University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

# Votemind: Using Sharemind for Privacy and Verifiability in Internet Voting

A special problem in partial fulfillment
of the requirements for the degree of
**Bachelor of Science in Computer Science**

Submitted by:

Ruahden F. Dang-awan

June 2016

Permission is given for the following people to have access to this SP:

| | |
|---|---|
| Available to the general public | Yes |
| Available only after consultation with author/SP adviser | No |
| Available only to those bound by confidentiality agreement | No |

## ACCEPTANCE SHEET

The Special Problem entitled "Votemind: Using Sharemind for Privacy and Verifiability in Internet Voting" prepared and submitted by Ruahden F. Dang-awan in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

**Richard Bryann L. Chua, M.Sc.**
Adviser

**EXAMINERS:**

| | **Approved** | **Disapproved** |
|---|---|---|
| 1. Gregorio B. Baes, Ph.D. (*candidate*) | | |
| 2. Avegail D. Carpio, M.Sc. | | |
| 3. Perlita E. Gasmen, M.Sc. (*candidate*) | | |
| 4. Marvin John C. Ignacio, M.Sc. (*cand.*) | | |
| 5. Ma. Sheila A. Magboo, M.Sc. | | |
| 6. Vincent Peter C. Magboo, M.D., M.Sc. | | |

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

**Ma. Sheila A. Magboo, M.Sc.**
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences
and Mathematics

**Marcelina B. Lirazan, Ph.D.**
Chair
Department of Physical Sciences
and Mathematics

**Leonardo R. Estacio Jr., Ph.D.**
Dean
College of Arts and Sciences

## Abstract

Electronic voting (e-voting) has been replacing traditional voting because it can automate the counting process of ballots. A type of e-voting – Internet voting – enables the voter to vote from home which increases voter turnout. In order for an I-voting system to be considered secure, it must satisfy both privacy and verifiability. A problem with most I-voting systems is that it usually trusts a third party to store and compute the votes, which risks the privacy and verifiabilty of the votes if that party becomes malicious. Secure Multiparty Computation (SMC) protocols can be implemented to meet these requirements by removing the need of trusting a third party. It achieves this by using multiple parties to do the storing and computing. Sharemind is a framework for SMC that can be used to create a secure I-voting system. This was done in Piscos' work, but her system still trusts in an intermediary web server which must be trusted. Talviste proposed an architecture that eliminates the need for the intermediary web server. This architecture uses a Javascript library that can split the data at the source. With the use of Sharemind 3 and the Javascript library, we created Votemind, which is an improvement of the Internet voting system of Piscos.

*Keywords*: Internet voting, Privacy, Verifiability, Secure Multiparty Computation, Sharemind

# Contents

# List of Figures

# List of Tables

# I. Introduction

## A. Background of the Study

Electronic voting (e-voting), in the past decade, has been evolving to replace the traditional voting system. It is used because it automates the counting process of ballots. In the case of Internet voting (I-voting), a type of e-voting, it allows the voter to vote from their respective homes which increases voter turnout. An I-voting system has the same requirements as traditional voting, and for it to be considered secure, it must fulfill both privacy and verifiability of the votes. [2, 3] I-voting has been used in legally binding elections in Estonia [2, 4], Norway [2, 5], and Australia [6]. It has also been used in a non-legally binding election in Université catholique de Louvain (UCL) [7].

I-voting systems were used in Estonia and UCL, but these systems are not fully secure from malicious attacks. One of the main problems of the Estonian and UCL I-voting system is that it trusts a third party to store and compute the votes. When this party becomes corrupted, it can jeopardize the privacy and verifiability of the votes.

A protocol that can be implemented to remove the need to trust a third party is Secure Multiparty Computing (SMC). SMC outputs secret shared results from secret shared data inputs. [8] It allows joint computations and keeps the privacy of each parties' individual infomation. [9, 10] An I-voting system, by Piscos [11], was able to apply SMC protocols with the use of a framework called Sharemind.

## B. Statement of the Problem

E-voting is now being used to replace tradional voting in other countries because it can count ballots faster and more accurately than manual counting. I-voting usually trusts a web server do computations on received encrypted votes. [4, 7] Trusting a third party or web server to collect and compute would be risky because a if that server becomes corrupt or malicious it could expose or change the votes. In order to remove the need to trust a third party, SMC protocols can be implemented. SMC can guarantee the correctness of the election results while preserving the privacy of the voters. [9, 10]

Piscos' I-voting system [11] implements SMC protocols to compute for the results. In her system, the cast ballot is sent by the browser to an intermediary server, which does secret-sharing of the cast ballot to the Sharemind miners. The problem with this system is there is a need to trust the intermediary server. This trust on an intermediary server can be eliminated if the secret-sharing to the Sharemind servers are to be performed in the browser instead. In the Sharemind version she used, Sharemind 2, there was no available technology that could be used to split the ballots into secret shares from the client's browser and send it to 3 Sharemind miners.

An architecture proposed by Talviste [1] can help solve the problem of trusting a web server with the use of a Javascript controller library for Sharemind. This Javascript controller is used to communicate with the 3 miners. However, there is a problem in using the Javascript library to communicate because modern browsers follow the Same Origin Policy. It is a policy that restricts Javascript applications from connecting to other web pages other than the web page origin domain. This means that the Javascript controller is prohibited from connecting to the 3 independent miners which are most likely on different hosts. Talviste proposed two solutions to this problem: the script tag workaround technique and HTML5 cross-document messaging API. [8] If the client's browser supports HTML5 cross-document messaging, then its API can be used. This allows communication between documents without any restriction on their domains using iframes. On the other hand, if the client's browser does not support HTML5 cross-document messaging, the script tag workaround technique is used. This technique uses HTML resource tags which are not affected by the Same Origin Policy. By using one of these methods, it enables the client's browser to split the shares at the source using the Javascript controller and send it to 3 different servers. Therefore, eliminating the need for an intermediary web server to split the shares.

## C. Objectives of the Study

To create a web-based I-voting system that uses Sharemind 3 which has the following user roles and their respective functionalities:

1. Allows a voter to
    (a) Cast a vote in an election
    (b) View the the encrypted format of his vote
    (c) Notify a voting official
2. Allows a voting official to
    (a) Setup an election
        i. Manage positions
        ii. Manage candidates
        iii. Manage parties
        iv. Manage election period
    (b) Setup official voters' list
    (c) Tally the results
    (d) Receive notifications
3. Allows a public user to
    (a) View election results
    (b) Download election results
    (c) View bulletin board of votes
    (d) Download bulletin board of votes
    (e) Notify a voting official

4. Allows an administrator to

    (a) Manage voting officials list

## D. Significance of the Project

An I-voting system can increase voter turnout because it is accessible from any place where there is a computer connected to the Internet. It speeds up the counting process and increases the accuracy because the counting will be automated. In addition, it costs less as compared to other voting schemes which require setting up stations and printing ballots. However, an I-voting system must fulfill the requirements of privacy and verifiability.

With the application of SMC protocols on an I-voting system, votes and voters are able to remain anonymous because computation can be performed on partial and encrypted data. It also allows the voters to verify their votes through the use of a web public bulletin board which allows the voters to see if their votes were included in the counted ballots. The system's security is enhanced by the use of 3 separate Sharemind miners for storing and counting. This decreases the chances of a malicious miner being successful on decoding the vote stored on its database or altering it because each miner only holds partial and encrypted data.

This I-voting system is an improvement of Piscos' I-voting system because it encrypts and splits the votes at the browser and not on an intermediary web server. It also uses Sharemind 3 which is a faster version of Sharemind 2 – the version used in Piscos' system.

## E. Scope and Limitations

1. This project handles only structual organization voting.

2. This project does not handle voter registration.

3. This project does not contain the analysis of used algorithms.

4. This project uses Sharemind version 3.

5. This project does not handle external attacks.

## F. Assumptions

1. Miners are online and working.

2. Miners will not collaborate with each other for conspiracy.

3. Miners will follow the given protocol.

4. The voter is not coerced when voting.

5. There is an existing list of voters.

6. There are no nuisance candidates.

7. The voter is not malicious.

# II. Review of Related Literature

In the past decade, e-voting has been used in some elections because it can automate counting when there are too many manual ballots to count. [2] Internet voting, a type of e-voting, allows the voter to vote from home, avoiding long travels. The central properties of modern I-voting systems are privacy and verifiability. Privacy means that the voter's vote information is not leaked by the system to anyone. Verifiability means that the voters can make sure their votes were actually counted and that the results are legitimate. [12] Therefore, in order to pull off I-voting, there is a need to have a secure, reliable, and trustworthy I-voting system that fulfills these central properties.

In some countries, I-voting is used for legally binding elections while other countries have banned I-voting for at least some time. But in non-legally binding elections, I-voting is likely to be used in most developed countries worldwide. [2]

An example of I-voting being used in legally binding elections is in Estonia's Internet voting system. Estonia was the first country to use Internet voting nationally with 30% of its ballots cast online. When voters cast their vote, their votes are encypted and they sign the corresponding ballot using their National ID. The system does not have end-to-end verifiability and it implicitly trusts the integrity of voters' computers, server components, and the election staff. Springall et. al. [4] discovered that the Estonian I-voting system has unsafe architectural inhibitions and procedural gaps that risks election's integrity. They saw that there are many ways attackers could successfully infiltrate the Estonian I-voting system and recommended that the I-voting sytem be terminated.

A generally more secure and distributed system that was used in legally binding elections is the Norweigan I-voting system. It was launched in some of its municipalites in 2011. [5] Its architecture defends against two main threats: compromised computers and coercion in I-voting. It uses two independent channels (postal mail and SMS) to guard against compromised voter's computer. Before voting, the voter gets a post mail containing a list of all candidates and his verification code. For the voter to vote on the application, he must first login with his National ID to be authenticated. Once they have voted, the system sends a code to the voters via SMS. The code can be used to verify that their ballot belongs to the box. [2] When the voting is over, the hashes of every ballot is disclosed by the auditor so that voters can check if their ballots were included in the count. Unfortunately, in 2014, Norway ended the trials of their I-voting system in national and local elections. According to their government, the reason for ending it was because the democratic processes could be threatened by voters' fears about their votes becoming known to the public. [13]

vVote [6] is an example of an I-voting system that ran successfully in the state election in Victoria, Australia. It was used in November 2012, taking a total of 1,121 votes. It is based on the Prét á Voter cryptographic end-to-end verifiable voting system. Before the vVote voting process starts, Prét á Voter ballot forms are printed. These are the ballots that will be filled up by the voter during the election. When the election is ongoing, a print-on-demand printer will print a human-readable candidate list in random order with its serial number that links it to its encrypted representation on the public Web Bulletin Board (WBB). The voter will then use an Electronic Ballot Marker (EBM) to assist her in filling up the Prét á Voter ballot according to her printed candidate list. After voting, the voter leaves the booth with the candidate list folded and the printed preference list (receipt) from the EBM. Before leaving the polling place, the voter must verify if the data signed by the WBB is the same as the data printed on the receipt using a smartphone application. Then, the voter shreds the list of candidates to prevent her from having proof of how she voted. When the public WBB is posted, the voter can use her receipt to check if the WBB recorded her vote properly. Anyone can then verify if the votes are correctly mixed and decrypted. This process provides cast-as-intended verification, recorded-as-cast verification, and an output list of decrypted recorded votes, with a universally verifiable proof of the decryption.

I-voting, however, can be used in both legally and non-legally binding elections.

An example of its use in non legally binding elections is when Helios, a web-based voting system, was used by the Université catholique de Louvain (UCL) when it elected its President in March 2009. [7] Helios was one of the first web-based open-audit voting systems. Distributed decryption is used in Helios to ensure that various trustees are needed for decryption. Voters are authenticated by their email address and an election-specific password which was given upon registration. A browser-based ballot encryption program encrypts the votes using LiveConnect to access the Java Virtual Machine from JavaScript. Ballots are submitted in behalf of the voters by a separate, trusted server. The Benaloh cast-or-audit voting protocol ensures ballot casting which is implemented in part by the ballot verifier. The votes are then shuffled by a Benaloh mixnet, and the election server decrypts the shuffled votes and tallies the results using UCL's own, high-speed, offline tallier and verifier. To ensure correctness of the election process, an election verification program downloads all encrypted votes, shuffled votes, decrypted ballots and proofs. When the voting was done, they froze the web bulletin board containing all the voter IDs and vote hashes. A signed receipt was published for each vote, together with a signed version of the full bulletin board content. Then, to attain universal verifiability, all the data and operations needed to audit the election were disclosed.

Helios has unconditional integrity but compromisable privacy. [14] Helios trusts in only one server, thus if this server is corrupted, privacy can be compromised. The Estonian I-voting system also trusts in a single server for privacy. Its ballot secrecy relies on the counting server not to expose data about the correspondence of the encrypted and decrypted votes. [4] The Norwegian I-voting system also relies on a single electronic ballot box to store their votes. [15] Generally, this shows that the common problem among many existing I-voting systems is that they trust a single server to store important and private data.

A secure protocol can be implemented to prevent the need for trusting a single third party which can improve the privacy of the votes. Secure multiparty computation (SMC) is a secure protocol that fulfill these requirements (privacy and verifiability) of I-voting without trusting a single third party. It can be described as a problem where joint computations are done but no party knows the individual data of other parties while still ensuring correctness of the computation and privacy of the parties' information. [9, 10]

SMC protocols can be used in different fields like banking and surverying [16]. Commonly, it is used for statistical computations on data that cannot be exposed. It can be implemented in different ways, which means that it can be applied in different architectures. Pathak et al. [16] proposed a novel architecture for SMC that can be used in statistical computations. Tri-TTP (Trusted Third Party) architecture is the foundation of the system. Each party included in the system will create virtual parties and share the data among them. The data has two parts, encrypted data and modifier tokens. Encrypted data is sent to the Data Accumulator (DA) and the modifier tokens are sent to the Modifier Token Accumulator (MTA). The computation TTP then receives both of these data from the two sources and computes the results which are announced publicly. According to the security analysis Pathak et al. [16] made, the use of the DA and MTA to store data has improved the security of the system.

Mishra et al. [9] proposed another novel SMC protocol for statistical computations. It is an enhanced version of [16] which used Virtual Parties. On [16], the number of nodes is directly proportional to the complexity of the system so an increase in the number of nodes creates negative effect on the performance of system. So in [9], they proposed a system that is as secure but less complex. In order to do this, they used a token ring network where computation will be executed. The ring will have alloted positions for each party and one position for TTP. Each party will create virtual parties and disseminate their encrypted data among the VP's. Using the encrypted data, the TTP will do the computations and publish the results. According to Mishra et al. [9], the general security of the system increases and the complexity decreases with the usage of the token ring network.

A generic system for SMC is FairplayMP. [17] FairplayMP is an extension of Fairplay which only supports secures computation by two parties so they extended it to the multi-party case. The system has two components: the SFDL 2.0 compiler and the Fairplay MP cryptographic engine - which is used by the SMC part. The SFDL 2.0 compiler is an enhanced version of SFDL 1.0, where the main feature is multi-player capabilities. Two steps are required by the program to execute SMC of a Boolean circuit: A garbled input and garbled circuit. The garbled circuit is made from the boolean circuit according to the Beaver, Micali and Rogoway (BMR) protocol. [18] The players then evaluate the garbled circuit which provides different outputs per player. The secure computations go through 8 communication rounds that use an enhanced version of BMR protocol. A weakness of FairplayMP is that it only uses protocols which safeguard against semi-honest attackers and not malicious attackers.

An example of a system for SMC that protects against malicious adversaries is Canon-MPC (CAsual NON-interactive secure Multi-Party Computation) [19]. It is "casual" because users only need a web browser and are not required to download or install any software. It is "non-interactive" in the sense that users do not need to be online simultaneously. The system is based on one-pass MPC protocols of Halevi et al. [20] and Native Client technology. Registration of users is only done once by providing an email address and public key. After registration, users can start a secure computation and choose the other users to be involved. It is implemented by a web site and client software which runs native code in a sandbox that is implemented by Native Client technology. The client side calls the Native Client using Pepper API implemented in Javascript. The server side stores its data in a MySQL server and was built using PHP. All computation is executed in client side and the server receives the results. Though the system is safe from malicious attackers, it does not use Transport Layer Security (TLS) encryption [21] – a cryptographic protocol that provides secure communication between server and client over a computer network – and secure coding which makes it vulnerable to SQL injections and other attacks.

A framework that was made for SMC and privacy-preserving computations is Sharemind [22]. It was designed for efficient and easy programming on a platform for developing and testing different privacy-preserving algorithms. Sharemind includes a computation runtime environment and a programming library so that users can create safe multi-party protocols without knowing the implementation details. Users can construct complex tasks because the framework provides efficient protocols for basic mathematical operations which are universally composable. Sharemind uses nonorthodox secret sharing and share computing protocols to optimize effiency. Three dedicated miner parties are used to do the computations. They are less susceptible to corruption from outside attacks and can tolerate semi-honest corruption of one miner. When data is sent by the users, information and rights are split between miners by using secret sharing. Each miner has a local database and local stack that contains shared values distributed by an additive secret sharing protocol.

Piscos [11] used Sharemind as the SMC framework for her Internet voting system. She developed a web-based application that provides protection for the voter's privacy and anonymity. Sharemind was used to handle SMC processes such as storing votes and computation of results. The system allowed the voters to cast a vote securely and the voters could verify their votes by viewing the bulletin board which contained the results of the election. By using the secret-shared database of Sharemind, stored votes were considered secure. The system generated unique hashes per voter for each candidate. The voter could then verify his/her votes by checking the bulletin board if the hashes he/she had was the same with what was posted. The correctness of the results were also guaranteed if the databases of the miners were not corrupted. The main issue encountered by Piscos was the slow running time when performing SMC in Sharemind. To enhance the performance of the system, Piscos [11] recommended that the system be implemented in the newer version of Sharemind, Sharemind 3. Another issue found in the system is that data from the user first had to go through the web server before the Sharemind server. Apache Tomcat was used as the systems web server, with MySQL as its database server. If the web server is attacked, data may be compromised before it goes to the safe Sharemind database server. For example, if an attacker hacks into the Apache Tomcat web server, he could gain access to the incoming votes in plain text form (since the votes are still going to be encrypted). Since the data is whole and unencrypted, the attacker could easily alter or erase the votes before it gets encrypted and sent to the Sharemind server. A problem also lies on how the verifiability of the system was implemented. The hashes used for verification were stored in the Apache Tomcat's MySQL database server and not in Sharemind's secure database. Because the data is in a vulnerable database, attackers may be able to change the hashes which could cover up anomalies in altered votes. Attackers can also make it seem like the system recorded the wrong vote by changing the hash. (Figure 1)

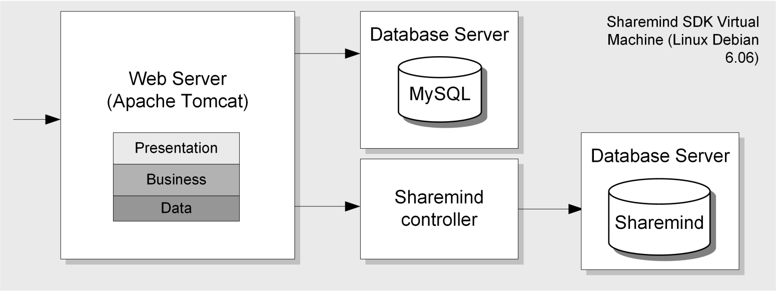


Figure 1: System Architecture of Piscos' I-voting system

To the prevent risk of having the web server attacked and exposing the private data, Talviste [1] proposed an system architecture for Sharemind (Figure 2).

This architecture solves the problem because there is no intermediary web server between the user and the miners. Data is directly communicated. This can be implemented by either using JavaScript or Flex. The user has a unique session identifier that is saved in the web server's database. When the user has finished filling in forms and sends the data, the data split into shares which is distributed to the three miners. Data analysis algorithms can then be run on each miner using

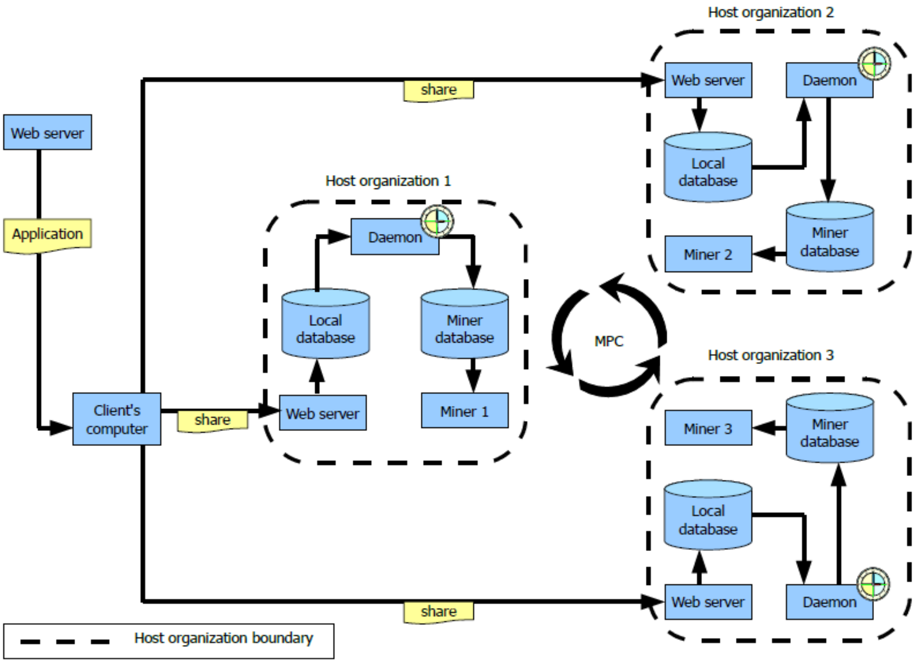


Figure 2: Data movement in Talviste's proposed architecture

their internal databases and SMC protocols.

# III. Theoretical Framework

## A. Voting Scheme

A voting scheme is a method for conducting an election. It is defined as a protocol which can receive votes and produce results based on the tally of the votes. [23] The most commonly used voting scheme is the traditional voting scheme. Examples of this scheme are the use of paper ballots, lever systems, or punch-card ballots. The electronic voting scheme (e-voting) is a scheme that is slowly replacing traditional schemes but it is still growing and needs more research and development to make it secure. E-voting uses electronic devices to record and tally votes. It can be classified into 3 types: [24]

1. Remote - The user has control over the voting client and operating environment. If internet is used then it is called internet voting or I-voting.
2. Kiosk - The voting environment is not managed by the election officials but the voting client is installed by them.
3. Poll-site - Both the voting client and operating environment is managed by the election officials.

In order for voting schemes to be considered secure, they must meet the following requirements: [2, 23]

1. Privacy - the voter's votes must be known to the voter only. It should be difficult or impossible for others to know who she voted.
2. End-to-end verifiability
   (a) Individual verifiability - the voter can check if her ballot appeared in the ballot box.
   (b) Universal verifiability - any third party should be able to check if the results correspond to the ballots on the box.

(c) Eligibility verifiability - any third party should be able to check if only the legitimate voters voted.

The following are extended requirements; [3, 23] they focus more on reliability, practicality and acceptability of the voting scheme:

1. Robustness - the scheme should have a mechanism that can correct or contain an error.

2. Uncoercability - the scheme should allow the voter to vote without being coerced or forced.

3. Efficiency - the scheme should be able produce an accurate tally in an appropriate amount of time.

4. Flexibility - the scheme should be able to support different ballots, different precincts and different voting styles.

5. Certifiability - the scheme must be able to provide a way for it to be tested by Independent Testing Authorities.

6. Scalability - the scheme should be able to handle any number of votes efficiently.

7. Usability - the scheme should be convenient for people to use.

8. Accessibility - the scheme should be accessible to all qualified voters.

## B. Secure Multiparty Computation

Secure Multiparty Computation (SMC) is a term used to describe the problem of computing any collaborative function securely. [25] SMC protocols are cryptographic techniques used to compute joint results while keeping a participating party's individual information unknown to the other parties. If the parties want to compute for the function $y = f(x_1, x_2, ..., x_n)$, then a SMC protocol can reveal and compute $y$ correctly without revealing the $x$'s.

There are two basic types of adversaries considered in SMC. [25] The first type is the honest-but-curious (HBC) adversary model (also called semi-honest or passive adversaries). In an HBC adversary model, the adversaries will follow the protocol but will try to learn additional information about the other inputs. The second type is the malicious or active adversary model. These adversaries will try to learn more about the other parties by deviating from the protocol. They will also try to change the result of the protocol and abort the protocol after they have learned additional information about other parties. Fortunately, even if a model includes passive or malicious adversaries, a function can still be computed securely as long as the majority of the participating parties are honest. [26]

Given the model that the majority of participants are honest, one way of doing SMC is by first having all the parties participate in a secure protocol. (Figure 3)

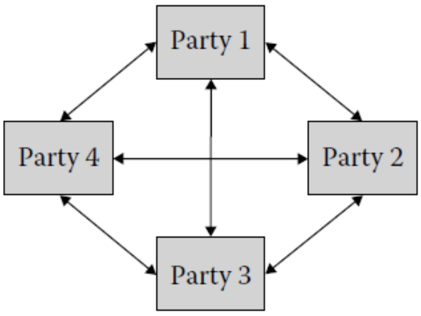


Figure 3: Example of an architecture where all parties participate [25]

This architecture allows for robust protocols that can still compute for the result even when other parties abort the protocol after the starting rounds. But the disadvantage of this protocol is that it requires the parties to be all online simultaneously. Another downside is that it is costly for a large number of parties. In other words, it only works well for a small number of parties. [25]

A representative-based architecture that increases the scalability of the participating parties was formalized by Damgard et. al. in [27]. This architecture splits the participating parties into 3 groups that are not necessarily mutually exclusive: input servers, computation servers, and output servers. The input servers share their input to the computation servers so that not one server has the whole data.

The computation servers then compute for partial outputs using the partial inputs. When the computations are done, the shared results are sent to the output servers that learns the complete result. The advantage of this architecture is that the input servers need not be all online at the same time because when they have sent their inputs, they can already log out. Another benefit of this architecture is that the computation servers are smaller in number compared to the input servers, which makes it more scalable. The disadvantage of this architecture is that it requires some level of trust on an outside party. (e.g. the computational parties are trusted not to collude) [25] (Figure 4)

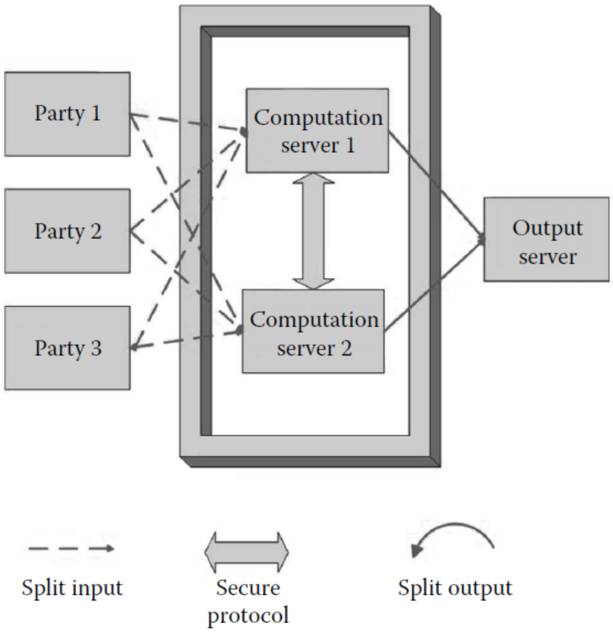


Figure 4: Data movement in the example of representative-based architecture [25]

## C. Sharemind

Sharemind is a framework for SMC protocols that is implemented using 3 servers which are called miners. It uses additive secret sharing scheme over the ring of 32-bit integers to gain direct computational advantages and to simplify many share computing protocols. Sharemind implemented its own multiplication protocol because the standard multiplication techniques are not applicable for the ring of 32-bit integers. Data is split into shares at the source and each share is sent to the corresponding miner. Once the miner gets the data needed for the computation, they do not need further interaction with the participating parties. The miners do all secure computations and output the collective result. [22]

The Sharemind framework was made for developing and testing different privacy-preserving algorithms in an efficient and easily programmable platform. It consists of a computation runtime environment and a programming library for making applications. Hence, SMC protocols can be developed without the knowing the implementation details. [22] (Figure 5)

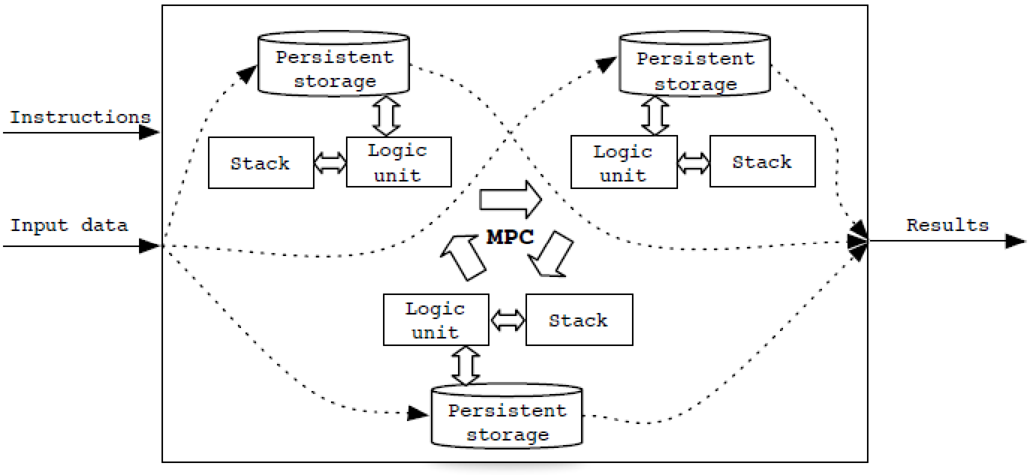


Figure 5: High-level description of Sharemind: 3 miners are given input data and instructions which use SMC protocols. When all computations are done, the result is revealed.

Each miner in Sharemind has a local database and a local stack which makes it like a secret storage for shared inputs. The miners use this stored data to do privacy-preserving operations. Sharemind can deal with only one semi-honest miner, which means that no data is leaked unless 2 miners collude. [22]

The very first full-featured version of Sharemind was released in 2008. In 2011, Sharemind 2 was released with development tools. [28] In Sharemind 2, additive 3-party secret sharing scheme was supported. The developer tool SecreC was added and it had two data visibility types: public and private, wherein private values can only become public through declassify. In 2012, Sharemind 3 was released. It had various new secure computation schemes, new underlying virtual machine, and a complete rewrite of SecreC with new features. Some of the features of the new SecreC are more primitive data types, arbitrary dimensional arrays, protection domain polymorphism and simple module system. [29]

Sharemind uses the designed developer tool named SecreC programming language to simplify the creation of secure Sharemind applications. SecreC can be used to separate private and public data. In order for this to be done during Sharemind runtime, Sharemind has a compiler that translates SecreC programs into Sharemind assembly language and a built-in assembly intepreter. [30]

In order to create an application using Sharemind, 3 steps are required. Firstly, there must be 3 independent parties that will host the 3 miners. Each host party must set up a server, install Sharemind miner software and be configured to be able to communicate with the other miners. Secondly, the data-mining applications coded in SecreC must be compiled into Sharemind assembly and be given to each of the miners so that it can be executed by the Sharemind virtual machine. Thirdly, the Sharemind controller library is used to create end-user applications. [1]

On Talviste's real deployment scenario, each Sharemind miner looked like this: (Figure 6)

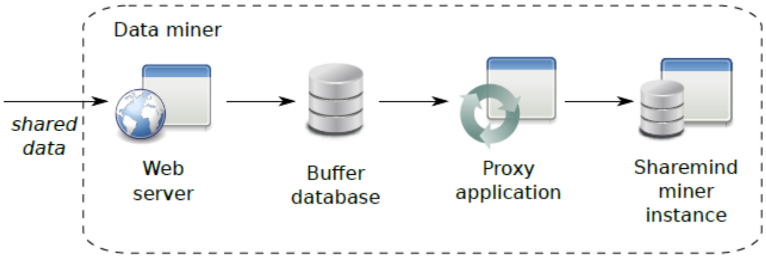


Figure 6: Components of a Sharemind miner in a real deployment scenario [1]

Each Sharemind miner is coupled with a web server (Apache), an SQL database engine (MySQL) and a simple web application. The web application receives the shares and identifies the source. The data received by the application is secret shared at the source before being sent to each of the miners to ensure privacy. In Talviste's real deployment scenario of Sharemind [1], the online data entry form used to pass information to the miners was made using HTML with a Javascript library that managed secret sharing. In Sharemind 3, Node.js can be used as the miner's web server that communicates with it directly, which means that there is no longer a need for a buffer database and a proxy application per miner.[1]

[1] According to Reimo Rebane, a Sharemind developer.

# IV. Design and Implementation

## A. Use Cases

There are 4 users in the system: voter, voting official, public user and administrator. (Figure 7)

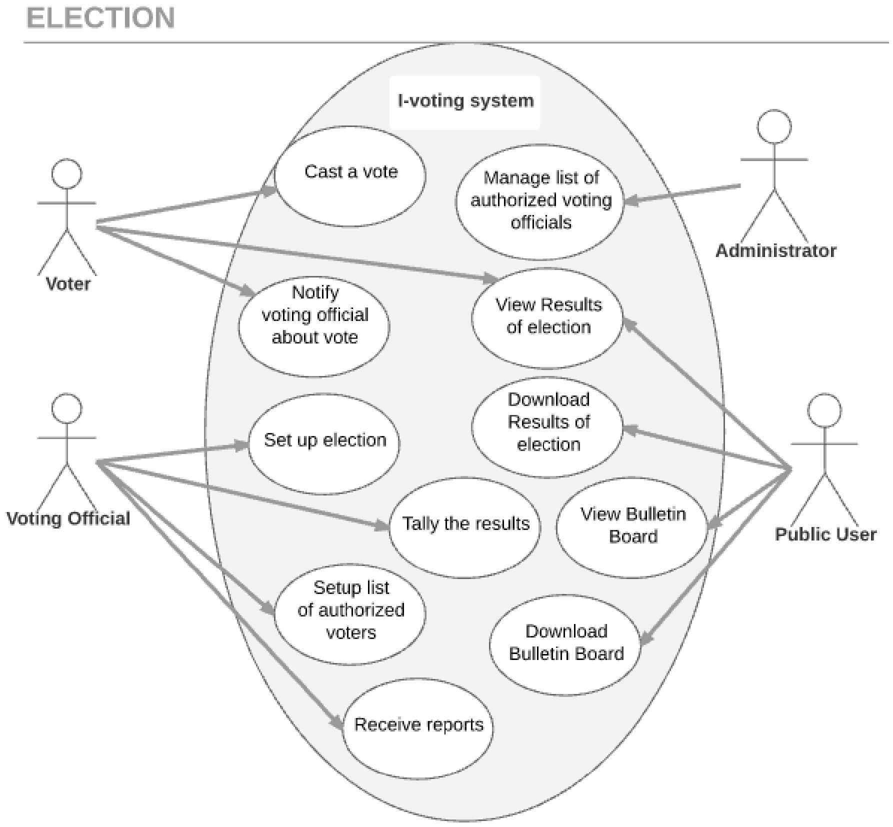


Figure 7: I-voting system's Top Level Use Case Diagram

Before the election, the administrator uploads a list of authorized voting officials. The voting officials then set up the elections, upload a list of authorized voters and start the elections.

During the election, the voter can log in to the system by using login credentials given by the voting officials via email. The voter can then fill up the ballot and submit it. When the voter is done submitting the electronic ballot, the system shows the voter the receipt which the voter can save. The receipt contains the hash values the votes.

When the election is stopped by the voting officials, the results are automatically computed and posted. The voters can then view and verify if their votes were counted by checking if their receipt and hash values are included in the published public web bulletin board. If there is any discrepancy in the votes, the voter can report this event to the voting officials. Any user can view and download the results of the election and the bulletin board of votes.

## B. Database Design

The I-voting system has 2 databases:

1. Database stored in a webserver - This database that can be altered by the voting officials and administrators. It contains the list of authorized users and the information needed in the ballot. (Figure 8)

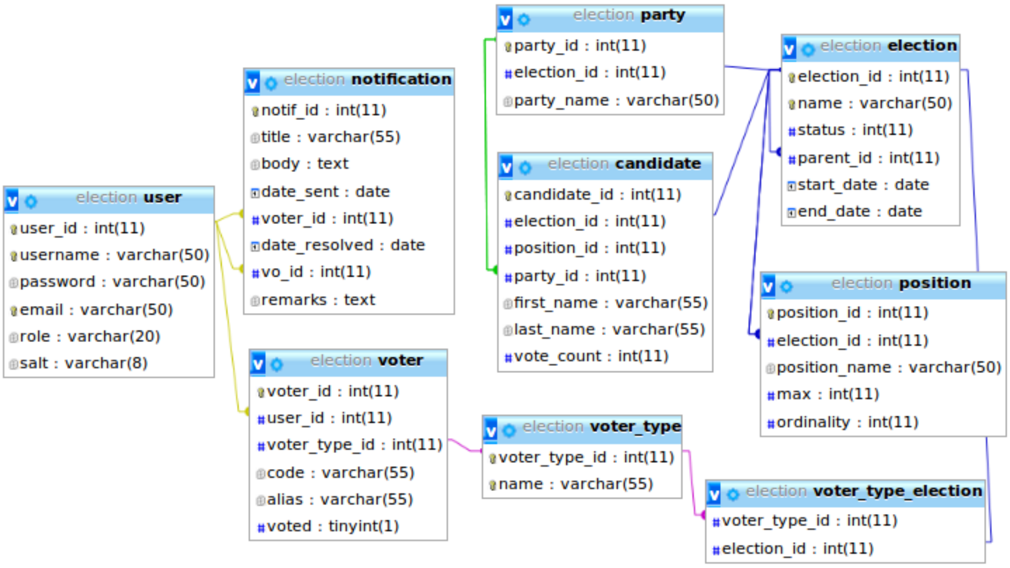
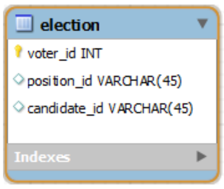


Figure 8: Database stored in webserver for the ballot and users

2. Secret-shared Sharemind database - This database contains the votes. Each vote has: voter_id – a unique identifier for the voter, position_id – defines what position is being voted for, and candidate_id – the candidate for the specified position. (Figure 9) Each election is allocated one table in the Sharemind database, when that election is deleted, its corresponding table is deleted too.



election

voter_id INT

position_id VARCHAR(45)

candidate_id VARCHAR(45)

Figure 9: Secret-shared Sharemind database

## C. Data Dictionary

| Column | Type | Description |
|---|---|---|
| **user_id** | int(11) | ID of the user |
| username | varchar(50) | Username of the user |
| password | varchar(50) | Hashed password of the user |
| email | varchar(55) | Email of the user |
| role | varchar(55) | Role of the user, may be:<br>voter, voting official or admin |
| salt | varchar(8) | Random string added to user's password for security |

Table 1: user table

| Column | Type | Description |
|---|---|---|
| **voter_id** | int(11) | ID of the voter |
| user_id | int(11) | User ID of the voter |
| voter_type_id | int(11) | Votertype ID of the voter |
| code | varchar(55) | Verification code of voter's ballot |
| alias | varchar(55) | Alias of voter posted at the bulletin board |
| voted | tinyint(1) | Boolean value that states if the voter has voted |

Table 2: voter table

| Column | Type | Description |
|---|---|---|
| **voter_type_id** | int(11) | ID of the voter type |
| name | varchar(55) | Name of voter type |

Table 3: voter_type table

| Column | Type | Description |
|---|---|---|
| **election_id** | int(11) | ID of the election |
| name | varchar(50) | Name of election |
| status | int(11) | A number that represents status of election (-2=not ready, -1=ready, 0=running, 1=stopped & posted) |
| parent_id | int(11) | ID of parent election |
| start_date | date | Start date of election |
| end_date | date | End date of election |

Table 4: election table

| Column | Type | Description |
|---|---|---|
| voter_type_id | int(11) | ID of voter type |
| election_id | int(11) | ID of election |

Table 5: voter_type_election table

| Column | Type | Description |
|---|---|---|
| **position_id** | int(11) | ID of the position |
| election_id | int(11) | Election ID of position |
| position_name | varchar(55) | Name of position |
| max | int(11) | Maximum number of candidates for this position |
| ordinality | int(11) | Order in the ballot |

Table 6: position table

| Column | Type | Description |
|---|---|---|
| **party_id** | int(11) | ID of the party |
| election_id | int(11) | Election ID of party |
| party_name | varchar(55) | Name of party |

Table 7: party table

| Column | Type | Description |
|---|---|---|
| **candidate_id** | int(11) | ID of the candidate |
| election_id | int(11) | Election ID of candidate |
| position_id | int(11) | Position ID of candidate |
| first_name | varchar(55) | First name of candidate |
| last_name | varchar(55) | Last name of candidate |
| vote_count | int(11) | Number of votes of candidate (set after election) |

Table 8: candidate table

| Column | Type | Description |
|---|---|---|
| **notif_id** | int(11) | ID of notification |
| title | varchar(55) | Title of notification |
| body | text | Body of notification |
| date_sent | dat | Date when notification was sent |
| voter_id | int(11) | ID of voter who sent the notification |
| date_resolved | date | Date when notification was resolved |
| vo_id | int(11) | ID of voting official who resolved the case |
| remarks | text | Remarks on the notification when resolved |

Table 9: notification table

## D. System Architecture

Our system runs on the virtual machine provided by Cybernetica. The web application is stored on the Apache 2 webserver in the virtual machine. MySQL server is used, which acts as a storage for the auxillary data needed for the elections. The Sharemind servers (or miners) are configured to communicate with their corresponding Node.js servers.

The Node.js server is used to communicate to the miner it is assigned to. The client can communicate with the Node.js server by using any JavaScript-enabled browser. By using the JavaScript controller provided by Cybernetica, the voter can send requests from the browser to the Sharemind servers through the Node.js server. Upon receiving the request or data, the Sharemind servers run the required script which is written in SecreC and use their local independent databases for the operation.

## E. System Development

Votemind is implemented in PHP using the CodeIgniter framework. Since it uses CodeIgniter, the Model-View-Controller architecture is applied. Models are used to communicate with the MySQL database. Views are the pages that are seen by the user. Controllers are used to manage the back-end of the web application. It handles what can be viewed by the user and calls models when needed.

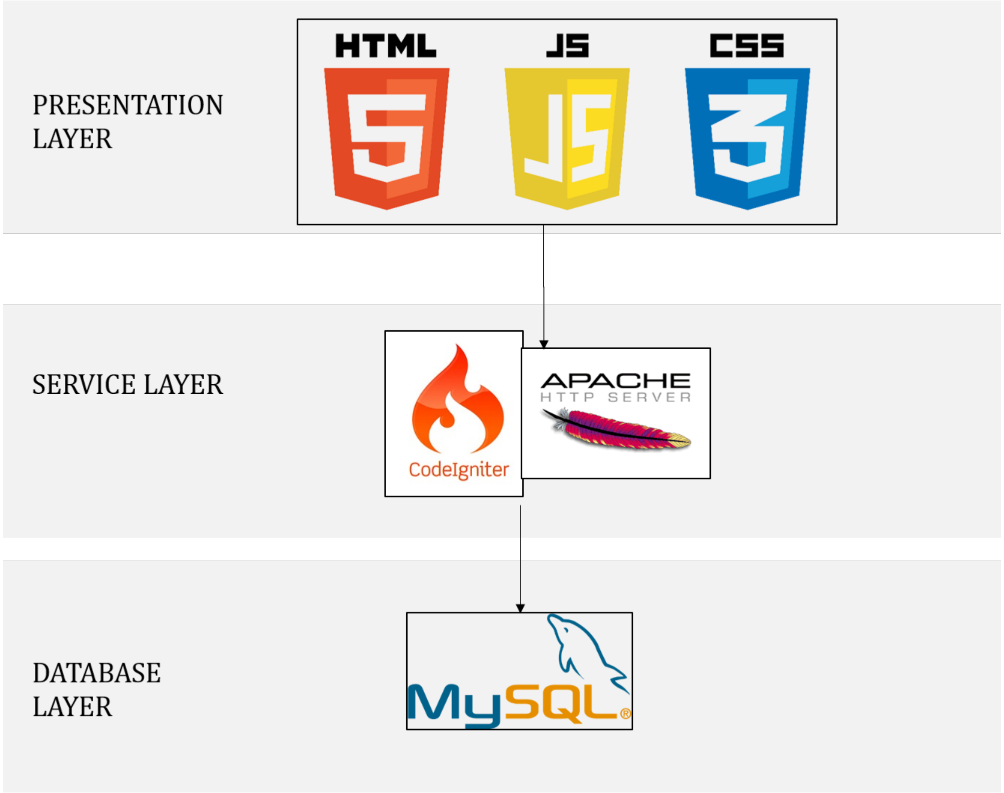


Figure 10: Votemind System Architecture

Votemind uses 2 different databases: MySQL and Sharemind's database. CodeIgniter is used to handle the communication of the webserver and MySQL database while the JavaScript controller given by Sharemind is used to communicate with Sharemind's database. Auxilliary data are stored in the MySQL database. These data include users, voters, and election components. The Sharemind database is used to store the votes of the users. In Sharemind's database, there is an alloted table for each election. Each table contains rows that has the voter id, position id and candidate id of the votes. Each column is stored in an encrypted 32-bit integer format.

When voters access the vote page, CodeIgniter sends the JavaScript controller to the client's browser along with the ballot. When a voter selects his/her candidates, the hash value of that candidate appears on the right side of the candidate name. This hash value is the SHA1 hash of the CRC32 value of the concatination of the voter id, position id, and candidate id. When the voter submits the ballot, the browser calls the JavaScript controller. The JavaScript controller encrypts

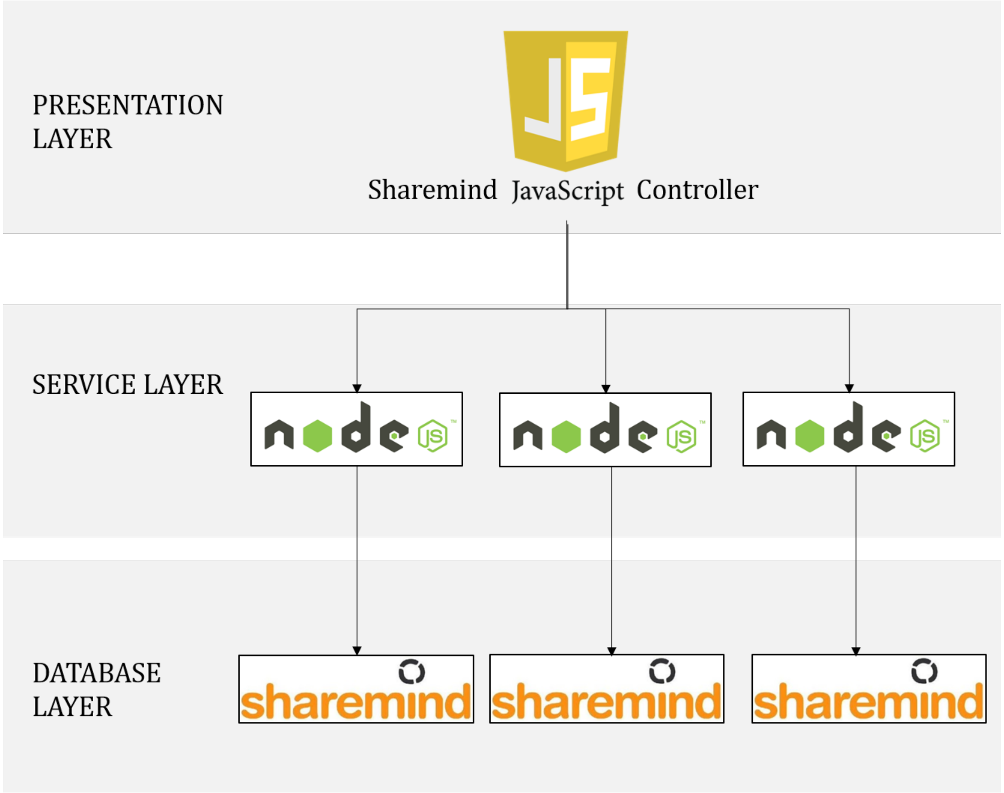


Figure 11: Sharemind Framework Architecture

and splits the votes at the browser and sends it to 3 different Node.js servers that are on top of the miners. These Node.js servers are independent from each other. When a Node.js server has received the data, it sends the data to the miner assigned to it, which may be located in another machine. The miner receives the data, runs the SecreC script needed, and stores the partial votes in encrypted format in its local database. When the votes are stored, the miners return the hash values of the concatination of the voter id, position id, and candidate id of each vote in CRC32 format to verify if the stored votes are the same with the votes sent from the browser. When the browser receives the hash values it hashes it again in SHA1 then checks if the hash values are the same with the values generated before. Afterwards, it informs the voter the statues of the verification and shows the receipt with the voter's alias, ballot code and the hash values. The voting process is done without the intervention the web server, the web server is only used to provide the JavaScript controller in the beginning of the process.

When an election is stopped, the voting official's browser calls the JavaScript controller to retrieve the results in the Sharemind database. The browser sends the election id and candidate ids of the candidates involved the election that is going to be stopped. The miners receive the ids and compute for the number of votes by running the SecreC script for that. After the computation, it returns the number of votes to the browser. When the browser receives the number of votes, it sends the results to the webserver to store it in the MySQL database for posting.

When public users access the bulletin board of a specific election, the browser calls the Javascript controller to request for the hashes in that election. Since each table is one election, Sharemind returns the hash values of the rows in the specified table/election. The format of the return values are still in CRC32, but before being shown to the user, it is hashed again using SHA1.

## F. Technical Architecture

The web application is saved in a Sharemind Virtual Machine provided by the developers of Sharemind. It is running on Debian OS. We have installed LAMPP structure on the virtual machine to be able to host the web application. The web application can be accessed by using HTML5, JavaScript enabled browsers.

The Sharemind miners are run in the Virtual Machine as well as the node.js servers.

Host OS Requirements

- Windows 7 and up, 64-bit or Ubuntu 14, 64-bit
- Oracle VM VirtualBox 5 and up
- RAM greater than or equal to 2GB
- Memory with at least 100GB

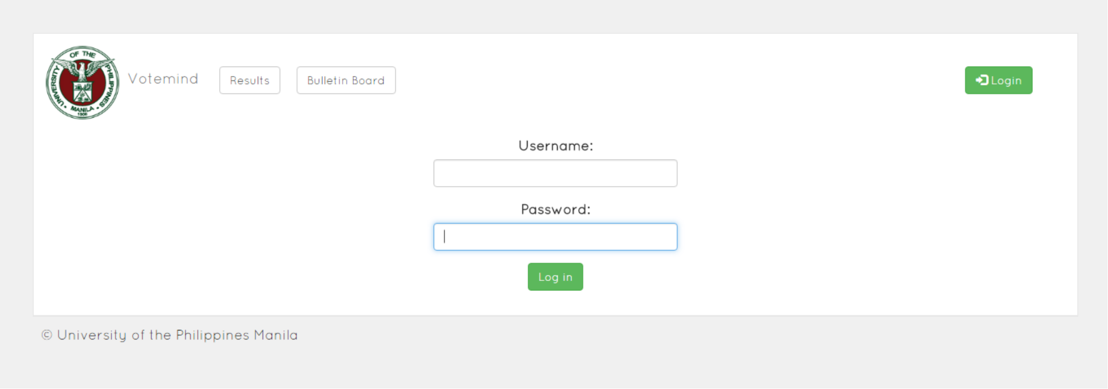
Virtual Machine OS Requirements

- Linux Debian 8.2.0 64-bit

- MySQL 5.5.x server
- Apache2 server
- Node.js server
- Sharemind 3 Academic Server SDK
- 1GB RAM
- PHP 5 server
- JavaScript-enabled browser

# V. Results

The following are the pages any user can access. The home page is the login page of the system.



Username:

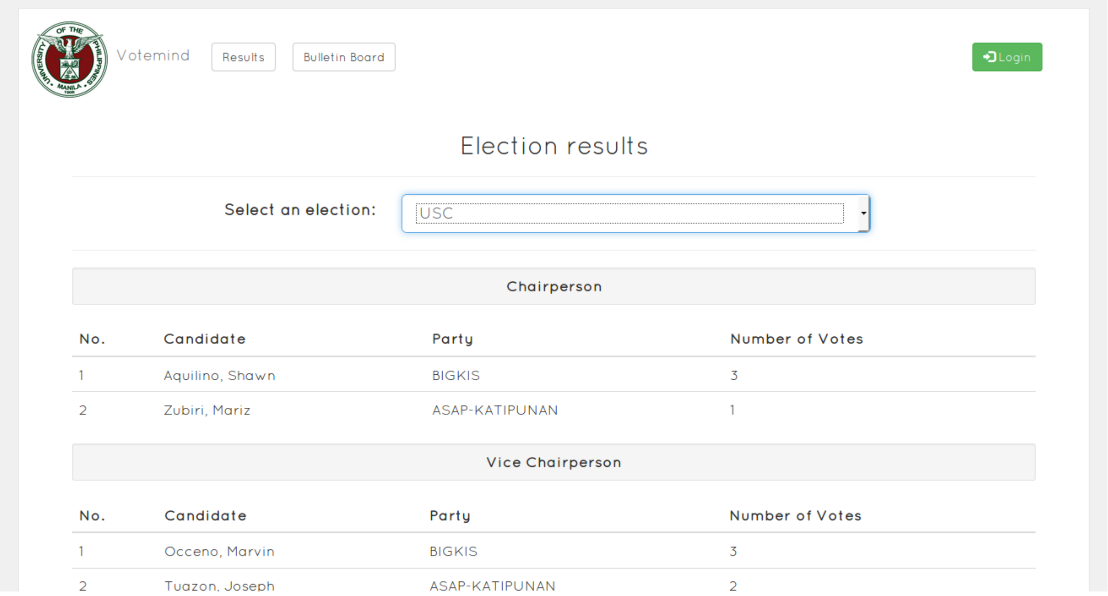
Password:

Log in

© University of the Philippines Manila

Figure 12: Login page

When an election is finished, the results are posted on the results page. This can be accessed by clicking the Results button on the top of the page. A public user can choose the election he/she wants to view by selecting the election name on the dropdown menu.



Election results

Select an election: USC

Chairperson

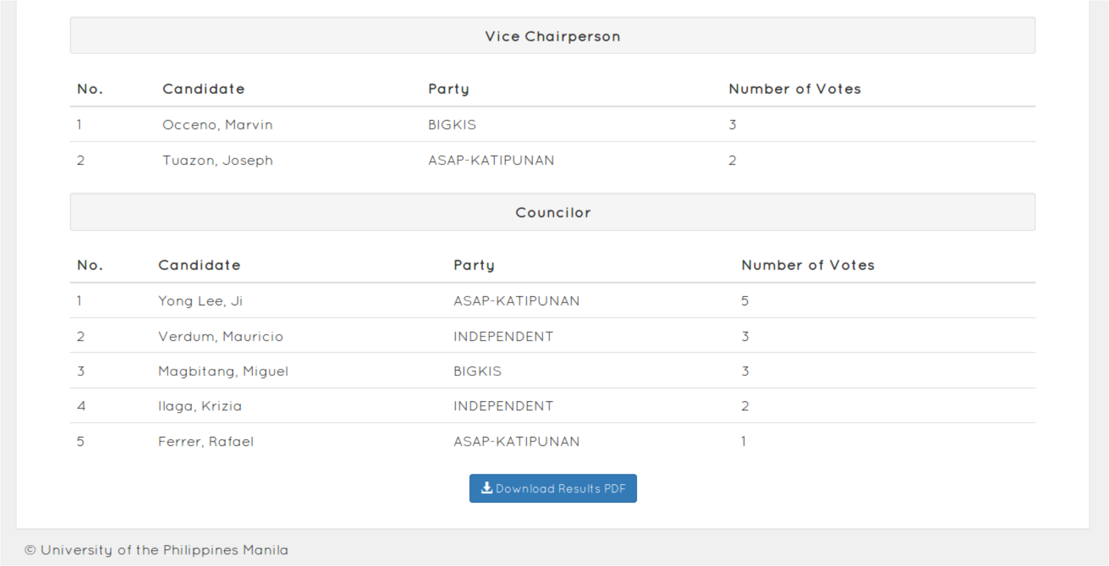
| No. | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Aquilino, Shawn | BIGKIS | 3 |
| 2 | Zubiri, Mariz | ASAP-KATIPUNAN | 1 |

Vice Chairperson

| No. | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Occeno, Marvin | BIGKIS | 3 |
| 2 | Tuazon, Joseph | ASAP-KATIPUNAN | 2 |

Figure 13: Results page

The public user can download a pdf copy of the results by clicking the "Download Results PDF" button located at the bottom of the page



Vice Chairperson

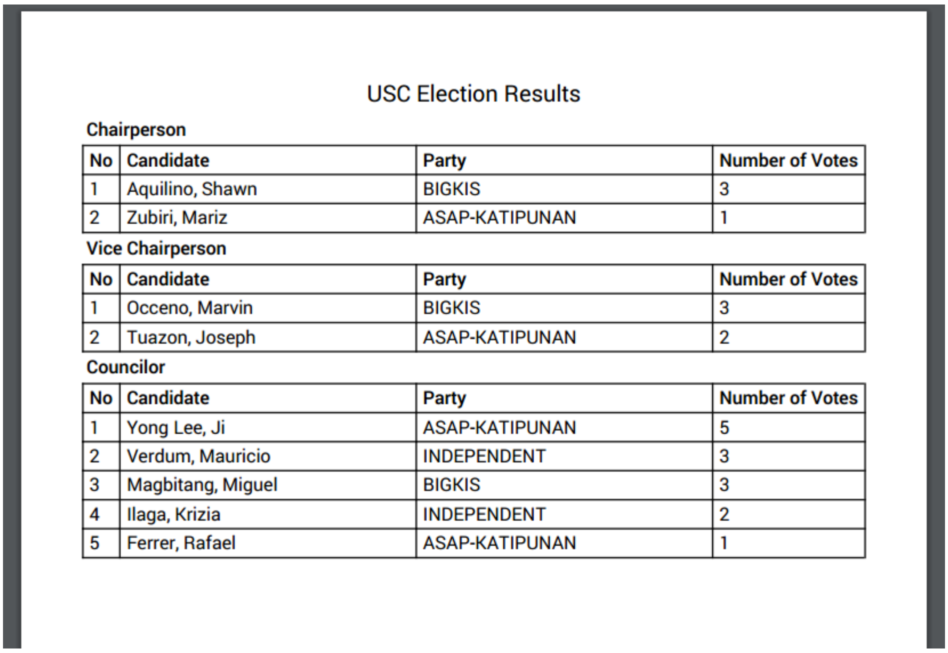
| No. | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Occeno, Marvin | BIGKIS | 3 |
| 2 | Tuazon, Joseph | ASAP-KATIPUNAN | 2 |

Councilor

| No. | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Yong Lee, Ji | ASAP-KATIPUNAN | 5 |
| 2 | Verdum, Mauricio | INDEPENDENT | 3 |
| 3 | Magbitang, Miguel | BIGKIS | 3 |
| 4 | Ilaga, Krizia | INDEPENDENT | 2 |
| 5 | Ferrer, Rafael | ASAP-KATIPUNAN | 1 |

© University of the Philippines Manila

Figure 14: Bottom of results page



USC Election Results

Chairperson

| No | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Aquilino, Shawn | BIGKIS | 3 |
| 2 | Zubiri, Mariz | ASAP-KATIPUNAN | 1 |

Vice Chairperson

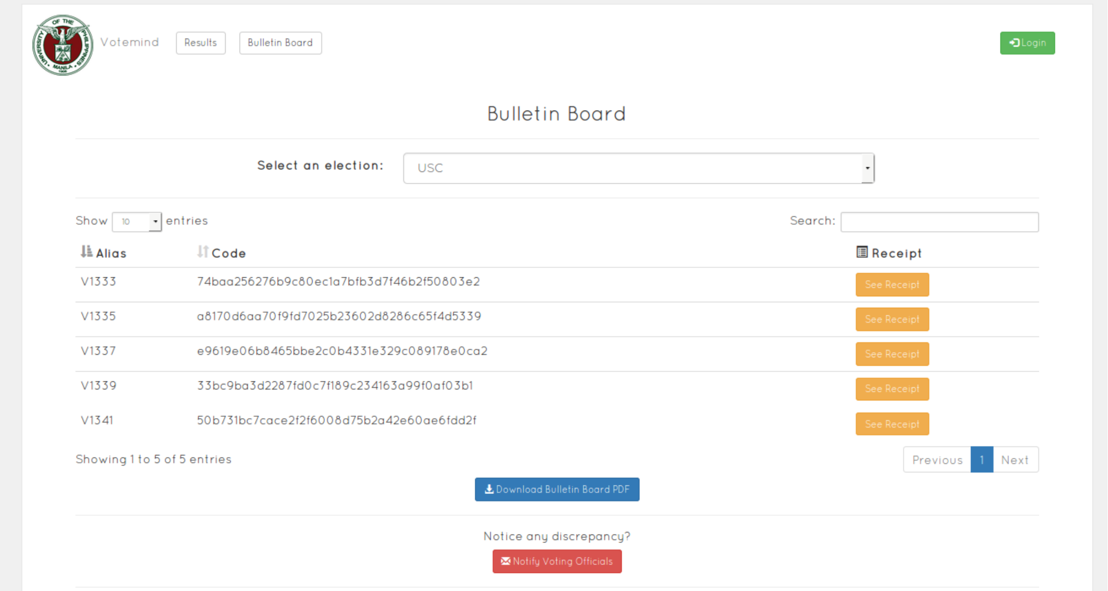
| No | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Occeno, Marvin | BIGKIS | 3 |
| 2 | Tuazon, Joseph | ASAP-KATIPUNAN | 2 |

Councilor

| No | Candidate | Party | Number of Votes |
|---|---|---|---|
| 1 | Yong Lee, Ji | ASAP-KATIPUNAN | 5 |
| 2 | Verdum, Mauricio | INDEPENDENT | 3 |
| 3 | Magbitang, Miguel | BIGKIS | 3 |
| 4 | Ilaga, Krizia | INDEPENDENT | 2 |
| 5 | Ferrer, Rafael | ASAP-KATIPUNAN | 1 |

Figure 15: Results pdf

A public user can also view the bulletin board, which is also posted with the results. It can be accessed by clicking the Bulletin Board button at the top of the page. This contains the receipts of the voters per election. The public user can choose the bulletin board he/she wants to view by selecting the corresponding name of the election in the dropdown menu. The voter can view the hash values of each ballot by clicking the “See Receipt” button of the ballot’s row.



Bulletin Board

Select an election: USC

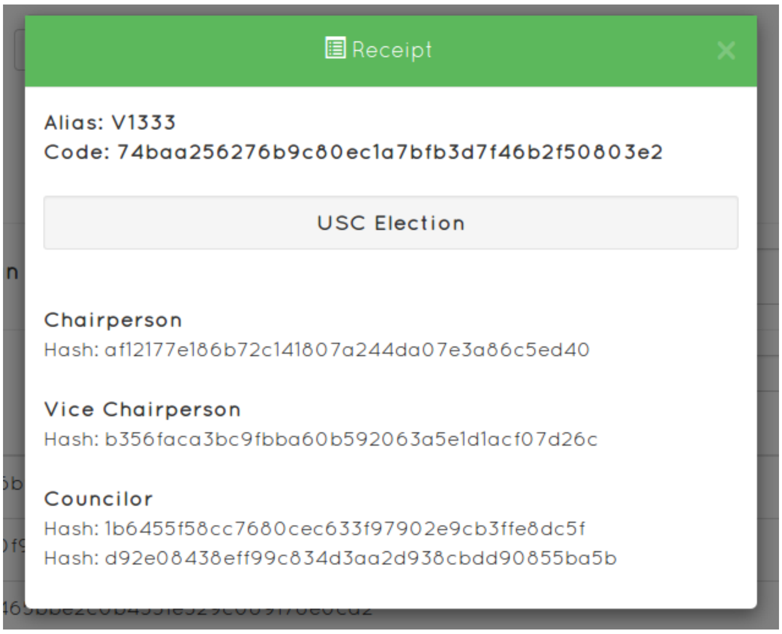
Show 10 entries    Search:

| Alias | Code | Receipt |
|---|---|---|
| V1333 | 74baa256276b9c80ec1a7bfb3d7f46b2f50803e2 | See Receipt |
| V1335 | a8170d6aa70f9fd7025b23602d8286c65f4d5339 | See Receipt |
| V1337 | e9619e06b8465bbe2c0b4331e329c089178e0ca2 | See Receipt |
| V1339 | 33bc9ba3d2287fd0c7f189c234163a99f0af03b1 | See Receipt |
| V1341 | 50b731bc7cace2f2f6008d75b2a42e60ae6fdd2f | See Receipt |

Showing 1 to 5 of 5 entries

Notice any discrepancy?

Figure 16: Bulletin board page



Alias: V1333
Code: 74baa256276b9c80ec1a7bfb3d7f46b2f50803e2

USC Election

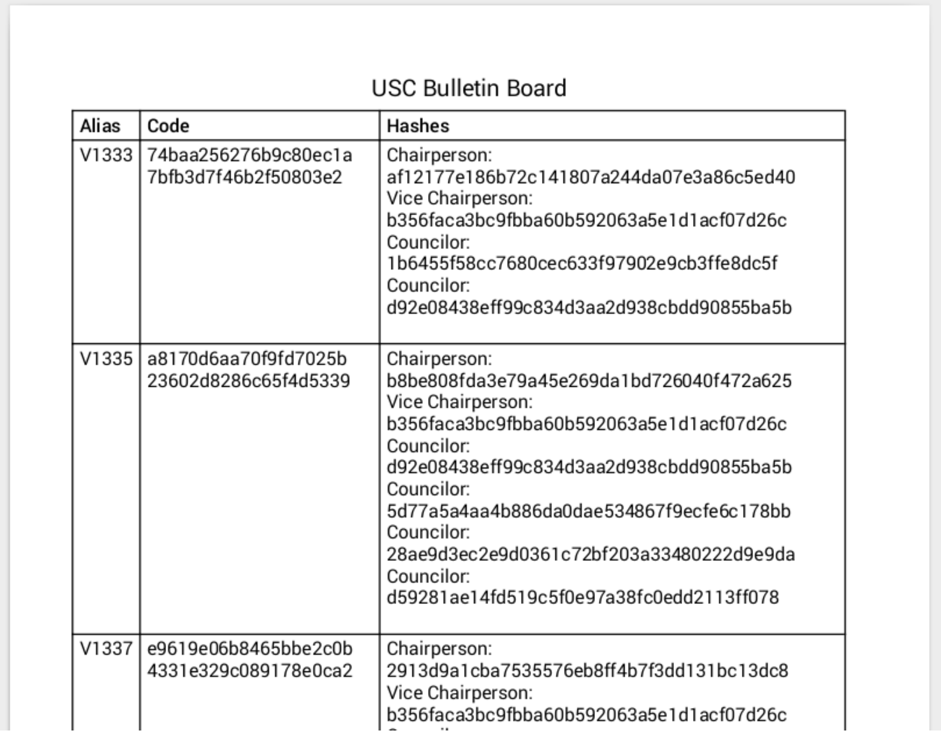
Chairperson
Hash: af12177e186b72c141807a244da07e3a86c5ed40

Vice Chairperson
Hash: b356faca3bc9fbba60b592063a5e1d1acf07d26c

Councilor
Hash: 1b6455f58cc7680cec633f97902e9cb3ffe8dc5f
Hash: d92e08438eff99c834d3aa2d938cbdd90855ba5b

Figure 17: View receipt of a ballot

The public user can also download the pdf version of the bulletin board by clicking the "Download Bulletin Board PDF" button located at the bottom of the page. The user can notify a voting official if he/she finds discrepancies on the hash values by clicking the "Notify Voting Officials" button.

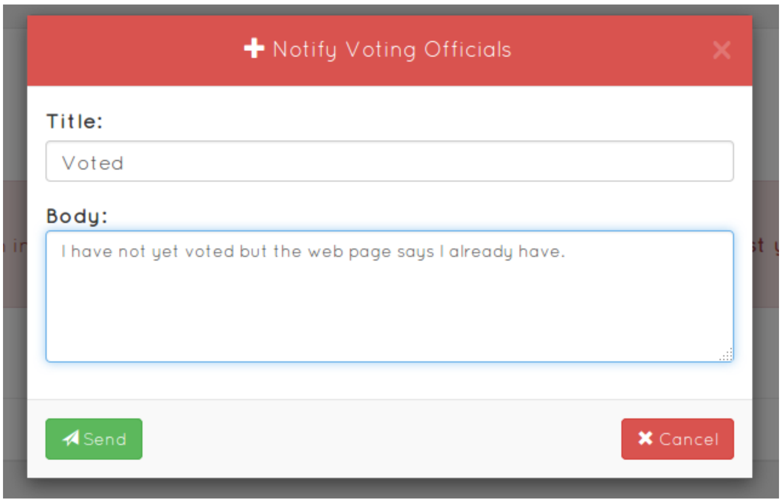


USC Bulletin Board

| Alias | Code | Hashes |
|---|---|---|
| V1333 | 74baa256276b9c80ec1a7bfb3d7f46b2f50803e2 | Chairperson:<br>af12177e186b72c141807a244da07e3a86c5ed40<br>Vice Chairperson:<br>b356faca3bc9fbba60b592063a5e1d1acf07d26c<br>Councilor:<br>1b6455f58cc7680cec633f97902e9cb3ffe8dc5f<br>Councilor:<br>d92e08438eff99c834d3aa2d938cbdd90855ba5b |
| V1335 | a8170d6aa70f9fd7025b23602d8286c65f4d5339 | Chairperson:<br>b8be808fda3e79a45e269da1bd726040f472a625<br>Vice Chairperson:<br>b356faca3bc9fbba60b592063a5e1d1acf07d26c<br>Councilor:<br>d92e08438eff99c834d3aa2d938cbdd90855ba5b<br>Councilor:<br>5d77a5a4aa4b886da0dae534867f9ecfe6c178bb<br>Councilor:<br>28ae9d3ec2e9d0361c72bf203a33480222d9e9da<br>Councilor:<br>d59281ae14fd519c5f0e97a38fc0edd2113ff078 |
| V1337 | e9619e06b8465bbe2c0b4331e329c089178e0ca2 | Chairperson:<br>2913d9a1cba7535576eb8ff4b7f3dd131bc13dc8<br>Vice Chairperson:<br>b356faca3bc9fbba60b592063a5e1d1acf07d26c |

Figure 18: Bulletin board pdf

The user must fill up the form with the title and the details report before clicking send.



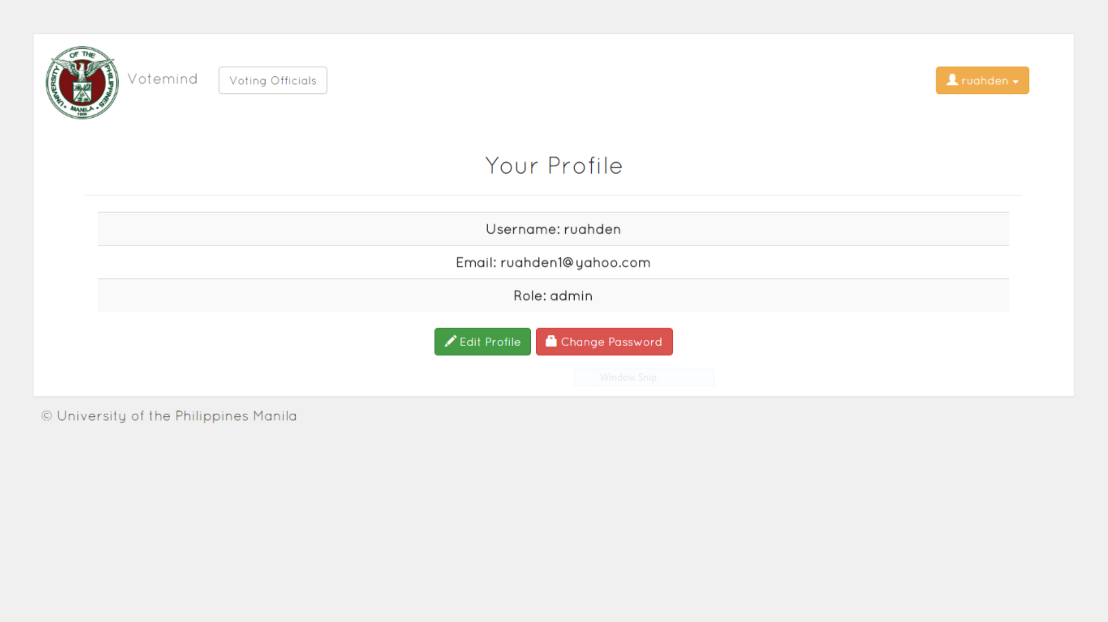
Notify Voting Officials

Title: Voted

Body: I have not yet voted but the web page says I already have.

Figure 19: Notify voting officials

The next pages can only be accessed by either a voting official or an admin. When they have logged in, both of them can view their profiles by clicking their username in the upper right corner of the page and clicking the "Profile" option in the dropdown that will appear.



Your Profile

Username: ruahden

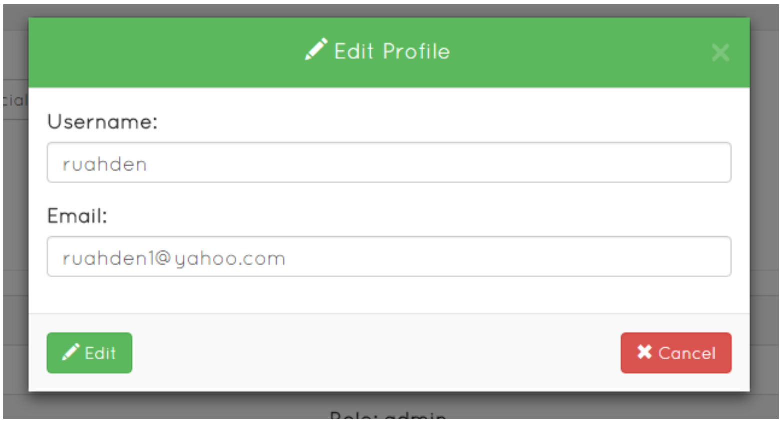
Email: ruahden1@yahoo.com

Role: admin

© University of the Philippines Manila

Figure 20: Manage profile page

They can edit their username or email by clicking the "Edit Profile" button.

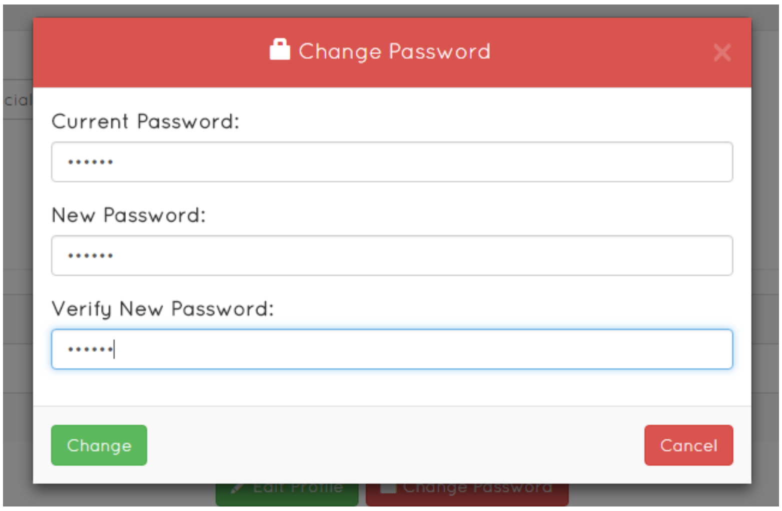


Username: ruahden

Email: ruahden1@yahoo.com

Figure 21: Edit profile

They can change their password by clicking the “Change Password” button. In order for a password change to be successful, the user must be able to type his/her password and type the new password twice.



Change Password

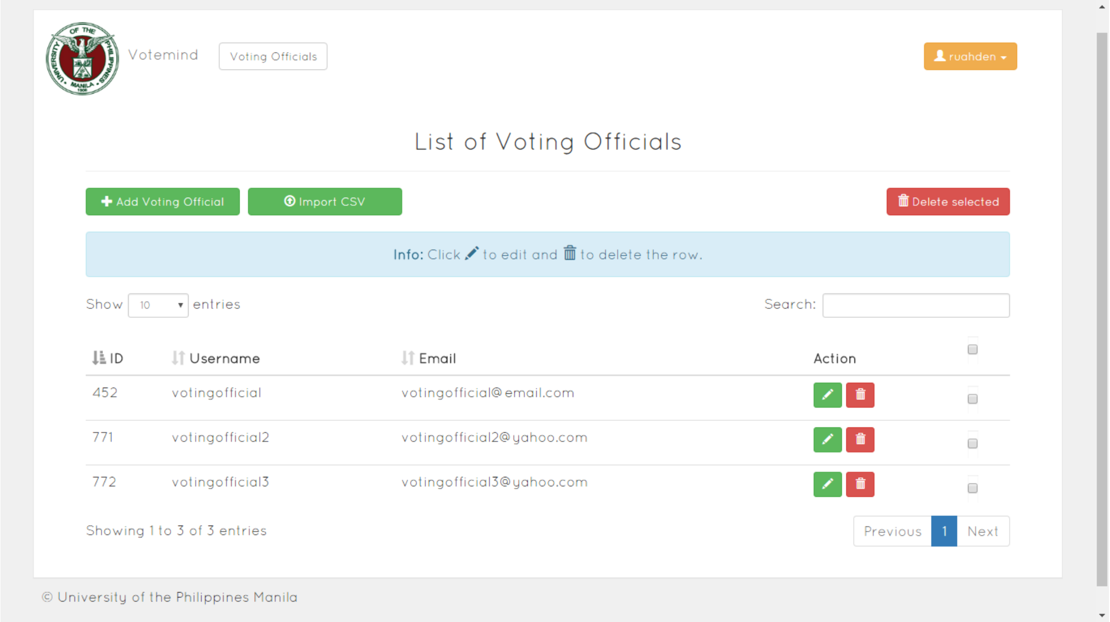
Current Password: ••••••

New Password: ••••••

Verify New Password: ••••••

Figure 22: Change password

The next pages can only be accessed by an admin user. The admin user is responsible for managing the voting officials. His/her homepage is the list of voting officials.



List of Voting Officials

Info: Click to edit and to delete the row.

Show 10 entries

Search:

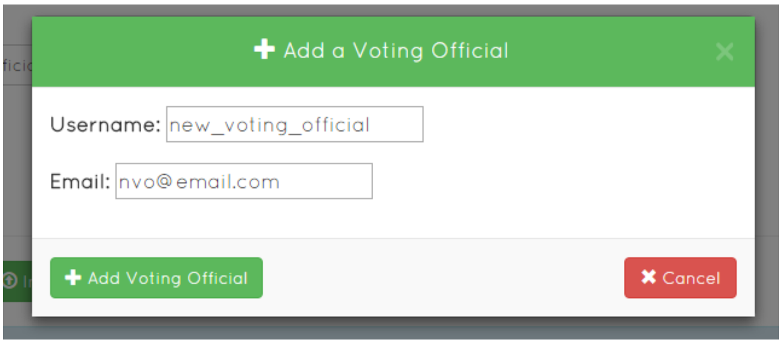
| ID | Username | Email | Action |
|---|---|---|---|
| 452 | votingofficial | votingofficial@email.com | |
| 771 | votingofficial2 | votingofficial2@yahoo.com | |
| 772 | votingofficial3 | votingofficial3@yahoo.com | |

Showing 1 to 3 of 3 entries

© University of the Philippines Manila

Figure 23: Admin home page

An admin can add a voting official by clicking the "Add Voting Official" button. The name and email address must be unique in order for the voting official to be added. The voting official's password will be randomly generated and will be sent via email.

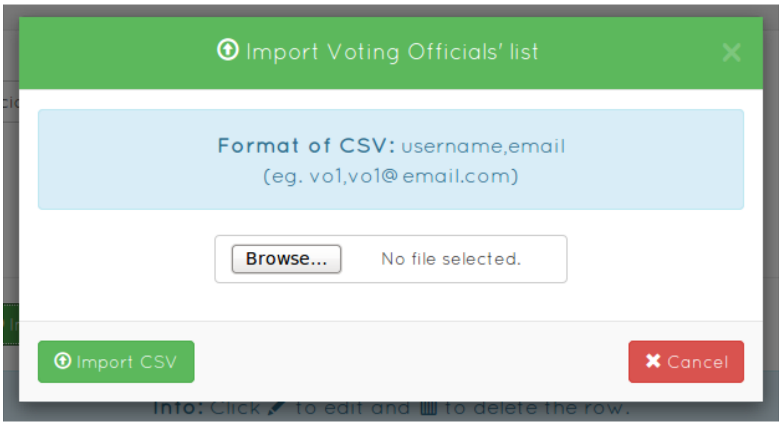


Username: new_voting_official

Email: nvo@email.com

Figure 24: Add a voting official

An admin can import a list of voting officials by clicking the "Import CSV" button.



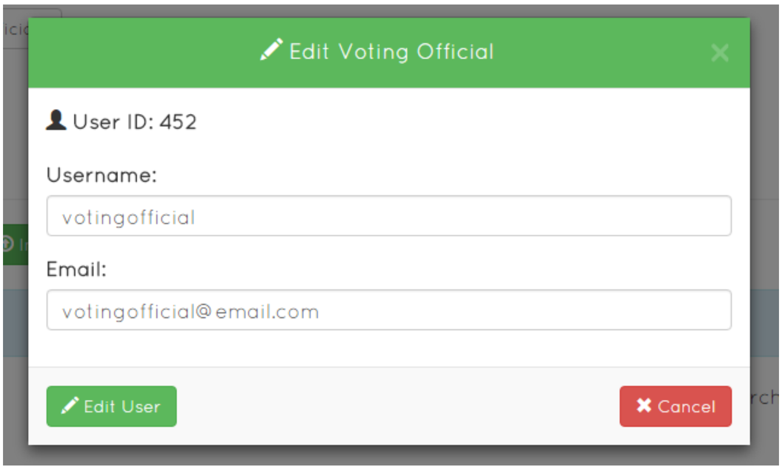
Import Voting Officials' list

**Format of CSV:** username,email
(eg. vo1,vo1@email.com)

Browse...
No file selected.
Import CSV
Cancel
Info: Click ✎ to edit and 🗑 to delete the row.

Figure 25: Import voting officials

An admin can edit an existing voting official by clicking the pencil button on the row of the voting official.



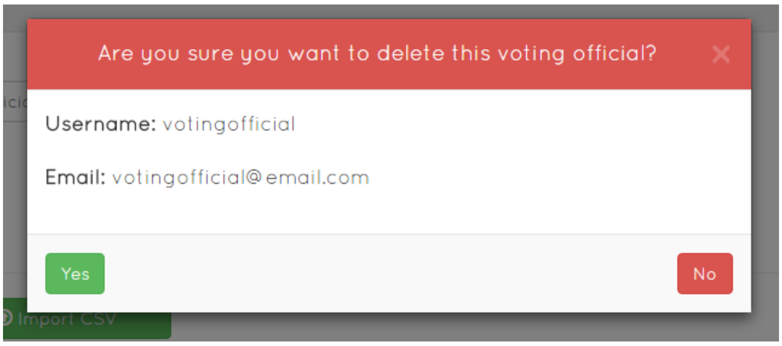
User ID: 452

Username: votingofficial

Email: votingofficial@email.com

Figure 26: Edit a voting official

An admin can delete an existing voting official by clicking the trash button on the row of the voting official and clicking yes on the modal that appears.

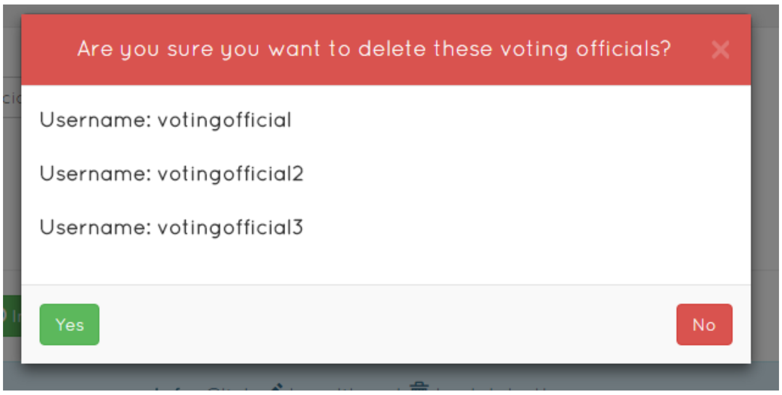

Are you sure you want to delete this voting official?

Username: votingofficial

Email: votingofficial@email.com

Figure 27: Delete a voting official

An admin can delete many voting officials at once by checking the checkbox of their row, clicking the "Delete Selected" button on the top-right corner of the page and clicking yes on the modal that appears.



Are you sure you want to delete these voting officials?

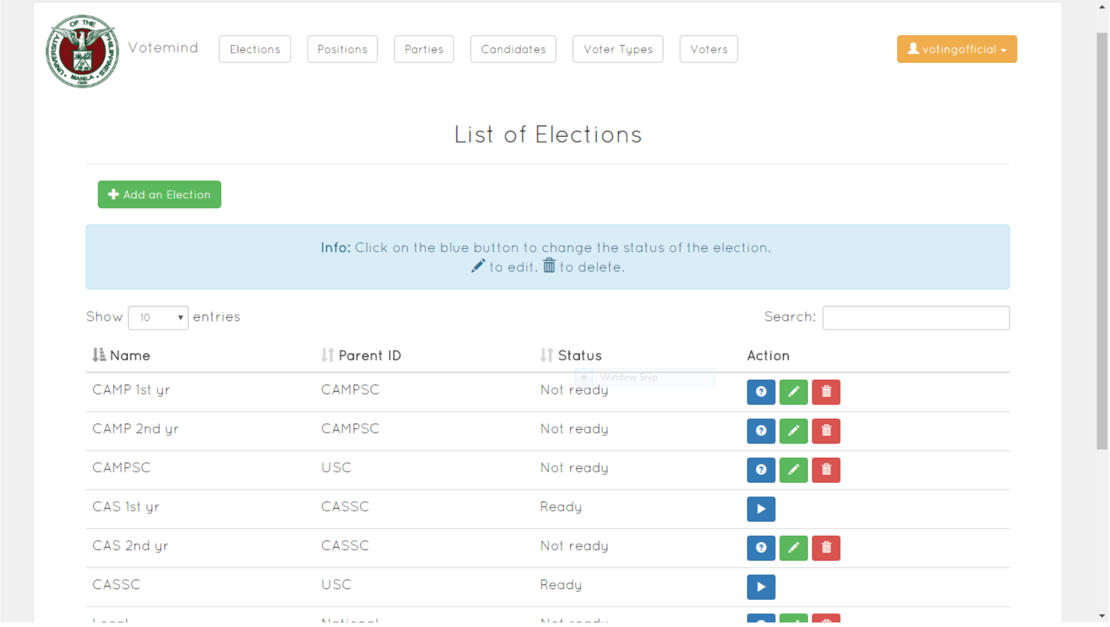
Username: votingofficial

Username: votingofficial2

Username: votingofficial3

Figure 28: Delete many voting officials

The following pages can only be accessed by voting officials. The home page of a voting official is the list of elections.



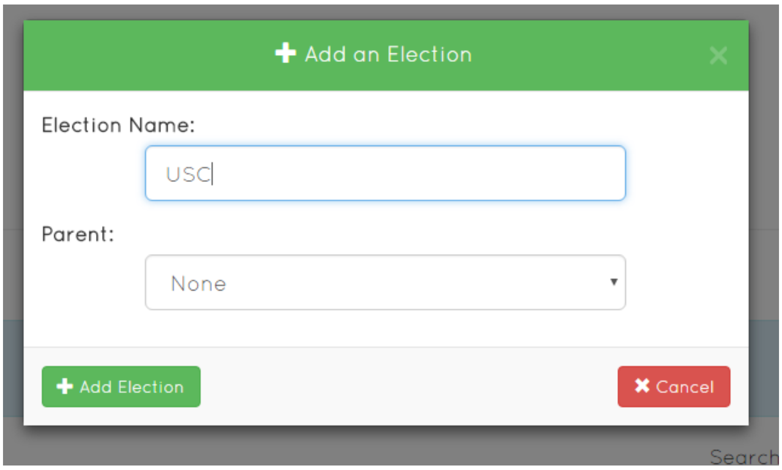
List of Elections

Info: Click on the blue button to change the status of the election. to edit. to delete.

| Name | Parent ID | Status | Action |
|---|---|---|---|
| CAMP 1st yr | CAMPSC | Not ready | |
| CAMP 2nd yr | CAMPSC | Not ready | |
| CAMPSC | USC | Not ready | |
| CAS 1st yr | CASSC | Ready | ▶ |
| CAS 2nd yr | CASSC | Not ready | |
| CASSC | USC | Ready | ▶ |
| Local | National | Not ready | |

Figure 29: Elections page

A voting official can add an election by clicking the “Add Election” button on the upper left corner of the page. The name of the election is required and must be unique. The parent is optional.

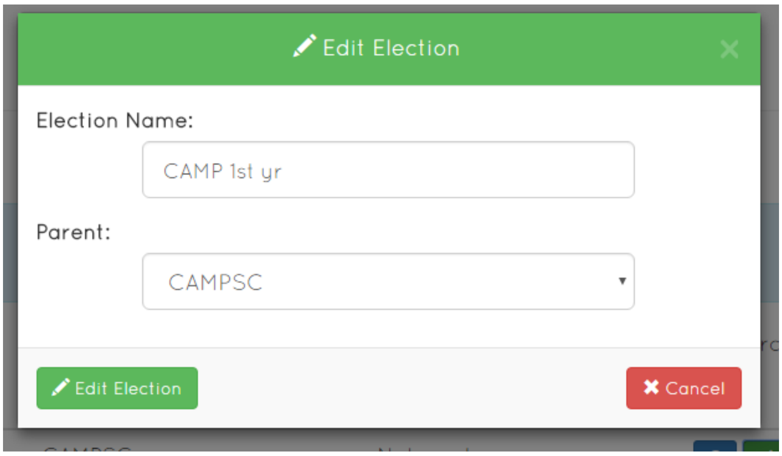


Election Name: USC

Parent: None

Figure 30: Add an election

An existing election can be edited by clicking the pencil button on its row. An election cannot be edited once it is involved in a votertype or if it has a status of running or posted.

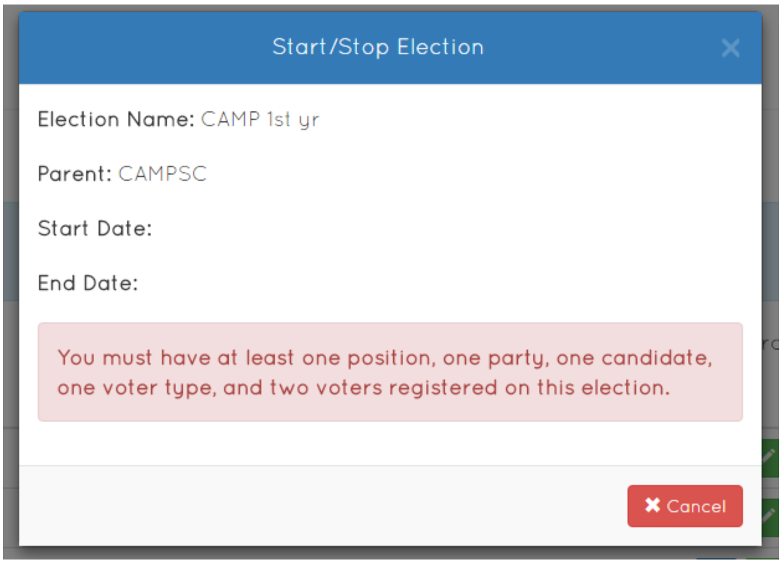


Election Name: CAMP 1st yr

Parent: CAMPSC

Figure 31: Edit an election

A voting official can start, stop or repeat an election. This can be done by clicking the blue button on the election's row. If an election does not have the requirements for it to start, its status will be "Not Ready". If you click the "i" button on its row, this informative modal will appear.



Start/Stop Election

Election Name: CAMP 1st yr

Parent: CAMPSC

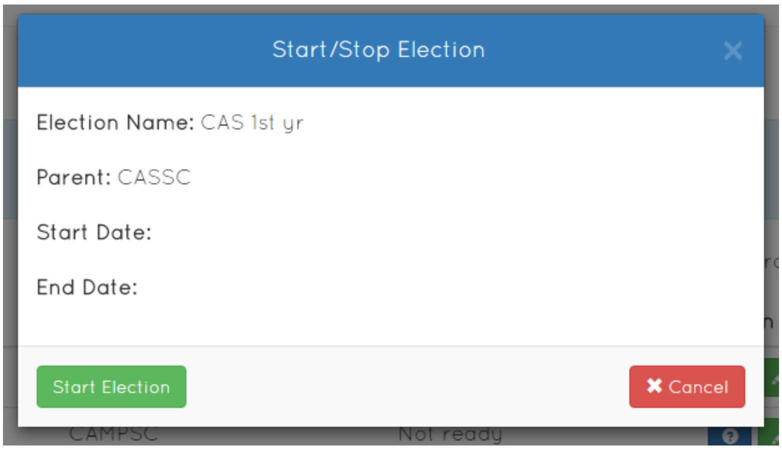
Start Date:

End Date:

You must have at least one position, one party, one candidate, one voter type, and two voters registered on this election.

Figure 32: View election info

If an election is "Ready" it can be started by clicking "Start Election" on the modal that appears after clicking the blue button on its row.



Start/Stop Election

Election Name: CAS 1st yr

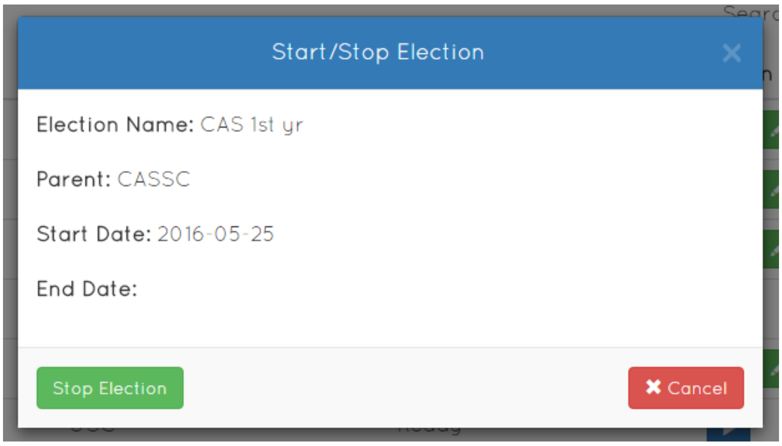
Parent: CASSC

Start Date:

End Date:

Figure 33: Start an election

During an election, its positions, parties, candidates, votertypes and voters cannot be deleted or edited. Voters also cannot be added to an ongoing election. A voting official can stop the election, and when it is stopped, its results and bulletin board will be automatically posted. An election is stopped by clicking the blue button on the election's row and clicking "Stop Election" on the modal that appears.



Election Name: CAS 1st yr

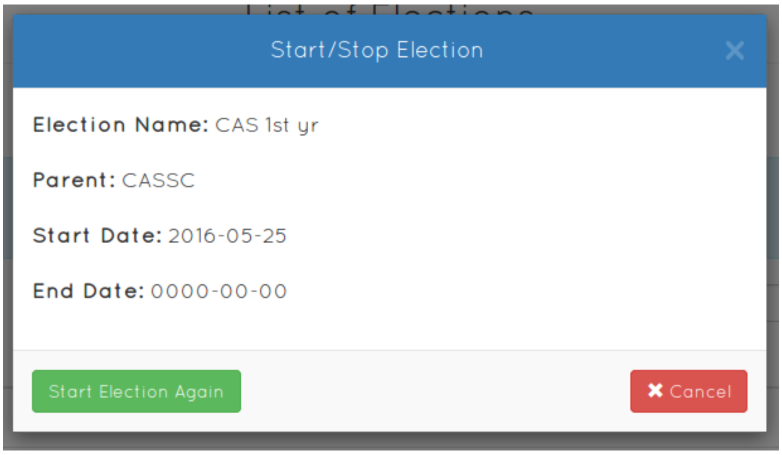
Parent: CASSC

Start Date: 2016-05-25

End Date:

Figure 34: Stop an election

When an election is posted, its positions, parties and candidates cannot be edited or deleted. A voting official can start an election again by clicking the "Start Election Again" button on the modal that appears after clicking the blue button on the election's row. When an election is "Started Again", it goes back to the "Ready" state.



Election Name: CAS 1st yr

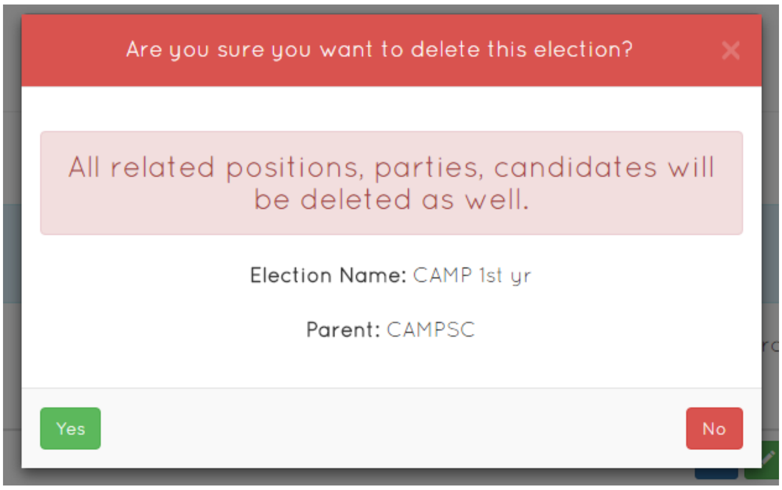
Parent: CASSC

Start Date: 2016-05-25

End Date: 0000-00-00

Figure 35: Repeat an election

If an election is not running or posted and not attached to any votertype, voting officials can delete it by clicking the trash button on its row and clicking yes on the modal.



Are you sure you want to delete this election?

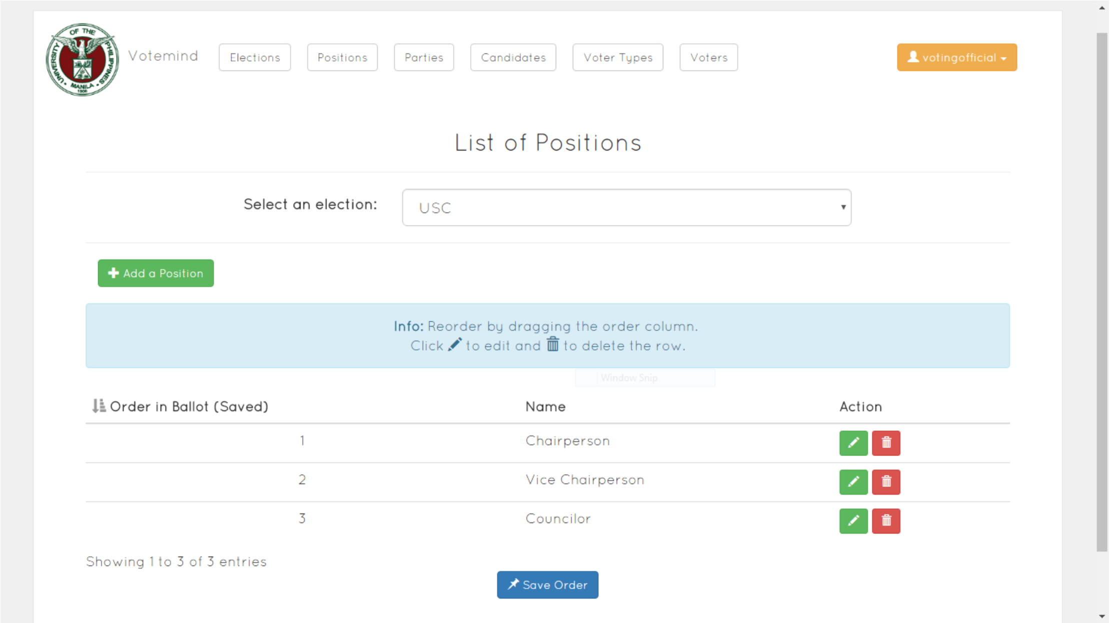
All related positions, parties, candidates will be deleted as well.

Election Name: CAMP 1st yr

Parent: CAMPSC

Figure 36: Delete an election

Once an election is made, a voting official can now add the components needed for the election. Positions can be managed by clicking the Positions button on the navigation bar. In order to manage the positions in a specific election, the voting official must choose the election in the dropdown menu.



List of Positions

Select an election: USC

Info: Reorder by dragging the order column.
Click ✎ to edit and 🗑 to delete the row.

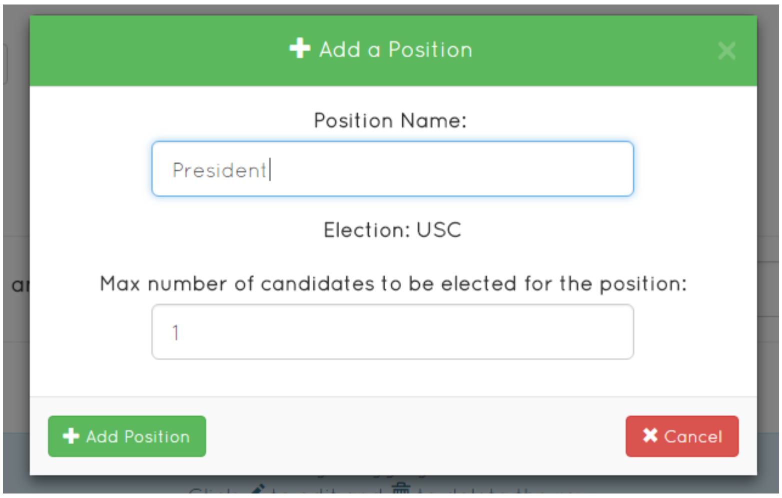
| Order in Ballot (Saved) | Name | Action |
|---|---|---|
| 1 | Chairperson | ✎ 🗑 |
| 2 | Vice Chairperson | ✎ 🗑 |
| 3 | Councilor | ✎ 🗑 |

Showing 1 to 3 of 3 entries

Figure 37: Positions page

The ordering of the positions can be rearranged by using the drag and drop feature of the page. The voting official can drag a row by its order column and drop it to its new position. When the ordering is complete, the "Save Order" button on the bottom of the page can be clicked to save the order.

Positions can be added by clicking the "Add Position" button at the upper left corner of the page. The user must specify the name of the position and the maximum number of candidates that must be voted for that position. It will be automatically ordered as last in the ballot when added.



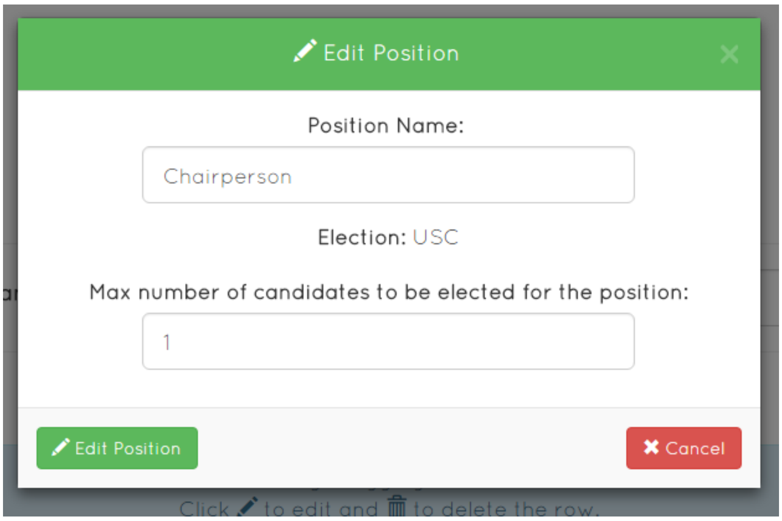
Position Name: President

Election: USC

Max number of candidates to be elected for the position: 1

Figure 38: Add a position

Positions can be edited by clicking the pencil button at the row of that position.



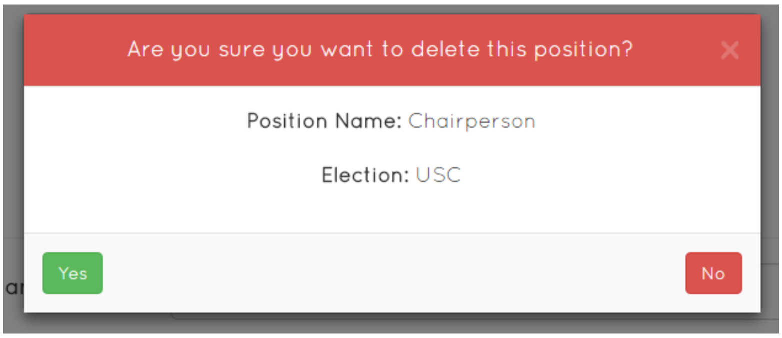
Position Name: Chairperson

Election: USC

Max number of candidates to be elected for the position: 1

Figure 39: Edit a position

A position can be deleted by clicking the trash button on its row and clicking yes on the modal. When a position is deleted, the ordering of the other positions in the ballot is updated.



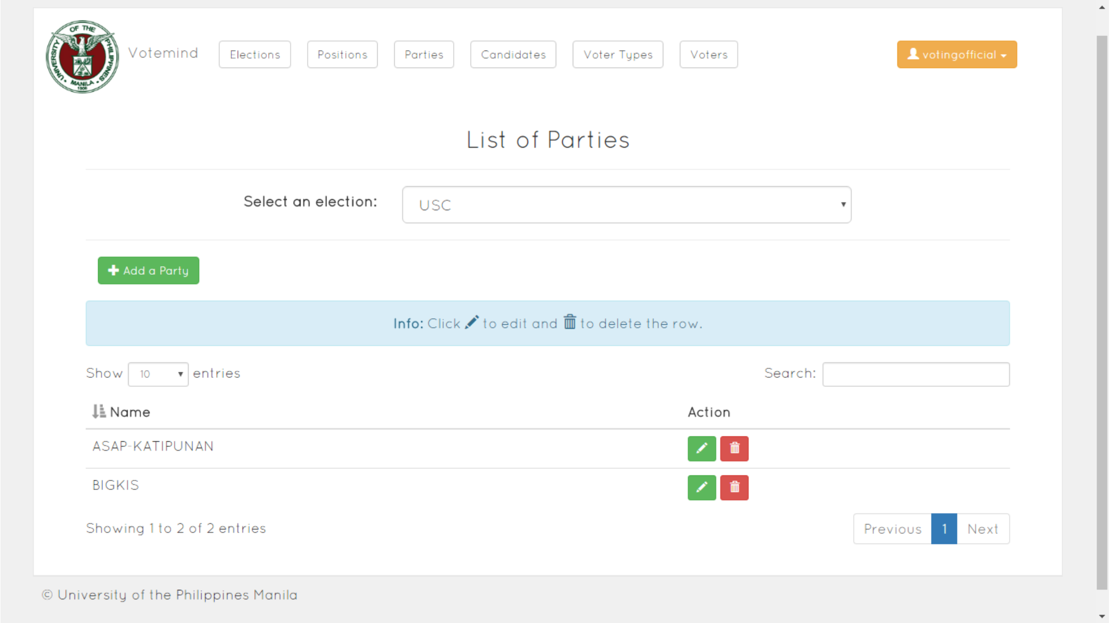
Are you sure you want to delete this position?

**Position Name:** Chairperson

**Election:** USC

Figure 40: Delete a position

Parties in an election can be managed by clicking the Parties button on the navigation bar and choosing an election in the dropdown menu.



List of Parties

Select an election: USC

**Info:** Click ✎ to edit and 🗑 to delete the row.

Show 10 entries

Search:

| Name | Action |
|---|---|
| ASAP-KATIPUNAN | ✎ 🗑 |
| BIGKIS | ✎ 🗑 |

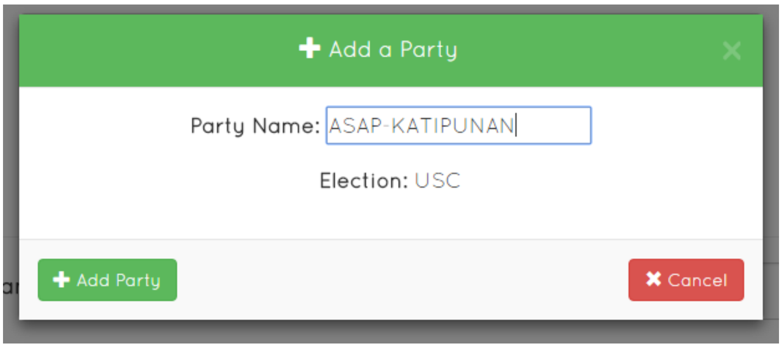
Showing 1 to 2 of 2 entries

© University of the Philippines Manila

Figure 41: Parties page

A voting official can add a party by clicking the "Add Party" button on the upper left corner of the page. The name of the party must be provided.

Note: The independent party need not be added, it is already in the option of the candidates.

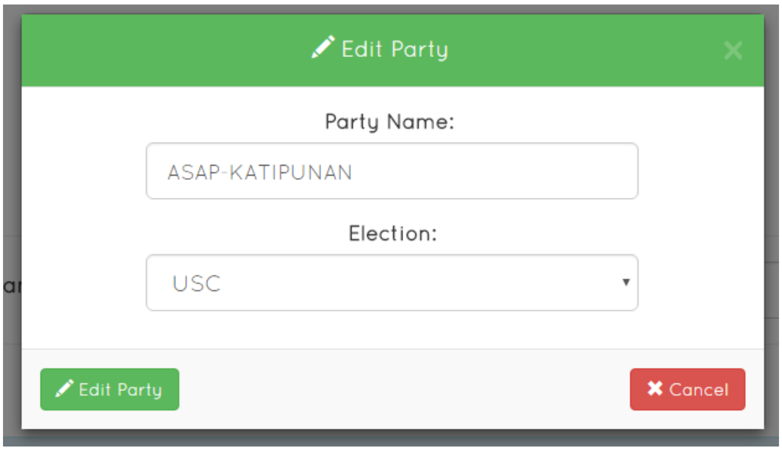


Party Name: ASAP-KATIPUNAN

Election: USC

Figure 42: Add a party

Parties can be edited by clicking the pencil button on the row of the party to be edited.

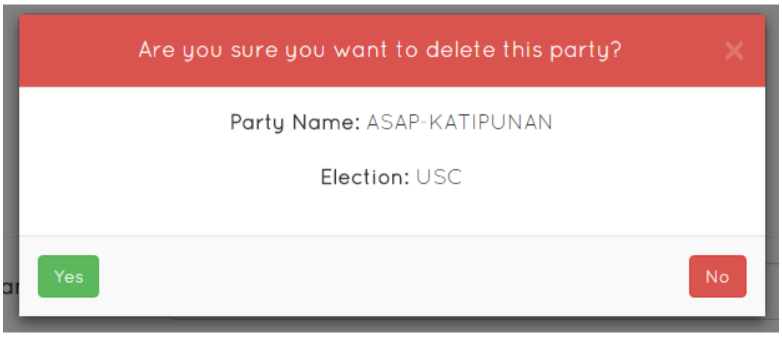


Party Name: ASAP-KATIPUNAN

Election: USC

Figure 43: Edit a party

A party can be deleted by clicking the trash button on the party row and clicking yes on the modal that appears.



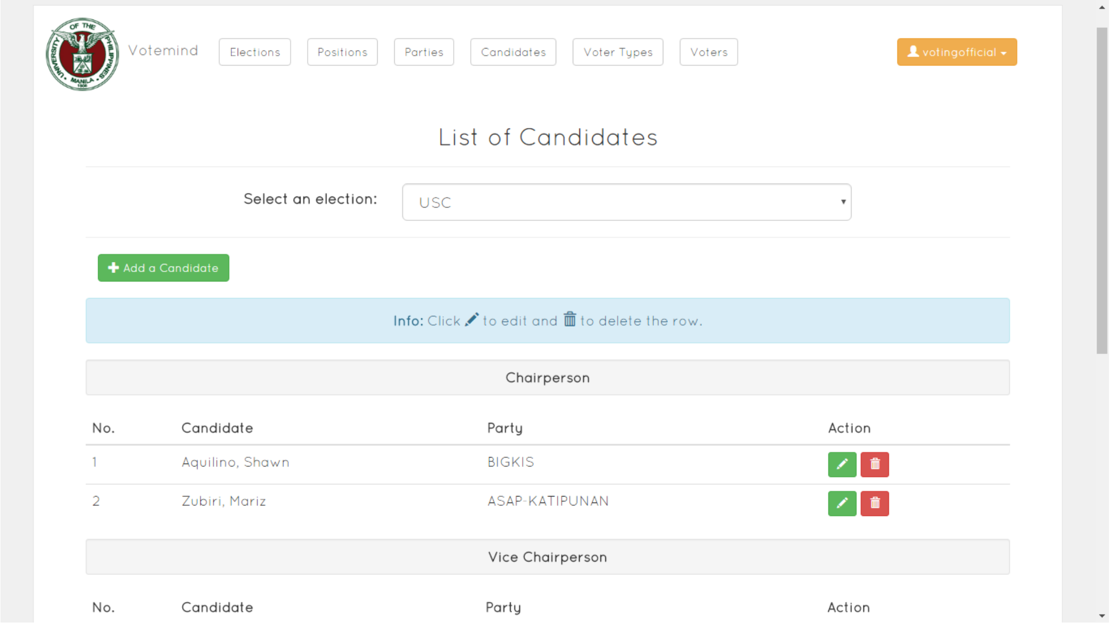
Are you sure you want to delete this party?

**Party Name:** ASAP-KATIPUNAN

**Election:** USC

Figure 44: Delete a party

Candidates in an election can be managed by clicking the Candidates button on the navigation bar and choosing an election in the dropdown menu.



List of Candidates

Select an election: USC

**Info:** Click ✎ to edit and 🗑 to delete the row.

Chairperson

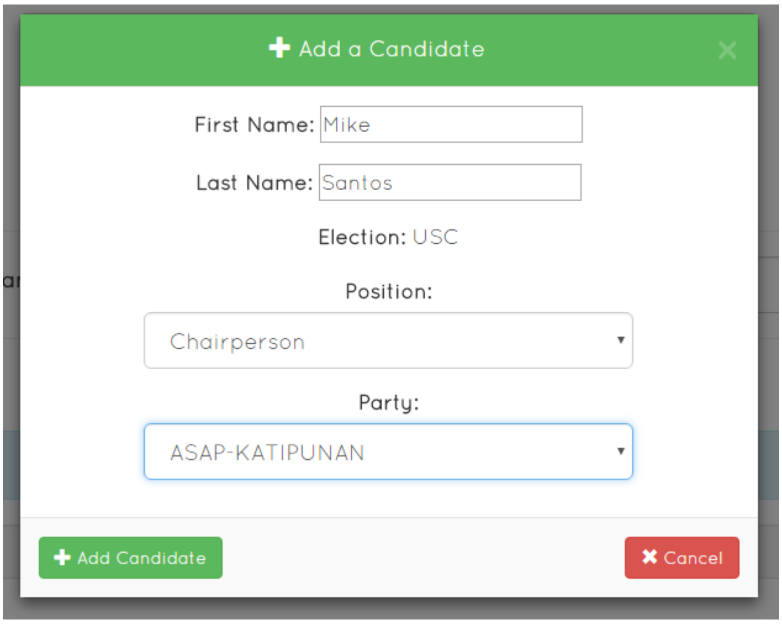
| No. | Candidate | Party | Action |
|---|---|---|---|
| 1 | Aquilino, Shawn | BIGKIS | ✎ 🗑 |
| 2 | Zubiri, Mariz | ASAP-KATIPUNAN | ✎ 🗑 |

Vice Chairperson

| No. | Candidate | Party | Action |
|---|---|---|---|

Figure 45: Candidates page

A candidate can be added by clicking the “Add Candidate” button at the upper left corner of the page. Its first name, last name, position and party must be provided.



First Name: Mike

Last Name: Santos

Election: USC

Position:

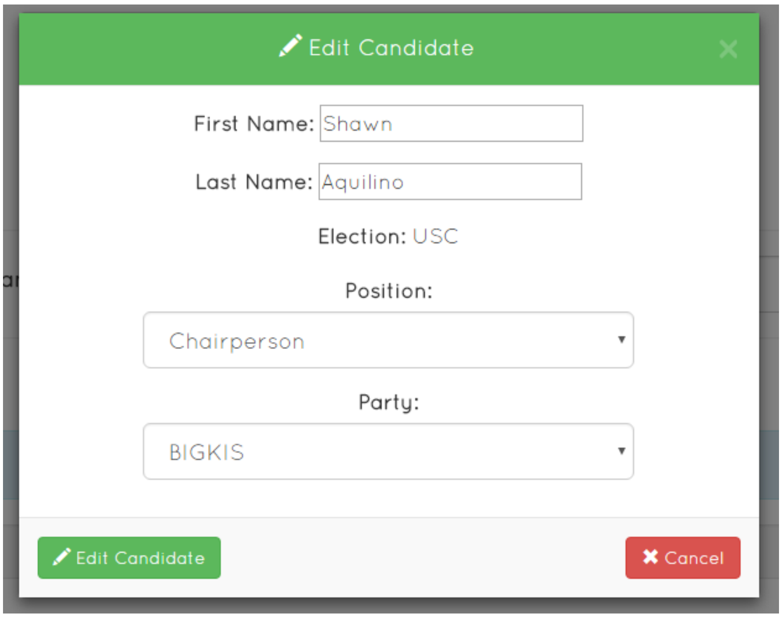
Chairperson

Party:

ASAP-KATIPUNAN

Figure 46: Add a candidate

Candidates can be edited by clicking the pencil button at the row of the candidate to be edited.



First Name: Shawn

Last Name: Aquilino

Election: USC

Position:

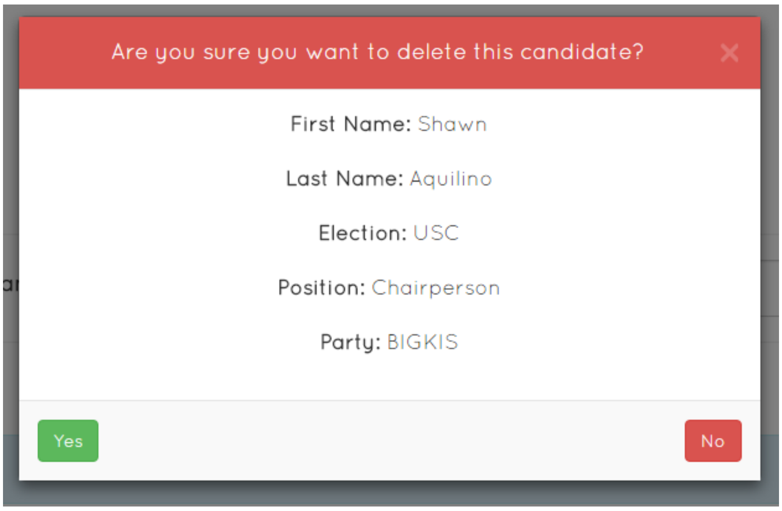
Chairperson

Party:

BIGKIS

Figure 47: Edit a candidate

A candidate can be deleted by clicking the trash button on the candidate's row and clicking yes on the modal that appears.



Are you sure you want to delete this candidate?

First Name: Shawn

Last Name: Aquilino

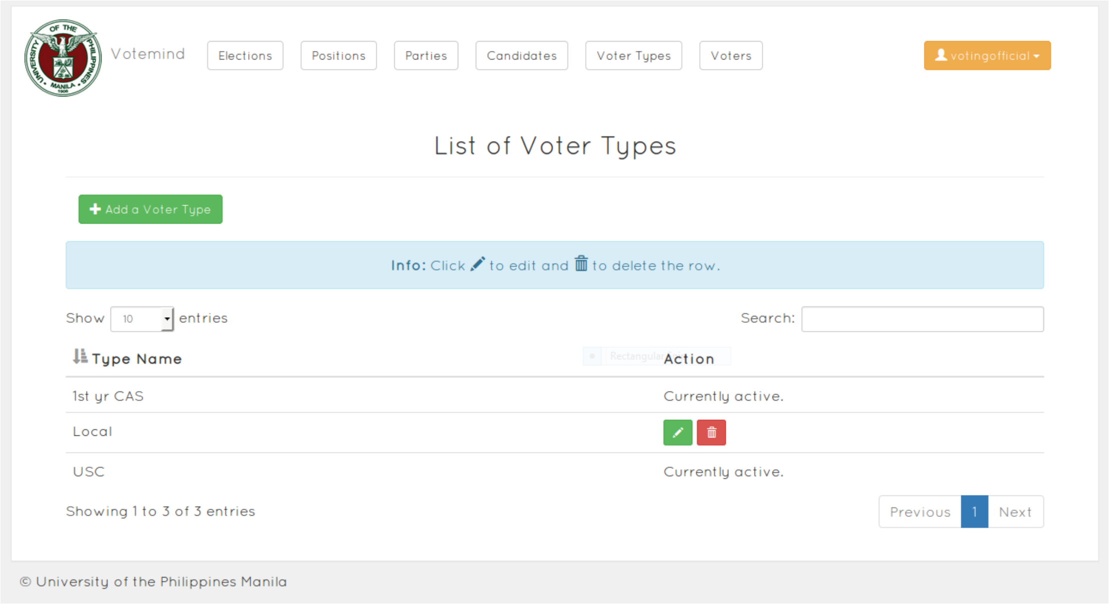
Election: USC

Position: Chairperson

Party: BIGKIS

Figure 48: Delete a candidate

The voting official can also manage the votertypes, its page can be viewed by clicking the "Voter Types" button on the navigation bar. A votertype is used to identify the elections in which a voter is included in. A votertype cannot be edited or deleted once it includes an active election.



List of Voter Types

Info: Click ✎ to edit and 🗑 to delete the row.

Show 10 entries Search:

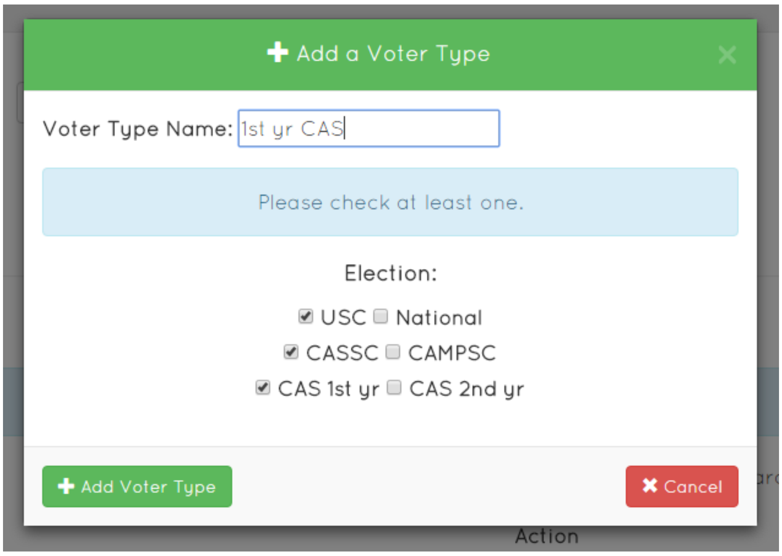
| Type Name | Action |
|---|---|
| 1st yr CAS | Currently active. |
| Local | ✎ 🗑 |
| USC | Currently active. |

Showing 1 to 3 of 3 entries

© University of the Philippines Manila

Figure 49: Votertypes page

A votertype can be added by clicking the "Add a voter type" button on the upper left corner of the page. The user must identify its name and choose the elections that are included in it. The first elections that will appear in the choices will be the elections without a parent. If you check an election, its children will appear.



Voter Type Name: 1st yr CAS

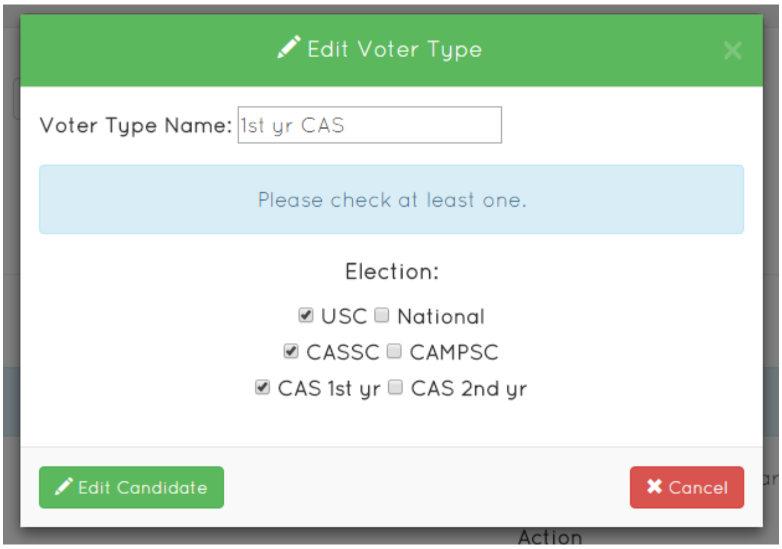
Please check at least one.

Election:

☑ USC □ National
☑ CASSC □ CAMPSC
☑ CAS 1st yr □ CAS 2nd yr

Figure 50: Add a votertype

A votertype can be edited by clicking the pencil button on the votertype's row.



Voter Type Name: 1st yr CAS

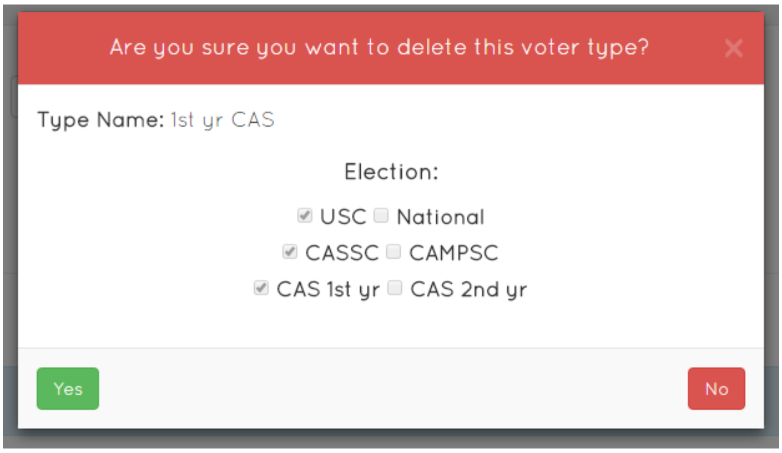
Please check at least one.

Election:

☑ USC □ National
☑ CASSC □ CAMPSC
☑ CAS 1st yr □ CAS 2nd yr

Figure 51: Edit a votertype

A votertype can be deleted by clicking the trash button on the votertype's row and clicking yes on the modal that appears.



Are you sure you want to delete this voter type?

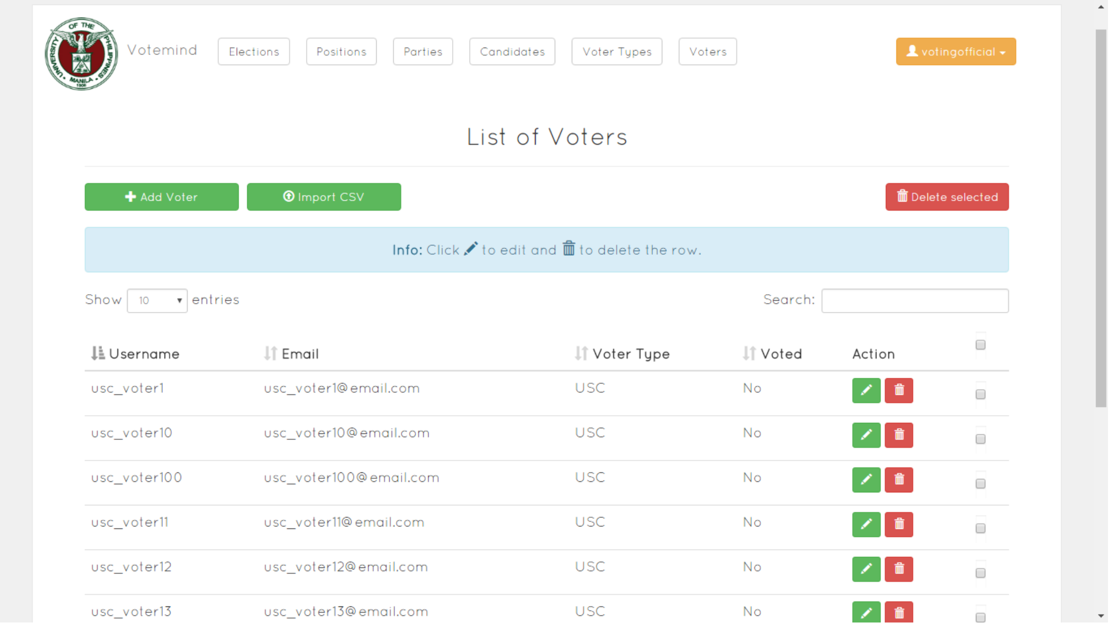
Type Name: 1st yr CAS

Election:

☑ USC □ National
☑ CASSC □ CAMPSC
☑ CAS 1st yr □ CAS 2nd yr

Figure 52: Delete a votertype

The voting official is responsible for managing the voters. The list of voters can be accessed by clicking the “Voters” button on the navigation bar.



List of Voters

Info: Click ✎ to edit and 🗑 to delete the row.

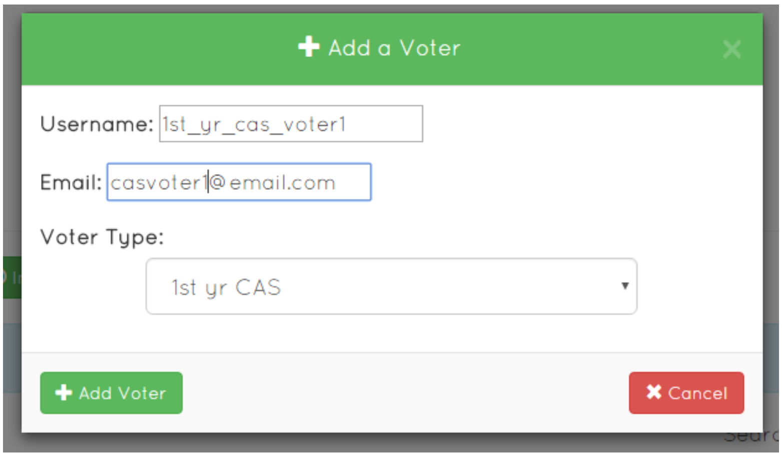
Show 10 entries

Search:

| Username | Email | Voter Type | Voted | Action | □ |
|---|---|---|---|---|---|
| usc_voter1 | usc_voter1@email.com | USC | No | ✎ 🗑 | □ |
| usc_voter10 | usc_voter10@email.com | USC | No | ✎ 🗑 | □ |
| usc_voter100 | usc_voter100@email.com | USC | No | ✎ 🗑 | □ |
| usc_voter11 | usc_voter11@email.com | USC | No | ✎ 🗑 | □ |
| usc_voter12 | usc_voter12@email.com | USC | No | ✎ 🗑 | □ |
| usc_voter13 | usc_voter13@email.com | USC | No | ✎ 🗑 | □ |

Figure 53: Voters page

The voting official can add voters by clicking the "Add Voter" button on the upper left corner of the page. Its username, email and votertype must be specified. The username and email must be unique in order for it to be added. When added, a randomly generated password will be sent via email.



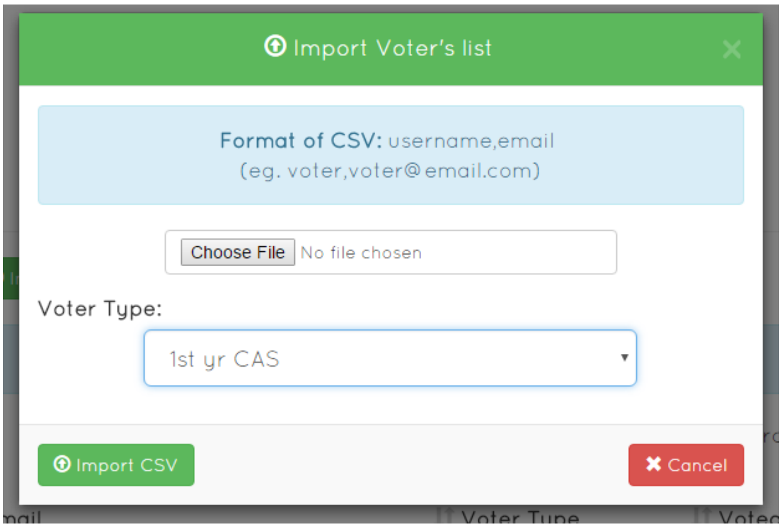
Username: 1st_yr_cas_voter1

Email: casvoter1@email.com

Voter Type: 1st yr CAS

Figure 54: Add a voter

The voting official can also add multiple voters at once by clicking the "Import CSV" button and uploading a CSV file containing the voters' usernames and emails.



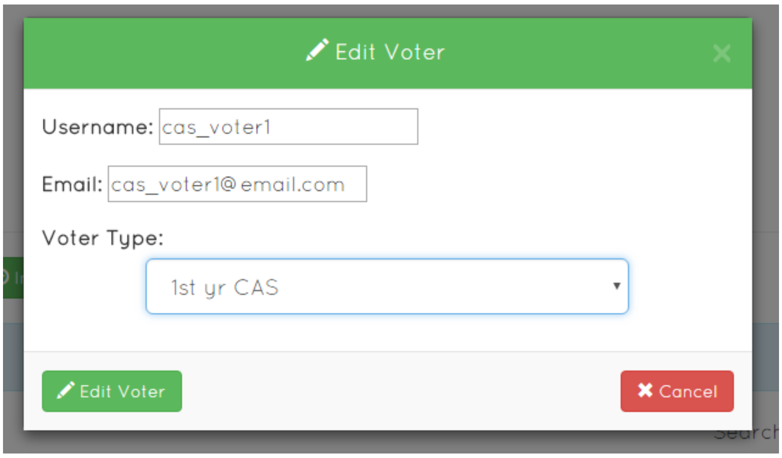
**Format of CSV:** username,email
(eg. voter,voter@email.com)

No file chosen

Voter Type: 1st yr CAS

Figure 55: Import a voters list

Voters can be edited by clicking the pencil button on the voter's row.



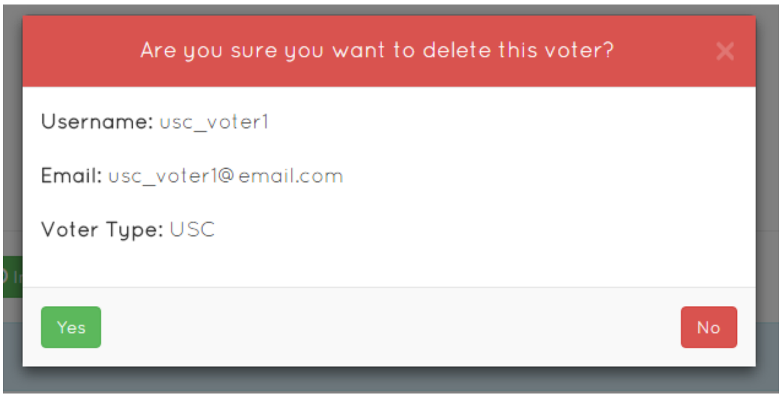
Username: cas_voter1

Email: cas_voter1@email.com

Voter Type: 1st yr CAS

Figure 56: Edit a voter

The voting official can delete a voter by clicking the trash button on the voter's row and clicking yes on the modal that appears.


Are you sure you want to delete this voter?

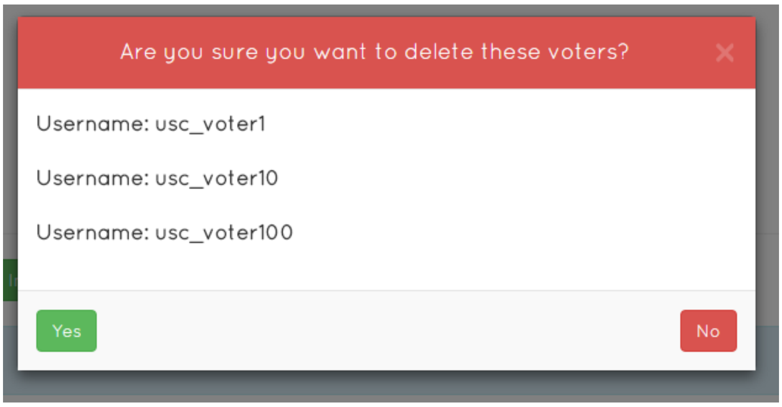
Username: usc_voter1

Email: usc_voter1@email.com

Voter Type: USC

Figure 57: Delete a voter

The voting official can delete multiple voters at once by checking the checkboxes of the voters to be deleted, clicking the "Delete Selected" button on the upper right corner of the page and clicking yes on the modal that appears.



Are you sure you want to delete these voters?

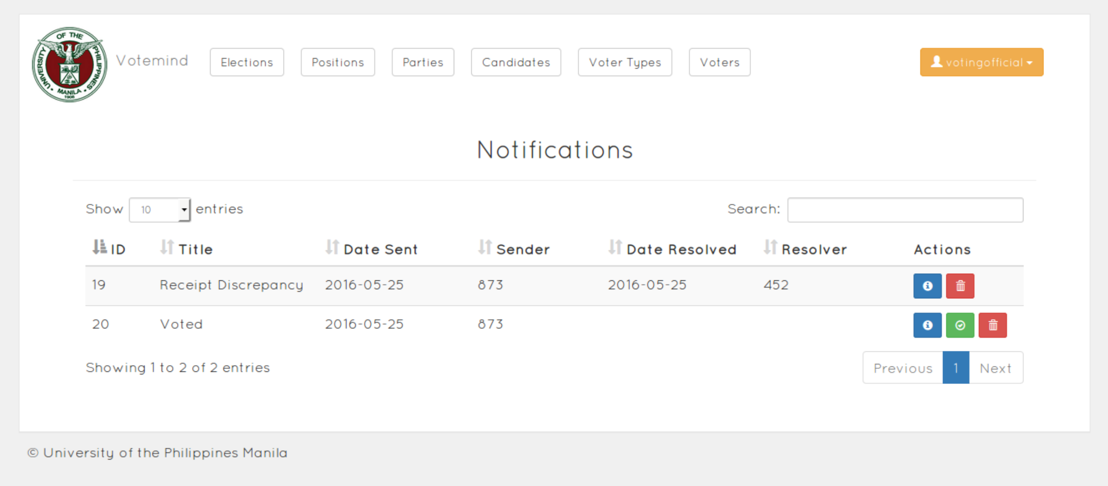
Username: usc_voter1

Username: usc_voter10

Username: usc_voter100

Figure 58: Delete voters

The voting official receives and manages the notifications from public users and voters. The notifications page can be viewed by clicking his/her username at the upperleft corner of the page and clicking on the notifications button from the dropdown menu.



Notifications

Show 10 entries

Search:

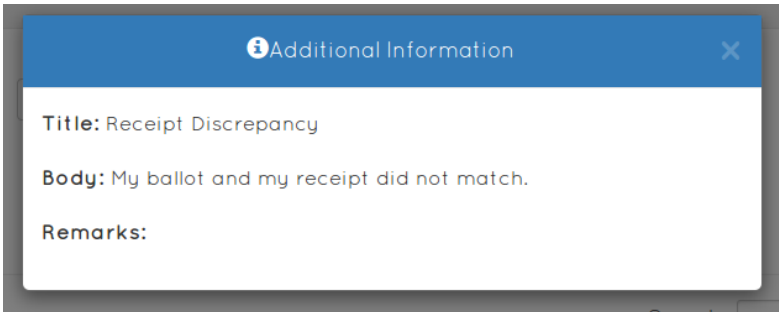
| ID | Title | Date Sent | Sender | Date Resolved | Resolver | Actions |
|---|---|---|---|---|---|---|
| 19 | Receipt Discrepancy | 2016-05-25 | 873 | 2016-05-25 | 452 | ⓘ 🗑 |
| 20 | Voted | 2016-05-25 | 873 | | | ⓘ ✓ 🗑 |

Showing 1 to 2 of 2 entries

© University of the Philippines Manila

Figure 59: Notifications Page

The voting official can see the details of the notification by clicking on the i button on the notification's row.



Additional Information

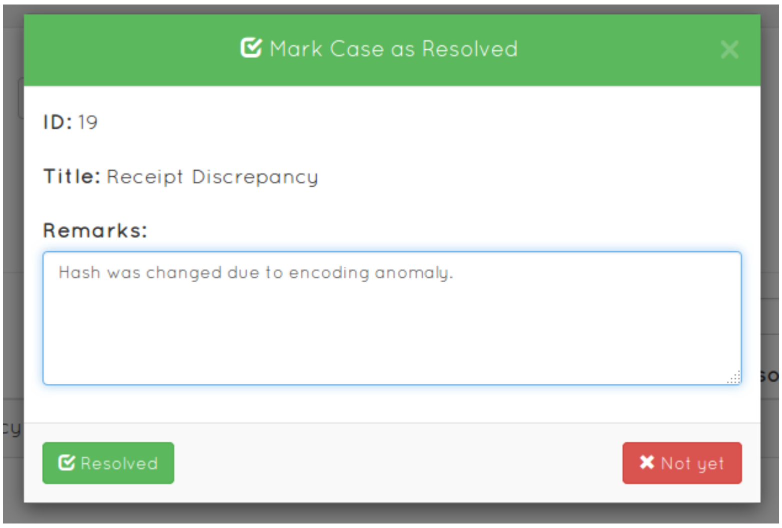
**Title:** Receipt Discrepancy

**Body:** My ballot and my receipt did not match.

**Remarks:**

Figure 60: See details of a notification

The voting official can mark an issue as resolved if it has already been solved. Remarks must be added when an issue is resolved.



Mark Case as Resolved

**ID:** 19

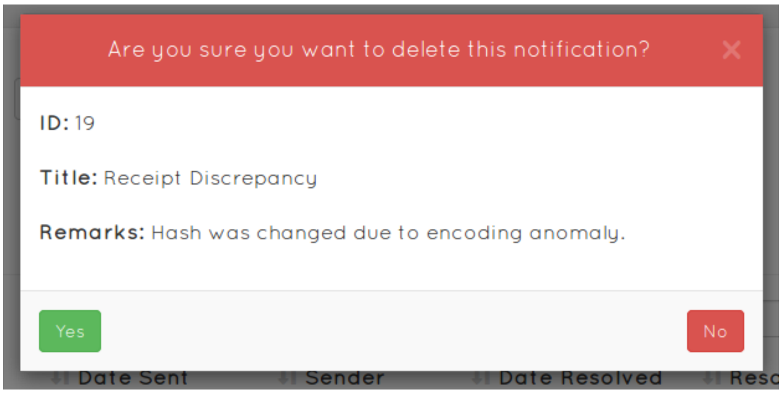
**Title:** Receipt Discrepancy

**Remarks:**

Hash was changed due to encoding anomaly.

Figure 61: Resolve a notification

The voting official can also delete a notification if it is no longer needed by clicking on the trash button at the notification's row and clicking yes on the modal that appears.



Are you sure you want to delete this notification?

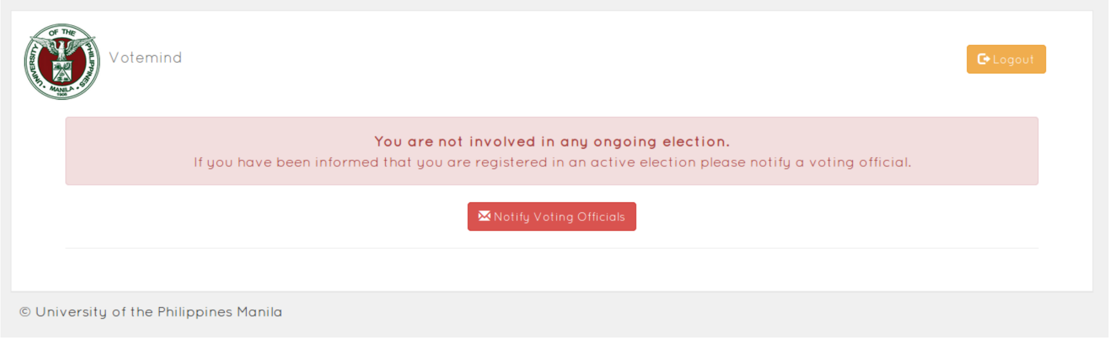
**ID:** 19

**Title:** Receipt Discrepancy

**Remarks:** Hash was changed due to encoding anomaly.

Figure 62: Delete a notification

The following pages are the pages accessible only to voters. If a voter logs in but the elections he/her is registered to has not yet started, the voter will be directed to this page.



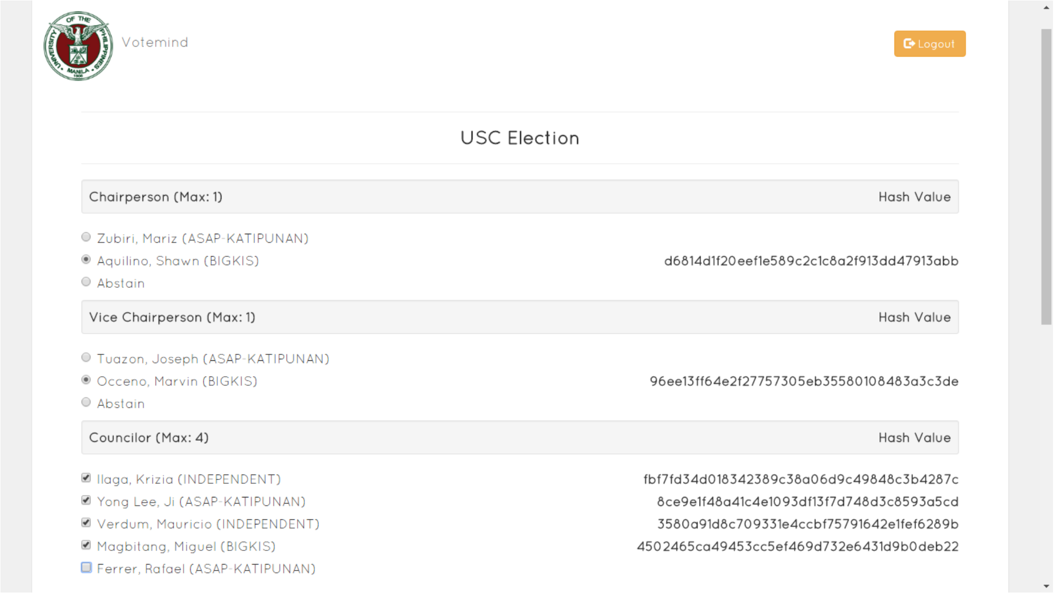
You are not involved in any ongoing election.

If you have been informed that you are registered in an active election please notify a voting official.

© University of the Philippines Manila

Figure 63: Inactive voter page

If the votertype of the voter is active, he/she can now vote. While selecting his/her candidates, the hash value of the candidate appears on the right side. The hash value is unique per voter and candidate.



USC Election

Chairperson (Max: 1) — Hash Value

○ Zubiri, Mariz (ASAP-KATIPUNAN)
● Aquilino, Shawn (BIGKIS) — d6814d1f20eef1e589c2c1c8a2f913dd47913abb
○ Abstain

Vice Chairperson (Max: 1) — Hash Value

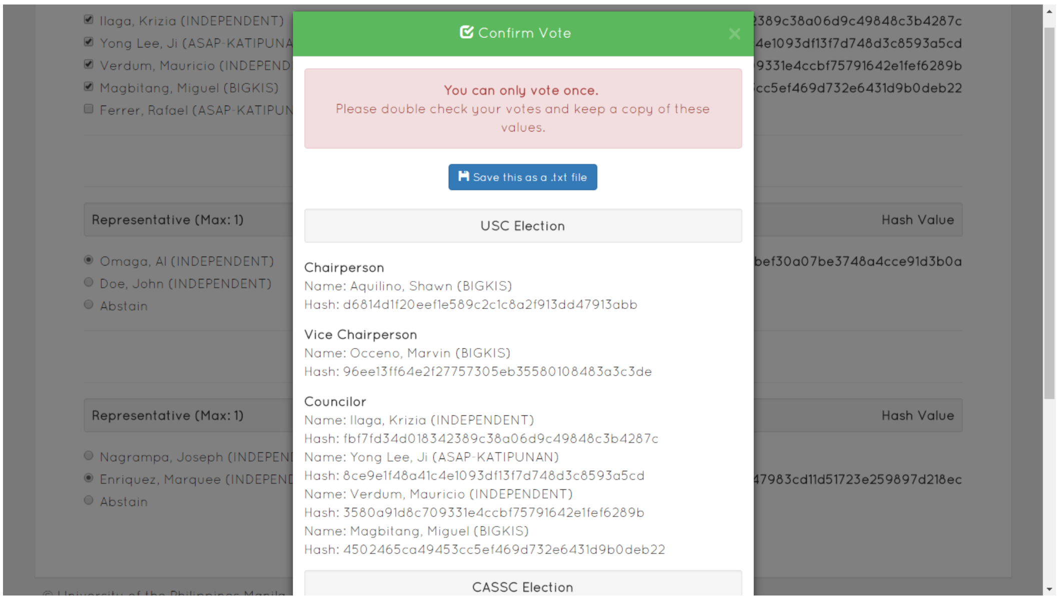
○ Tuazon, Joseph (ASAP-KATIPUNAN)
● Occeno, Marvin (BIGKIS) — 96ee13ff64e2f27757305eb35580108483a3c3de
○ Abstain

Councilor (Max: 4) — Hash Value

☑ Ilaga, Krizia (INDEPENDENT) — fbf7fd34d018342389c38a06d9c49848c3b4287c
☑ Yong Lee, Ji (ASAP-KATIPUNAN) — 8ce9e1f48a41c4e1093df13f7d748d3c8593a5cd
☑ Verdum, Mauricio (INDEPENDENT) — 3580a91d8c709331e4ccbf75791642e1fef6289b
☑ Magbitang, Miguel (BIGKIS) — 4502465ca49453cc5ef469d732e6431d9b0deb22
□ Ferrer, Rafael (ASAP-KATIPUNAN)

Figure 64: Vote page

Once the voter has filled the ballot, he/she can submit it by clicking on the submit button at the bottom of the page. If the voter missed to fill out parts of the ballot or if the voter chose too many candidates for a position, he/she will be notified.


Ilaga, Krizia (INDEPENDENT)
Yong Lee, Ji (ASAP-KATIPUNA
Verdum, Mauricio (INDEPEND
Magbitang, Miguel (BIGKIS)
Ferrer, Rafael (ASAP-KATIPUN
389c38a06d9c49848c3b4287c
4e1093df13f7d748d3c8593a5cd
9331e4ccbf75791642e1fef6289b
cc5ef469d732e6431d9b0deb22
Representative (Max: 1)
Hash Value
Omaga, Al (INDEPENDENT)
Doe, John (INDEPENDENT)
Abstain
bef30a07be3748a4cce91d3b0a
Representative (Max: 1)
Hash Value
Nagrampa, Joseph (INDEPEN
Enriquez, Marquee (INDEPEN
Abstain
47983cd11d51723e259897d218ec
Save this as a .txt file

Confirm Vote

You can only vote once.
Please double check your votes and keep a copy of these values.

USC Election

Chairperson
Name: Aquilino, Shawn (BIGKIS)
Hash: d6814d1f20eef1e589c2c1c8a2f913dd47913abb

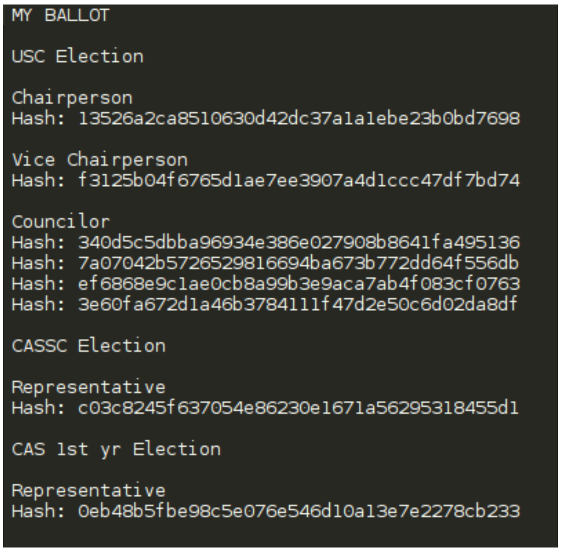
Vice Chairperson
Name: Occeno, Marvin (BIGKIS)
Hash: 96ee13ff64e2f27757305eb35580108483a3c3de

Councilor
Name: Ilaga, Krizia (INDEPENDENT)
Hash: fbf7fd34d018342389c38a06d9c49848c3b4287c
Name: Yong Lee, Ji (ASAP-KATIPUNAN)
Hash: 8ce9e1f48a41c4e1093df13f7d748d3c8593a5cd
Name: Verdum, Mauricio (INDEPENDENT)
Hash: 3580a91d8c709331e4ccbf75791642e1fef6289b
Name: Magbitang, Miguel (BIGKIS)
Hash: 4502465ca49453cc5ef469d732e6431d9b0deb22

CASSC Election

Figure 65: Review ballot

If the ballot is valid, the system shows the voter his/her ballot and gives the option to save it as a text file for future reference.

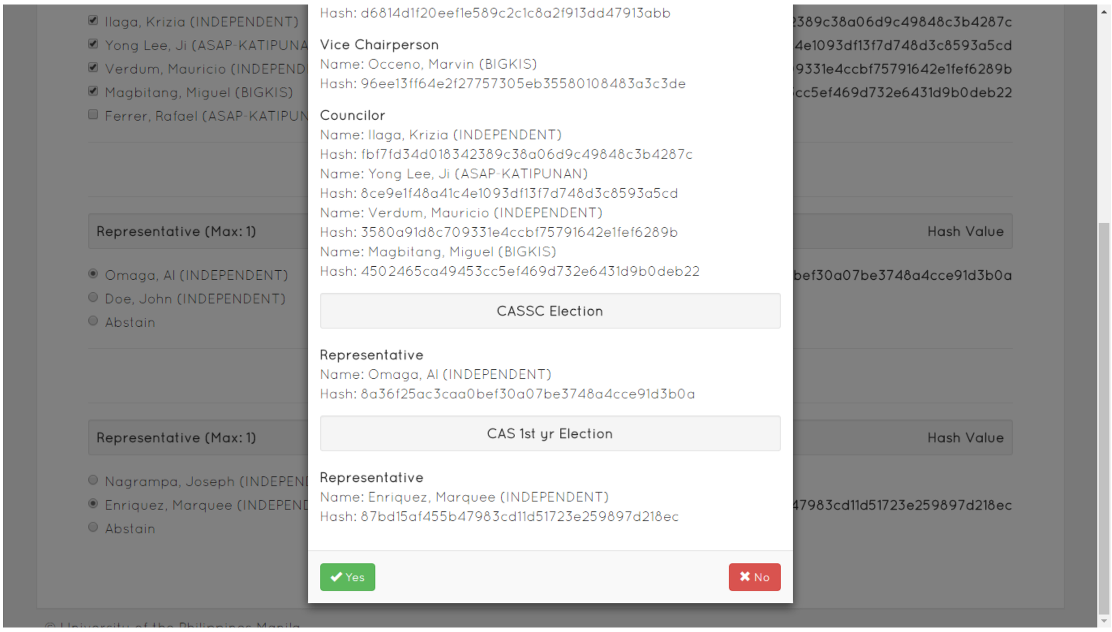


```
MY BALLOT

USC Election

Chairperson
Hash: 13526a2ca8510630d42dc37a1a1ebe23b0bd7698

Vice Chairperson
Hash: f3125b04f6765d1ae7ee3907a4d1ccc47df7bd74

Councilor
Hash: 340d5c5dbba96934e386e027908b8641fa495136
Hash: 7a07042b5726529816694ba673b772dd64f556db
Hash: ef6868e9c1ae0cb8a99b3e9aca7ab4f083cf0763
Hash: 3e60fa672d1a46b3784111f47d2e50c6d02da8df

CASSC Election

Representative
Hash: c03c8245f637054e86230e1671a56295318455d1

CAS 1st yr Election

Representative
Hash: 0eb48b5fbe98c5e076e546d10a13e7e2278cb233
```

Figure 66: Contents of myBallot.txt

If the voter has verified that the ballot is correct. The voter can click yes to send his/her vote.

Note: The voter can only vote once.



Hash: d6814d1f20eef1e589c2c1c8a2f913dd47913abb

Vice Chairperson
Name: Occeno, Marvin (BIGKIS)
Hash: 96ee13ff64e2f27757305eb35580108483a3c3de

Councilor
Name: Ilaga, Krizia (INDEPENDENT)
Hash: fbf7fd34d018342389c38a06d9c49848c3b4287c
Name: Yong Lee, Ji (ASAP-KATIPUNAN)
Hash: 8ce9e1f48a41c4e1093df13f7d748d3c8593a5cd
Name: Verdum, Mauricio (INDEPENDENT)
Hash: 3580a91d8c709331e4ccbf75791642e1fef6289b
Name: Magbitang, Miguel (BIGKIS)
Hash: 4502465ca49453cc5ef469d732e6431d9b0deb22

CASSC Election

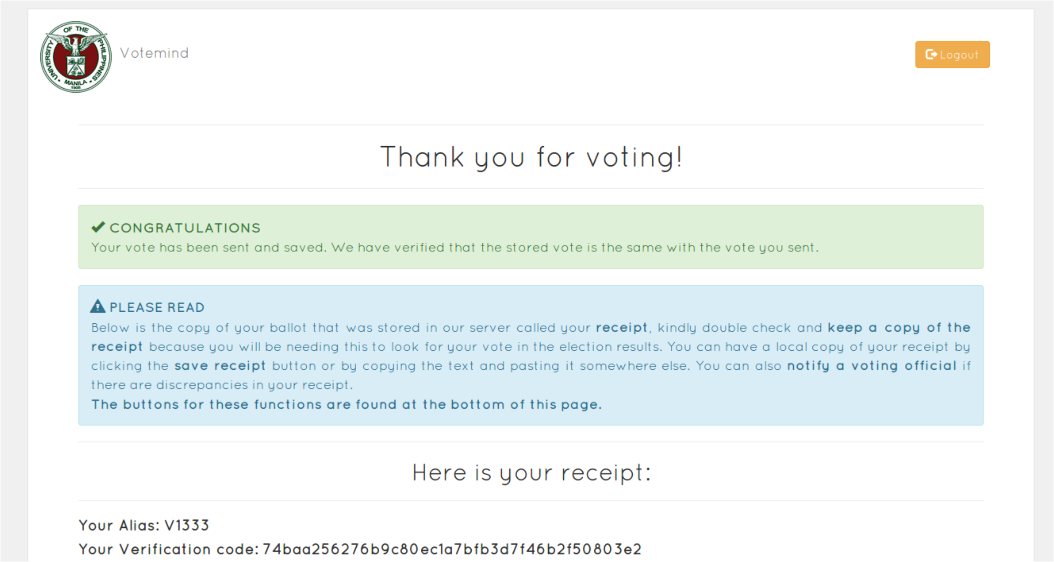
Representative
Name: Omaga, Al (INDEPENDENT)
Hash: 8a36f25ac3caa0bef30a07be3748a4cce91d3b0a

CAS 1st yr Election

Representative
Name: Enriquez, Marquee (INDEPENDENT)
Hash: 87bd15af455b47983cd11d51723e259897d218ec

Figure 67: Submit ballot

Once the server has received the votes, it redirects the voter to another page. The new page gives information about the verification of the ballot and instructions about the receipt.



Thank you for voting!

✓ CONGRATULATIONS
Your vote has been sent and saved. We have verified that the stored vote is the same with the vote you sent.

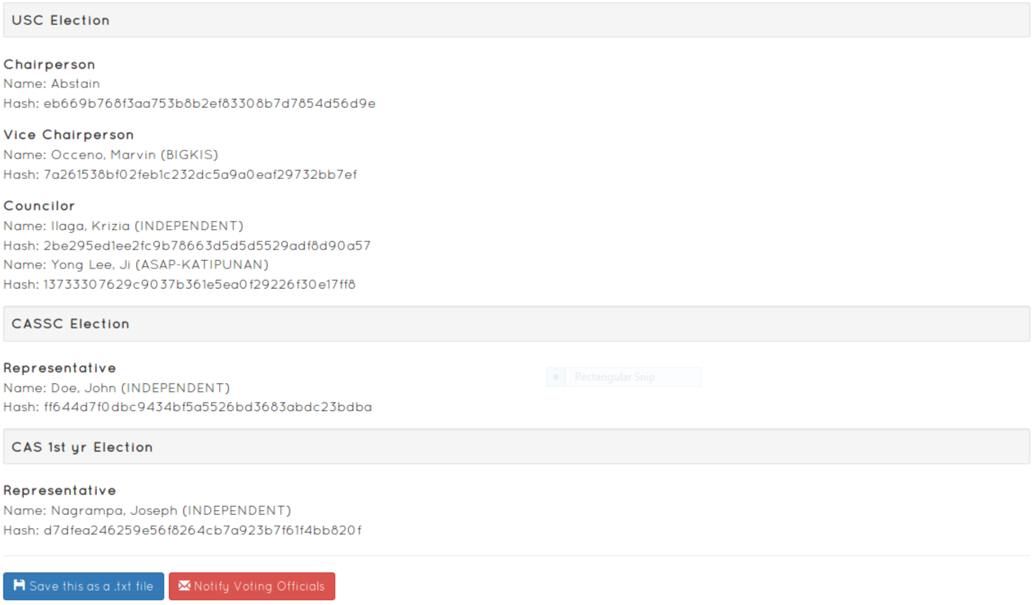
⚠ PLEASE READ
Below is the copy of your ballot that was stored in our server called your **receipt**, kindly double check and **keep a copy of the receipt** because you will be needing this to look for your vote in the election results. You can have a local copy of your receipt by clicking the **save receipt** button or by copying the text and pasting it somewhere else. You can also **notify a voting official** if there are discrepancies in your receipt.
**The buttons for these functions are found at the bottom of this page.**

Here is your receipt:

Your Alias: V1333
Your Verification code: 74baa256276b9c80ec1a7bfb3d7f46b2f50803e2

Figure 68: Vote received page top part

The system will inform the voter if his/her ballot corresponds to the votes stored in the server. The voter can double check by looking at the hash values in the receipt. If there are discrepancies, the voter can notify a voting official.



USC Election

Chairperson
Name: Abstain
Hash: eb669b768f3aa753b8b2ef83308b7d7854d56d9e

Vice Chairperson
Name: Occeno, Marvin (BIGKIS)
Hash: 7a261538bf02feb1c232dc5a9a0eaf29732bb7ef

Councilor
Name: Ilaga, Krizia (INDEPENDENT)
Hash: 2be295ed1ee2fc9b78663d5d5d5529adf8d90a57
Name: Yong Lee, Ji (ASAP-KATIPUNAN)
Hash: 13733307629c9037b361e5ea0f29226f30e17ff8

CASSC Election

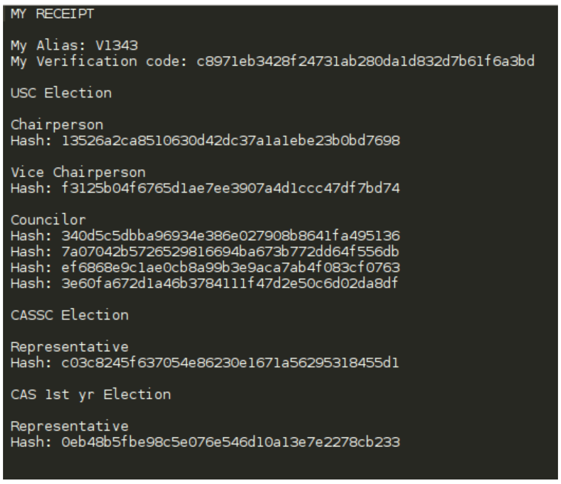
Representative
Name: Doe, John (INDEPENDENT)
Hash: ff644d7f0dbc9434bf5a5526bd3683abdc23bdba

CAS 1st yr Election

Representative
Name: Nagrampa, Joseph (INDEPENDENT)
Hash: d7dfea246259e56f8264cb7a923b7f61f4bb820f

Figure 69: Vote received page bottom part

It is recommended that the voter keeps a local copy of the receipt by saving it as a text file.

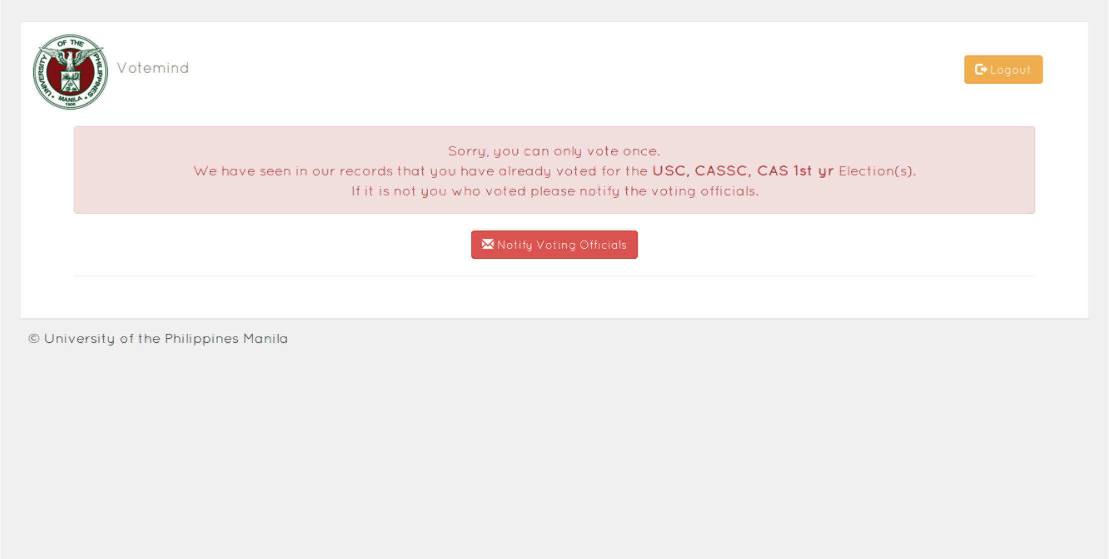


```
MY RECEIPT

My Alias: V1343
My Verification code: c8971eb3428f24731ab280da1d832d7b61f6a3bd

USC Election

Chairperson
Hash: 13526a2ca8510630d42dc37a1a1ebe23b0bd7698

Vice Chairperson
Hash: f3125b04f6765d1ae7ee3907a4d1ccc47df7bd74

Councilor
Hash: 340d5c5dbba96934e386e027908b8641fa495136
Hash: 7a07042b5726529816694ba673b772dd64f556db
Hash: ef6868e9c1ae0cb8a99b3e9aca7ab4f083cf0763
Hash: 3e60fa672d1a46b3784111f47d2e50c6d02da8df

CASSC Election

Representative
Hash: c03c8245f637054e86230e1671a56295318455d1

CAS 1st yr Election

Representative
Hash: 0eb48b5fbe98c5e076e546d10a13e7e2278cb233
```

Figure 70: Contents of myReceipt.txt

Once the voter has checked all values and verified that it is correct or notified the voting officials of any discrepancies, he/she can now log out and wait for the results of the election.

If the voter tries to access his/her account again, the system will redirect the voter to a page informing the voter that he/she already voted.



Sorry, you can only vote once.
We have seen in our records that you have already voted for the **USC, CASSC, CAS 1st yr** Election(s).
If it is not you who voted please notify the voting officials.

© University of the Philippines Manila

Figure 71: Voted info page

# VI. Discussions

Votemind, an Internet Voting System, is a web-based application that is accessible on any computer that has Internet and a Javascript enabled browser. It fulfills the main requirements of an election, which are privacy and verifiability. It achieves this by applying Secure Multiparty Computation using the Sharemind 3 framework. The Sharemind miners are used to store the votes, to count the votes, and to retrieve the votes' hashes for the bulletin board.

The system has 4 roles: public user, voter, voting official, and administrator. The public user can view the results and the bulletin board of the elections for verification. If they see any discrepancy in the hashes, they can notify the voting officials. Voters can vote and keep a copy of their receipt. Voting officials can configure the election and manage the list of voters. They also manage the notifications sent by the public users or voters. Administrators manage the list of voting officials. Both voting officials and administrators have the function to manage their account and change their passwords.

Because of the use of the Sharemind miners for storage, the votes are considered private and secure. They are split at the browser with the use of the Sharemind Javascript library and sent to the 3 independent node.js servers that communicate with each of the Sharemind miners. The data received by the Sharemind miners are partial and encrypted. Sharemind can handle one semi-honest corrupt miner, which means that no data is leaked unless two miners collude. [22]

```
/*app_save_vote.sc: How sent encrypted values are saved*/
for (uint i = 0; i < size(candidateId); ++i) {
    table = arrayToString(declassify(electionId[i]));
    tdbInsertRow(datasource, table, {voterId, positionId[i],
       candidateId[i]});
    print("ROW INSERTED");
    message = bl_str(arrayToString(declassify(voterId))+
       arrayToString(declassify(positionId[i]))+arrayToString(
       declassify(candidateId[i])));
    hash[i] = CRC32(message); // these values will be returned
       by the miners when the insertion is completed
    }
}
```

Votemind is able to eliminate the need of trusting an intermediary server when sending the votes by using the Javascript controller library provided by Sharemind. Using the library, the system is able to encrypt and split the votes at the browser and send it to the miners. When the votes are sent, the data is received by the Node.js servers on top of the miners. These servers are independent and they are responsible for sending the encrypted and partial data to the miners. They also collect outputs from miners that are sent to the client's browser.

```
/*App.js: How the values are encrypted at the browser and split
    to the 3 miners*/

//Candidate Ids are being encrypted
var pub_value_c = new pub.Uint32Array(candidateId.length);
for(var i=0; i<candidateId.length; i++){
  pub_value_c.set(i, new BigInteger(’’+candidateId[i], 10));
}
var private_value_c = new priv.Uint32Array(pub_value_c);

//Preparing arguments to be sent and specifying the SecreC
   script to be run
var args = {};
args["proxyParams"] = {"codefile": "app_save_vote.sb"}
args["smParams"] = {"electionId": private_value_e, "voterId":
   private_value_v, "candidateId": private_value_c, "positionId
   ": private_value_p};

//Splitting up the votes and sending them to the 3 miners
//This is function is defined at the sharemind controller
servers.emit("run_code", args, function (data) {
     app.log("Successfully saved vote.");
     $(window).trigger("allDone");
     // Trigger callback:
      if (callback) callback(’ok’,data);
 });
});
```

Verifiability is fulfilled with the use of the bulletin board. A user can check if his/her vote was recorded properly by checking if the posted hash values are the same with his/her receipt hash values. The hashes are computed whenever the bulletin board is viewed, ensuring that the hashes are directly from the Sharemind miners. When the bulletin board of a specific election is viewed, the browser of the user requests the hash values stored in that election from the Sharemind database using the Javascript controller. When the miners receive the request, they get the values of the rows in the table/election specified. After the values are retrieved, they generate the hashes by concatinating the voter id, position id and candidate id of each row and using CRC32. CRC32 is used because among the few hashing functions in Sharemind because it does not require a seed and it is the only hashing function that returns a non-array hash value, which is necessary because it needs to be converted to a string and returned to the user.

```
//SecreC code for generating hashes in Sharemind miners
for (uint i = 0; i < size(candidateId); ++i) {
   message=bl_str(arrayToString(declassify(voterId))+
      arrayToString(declassify(positionId[i]))+arrayToString(
      declassify(candidateId[i])));
   hash[i] = CRC32(message);
}
publish("hashes", declassify(reshare(hash)));
```

When all rows are hashed, the values are returned to the browser, hashed again using SHA1 before being posted in the bulletin board. It is hashed again in SHA1 to lessen the chances of it being cracked.

```
//Javascript code for hashing the received results in SHA1
for(i=0;i<results['hashes'].length;i++){
   hashes[i] = CryptoJS.SHA1(""+results['hashes'][i]);
}
```

Although this may be effective, it is not efficient because computing for the hashes takes time, especially when there are a large amount of votes. The results of the election are only computed once to lessen the loading time when being viewed. It is still computed using the Sharemind miners but the results are stored in the MySQL database for faster loading.

```
/*app_get_histogram.sc: how results are computed*/
// Create vector where the votes repeat nrOfCandidates times:
pd_shared3p uint32[[1]] exVotes (nrOfVotes*nrOfCandidates);
for (uint i = 0; i < nrOfCandidates; i++) {
        exVotes[i*nrOfVotes:(i+1)*nrOfVotes] = votes;
}
assert(size(exVotes) == nrOfVotes*nrOfCandidates);

// Create a comparison vector, where each
// candidate id is repeated nrOfVotes times:
```

```
pd_shared3p uint32[[1]] exCandidates (nrOfVotes*nrOfCandidates);
for (uint i = 0; i < nrOfCandidates; i++) {
exCandidates[i*nrOfVotes:(i+1)*nrOfVotes] = (uint32)(candidateId[i]);
}
assert(size(exCandidates) == nrOfVotes*nrOfCandidates);

// Do equality check:
pd_shared3p bool[[1]] eq = (exVotes == exCandidates);
pd_shared3p uint[[1]] sums = sum(eq, nrOfCandidates);
public uint[[1]] psums = declassify(sums);

//printVector(psums);
publish("histogram", psums);
```

In order for the Javascript controller to work, the IP addresses of the Node.js servers and miners should be known so that they can be configured. The configuration of the miner.js file ,which contains the IP addresses of the Node.js servers, is done during installation.

```
//Miners.js
(function(app, sm, $){
  app.hosts = [
    "http://126.172.16.2:8081",
    "http://126.172.16.3:8082",
    "http://126.172.16.4:8083"
  ];
})(this.app = this.app === undefined ? {} : this.app, sm,
    jQuery);
```

The 3 Node.js servers also need to be configured before being run. To configure them, their IP addresses should be known beforehand. The following files need to be configured for each Node.js server:

```
#run_servers.sh
#If this Node.js server is server1
#If its IP address is 126.172.16.2
#!/bin/sh
nodejs server.js Server1 server1.cfg 8081 126.172.16.2 &
#nodejs server.js Server2 server2.cfg 8082 &
#nodejs server.js Server3 server3.cfg 8083 &
wait

;server1.cfg
[MinerNode1]
; Assumption: Miner and Node.js is at the same machine
; Address of the Miner
Address=126.172.16.2
Port=30001
```

The miners also need to be configured in order for them to see their corresponding Node.js server and their fellow miners. The following files need to be configured:

```
;change these values in server.cfg of miner1
ListenInterfaces=126.172.16.2:30001 [::1]:30001

; -----------------------------------------------------
; The server address book
; Describes the addresses and identities of servers on the network
; -----------------------------------------------------
[Server 1]
Address = 126.172.16.2
Port = 30001
PublicIdentity = server1-public-key
;OutgoingTlsPriorities = NONE:+CTYPE-X509:+VERS-TLS1.2:+AES-256-GCM:+
   ECDHE-RSA:+AEAD:+ECDHE-RSA:+COMP-NULL:+SIGN-RSA-SHA512:+CURVE-
   SECP521R1
;MaxPayloadSize = 131059

[Server 2]
Address =126.172.16.3
Port = 30002
PublicIdentity = server2-public-key
;OutgoingTlsPriorities = NONE:+CTYPE-X509:+VERS-TLS1.2:+AES-256-GCM:+
   ECDHE-RSA:+AEAD:+ECDHE-RSA:+COMP-NULL:+SIGN-RSA-SHA512:+CURVE-
   SECP521R1
;MaxPayloadSize = 131059

[Server 3]
Address =126.172.16.4
Port = 30003
PublicIdentity = server3-public-key
;OutgoingTlsPriorities = NONE:+CTYPE-X509:+VERS-TLS1.2:+AES-256-GCM:+
   ECDHE-RSA:+AEAD:+ECDHE-RSA:+COMP-NULL:+SIGN-RSA-SHA512:+CURVE-
   SECP521R1
;MaxPayloadSize = 131059
```

When a miner's database becomes corrupted, it can be simply brought back if the miner has a back up copy. Administrators can regularly back up the miner's database by simply copying the database files and pasting it somewhere else. When the previous data is needed, the admins can simply copy the old back up files and paste it in the miner's database.

Miners secure a TLS connection with each other and their Node.js servers. An administrator can generate their own certificates using these commands:

```
openssl req -x509 -days 3650 -nodes -newkey rsa:4096 -keyout
   serverX-private-key -out serverX-public-key -outform der;
```

```
openssl rsa -in serverX-private-key -out serverX-private-key -
   outform der
```

When the certificates are generated, each miner must share to the other miners and Node.js server their serverX-public-key and uncomment the OutgoingTlsPriorities parts of their server.cfg files. Node.js servers can also do this, and when they have generated their own certificates, they must share to their miners their client-public-key files. Also, in order for the TLS connection to work they must uncomment the OutgoingTlsPriorities part of the serverX.cfg of the miner assigned to them.

An issue that was encountered during development was the incapability of the miners to multi-thread. They were only able to run one script at a time, implying that voters can only vote one at a time. Another issue is that the SecreC of Sharemind 3 does not allow row editing or row deletion on its tables, that is why voters are not allowed to revote.

# VII. Conclusions

Votemind, an Internet Voting System, is a web-based application that fulfills the privacy and verifiability requirements of an election. It was achieved by applying Secure Multiparty Computation using the Sharemind framework and JavaScript library. The votes are encrypted and split at the browser then stored in the Sharemind miners to ensure privacy and remove the need to trust an intermediary server. The bulletin board values are directly pulled from the Sharemind miners to ensure that the values can be used for verification. Votemind is accessible from any Javascript enabled browser that is connected to the Internet.

The system allows a public user to view the results and bulletin board of the election, and to notify voting officials about discrepancies in the hashe values of the votes. It allows voters to vote and keep a copy of their receipts. Voting officials can manage the election configuration and the voters list. Administrators can manage the list of voting officials.

# VIII. Recommendations

It is highly recommended that the Javascript files be obscured when Votemind is deployed in a real setting. JavaScript Obfuscator and Jscrambler are samples of available obfuscators in the Internet that can be used. This is to prevent the users from understanding and replacing the code of Sharemind's Javascript controller with malicious code. Also, it is recommended that the Sharemind miners be configured to receive incoming requests simultaneously. So that when it is deployed, voters can vote simultaneously. The miner's databases should also be regularly backed up in case of failures. Node.js servers should be monitored by other applications so that when it crashes, it will automatically be rebooted.

# IX. Bibliography

[1] R. Talviste, “Web-based data entry in privacy-preserving applications,” Undergraduate Thesis, University of Tartu, Tartu, Estonia, 2009.

[2] V. Cortier, “Formal verification of e-voting: Solutions and challenges,” *ACM SIGLOG News*, vol. 2, pp. 25–34, Jan. 2015.

[3] R. Kofler, R. Krimmer, and A. Prosser, “Electronic voting: algorithmic and implementation issues,” in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pp. 7 pp.–, Jan 2003.

[4] D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. Halderman, “Security analysis of the Estonian internet voting system,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’14, (New York, NY, USA), pp. 703–715, ACM, 2014.

[5] M. J. M. Chowdhury, “Article: Comparison of e-voting schemes: Estonian and Norwegian solutions,” *International Journal of Applied Information Systems*, vol. 6, pp. 60–66, September 2013. Published by Foundation of Computer Science, New York, USA.

[6] C. Culnane, P. Y. A. Ryan, S. Schneider, and V. Teague, “vvote: A verifiable voting system,” *ACM Trans. Inf. Syst. Secur.*, vol. 18, pp. 3:1–3:30, June 2015.

[7] B. Adida, O. De Marneffe, O. Pereira, and J.-J. Quisquater, “Electing a university president using open-audit voting: Analysis of real-world use of helios,” in *Proceedings of the 2009 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE’09, (Berkeley, CA, USA), pp. 10–10, USENIX Association, 2009.

[8] R. Talviste, "Deploying secure multiparty computation for joint data analysis—a case study," Master's thesis, Institute of Computer Science, University of Tartu, 2011.

[9] D. Mishra, R. Pathak, S. Joshi, and A. Ludhiyani, "Secure multi-party computation for statistical computations using virtual parties on a token ring network," in *Wireless And Optical Communications Networks (WOCN), 2010 Seventh International Conference On*, pp. 1–6, Sept 2010.

[10] R. Cramer, I. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing - An Information Theoretic Approach*. 2013.

[11] J. A. Piscos, "Secure multiparty computation for internet voting," Undergraduate Thesis, University of the Philippines Manila, Manila, Philippines, 2015.

[12] A. R.Kusters, T.Truderung, "Clash Attacks on the Verifiability of E-Voting Systems," in *IEEE Symposium on Security and Privacy*, pp. 395–409, IEEE Computer Society, 2012.

[13] "E-voting experiments end in norway amid security fears." http://www.bbc.com/news/technology-280556781. Accessed: 2015-12-05.

[14] B. Adida, "Helios: Web-based open-audit voting," in *Proceedings of the 17th Conference on Security Symposium*, SS'08, (Berkeley, CA, USA), pp. 335–348, USENIX Association, 2008.

[15] M. W. Øberg, "Improving the norwegian internet voting protocol," 2011.

[16] R. Pathak, S. Joshi, D. Mishra, and A. Ludhiyani, "Tri-TTP based architecture for secure multi-party computations using virtual parties," in *Wireless And Optical Communications Networks (WOCN), 2010 Seventh International Conference On*, pp. 1–6, Sept 2010.

[17] A. Ben-David, N. Nisan, and B. Pinkas, "FairplayMP: A system for secure multi-party computation," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, CCS '08, (New York, NY, USA), pp. 257–266, ACM, 2008.

[18] D. Beaver, S. Micali, and P. Rogaway, "The round complexity of secure protocols," in *Proceedings of the Twenty-second Annual ACM Symposium on Theory of Computing*, STOC '90, (New York, NY, USA), pp. 503–513, ACM, 1990.

[19] A. Jarrous and B. Pinkas, "Canon-MPC, a system for casual non-interactive secure multi-party computation using native client," in *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*, WPES '13, (New York, NY, USA), pp. 155–166, ACM, 2013.

[20] S. Halevi, Y. Lindell, and B. Pinkas, "Secure computation on the web: Computing without simultaneous interaction," in *Proceedings of the 31st Annual Conference on Advances in Cryptology*, CRYPTO'11, (Berlin, Heidelberg), pp. 132–150, Springer-Verlag, 2011.

[21] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol version 1.2." https://tools.ietf.org/html/rfc5246. Accessed: 2015-12-21.

[22] D. Bogdanov, S. Laur, and J. Willemson, "Sharemind: A Framework for Fast Privacy-Preserving Computations," in *Proceedings of the 13th European Symposium on Research in Computer Security - ESORICS'08* (S. Jajodia and J. Lopez, eds.), vol. 5283 of *Lecture Notes in Computer Science*, pp. 192–206, Springer Berlin / Heidelberg, 2008.

[23] J. A. Staub, "An analysis of Chaum's voter-verifiable election scheme," Master's thesis, Graduate School of the University of Maryland, College Park, 2005.

[24] A. Kiayias, M. Korman, and D. Walluck, "An internet voting system supporting user privacy," in *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, pp. 165–174, Dec 2006.

[25] K. B. Frikken, "Algorithms and theory of computation handbook," ch. Secure Multiparty Computation, pp. 14–14, Chapman & Hall/CRC, 2010.

[26] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, (New York, NY, USA), pp. 218–229, ACM, 1987.

[27] I. Damgård and Y. Ishai, "Constant-round multiparty computation using a black-box pseudorandom generator," in *Proceedings of the 25th Annual International Conference on Advances in Cryptology*, CRYPTO'05, (Berlin, Heidelberg), pp. 378–394, Springer-Verlag, 2005.

[28] "Sharemind: Your secure service platform for data collection and analysis." http://cyber.ee/uploads/2013/06/Sharemind_NEW_A4_web.pdf. Accessed: 2016-01-13.

[29] J. R. Dan Bogdanov, Peeter Laud, "Secrec for sharemind 3." http://kodu.ut.ee/~varmo/tday-otepaa/randmets-slides.pdf, 2013.

[30] D. Bogdanov, *Sharemind: programmable secure computations with practical applications*. PhD thesis, University of Tartu, 2013.

# X. Appendix

## A. Source Code

### A..1 Sharemind

1. app_save_vote.sc

```
/*
 * This file is a part of the Sharemind
     framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

import shared3p;
import stdlib;
import table_database;
import shared3p_table_database;
import shared3p_string;
import shared3p_aes;

domain pd_shared3p shared3p;

void main() {

  public string datasource = "DS1";
  // Open the database connection
  tdbOpenConnection(datasource);

  pd_shared3p uint32[[1]] electionId =
      argument("electionId");
  pd_shared3p uint32 voterId = argument("
      voterId");
  pd_shared3p uint32[[1]] positionId =
      argument("positionId");
  pd_shared3p uint32[[1]] candidateId =
      argument("candidateId");

  pd_shared3p xor_uint8[[1]] message;
  pd_shared3p xor_uint32[[1]] hash (size(
      candidateId));

  public string table = "";

   for (uint i = 0; i < size(candidateId); ++i
       ) {

        table = arrayToString(declassify(
            electionId[i]));

        tdbInsertRow(datasource, table, {
            voterId, positionId[i],
            candidateId[i]});
        print("ROW INSERTED");
        message = bl_str(arrayToString(
            declassify(voterId))+
            arrayToString(declassify(
            positionId[i]))+arrayToString(
            declassify(candidateId[i])));
        hash[i] = CRC32(message);
   }

  //tdbVmapAddValue(params_dbody, "values",
      positionId);
  //tdbVmapAddValue(params_dbody, "values",
      candidateId);
  //tdbVmapAddValue(params_dbody, "values",
      vote);

  // Check whether the tables exist
  // If not, create it
//  if (!tdbTableExists(datasource, table)) {
//    pd_shared3p uint8 nuint8;
//    pd_shared3p uint32 nuint32;

//    // Create a parameter map.
//    uint64 params = tdbVmapNew();

//    // voterId
//    tdbVmapAddType(params, "types", nuint8);
//    tdbVmapAddString(params, "names", "0");

//    // positionId
//    tdbVmapAddType(params, "types", nuint8);
//    tdbVmapAddString(params, "names", "1");

//    // candidateId
//    tdbVmapAddType(params, "types", nuint8);
//    tdbVmapAddString(params, "names", "2");

//    // Create the table.
//    tdbTableCreate(datasource, table, params
    );

//    // Clean up the parameters.
//    tdbVmapDelete(params);
//  }

//  // Save declaration body.
//  tdbInsertRow(datasource, table,
    params_dbody);

//  // Clean up the parameters
//  tdbVmapDelete(params_dbody);

  tdbCloseConnection(datasource);

  publish("hashes", declassify(reshare(hash)))
      ;
}
```

2. app_get_histogram.sc

```
/*
 * This file is a part of the Sharemind
     framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

import shared3p;
import stdlib;
import table_database;
import shared3p_table_database;

domain pd_shared3p shared3p;

void main() {

  public string datasource = "DS1";
  public uint32 electionId = argument("
      electionId");
  public uint32[[1]] positionId = argument("
      positionId");
  public uint32[[1]] candidateId = argument
      ("candidateId");
  public string table = arrayToString(
      electionId);

  public uint nrOfCandidates = size(
      candidateId);
  print("nrOfCandidates:",nrOfCandidates);

  // Open the database connection
  tdbOpenConnection(datasource);

  // Check whether the tables exist.
  assert(tdbTableExists(datasource, table));

  pd_shared3p uint32[[1]] votes =
        tdbReadColumn(datasource, table, (
        uint64)2); // Column '2' holds
        candidate id
    print("votes:", _vectorToString(declassify
        (votes)));

    public uint nrOfVotes = size(votes);
    //print(nrOfVotes);

        // Copy vector nrOfCandidates times:
        pd_shared3p uint32[[1]] exVotes (
            nrOfVotes*nrOfCandidates);
        for (uint i = 0; i < nrOfCandidates; i
            ++) {
                exVotes[i*nrOfVotes:(i+1)*
                    nrOfVotes] = votes;
        }
        assert(size(exVotes) == nrOfVotes*
            nrOfCandidates);

        // Create a comparison vector, where
            each
        // candidate id is repeated nrOfVotes
            times:
        pd_shared3p uint32[[1]] exCandidates (
            nrOfVotes*nrOfCandidates);
        for (uint i = 0; i < nrOfCandidates; i
            ++) {
                exCandidates[i*nrOfVotes:(i+1)
                    *nrOfVotes] = (uint32)(
                    candidateId[i]); //this
                    is candidateId
        }
        assert(size(exCandidates) == nrOfVotes
            *nrOfCandidates);

        // Do equality check:
        pd_shared3p bool[[1]] eq = (exVotes ==
             exCandidates);
        pd_shared3p uint[[1]] sums = sum(eq,
            nrOfCandidates);
        public uint[[1]] psums = declassify(
            sums);

        //printVector(psums);
        publish("histogram", psums);
}
```

3. app_delete_table_vm.sc

```
/*
 * This file is a part of the Sharemind
     framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

import shared3p;
import stdlib;
import table_database;
import shared3p_table_database;

domain pd_shared3p shared3p;

void main() {
    public uint32 electionId = argument("
        electionId");
    //public uint electionId = 1;

    // Open the database connection to
        datasource_main
    public string datasource = "DS1";
    tdbOpenConnection(datasource);

    public string table = arrayToString(
        electionId);

    print("DELETING TABLE: ", electionId);
    if(tdbTableExists(datasource, table)){
        tdbTableDelete(datasource, table);
        print("TABLE DELETED: ", electionId);
    }

    // Close the database connection.
    tdbCloseConnection(datasource);
}
```

4. app_get_hashes.sc

```
/*
 * This file is a part of the Sharemind
     framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

import shared3p;
import stdlib;
import table_database;
import shared3p_table_database;
import shared3p_string;
import shared3p_aes;

domain pd_shared3p shared3p;

void main() {

  public string datasource = "DS1";
  // Open the database connection
  tdbOpenConnection(datasource);

  public uint32 electionId = argument("
      electionId");
  public string table = arrayToString(
      electionId);

  assert(tdbTableExists(datasource, table));

  if(tdbTableExists(datasource, table)){
      pd_shared3p uint32[[1]] voterId =
          tdbReadColumn(datasource, table, (
          uint64)0); // Column '0' holds
          voter id
    publish("voterIds", declassify(voterId))
        ;
    pd_shared3p uint32[[1]] positionId =
        tdbReadColumn(datasource, table, (
        uint64)1); // Column '1' holds
        position id
    publish("positionIds", declassify(
        positionId));
    pd_shared3p uint32[[1]] candidateId =
        tdbReadColumn(datasource, table, (
        uint64)2); // Column '2' holds
        candidate id

    pd_shared3p xor_uint8[[1]] message;
    pd_shared3p xor_uint32[[1]] hash (size(
        candidateId));

    for (uint i = 0; i < size(candidateId);
        ++i) {
      message = bl_str(arrayToString(
          declassify(voterId))+
          arrayToString(declassify(
          positionId[i]))+arrayToString(
          declassify(candidateId[i])));
      hash[i] = CRC32(message);
    }

    publish("hashes", declassify(reshare(
        hash)));
  }

  tdbCloseConnection(datasource);
}
```

5. get_random.sc

```
/*
 * This file is a part of the EMTA KMD INF
     application on Sharemind framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

import shared3p;
import stdlib;
import shared3p_random;

domain pd_shared3p shared3p;
```

```
void main() {
    // Create ”bytes” nr of random bytes and
        return them.
    // Example: Create 32 bytes (uint8 = 8 bit
         = 1 byte x 32) => uint8[[1]] rnd
        (32).

    uint32 bytes_cnt = argument(”bytes”);

    pd_shared3p uint8[[1]] rnd ((uint)
         bytes_cnt);

    rnd = randomize(rnd);

    publish(”random”, rnd);
}
```

## A..2 CodeIgniter

1. Controllers

(a) Login.php

```
<?php if ( ! defined(’BASEPATH’)) exit(’No
     direct script access allowed’);

class Login extends CI_Controller {

 function __construct()
 {
    parent::__construct();
    if(!$this->config->item(’installed’)){
      redirect(’install’);
    }else{
      if(file_exists(”install/index.php”)){
        $this->load->helper(”file”); // load
            the helper
        delete_files(”install/assets”);
        delete_files(”install/config”);
        delete_files(”install/includes”);
        rmdir(”install/assets”);
        rmdir(”install/config”);
        rmdir(”install/includes”);
        delete_files(”install”);
      }
    }
    $this->load->model(’login_model’,’’,TRUE);
    $this->load->model(’votingofficial_model
         ’,’’,TRUE);
 }

 function index()
 {
    //This method will have the credentials
         validation
    $this->load->library(’form_validation’);
    $this->form_validation->set_rules(’
         username’, ’Username’, ’trim|required
         ’);
    $this->form_validation->set_rules(’
         password’, ’Password’, ’trim|required
         |callback_check_database’);

    if($this->form_validation->run() == FALSE)
    {
      //Field validation failed. User
           redirected to login page
      $this->load->view(’templates/header’);
      $this->load->view(’login/navbar’);
        $this->load->view(’login/view’);
          $this->load->view(’templates/footer
               ’);
    }
    else
    {
      //Go to private area
      $session_data = $this->session->userdata
           (’logged_in’);
      $role = $session_data[’role’];
      if($role==’voter’) $role .= ’/vote’;
      else if($role==’votingofficial’) $role
           .= ’/elections’;
      else if($role==’admin’) $role .= ’/vos’;
      redirect($role, ’refresh’);
    }
 }

 function install(){

 }

  function bulletin($function=””){
```

```
    if($function==”getBulletin”){

      $election_id = $this->input->post(’
          election_id’);

      if(!empty($election_id) && $election_id
          != ’None’){

        $this->session->set_userdata(’
            election_id’, $election_id);

        $preVoterIds = $this->input->post(’
            voterIds’);
        $positionIds = $this->input->post(’
            positionIds’);
        // $hashes = $this->input->post(’
            hashes’);


        $voterIds[0] = $this->login_model->
            getVoterAliasAndCode($preVoterIds
            [0]);
        $voterIds[0][’positions’] = array(
            $this->login_model->
            getPositionName($positionIds[0]))
            ;
        // $voterIds[0][’hashes’] = array(
            $hashes[0]);

        for($i=1,$j=0; $i<count($preVoterIds);
            $i++){
          if($preVoterIds[$i] == $preVoterIds[
              $i-1]){
            array_push($voterIds[$j][’
                positions’], $this->
                login_model->getPositionName(
                $positionIds[$i]));
            // array_push($voterIds[$j][’
                hashes’], $hashes[$i]);
          }else{
            $j++;
            $voterIds[$j] = $this->login_model
                ->getVoterAliasAndCode(
                $preVoterIds[$i]);
            $voterIds[$j][’positions’] = array
                ($this->login_model->
                getPositionName($positionIds[
                $i]));
            // $voterIds[$j][’hashes’] = array
                ($hashes[$i]);
          }
        }

        // print_r($voterIds);
        $data[’voters’] = $voterIds;

        //$data[’election_id’] = $election_id;
        //$data[’voters’] =  $this->
            login_model->getAliasAndCode(
            $election_id);
        //$data[’positions’] = $this->
            votingofficial_model->
            getPositions($election_id);
        $data[’election_name’] = $this->
            votingofficial_model->
            getElectionName($election_id);

        $this->load->view(’login/bulletin/
            table’, $data);
        $this->load->view(’login/bulletin/
            table_script',$data);
        $this->load->view('login/bulletin/
            receipt_modal',$data);
        $this->load->view('voter/notify_modal
            ');
      }

    }else{

      $election_id = $this->session->userdata
          ('election_id');
      $data['elections'] = $this->login_model
          ->getFinishedElections();

      if(!empty($election_id) && $election_id
          != 'None'){
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
      }

      $this->load->view('templates/header');
      $this->load->view('login/navbar');
      $this->load->view('login/bulletin/
          bulletin',$data);
      $this->load->view('login/bulletin/footer
          ');
    }
  }

  function notify(){

    $this->load->helper('date');

    $title = $this->input->post('title');
    $body = $this->input->post('body');
    $datestring = "%Y-%m-%d";
    $time = time();
    $date = mdate($datestring, $time);

    if($this->login_model->sendNotif($title,
        $body, $date)){
      echo "sent";
    }

  }


  function results($function=""){

    if($function=="getResults"){

      $election_id = $this->input->post('
          election_id');

      if(!empty($election_id) && $election_id
          != 'None'){
        $this->session->set_userdata('
            election_id', $election_id);
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
        $data['candidates'] = $this->
            votingofficial_model->
            getCandidates($election_id);

        $this->load->view('login/results/table
            ',$data);
      }
    }else{
      $election_id = $this->session->userdata
          ('election_id');
      $data['elections'] = $this->login_model
          ->getFinishedElections();

      if(!empty($election_id) && $election_id
          != 'None'){
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
      }

      $this->load->view('templates/header');
      $this->load->view('login/navbar');
      $this->load->view('login/results/results
          ',$data);
      $this->load->view('login/results/footer
          ');
    }
  }

  function check_database($password)
  {
    //Field validation succeeded.  Validate
        against database
    $username = $this->input->post('username');

    //query the database
    $result = $this->login_model->login(
        $username, $password);

    if($result)
    {
      $sess_array = array();
      foreach($result as $row)
      {
        $sess_array = array(
          'user_id' => $row->user_id,
          'username' => $row->username,
          'role' => $row->role,
          'email' => $row->email
        );
        $this->session->set_userdata('logged_in
            ', $sess_array);
      }
      return TRUE;
    }
    else
    {
      $this->form_validation->set_message('
          check_database', 'Invalid username
          or password');
      return false;
    }
  }
}

?>
```

(b) Admin.php

```
<?php if ( ! defined('BASEPATH')) exit('No
    direct script access allowed');

// session_start(); //we need to call PHP's
    session object to access it through CI

class Admin extends CI_Controller{

  function __construct(){
    parent::__construct();
    $this->load->model('admin_model','',TRUE);
  }

  function vos($function=""){
    if($this->session->userdata('logged_in')&&
         $this->session->userdata('logged_in
         ')['role'] == 'admin'){

      $session_data = $this->session->userdata
          ('logged_in');
      $data['username'] = $session_data['
          username']; // for future use

      if($this->input->post() || !empty(
          $_FILES['userfile']['name']))
      {
        if($function=='add'){
          $this->load->helper('string');
          $username = $this->input->post('
              username');
          $email = $this->input->post('email')
              ;
          $salt = random_string('alnum', 8);
          $password = random_string('alnum',
              8);
          //$password = "12345";

          if($this->admin_model->insert(
              $username, $email, $password,
              $salt) != true){
            $data['error'] = '<strong>Error!</
                strong> Add Voter
                unsuccessful.';
          }else{
            //write_file('application/uploads/
                passwords.txt', "\n Username
                :".$username.',Password:'.
                $password, "a+");
             if(!$this->email($username,
                 $password,$email))
               $data['error'] = '<strong>Error
                   !</strong> Email Voter
                   unsuccessful.';
          }

        }else if($function=='edit'){
```

```
      $userid = trim($this->input->post('
          userid'));
      $username = $this->input->post('
          username');
      $email = $this->input->post('email')
          ;

      if($this->admin_model->edit($userid,
           $username,$email, '
          votingofficial')){
        $data['success'] = '<strong>
            Success!</strong> You have
            edited User '.$userid.'.';
      }else{
        $data['error'] = '<strong>Error!</
            strong> User '.$userid.' was
            not edited.';
      }

    }else if($function=='delete'){

      $userid = $this->input->post('
          user_id');
      foreach($userid as $user_id){
        if(!$this->admin_model->delete(
            $user_id)){
          $data['error'] .= '<strong>Error
              !</strong> User '.$userid.'
               was not deleted.';
        }
      }

    }else if($function=='import'){

      if(!empty($_FILES['userfile']['name
          '])){
        $data['error'] = "";
        $config['upload_path']          =
            'application/uploads/';
        $config['allowed_types']        =
            'csv';
        $config['max_size']             =
            0;
        $config['file_name'] = '
            votingofficials';
        $config['overwrite'] = TRUE;
        $this->load->library('upload',
            $config);

        if (!$this->upload->do_upload('
            userfile')){
          $data['error'] .= "<strong>Error
              !</strong> Import Voting
              officials unsuccessful. \n
              ";
        }

        $filePath = $config['upload_path
            '].$config['file_name'].'.csv
            ';
        $parsed_file_content = array_map('
            str_getcsv', file($filePath))
            ;

        if(!delete_files($config['
            upload_path'])){
          $data['error'] .= "<strong>Error
              !</strong> Server-side
              deletion unsuccessful. \n";
        }

        if(!$this->insert_database(
            $parsed_file_content,$config
            ['upload_path'].'passwords.
            txt')){
          $data['error'] .= "<strong>Error
              !</strong> Import voting
              officials unsuccessful. \n
              ";
        }

      }
    }
  }

  $data['votingOfficials'] = $this->
      admin_model->get_vos();
  $this->load->view('templates/header');
  $this->load->view('admin/navbar',$data);
  $this->load->view('admin/view',$data);
  $this->load->view('admin/add_modal');
  $this->load->view('admin/edit_modal');
  $this->load->view('admin/delete_modal');
  $this->load->view('admin/
      delete_selected_modal');
  $this->load->view('admin/import_modal');
  $this->load->view('admin/footer');
}
```

```
   else
   {
     //If no session, redirect to login page
     redirect('login', 'refresh');
   }
 }

 function profile($function=""){
   if($this->session->userdata('logged_in')&&
        $this->session->userdata('logged_in
       ')['role'] == 'admin'){

     $session_data = $this->session->userdata
         ('logged_in');
     $user_id = $session_data['user_id'];

     if($this->input->post()){

       if($function == "edit"){

         $username = $this->input->post('
             username');
         $email = $this->input->post('email')
             ;
         $this->admin_model->edit($user_id,
             $username,$email, 'admin');

         $sess_array = $this->admin_model->
             getUserDetails($user_id, 'admin
             ');
         $this->session->set_userdata('
             logged_in', $sess_array);

       }else if($function == "delete"){

         $current_password = $this->input->
             post('current_password');
         $new_password = $this->input->post('
             new_password');
         $verify_password = $this->input->
             post('verify_password');

         if($this->admin_model->
             isPasswordCorrect($user_id,
             $current_password)){
           if($new_password ==
               $verify_password){
             if($this->admin_model->
                 changePassword($user_id,
                 $new_password)){
               $data['success'] = "Success!
                   Password was changed.";
             }else{
               $data['error'] = "Error! The
                   server was not able to
                   change your password.";
             }
           }else{
             $data['error'] = "New Password
                 and Verify New Password did
                  not match.";
           }
         }else{
           $data['error'] = "Current password
                incorrect.";
         }
       }
     }

     $session_data = $this->session->userdata
         ('logged_in');
     $data['username'] = $session_data['
         username']; // for future use
     $data['role'] = $session_data['role'];
     $data['email'] = $session_data['email'];

     $this->load->view('templates/header');
     $this->load->view('admin/navbar',$data);
     $this->load->view('admin/profile', $data
         );
     $this->load->view('admin/
         edit_profile_modal');
     $this->load->view('admin/
         change_password_modal');
     $this->load->view('admin/footer');
   }else{
     //If no session, redirect to login page
     redirect('login', 'refresh');
   }
 }

function insert_database($file_content,
    $upload_path)
{
  $this->load->helper('string');

  foreach($file_content as $row){
   $username = trim($row[0]);
   $email = trim($row[1]);
```

```
    $salt = random_string('alnum', 8);
    $password = random_string('alnum', 8);
    //$password = "12345";

    if(!$this->admin_model->insert($username,
        $email, $password, $salt)){
      return false;
    }else if(!$this->email($username,$password
        ,$email)){
     //write_file($upload_path, "\n Username
         :".$username.',Password:'.$password,
          "a+");
       return false;
    }
   }
   return true;
  }

  function email($username, $password, $email){

    $this->load->helper('email');
    $this->load->library('email');

    $this->email->from('votemind@gmail.com', '
        Votemind Admin');
    $this->email->to($email);

    $this->email->subject('Your Voting
        official account');
```

```
    $html = '<h2> Your voting official account
         details: </h2>';
    $html .= '<h3> Username:'.$username.'</h3
        >';
    $html .= '<h3> Password:'.$password.'</h3
        >';
    $this->email->message($html);

    if(!$this->email->send()){
      echo $this->email->print_debugger();
      return false;
    }
    return true;

  }

  function logout()
  {
    $this->session->unset_userdata('logged_in')
        ;
    session_destroy();
    redirect('login', 'refresh');
  }

}

?>
```

(c) Votingofficial.php

```
<?php if ( ! defined('BASEPATH')) exit('No
    direct script access allowed');

// session_start(); //we need to call PHP's
    session object to access it through CI

class Votingofficial extends CI_Controller{

 function __construct()
 {
   parent::__construct();
   $this->load->model('votingofficial_model
       ','',TRUE);
   $this->load->model('admin_model','',TRUE);
 }

 function index()
 {
   if($this->session->userdata('logged_in') &&
        $this->session->userdata('logged_in')
        ['role'] == 'votingofficial')
   {
     $session_data = $this->session->userdata
         ('logged_in');
     $data['username'] = $session_data['
         username']; // for future use

     $this->load->view('templates/header');
     $this->load->view('votingofficial/navbar
         ',$data);
     $this->load->view('templates/footer');
   }
   else
   {
     redirect('login', 'refresh');
   }
 }

 function profile($function=""){
    if($this->session->userdata('logged_in')&&
         $this->session->userdata('logged_in
         ')['role'] == 'votingofficial'){

      $session_data = $this->session->userdata
          ('logged_in');
      $user_id = $session_data['user_id'];

      if($this->input->post()){

        if($function == "edit"){

          $username = $this->input->post('
              username');
          $email = $this->input->post('email')
              ;
          $this->admin_model->edit($user_id,
              $username,$email, '
              votingofficial');

          $sess_array = $this->admin_model->
              getUserDetails($user_id, '
              votingofficial');
          $this->session->set_userdata('
              logged_in', $sess_array);
```

```
        }else if($function == "delete"){

          $current_password = $this->input->
              post('current_password');
          $new_password = $this->input->post('
              new_password');
          $verify_password = $this->input->
              post('verify_password');

          if($this->admin_model->
              isPasswordCorrect($user_id,
              $current_password)){
            if($new_password ==
                $verify_password){
              if($this->admin_model->
                  changePassword($user_id,
                  $new_password)){
                $data['success'] = "Success!
                    Password was changed.";
              }else{
                $data['error'] = "Error! The
                    server was not able to
                    change your password.";
              }
            }else{
              $data['error'] = "New Password
                  and Verify New Password did
                  not match.";
            }
          }else{
            $data['error'] = "Current password
                 incorrect.";
          }
        }
      }

      $session_data = $this->session->userdata
          ('logged_in');
      $data['username'] = $session_data['
          username']; // for future use
      $data['role'] = $session_data['role'];
      $data['email'] = $session_data['email'];

      $this->load->view('templates/header');
      $this->load->view('votingofficial/navbar
          ',$data);
      $this->load->view('votingofficial/
          profile/profile', $data);
      $this->load->view('votingofficial/
          profile/edit_profile_modal');
      $this->load->view('votingofficial/
          profile/change_password_modal');
      $this->load->view('templates/footer');
    }else{
      //If no session, redirect to login page
      redirect('login', 'refresh');
    }
  }

  function notifications($function=""){
    if($this->session->userdata('logged_in')&&
         $this->session->userdata('logged_in
         ')['role'] == 'votingofficial')
    {
      $session_data = $this->session->userdata
```

```
        ('logged_in');
    $user_id = $session_data['user_id'];
    $data['username'] = $session_data['
        username']; // for future use
    $data['role'] = $session_data['role'];
    $data['email'] = $session_data['email'];

    if($this->input->post()){
      if($function=="delete"){
        $notif_id = $this->input->post('
            notif_id');
        if(!$this->votingofficial_model->
            deleteNotification($notif_id)){
          $data['error'] = "<strong>Error!</
              strong> Delete notification
              failed.";
        }
      }else if($function=="resolve"){
        $notif_id = $this->input->post('
            notif_id');
        $remarks = $this->input->post('
            remarks');
        $this->load->helper('date');
        $datestring = "%Y-%m-%d";
        $time = time();
        $date = mdate($datestring, $time);
        if(!$this->votingofficial_model->
            resolveNotification($notif_id,
            $remarks, $user_id, $date)){
          $data['error'] = "<strong>Error!</
              strong> Resolve notification
              failed.";
        }
      }
    }

    $data['notifications'] = $this->
        votingofficial_model->
        getNotifications();
    $this->load->view('templates/header');
    $this->load->view('votingofficial/navbar
        ',$data);
    $this->load->view('votingofficial/
        notification/mainpage',$data);
    $this->load->view('votingofficial/
        notification/info_modal');
    $this->load->view('votingofficial/
        notification/resolve_modal');
    $this->load->view('votingofficial/
        notification/delete_modal');
    $this->load->view('votingofficial/
        notification/footer');

  }else{
    //If no session, redirect to login page
    redirect('login', 'refresh');
  }
}

function elections($function = ""){

  if($this->session->userdata('logged_in')
      && $this->session->userdata('
      logged_in')['role'] == '
      votingofficial')
  {
    $session_data = $this->session->userdata
        ('logged_in');
    $data['username'] = $session_data['
        username']; // for future use

    if($function=='postResults'){ //AJAX
        PART

      $election_id = $this->input->post('
          election_id');

      if(!empty($election_id) &&
          $election_id != 'None'){
        $this->session->set_userdata('
            election_id', $election_id);
        $candidates = $this->
            votingofficial_model->
            getCandidatesForSm($election_id
            );
        header('Content-type: text/plain');
        header('Content-type: application/
            json');
        echo json_encode($candidates);
      }
    }else{

      if($this->input->post()){

        if($function=="add"){

          $election_name = $this->input->
              post('name');
          $parent_id = $this->input->post('
              parent');

          if(!$this->votingofficial_model->
              addElection($election_name,
              $parent_id)){
            $data['error'] = '<strong>Error
                !</strong> Adding election
                '.$election_name.'
                unsuccessful.';
          }

        }else if($function=="change"){

          $this->load->helper('date');

          $election_id = $this->input->post
              ('election_id');
          $status = $this->input->post('
              status');

          if($status == 1){
            $num_of_votes = $this->input->
                post('num_of_votes');
            if(!empty($num_of_votes)){
              $candidates = $this->
                  votingofficial_model->
                  getCandidateIdsOnly(
                  $election_id);
              $date = "";
              if(!$this->
                  votingofficial_model->
                  changeStatus($election_id
                  ,$status,$date)){
                $data['error'] = '<strong>
                    Error!</strong> Update
                    election unsuccessful
                    .';
              }else{
                if(!$this->
                    votingofficial_model->
                    addVotesToCandidates(
                    $candidates,
                    $num_of_votes)){
                  $data['error'] = '<strong>
                      Error!</strong>
                      Update vote counts
                      unsuccessful.';
                }
              }
            }else{
              $candidates = $this->
                  votingofficial_model->
                  getCandidateIdsOnly(
                  $election_id);
              $i=0;
              foreach($candidates as
                  $candidate){
                $num_of_votes[$i]=0;
                $i++;
              }
              $date = "";
              if(!$this->
                  votingofficial_model->
                  changeStatus($election_id
                  ,$status,$date)){
                $data['error'] = '<strong>
                    Error!</strong> Update
                    election unsuccessful
                    .';
              }else{
                if(!$this->
                    votingofficial_model->
                    addVotesToCandidates(
                    $candidates,
                    $num_of_votes)){
                  $data['error'] = '<strong>
                      Error!</strong>
                      Update vote counts
                      unsuccessful.';
                }
              }
            }
          }else{
            $datestring = "%Y-%m-%d";
            $time = time();
            $date = mdate($datestring, $time
                );

            if(!$this->votingofficial_model
                ->changeStatus($election_id
                , $status, $date)){
              $data['error'] = '<strong>
                  Error!</strong> Edit
                  election unsuccessful.';
            }
          }


        }else if($function=="edit"){
```

```
    $election_id = $this->input->post
          ('election_id');
    $election_name = $this->input->
          post('name');
    $parent_id = $this->input->post('
          parent');

    if(!$this->votingofficial_model->
          editElection($election_id,
          $election_name,$parent_id)){
      $data['error'] = '<strong>Error
            !</strong> Edit election
            unsuccessful.';
    }

  }else if($function=="delete"){

    $election_id = $this->input->post
          ('election_id');

    if(!$this->votingofficial_model->
          deleteElection($election_id))
          {
      $data['error'] = '<strong>Error
            !</strong> Delete election
            unsuccessful.';
    }
    $this->session->set_userdata('
          election_id', NULL);

  }
}

$data['elections'] = $this->
      votingofficial_model->
      getElections();
foreach($data['elections'] as
      $election){
  if($election['status'] == -2 ||
        $election['status'] == -1){
    if(!$this->votingofficial_model->
          updateStatus($election["
          election_id"])){
      $data['error'] = '<strong>Error
            !</strong> Checking and
            updating elections
            unsuccessful.';
    }
  }
}

$data['elections'] = $this->
      votingofficial_model->
      getElections();
for($i=0; $i<count($data['elections'])
      ; $i++){
  $election = $data['elections'][$i];
  if($this->votingofficial_model->
        partOfVoterType($election["
        election_id"]))
    $data['elections'][$i]['
          partOfVoterType'] = true;
  else
    $data['elections'][$i]['
          partOfVoterType'] = false;
}

for($i=0; $i<count($data['elections'])
      ; $i++){
  if(!empty($data['elections'][$i]['
        parent_id']))
    $data['elections'][$i]['
          parent_name'] = $this->
          votingofficial_model->
          getElectionName($data['
          elections'][$i]['parent_id'])
          ;
  else{
    $data['elections'][$i]['parent_id
          '] = -1;
    $data['elections'][$i]['
          parent_name'] = 'None';
  }
}

$this->load->view('templates/header');
$this->load->view('votingofficial/
      navbar',$data);
$this->load->view('votingofficial/
      elections/elections',$data);
$this->load->view('votingofficial/
      elections/add_modal',$data);
$this->load->view('votingofficial/
      elections/change_modal',$data);
$this->load->view('votingofficial/
      elections/edit_modal',$data);
$this->load->view('votingofficial/
      elections/delete_modal',$data);
```

```
      $this->load->view('votingofficial/
            elections/result_modal',$data);
      $this->load->view('votingofficial/
            elections/footer');
    }
  }
  else
  {
    redirect('login', 'refresh');
  }
}

function positions($function = ""){
  if($this->session->userdata('logged_in')
        && $this->session->userdata('
        logged_in')['role'] == '
        votingofficial')
  {
    $session_data = $this->session->userdata
          ('logged_in');
    $data['username'] = $session_data['
          username']; // for future use

    if($function=='getPositions'){ //AJAX
          PART

      $election_id = $this->input->post('
            election_id');

      if(!empty($election_id) &&
            $election_id != 'None'){
        $this->session->set_userdata('
              election_id', $election_id);
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
              votingofficial_model->
              getElectionName($election_id);
        $data['election_status'] = $this->
              votingofficial_model->
              getElectionStatus($election_id)
              ;
        $data['positions'] = $this->
              votingofficial_model->
              getPositions($election_id);
        $data['elections'] = $this->
              votingofficial_model->
              getElectionIdsAndNames();
        $this->load->view('votingofficial/
              positions/table',$data);
        $this->load->view('votingofficial/
              positions/add_modal',$data);
        $this->load->view('votingofficial/
              positions/edit_modal',$data);
        $this->load->view('votingofficial/
              positions/delete_modal',$data);
        $this->load->view('votingofficial/
              positions/modal_scripts',$data)
              ;
      }

    }else if($function=="reorder" && $this->
          input->post('position_ids')){ //
          AJAX

      $position_ids = $this->input->post('
            position_ids');
      if(!$this->votingofficial_model->
            arrangePositions($position_ids)){
        $data['error'] = '<strong>Error!</
              strong> Reorder positions
              unsuccessful.';
        echo 'false';
      }else{
        echo 'true';
      }

    }else{

      $election_id = $this->session->
            userdata('election_id');

      if($this->input->post()){

        if($function=='add'){

          $position_name = $this->input->
                post('position_name');
          $max = $this->input->post('max');

          if(!$this->votingofficial_model->
                addPosition($position_name,
                $election_id, $max)){
            $data['error'] = '<strong>Error
                  !</strong> Add position
                  unsuccessful.';
          }else{
            $data['success'] = '<strong>
                  Error!</strong> Add
                  position unsuccessful.';
          }

        }else if($function=='edit'){

          $position_id = $this->input->post
              ('position_id');
          $position_name = $this->input->
              post('position_name');
          $max = $this->input->post('max');

          if(!$this->votingofficial_model->
              editPosition($position_id,
              $position_name, $max)){
            $data['error'] = '<strong>Error
                !</strong> Edit position
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Edit
                position unsuccessful.';
          }

        }else if($function=='delete'){

          $position_id = $this->input->post
              ('position_id');
          $position_ids = $this->input->post
              ('position_ids');

          if(!$this->votingofficial_model->
              deletePosition($position_id,
              $position_ids)){
            $data['error'] = '<strong>Error
                !</strong> Delete position
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Delete
                position unsuccessful.';
          }

        }
      }

      $data['elections'] = $this->
          votingofficial_model->
          getElectionIdsAndNames();

      if(!empty($election_id) &&
          $election_id != 'None'){
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
      }

      $this->load->view('templates/header');
      $this->load->view('votingofficial/
          navbar',$data);
      $this->load->view('votingofficial/
          positions/positions',$data);
      $this->load->view('votingofficial/
          positions/footer');

    }
  }
  else
  {
    redirect('login', 'refresh');
  }
}

function parties($function = ""){
  if($this->session->userdata('logged_in')
      && $this->session->userdata('
      logged_in')['role'] == '
      votingofficial')
  {
    $session_data = $this->session->userdata
        ('logged_in');
    $data['username'] = $session_data['
        username']; // for future use

    if($function=='getParties'){ //AJAX PART

      $election_id = $this->input->post('
          election_id');

      if(!empty($election_id) &&
          $election_id != 'None'){
        $this->session->set_userdata('
            election_id', $election_id);
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
        $data['election_status'] = $this->
            votingofficial_model->
            getElectionStatus($election_id)
            ;
        $data['parties'] = $this->
            votingofficial_model->
            getParties($election_id);
        $data['elections'] = $this->
            votingofficial_model->
            getElectionIdsAndNames();
        $this->load->view('votingofficial/
            parties/table',$data);
        $this->load->view('votingofficial/
            parties/add_modal');
        $this->load->view('votingofficial/
            parties/edit_modal');
        $this->load->view('votingofficial/
            parties/delete_modal');
        $this->load->view('votingofficial/
            parties/modal_scripts');
      }

    }else{

      $election_id = $this->session->
          userdata('election_id');

      if($this->input->post()){
        if($function=='add'){

          $party_name = $this->input->post('
              party_name');

          if(!$this->votingofficial_model->
              addParty($party_name,
              $election_id)){
            $data['error'] = '<strong>Error
                !</strong> Add party
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Error!</strong> Add party
                unsuccessful.';
          }

        }else if($function=='edit'){

          $newelection_id = $this->input->
              post('election_id');
          $party_id = $this->input->post('
              party_id');
          $party_name = $this->input->post('
              party_name');

          if(!$this->votingofficial_model->
              editParty($party_id,
              $party_name, $newelection_id)
              ){
            $data['error'] = '<strong>Error
                !</strong> Edit party
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Edit
                party unsuccessful.';
          }

        }else if($function=='delete'){

          $party_id = $this->input->post('
              party_id');

          if(!$this->votingofficial_model->
              deleteParty($party_id)){
            $data['error'] = '<strong>Error
                !</strong> Delete party
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Delete
                party unsuccessful.';
          }

        }
      }

      $data['elections'] = $this->
          votingofficial_model->
          getElectionIdsAndNames();

      if(!empty($election_id) &&
          $election_id != 'None'){
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
      }

      $this->load->view('templates/header');
      $this->load->view('votingofficial/
          navbar',$data);
      $this->load->view('votingofficial/
          parties/parties',$data);
      $this->load->view('votingofficial/
          parties/footer');
    }
  }
  else
  {
    redirect('login', 'refresh');
  }
}

function candidates($function = ""){
  if($this->session->userdata('logged_in')
      && $this->session->userdata('
      logged_in')['role'] == '
      votingofficial')
  {
    $session_data = $this->session->userdata
        ('logged_in');
    $data['username'] = $session_data['
        username']; // for future use
    if($function=='getCandidates'){ //AJAX
        PART

      $election_id = $this->input->post('
          election_id');

      if(!empty($election_id) &&
          $election_id != 'None'){
        $this->session->set_userdata('
            election_id', $election_id);
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
        $data['election_status'] = $this->
            votingofficial_model->
            getElectionStatus($election_id)
            ;
        $data['elections'] = $this->
            votingofficial_model->
            getElectionIdsAndNames();
        $data['positions'] = $this->
            votingofficial_model->
            getPositions($election_id);
        $data['parties'] = $this->
            votingofficial_model->
            getParties($election_id);
        $data['candidates'] = $this->
            votingofficial_model->
            getCandidates($election_id);

        $this->load->view('votingofficial/
            candidates/table',$data);
        $this->load->view('votingofficial/
            candidates/add_modal');
        $this->load->view('votingofficial/
            candidates/edit_modal');
        $this->load->view('votingofficial/
            candidates/delete_modal');
        $this->load->view('votingofficial/
            candidates/modal_scripts');
      }

    }else{

      $election_id = $this->session->
          userdata('election_id');

      if($this->input->post()){
        if($function=='add'){

          $first_name = $this->input->post('
              first_name');
          $last_name = $this->input->post('
              last_name');
          $position_id = $this->input->post
              ('position_id');
          $party_id = $this->input->post('
              party_id');

          if(!$this->votingofficial_model->
              addCandidate($first_name,
              $last_name, $election_id,
              $position_id, $party_id)){
            $data['error'] = '<strong>Error
                !</strong> Add Candidate
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Add
                Candidate successful.';
          }

        }else if($function=='edit'){

          $candidate_id = $this->input->post
              ('candidate_id');
          $first_name = $this->input->post('
```

```
              first_name');
          $last_name = $this->input->post('
              last_name');
          $position_id = $this->input->post
              ('position_id');
          $party_id = $this->input->post('
              party_id');

          if(!$this->votingofficial_model->
              editCandidate($candidate_id,
              $first_name, $last_name,
              $position_id, $party_id)){
            $data['error'] = '<strong>Error
                !</strong> Edit Candidate
                unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Edit
                Candidate unsuccessful.';
          }

        }else if($function=='delete'){

          $candidate_id = $this->input->post
              ('candidate_id');

          if(!$this->votingofficial_model->
              deleteCandidate($candidate_id
              )){
            $data['error'] = '<strong>Error
                !</strong> Delete Candidate
                 unsuccessful.';
          }else{
            $data['success'] = '<strong>
                Success!</strong> Delete
                Candidate unsuccessful.';
          }

        }
      }

      $data['elections'] = $this->
          votingofficial_model->
          getElectionIdsAndNames();

      if(!empty($election_id) &&
          $election_id != 'None'){
        $data['election_id'] = $election_id;
        $data['election_name'] = $this->
            votingofficial_model->
            getElectionName($election_id);
      }

      $this->load->view('templates/header');
      $this->load->view('votingofficial/
          navbar',$data);
      $this->load->view('votingofficial/
          candidates/candidates',$data);
      $this->load->view('votingofficial/
          candidates/footer');
    }
  }
  else
  {
    redirect('login', 'refresh');
  }
}

function votertypes($function = ""){
  if($this->session->userdata('logged_in')
      && $this->session->userdata('
      logged_in')['role'] == '
      votingofficial')
  {
    // if($function=='getChildren' && $this
        ->input->post('election_id')){ //
        called by ajax

    //   $election_id = $this->input->post('
        election_id');
    //   $children = $this->
        votingofficial_model->
        getElectionChildren($election_id);

    //   if($children){
    //     header('Content-type: text/plain
        ');
    //     header('Content-type: application
        /json');
    //     echo json_encode($children);
    //   }

    // }else{

      $session_data = $this->session->
          userdata('logged_in');
      $data['username'] = $session_data['
          username']; // for future use
      if($this->input->post()){
```

```
  if($function=='add'){

    $name = $this->input->post('name')
        ;
    $checked = $this->input->post('
        election');
    if(!empty($checked)){
      if(!$this->votingofficial_model
          ->addVotertype($name,
          $checked)){
        $data['error'] = '<strong>
            Error!</strong> Add Voter
             Type unsuccessful.';
      }else{
        $data['success'] = '<strong>
            Error!</strong> Add Voter
             Type unsuccessful.';
      }
    }

  }else if($function=='edit'){
    $voter_type_id = $this->input->
        post('voter_type_id');
    $name = $this->input->post('name')
        ;
    $checked = $this->input->post('
        election');

    if(!empty($checked)){
      if(!$this->votingofficial_model
          ->editVotertype(
          $voter_type_id, $name,
          $checked)){
        $data['error'] = '<strong>
            Error!</strong> Edit
            Voter Type unsuccessful
            .';
      }else{
        $data['success'] = '<strong>
            Success!</strong> Edit
            Voter Type unsuccessful
            .';
      }
    }

  }else if($function=='delete'){

    $voter_type_id = $this->input->
        post('voter_type_id');

    if(!$this->votingofficial_model->
        deleteVotertype(
        $voter_type_id)){
      $data['error'] = '<strong>Error
          !</strong> Delete Voter
          Type unsuccessful.';
    }else{
      $data['success'] = '<strong>
          Success!</strong> Delete
          Voter Type unsuccessful.';
    }

  }
}

$data['elections'] = $this->
    votingofficial_model->
    getElectionsForVoterTypes();
$data['votertypes'] = $this->
    votingofficial_model->
    getVoterTypes();

for($i=0; $i<count($data['votertypes
    ']); $i++){
  $data['votertypes'][$i]['
      election_ids'] =
    $this->votingofficial_model->
        getVoterTypeElections($data['
        votertypes'][$i]['
        voter_type_id']);
  $data['votertypes'][$i]['active'] =
      $this->votingofficial_model->
      getVoterTypeStatus($data['
      votertypes'][$i]['voter_type_id
      ']);
}

$this->load->view('templates/header');
$this->load->view('votingofficial/
    navbar',$data);
$this->load->view('votingofficial/
    votertypes/votertypes',$data);
$this->load->view('votingofficial/
    votertypes/add_modal',$data);
$this->load->view('votingofficial/
    votertypes/edit_modal',$data);
$this->load->view('votingofficial/
    votertypes/delete_modal',$data);
$this->load->view('votingofficial/
```

```
          votertypes/footer');
      }
    // }

  else
  {
    redirect('login', 'refresh');
  }
}

function voters($function = "")
{
  if($this->session->userdata('logged_in')
      && $this->session->userdata('
      logged_in')['role'] == '
      votingofficial')
  {
    $session_data = $this->session->userdata
        ('logged_in');
    $data['username'] = $session_data['
        username']; // for future use
    if($this->input->post() || !empty(
        $_FILES['userfile']['name'])){

      if($function=='add'){

        $this->load->helper('string');

        $username = $this->input->post('
            username');
        $email = $this->input->post('email')
            ;
        $voter_type_id = $this->input->post
            ('voter_type_id');
        $salt = random_string('alnum', 8);
        $password = random_string('alnum',
            8);
        //$password = "qwerty";

        if(!$this->votingofficial_model->
            addVoter($username, $email,
            $password, $salt,
            $voter_type_id)){

          $data['error'] = '<strong>Error!</
              strong> Add Voter
              unsuccessful.';

        }else{

          //write_file('application/uploads/
              passwords.txt', "\n Username
              :".$username.',Password:'.
              $password, "a+");

          if(!$this->email($username,
              $password,$email))
            $data['error'] = '<strong>Error
                !</strong> Email Voter
                unsuccessful.';
          else
            $data['success'] = '<strong>
                Success!</strong> Add Voter
                 successful.';

        }

      }else if($function=='import'){

        if(!empty($_FILES['userfile']['name
            '])){

          $voter_type_id = $this->input->
              post('voter_type_id');
          $data['error'] = "";

          $config['upload_path']          =
              'application/uploads/';
          $config['allowed_types']        =
              'csv';
          $config['max_size']             =
              0;
          $config['file_name'] = 'voters';
          $config['overwrite'] = TRUE;

          $this->load->library('upload',
              $config);

          if (!$this->upload->do_upload('
              userfile')){
            $data['error'] .= "<strong>Error
                !</strong> Import Voters
                unsuccessful. \n";
          }

          $filePath = $config['upload_path
              '].$config['file_name'].'.csv
              ';
          $parsed_file_content = array_map('
          str_getcsv', file($filePath))
          ;

      if(!delete_files($config['
          upload_path'])){
        $data['error'] .= "<strong>Error
            !</strong> Server-side
            deletion unsuccessful. \n";
      }

      if(!$this->insert_database(
          $parsed_file_content,$config
          ['upload_path'].'passwords.
          txt', $voter_type_id)){
        $data['error'] .= "<strong>Error
            !</strong> Import Voters
            unsuccessful. \n";
      }

    }

  }else if($function=='edit'){

    $user_id = $this->input->post('
        user_id');
    $username = $this->input->post('
        username');
    $email = $this->input->post('email')
        ;
    $voter_type_id = $this->input->post
        ('voter_type_id');

    if(!$this->votingofficial_model->
        editVoter($user_id, $username,
        $email, $voter_type_id)){
      $data['error'] = '<strong>Error!</
          strong> Edit Voter
          unsuccessful.';
    }else{
      $data['success'] = '<strong>
          Success!</strong> Edit Voter
          unsuccessful.';
    }

  }else if($function=='delete'){

    $userid = $this->input->post('
        user_id');
    foreach($userid as $user_id){
      if(!$this->votingofficial_model->
          deleteVoter($user_id)){
        $data['error'] = '<strong>Error
            !</strong> Delete Voter
            unsuccessful.';
      }
    }

  }
}

$data['votertypes'] = $this->
    votingofficial_model->getVoterTypes
    ();
$num_of_votertypes = count($data['
    votertypes']);
for($i=0,$j=0;$j<$num_of_votertypes;$j
    ++){
  $row = $data['votertypes'][$i];
  if($this->votingofficial_model->
      getVoterTypeStatus($row['
      voter_type_id']) == true){
    array_splice($data['votertypes'],$i
        ,1);
  }else{
    $i++;
  }
}
$data['voters'] = $this->
    votingofficial_model->getVoters();
$i=0;
foreach ($data['voters'] as $voter) {
  if($this->votingofficial_model->
      getVoterTypeStatus($voter['
      voter_type_id']) == true){
    $data['voters'][$i]['active'] = true
        ;
  }else{
    $data['voters'][$i]['active'] =
        false;
  }
  $i++;
}

$this->load->view('templates/header');
$this->load->view('votingofficial/navbar
```

```
          ',$data);
      $this->load->view('votingofficial/voters
          /voters', $data);
      $this->load->view('votingofficial/voters
          /add_modal',$data);
      $this->load->view('votingofficial/voters
          /import_modal',$data);
      $this->load->view('votingofficial/voters
          /edit_modal',$data);
      $this->load->view('votingofficial/voters
          /delete_modal',$data);
      $this->load->view('votingofficial/voters
          /delete_selected_modal',$data);
      $this->load->view('votingofficial/voters
          /footer');

    }
    else
    {
      redirect('login', 'refresh');
    }
  }

  function insert_database($file_content,
      $upload_path, $voter_type_id)
  {

    $this->load->helper('string');

    foreach($file_content as $row){

      $username = trim($row[0]);
      $email = trim($row[1]);
      $salt = random_string('alnum', 8);
      $password = random_string('alnum', 8);
      //$password = "qwerty";

      //write_file($upload_path, "\n Username
          :".$username.',Password:'.$password
          , "a+");

      if(!$this->email($username,$password,
          $email)){
        return false;
      }

      if(!$this->votingofficial_model->
          addVoter($username, $email,
          $password, $salt, $voter_type_id)){
        return false;
      }
    }
    return true;
  }

 function email($username, $password, $email){

    $this->load->helper('email');
    $this->load->library('email');

    $this->email->from('votemind@gmail.com', '
        Votemind Voting Official');
    $this->email->to($email);

    $this->email->subject('Your Voter account
        ');

    $html = '<h2> Your voter account details:
        </h2>';
    $html .= '<h3> Username:'.$username.'</h3
        >';
    $html .= '<h3> Password:'.$password.'</h3
        >';
    $this->email->message($html);

    if(!$this->email->send()){
      echo $this->email->print_debugger();
      return false;
    }
    return true;

  }

 function logout()
 {
   $this->session->unset_userdata('logged_in')
       ;
   session_destroy();
   redirect('login', 'refresh');
 }

}

?>
```

(d) Voter.php

```php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Voter extends CI_Controller {

  function __construct()
  {
    parent::__construct();
    $this->load->model('voter_model','',TRUE);
  }

  function vote($function=""){
    if($this->session->userdata('logged_in') && $this->session->userdata('logged_in')['role'] == 'voter')
    {
      $session_data = $this->session->userdata('logged_in');
      $data['username'] = $session_data['username']; // for future use

      $user_id = $session_data['user_id'];
      $voter = $this->voter_model->getVoterTypeId($user_id);
      $voter_id = $voter['voter_id'];
      $voter_type_id = $voter['voter_type_id'];
      $voted = $voter['voted'];
      $data['elections'] = $this->voter_model->getVoterTypeElections($voter_type_id);
      $data['voter_id'] = $voter_id;
      $data['alias'] = "V".($user_id+$voter_id);

      if($voted && $this->voter_model->getVoterTypeStatus($voter_type_id) == true){

        $this->load->view('templates/header');
        $this->load->view('voter/navbar');
        $this->load->view('voter/voted', $data);
        $this->load->view('voter/notify_modal');
        $this->load->view('voter/voted_footer');

      }else{

        if($this->voter_model->getVoterTypeStatus($voter_type_id) == true){
          $i=0;
          //print_r($data['elections']);
          foreach($data['elections'] as $election){
            if($election['status'] == 0){
              $data['elections'][$i]['candidates'] = $this->voter_model->getCandidates($election['election_id']);
              $data['elections'][$i]['positions'] = $this->voter_model->getPositions($election['election_id']);
              $i++;
            }
          }

          if($function=="getReceipt")
          {
            $codeString = $this->input->post("codeString");
            if(!$this->voter_model->setVoted($data['alias'],$codeString,$voter_id)){
              $data['error'] = "<strong>Error!</strong> Please notify a voting official that your status was not changed to voted.";
            }
            $this->load->view('voter/receipt', $data);
            $this->load->view('voter/notify_modal');
          }
          else
          {
            $this->load->view('templates/header');
            $this->load->view('voter/navbar');
            $this->load->view('voter/vote', $data);
            $this->load->view('voter/vote_modal', $data);
            $this->load->view('voter/footer');
          }

        }else{
          $this->load->view('templates/header');
          $this->load->view('voter/navbar');
          $this->load->view('voter/inactive');
          $this->load->view('voter/notify_modal');
          $this->load->view('voter/voted_footer');
        }
      }
    }
    else
    {
      redirect('login', 'refresh');
    }
  }

  function notify(){
    if($this->session->userdata('logged_in') && $this->session->userdata('logged_in')['role'] == 'voter')
    {
      $this->load->helper('date');

      $session_data = $this->session->userdata('logged_in');
      $user_id = $session_data['user_id']; // for future use
      $title = $this->input->post('title');
      $body = $this->input->post('body');
      $datestring = "%Y-%m-%d";
      $time = time();
      $date = mdate($datestring, $time);

      if($this->voter_model->sendNotif($user_id, $title, $body, $date)){
        echo "sent";
      }
    }
    else
    {
      redirect('login', 'refresh');
    }
  }

  function logout()
  {
    $this->session->unset_userdata('logged_in');
    session_destroy();
    redirect('login', 'refresh');
  }

}

?>
```

2. Models

(a) Login_model.php

```php
<?php
Class Login_model extends CI_Model
{
  function login($username, $password)
  {
    $this -> db -> select('salt');
    $this -> db -> from('user');
    $this -> db -> where('username', $username);
    $this -> db -> limit(1);
    $query = $this -> db -> get();
    $result = $query->result();

    if($result){
      foreach($result as $row){
        $salt = $row->salt;
      }
      $this -> db -> select('user_id, username, role, email');
      $this -> db -> from('user');
      $this -> db -> where('username', $username);
      $this -> db -> where('password', SHA1($password.$salt));
      $this -> db -> limit(1);
      $query = $this -> db -> get();

      if($query -> num_rows() == 1)
        return $query->result();

    }else{
      return false;
    }
  }

  function getFinishedElections(){

    $this->db->select('election_id, name');
    $this->db->from('election');
    $this->db->where('status',1);
    $query = $this->db->get();

    if($query){
      return $query->result_array();
    }

    return false;
  }

  function getAliasAndCode($election_id){

    $this->db->select('alias, code, voter_id')
        ;
    $this->db->from('voter');
    $this->db->join('voter_type_election','
        voter.voter_type_id=
        voter_type_election.voter_type_id');
    $this->db->where('election_id',
        $election_id);
    $query = $this->db->get();

    if($query){
      return $query->result_array();
    }

    return false;
  }

  function getVoterAliasAndCode($voter_id){

    $this->db->select('alias,code');
    $this->db->from('voter');
    $this->db->where('voter_id',$voter_id);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->result_array()[0];
    }

    return false;
  }

  function getPositionName($position_id){

    $this->db->select('position_name');
    $this->db->from('position');
    $this->db->where('position_id',
        $position_id);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->row('position_name');
    }

    return false;
  }

  function sendNotif($title, $body, $date){
    $data = array(
      'title' => $title,
      'body' => $body,
      'date_sent' => $date
    );

    if($this->db->insert('notification', $data
        )){
      return TRUE;
    }else{
      return FALSE;
    }
  }

}
?>
```

(b) Admin_model.php

```
<?php
Class Admin_model extends CI_Model
{

 function insert($username, $email, $password,
      $salt)
 {
    $data = array(
      'username' => $username,
      'password' => SHA1($password.$salt),
      'role' => 'votingofficial',
      'email' => $email,
      'salt' => $salt
    );

    if($this->db->insert('user', $data)){
      return TRUE;
    }else{
      echo "Error: ".$this->db->error()['
          message'];
      return FALSE;
    }
 }

  function edit($userid, $username, $email,
      $role)
  {
    $data = array('username' => $username, '
        email' => $email);
    $where = "user_id=".$userid." AND role =
        '".$role."'";
    if($this->db->update('user', $data, $where
        )){
      return TRUE;
    }else{
      echo "Error: ".$this->db->error()['
          message'];
      return FALSE;
    }
  }

  function delete($userid)
  {
    $where = array('user_id' => $userid, 'role
        ' => 'votingofficial');
    if($this->db->delete('user', $where)){
      return TRUE;
    }else{
      echo "Error: ".$this->db->error()['
          message'];
      return FALSE;
    }
  }

 function get_vos()
 {
  $this->db->select('user_id, username, email
      ');
  $this->db->from('user');
  $this->db->where('role','votingofficial');
  $this->db->order_by('username');
  $query = $this->db->get();

  if($query){
    return $query->result_array();
  }
  echo "Error: ".$this->db->error()['message
      '];
  return false;
 }

  function getUserDetails($user_id, $role){
    $this->db->select('user_id, username,
        email, role');
    $this->db->from('user');
    $this->db->where('user_id',$user_id);
    $this->db->where('role',$role);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->result_array()[0];
    }
    echo "Error: ".$this->db->error()['message
        '];
    return false;
  }

  function isPasswordCorrect($user_id,
      $password){
    $this -> db -> select('salt');
    $this -> db -> from('user');
    $this -> db -> where('user_id', $user_id);
    $this -> db -> limit(1);
    $query = $this -> db -> get();
    $salt = $query->row('salt');

    $this->db->select('*');
    $this->db->from('user');
    $this->db->where('password',SHA1($password
        .$salt));
    if($this->db->count_all_results() == 1){
      return true;
    }else{
      return false;
    }
  }

  function changePassword($user_id, $password)
      {
    $this -> db -> select('salt');
    $this -> db -> from('user');
    $this -> db -> where('user_id', $user_id);
    $this -> db -> limit(1);
    $query = $this -> db -> get();
    $salt = $query->row('salt');

    $this->db->set('password', SHA1($password.
        $salt));
    $this->db->where('user_id',$user_id);
    if($this->db->update('user')){
      return true;
    }else{
      echo "Error: ".$this->db->error()['
          message'];
      return false;
    }
  }

}
?>
```

(c) Votingofficial_model.php

```
<?php
Class Votingofficial_model extends CI_Model
{

  function insert($username, $email, $password
      , $salt)
  {
    $data = array(
      'username' => $username,
      'password' => SHA1($password.$salt),
      'role' => 'voter',
      'email' => $email,
      'salt' => $salt
    );

    if($this->db->insert('user', $data)){
      return TRUE;
    }else{
      return FALSE;
    }
 }

  /* ELECTIONS */
  function getElections()
  {
    $query = $this->db->query('SELECT * FROM
        election ORDER BY name');
    return $query->result_array();
  }

  function updateStatus($election_id)
  {
    $election_id = intval($election_id);
    $query = $this->db->query("SELECT COUNT(*)
         FROM position WHERE election_id=".
        $this->db->escape($election_id));

    if($query->row('COUNT(*)') > 0){
      $query = $this->db->query("SELECT COUNT
          (*) FROM candidate WHERE
          election_id=".$this->db->escape(
          $election_id));
      if($query->row('COUNT(*)') > 0){
        $query = $this->db->query("SELECT
            COUNT(*) FROM voter as v JOIN
            voter_type_election as vte ON v.
            voter_type_id = vte.voter_type_id
             WHERE election_id=".$this->db->
            escape($election_id));
        if($query->row('COUNT(*)') > 1){
          if(!$this->db->query("UPDATE
              election SET status = -1 WHERE
              election_id =".$this->db->
              escape($election_id))){
            return false;
          }else{
            return true;
          }
        }
      }
    }


    if(!$this->db->query("UPDATE election SET
        status = -2 WHERE election_id =".
        $this->db->escape($election_id))){
      return false;
    }else{
      return true;
    }

  }

  function partOfVoterType($election_id)
  {
    $this->db->select('*');
    $this->db->from('voter_type_election');
    $this->db->where('election_id',
        $election_id);
    if($this->db->count_all_results() >0)
      return true;
    return false;
  }

  function getElectionIdsAndNames()
  {
    $query = $this->db->query('SELECT
        election_id,name FROM election ORDER
        BY name');
    return $query->result_array();
  }

  function changeStatus($election_id, $status,
      $date){
    if($status == 0){
      $data = array(
        'status' => $status,
        'start_date' => $date,
        'end_date' => NULL
      );
    } else if($status == 1){

      $data = array(
        'status' => $status,
        'end_date' => $date
      );
    } else{
      $data = array(
        'status' => $status,
        'start_date' => NULL,
        'end_date' => NULL
      );
    }

    $this->db->where('election_id',
        $election_id);
    if(!$this->db->update('election', $data))
      return FALSE;

    return TRUE;
  }

  function getElectionName($election_id){
    if(!empty($election_id)){
      $query = $this->db->query('SELECT name
          FROM election WHERE election_id='.
          $this->db->escape($election_id));
      $row = $query->row();
      return $row->name;
    }else{
      return 'None';
    }
  }

  function getElectionStatus($election_id){
    $this->db->select('status');
    $this->db->from('election');
    $this->db->where('election_id',
        $election_id);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->row('status');
    }

    return false;
  }

  function addElection($election_name,
    $parent_id)
{
  if($parent_id!=-1){
    $data = array(
      'name' => $election_name,
      'parent_id' => $parent_id,
      'status' => -2
    );
  }else{
    $data = array(
      'name' => $election_name,
      'status' => -2
    );
  }

  if($this->db->insert('election', $data)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function editElection($election_id,
    $election_name, $parent_id)
{
  if($parent_id!=-1)
    $data = array('name' => $election_name,
        'parent_id' => $parent_id);
  else
    $data = array('name' => $election_name,
        'parent_id' => null);
  $where = "election_id=".$election_id;
  if($this->db->update('election', $data,
      $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function deleteElection($election_id)
{
  $where = array('election_id' =>
      $election_id);
  if($this->db->delete('election', $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

/* POSITIONS */

function getPositions($election_id)
{

  $query = $this->db->query('SELECT * FROM
      position WHERE election_id='.$this->
      db->escape($election_id).' ORDER BY
      ordinality');
  if($query){
    return $query->result_array();
  }
  return false;
}

function arrangePositions($positions_ids)
{
  for($i=0;$i<count($positions_ids);$i++){
    $data[$i] = array('ordinality' => ($i+1)
        ,'position_id' => $positions_ids[$i
        ]);
  }
  if($this->db->update_batch('position',
      $data, 'position_id')){
    return TRUE;
  }else{
    return FALSE;
  }
}

function addPosition($position_name,
    $election_id, $max)
{
  //how to get ordinality

  $this->db->select_max('ordinality','
      ordinality');
  $this->db->from('position');
  $this->db->where('election_id',
      $election_id);
  $query = $this->db->get();
  $ordinality = $query->row('ordinality') +
      1;

  $data = array(
    'position_name' => $position_name,
    'election_id' => $election_id,
    'max' => $max,
    'ordinality' => $ordinality
  );

  if($this->db->insert('position', $data)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function editPosition($position_id,
    $position_name, $max)
{
  $data = array('position_name' =>
      $position_name, 'max' => $max);
  $where = "position_id=".$position_id;
  if($this->db->update('position', $data,
      $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function deletePosition($position_id,
    $positions_ids)
{

  $where = array('position_id' =>
      $position_id);

  if($this->db->delete('position', $where)){

    if(!empty($positions_ids)){

      $this->db->reset_query();

      for($i=0;$i<count($positions_ids);$i
          ++){
        $data[$i] = array('ordinality' => (
            $i+1),'position_id' =>
            $positions_ids[$i]);
      }
      if($this->db->update_batch('position',
          $data, 'position_id')){
        return TRUE;
      }else{
        return FALSE;
      }

    }

    return TRUE;

  }else{
    return FALSE;
  }
}

/* PARTIES */

function getParties($election_id)
{

  $query = $this->db->query('SELECT * FROM
      party WHERE election_id='.$this->db->
      escape($election_id).' ORDER BY
      party_name');
  if($query){
    return $query->result_array();
  }
  return false;
}

function addParty($party_name, $election_id)
{
  $data = array(
    'party_name' => $party_name,
    'election_id' => $election_id
  );

  if($this->db->insert('party', $data)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function editParty($party_id, $party_name,
    $election_id)
{
  $data = array('party_name' => $party_name,
      'election_id' => $election_id);
  $where = "party_id=".$party_id;
  if($this->db->update('party', $data,
      $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function deleteParty($party_id)
{
  $where = array('party_id' => $party_id);
  if($this->db->delete('party', $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

/* CANDIDATES */

function getCandidates($election_id)
{
  $query = $this->db->query('SELECT c.*, pos
       .position_name, par.party_name  FROM
       candidate AS c JOIN position AS pos
       on pos.position_id = c.position_id
       LEFT JOIN party as par on par.
       party_id = c.party_id WHERE c.
       election_id = '.$this->db->escape(
       $election_id).' ORDER BY pos.
       ordinality, c.last_name, c.first_name
       ');

  if($query){
    return $query->result_array();
  }
  return false;
}

function getCandidatesForSm($election_id)
{
  $this->db->select('candidate_id,
       position_id');
  $this->db->from('candidate');
  $this->db->where('election_id',
       $election_id);
  $query = $this->db->get();

  if($query){
    $i=0;
    foreach ($query->result() as $row)
    {
      $return_array[$i] = array("
           candidate_id" => $row->
           candidate_id, "position_id" =>
           $row->position_id);
      $i++;
    }
    return $return_array;
  }

  return false;
}

function getCandidateIdsOnly($election_id)
{
  $this->db->select('candidate_id');
  $this->db->from('candidate');
  $this->db->where('election_id',
       $election_id);
  $query = $this->db->get();

  if($query){
    $i=0;
    foreach ($query->result() as $row)
    {
      $return_array[$i] = $row->candidate_id
           ;
      $i++;
    }
    return $return_array;
  }

  return false;
}

function addCandidate($first_name,
     $last_name, $election_id, $position_id,
      $party_id)
{

  if($party_id == -1){
    $data = array(
      'first_name' => $first_name,
      'last_name' => $last_name,
      'election_id' => $election_id,
      'position_id' => $position_id
    );

  }else{
    $data = array(
      'first_name' => $first_name,
      'last_name' => $last_name,
      'election_id' => $election_id,
      'position_id' => $position_id,
      'party_id' => $party_id
    );
  }

  if($this->db->insert('candidate', $data)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function addVotesToCandidates($candidates,
     $num_of_votes)
{
  for($i=0;$i<count($candidates);$i++){
    $data[$i] = array('vote_count' =>
         $num_of_votes[$i],'candidate_id' =>
          $candidates[$i]);
  }
  if($this->db->update_batch('candidate',
       $data, 'candidate_id')){
    return TRUE;
  }else{
    return FALSE;
  }
}

function editCandidate($candidate_id,
     $first_name, $last_name, $position_id,
     $party_id)
{
  if($party_id == -1){
    $data = array(
      'first_name' => $first_name,
      'last_name' => $last_name,
      'position_id' => $position_id,
      'party_id' => null
    );
  }else{
    $data = array(
      'first_name' => $first_name,
      'last_name' => $last_name,
      'position_id' => $position_id,
      'party_id' => $party_id
    );
  }

  $where = "candidate_id=".$candidate_id;
  if($this->db->update('candidate', $data,
       $where)){
    return TRUE;
  }else{
    return FALSE;
  }
}

function deleteCandidate($candidate_id)
{
  $where = array('candidate_id' =>
       $candidate_id);
  if($this->db->delete('candidate', $where))
       {
    return TRUE;
  }else{
    return FALSE;
  }
}

/* VOTER TYPES */

function getVoterTypes()
{
  $query = $this->db->query('SELECT * FROM
       voter_type ORDER BY name');
  if($query){
    return $query->result_array();
  }
  return false;
}

function getVoterTypeElections(
     $voter_type_id)
{
  $query = $this->db->query('SELECT
       election_id FROM voter_type_election
      WHERE voter_type_id='.$this->db->
       escape($voter_type_id));
  if($query){
    $i=0;
    $return_array = array();
    foreach($query->result_array() as $row){
      $return_array[$i]=$row['election_id'];
      $i++;
    }
    return $return_array;
  }
  return false;
}
```

```
function getElectionsForVoterTypes()
{
  $query = $this->db->query('SELECT
       election_id,name,parent_id FROM
       election ORDER BY parent_id');
  return $query->result_array();
}

function getElectionChildren($election_id)
{
  $this->db->select('*');
  $this->db->from('election');
  $this->db->where('parent_id',$election_id)
       ;
  $query = $this->db->get();
  if($query)
    return $query->result_array();
  return false;
}

function getVoterTypeStatus($voter_type_id)
{
  if(!empty($voter_type_id)){

    $this->db->select('*');
    $this->db->from('voter_type_election');
    $this->db->join('election', '
         voter_type_election.election_id =
         election.election_id');
    $this->db->where('voter_type_id',
         $voter_type_id);
    $this->db->where('status',0);
    $active_elections = $this->db->
         count_all_results();

    $this->db->select('*');
    $this->db->from('voter_type_election');
    $this->db->join('election', '
         voter_type_election.election_id =
         election.election_id');
    $this->db->where('voter_type_id',
         $voter_type_id);
    $total_elections = $this->db->
         count_all_results();

    if($total_elections == 0){
      $this->db->where('voter_type_id',
           $voter_type_id);
      $this->db->delete('voter_type');
      return true;
    }

    if($active_elections == $total_elections
         ){
      return true;
    }else
      return false;

  }else{
    return false;
  }
}

function addVotertype($name, $elections)
{
  $data = array(
    'name' => $name
  );

  if($this->db->insert('voter_type', $data))
       {

    $voter_type_id =  $this->db->insert_id()
         ;

    foreach($elections as $election_id){
      $data = array(
        'voter_type_id' => $voter_type_id,
        'election_id' => $election_id
      );
      if(!$this->db->insert('
           voter_type_election', $data)){
        return FALSE;
      }
    }

    return TRUE;
  }else{
    return FALSE;
  }
}

function editVotertype($voter_type_id, $name
     , $elections)
{
  $data = array(
    'name' => $name
  );
```

```
  $where = ”voter_type_id=”.$voter_type_id;

  if($this->db->update('voter_type', $data,
       $where)){

    if(!$this->db->delete('
         voter_type_election', $where)){
      return FALSE;
    }

    foreach($elections as $election_id){
      $data = array(
        'voter_type_id' => $voter_type_id,
        'election_id' => $election_id
      );
      if(!$this->db->insert('
           voter_type_election', $data)){
        return FALSE;
      }
    }

    return TRUE;

  }else{
    return FALSE;
  }
}

function deleteVotertype($voter_type_id)
{
  $where = ”voter_type_id=”.$voter_type_id;

  if($this->db->delete('voter_type', $where)
       ){
    return TRUE;
  }
  return FALSE;
}

/* VOTERS */

function getVoters()
{
  $query = $this->db->query('SELECT * FROM
       voter AS v JOIN user AS u on v.
       user_id = u.user_id
    LEFT JOIN voter_type AS vt ON v.
         voter_type_id = vt.voter_type_id
         ORDER BY u.username');

  if($query){
    return $query->result_array();
  }
  return false;
}

function addVoter($username, $email,
     $password, $salt, $voter_type_id)
{
  $data = array(
    'username' => $username,
    'password' => SHA1($password.$salt),
    'email' => $email,
    'role' => 'voter',
    'salt' => $salt
  );

  if($this->db->insert('user', $data)){

    $data = array(
      'user_id' => $this->db->insert_id(),
      'voter_type_id' => $voter_type_id
    );

    if($this->db->insert('voter', $data))
      return TRUE;

    return FALSE;

  }else{

    return FALSE;

  }
}

function editVoter($user_id, $username,
     $email, $voter_type_id)
{
  $data = array(
    'username' => $username,
    'email' => $email
  );

  $this->db->where('user_id', $user_id);

  if($this->db->update('user', $data)){

    $data = array(
        'voter_type_id' => $voter_type_id
      );

      $this->db->where('user_id', $user_id);

      if($this->db->update('voter', $data))
        return TRUE;

      return FALSE;

    }else{

      return FALSE;

    }
  }

  function deleteVoter($user_id)
  {
    $this->db->where('user_id', $user_id);

    if($this->db->delete('user'))
      return TRUE;

    return FALSE;
  }

  function getNotifications()
  {
    $this->db->select('*');
    $this->db->from('notification');
    $query = $this->db->get();
    if($query){
      return $query->result_array();
    }else{
      return FALSE;
    }
  }

  function deleteNotification($notif_id)
  {
    $this->db->where('notif_id', $notif_id);
    if($this->db->delete('notification')){
      return TRUE;
    }else{
      return FALSE;
    }
  }

  function resolveNotification($notif_id,
      $remarks, $user_id, $date)
  {
    $data = array(
      'date_resolved' => $date,
      'remarks' => $remarks,
      'vo_id' => $user_id
    );

    $this->db->where('notif_id', $notif_id);

    if($this->db->update('notification', $data
        )){
      return TRUE;
    }

    return FALSE;
  }

}
?>
```

(d) Voter_model.php

```
<?php
Class Voter_model extends CI_Model
{

  function getVoterTypeId($user_id){
    $this->db->select('voter_id, voter_type_id
        , voted');
    $this->db->from('voter');
    $this->db->where('user_id',$user_id);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->row_array();
    }

    return false;
  }

  function getVoterTypeElections(
      $voter_type_id)
  {
    $this->db->select('*');
    $this->db->from('voter_type_election');
    $this->db->join('election', '
        voter_type_election.election_id =
        election.election_id');
    $this->db->where('voter_type_id',
        $voter_type_id);
    $this->db->order_by('parent_id');
    $query = $this->db->get();

    if($query){
      return $query->result_array();
    }

    return false;
  }

  function getElectionDetails($election_id){
    $this->db->select('*');
    $this->db->from('election');
    $this->db->where('election_id',
        $election_id);
    $this->db->limit(1);
    $query = $this->db->get();

    if($query){
      return $query->row_array();
    }

    return false;
  }

  function getPositions($election_id)
  {
    $this->db->select('*');
    $this->db->from('position');
    $this->db->where('election_id',
        $election_id);
    $this->db->order_by('ordinality');
    $query = $this->db->get();

    if($query){
      return $query->result_array();
    }

    return false;
  }

  function getCandidates($election_id)
  {
    $query = $this->db->query('SELECT c.*, pos
        .*, par.party_name FROM candidate AS
         c JOIN position AS pos on pos.
        position_id = c.position_id LEFT JOIN
         party as par on par.party_id = c.
        party_id WHERE c.election_id = '.
        $this->db->escape($election_id).'
       ORDER BY pos.ordinality, c.last_name,
         c.first_name');

    if($query){
      return $query->result_array();
    }
    return false;
  }

  function setVoted($alias, $codeString,
      $voter_id){
    $data = array('alias' => $alias,'code'=>
        $codeString,'voted' => true);
    $where = "voter_id=".$voter_id;
    if($this->db->update('voter', $data,
        $where)){
      return TRUE;
    }else{
      return FALSE;
    }
  }

  function getVoterTypeStatus($voter_type_id)
  {
    if(!empty($voter_type_id)){

      $this->db->select('*');
      $this->db->from('voter_type_election');
      $this->db->join('election', '
          voter_type_election.election_id =
          election.election_id');
      $this->db->where('voter_type_id',
          $voter_type_id);
      $this->db->where('status',0);
      $active_elections = $this->db->
          count_all_results();
        $this->db->select('*');
        $this->db->from('voter_type_election');
        $this->db->join('election', '
             voter_type_election.election_id =
             election.election_id');
        $this->db->where('voter_type_id',
             $voter_type_id);
        $total_elections = $this->db->
             count_all_results();

        if($total_elections == 0){
          $this->db->where('voter_type_id',
               $voter_type_id);
          $this->db->delete('voter_type');
          return false;
        }

        if($active_elections == $total_elections
             ){
          return true;
        }else
          return false;

      }else{
          return false;
        }
      }

      function sendNotif($user_id, $title, $body,
           $date){
        $data = array(
          'voter_id' => $user_id,
          'title' => $title,
          'body' => $body,
          'date_sent' => $date
        );

        if($this->db->insert('notification', $data
             )){
          return TRUE;
        }else{
          return FALSE;
        }
      }

    }
    ?>
```

3. Views

(a) login

i. bulletin

A. bulletin.php

```
<div class="content container-fluid text-
     center">

<h2>Bulletin Board</h2><hr>

<?php if(!empty($elections)){ ?>
  <div class='form-horizontal'>
    <div class='form-group'>
      <label class='control-label input-lg col
           -sm-4'> Select an election: </label
           >
      <div class="col-sm-6">
        <select name='election_name' id='
             select_election_id' class='form-
             control input-lg'>
          <?php if(!empty($election_id)){ ?>
              <option value=<?php echo
                   $election_id; ?> default> <?
                   php echo $election_name; ?>
                   </option>
          <?php } else { ?>
            <option id='option_placeholder'
                 default> None </option>
          <?php } foreach($elections as $row){
            if($row['election_id']!=$election_id
                 ) {?>
              <option value=<?php echo $row['
                   election_id'];?>><?php echo
                   $row['name'];?></option>
            <?php }
              } ?>
          </select>
          </div>
        </div>
      </div>

      <hr>

      <div id='bulletin_board'>

      </div>

    <?php }else{ ?>

      <div class='alert alert-danger'><strong>
           Error!</strong> Bulletin board is empty
           .</div>

    <?php } ?>

      <div id='noVotesMsg'> </div>

    </div>
```

B. footer.php

```
<!-- <div id="footer">Copyright &copy; 2016
     Cybernetica AS</div> -->
<footer class="container-fluid text-left
     navbar-default">
<p>&copy; University of the Philippines Manila
     </p>
</footer>
<script src="<?php echo asset_url();?>js/
     jquery-1.12.0.min.js"></script>
<script src="<?php echo asset_url();?>js/
     bootstrap.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/bootbox.min.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/ext/typedarray.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/ext/jsaes.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/ext/jsbn.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/ext/jsbn2.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/ext/socket.io.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/prng.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/sm.types.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/sm.types.util.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/sm.types.base.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/sm.types.shared3p.js
     "></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/serialization.js"></
     script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/sm.ctrl.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/miners.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/app.js"></script>
<script type="text/javascript" src="<?php echo
     asset_url();?>js/page.js"></script>
<script src="<?php echo asset_url();?>crypto-
     js/sha1.js"></script>
<script type="text/javascript">

var hashes = [];

$(document).ready(function () {
$('.content').hide();
$('.content').delay(200).slideDown(800);

var election_id = $("#select_election_id").val
     ();

if(election_id!=null && election_id!="None"){

$('body').css({'cursor' : 'wait'});
var clbck = function(err_msg, results) {
if (!results) {
$('body').css({'cursor' : 'default'});
$('#noVotesMsg').hide();
$('#noVotesMsg').addClass('alert alert-danger
    ');
$('#noVotesMsg').html('There were no votes in
    this election.');
$('#noVotesMsg').slideDown();
}else{

for(i=0;i<results['hashes'].length;i++){
hashes[i] = CryptoJS.SHA1(""+results['hashes
    '][i]);
}

//console.log(results['voterIds']);

jQuery.ajax({
type: "POST",
url: "<?php echo linkbase_url(); ?>" + "login/
    bulletin/getBulletin",
data: {election_id: election_id, voterIds:
    results['voterIds'],
positionIds: results['positionIds']},
success: function(res) {
if (res)
{
//pass hashes to bulletin board
$('#bulletin_board').html(res);
$('#bulletin_board').delay(200).slideDown(800)
    ;
}
}
});

$('body').css({'cursor' : 'default'});
}
};

app.getHashes(election_id,clbck);
}

$('#select_election_id').change(function() {
$('#bulletin_board').slideUp(800,function(){

$('#option_placeholder').remove();
var election_id = $("#select_election_id").val
    ();
$('body').css({'cursor' : 'wait'});

var clbck = function(err_msg, results) {
if (!results) {
$('body').css({'cursor' : 'default'});
$('#noVotesMsg').hide();
$('#noVotesMsg').addClass('alert alert-danger
    ');
$('#noVotesMsg').html('There were no votes in
    this election.');
$('#noVotesMsg').slideDown();
}else{

for(i=0;i<results['hashes'].length;i++){
hashes[i] = CryptoJS.SHA1(""+results['hashes
    '][i]);
}

jQuery.ajax({
type: "POST",
url: "<?php echo linkbase_url(); ?>" + "login/
    bulletin/getBulletin",
data: {election_id: election_id, voterIds:
    results['voterIds'],
positionIds: results['positionIds']},
success: function(res) {
if (res)
{
//pass hashes to bulletin board
$('#bulletin_board').html(res);
$('#bulletin_board').delay(200).slideDown(800)
    ;
}
}
});
$('body').css({'cursor' : 'default'});
}
};

app.getHashes(election_id,clbck);
});
});

app.init();
});

</script>
</body>
</html>
```

C. receipt_modal.php

```
<!-- Receipt Modal -->
<div id="receiptModal" class="modal fade" role
    ="dialog">
  <div class="modal-dialog">

    <!-- Receipt Modal content-->
    <div class="modal-content">

      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class='
             glyphicon glyphicon-list-alt'></
             span> Receipt</h4>
      </div>

       <div class="modal-body">
        <div class="modal-form">

          <div class='text-left'>
            <strong id='receipt_alias'></
                 strong>
          </div>

          <div class='text-left'>
            <strong id='receipt_code'></strong
                 >
          </div>

          <br>

          <div class='well well-sm text-center
               '>
            <strong><?php echo $election_name
                 ." Election";?></strong>
          </div>
          <div class='form-group text-left' id
               ='receipt_positions_hashes'>

          </div>

          <!-- <?php foreach($positions as
               $row){ ?>

            <div class='form-group text-left'>

              <div>
                <strong><?php echo $row['
                     position_name'];?></
                     strong>
                <input name='position_dl'
                     class='downloadable' type
                     ='hidden' value="<?php
                     echo $row['position_name
                     '];?>" >
              </div>

              <div class='downloadable' id="<?
                   php echo $row['position_id
                   '].'_confirm';?>">
                <!-- jQuery will insert values
                     here -->
              </div>

            </div>

          <?php } ?>    -->

        </div>
      </div>
    </div>
  </div>
</div>
```

D. table.php

```
<?php if(!empty($error)){ ?>
  <div class=’alert alert−error fade in’>
      <a href=”#” class=”close” data−dismiss
          =”alert” aria−label=”close”>&
          times;</a>
      <?php echo $error;?>
  </div>
<?php } ?>

<?php if(!empty($voters)){ ?>

<div class=’text−left’>
  <table id=’bulletin_table’ class=”table
      table−hover text−left” style=”width
      :100%”>
    <thead>
      <tr>
        <th>Alias</th>
        <th>Code</th>
        <th><span class=’glyphicon glyphicon
            −list−alt’></span> Receipt</th>
      </tr>
    </thead>
    <tbody>
      <?php $index = 0;
      foreach ($voters as $row) {?>
        <tr class=’table−row’>
          <td><?php echo $row[’alias’];?></
              td>
          <td><?php echo $row[’code’];?></td
              >
          <td>
            <button class=”btn btn−warning
                btn−md” data−alias=”<?php
                echo $row[’alias’];?>” data
                −code=”<?php echo $row[’
                code’];?>”
            data−index=<?php echo $index;?>

            data−positions=”<?php
            $isFirst = true;
            foreach($row[’positions’] as
                $position){
              if($isFirst == true){
                echo $position;
                $isFirst = false;
              }else{
                echo ”,”.$position;
              }
              $index++;
            }?>””

            data−toggle=”modal” data−target
                =”#receiptModal”>
            See Receipt</button>
          </td>
        </tr>
      <?php } ?>
    </tbody>
  </table>

<div class=’text−center’>
    <button class=’btn btn−primary’ id=’
        download_button’>
    <span class=’glyphicon glyphicon−
        download−alt’></span> Download
        Bulletin Board PDF</button>
  </div>
</div>

<hr>

<h4>Notice any discrepancy?</h4>

<div class=’text−center’>
  <button class=”btn btn−success btn−danger”
      data−toggle=”modal” data−target=”#
      notifyModal” id=’mail_button’>
    <span class=’glyphicon glyphicon−
        envelope’></span> Notify Voting
        Officials</button>
</div>

<hr>

<div id=’notify_done’>
</div>

<?php }else{ ?>
  <div class=’alert alert−danger’><strong>
      Error!</strong> There are no aliases
      and codes in this election.</div>
<?php } ?>
```

E. table_script.php

```
<script src=”<?php echo asset_url();?>js/
    jquery.dataTables.min.js”></script>
<script src=”<?php echo asset_url();?>js/
    dataTables.bootstrap.min.js”></script>
<script src=”<?php echo asset_url();?>jsPDF/
    pdfmake.min.js”></script>
<script src=”<?php echo asset_url();?>jsPDF/
    vfs_fonts.js”></script>
<script type=”text/javascript”>

$(document).ready(function(){
$(”#bulletin_table”).dataTable({
”aoColumnDefs”: [
{ ’bSortable’: false, ’aTargets’: [2] }
]
});

$(’#notify_button’).on(’click’, function(e){
var title = $(’#notify_title’).val();
var body = $(’#notify_body’).val();

jQuery.ajax({
type: ”POST”,
url: ”<?php echo linkbase_url(); ?>” + ”login/
    notify”,
data: {title:title, body:body},
success: function(res) {
if (res){
$(’#notify_done’).addClass(’alert alert−
    success text−center’);
$(’#notify_done’).html(”SUCCESS: Your
    notification was sent.”);
$(’#mail_button’).attr(’disabled’,true);
}else{
$(’#notify_done’).addClass(’alert alert−danger
    text−center’);
$(’#notify_done’).html(”ERROR: Your
    notification was not sent.”);
}
}
});

});

$(”#receiptModal”).on(’show.bs.modal’,
    function(e){
var alias = $(e.relatedTarget).data(’alias’);
var code = $(e.relatedTarget).data(’code’);
var index = $(e.relatedTarget).data(’index’);
var positions = $(e.relatedTarget).data(’
    positions’);
$(e.currentTarget).find(’#receipt_alias’).html
    (’Alias: ’+alias);
$(e.currentTarget).find(’#receipt_code’).html
    (’Code: ’+code);

var content = ””;
positions_array = positions.split(”,”);
console.log(index);
for(i=0, j= index; i<positions_array.length; i
    ++, j++){
if(i==0){
content += ”<br><div> <strong>”+
    positions_array[i]+”</strong> </div>”;
content += ”<div> Hash: ”+hashes[j]+”</div>”;
}else if(positions_array[i] == positions_array
    [i−1]){
content += ”<div> Hash: ”+hashes[j]+”</div>”;
}else{
content += ”<br><div> <strong>”+
    positions_array[i]+”</strong> </div>”;
content += ”<div> Hash: ”+hashes[j]+”</div>”;
}
}
$(e.currentTarget).find(’#
    receipt_positions_hashes’).html(content);
});

$(’#download_button’).on(’click’, function(e){
var docDefinition = {

footer: function(currentPage, pageCount) {
    return {text: currentPage.toString() + ’
    of ’ + pageCount, alignment:’center’ };},

content: [
{ text: ”<?php echo $election_name.” Bulletin
    Board”;?> ”, fontSize: 15, alignment:’
    center’, margin: 5 },

{
table: {
headerRows: 1,
widths: [ ’auto’, ’30%’ ,’60%’],
body: [
[ {text: ’Alias’,bold:true}, {text:’Code’,bold
    :true}, {text:’Hashes’,bold:true}],

<?php
$i=0; $index = 0;
foreach ($voters as $row) {
if($i!=0) echo ”,”;
?>
[
”<?php echo $row[’alias’];?>”,
{text:”<?php
$half = (int) ( (strlen($row[’code’]) / 2) );
    // cast to int incase str length is odd
$left = substr($row[’code’], 0, $half);
$right = substr($row[’code’], $half);
echo $left.’ ’.$right;
?>”},
”<?php
foreach($row[’positions’] as $position){
echo $position.’: ’;
?> \n”+hashes[<?php echo intval($index);?>]+”\
    n <?php
$index++;
}
?>”
]

<?php
$i++;
}
?>

]
}
}
]
};

//pdfMake.createPdf(docDefinition).open();
pdfMake.createPdf(docDefinition).download(’<?
    php echo str_replace(” ”,”_”,
    $election_name.”_bulletin_board”);?>.pdf
    ’);
});

});

</script>
```

ii. results

A. footer.php

```
    <!-- <div id=”footer”>Copyright &copy;
         2016 Cybernetica AS</div> -->
  <footer class=”container-fluid text-left
      navbar-default”>
    <p>&copy; University of the Philippines
        Manila</p>
  </footer>
  <script src=”<?php echo asset_url();?>js/
      jquery-1.12.0.min.js”></script>
  <script src=”<?php echo asset_url();?>js/
      bootstrap.js”></script>
  <script type=”text/javascript”>

    $(document).ready(function () {
      $(’.content’).hide();
      $(’.content’).delay(200).slideDown(800);

      var election_id = $(”#select_election_id
          ”).val();

      if(election_id != ”None”){
        jQuery.ajax({
          type: ”POST”,
          url: ”<?php echo linkbase_url(); ?>”
              + ”login/results/getResults”,
          data: {election_id: election_id},
          success: function(res) {
            if (res)
            {
              $(’#results_table’).html(res);
              $(’#results_table’).delay(200).
                  slideDown(800);
            }
          }
        });
      }

      $(’#select_election_id’).change(function
          () {
        $(’#results_table’).slideUp(800,
            function(){
          $(’#option_placeholder’).remove();
          var election_id = $(”#
              select_election_id”).val();

          jQuery.ajax({
            type: ”POST”,
            url: ”<?php echo linkbase_url();
                ?>” + ”login/results/
                getResults”,
            data: {election_id: election_id},
            success: function(res) {
              if (res)
              {
                $(’#results_table’).html(res);
                $(’#results_table’).delay(200)
                    .slideDown(800);
              }
            }
          });
        });
      });

    });

  </script>
</body>
</html>
```

B. results.php

```
<div class=”content container-fluid text-
    center”>

<h2>Election results</h2><hr>

<?php if(!empty($elections)){ ?>
  <div class=’form-horizontal’>
    <div class=’form-group’>
      <label class=’control-label input-lg col
          -sm-4’> Select an election: </label
          >
      <div class=”col-sm-6”>
        <select name=’election_name’ id=’
            select_election_id’ class=’form-
            control input-lg’>
        <?php if(!empty($election_id)){ ?>
            <option value=<?php echo
                $election_id; ?> default> <?
                php echo $election_name; ?>
                </option>
        <?php } else { ?>
          <option id=’option_placeholder’
              default> None </option>
        <?php } foreach($elections as $row){
          if($row[’election_id’]!=$election_id
              ) {?>
            <option value=<?php echo $row[’
                election_id’];?>><?php echo
                $row[’name’];?></option>
          <?php }
            } ?>
        </select>
        </div>
      </div>
    </div>

  <hr>

  <div id=’results_table’>

  </div>
<?php }else{ ?>

  <div class=’alert alert-danger’><strong>
      Error!</strong> There are no finished
      election at the moment.</div>

<?php } ?>
</div>
```

C. table.php

```
  <?php if(!empty($error)){ ?>
    <div class='alert alert-error fade in'>
        <a href="#" class="close" data-dismiss
             ="alert" aria-label="close">&
             times;</a>
        <?php echo $error;?>
    </div>
  <?php } ?>

  <?php if(!empty($candidates)){
    $i=0;
    for($i=0;$i<count($candidates);){
      $row = $candidates[$i];
      $position_id = $row['position_id'];?>

    <div class='well well-sm'><strong><?php
         echo $row['position_name'];?></strong
         ></div>
    <table class="table text-left">
      <thead >
        <tr>
          <th>No.</th>
          <th>Candidate</th>
          <th>Party</th>
          <th>Number of Votes</th>
        </tr>
      </thead>
      <tbody>
          <?php $k=0; $sortedarray = array();
          while($i<count($candidates) && $row
               ['position_id'] == $position_id
               ){

            if(empty($row['party_name'])) $row
                 ['party_name'] = "INDEPENDENT
                 ";

            $sortedarray[$k] = array("
                 last_name" => $row['last_name
                 '], "first_name" =>$row["
                 first_name"],
              "party_name" =>$row['party_name
                   '], "vote_count" =>$row['
                   vote_count']);
            $i++; $k++;
            if($i < count($candidates)){
              $row = $candidates[$i];
            }
          }

          uasort($sortedarray, function ($x,
               $y) {
            $a = $x['vote_count'];
            $b = $y['vote_count'];
            if ($a == $b) return 0;
            elseif ($a > $b) return -1;
            else return 1;
          });

          $j = 1;
          foreach($sortedarray as $srow){ ?>
            <tr class='table-row'>
            <td><?php echo $j;?></td>
            <td><?php echo $srow['last_name
                 ']." , ".$srow["first_name
                 "];?></td>
            <td><?php echo $srow['party_name
                 '];?></td>
            <td><?php echo $srow['vote_count
                 '];?></td>
            </tr>
            <?php $j++;
          }?>
      </tbody>
    </table>
    <?php } ?>
    <button class='btn btn-primary' id='
         download_button'>
      <span class='glyphicon glyphicon-
           download-alt'></span> Download
           Results PDF</button>
    <?php }else{ ?>
      <div class='alert alert-danger'><strong>
           Error!</strong> There are no
           candidates in this election.</div>
    <?php } ?>

<script src="<?php echo asset_url();?>jsPDF/
     pdfmake.min.js"></script>
<script src="<?php echo asset_url();?>jsPDF/
     vfs_fonts.js"></script>
<script type="text/javascript">

  $(document).ready(function () {

    $('#download_button').on('click', function
         (e){
```

```
    var docDefinition = {

      footer: function(currentPage,
           pageCount) { return {text:
           currentPage.toString() + ' of ' +
            pageCount, alignment:'center'
           };},
      content: [
        { text: "<?php echo $election_name."
             Election Results";?> ",
            fontSize: 15, alignment:'center
            ', margin: 5 },

        <?php $i=0;
        for($i=0;$i<count($candidates);){
          $row = $candidates[$i];
          $position_id = $row['position_id
               '];

        if($i!=0) echo ","; ?>{ text: "<?php
             echo $row['position_name'];?>
            ", bold:true, margin:3},

        {
          table: {
            headerRows: 1,
            widths: [ 'auto', '*', '*', '
                 auto'],
            body: [
              [ {text: 'No',bold:true}, {
                   text:'Candidate',bold:
                   true}, {text:'Party',bold
                   :true}, {text:'Number of
                   Votes',bold:true}],

            <?php $k=0; $sortedarray = array
                 ();
            while($i<count($candidates) &&
                 $row['position_id'] ==
                 $position_id){
              if(empty($row['party_name']))
                   $row['party_name'] = "
                   INDEPENDENT";
              $sortedarray[$k] = array("
                   last_name" => $row['
                   last_name'], "first_name"
                    =>$row["first_name"],
                "party_name" =>$row['
                     party_name'], "
                     vote_count" =>$row['
                     vote_count']);
              $i++; $k++;
              if($i < count($candidates)){
                $row = $candidates[$i];
              }
            }

            uasort($sortedarray, function (
                 $x, $y) {
              $a = $x['vote_count'];
              $b = $y['vote_count'];
              if ($a == $b) return 0;
              elseif ($a > $b) return -1;
              else return 1;
            });

            $j = 1;
            foreach($sortedarray as $srow){
              if ($j!=1) echo ",";?>
              [ "<?php echo $j;?>", "<?php
                   echo $srow['last_name
                   ']." , ".$srow["first_name
                   "];?>", "<?php echo $srow
                   ['party_name'];?>", "<?
                   php echo $srow['
                   vote_count'];?>" ]
              <?php $j++;
            }?>
            ]
          }
        }

        <?php } ?>
      ]
    };

    //pdfMake.createPdf(docDefinition).open
         ();
    pdfMake.createPdf(docDefinition).
         download('<?php echo str_replace("
         ","_",$election_name."_election")
         ;?>.pdf');
  });

});

</script>
```

[leftmargin=0cm,itemindent=0cm]

iii. navbar.php

```
<nav class=”navbar navbar−default”>
<img src=”<?php echo asset_url();?>img/
     upm_logo.png” alt=”UPM Logo” align=’
     left’>
  <div class=”container−fluid”>
    <div class=”navbar−header”>
      <a class=”navbar−brand” href=”#”>
           Votemind</a>
    </div>
    <div class=”collapse navbar−collapse” id
         =”myNavbar”>
      <ul class=”nav navbar−nav”>
        <li>
          <div class=’a−btn’><a class=’btn
               btn−default’ href=”<?php echo
                linkbase_url();?>login/
               results”>Results</a></div>
        </li>
      </ul>
      <ul class=”nav navbar−nav”>
        <li>
          <div class=’a−btn’><a class=’btn
               btn−default’ href=”<?php echo
                linkbase_url();?>login/
               bulletin”>Bulletin Board</a
               ></div>
        </li>
      </ul>
      <ul class=”nav navbar−nav navbar−right
           ”>
        <li><div class=’a−btn’><a class=’btn
             btn−success’ href=”<?php echo
             linkbase_url();?>login”>
          <span class=”glyphicon glyphicon−
               log−in”></span> Login</a></
               div>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

iv. view.php

```
<div class=”content container−fluid text−
     center”>
  <div class=”form”>
    <?php echo form_open(’login’); ?>
      <div class=’form−group’ align=’center’>
        <label for=”username ”>Username:</
             label>
        <input type=”text” name=”username” id
             =”username” class=”form−control”
             style=’width:300px;’>
      </div>
      <div class=’alert−danger’><?php echo
          form_error(’username’); ?></div>
      <div class=’form−group’ align=’center’>
        <label for=”password ”>Password:</
             label>
        <input type=”password” name=”password”
             id=”password” class=”form−
             control” style=’width:300px’>
      </div>
      <div class=’alert−danger’><?php echo
          form_error(’password’); ?></div>
      <input type=”submit” value=”Log in”
          class=’btn btn−success’>
    </form>
  </div>
</div>
```

(b) admin

i. add_modal.php

```
<!−− Add Modal −−>
<div id=”addModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal−dialog”>

    <!−− Add Modal content−−>
    <?php echo form_open(’admin/vos/add’); ?>
    <div class=”modal−content”>

      <div class=”modal−header”>
        <button type=”button” class=”close”
             data−dismiss=”modal”>&times;</
             button>
        <h4 class=”modal−title”><span class=”
             glyphicon glyphicon−plus”></span>
              Add a Voting Official</h4>
      </div>

      <div class=”modal−body”>
        <div class=”modal−form”>

            <div class=’form−group’>
              <label>Username: </label>
              <input type=’text’ name=’
                   username’ required>
            </div>

            <div class=’form−group’>
              <label>Email: </label>
              <input type=’text’ name=’email’
                   required>
            </div>

        </div>
      </div>

      <div class=”modal−footer”>
        <button type=”submit” class=”btn btn−
             success pull−left”>
          <span class=”glyphicon glyphicon−
               plus”></span> Add Voting
               Official
        </button>
        <button type=”submit” class=”btn btn−
             danger pull−right” data−dismiss=”
             modal”><span class=”glyphicon
             glyphicon−remove”></span> Cancel
             </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

ii. change_passowrd_modal.php

```
<!−− Delete Modal −−>
<div id=”changePasswordModal” class=”modal
     fade” role=”dialog”>
  <div class=”modal−dialog”>

    <!−− Delete Modal content−−>
    <?php echo form_open(’admin/profile/delete
         ’); ?>
    <div class=”modal−content”>

      <div class=”modal−header modal−header−
           red”>
        <button type=”button” class=”close”
             data−dismiss=”modal”>&times;</
             button>
        <h4 class=”modal−title”><span class=’
             glyphicon glyphicon−lock’></span>
              Change Password</h4>
      </div>

      <div class=”modal−body”>
        <div class=”modal−form”>

            <div class=’form−group’>
              <label>Current Password: </label
                   >
              <input type=”password” name=”
                   current_password” class=”
                   form−control” required>
            </div>

            <div class=’form−group’>
```

```
              <label>New Password: </label>
              <input type=”password” name=”
                   new_password” class=”form-
                   control” required>
            </div>

            <div class=’form-group’>
              <label>Verify New Password: </
                   label>
              <input type=”password” name=”
                   verify_password” class=”
                   form-control” required>
            </div>

        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”>Change</button
             >
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”>Cancel</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

iii. delete.php

```
<div class=”content container-fluid text-
     center”>
  <h2>List of voting officials</h2>
  <hr>
  <?php if(!empty($vos)){ ?>
    <div class=’alert alert-info fade in’>
      <strong>Info:</strong> Check the table
           rows you want to delete.
    </div>
    <?php echo form_open(’admin/delete’); ?>
      <table id=’delete_table’ class=”table
           text-left”>
        <thead>
          <tr>
            <th><input class=’allcheckable’
                 type=”checkbox” name=”
                 votingofficialall” value=”all
                 ” onclick=”toggle(this);”></
                 th>
            <th>User ID</th>
            <th>Username</th>
            <th>Email</th>
          </tr>
        </thead>
        <tbody>
        <?php
        foreach($vos as $row){ ?>
          <tr class=’table-row’>
            <td><input type=”checkbox” class=’
                 checkable’ name=”
                 votingofficial[]” id=<?php
                 echo $row[’user_id’];?>
               value=<?php echo $row[’user_id
                    ’];?>></td>
            <td><?php echo $row[’user_id
                 ’];?></td>
            <td><?php echo $row[’username
                 ’];?></td>
            <td><?php echo $row[’email’];?></
                 td>
          </tr>
        <?php }?>
        </tbody>
      </table>
      <button id=’delete_button’ type=”submit”
           class=”btn btn-danger btn-md pull-
           center”>
        <span class=”glyphicon glyphicon-
             remove”></span> Delete Rows</
             button>
    </form>
  <?php }else{ ?>
    <div class=’alert alert-danger’><strong>
         Error!</strong> There are no voting
         officials.</div>
    <a href=”<?php echo linkbase_url();?>admin
         /add” class=’btn btn-default’>Add a
         list</a>
  <?php } ?>
</div>
```

iv. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
     =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open(’admin/vos/delete’);
         ?>
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
           red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Are you sure
             you want to delete this voting
             official?</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

            <input type=’hidden’ name=’user_id
                 []’ id=’deleteModal_userid’>
            <div class=’form-group’>
              <label>Username: </label>
              <span id=’deleteModal_username
                   ’></span>
            </div>

            <div class=’form-group’>
              <label>Email: </label>
              <span id=’deleteModal_email’></
                   span>
            </div>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”>Yes</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”>No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

v. delete_selected_modal.php

```
<!-- Delete Modal -->
<div id=”deleteSelectedModal” class=”modal
     fade” role=”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open(’admin/vos/delete’);
         ?>
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
           red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Are you sure
             you want to delete these voting
             officials?</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left” id=’
             delete_selected_button’>Yes</
             button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”>No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

vi. edit_modal.php

```
<!-- Edit Modal -->
<div id=”editModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal-dialog”>

    <!-- Modal content-->
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit Voting Official</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

          <?php echo form_open(’admin/vos/edit
               ’); ?>
            <div class=’form-group’>
              <label><span class=”glyphicon
                   glyphicon-user”></span>
                   User ID: <span id=’
                   editModal_userID’> </span>
                   </label>
              <input type=’hidden’ name=’
                   userid’ id=’
                   editModal_huserid’>
            </div>

            <div class=’form-group’>
              <label for=”username”>Username
                   :</label>
              <input type=”text” name=”
                   username” id=”
                   editModal_username” class=”
                   form-control” required>
            </div>

            <div class=’form-group’>
              <label for=”email”>Email:</
                   label>
              <input type=”text” name=”email”
                   id=”editModal_email” class
                   =”form-control” required>
            </div>
            <div class=’alert-danger’><?php
                 echo form_error(’email’);
                 ?></div>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit User</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Cancel
             </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

vii. edit_profile_modal.php

```
<!-- Edit Modal -->
<div id=”editProfileModal” class=”modal fade”
     role=”dialog”>
  <div class=”modal-dialog”>

    <!-- Modal content-->
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit Profile</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

          <?php echo form_open(’admin/profile/
               edit’); ?>

            <div class=’form-group’>
              <label for=”username”>Username
                   :</label>
              <input type=”text” name=”
                   username” id=”
                   editProfileModal_username”
                   class=”form-control”
                value=”<?php echo $username
                     ;?>” required>
            </div>

            <div class=’form-group’>
              <label for=”email”>Email:</
                   label>
              <input type=”text” name=”email”
                   id=”editProfileModal_email”
                    class=”form-control” value
                   =”<?php echo $email;?>”
                   required>
            </div>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Cancel
             </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

viii. footer.php

```
      <!-- <div id=”footer”>Copyright &copy;
           2016 Cybernetica AS</div> -->
  <footer class=”container-fluid text-left
       navbar-default”>
    <p>&copy; University of the Philippines
         Manila</p>
  </footer>
  <script src=”<?php echo asset_url();?>js/
       jquery-1.12.0.min.js”></script>
  <script src=”<?php echo asset_url();?>js/
       bootstrap.js”></script>
  <script src=”<?php echo asset_url();?>js/
       jquery.dataTables.min.js”></script>
  <script src=”<?php echo asset_url();?>js/
       dataTables.bootstrap.min.js”></script>
  <script type=”text/javascript”>
```

```
// function toggle(source) {
//   checkboxes = document.
     getElementsByName(’votingofficial[]’)
     ;
//   for(var i=0, n=checkboxes.length;i<n;
     i++) {
//     checkboxes[i].checked = source.
     checked;
//   }
// }

// function isThereChecked() {
//   checkboxes = document.
     getElementsByName(’votingofficial[]’)
     ;
//   for(var i=0, n=checkboxes.length;i<n;
     i++) {
//     if(checkboxes[i].checked)
//       return true;
//   }
//   return false;
// }

$(document).ready(function () {
  $(’.content’).hide();
  $(’.content’).delay(200).slideDown(800);
  $(”#admin_table”).dataTable({
    ”aoColumnDefs”: [
      { ’bSortable’: false, ’aTargets’: [
          3,4 ] }
    ]
  });

  $(’#editModal’).on(’show.bs.modal’,
      function(e) {
    var userid = $(e.relatedTarget).data(’
        userid’);
    var username = $(e.relatedTarget).data
        (’username’);
    var email = $(e.relatedTarget).data(’
        email’);
    $(e.currentTarget).find(’#
        editModal_userID’).html(’ ’+
        userid);
    $(e.currentTarget).find(’#
        editModal_huserid’).val(userid);
    $(e.currentTarget).find(’#
        editModal_username’).val(username
        );
    $(e.currentTarget).find(’#
        editModal_email’).val(email);
  });

  $(’#deleteModal’).on(’show.bs.modal’,
      function(e) {
    var userid = $(e.relatedTarget).data(’
        userid’);
    var username = $(e.relatedTarget).data
        (’username’);
    var email = $(e.relatedTarget).data(’
        email’);
    $(e.currentTarget).find(’#
        deleteModal_userid’).val(userid);
    $(e.currentTarget).find(’#
        deleteModal_username’).html(
        username);
    $(e.currentTarget).find(’#
        deleteModal_email’).html(email);
  });

  $(’#allvotingofficials’).on(’click’,
      function(e){
    $(”.checkable”).prop(’checked’,$(this)
        .prop(’checked’));
  });

  $(’#deleteSelectedModal’).on(’show.bs.
      modal’, function(e){
    var html = ””;
    var isThereChecked = false;
    $(”.checkable”).each(function(e){
      if($(this).attr(’name’)!=’
          allvotingofficials’ && $(this).
          prop(’checked’) == true){
        isThereChecked = true;
        html += ”<div class=’form-group’>
            <label>Username: ”+$(this).
            attr(’username’)+”</label></
            div>”;
        html += ”<input type=’hidden’ name
            =’user_id[]’ value=’”+$(this)
            .val()+”’>”;
      }
    });

    if(isThereChecked == true){
      $(e.currentTarget).find(’#
          delete_selected_button’).show()
          ;
      $(e.currentTarget).find(’.modal-form
          ’).html(html);
    }else{
      $(e.currentTarget).find(’#
          delete_selected_button’).hide()
          ;
      $(e.currentTarget).find(’.modal-form
          ’).html(”There are no selected
          voting officials.”);
    }
  });

  // $(’#delete_button’).hide();

  // $(’.checkable’).change(function() {
  //   $(this).closest(’tr’).toggleClass(’
      alert-danger’, this.checked);
  //   if($(’.checkable’).is(’:checked’)){
  //     $(’#delete_button’).fadeIn();
  //   }else{
  //     $(’#delete_button’).fadeOut();
  //   }
  // });

  // $(’.allcheckable’).change(function()
      {
  //   $(’.table-row’).toggleClass(’alert-
      danger’, this.checked);
  //   if(this.checked){
  //     $(’#delete_button’).fadeIn();
  //   }else{
  //     $(’#delete_button’).fadeOut();
  //   }
  // });

  // $(’.table-row’).click(function(e){
  //   var userid = $(e.currentTarget).
      find(’#userid’).html();
  //   alert(userid);
  //   var isChecked = $(e.currentTarget).
      find(”input[name=’votingofficial
      ’]”).is(’:checked’);
  //   $(e.currentTarget).find(”input[name
      =’votingofficial’]”).prop(’checked
      ’, !isChecked);
  // });

});
</script>
</body>
</html>
```

ix. import_modal.php

```
<!-- Import Modal -->
<div id=”importModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Import Modal content-->
    <?php echo form_open_multipart(’admin/vos/
        import’); ?>
    <div class=”modal-content”>

      <div class=”modal-header”>
        <button type=”button” class=”close”
            data-dismiss=”modal”>&times;</
            button>
        <h4 class=”modal-title”><span class=”
            glyphicon glyphicon-upload”></
            span> Import Voting Officials’
            list</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>
          <div class=’form-group’>

            <div class=’alert alert-info fade
                in text-center’>
              <strong>Format of CSV:</strong>
                  username,email <br> (eg.
                  vo1,vo1@email.com)
            </div>
            <input type=”file” name=”userfile”
                class=’file btn btn-default’
                required>

          </div>
        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
            success pull-left”>
          <span class=”glyphicon glyphicon-
              upload”></span> Import CSV
        </button>
        <button type=”submit” class=”btn btn-
            danger pull-right” data-dismiss=”
            modal”><span class=”glyphicon
            glyphicon-remove”></span> Cancel
            </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

x. navbar.php

```
<nav class=”navbar navbar-default”>
<img src=”<?php echo asset_url();?>img/
    upm_logo.png” alt=”UPM Logo” align=’
    left’>
  <div class=”container-fluid”>
    <div class=”navbar-header”>
      <a class=”navbar-brand” href=”#”>
          Votemind</a>
    </div>
    <div class=”collapse navbar-collapse” id
        =”myNavbar”>
      <ul class=”nav navbar-nav”>
        <li>
          <div class=’a-btn’><a class=’btn
              btn-default’ href=”<?php echo
               linkbase_url();?>admin/vos”>
              Voting Officials</a></div>
        </li>
      </ul>
      <ul class=”nav navbar-nav navbar-right
          ”>
          <li>
          <div class=”dropdown”>
            <button class=”btn btn-warning
                dropdown-toggle” type=”
                button” data-toggle=”
                dropdown”>
              <span class=’glyphicon
                  glyphicon-user’></span>
                  <?php echo $username;?>
            <span class=”caret”></span></
                button>
            <ul class=”dropdown-menu”>
              <li><a href=”<?php echo
                  linkbase_url();?>admin/
                  profile”><span class=’
                  glyphicon glyphicon-cog
                  ’></span> Profile</a></li
                  >
              <li><a href=”<?php echo
                  linkbase_url();?>admin/
                  logout”><span class=”
                  glyphicon glyphicon-log-
                  out”></span> Logout</a></
                  li>
            </ul>
          </div>
          </li>
      </ul>
    </div>
  </div>
</nav>
```

xi. profile.php

```
<div class=”content container-fluid text-
    center”>

  <h2>Your Profile</h2>

  <hr>

  <div class=’container-fluid’ style=’padding-
      bottom: 20px;’>

    <?php if(!empty($error)){ ?>
      <div class=’alert alert-danger’><strong
          ><?php echo $error;?></strong></div
          >
    <?php } ?>

    <?php if(!empty($success)){ ?>
      <div class=’alert alert-success’><strong
          ><?php echo $success;?></strong></
          div>
    <?php } ?>

    <table id=’profile_table’ class=”table
        table-striped text-center” style=’
        width:100%;’>
      <tr>
        <td>
          <strong>Username: <?php echo
              $username;?></strong>
        </td>
      </tr>

      <tr>
        <td>
          <strong>Email: <?php echo $email
              ;?></strong>
        </td>
      </tr>

      <tr>
        <td>
          <strong>Role: <?php echo $role;?></
              strong>
        </td>
      </tr>
    </table>

    <div class=’text-center’>

      <button class=”btn btn-success btn-md”
          data-toggle=”modal” data-target=”#
          editProfileModal”>
        <span class=”glyphicon glyphicon-
            pencil”></span> Edit Profile
      </button>

      <button class=”btn btn-danger btn-md”
          data-toggle=”modal” data-target=”#
          changePasswordModal”>
        <span class=”glyphicon glyphicon-lock
            ”></span> Change Password
      </button>

    </div>

  </div>

</div>
```

xii. view.php

```
<div class=”content container-fluid text-
    center”>

  <h2>List of Voting Officials</h2>

  <hr>

  <div class=’container-fluid’ style=’padding-
      bottom: 20px;’>

    <div class=’row’>

      <button class=”btn btn-success btn-md
          pull-left col-sm-2” data-toggle=”
          modal” data-target=”#addModal”>
        <span class=”glyphicon glyphicon-plus
            ”></span> Add Voting Official
      </button>

      <button class=”btn btn-success btn-md
          col-sm-2” style=’margin-left: 10px
          ;’ data-toggle=”modal” data-target
          =”#importModal”>
        <span class=”glyphicon glyphicon-
            upload”></span> Import CSV
      </button>
```

```
        <button class='btn btn-danger btn-md
             pull-right' data-toggle=”modal”
             data-target=”#deleteSelectedModal”>
        <span class=”glyphicon glyphicon-trash
             ”></span> Delete selected</button>

      </div>

    </div>

    <?php if(!empty($error)){ ?>
      <div class='alert alert-danger fade in'>
          <a href=”#” class=”close” data-dismiss
               =”alert” aria-label=”close”>&
               times;</a>
          <?php echo $error;?>
      </div>
    <?php } ?>

    <?php if(!empty($votingOfficials)){ ?>

      <div class='alert alert-info fade in'>
        <strong>Info:</strong> Click <span class
             =”glyphicon glyphicon-pencil”></
             span> to edit and
        <span class=”glyphicon glyphicon-trash
             ”></span> to delete the row.
      </div>

      <div class='text-left'>
      <table id='admin_table' class=”table table
           -hover text-left” style='width
           :100%;'>
        <thead >
          <tr>
            <th>ID</th>
            <th>Username</th>
            <th>Email</th>
            <th>Action</th>
            <th>
              <label><input type=”checkbox”
                   class='checkable form-control
                   ' name=”allvotingofficials”
                   id=”allvotingofficials”></
                   label>
            </th>
          </tr>
        </thead>
        <tbody>
        <?php
        foreach($votingOfficials as $row){ ?>
          <tr class='table-row'>
            <td><?php echo $row['user_id'];?></
```

```
               td>
          <td><?php echo $row['username'];?></
               td>
          <td><?php echo $row['email'];?></td>
          <td>
            <button class=”btn btn-success btn
                 -sm” data-toggle=”modal” data
                 -userid=<?php echo $row['
                 user_id'];?>
              data-username=”<?php echo $row['
                   username'];?>” data-email
                   =”<?php echo $row['email
                   '];?>”
              data-target=”#editModal”>
              <span class=”glyphicon glyphicon
                   -pencil”></span></button>
            <button class=”btn btn-danger btn-
                 sm” data-toggle=”modal” data-
                 userid=<?php echo $row['
                 user_id'];?>
              data-username=”<?php echo $row['
                   username'];?>” data-email
                   =”<?php echo $row['email
                   '];?>”
              data-target=”#deleteModal”>
              <span class=”glyphicon glyphicon
                   -trash”></span></button>
          </td>
          <td>
            <input type=”checkbox” class='
                 checkable form-control'
                 username=”<?php echo $row['
                 username'];?>”
             value=<?php echo $row['user_id
                 '];?>>
          </td>
        </tr>
      <?php }?>
      </tbody>
    </form>
    </table>
    </div>

    <?php }else{ ?>

      <div class='alert alert-danger'><strong>
           Error!</strong> There are no voting
            officials.</div>

    <?php } ?>

</div>
```

(c) votingofficial

i. candidates

A. add_modal.php

```
<!-- Add Modal -->
<div id=”addModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal-dialog”>

    <!-- Add Modal content-->
    <?php echo form_open('votingofficial/
         candidates/add'); ?>
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-plus”></span>
              Add a Candidate</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

            <div class='form-group'>
              <label>First Name: </label>
              <input type='text' name='
                   first_name' required>
            </div>

            <div class='form-group'>
              <label>Last Name: </label>
              <input type='text' name='
                   last_name' required>
            </div>

            <div class='form-group'>
              <label>Election: </label>
              <span><?php echo $election_name
                   ;?></span>
            </div>
```

```
            <div class='form-group'>
              <label>Position: </label>
                <select name='position_id'
                     class='form-control input
                     -lg modal-form-control'
                     required>
                <option disabled selected
                     value> -- select an
                     option -- </option>
                <?php foreach($positions as
                     $row){ ?>
                  <option value=<?php echo
                       $row['position_id
                       '];?>><?php echo $row['
                       position_name'];?></
                       option>
                <?php
                } ?>
                </select>
            </div>

            <div class='form-group'>
              <label>Party: </label>
                <select name='party_id' class
                     ='form-control input-lg
                     modal-form-control'
                     required>
                <option disabled selected
                     value> -- select an
                     option -- </option>
                <?php foreach($parties as $row
                     ){ ?>
                  <option value=<?php echo
                       $row['party_id'];?>><?
                       php echo $row['
                       party_name'];?></option
            >
        <?php
        } ?>
          <option value=”-1”>
               INDEPENDENT</option>
        </select>
      </div>
    </div>
  </div>
  <div class=”modal-footer”>
    <button type=”submit” class=”btn btn-
         success pull-left”><span class=”
         glyphicon glyphicon-plus”></span>
         Add Candidate</button>
    <button type=”submit” class=”btn btn-
         danger pull-right” data-dismiss=”
         modal”><span class=”glyphicon
         glyphicon-remove”></span> Cancel
         </button>
    </form>
  </div>
</div>

   </div>
</div>
```

B. candidates.php

```
<div class=”content container-fluid text-
     center”>

  <h2>List of Candidates</h2>
  <hr>
  <div class='form-horizontal'>
    <div class='form-group'>
      <label class='control-label input-lg col
           -sm-4'> Select an election: </label
           >
      <div class=”col-sm-6”>
        <select name='election_name' id='
             select_election_id' class='form-
             control input-lg'>
        <?php if(!empty($election_id)){ ?>
            <option value=<?php echo
                 $election_id; ?> default> <?
                 php echo $election_name; ?>
                 </option>
        <?php } else { ?>
          <option id='option_placeholder'
               default> None </option>
        <?php } foreach($elections as $row){
          if($row['election_id']!=$election_id
              ) {?>
            <option value=<?php echo $row['
                 election_id'];?>><?php echo
                 $row['name'];?></option>
        <?php }
          } ?>
      </select>
      </div>
    </div>
  </div>

  <hr>

  <div id='candidates_table'>

  </div>

</div>
```

C. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
     =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open('votingofficial/
         candidates/delete'); ?>
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
           red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Are you sure
             you want to delete this candidate
             ?</h4>
      </div>

      <div class=”modal-body”>

        <div class=”modal-form”>

            <input type='hidden' name='
                 candidate_id' id='
                 deleteModal_candidateid' >

            <div class='form-group'>
              <label>First Name: </label>
              <span id='deleteModal_firstname
                   '></span>
            </div>

            <div class='form-group'>
              <label>Last Name: </label>
              <span id='deleteModal_lastname
                   '></span>
            </div>
            <div class='form-group'>
              <label>Election: </label>
              <span><?php echo $election_name
                   ;?></span>
            </div>

            <div class='form-group'>
              <label>Position: </label>
                <span id='
                     deleteModal_positionname
                     '></span>
            </div>

            <div class='form-group'>
              <label>Party: </label>
                <span id='
                     deleteModal_partyname'></
                     span>
            </div>

        </div>

      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”>Yes</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”>No</button>
        </form>
      </div>

    </div>

  </div>
</div>
```

D. edit_modal.php

```
<!-- Edit Modal -->
<div id=”editModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal-dialog”>

    <!-- Edit Modal content-->
    <?php echo form_open('votingofficial/
         candidates/edit'); ?>
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit Candidate</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

            <input type='hidden' name='
                 candidate_id' id='
                 editModal_candidateid' >

            <div class='form-group'>
              <label>First Name: </label>
              <input type='text' name='
                   first_name' id='
                   editModal_firstname'
                   required>
            </div>

            <div class='form-group'>
              <label>Last Name: </label>
              <input type='text' name='
                   last_name' id='
                   editModal_lastname'
                   required>
            </div>

            <div class='form-group'>
              <label>Election: </label>
              <span><?php echo $election_name
                   ;?></span>
            </div>

            <div class='form-group'>
              <label>Position: </label>
                <select name='position_id' id
                     ='editModal_positionid'
                     class='form-control input
                     -lg modal-form-control'
                     required>
                <?php foreach($positions as
                     $row){ ?>
                  <option value=<?php echo
                       $row['position_id
                       '];?>><?php echo $row['
                       position_name'];?></
                       option>
                <?php
                } ?>
                </select>
            </div>
            <div class='form-group'>
              <label>Party: </label>
                <select name='party_id' id='
                     editModal_partyid' class
                     ='form-control input-lg
                     modal-form-control'
                     required>
                <?php foreach($parties as $row
                     ){ ?>
                  <option value=<?php echo
                       $row['party_id'];?>><?
                       php echo $row['
                       party_name'];?></option
                       >
                <?php
                } ?>
                  <option value="-1">
                       INDEPENDENT</option>
                </select>
            </div>
        </div>
      </div>

      <div class="modal-footer">
        <button type="submit" class="btn btn-
             success pull-left"><span class="
             glyphicon glyphicon-pencil"></
             span> Edit Candidate</button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal"><span class="glyphicon
             glyphicon-remove"></span> Cancel
             </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

E. footer.php

```
    <!-- <div id="footer">Copyright &copy;
         2016 Cybernetica AS</div> -->
<footer class="container-fluid text-left
     navbar-default">
  <p>&copy; University of the Philippines
       Manila</p>
</footer>
<script src="<?php echo asset_url();?>js/
     jquery-1.12.0.min.js"></script>
<script src="<?php echo asset_url();?>js/
     bootstrap.js"></script>
<script type="text/javascript">

  $(document).ready(function () {
    $('.content').hide();
    $('.content').delay(200).slideDown(800);

    var election_id = $("#select_election_id
         ").val();

        jQuery.ajax({
          type: "POST",
          url: "<?php echo linkbase_url();
               ?>" + "votingofficial/
               candidates/getCandidates",
          data: {election_id: election_id},
          success: function(res) {
            if (res)
            {
              $('#candidates_table').html(
                   res);
              $('#candidates_table').delay
                   (200).slideDown(800);
            }
          }
          });

      $('#select_election_id').change(function
           () {
        $('#candidates_table').slideUp(800,
             function(){
           $('#option_placeholder').remove();
          var election_id = $("#
               select_election_id").val();

          jQuery.ajax({
            type: "POST",
            url: "<?php echo linkbase_url();
                 ?>" + "votingofficial/
                 candidates/getCandidates",
            data: {election_id: election_id},
            success: function(res) {
              if (res)
              {
                $('#candidates_table').html(
                     res);
                $('#candidates_table').delay
                     (200).slideDown(800);
              }
            }
          });
        });
      });

    });

  </script>
</body>
</html>
```

F. modal_scripts.php

```
<script type="text/javascript">

  $(document).ready(function () {

    $('#editModal').on('show.bs.modal',
         function(e) {
      var candidate_id = $(e.relatedTarget).
           data('candidateid');
      var first_name = $(e.relatedTarget).
           data('firstname');
      var last_name = $(e.relatedTarget).
           data('lastname');
      var position_id = $(e.relatedTarget).
           data('positionid');
      var party_id = $(e.relatedTarget).data
           ('partyid');
      $(e.currentTarget).find('#
           editModal_candidateid').val(
           candidate_id);
      $(e.currentTarget).find('#
           editModal_firstname').val(
           first_name);
      $(e.currentTarget).find('#
           editModal_lastname').val(
           last_name);
      $(e.currentTarget).find('#
             editModal_positionid ').val(
             position_id);
        $(e.currentTarget).find('#
             editModal_partyid ').val(party_id)
             ;
      });

      $('#deleteModal').on('show.bs.modal',
           function(e) {
        var candidate_id = $(e.relatedTarget).
             data('candidateid');
        var first_name = $(e.relatedTarget).
             data('firstname');
        var last_name = $(e.relatedTarget).
             data('lastname');
        var position_name = $(e.relatedTarget)
             .data('positionname');
        var party_name = $(e.relatedTarget).
             data('partyname');
        $(e.currentTarget).find('#
             deleteModal_candidateid ').val(
             candidate_id);
        $(e.currentTarget).find('#
             deleteModal_firstname ').html(
             first_name);
        $(e.currentTarget).find('#
             deleteModal_lastname ').html(
             last_name);
        $(e.currentTarget).find('#
             deleteModal_positionname ').html(
             position_name);
        $(e.currentTarget).find('#
             deleteModal_partyname ').html(
             party_name);
      });

    });

</script>
```

G. table.php

```
<?php if($election_status != 0 &&
     $election_status != 1){ ?>
<div class='container-fluid' style='padding-
     bottom: 20px;'>
  <button class="btn btn-success btn-md pull
       -left" data-toggle="modal" data-
       target="#addModal">
  <span class="glyphicon glyphicon-plus"></
       span> Add a Candidate</button>
</div>
<?php } ?>

<?php if(!empty($error)){ ?>
  <div class='alert alert-danger fade in'>
      <a href="#" class="close" data-dismiss
           ="alert" aria-label="close">&
           times;</a>
      <?php echo $error;?>
  </div>
<?php } ?>

<?php if(!empty($candidates)){ ?>

  <?php if($election_status != 0 &&
       $election_status != 1){ ?>
  <div class='alert alert-info fade in'>
    <strong>Info:</strong> Click <span class
         ="glyphicon glyphicon-pencil"></
         span> to edit and
      <span class="glyphicon glyphicon-trash
           "></span> to delete the row.
  </div>
  <?php } ?>

  <?php $i=0;
  for($i=0;$i<count($candidates);){
    $row = $candidates[$i];
    $position_id = $row['position_id'];?>
  <div class='well well-sm'><strong><?php
       echo $row['position_name'];?></strong
       ></div>
  <table class="table table-hover text-left
       ">
    <thead >
      <tr>
        <th>No.</th>
        <th>Candidate</th>
        <th>Party</th>
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
        <th>Action</th>
        <?php } ?>
      </tr>
    </thead>
    <tbody>
      <tr class='table-row'>
        <?php $j = 1;
          while($i<count($candidates) &&
               $row['position_id'] ==
               $position_id){ ?>
        <td><?php echo $j;?></td>
        <td><?php echo $row['last_name']."，
             ".$row["first_name"];?></td>
        <td><?php if(!empty($row['party_name
             '])) echo $row['party_name'];
             else echo "INDEPENDENT";?></td>
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
          <td>
            <button class="btn btn-success
                 btn-sm" data-toggle="modal"
                  data-candidateid=<?php
                 echo $row['candidate_id
                 '];?>
            data-firstname="<?php echo $row
                 ["first_name"];?>" data-
                 lastname="<?php echo $row["
                 last_name"];?>"
            data-partyid=<?php if(!empty(
                 $row['party_id'])) echo
                 $row['party_id']; else echo
                  "-1";?> data-positionid=<?
                 php echo $row['position_id
                 '];?> data-target="#
                 editModal">
              <span class="glyphicon
                   glyphicon-pencil"></span
                   ></button>
            <button class="btn btn-danger
                 btn-sm" data-toggle="modal"
                  data-candidateid=<?php
                 echo $row['candidate_id
                 '];?>
            data-firstname="<?php echo $row
                 ["first_name"];?>" data-
                 lastname="<?php echo $row["
                 last_name"];?>"
            data-partyname=<?php if(!empty(
                 $row['party_name'])) echo
                 $row['party_name']; else
                 echo "INDEPENDENT";?>
            data-positionname=<?php echo
                 $row['position_name'];?>
            data-target="#deleteModal">
              <span class="glyphicon
                   glyphicon-trash"></span
                   ></button>
          </td>
        <?php } ?>
      </tr>
    <?php $i++; $j++;
      if($i < count($candidates))
        $row = $candidates[$i];
    }?>
    </tbody>
  </table>
  <?php }
  }else{ ?>
  <div class='alert alert-danger'><strong>
       Error!</strong> There are no
       candidates in this election.</div>
<?php } ?>
```

ii. elections

A. add_modal.php

```
<!-- Add Modal -->
<div id="addModal" class="modal fade" role="
     dialog">
  <div class="modal-dialog">

    <!-- Add Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class="
```

```
             glyphicon glyphicon-plus”></span>
              Add an Election</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>
          <?php echo form_open(’votingofficial
               /elections/add’); ?>
            <div class=’form-group’>
              <label>Election Name: </label>
              <input type=’text’ name=’name’
                   class=’form-control input-
                   lg modal-form-control’
                   required>
            </div>
            <div class=’form-group’>
              <label>Parent: </label>
              <select name=’parent’ class=’
                   form-control input-lg modal
                   -form-control’ required>
                <option value=’-1’ default>
                     None </option>
              <?php foreach($elections as $row
                   ){ ?>
                <option value=<?php echo $row
                     [’election_id’];?>><?php
                     echo $row[’name’];?></
                     option>
              <?php } ?>
              </select>
            </div>
        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”><span class=”
             glyphicon glyphicon-plus”></span>
              Add Election</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Cancel
             </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

B. change_modal.php

```
<!-- Change Modal -->
<div id=”changeModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Change Modal content-->
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
           blue”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Start/Stop
             Election</h4>
      </div>

      <div class=”modal-body”>

        <div class=”modal-form”>

          <?php $attributes = array(’id’ => ’
               change_election_form’);
          echo form_open(’votingofficial/
               elections/change’, $attributes)
               ; ?>
            <input type=’hidden’ name=’
                 election_id’ id=’
                 changeModal_eid’>
            <input type=’hidden’ name=’status’
                  id=’changeModal_status’>
          </form>

            <div class=’form-group’>
              <label>Election Name: </label>
              <span id=’changeModal_name’></
                   span>
            </div>

            <div class=’form-group’>
              <label>Parent: </label>
              <span id=’changeModal_parent’>
                   </span>
            </div>

            <div class=’form-group’>
              <label>Start Date: </label>
              <span id=’changeModal_startdate
                   ’> </span>
            </div>

            <div class=’form-group’>
              <label>End Date: </label>
              <span id=’changeModal_enddate’>
                   </span>
            </div>

            <div class=’form-group’ id=’
                 message’>
            </div>

        </div>

      </div>

      <div class=”modal-footer”>

        <button type=”submit” class=”btn btn-
             success pull-left” id=’
             changeButton’>
          <span class=”glyphicon”></span>
        </button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Cancel
             </button>

      </div>
    </div>

  </div>
</div>
```

C. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <div class=”modal-content”>
      <div class=”modal-header modal-header-
           red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Are you sure
             you want to delete this election
             ?</h4>
      </div>
      <div class=”modal-body text-center”>
        <h3 class=’alert alert-danger’>All
             related positions, parties,
             candidates will be deleted as
             well.</h4>
        <div class=”modal-form”>
          <?php $attributes = array(’id’ => ’
               delete_election_form’);
          echo form_open(’votingofficial/
               elections/delete’,$attributes);
                ?>
          <input type=’hidden’ name=’
               election_id’ id=’
               deleteModal_eid’>
          </form>
            <div class=’form-group’>
              <label>Election Name: </label>
              <span id=’deleteModal_name’></
                   span>
            </div>
            <div class=’form-group’>
              <label>Parent: </label>
              <span id=’deleteModal_parent’></
                   span>
            </div>
        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left” id=’
             delete_election_yes_button'>Yes</
             button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal">No</button>
      </div>
    </div>
  </div>
</div>
```

D. edit_modal.php

```
<!-- Edit Modal -->
<div id="editModal" class="modal fade" role="
    dialog">
  <div class="modal-dialog">

    <!-- Edit Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
            data-dismiss="modal">&times;</
            button>
        <h4 class="modal-title"><span class="
            glyphicon glyphicon-pencil"></
            span> Edit Election</h4>
      </div>
      <div class="modal-body">
        <div class="modal-form">
          <?php echo form_open('votingofficial
              /elections/edit'); ?>
          <input type='hidden' name='
              election_id' id='editModal_eid'
               required>
            <div class='form-group'>
              <label>Election Name: </label>
              <input type='text' name='name'
                  id='editModal_name' class='
                  form-control input-lg modal
                  -form-control' required>
            </div>
            <div class='form-group'>
              <label>Parent: </label>
              <div class='editModal_parent'>
                <select name='parent' class='
                    form-control input-lg
                    modal-form-control'
                    required>
                  <option value='-1'> None </
                      option>
                <?php foreach($elections as
                    $row){ ?>
                  <option value=<?php echo
                      $row['election_id
                      '];?>><?php echo $row['
                      name'];?></option>
                <?php } ?>
                </select>
              </div>
            </div>
        </div>
      </div>
      <div class="modal-footer">
        <button type="submit" class="btn btn-
            success pull-left"><span class="
            glyphicon glyphicon-pencil"></
            span> Edit Election</button>
        <button type="submit" class="btn btn-
            danger pull-right" data-dismiss="
            modal"><span class="glyphicon
            glyphicon-remove"></span> Cancel
            </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

E. elections.php

```
<div class="content container-fluid text-
    center">

  <h2>List of Elections</h2>
  <hr>

  <?php if(!empty($error)){ ?>
    <div class='alert alert-danger fade in'>
        <a href="#" class="close" data-dismiss
            ="alert" aria-label="close">&
            times;</a>
        <?php echo $error;?>
    </div>
  <?php } ?>

  <div class='container-fluid' style='padding-
      bottom: 20px;'>
      <button class="btn btn-success btn-md
          pull-left" data-toggle="modal" data
          -target="#addModal">
      <span class="glyphicon glyphicon-plus
          "></span> Add an Election</button>
    </div>

  <?php if(!empty($elections)){ ?>

    <div class='alert alert-info fade in'>
      <strong>Info:</strong> Click on the blue
            button to change the status of the
            election. <br><span class="
           glyphicon glyphicon-pencil"></span>
            to edit. <span class="glyphicon
           glyphicon-trash"></span> to delete.
    </div>

    <div class='text-left'>
    <table id='elections_table' class="table
        table-hover text-left" style='width
        :100%;'>
      <thead >
        <tr>
          <th>Name</th>
          <th>Parent ID</th>
          <th>Status</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
      <?php $i=0;
      foreach($elections as $row){ ?>
        <tr class='table-row'>
          <td><?php echo $row['name'];?></td>
          <td><?php echo $row['parent_name
              '];?></td>
          <td>
            <?php
              if($row['status']==-2){
                echo 'Not ready';
              }else if($row['status']==-1){
                echo 'Ready';
              }else if($row['status']==0){
                echo 'Started';
              }else if($row['status']==1){
                echo 'Ended and Posted';
              }
            ?>
          </td>
          <td>
            <button class="btn btn-primary btn
                -sm" data-toggle="modal" data
                -electionid=<?php echo $row['
                election_id'];?>
              data-name="<?php echo $row['name
                  '];?>" data-parentid=<?php
                  echo $row['parent_id'];?>
              data-parentname="<?php echo $row
                  ['parent_name'];?>" data-
                  startdate="<?php echo $row
                  ['start_date'];?>"
              data-enddate="<?php echo $row['
                  end_date'];?>" data-status
                  =<?php echo $row['status
                  '];?> data-target="#
                  changeModal">
              <span class="glyphicon
                <?php
                  if($row['status']==-2){
                    echo 'glyphicon-question-
                        sign';
                  }else if($row['status']==-1)
                      {
                    echo 'glyphicon-play';
                  }else if($row['status']==0){
                    echo 'glyphicon-stop';
                  }else if($row['status']==1){
                    echo 'glyphicon-repeat';
                  }
                ?>"
              ></span>
            </button>
```

```
<?php if($row['status'] != 0 &&
     $row['status'] != 1 && $row['
     partOfVoterType'] == false)
     {?>

<button class="btn btn-success btn
     -sm" data-toggle="modal" data
     -electionid=<?php echo $row['
     election_id'];?>
  data-name="<?php echo $row['name
       '];?>" data-parentid=<?php
       echo $row['parent_id'];?>
  data-parentname="<?php echo $row
       ['parent_name'];?>" data-
       target="#editModal">
  <span class="glyphicon glyphicon
       -pencil"></span></button>

<button class="btn btn-danger btn-
     sm" data-toggle="modal" data-
     electionid=<?php echo $row['
     election_id'];?>
  data-name="<?php echo $row['name
       '];?>" data-parentname="<?
       php echo $row['parent_name
       '];?>"data-target="#
       deleteModal">
  <span class="glyphicon glyphicon
       -trash"></span></button>

  <?php /*if($row['status']==1){
       ?>

    <button class="btn btn-warning
          btn-sm" data-toggle="
```

```
                    modal" data-electionid=<?
                    php echo $row['
                    election_id'];?>
                data-name="<?php echo $row['
                    name'];?>" data-parentid
                    =<?php echo $row['
                    parent_id'];?>
                data-parentname="<?php echo
                    $row['parent_name'];?>"
                    data-startdate="<?php
                    echo $row['start_date
                    '];?>"
                data-enddate="<?php echo $row
                    ['end_date'];?>" data-
                    target="#resultModal">
                <span class="glyphicon
                    glyphicon-list-alt"></
                    span></button>

              <?php } */
            }?>

          </td>
        </tr>
      <?php }?>
      </tbody>
    </table>
    </div>

    <?php }else{ ?>
      <div class='alert alert-danger'><strong>
           Error!</strong> There are no
           elections.</div>
    <?php } ?>
</div>
```

F. footer.php

```
    <!-- <div id="footer">Copyright &copy;
         2016 Cybernetica AS</div> -->
<footer class="container-fluid text-left
     navbar-default">
  <p>&copy; University of the Philippines
       Manila</p>
</footer>
<script src="<?php echo asset_url();?>js/
     jquery-1.12.0.min.js"></script>
<script src="<?php echo asset_url();?>js/
     bootstrap.js"></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/bootbox.min.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/ext/typedarray.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/ext/jsaes.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/ext/jsbn.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/ext/jsbn2.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/ext/socket.io.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/prng.js"></script
     >
<script type="text/javascript" src="<?php
     echo asset_url();?>js/sm.types.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/sm.types.util.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/sm.types.base.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/sm.types.shared3p
     .js"></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/serialization.js
     "></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/sm.ctrl.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/miners.js"></
     script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/app.js"></script>
<script type="text/javascript" src="<?php
     echo asset_url();?>js/page.js"></script
     >
```

```
<script src="<?php echo asset_url();?>js/
     jquery.dataTables.min.js"></script>
<script src="<?php echo asset_url();?>js/
     dataTables.bootstrap.min.js"></script>
<script type="text/javascript">

  $(document).ready(function () {
    $('.content').hide();
    $('.content').delay(200).slideDown(800);
    $("#elections_table").dataTable({
      "aoColumnDefs": [
        { 'bSortable': false, 'aTargets':
             [3] }
     ]
    });

    $('body').css({'cursor' : 'default'});

    $('#changeModal').on('show.bs.modal',
         function(e) {
      var election_id = $(e.relatedTarget).
           data('electionid');
      var name = $(e.relatedTarget).data('
           name');
      var parent_name = $(e.relatedTarget).
           data('parentname');
      var status = $(e.relatedTarget).data('
           status');
      var start_date = $(e.relatedTarget).
           data('startdate');
      var end_date = $(e.relatedTarget).data
           ('enddate');
      $(e.currentTarget).find('#
           changeModal_eid').val(election_id
           );
      $(e.currentTarget).find('#
           changeModal_name').html(name);
      $(e.currentTarget).find('#
           changeModal_parent').html(
           parent_name);
      $(e.currentTarget).find('#
           changeModal_startdate').html(
           start_date);
      $(e.currentTarget).find('#
           changeModal_enddate').html(
           end_date);
      $(e.currentTarget).find('#changeButton
           ').show();

      if(status == 0){
        $(e.currentTarget).find('#message').
             html("");
        $(e.currentTarget).find('#
             changeButton').html("Stop
             Election");
        $(e.currentTarget).find('#
             changeModal_status').val(1);
      }else if(status == -1){
```

```
      $(e.currentTarget).find('#message').
          html("");
      $(e.currentTarget).find('#
          changeButton').html("Start
          Election");
      $(e.currentTarget).find('#
          changeModal_status').val(0);
    }else if(status == 1){
      $(e.currentTarget).find('#message').
          html("");
      $(e.currentTarget).find('#
          changeButton').html("Start
          Election Again");
      $(e.currentTarget).find('#
          changeModal_status').val(-1);
    }
    else if(status == -2){
      $(e.currentTarget).find('#message').
          html("<div class='alert alert-
          danger'><strong>You must have
          at least one position, one
          party, one candidate, one voter
           type, and two voters
          registered on this election.</
          strong></div>");
      $(e.currentTarget).find('#
          changeButton').hide();
      $(e.currentTarget).find('#
          changeModal_status').val(-2);
    }

  });

  $('#changeButton').on('click',function(e
      ){

    var status = $(this).html();

    if(status == "Start Election Again" ){

      $('body').css({'cursor' : 'wait'});
      var election_id = $('#
          changeModal_eid').val();
      var clbck = function(err_msg) {
        if (err_msg) {
          showMessage('Something went
              wrong. Look for details in
              the console.', 'error');
        }else{
          setTimeout(function(){
            $('#change_election_form').
                submit();
          }, 500);
        }
      };
      app.deleteElection(election_id,clbck
          );

    }else if(status == "Stop Election"){

      $('body').css({'cursor' : 'wait'});
      var election_id = $('#
          changeModal_eid').val();
      var clbck = function(err_msg,
          results) {
        if (!results) {
          //showMessage('Something went
              wrong. Look for details in
              the console.', 'error');
          setTimeout(function(){
            $('#change_election_form').
                submit();
          }, 500);
        }else{
          //do something else
          var content = $('#
              change_election_form').html
              ();
          for(i=0; i<results['histogram'].
              length; i++){
            content += "<input type='
                hidden' name='
                num_of_votes[]' value="+
                results['histogram'][i
                ]+">";
          }
          $('#change_election_form').html(
              content);
          setTimeout(function(){
            $('#change_election_form').
                submit();
          }, 500);
        }
      };

      jQuery.ajax({
        type: "POST",
        url: "<?php echo linkbase_url();
            ?>" + "votingofficial/
            elections/postResults",
        data: {election_id: election_id},
        dataType : 'json',
        success: function(res) {
          if (res)
          {
            var positionIds = [];
            var candidateIds = [];
            for(i=0;i<res.length;i++){
              positionIds.push(res[i]['
                  position_id']);
              candidateIds.push(res[i]['
                  candidate_id']);
            }
            app.getResults(election_id,
                positionIds,candidateIds,
                clbck);
          }
        }
      });
    }
    else{
      $('#change_election_form').submit();
    }

  });

   $('#resultModal').on('show.bs.modal',
       function(e) {
    var election_id = $(e.relatedTarget).
        data('electionid');
    var name = $(e.relatedTarget).data('
        name');
    var parent_name = $(e.relatedTarget).
        data('parentname');
    var start_date = $(e.relatedTarget).
        data('startdate');
    var end_date = $(e.relatedTarget).data
        ('enddate');
    $(e.currentTarget).find('#
        resultModal_eid').val(election_id
        );
    $(e.currentTarget).find('#
        resultModal_name').html(name);
    $(e.currentTarget).find('#
        resultModal_parent').html(
        parent_name);
    $(e.currentTarget).find('#
        resultModal_startdate').html(
        start_date);
    $(e.currentTarget).find('#
        resultModal_enddate').html(
        end_date);
  });

  $('#editModal').on('show.bs.modal',
      function(e) {
    var election_id = $(e.relatedTarget).
        data('electionid');
    var name = $(e.relatedTarget).data('
        name');
    var parent_id = $(e.relatedTarget).
        data('parentid');
    var parent_name = $(e.relatedTarget).
        data('parentname');
    $(e.currentTarget).find('#
        editModal_eid').val(election_id);
    $(e.currentTarget).find('#
        editModal_name').val(name);
    $(e.currentTarget).find("div.
        editModal_parent select").val(
        parent_id);
    $(e.currentTarget).find("option").show
        ();
    $(e.currentTarget).find("option[value
        ="+election_id+"]").hide();
  });

  $('#deleteModal').on('show.bs.modal',
      function(e) {
    var election_id = $(e.relatedTarget).
        data('electionid');
    var name = $(e.relatedTarget).data('
        name');
    var parent_name = $(e.relatedTarget).
        data('parentname');
    $(e.currentTarget).find('#
        deleteModal_eid').val(election_id
        );
    $(e.currentTarget).find('#
        deleteModal_name').html(name);
    $(e.currentTarget).find('#
        deleteModal_parent').html(
        parent_name);
  });

  $('#delete_election_yes_button').on('
      click',function(e){
    $('body').css({'cursor' : 'wait'});
```

```
      var election_id = $(’#deleteModal_eid
          ’).val();
      var clbck = function(err_msg) {
        if (err_msg) {
          showMessage(’Something went wrong.
               Look for details in the
               console.’, ’error’);
        }else{
          setTimeout(function(){
            $(’#delete_election_form’).
                submit();
          }, 500);
        }
      };
      app.deleteElection(election_id,clbck);
    });

    // $(’#postButton’).on(’click’, function
        (e) {
    //   var election_id = $(’#
        resultModal_eid’).val();
    //   jQuery.ajax({
    //     type: ”POST”,
    //     url: ”<?php echo linkbase_url();
        ?>” + ”votingofficial/elections/
        postResults”,
    //     data: {election_id: election_id},
    //     dataType : ’json’,
    //     success: function(res) {
    //       if (res)
    //       {

    //         var positionIds = [];
    //         var candidateIds = [];
    //         for(i=0;i<res.length;i++){
    //           positionIds.push(res[i][’
        position_id’]);
    //           candidateIds.push(res[i][’
        candidate_id’]);
    //         }

    //         var clbck = function(err_msg,
         results) {
    //           if (!results) {
    //             showMessage(’Something
        went wrong. Look for details in the
         console.’, ’error’);
    //           }else{
    //             //pag nakuha results
        ilagay sa server tapos refresh page
    //           }
    //         };

    //         app.getResults(positionIds,
        candidateIds,clbck);
    //       }
    //     }
    //   });
    // });

    app.init();
  });
  </script>
</body>
</html>
```

G. result_modal.php

```
<!−− Result Modal −−>
<div id=”resultModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal−dialog”>

    <!−− Change Modal content−−>
    <div class=”modal−content”>

      <div class=”modal−header modal−header−
          orange”>
        <button type=”button” class=”close”
            data−dismiss=”modal”>&times;</
            button>
        <h4 class=”modal−title”>Post Results
            of Election</h4>
      </div>

      <div class=”modal−body”>

        <div class=”modal−form”>

          <?php $attributes = array(’id’ => ’
              post_election_form’);
          echo form_open(’votingofficial/
              elections/results’,$attributes)
              ; ?>
          <input type=’hidden’ name=’
              election_id’ id=’
              resultModal_eid’>
          </form>

            <div class=’form−group’>
              <label>Election Name: </label>
              <span id=’resultModal_name’></
                  span>
            </div>

            <div class=’form−group’>
              <label>Parent: </label>
              <span id=’resultModal_parent’>
                  </span>
            </div>

            <div class=’form−group’>
              <label>Start Date: </label>
              <span id=’resultModal_startdate
                  ’> </span>
            </div>

            <div class=’form−group’>
              <label>End Date: </label>
              <span id=’resultModal_enddate’>
                  </span>
            </div>

        </div>

      </div>

      <div class=”modal−footer”>

        <button type=”submit” class=”btn btn−
            success pull−left” id=’postButton
            ’>
          <span class=”glyphicon glyphicon−
              list−alt”></span> Post
        </button>
        <button type=”submit” class=”btn btn−
            danger pull−right” data−dismiss=”
            modal”><span class=”glyphicon
            glyphicon−remove”></span> Cancel
            </button>

      </div>
    </div>

  </div>
</div>
```

iii. notification

A. delete_modal.php

```
<!−− Delete Modal −−>
<div id=”deleteModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal−dialog”>

    <!−− Delete Modal content−−>
    <?php echo form_open(’votingofficial/
        notifications/delete’); ?>
    <div class=”modal−content”>

      <div class=”modal−header modal−header−
          red”>
        <button type=”button” class=”close”
            data−dismiss=”modal”>&times;</
            button>
        <h4 class=”modal−title”>Are you sure
            you want to delete this
            notification?</h4>
      </div>

      <div class=”modal−body”>
        <div class=”modal−form”>

            <input type=’hidden’ name=’
                notif_id’ id=’notif_id’>
            <div class=’form−group’>
              <label>ID: </label>
              <span id=’deleteModal_id’></span
                  >
            </div>
          <div class='form-group'>
            <label>Title: </label>
            <span id='deleteModal_title'></
                span>
          </div>

          <div class='form-group'>
            <label>Remarks: </label>
            <span id='deleteModal_remarks
                '></span>
          </div>

      </div>
    </div>
```

```
      <div class="modal-footer">
        <button type="submit" class="btn btn-
            success pull-left">Yes</button>
        <button type="submit" class="btn btn-
            danger pull-right" data-dismiss="
            modal">No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

B. footer.php

```
    <!-- <div id="footer">Copyright &copy;
        2016 Cybernetica AS</div> -->
<footer class="container-fluid text-left
    navbar-default">
  <p>&copy; University of the Philippines
      Manila</p>
</footer>
<script src="<?php echo asset_url();?>js/
    jquery-1.12.0.min.js"></script>
<script src="<?php echo asset_url();?>js/
    bootstrap.js"></script>
<script src="<?php echo asset_url();?>js/
    jquery.dataTables.min.js"></script>
<script src="<?php echo asset_url();?>js/
    dataTables.bootstrap.min.js"></script>
<script type="text/javascript">

  $(document).ready(function () {
    $('.content').hide();
    $('.content').delay(200).slideDown(800);
    $("#notif_table").dataTable({
      "aoColumnDefs": [
        { 'bSortable': false, 'aTargets':
            [6] }
       ]
    });

    $('#infoModal').on('show.bs.modal',
        function(e) {
      var title = $(e.relatedTarget).data('
          title');
      var body = $(e.relatedTarget).data('
          body');
      var remarks = $(e.relatedTarget).data
          ('remarks');
      $(e.currentTarget).find('#
          infoModal_title').html(title);
      $(e.currentTarget).find('#
          infoModal_body').html(body);
      $(e.currentTarget).find('#
          infoModal_remarks').html(remarks)
```

```
            ;
      });

      $('#deleteModal').on('show.bs.modal',
          function(e) {
        var notif_id = $(e.relatedTarget).data
            ('id');
        var title = $(e.relatedTarget).data('
            title');
        var remarks = $(e.relatedTarget).data
            ('remarks');
        $(e.currentTarget).find('#notif_id').
            val(notif_id);
        $(e.currentTarget).find('#
            deleteModal_title').html(title);
        $(e.currentTarget).find('#
            deleteModal_id').html(notif_id);
        $(e.currentTarget).find('#
            deleteModal_remarks').html(
            remarks);
      });

      $('#resolveModal').on('show.bs.modal',
          function(e) {
        var notif_id = $(e.relatedTarget).data
            ('id');
        var title = $(e.relatedTarget).data('
            title');
        $(e.currentTarget).find('#notif_id').
            val(notif_id);
        $(e.currentTarget).find('#
            resolveModal_title').html(title);
        $(e.currentTarget).find('#
            resolveModal_id').html(notif_id);
      });

    });
  </script>
</body>
</html>
```

C. info_modal.php

```
<!-- Delete Modal -->
<div id="infoModal" class="modal fade" role="
    dialog">
  <div class="modal-dialog">

    <!-- Delete Modal content-->
    <div class="modal-content">

      <div class="modal-header modal-header-
          blue">
        <button type="button" class="close"
            data-dismiss="modal">&times;</
            button>
        <h4 class="modal-title"><span class="
            glyphicon glyphicon-info-sign"></
            span>Additional Information</h4>
      </div>

      <div class="modal-body">
        <div class="modal-form">

            <div class='form-group'>
              <label>Title: </label>
```

```
              <span id='infoModal_title'></
                  span>
            </div>

            <div class='form-group'>
              <label>Body: </label>
              <span id='infoModal_body'></span
                  >
            </div>

            <div class='form-group'>
              <label>Remarks: </label>
              <span id='infoModal_remarks'></
                  span>
            </div>

        </div>
      </div>
    </div>
  </div>
</div>
```

D. mainpage.php

```
<div class="content container-fluid text-
    center">

  <h2>Notifications</h2>

  <hr>
```

```
  <div class='container-fluid' style='padding-
      bottom: 20px;'>

    <?php if(!empty($error)){ ?>
      <div class='alert alert-danger'><strong
```

```
           ><?php echo $error;?></strong></div
           >
    <?php } ?>

    <?php if(!empty($success)){ ?>
      <div class=’alert alert-success’><strong
           ><?php echo $success;?></strong></
           div>
    <?php } ?>

    <?php if(!empty($notifications)){ ?>
    <div class=’text-left’>
    <table id=’notif_table’ class=”table table
         -striped” style=’width:100%;’>

      <thead>
        <tr>
          <th>ID</th>
          <th>Title</th>
          <th>Date Sent</th>
          <th>Sender</th>
          <th>Date Resolved</th>
          <th>Resolver</th>
          <th>Actions</th>
        </tr>
      </thead>

      <tbody>
      <?php
      foreach($notifications as $row){ ?>
        <tr class=’table-row’>
          <td><?php echo $row[’notif_id’];?></
               td>
          <td><?php echo $row[’title’];?></td>
          <td><?php echo $row[’date_sent
               ’];?></td>
          <td><?php echo $row[’voter_id’];?></
               td>
          <td><?php echo $row[’date_resolved
               ’];?></td>
          <td><?php echo $row[’vo_id’];?></td>
          <td>
            <button class=”btn btn-primary btn
                 -sm” data-toggle=”modal” data
                 -title=”<?php echo $row[’
                 title’];?>”
            data-body=”<?php echo $row[’body
                 ’];?>” data-remarks=”<?php
                 echo $row[’remarks’];?>” data
                 -target=”#infoModal”>
              <span class=”glyphicon glyphicon
                   -info-sign”></span></button
                   >
            <?php if(empty($row[’date_resolved
                 ’])){ ?>
            <button class=”btn btn-success btn
                 -sm” data-toggle=”modal” data
                 -title=”<?php echo $row[’
                 title’];?>”
            data-id=”<?php echo $row[’notif_id
                 ’];?>” data-target=”#
                 resolveModal”>
              <span class=”glyphicon glyphicon
                   -ok-circle”></span></button
                   >
            <?php } ?>
            <button class=”btn btn-danger btn-
                 sm” data-toggle=”modal” data-
                 title=”<?php echo $row[’title
                 ’];?>”
            data-id=”<?php echo $row[’notif_id
                 ’];?>” data-remarks=”<?php
                 echo $row[’remarks’];?>” data
                 -target=”#deleteModal”>
              <span class=”glyphicon glyphicon
                   -trash”></span></button>
          </td>
        </tr>
      <?php }?>
      </tbody>

    </table>
    </div>

    <?php }else{ ?>

      <div class=’alert alert-danger’>There
           are no notifications.</div>

    <?php } ?>

  </div>

</div>
```

E. resolve_modal.php

```
<!-- Add Modal -->
<div id=”resolveModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Add Modal content-->
    <?php echo form_open(’votingofficial/
         notifications/resolve’); ?>
    <div class=”modal-content”>

      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-check”></span
             > Mark Case as Resolved</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

            <input type=’hidden’ name=’
                 notif_id’ id=’notif_id’>
            <div class=’form-group’>
              <label>ID: </label>
              <span id=’resolveModal_id’></
                   span>
            </div>

            <div class=’form-group’>
              <label>Title: </label>
              <span id=’resolveModal_title’></
                   span>
            </div>

            <div class=’form-group’>
              <label>Remarks: </label>
              <div class=’text-center’><
                   textarea rows=”4” name=”
                   remarks” id=”remarks” class
                   =”form-control” required></
                   textarea>
                </div>
            </div>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”>
          <span class=”glyphicon glyphicon-
               check”></span> Resolved
        </button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Not yet
             </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

iv. parties

A. add_modal.php

```
<!-- Add Modal -->
<div id=”addModal” class=”modal fade” role=”
    dialog”>
  <div class=”modal-dialog”>

    <!-- Add Modal content-->
    <?php echo form_open(’votingofficial/
         parties/add’); ?>
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
```

```
              glyphicon glyphicon-plus”></span>
               Add a Party</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

            <div class=’form-group’>
              <label>Party Name: </label>
              <input type=’text’ name=’
                  party_name’ required>
            </div>
            <div class=’form-group’>
              <label>Election: </label>
              <span><?php echo $election_name
                  ;?></span>
            </div>
        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
            success pull-left”><span class=”
            glyphicon glyphicon-plus”></span>
             Add Party</button>
        <button type=”submit” class=”btn btn-
            danger pull-right” data-dismiss=”
            modal”><span class=”glyphicon
            glyphicon-remove”></span> Cancel
            </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

B. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open(’votingofficial/
        parties/delete’); ?>
    <div class=”modal-content”>
      <div class=”modal-header modal-header-
          red”>
        <button type=”button” class=”close”
            data-dismiss=”modal”>&times;</
            button>
        <h4 class=”modal-title”>Are you sure
            you want to delete this party?</
            h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>
          <input type=’hidden’ name=’party_id’
              id=’deleteModal_pid’>
            <div class=’form-group’>
              <label>Party Name: </label>
              <span id=’deleteModal_name’></
                  span>
            </div>
            <div class=’form-group’>
              <label>Election: </label>
              <span id=’deleteModal_election
                  ’><?php echo $election_name
                  ;?></span>
            </div>
        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
            success pull-left”>Yes</button>
        <button type=”submit” class=”btn btn-
            danger pull-right” data-dismiss=”
            modal”>No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

C. edit_modal.php

```
<!-- Edit Modal -->
<div id=”editModal” class=”modal fade” role=”
    dialog”>
  <div class=”modal-dialog”>

    <!-- Edit Modal content-->
    <?php echo form_open(’votingofficial/
        parties/edit’); ?>
    <div class=”modal-content”>
      <div class=”modal-header”>
        <button type=”button” class=”close”
            data-dismiss=”modal”>&times;</
            button>
        <h4 class=”modal-title”><span class=”
            glyphicon glyphicon-pencil”></
            span> Edit Party</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

          <input type=’hidden’ name=’party_id’
              id=’editModal_partyid’>
          <input type=’hidden’ name=’
              original_eid’ value=<?php echo
              $election_id;?>>

            <div class=’form-group’>
              <label>Party Name: </label> <br>
              <input type=’text’ name=’
                  party_name’ id=’
                  editModal_name’ class=’form
                  -control input-lg modal-
                  form-control’ required>
            </div>

            <div class=’form-group’>
              <label>Election: </label>
              <div class=’editModal_election’>
                <select name=’election_id’
                    class=’form-control input
                    -lg modal-form-control’
                    required>
                  <option value=<?php echo
                      $election_id;?> default
                      > <?php echo
                      $election_name;?> </
                      option>
                <?php foreach($elections as
                    $row){
                  if($row[’election_id’]!=
                      $election_id){?>
                  <option value=<?php echo
                      $row[’election_id
                      ’];?>><?php echo $row[’
                      name’];?></option>
                <?php }
                } ?>
                  </select>
              </div>
            </div>

        </div>
      </div>
      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
            success pull-left”><span class=”
            glyphicon glyphicon-pencil”></
            span> Edit Party</button>
        <button type=”submit” class=”btn btn-
            danger pull-right” data-dismiss=”
            modal”><span class=”glyphicon
            glyphicon-remove”></span> Cancel
            </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

D. footer.php

```
    <!-- <div id=”footer”>Copyright &copy;
         2016 Cybernetica AS</div> -->
  <footer class=”container-fluid text-left
       navbar-default”>
    <p>&copy; University of the Philippines
         Manila
  <script src=”<?php echo asset_url();?>js/
       jquery-1.12.0.min.js”></script>
  <script src=”<?php echo asset_url();?>js/
       bootstrap.js”></script>

  <script type=”text/javascript”>

    $(document).ready(function () {
      $('.content').hide();
      $('.content').delay(200).slideDown(800);

      var election_id = $(”#select_election_id
           ”).val();

          jQuery.ajax({
            type: ”POST”,
            url: ”<?php echo linkbase_url();
                 ?>” + ”votingofficial/parties
                 /getParties”,
            data: {election_id: election_id},
            success: function(res) {
              if (res)
              {
                $('#parties_table').html(res);
                $('#parties_table').delay(200)
                     .slideDown(800);
              }
            }
          });

      $('#select_election_id').change(function
           () {
        $('#parties_table').slideUp(800,
             function(){
          $('#option_placeholder').remove();
          var election_id = $(”#
               select_election_id”).val();

          jQuery.ajax({
            type: ”POST”,
            url: ”<?php echo linkbase_url();
                 ?>” + ”votingofficial/parties
                 /getParties”,
            data: {election_id: election_id},
            success: function(res) {
              if (res)
              {
                $('#parties_table').html(res);
                $('#parties_table').delay(200)
                     .slideDown(800);
              }
            }
          });
        });
      });

    });

  </script>
</body>
</html>
```

E. modal_scripts.php

```
  <script src=”<?php echo asset_url();?>js/
       jquery.dataTables.min.js”></script>
  <script src=”<?php echo asset_url();?>js/
       dataTables.bootstrap.min.js”></script>
  <script type=”text/javascript”>
    $(document).ready(function () {
      $(”#elections_table”).dataTable({
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
        ”aoColumnDefs”: [
          { 'bSortable': false, 'aTargets': [
               1] }
       ]<?php } ?>
      });

      $('#editModal').on('show.bs.modal',
           function(e) {
        var party_id = $(e.relatedTarget).data
             ('partyid');
        var party_name = $(e.relatedTarget).
             data('partyname');
        $(e.currentTarget).find('#
             editModal_partyid').val(party_id)
             ;
        $(e.currentTarget).find('#
             editModal_name').val(party_name);
      });

      $('#deleteModal').on('show.bs.modal',
           function(e) {
        var party_id = $(e.relatedTarget).data
             ('partyid');
        var party_name = $(e.relatedTarget).
             data('partyname');
        $(e.currentTarget).find('#
             deleteModal_pid').val(party_id);
        $(e.currentTarget).find('#
             deleteModal_name').html(
             party_name);
      });

    });

  </script>
```

F. parties.php

```
<div class=”content container-fluid text-
     center”>

  <h2>List of Parties</h2>
  <hr>
  <div class='form-horizontal'>
    <div class='form-group'>
      <label class='control-label input-lg col
           -sm-4'> Select an election: </label
           >
      <div class=”col-sm-6”>
        <select name='election_name' id='
             select_election_id' class='form-
             control input-lg'>
        <?php if(!empty($election_id)){ ?>
            <option value=<?php echo
                 $election_id; ?> default> <?
                 php echo $election_name; ?>
                 </option>
        <?php } else { ?>
          <option id='option_placeholder'
               default> None </option>
        <?php } foreach($elections as $row){
          if($row['election_id']!=$election_id
               ) {?>
            <option value=<?php echo $row['
                 election_id'];?>><?php echo
                 $row['name'];?></option>
        <?php }
          } ?>
        </select>
        </div>
      </div>
    </div>

    <hr>

    <div id='parties_table'>

    </div>

</div>
```

G. table.php

```
  <?php if($election_status != 0 &&
       $election_status != 1){ ?>
  <div class='container-fluid' style='padding-
       bottom: 20px;'>
    <button class=”btn btn-success btn-md pull
         -left” data-toggle=”modal” data-
         target=”#addModal”>
      <span class=”glyphicon glyphicon-plus”></
           span> Add a Party</button>
    </div>
    <?php } ?>
```

```
<?php if(!empty($error)){ ?>
  <div class='alert alert-danger fade in'>
      <a href="#" class="close" data-dismiss
           ="alert" aria-label="close">&
           times;</a>
      <?php echo $error;?>
  </div>
<?php } ?>
<?php if(!empty($parties)){ ?>

  <?php if($election_status != 0 &&
       $election_status != 1){ ?>
  <div class='alert alert-info fade in'>
    <strong>Info:</strong> Click <span class
         ="glyphicon glyphicon-pencil"></
         span> to edit and
      <span class="glyphicon glyphicon-trash
           "></span> to delete the row.
  </div>
  <?php } ?>

  <div class='text-left'>
  <table id='elections_table' class="table
       table-hover text-left" style='width
       :100%'>
    <thead >
      <tr>
        <th>Name</th>
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
        <th>Action</th>
        <?php } ?>
      </tr>
    </thead>
    <tbody>
    <?php foreach($parties as $row){ ?>
      <tr class='table-row'>
      <td><?php echo $row['party_name
           '];?></td>
      <?php if($election_status != 0 &&
           $election_status != 1){ ?>
      <td>
        <button class="btn btn-success btn
             -sm" data-toggle="modal" data
             -partyid='<?php echo $row['
             party_id'];?>'
        data-partyname='<?php echo $row['
             party_name'];?>' data-target
             ="#editModal">
          <span class="glyphicon glyphicon
               -pencil"></span></button>
        <button class="btn btn-danger btn-
             sm" data-toggle="modal" data-
             partyid='<?php echo $row['
             party_id'];?>'
        data-partyname='<?php echo $row['
             party_name'];?>' data-target
             ="#deleteModal">
          <span class="glyphicon glyphicon
               -trash"></span></button>
      </td>
      <?php } ?>
    </tr>
  <?php }?>
  </tbody>
</table>
</div>

<?php }else{ ?>
  <div class='alert alert-danger'><strong>
       Error!</strong> There are no
       parties in this election.</div>
<?php } ?>
```

v. positions

A. add_modal.php

```
<!-- Add Modal -->
<div id="addModal" class="modal fade" role="
     dialog">
  <div class="modal-dialog">

    <!-- Add Modal content-->
    <?php echo form_open('votingofficial/
         positions/add'); ?>
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class="
             glyphicon glyphicon-plus"></span>
              Add a Position</h4>
      </div>
      <div class="modal-body">
        <div class="modal-form">

            <div class='form-group'>
              <label>Position Name: </label> <
                   br>
              <input type='text' name='
                   position_name' class='form-
                   control input-lg modal-form
                   -control' required>
            </div>
            <div class='form-group'>
              <label>Election: <?php echo
                   $election_name;?> </label>
              <input type='hidden' name='
                   election_id' value=<?php
                   echo $election_id;?>>
            </div>
            <div class='form-group'>
              <label>Max number of candidates
                   to be elected for the
                   position: </label>
              <input type='number' name='max'
                   min=1 value=1 class='form-
                   control input-lg modal-form
                   -control' required>
            </div>
          </div>
      </div>
      <div class="modal-footer">
        <button type="submit" class="btn btn-
             success pull-left"><span class="
             glyphicon glyphicon-plus"></span>
              Add Position</button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal"><span class="glyphicon
             glyphicon-remove"></span> Cancel
             </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

B. delete_modal.php

```
<!-- Delete Modal -->
<div id="deleteModal" class="modal fade" role
     ="dialog">
  <div class="modal-dialog">

    <!-- Delete Modal content-->
    <?php echo form_open('votingofficial/
         positions/delete'); ?>
    <div class="modal-content">
      <div class="modal-header modal-header-
           red">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title">Are you sure
             you want to delete this position
             ?</h4>
      </div>
      <div class="modal-body">
        <div class="modal-form">
          <input type='hidden' name='
               position_id' id='
               deleteModal_pid'>
          <div id='deleteModal_pids'></div>

            <div class='form-group'>
              <label>Position Name: </label>
              <span id='deleteModal_name'></
                   span>
            </div>
            <div class='form-group'>
              <label>Election: </label>
              <span id='deleteModal_election
                   '><?php echo $election_name
                   ;?></span>
            </div>
        </div>
      </div>
      <div class="modal-footer">
        <button type="submit" class="btn btn-
             success pull-left">Yes</button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal">No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

C. edit_modal.php

```
<!-- Edit Modal -->
<div id="editModal" class="modal fade" role="
     dialog">
  <div class="modal-dialog">

    <!-- Edit Modal content-->
    <?php echo form_open('votingofficial/
         positions/edit'); ?>
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class="
             glyphicon glyphicon-pencil"></
             span> Edit Position</h4>
      </div>
      <div class="modal-body">
        <div class="modal-form">

          <input type='hidden' name='
               position_id' id='
               editModal_positionid'>

            <div class='form-group'>
              <label>Position Name: </label> <
                   br>
              <input type='text' name='
                   position_name' id='
                   editModal_name' class='form
                   -control input-lg modal-
                   form-control' required>
            </div>

            <div class='form-group'>
              <label>Election: </label>
              <span id='editModal_election'><?
                   php echo $election_name;
                   ?></span>
            </div>

            <div class='form-group'>
              <label>Max number of candidates
                   to be elected for the
                   position: </label>
              <input type='number' name='max'
                   min=1 id='editModal_max'
                   class='form-control input-
                   lg modal-form-control'
                   required>
            </div>

        </div>
      </div>
      <div class="modal-footer">
        <button type="submit" class="btn btn-
             success pull-left"><span class="
             glyphicon glyphicon-pencil"></
             span> Edit Position</button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal"><span class="glyphicon
             glyphicon-remove"></span> Cancel
             </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

D. footer.php

```
    <!-- <div id="footer">Copyright &copy;
         2016 Cybernetica AS</div> -->
<footer class="container-fluid text-left
     navbar-default">
  <p>&copy; University of the Philippines
       Manila</p>
</footer>
<script src="<?php echo asset_url();?>js/
     jquery-1.12.0.min.js"></script>
<script src="<?php echo asset_url();?>js/
     bootstrap.js"></script>
<script type="text/javascript">

  $(document).ready(function () {
    $('.content').hide();
    $('.content').delay(200).slideDown(800);

    var election_id = $("#select_election_id
         ").val();

        jQuery.ajax({
          type: "POST",
          url: "<?php echo linkbase_url();
               ?>" + "votingofficial/
               positions/getPositions",
          data: {election_id: election_id},
          success: function(res) {
            if (res)
            {
              $('#positions_table').html(res
                   );
              $('#positions_table').delay
                   (200).slideDown(800);
            }
          }
          });

      $('#select_election_id').change(function
           () {
        $('#positions_table').slideUp(800,
             function(){
          $('#option_placeholder').remove();
          var election_id = $("#
               select_election_id").val();

          jQuery.ajax({
            type: "POST",
            url: "<?php echo linkbase_url();
                 ?>" + "votingofficial/
                 positions/getPositions",
            data: {election_id: election_id},
            success: function(res) {
              if (res)
              {
                $('#positions_table').html(res
                     );
                $('#positions_table').delay
                     (200).slideDown(800);
              }
            }
          });
        });
      });

    });

  </script>
</body>
</html>
```

E. modal_scripts.php

```
<script src="<?php echo asset_url();?>js/
     jquery.dataTables.min.js"></script>
<script src="<?php echo asset_url();?>js/
     dataTables.bootstrap.min.js"></script>
<script src="<?php echo asset_url();?>js/
     dataTables.rowReorder.min.js"></script>
<script type="text/javascript">
    $(document).ready(function () {

      var isFirst = true;

      var elections_table = $("#
           elections_table").dataTable({
        "paging": false,
      "search":false,
      "bFilter": false,
      <?php if($election_status != 0 && $election_status != 1){ ?>
      "aoColumnDefs": [
        { 'bSortable': false, 'aTargets': [2] },
        { orderable: true, className: 'reorder', targets: 0 },
        { orderable: false, targets: '_all' }
     ],rowReorder: true,
     "fnDrawCallback": function( oSettings ) {
      if(isFirst) isFirst = false;
      else $('#save_status').html('(Not Saved)');
     }
     <?php } ?>
    });

    $('#editModal').on('show.bs.modal', function(e) {
      var position_id = $(e.relatedTarget).data('positionid');
      var position_name = $(e.relatedTarget).data('positionname');
      var max = $(e.relatedTarget).data('max');
      $(e.currentTarget).find('#editModal_positionid').val(position_id);
      $(e.currentTarget).find('#editModal_name').val(position_name);
      $(e.currentTarget).find('#editModal_max').val(max);
    });

    $('#deleteModal').on('show.bs.modal', function(e) {

      var position_id = $(e.relatedTarget).data('positionid');
      var position_name = $(e.relatedTarget).data('positionname');

      var position_ids = [];
      $('.position-id').each(function(){
        if(position_id != $(this).val())
          position_ids.push($(this).val());
      });
      var deleteModal_pids = "";
      for(i=0;i<position_ids.length;i++){
        deleteModal_pids += "<input type='hidden' name='position_ids[]' value="+position_ids[i]+">";
      }

      $(e.currentTarget).find('#deleteModal_pid').val(position_id);
      $(e.currentTarget).find('#deleteModal_pids').html(deleteModal_pids);
      $(e.currentTarget).find('#deleteModal_name').html(position_name);
    });

    $('#reorder_button').on('click', function(e){
      var position_ids = [];
      $('.position-id').each(function(){
        position_ids.push($(this).val());
      });

      jQuery.ajax({
        type: "POST",
        url: "<?php echo linkbase_url(); ?>" + "votingofficial/positions/reorder",
        data: {position_ids: position_ids},
        success: function(res) {
          if (res == 'true')
          {
            $('#save_status').html('(Saved)');
            $('#success_alert').modal('show');
          }else{
            $('#error_alert').modal('show');
          }
        }
      });

    });

  });

</script>
```

### F. positions.php

```
<div class="content container-fluid text-center">

  <h2>List of Positions</h2>
  <hr>
  <div class='form-horizontal'>
    <div class='form-group'>
      <label class='control-label input-lg col-sm-4'> Select an election: </label>
      <div class="col-sm-6">
        <select name='election_name' id='select_election_id' class='form-control input-lg'>
        <?php if(!empty($election_id)){ ?>
            <option value=<?php echo $election_id; ?> default> <?php echo $election_name; ?> </option>
        <?php } else { ?>
          <option id='option_placeholder' default> None </option>
        <?php } foreach($elections as $row){
          if($row['election_id']!= $election_id) {?>
            <option value=<?php echo $row['election_id'];?>><?php echo $row['name'];?></option>
        <?php }
          } ?>
      </select>
      </div>
    </div>
  </div>

  <hr>

  <div id='positions_table'>

  </div>

</div>
```

### G. table.php

```
<?php if($election_status != 0 && $election_status != 1){ ?>
<div class='container-fluid' style='padding-bottom: 20px;'>
  <button class="btn btn-success btn-md pull-left" data-toggle="modal" data-target="#addModal">
  <span class="glyphicon glyphicon-plus"></span> Add a Position</button>
</div>
<?php } ?>

<?php if(!empty($error)){ ?>
  <div class='alert alert-danger fade in'>
      <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a>
      <?php echo $error;?>
  </div>
<?php } ?>

<?php if(!empty($positions)){ ?>

  <?php if($election_status != 0 && $election_status != 1){ ?>
  <div class='alert alert-info fade in'>
    <strong>Info:</strong> Reorder by dragging the order column. <br>
      Click <span class="glyphicon glyphicon-pencil"></span> to edit and
      <span class="glyphicon glyphicon-trash
```

```
         "></span> to delete the row.
  </div>
  <?php } ?>

  <div class='text-left'>
  <table id='elections_table' class="table
       table-hover text-left" style="width
       :100%">
    <thead >
      <tr>
        <th>Order in Ballot <span id='
             save_status'>(Saved)</span></th
             >
        <th>Name</th>
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
        <th>Action</th>
        <?php } ?>
      </tr>
    </thead>
    <tbody>
    <?php
    foreach($positions as $row){ ?>
      <tr class='table-row'>
        <td><?php echo $row['ordinality
             '];?></td>
        <input type='hidden' class='position
             -id' name='position_id[]' value
             ="<?php echo $row['position_id
             '];?>" >
        <td><?php echo $row['position_name
             '];?></td>
        <?php if($election_status != 0 &&
             $election_status != 1){ ?>
        <td>
          <button class="btn btn-success btn
               -sm" data-toggle="modal" data
               -positionid=<?php echo $row['
               position_id'];?>
          data-positionname="<?php echo $row
               ['position_name'];?>" data-
               max=<?php echo $row['max'];?>
          data-target="#editModal">
            <span class="glyphicon glyphicon
                 -pencil"></span></button>
          <button class="btn btn-danger btn-
               sm" data-toggle="modal" data-
               positionid=<?php echo $row['
               position_id'];?>
          data-positionname="<?php echo $row
               ['position_name'];?>" data-
               target="#deleteModal">
            <span class="glyphicon glyphicon
                 -trash"></span></button>
        </td>
        <?php } ?>
      </tr>
    <?php }?>
    </tbody>
  </table>

  <div class="text-center">
    <button id='reorder_button' class="btn
         btn-primary btn-md">
      <span class="glyphicon glyphicon-
           pushpin"></span> Save Order
    </button>
  </div>
```

```
    </div>

    <?php }else{ ?>
      <div class='alert alert-danger'><strong>
           Error!</strong> There are no
           positions in this election.</div>
    <?php } ?>

<!-- Success Modal -->
<div id="success_alert" class="modal fade"
     role="dialog">
  <div class="modal-dialog">

    <!-- Modal content -->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class='
             glyphicon glyphicon-ok-sign'></
             span> Success</h4>
      </div>
      <div class="modal-body">
        <p>Positions have been successfully
             reordered.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-
             default" data-dismiss="modal">
             Close</button>
      </div>
    </div>

  </div>
</div>

<!-- Error Modal -->
<div id="error_alert" class="modal fade" role
     ="dialog">
  <div class="modal-dialog">

    <!-- Modal content -->
    <div class="modal-content">
      <div class="modal-header modal-header-
           red">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class='
             glyphicon glyphicon-remove-sign
             '></span> Error</h4>
      </div>
      <div class="modal-body">
        <p>Error! Positions have been not
             successfully reordered.</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-
             default" data-dismiss="modal">
             Close</button>
      </div>
    </div>

  </div>
</div>
```

vi. profile

A. change_password_modal.php

```
<!-- Delete Modal -->
<div id="changePasswordModal" class="modal
     fade" role="dialog">
  <div class="modal-dialog">

    <!-- Delete Modal content-->
    <?php echo form_open('votingofficial/
         profile/delete'); ?>
    <div class="modal-content">

      <div class="modal-header modal-header-
           red">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class='
             glyphicon glyphicon-lock'></span>
              Change Password</h4>
      </div>

      <div class="modal-body">
        <div class="modal-form">

            <div class='form-group'>
              <label>Current Password: </label
```

```
                   >
              <input type="password" name="
                   current_password" class="
                   form-control" required>
            </div>

            <div class='form-group'>
              <label>New Password: </label>
              <input type="password" name="
                   new_password" class="form-
                   control" required>
            </div>

            <div class='form-group'>
              <label>Verify New Password: </
                   label>
              <input type="password" name="
                   verify_password" class="
                   form-control" required>
            </div>

        </div>
      </div>

      <div class="modal-footer">
```

```
        <button type="submit" class="btn btn-
            success pull-left">Change</button
            >
        <button type="submit" class="btn btn-
            danger pull-right" data-dismiss="
            modal">Cancel</button>
        </form>
```

```
      </div>
    </div>
  </div>
</div>
```

B. edit_profile_modal.php

```
<!-- Edit Modal -->
<div id="editProfileModal" class="modal fade"
    role="dialog">
  <div class="modal-dialog">

    <!-- Modal content-->
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
            data-dismiss="modal">&times;</
            button>
        <h4 class="modal-title"><span class="
            glyphicon glyphicon-pencil"></
            span> Edit Profile</h4>
      </div>

      <div class="modal-body">
        <div class="modal-form">

          <?php echo form_open('votingofficial
              /profile/edit'); ?>

            <div class='form-group'>
              <label for="username">Username
                  :</label>
              <input type="text" name="
                  username" id="
                  editProfileModal_username"
                  class="form-control"
                value="<?php echo $username
                    ;?>" required>
            </div>
```

```
            <div class='form-group'>
              <label for="email">Email:</
                  label>
              <input type="text" name="email"
                  id="editProfileModal_email"
                   class="form-control" value
                  ="<?php echo $email;?>"
                  required>
            </div>

        </div>
      </div>

      <div class="modal-footer">
        <button type="submit" class="btn btn-
            success pull-left"><span class="
            glyphicon glyphicon-pencil"></
            span> Edit</button>
        <button type="submit" class="btn btn-
            danger pull-right" data-dismiss="
            modal"><span class="glyphicon
            glyphicon-remove"></span> Cancel
            </button>
        </form>
      </div>

    </div>
  </div>
</div>
```

C. profile.php

```
<div class="content container-fluid text-
    center">

  <h2>Your Profile</h2>

  <hr>

  <div class='container-fluid' style='padding-
      bottom: 20px;'>

    <?php if(!empty($error)){ ?>
      <div class='alert alert-danger'><strong
          ><?php echo $error;?></strong></div
          >
    <?php } ?>

    <?php if(!empty($success)){ ?>
      <div class='alert alert-success'><strong
          ><?php echo $success;?></strong></
          div>
    <?php } ?>

    <table id='profile_table' class="table
        table-striped text-center" style='
        width:100%;'>
      <tr>
        <td>
          <strong>Username: <?php echo
              $username;?></strong>
        </td>
      </tr>

      <tr>
        <td>
          <strong>Email: <?php echo $email
```

```
              ;?></strong>
        </td>
      </tr>

      <tr>
        <td>
          <strong>Role: <?php echo $role;?></
              strong>
        </td>
      </tr>
    </table>

    <div class='text-center'>

      <button class="btn btn-success btn-md"
          data-toggle="modal" data-target="#
          editProfileModal">
        <span class="glyphicon glyphicon-
            pencil"></span> Edit Profile
      </button>

      <button class="btn btn-danger btn-md"
          data-toggle="modal" data-target="#
          changePasswordModal">
        <span class="glyphicon glyphicon-lock
            "></span> Change Password
      </button>

    </div>

  </div>

</div>
```

vii. voters

A. add_modal.php

```
<!-- Add Modal -->
<div id="addModal" class="modal fade" role="
    dialog">
  <div class="modal-dialog">

    <!-- Add Modal content-->
    <?php echo form_open('votingofficial/
        voters/add'); ?>
    <div class="modal-content">
```

```
      <div class="modal-header">
        <button type="button" class="close"
            data-dismiss="modal">&times;</
            button>
        <h4 class="modal-title"><span class="
            glyphicon glyphicon-plus"></span>
             Add a Voter</h4>
      </div>

      <div class="modal-body">
```

```
        <div class=”modal-form”>

            <div class='form-group'>
              <label>Username: </label>
              <input type='text' name='
                  username' required>
            </div>

            <div class='form-group'>
              <label>Email: </label>
              <input type='text' name='email'
                  required>
            </div>

            <div class='form-group'>
              <label>Voter Type: </label>
                <select name='voter_type_id'
                    class='form-control input
                    -lg modal-form-control'
                    required>
                <option disabled selected
                    value> -- select an
                    option -- </option>
                <?php foreach($votertypes as
                    $row){ ?>
                  <option value=<?php echo
                      $row['voter_type_id
                      '];?>><?php echo $row['
                      name'];?></option>
                <?php } ?>
                </select>
            </div>

          </div>
        </div>

        <div class=”modal-footer”>

          <button type=”submit” class=”btn btn-
              success pull-left”><span class=”
              glyphicon glyphicon-plus”></span>
               Add Voter</button>
          <button type=”submit” class=”btn btn-
              danger pull-right” data-dismiss=”
              modal”><span class=”glyphicon
              glyphicon-remove”></span> Cancel
              </button>

          </form>
        </div>

      </div>
    </div>
  </div>
```

B. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open('votingofficial/
        voters/delete'); ?>
    <div class=”modal-content”>
      <div class=”modal-header modal-header-
          red”>
        <button type=”button” class=”close”
            data-dismiss=”modal”>&times;</
            button>
        <h4 class=”modal-title”>Are you sure
            you want to delete this voter?</
            h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

            <input type='hidden' name='user_id
                []' id='deleteModal_userid' >

            <div class='form-group'>
              <label>Username: </label>
              <span id='deleteModal_username
                  '></span>
            </div>

            <div class='form-group'>
              <label>Email: </label>
              <span id='deleteModal_email'></
                  span>
            </div>

            <div class='form-group'>
              <label>Voter Type: </label>
                <span id='
                    deleteModal_votertypename
                    '></span>
            </div>

          </div>
        </div>

        <div class=”modal-footer”>
          <button type=”submit” class=”btn btn-
              success pull-left”>Yes</button>
          <button type=”submit” class=”btn btn-
              danger pull-right” data-dismiss=”
              modal”>No</button>
          </form>
        </div>
      </div>

    </div>
  </div>
```

C. delete_selected_modal.php

```
<!-- Delete Modal -->
<div id=”deleteSelectedModal” class=”modal
    fade” role=”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content-->
    <?php echo form_open('votingofficial/
        voters/delete'); ?>
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
          red”>
        <button type=”button” class=”close”
            data-dismiss=”modal”>&times;</
            button>
        <h4 class=”modal-title”>Are you sure
            you want to delete these voters
            ?</h4>
      </div>

      <div class=”modal-body”>

          <div class=”modal-form”>

          </div>
        </div>

        <div class=”modal-footer”>
          <button type=”submit” class=”btn btn-
              success pull-left” id='
              delete_selected_button'>Yes</
              button>
          <button type=”submit” class=”btn btn-
              danger pull-right” data-dismiss=”
              modal”>No</button>
          </form>
        </div>
      </div>

    </div>
  </div>
```

D. edit_modal.php

```
<!-- Edit Modal -->
<div id=”editModal” class=”modal fade” role=”
    dialog”>
  <div class=”modal-dialog”>

    <!-- Edit Modal content-->
      <?php echo form_open('votingofficial/
          voters/edit'); ?>
      <div class=”modal-content”>

        <div class=”modal-header”>
          <button type=”button” class=”close”
```

```
           data-dismiss=”modal”>&times;</
           button>
      <h4 class=”modal-title”><span class=”
           glyphicon glyphicon-pencil”></
           span> Edit Voter</h4>
    </div>

    <div class=”modal-body”>
      <div class=”modal-form”>

          <input type='hidden' name='user_id
               ' id='editModal_userid' >

          <div class='form-group'>
            <label>Username: </label>
            <input type='text' name='
                 username' id='
                 editModal_username'
                 required>
          </div>

          <div class='form-group'>
            <label>Email: </label>
            <input type='text' name='email'
                 id='editModal_email'
                 required>
          </div>

          <div class='form-group'>
            <label>Voter Type: </label>
              <select name='voter_type_id'
                   id='editModal_votertypeid
                   ' class='form-control
                   input-lg modal-form-
                     control' required>
                <option disabled selected
                     value> -- select an
                     option -- </option>
                <?php foreach($votertypes as
                     $row){ ?>
                  <option value=<?php echo
                       $row['voter_type_id
                       '];?>><?php echo $row['
                       name'];?></option>
                <?php } ?>
                </select>
            </div>

        </div>
      </div>

      <div class=”modal-footer”>
        <button type=”submit” class=”btn btn-
             success pull-left”><span class=”
             glyphicon glyphicon-pencil”></
             span> Edit Voter</button>
        <button type=”submit” class=”btn btn-
             danger pull-right” data-dismiss=”
             modal”><span class=”glyphicon
             glyphicon-remove”></span> Cancel
             </button>
        </form>
      </div>
    </div>

  </div>
</div>
```

E. footer.php

```
    <!-- <div id=”footer”>Copyright &copy;
         2016 Cybernetica AS</div> -->
<footer class=”container-fluid text-left
     navbar-default”>
  <p>&copy; University of the Philippines
       Manila</p>
</footer>
<script src=”<?php echo asset_url();?>js/
     jquery-1.12.0.min.js”></script>
<script src=”<?php echo asset_url();?>js/
     bootstrap.js”></script>
<script src=”<?php echo asset_url();?>js/
     jquery.dataTables.min.js”></script>
<script src=”<?php echo asset_url();?>js/
     dataTables.bootstrap.min.js”></script>
<script type=”text/javascript”>

  $(document).ready(function () {
    $('.content').hide();
    $('.content').delay(200).slideDown(800);
    $(”#voters_table”).dataTable({
      ”aoColumnDefs”: [
        { 'bSortable': false, 'aTargets':
             [4,5] }
      ]
    });

    $('#editModal').on('show.bs.modal',
         function(e) {

      var userid = $(e.relatedTarget).data('
           userid');
      var username = $(e.relatedTarget).data
           ('username');
      var email = $(e.relatedTarget).data('
           email');
      var votertypeid = $(e.relatedTarget).
           data('votertypeid');

      $(e.currentTarget).find('#
           editModal_userid').val(userid);
      $(e.currentTarget).find('#
           editModal_username').val(username
           );
      $(e.currentTarget).find('#
           editModal_email').val(email);
      if(votertypeid!=”null”) $(e.
           currentTarget).find('#
           editModal_votertypeid').val(
           votertypeid);

    });

    $('#deleteModal').on('show.bs.modal',
         function(e) {

      var userid = $(e.relatedTarget).data('
           userid');
      var username = $(e.relatedTarget).data
           ('username');
      var email = $(e.relatedTarget).data('
           email');
      var votertypename = $(e.relatedTarget)
           .data('votertypename');

      $(e.currentTarget).find('#
           deleteModal_userid').val(userid);
      $(e.currentTarget).find('#
           deleteModal_username').html(
           username);
      $(e.currentTarget).find('#
           deleteModal_email').html(email);
      $(e.currentTarget).find('#
           deleteModal_votertypename').html(
           votertypename);
    });

    $('#allvoters').on('click',function(e){
      $(”.checkable”).prop('checked',$(this)
           .prop('checked'));
    });

    $('#deleteSelectedModal').on('show.bs.
         modal', function(e){

      var html = ””;
      var isThereChecked = false;
      $(”.checkable”).each(function(e){
        if($(this).attr('name')!='allvoters'
             && $(this).prop('checked') ==
             true){
          isThereChecked = true;
          html += ”<div class='form-group'>
               <label>Username: ”+$(this).
               attr('username')+”</label></
               div>”;
          html += ”<input type='hidden' name
               ='user_id[]' value='”+$(this)
               .val()+”'>”;
        }
      });

      if(isThereChecked == true){
        $(e.currentTarget).find('#
             delete_selected_button').show()
             ;
        $(e.currentTarget).find('.modal-form
             ').html(html);
      }else{
        $(e.currentTarget).find('#
             delete_selected_button').hide()
             ;
        $(e.currentTarget).find('.modal-form
             ').html(”There are no selected
             voters.”);
      }
    });

  });
  </script>
</body>
</html>
```

F. import_modal.php

```
<!-- Import Modal -->
<div id="importModal" class="modal fade" role
    ="dialog">
  <div class="modal-dialog">

    <!-- Import Modal content-->
    <?php echo form_open_multipart('
        votingofficial/voters/import'); ?>
    <div class="modal-content">

      <div class="modal-header">
        <button type="button" class="close"
            data-dismiss="modal">&times;</
            button>
        <h4 class="modal-title"><span class="
            glyphicon glyphicon-upload"></
            span> Import Voter's list</h4>
      </div>

      <div class="modal-body">

        <div class="modal-form">

          <div class='form-group'>

            <div class='alert alert-info fade
                in text-center'>
              <strong>Format of CSV:</strong>
                  username,email <br> (eg.
                  voter,voter@email.com)
            </div>

            <input type="file" name="userfile"
                class='file btn btn-default
                '>

          </div>

          <div class='form-group'>

            <label>Voter Type: </label>
            <select name='voter_type_id' class
                ='form-control input-lg modal
                -form-control' required>
              <option disabled selected value>
                  -- select an option -- </
                  option>
              <?php foreach($votertypes as
                  $row){ ?>
                <option value=<?php echo $row
                    ['voter_type_id'];?>><?
                    php echo $row['name
                    '];?></option>
              <?php
              } ?>
            </select>

          </div>

        </div>
      </div>

      <div class="modal-footer">

        <button type="submit" class="btn btn-
            success pull-left"><span class="
            glyphicon glyphicon-upload"></
            span> Import CSV</button>
        <button type="submit" class="btn btn-
            danger pull-right" data-dismiss="
            modal"><span class="glyphicon
            glyphicon-remove"></span> Cancel
            </button>

        </form>
      </div>

    </div>
  </div>
</div>
```

G. voters.php

```
<div class="content container-fluid text-
    center">

  <h2>List of Voters</h2>
  <hr>

  <div class='container-fluid' style='padding-
      bottom: 20px;'>
    <div class='row'>

      <button class="btn btn-success btn-md
          pull-left col-sm-2" data-toggle="
          modal" data-target="#addModal">
        <span class="glyphicon glyphicon-plus
            "></span> Add Voter
      </button>

      <button class="btn btn-success btn-md
          col-sm-2" style='margin-left: 10px
          ;' data-toggle="modal" data-target
          ="#importModal">
        <span class="glyphicon glyphicon-
            upload"></span> Import CSV
      </button>

      <button class='btn btn-danger btn-md
          pull-right' data-toggle="modal"
          data-target="#deleteSelectedModal">
      <span class="glyphicon glyphicon-trash
          "></span> Delete selected</button>

    </div>
  </div>

  <?php if(!empty($error)){ ?>
    <div class='alert alert-danger fade in'>
        <a href="#" class="close" data-dismiss
            ="alert" aria-label="close">&
            times;</a>
        <?php echo $error;?>
    </div>
  <?php } ?>

  <?php if(!empty($voters)){ ?>

    <div class='alert alert-info fade in'>
      <strong>Info:</strong> Click <span class
          ="glyphicon glyphicon-pencil"></
          span> to edit and
      <span class="glyphicon glyphicon-trash
          "></span> to delete the row.
    </div>

    <div class='text-left'>
    <table id='voters_table' class="table
        table-hover text-left" style='width
        :100%'>
      <thead >
        <tr>
          <th>Username</th>
          <th>Email</th>
          <th>Voter Type</th>
          <th>Voted</th>
          <th>Action</th>
          <th>
            <label><input type="checkbox"
                class='checkable form-control
                ' name="allvoters" id="
                allvoters"></label>
          </th>
        </tr>
      </thead>
      <tbody>
      <?php foreach($voters as $row){ ?>
        <tr class='table-row'>
          <td><?php echo $row['username'];?></
              td>
          <td><?php echo $row['email'];?></td>
          <td><?php echo $row['name'];?></td>
          <td><?php if($row['voted']==false)
              echo "No"; else echo "Yes";?></
              td>
          <td>
          <?php if($row['active'] == false){
              ?>
            <button class="btn btn-success btn
                -sm" data-toggle="modal" data
                -userid=<?php echo $row['
                user_id'];?>
              data-username="<?php echo $row['
                  username'];?>" data-email
                  ="<?php echo $row['email
                  '];?>"
              data-votertypeid=<?php if(!empty
        ($row['voter_type_id']))
        echo $row['voter_type_id'];
         else echo ”null”;?>
    data-target=”#editModal”>
    <span class=”glyphicon glyphicon
        -pencil”></span></button>
  <button class=”btn btn-danger btn-
      sm” data-toggle=”modal” data-
       userid=<?php echo $row['
       user_id'];?>
    data-username=”<?php echo $row['
        username'];?>” data-email
        =”<?php echo $row['email
        '];?>”
    data-votertypename=”<?php if(!
        empty($row['name'])) echo
        $row['name'];?>”” data-
        target=”#deleteModal”>
    <span class=”glyphicon glyphicon
        -trash”></span></button>
<?php } else echo ”Not allowed”;?>
</td>
<td>
<?php if($row['active'] == false){
    ?>
  <input type=”checkbox” class='
       checkable form-control'
                username=”<?php echo $row['
                username'];?>”
            votertypename=”<?php if(!empty(
                $row['name'])) echo $row['
                name'];?>” email=”<?php echo
                $row['email'];?>”
            value=<?php echo $row['user_id
                '];?>>
          <?php } ?>
          </td>
        </tr>
      <?php }?>
      </tbody>
    </table>
    </div>

    <?php }else{ ?>

      <div class='alert alert-danger'><strong>
          Error!</strong> There are no voters
          .</div>

    <?php } ?>

</div>
```

viii. votertypes

A. add_modal.php

```
<!-- Add Modal -->
<div id=”addModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal-dialog”>
    <!-- Add Modal content -->
    <?php
    echo form_open('votingofficial/votertypes/
         add'); ?>
    <div class=”modal-content”>

      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-plus”></span>
              Add a Voter Type</h4>
      </div>

      <div class=”modal-body”>
        <div class=”modal-form”>

          <div class='form-group'>
            <label>Voter Type Name: </label>
            <input type='text' name='name'
                 required>
          </div>

          <div class='form-group alert alert-
               info text-center check-one-info
               '>
            Please check at least one.
          </div>

          <div class='form-group text-center'>
            <h4><label>Election:</label></h4>
              <?php
              $isFirst = true;
              foreach($elections as $row){

                if($isFirst || $prevRow['
                    parent_id'] != $row['
                    parent_id']){
                  if($isFirst) $isFirst =
                      false;
                  else echo ”</div>”;?>
                  <div id=”<?php echo $row['
                       parent_id'].”_children
                       ”?>” class='children'>
                <?php }
                $prevRow = $row;
                ?>

                <label>
                  <input type=”checkbox” class
                      ='check-box checkable'
                      name=”election[]” value
                      =<?php echo $row['
                      election_id'];?>
                    parent_id=”<?php echo $row
                        ['parent_id'];?>” id
                        =”<?php echo $row['
                        election_id'];?>”>
                <?php echo $row['name'];?> </
                     label>

              <?php } ?>
              </div> <!-- For the last div -->
          </div>

        </div>
      </div>


      <div class=”modal-footer”>
        <button class=”btn btn-success pull-
             left” id=”add_votetype_button”>
        <span class=”glyphicon glyphicon-plus
            ”></span> Add Voter Type</button>
        <button class=”btn btn-danger pull-
             right” data-dismiss=”modal”><span
              class=”glyphicon glyphicon-
             remove”></span> Cancel</button>
      </div>
      </form>

    </div>
  </div>
</div>
```

B. delete_modal.php

```
<!-- Delete Modal -->
<div id=”deleteModal” class=”modal fade” role
     =”dialog”>
  <div class=”modal-dialog”>

    <!-- Delete Modal content -->
    <?php echo form_open('votingofficial/
         votertypes/delete'); ?>
    <div class=”modal-content”>
      <div class=”modal-header modal-header-
           red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”>Are you sure
             you want to delete this voter
             type?</h4>
      </div>
      <div class=”modal-body”>
        <div class=”modal-form”>

            <input type='hidden' name='
                 voter_type_id' id='
                 deleteModal_votertypeid' >

            <div class='form-group'>
              <label>Type Name: </label>
              <span id='
                   deleteModal_votertypename
                   '></span>
            </div>
            <div class='form-group text-center
                 '>
              <h4><label>Election:</label></h4
                   >
              <?php
              $isFirst = true;
              foreach($elections as $row){

                if($isFirst || $prevRow['
                     parent_id'] != $row['
                     parent_id']){
                  if($isFirst) $isFirst =
                       false;
                  else echo "</div>";?>
                  <div id="<?php echo $row['
                       parent_id']."
                       _children_delete"?>"
                       class='children'>
                <?php }
                $prevRow = $row;
                ?>

                <label>
                  <input type="checkbox" class
                       ='check-box checkable'
                       name="election[]" value
                       =<?php echo $row['
                       election_id'];?>
                    parent_id="<?php echo $row
                         ['parent_id'];?>" id
                         ="<?php echo $row['
                         election_id'];?>">
                <?php echo $row['name'];?> </
                     label>

              <?php } ?>
              </div> <!-- For the last div -->
            </div>

        </div>
      </div>

      <div class="modal-footer">
        <button type="submit" class="btn btn-
             success pull-left">Yes</button>
        <button type="submit" class="btn btn-
             danger pull-right" data-dismiss="
             modal">No</button>
        </form>
      </div>
    </div>

  </div>
</div>
```

C. edit_modal.php

```
<!-- Edit Modal -->
<div id="editModal" class="modal fade" role="
     dialog">
  <div class="modal-dialog">

    <!-- Edit Modal content-->
    <?php echo form_open('votingofficial/
         votertypes/edit'); ?>
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close"
             data-dismiss="modal">&times;</
             button>
        <h4 class="modal-title"><span class="
             glyphicon glyphicon-pencil"></
             span> Edit Voter Type</h4>
      </div>
      <div class="modal-body">
        <div class="modal-form">

            <input type='hidden' name='
                 voter_type_id' id='
                 editModal_votertypeid'>

            <div class='form-group'>
              <label>Voter Type Name: </label>
              <input type='text' name='name'
                   id='editModal_votertypename
                   ' required>
            </div>

            <div class='form-group alert alert
                 -info text-center check-one-
                 info'>
              Please check at least one.
            </div>

            <div class='form-group text-center
                 '>
              <h4><label>Election:</label></h4
                   >
              <?php
              $isFirst = true;
              foreach($elections as $row){

                if($isFirst || $prevRow['
                     parent_id'] != $row['
                     parent_id']){
                  if($isFirst) $isFirst =
                       false;
                  else echo "</div>";?>
                  <div id="<?php echo $row['
                       parent_id']."
                       _children_edit"?>"
                       class='children'>
                <?php }
                $prevRow = $row;
                ?>

                <label>
                  <input type="checkbox" class
                       ='check-box checkable'
                       name="election[]" value
                       =<?php echo $row['
                       election_id'];?>
                    parent_id="<?php echo $row
                         ['parent_id'];?>" id
                         ="<?php echo $row['
                         election_id'];?>">
                <?php echo $row['name'];?> </
                     label>

              <?php } ?>
              </div> <!-- For the last div -->
            </div>
        </div>
      </div>

      <div class="modal-footer">
        <button class="btn btn-success pull-
             left" id="edit_votetype_button">
        <span class="glyphicon glyphicon-
             pencil"></span> Edit Candidate</
             button>
        <button class="btn btn-danger pull-
             right" data-dismiss="modal"><span
              class="glyphicon glyphicon-
             remove"></span> Cancel</button>
      </div>
      </form>

    </div>
  </div>
</div>
```

D. footer.php

```
      <!-- <div id="footer">Copyright &copy;
           2016 Cybernetica AS</div> -->
  <footer class="container-fluid text-left
       navbar-default">
    <p>&copy; University of the Philippines
         Manila</p>
  </footer>
  <script src="<?php echo asset_url();?>js/
       jquery-1.12.0.min.js"></script>
  <script src="<?php echo asset_url();?>js/
       bootstrap.js"></script>
  <script src="<?php echo asset_url();?>js/
       jquery.dataTables.min.js"></script>
  <script src="<?php echo asset_url();?>js/
       dataTables.bootstrap.min.js"></script>
  <script type="text/javascript">

    $(document).ready(function () {

      // for add
      $('.children').hide();
      $('#_children').show(); //elections with
           no parents

      $('.content').hide();
      $('.content').delay(200).slideDown(800);
      $("#elections_table").dataTable({
        "aoColumnDefs": [
    { 'bSortable': false, 'aTargets':
        [1] }
 ]
});

$('#editModal').on('show.bs.modal',
    function(e) {

  var votertypeid = $(e.relatedTarget).
      data('votertypeid');
  var votertypename = $(e.relatedTarget)
      .data('votertypename');
  var election_ids = $(e.relatedTarget).
      data('electionids');
  var election_array = [];
  var intiail_election_id,
      initial_parent_id,
      initial_grandparent_id;
  if(election_ids.toString().indexOf
      (',')>=0){
    election_array = election_ids.split
        (",");
  }else{
    election_array[0] = election_ids.
        toString();
  }

  $(e.currentTarget).find('#
      editModal_votertypeid').val(
      votertypeid);
  $(e.currentTarget).find('#
      editModal_votertypename').val(
      votertypename);
  $(e.currentTarget).find("input[type='
      checkbox']").each(function(){
    $(this).prop('checked', false);
    if($.inArray($(this).val().toString
        (),election_array) >= 0){
      $(this).prop('checked', true);
      initial_election_id = $(this).val
          ();
      $('#'+initial_election_id+'
          _children_edit').show();

      initial_parent_id = $(this).attr('
          parent_id');
      $('#'+initial_parent_id+'
          _children_edit').show();

      initial_grandparent_id = $("#"+
          initial_parent_id).attr('
          parent_id');
      $("#"+initial_grandparent_id+"
          _children_edit :input").attr
          ("readonly", true);
      $("#"+initial_grandparent_id+"
          _children_edit").css("pointer
          -events","none");
    }
  });

  $(e.currentTarget).find("input[type='
      checkbox']").change(function(){

    var election_id = $(this).val();
    var parent_id = $(this).attr('
        parent_id');
    var grandparent_id = $("#"+parent_id
        ).attr('parent_id');
    var isThereCheck = false;

    if($(this).prop('checked') == true){

      $("#"+grandparent_id+"
          _children_edit :input").attr
          ("readonly", true);
      $("#"+grandparent_id+"
          _children_edit").css("pointer
          -events","none");
      $('#'+election_id+'_children_edit
          ').slideDown();

    }else{ // unchecked

      $("#"+parent_id+"_children_edit :
          input").each(function(){
        if($(this).prop('checked'))
            isThereCheck = true;
      });
      if(!isThereCheck) {
        $("#"+grandparent_id+"
            _children_edit :input").
            attr("readonly", false);
        $("#"+grandparent_id+"
            _children_edit").css("
            pointer-events","auto");
      }
      $('#'+election_id+'_children_edit
          ').slideUp();
```

```
    }
  });

});

$('#deleteModal').on('show.bs.modal',
    function(e) {
  var votertypeid = $(e.relatedTarget).
      data('votertypeid');
  var votertypename = $(e.relatedTarget)
      .data('votertypename');
  var election_ids = $(e.relatedTarget).
      data('electionids');
  var election_array = [];
  var initial_parent_id;
  if(election_ids.toString().indexOf
      (',')>=0){
    election_array = election_ids.split
        (",");
  }else{
    election_array[0] = election_ids.
        toString();
  }

  $(e.currentTarget).find('#
      deleteModal_votertypeid').val(
      votertypeid);
  $(e.currentTarget).find('#
      deleteModal_votertypename').html(
      votertypename);
  $(e.currentTarget).find("input[type='
      checkbox']").each(function (){
    $(this).prop('checked', false);
    if($.inArray($(this).val().toString
        (),election_array) >= 0){
      $(this).prop('checked', true);
      initial_parent_id = $(this).attr('
          parent_id');
      $('#'+initial_parent_id+'
          _children_delete').show();
    }
  });
  $(e.currentTarget).find("input[type='
      checkbox']").attr("disabled",
      true);
});

$('form').submit(function(e){
  var checkedAtLeastOne = false;
  $(e.currentTarget).find('input[type="
      checkbox"]').each(function() {
      if ($(this).is(":checked")) {
          checkedAtLeastOne = true;
      }
  });

  if(checkedAtLeastOne == true){
    return true;
  }else{
    $(e.currentTarget).find('.check-one-
        info').removeClass('alert-info
        ').addClass('alert-danger');
    e.preventDefault();
  }
});

$('#addModal').on('show.bs.modal',
    function(e) {

  $(e.currentTarget).find("input[type='
      checkbox']").change(function(){

    var election_id = $(this).val();
    var parent_id = $(this).attr('
        parent_id');
    var grandparent_id = $("#"+parent_id
        ).attr('parent_id');
    var isThereCheck = false;

    if($(this).prop('checked') == true){

      $("#"+grandparent_id+"_children :
          input").attr("readonly", true
          );
      $("#"+grandparent_id+"_children").
          css("pointer-events","none");
      $('#'+election_id+'_children').
          slideDown();

    }else{ // unchecked

      $("#"+parent_id+"_children :input
          ").each(function(){
        if($(this).prop('checked'))
            isThereCheck = true;
      });
      if(!isThereCheck) {
        $("#"+grandparent_id+"_children
            :input").attr("readonly",
                false);
            $("#"+grandparent_id+"_children
                ").css("pointer-events","
                auto");
          }
          $('#'+election_id+'_children').
              slideUp();
        }
      });
```

```
        });

      });
    </script>
  </body>
</html>
```

E. votertypes.php

```
<div class="content container-fluid text-
    center">

  <h2>List of Voter Types</h2>
  <hr>
  <div class='container-fluid' style='padding-
      bottom: 20px;'>
    <button class="btn btn-success btn-md pull
        -left" data-toggle="modal" data-
        target="#addModal">
    <span class="glyphicon glyphicon-plus"></
        span> Add a Voter Type</button>
  </div>
  <?php if(!empty($error)){ ?>
    <div class='alert alert-danger fade in'>
        <a href="#" class="close" data-dismiss
            ="alert" aria-label="close">&
            times;</a>
        <?php echo $error;?>
    </div>
  <?php } ?>
  <?php if(!empty($votertypes)){ ?>
    <div class='alert alert-info fade in'>
      <!-- <a href="#" class="close" data-
          dismiss="alert" aria-label="close
          ">&times;</a> -->
      <strong>Info:</strong> Click <span class
          ="glyphicon glyphicon-pencil"></
          span> to edit and
        <span class="glyphicon glyphicon-trash
            "></span> to delete the row.
    </div>

    <div class='text-left'>
    <table id='elections_table' class="table
        table-hover text-left" style='width
        :100%'>
      <thead >
        <tr>
          <th>Type Name</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
      <?php foreach($votertypes as $row){ ?>
        <tr class='table-row'>
          <td><?php echo $row['name'];?></td>
          <td>
          <?php if($row['active']==false){ ?>
            <button class="btn btn-success btn
                -sm" data-toggle="modal" data
                -votertypeid=<?php echo $row
                ['voter_type_id'];?>
              data-votertypename="<?php echo
                  $row['name'];?>"
              data-electionids="<?php for($i
                  =0; $i<count($row['
                  election_ids']); $i++){
                  if($i==0) echo $row['
                      election_ids'][$i];
                  else echo ','.$row['
                      election_ids'][$i];
                }?>"
              data-target="#editModal">
              <span class="glyphicon glyphicon
                  -pencil"></span></button>

            <button class="btn btn-danger btn-
                sm" data-toggle="modal" data-
                votertypeid=<?php echo $row['
                voter_type_id'];?>
              data-votertypename="<?php echo
                  $row['name'];?>"
              data-electionids="<?php for($i
                  =0; $i<count($row['
                  election_ids']); $i++){
                  if($i==0) echo $row['
                      election_ids'][$i];
                  else echo ",".$row['
                      election_ids'][$i];
                }?>"
              data-target="#deleteModal">
              <span class="glyphicon glyphicon
                  -trash"></span></button>
              <?php }else{ echo "Currently
                  active.";} ?>
          </td>
        </tr>
      <?php }?>
      </tbody>
    </table>
    </div>

    <?php }else{ ?>
      <div class='alert alert-danger'><strong>
          Error!</strong> There are no voter
          types.</div>
    <?php } ?>
</div>
```

ix. navbar.php

```
<nav class="navbar navbar-default">
<img src="<?php echo asset_url();?>img/
    upm_logo.png" alt="UPM Logo" align='
    left'>
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">
          Votemind</a>
    </div>
    <div class="collapse navbar-collapse" id
        ="myNavbar">
      <ul class="nav navbar-nav">
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/elections">
              Elections</a>
        </div></li>
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/positions">
              Positions</a>
        </div></li>
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/parties">
              Parties</a>
        </div></li>
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/candidates
              ">Candidates</a>
        </div></li>
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/votertypes
              ">Voter Types</a>
        </div></li>
        <li><div class='a-btn'>
          <a class='btn btn-default' href
              ="<?php echo linkbase_url()
              ;?>votingofficial/voters">
              Voters</a>
        </div></li>
      </ul>
      <ul class="nav navbar-nav navbar-right
          ">
          <li>
          <div class="dropdown">
            <button class="btn btn-warning
                dropdown-toggle" type="
                button" data-toggle="
                dropdown">
              <span class='glyphicon
                  glyphicon-user'></span>
                  <?php echo $username;?>
            <span class="caret"></span></
                button>
        <ul class=”dropdown-menu”>
          <li><a href=”<?php echo
              linkbase_url();?>
              votingofficial/
              notifications”>
            <span class=’glyphicon
                glyphicon-exclamation-
                sign’></span>
            Notifications</a></li>
          <li><a href=”<?php echo
              linkbase_url();?>
              votingofficial/profile”><
              span class=’glyphicon
              glyphicon-cog’></span>
            Profile</a></li>
          <li><a href=”<?php echo
              linkbase_url();?>
              votingofficial/logout”><
              span class=”glyphicon
              glyphicon-log-out”></span
              >
            Logout</a></li>
        </ul>
      </div>
      </li>
    </ul>
  </div>
</div>
</nav>
```

(d) voter

i. footer.php

```
    <!-- <div id=”footer”>Copyright &copy;
        2016 Cybernetica AS</div> -->
<footer class=”container-fluid text-left
    navbar-default”>
  <p>&copy; University of the Philippines
      Manila</p>
</footer>
<script src=”<?php echo asset_url();?>js/
    jquery-1.12.0.min.js”></script>
<script src=”<?php echo asset_url();?>js/
    bootstrap.js”></script>
<script src=”<?php echo asset_url();?>js/
    jquery.validate.min.js”></script>
<script src=”<?php echo asset_url();?>js/
    additional-methods.min.js”></script>
<script src=”<?php echo asset_url();?>crypto
    -js/crc32.js”></script>
<script src=”<?php echo asset_url();?>crypto
    -js/sha1.js”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/bootbox.min.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/ext/typedarray.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/ext/jsaes.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/ext/jsbn.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/ext/jsbn2.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/ext/socket.io.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/prng.js”></script
    >
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/sm.types.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/sm.types.util.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/sm.types.base.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/sm.types.shared3p
    .js”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/serialization.js
    ”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/sm.ctrl.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/miners.js”></
    script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/app.js”></script>
<script type=”text/javascript” src=”<?php
    echo asset_url();?>js/page.js”></script
    >
<script type=”text/javascript”>

  //When page has finished loading...
  $(document).ready(function () {

    $('.content').hide();
    $('.errors').hide();
    $(’#ballot’).delay(200).slideDown(800);

    $(’#verification_form’).validate({
      rules: {
        <?php
          $isFirst = true;
          foreach($elections as $election){
            $candidates = $election[’
                candidates’];
            for($i=0; $i<count($candidates);$i
                ++){
              $row = $candidates[$i];
              $position_id = $row[’position_id
                  ’];

              if($row[’max’]!=1){

                if($i!=0 && $position_id !=
                    $candidates[$i-1][’
                    position_id’]){

                  if($isFirst) $isFirst =
                      false;
                  else echo ”,”;
                  echo $position_id.”:{
                      maxlength:”.$row[’max
                      ’].”}”;

                }else if($i==0){
                  echo $position_id.”:{
                      maxlength:”.$row[’max
                      ’].”}”;
                  $isFirst = false;
                }
              }
            }
          }
        ?>
        },
        errorPlacement: function(error,element
            ) {
          var positionId = element.attr(’name
              ’);
          $(’#’+positionId+’_error’).slideDown
              ();
          return true;
        }
      });

      $(’#submit_vote_button’).on(’click’,
          function(e) {
          if ($(’#verification_form’).valid())
              {

          } else {
              $(’#error_alert’).modal(’show’);
              e.stopPropagation();
          }
      });

      $(”input[type=’radio’]”).on(’change’,
          function(e) {

        var voterId = <?php echo $voter_id;?>;
        var positionId = $(this).attr(”name”);
        var candidateId = $(this).val();
        //var cbchash = CRC32.buf([voterId,
            positionId, candidateId]);
        var cbchash = CRC32.bstr(voterId+””+
            positionId+””+candidateId);
        var sha1hash = CryptoJS.SHA1(””+(
            cbchash>>>0));

        $(’.’+positionId+’_hash’).fadeOut();

        if($(’#’+positionId+””+candidateId+’
            _hash’).html()==””){
          $(’#’+positionId+””+candidateId+’
              _hash’).html(””+sha1hash);
        }
        $(’#’+positionId+””+candidateId+’_hash
        ’).fadeIn();

  $(’#’+positionId+’_error’).slideUp();

});

$(”input[type=’checkbox’]”).on(’change’,
      function(e) {

  var voterId = <?php echo $voter_id;?>;
  var positionId = $(this).attr(”name”);
  var candidateId = $(this).val();
  //var cbchash = CRC32.buf([ voterId,
       positionId, candidateId]);
  var cbchash = CRC32.bstr(voterId+””+
       positionId+””+candidateId);
  var sha1hash = CryptoJS.SHA1(””+(
       cbchash>>>0));

  if($(this).is(’:checked’)){
    if($(’#’+positionId+””+candidateId+’
         _hash’).html()==””){
      $(’#’+positionId+””+candidateId+’
           _hash’).html(””+sha1hash);
    }
    $(’#’+positionId+””+candidateId+’
         _hash’).fadeIn();
  }else{
    $(’#’+positionId+””+candidateId+’
         _hash’).fadeOut();
  }

  if($(this).valid()){
    $(’#’+positionId+’_error’).slideUp()
         ;
  }

});

$(’#voteModal’).on(’show.bs.modal’,
     function(e) {

  var positionId = [];
  var candidateId = [];
  $(”input:checked”).each(function(){
    positionId.push($(this).attr(”name”)
         );
    candidateId.push($(this).val());
  });

  var i;
  var content;
  var name,hash;
  for(i=0;i<positionId.length;i++){
    name = ”Name: ”+$(’#’+positionId[i
         ]+’’+candidateId[i]+’_name’).
         text().trim();
    hash = ”Hash: ”+$(’#’+positionId[i
         ]+’’+candidateId[i]+’_hash’).
         text().trim();
    if(i>0 && positionId[i]==positionId[
         i-1]){
      content += ”<div>”+name+”</div
           >”+”<div>”+hash+”</div>” +
     ”<input class=’downloadable’ type
           =’hidden’ value=’”+hash+”’>”;
    }else{
      content = ”<div>”+name+”</div>”+”<
           div>”+hash+”</div>” +
     ”<input class=’downloadable’ type
           =’hidden’ value=’”+hash+”’>”;
    }

    $(’#’+positionId[i]+’_confirm’).html
         (content);
  }

});

function download(filename, text) {
  var pom = document.createElement(’a’);
  pom.setAttribute(’href’, ’data:text/
       plain;charset=utf-8,’ +
       encodeURIComponent(text));
  pom.setAttribute(’download’, filename)
       ;

  if (document.createEvent) {
      var event = document.createEvent(’
           MouseEvents’);
      event.initEvent(’click’, true,
           true);
      pom.dispatchEvent(event);
  }
  else {
      pom.click();
  }
}
```

```
$(’#downloadButton’).on(’click’,
     function(e) {
  var string = $(this).attr(”name”)+ ”\n
       ”;
  $(”.downloadable”).each(function(){
    if($(this).attr(”name”) == ”
         election_dl”) string += ”\n” +
         $(this).val() + ”\n”;
    else if($(this).attr(”name”) == ”
         position_dl”) string += ”\n” +
         $(this).val();
    else string += $(this).val() + ”\n”;
  });
  download(’myBallot.txt’,string);
});


$(’#submitButton’).on(’click’, function(
     e) {

  $(’body’).css({’cursor’ : ’wait’});
  var voterId = <?php echo $voter_id;?>;
  var electionId = [];
  var candidateId = [];
  var positionId = [];
  $(”input:checked”).each(function(){
    electionId.push($(this).attr(’
         electionId’));
    candidateId.push($(this).val());
    positionId.push($(this).attr(”name”)
         );
  });

  var clbck = function(err_msg, results)
       {
    if (!results) {
      showMessage(’Something went wrong.
            Look for details in the
           console.’, ’error’);
    } else {

      $(’#ballot’).delay(200).slideUp
           (800, function(){

        var i;
        var cbchash, cbchashes=””;
        var codeString = ””;
        var sha1hashes = [];
        for(i=0; i<results[’hashes’].
             length; i++){
          cbchash = results[’hashes’][i
               ];
          cbchashes += ”” + cbchash;
          sha1hashes[i] = CryptoJS.SHA1
               (””+cbchash);
        }
        codeString = CryptoJS.SHA1(
             cbchashes);

        jQuery.ajax({
        type: ”POST”,
        url: ”<?php echo linkbase_url();
             ?>” + ”voter/vote/
             getReceipt”,
        data: {codeString: ””+codeString
             },
        success: function(res) {
            if (res)
            {
              $(’#receipt’).html(res);

              var content;
              var prevhash;
              var isThereDiscrepancy =
                   false;
              for(i=0;i<positionId.
                   length;i++){

                prevhash = $(’#’+
                     positionId[i]+’’+
                     candidateId[i]+’
                     _hash’).html();
                if(prevhash !=
                     sha1hashes[i])
                     isThereDiscrepancy
                     = true;

                if(i>0 && positionId[i
                     ]==positionId[i-1])
                     {
                  content += ”<div>Name
                       : ”+$(’#’+
                       positionId[i
                       ]+’’+candidateId
                       [i]+’_name’).
                       html()+”</div>”
                     +”<div>Hash: ”+
                       sha1hashes[i
                       ]+”</div>”;
  }else{
    content = "<div>Name:
         "+$('#'+
         positionId[i]+''+
         candidateId[i]+'
         _name').html()
         +"</div>"
      +"<div>Hash: "+
           sha1hashes[i
           ]+"</div>";
  }

  $('#'+positionId[i]+'
       _receipt').html(
       content);
}

var status, instructions;
if(isThereDiscrepancy){
  status = "<strong><span
       class='glyphicon
       glyphicon-remove
       '></span> SORRY</
       strong><br> Your
       vote has been sent
       and saved. But we
       have seen that the
       stored vote is not
       the same with the
       vote you sent.";
  $('#status').addClass('
       alert-danger');
  $('#status').html(status
       );
  instructions = "<strong
       ><span class='
       glyphicon glyphicon
       -alert'></span>
       PLEASE READ</strong
       > <br>Below is the
       copy of your ballot
        that was stored in
        our server called
       your <strong>
       receipt</strong>,
       kindly double check
        and <strong> keep
       a copy of the
       receipt </strong>
       for future
       reference. You can
       have a local copy
       of your receipt by
       clicking the <
       strong>'save
       receipt'</strong>
       button or by
       copying the text
       and pasting it
       somewhere else. <br
       > Please <strong>
       notify a voting
       official</strong>
       so that we can be
       do something about
       this error. <br> <
       strong>The buttons
       for these functions
        are found at the
       bottom of this page
       .</strong>";
  $('#instructions').
       addClass('alert-
       warning');
  $('#instructions').html(
       instructions);
}else{
  status = "<strong><span
       class='glyphicon
       glyphicon-ok'></
       span>
       CONGRATULATIONS</
       strong> <br>Your
       vote has been sent
       and saved. We have
       verified that the
       stored vote is the
       same with the vote
       you sent.";
  $('#status').addClass('
       alert-success');
  $('#status').html(status
       );
  instructions = "<strong
       ><span class='
       glyphicon glyphicon
       -alert'></span>
       PLEASE READ</strong
       > <br>Below is the
       copy of your ballot
        that was stored in
        our server called
       your <strong>
       receipt</strong>,
       kindly double check
        and <strong> keep
       a copy of the
       receipt </strong>
       because you will be
        needing this to
       look for your vote
       in the election
       results. You can
       have a local copy
       of your receipt by
       clicking the <
       strong>save receipt
       </strong> button or
        by copying the
       text and pasting it
        somewhere else.
       You can also <
       strong>notify a
       voting official</
       strong> if there
       are discrepancies
       in your receipt.<br
       > <strong>The
       buttons for these
       functions are found
        at the bottom of
       this page.</strong
       >";
  $('#instructions').
       addClass('alert-
       info');
  $('#instructions').html(
       instructions);
}

$('#downloadButton').on('
     click', function(e) {
  var string = $(this).
       attr("name")+ "\n\n
       ";
  string += "My Alias: <?
       php echo $alias;?>
       \n";
  string += "My
       Verification code:
       "+codeString+"\n";
  $(".downloadable").each(
       function(){
    if($(this).attr("name
         ") == "
         election_dl")
         string += "\n" +
         $(this).val() +
         "\n";
    else if($(this).attr("
         name") == "
         position_dl")
         string += "\n" +
         $(this).val();
    else string += $(this)
         .val() + "\n";
  });
  download('myReceipt.txt
       ',string);
});

$('#notify_button').on('
     click', function(e){
  var title = $('#
       notify_title').val
       ();
  var body = $('#
       notify_body').val()
       ;

  jQuery.ajax({
    type: "POST",
    url: "<?php echo
         linkbase_url();
         ?>" + "voter/
         notify",
    data: {title:title,
         body:body},
    success: function(res)
          {
      if (res){
        $('#notify_done').
             addClass('
             alert alert-
             success text-
             center');
        $('#notify_done').
             html("SUCCESS
                    : Your
                    notification
                    was sent.”);
                $('#mail_button').
                    attr('
                    disabled',
                    true);
              }else{
                $('#notify_done').
                    addClass('
                    alert alert-
                    danger text-
                    center');
                $('#notify_done').
                    html(”ERROR:
                    Your
                    notification
                    was not sent
                    .”);
              }
            }
          });
        });

        $('#ballot').html(””);
        $('#code').html(” ”+
            codeString);
                  $('#receipt').delay(200).
                      slideDown(800,
                      function(){
                  $('body').css({'cursor' :
                      'default'});
                  });
                }
              }
            });
          });
        }
      };

      app.vote(electionId, voterId,
          positionId, candidateId, clbck);

    });

    app.init();

  });

  </script>

</body>
</html>
```

ii. inactive.php

```
<div class=”content container-fluid text-left”
    id='ballot'>

  <div class='alert alert-danger text-center'>
  <strong> You are not involved in any ongoing
      election. </strong>
  <br> If you have been informed that you are
     registered in an active election please
      notify a voting official.
  </div>

  <div class='text-center'>
    <button class=”btn btn-success btn-danger”
         data-toggle=”modal” data-target=”#
         notifyModal” id='mail_button'>
      <span class='glyphicon glyphicon-
          envelope'></span> Notify Voting
          Officials</button>
  </div>

  <hr>

  <div id='notify_done'>
  </div>

</div>
```

iii. navbar.php

```
  <nav class=”navbar navbar-default”>
  <img src=”<?php echo asset_url();?>img/
     upm_logo.png” alt=”UPM Logo” align='
     left'>
    <div class=”container-fluid”>
      <div class=”navbar-header”>
        <a class=”navbar-brand” href=”#”>
            Votemind</a>
      </div>
      <div class=”collapse navbar-collapse” id
          =”myNavbar”>
        <ul class=”nav navbar-nav navbar-right
            ”>
          <li><div class='a-btn'><a class='btn
               btn-warning' href=”<?php echo
               linkbase_url();?>voter/logout
               ”><span class=”glyphicon
               glyphicon-log-out”></span>
               Logout</a></div></li>
        </ul>
      </div>
    </div>
  </nav>
```

iv. notify_modal.php

```
<!-- Add Modal -->
<div id=”notifyModal” class=”modal fade” role
    =”dialog”>
  <div class=”modal-dialog”>

    <!-- Add Modal content-->
    <div class=”modal-content”>

      <div class=”modal-header modal-header-
          red”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=”
             glyphicon glyphicon-plus”></span>
              Notify Voting Officials</h4>
      </div>

      <div class=”modal-body”>

        <div class=”modal-form”>

            <div class='form-group'>
              <label>Title: </label>
              <input type='text' name='title'
                   class=”form-control” id='
                   notify_title' required>
            </div>

            <div class='form-group'>
              <div><strong>Body:</strong></div
                   >
              <div class='text-center'><
                   textarea rows=”4” name=”
                   body” id=”notify_body”
                   class=”form-control”
                   required></textarea>
                </div>
            </div>

        </div>
      </div>

      <div class=”modal-footer”>

        <button class=”btn btn-success pull-
             left” id='notify_button' data-
             dismiss=”modal”><span class=”
             glyphicon glyphicon-send”></span>
          Send</button>
        <button class=”btn btn-danger pull-
             right” data-dismiss=”modal”><span
              class=”glyphicon glyphicon-
             remove”></span> Cancel</button>

      </div>
    </div>
  </div>
</div>
```

v. receipt.php

```
<!−− <div class=”content container−fluid text−
     left”>   −−>

  <hr><h1 align=’center’> Thank you for voting
       ! </h1> <hr>
  <div id=’status’ class=’alert text−justify
       ’></div>
  <div id=’instructions’ class=’alert text−
       justify’></div><hr>
  <h2 align=’center’> Here is your receipt:</
       h2><hr>
  <h4><strong>Your Alias: <?php echo $alias
       ;?></strong></h4>
  <h4><strong>Your Verification code:<span id
       =’code’></span></strong></h4>

    <?php foreach($elections as $election){ ?>

    <div class=’well well−sm text−left’>
      <strong><?php echo $election[’name’].”
           Election”;?></strong>
    </div>

      <?php $positions = $election[’positions
           ’];

      for($i=0;$i<count($positions);$i++){

        $row = $positions[$i];?>

        <div class=’form−group’>

          <div>
            <strong><?php echo $row[’
                 position_name’];?></strong>
          </div>

          <div id=”<?php echo $row[’
               position_id’].’_receipt’;?>”>
            <!−− jQuery will insert values
                 here −−>
          </div>

        </div>

     <?php }
    } ?>

    <hr>
    <button name=’MY RECEIPT’ class=’btn btn−
         primary’ id=’downloadButton’>
      <span class=’glyphicon glyphicon−floppy−
           disk’></span> Save this as a .txt
           file
    </button>
    <button class=”btn btn−success btn−danger”
          data−toggle=”modal” data−target=”#
         notifyModal” id=’mail_button’>
      <span class=’glyphicon glyphicon−
           envelope’></span> Notify Voting
           Officials
    </button>

    <hr>
    <div id=’notify_done’>
    </div>

<!−− </div> −−>
```

vi. vote.php

```
<div class=”content container−fluid text−left”
      id=’ballot’>

    <?php $attributes = array(’id’=>’
         verification_form’);
    echo form_open(’’,$attributes);
    foreach($elections as $election){ ?>

      <hr>
      <div>
        <h3 align=’center’><strong><?php echo
             $election[’name’].” Election
             ”;?></strong></h3>
      </div>
      <hr>

      <?php
      $i=0;
      $candidates = $election[’candidates’];

      for($i=0;$i<count($candidates);){

        $row = $candidates[$i];
        $position_id = $row[’position_id’];?>

        <div class=’well well−sm’>
          <strong><?php echo $row[’
               position_name’];?> (Max: <?php
               echo $row[’max’];?>)</strong>
          <strong class=’pull−right’>Hash
               Value</strong>
        </div>

        <?php while($i<count($candidates) &&
             $row[’position_id’] ==
             $position_id){
           if($row[’max’] == 1){ ?>

              <div class=”radio”>
                <label>
                  <input type=”radio” name=”<?
                       php echo $row[’
                       position_id’];?>”
                       electionid=<?php echo
                       $election[’election_id
                       ’];?>
                   value=<?php echo $row[’
                       candidate_id’];?>
                       required>
                    <span id=”<?php echo $row
                         [’position_id’].$row
                         [’candidate_id’].’
                         _name’;?>”>
                      <?php echo $row[’
                           last_name’].”, ”.
                           $row[”first_name”];
                      if(!empty($row[’
                           party_name’])) echo
                            ” (”.$row[’
                           party_name’].”)”;
                      else echo ” (INDEPENDENT
                           )”;
                      ?>
                    </span>
                </label>
                <strong id=”<?php echo $row[’
                     position_id’].$row[’
                     candidate_id’].’_hash
                     ’;?>”
                   class=”<?php echo $row[’
                        position_id’].’_hash
                        ’;?> pull−right”></
                        strong>
              </div>

            <?php }else{ ?>

              <div class=”checkbox”>
                <label>
                  <input type=”checkbox” name
                       =”<?php echo $row[’
                       position_id’];?>” value
                       =<?php echo $row[’
                       candidate_id’];?>
                   electionid=<?php echo
                       $election[’election_id
                       ’];?>>
                    <span id=”<?php echo $row
                         [’position_id’].$row
                         [’candidate_id’].’
                         _name’;?>”>
                      <?php echo $row[’
                           last_name’].”, ”.
                           $row[”first_name”];
                      if(!empty($row[’
                           party_name’])) echo
                            ” (”.$row[’
                           party_name’].”)”;
                      else echo ” (INDEPENDENT
                           )”;
                      ?>
                    </span>
                </label>
                <strong id=”<?php echo $row[’
                     position_id’].$row[’
                     candidate_id’].’_hash
                     ’;?>”
                   class=”<?php echo $row[’
                        position_id’].’_hash
                        ’;?> pull−right”></
                        strong>
              </div>

            <?php }
        $prevrow = $row;
        $i++;

        if($i < count($candidates)) $row =
             $candidates[$i];
      }
      if($prevrow[’max’] == 1){?>

        <div class=”radio”>
          <label>
            <input type=”radio” name=”<?
                 php echo $prevrow[’
                 position_id’];?>” value
                 =’0’
             electionid=<?php echo
                 $election[’election_id
                 ’];?>>
              <span id=”<?php echo
                  $prevrow[’position_id
                  ’].’0_name’;?>”>
                Abstain
              </span>
          </label>
          <strong id=”<?php echo $prevrow
               [’position_id’].’0_hash
               ’;?>”
             class=”<?php echo $prevrow[’
                 position_id’].’_hash’;?>
                 pull-right”></strong>
        </div>

        <div class=”errors alert alert-
             danger” id=”<?php echo
             $prevrow[’position_id’].”
             _error”;?>”>
          You must select one
        </div>
    <?php } else { ?>
        <div class=”errors alert alert-
             danger” id=”<?php echo
             $prevrow[’position_id’].”
             _error”;?>”>
          You must choose within the
              maximum.
        </div>
    <?php }
```

```
      }
    }?>
    </form>
    <div class=’text-center’>
      <button id=’submit_vote_button’ class=”
           btn btn-success btn-lg” data-toggle
           =”modal” data-target=”#voteModal”>
           Submit Vote</button>
    </div>

  </div>

  <!-- Error Modal -->
  <div id=”error_alert” class=”modal fade” role
       =”dialog”>
    <div class=”modal-dialog”>

      <!-- Modal content -->
      <div class=”modal-content”>
        <div class=”modal-header modal-header-
             red”>
          <button type=”button” class=”close”
               data-dismiss=”modal”>&times;</
               button>
          <h4 class=”modal-title”><span class=’
               glyphicon glyphicon-remove-sign
               ’></span> Invalid Ballot</h4>
        </div>
        <div class=”modal-body”>
          <p>Your ballot seems to be invalid.
               Please double check.</p>
        </div>
        <div class=”modal-footer”>
          <button type=”button” class=”btn btn-
               default” data-dismiss=”modal”>
               Close</button>
        </div>
      </div>

    </div>
  </div>

  <div class=”content container-fluid text-left”
       id=’receipt’>

  </div>
```

vii. vote_modal.php

```
<!-- Vote Modal -->
<div id=”voteModal” class=”modal fade” role=”
     dialog”>
  <div class=”modal-dialog”>

    <!-- Vote Modal content-->
    <div class=”modal-content”>

      <div class=”modal-header”>
        <button type=”button” class=”close”
             data-dismiss=”modal”>&times;</
             button>
        <h4 class=”modal-title”><span class=’
             glyphicon glyphicon-check’></span
             > Confirm Vote</h4>
      </div>

       <div class=”modal-body”>

        <div class=’alert alert-danger fade in
             text-center’>
          <strong>You can only vote once.</
               strong> <br> Please double
               check your votes and keep a
               copy of these values.
        </div>

        <div class=’text-center’>
          <button name=’MY BALLOT’ class=’btn
               btn-primary pull-center’ id=’
               downloadButton’>
            <span class=’glyphicon glyphicon-
                 floppy-disk’></span> Save
                 this as a .txt file
          </button>
        </div>
        <br>

        <div class=”modal-form”>

          <?php

        foreach($elections as $election){ ?>

          <div class=’well well-sm text-center
               ’>
            <strong><?php echo $election[’name
                 ’].” Election”;?></strong>
            <input name=’election_dl’ class=’
                 downloadable’ type=’hidden’
                 value=”<?php echo $election[’
                 name’].’ Election’;?>” >
          </div>

          <?php

          $positions = $election[’positions’];

          for($i=0;$i<count($positions);$i++){

            $row = $positions[$i];?>

            <div class=’form-group’>

              <div>
                <strong><?php echo $row[’
                     position_name’];?></
                     strong>
                <input name=’position_dl’
                     class=’downloadable’ type
                     =’hidden’ value=”<?php
                     echo $row[’position_name
                     ’];?>” >
              </div>

              <div class=’downloadable’ id=”<?
                   php echo $row[’position_id
                   ’].’_confirm’;?>”>
                <!-- jQuery will insert values
                     here -->
              </div>

            </div>

          <?php }

        }?>

        </div>
      </div>

      <div class=”modal-footer”>
```

```
        <button class='btn btn-success pull-
            left' id='submitButton' data-
            dismiss='modal'>
          <span class='glyphicon glyphicon-ok
              '></span> Yes
        </button>

        <button class="btn btn-danger pull-
            right" data-dismiss="modal">
          <span class='glyphicon glyphicon-
              remove'></span> No
        </button>

      </div>

    </div>

  </div>
</div>
```

viii. voted.php

```
<div class="content container-fluid text-left"
    id='ballot'>

  <div class='alert alert-danger text-center'>
  Sorry, you can only vote once. <br> We have
      seen in our records that you have
      already voted for the <strong>
    <?php
    for($i=0;$i<count($elections);$i++){
      if($i==0)
        echo $elections[$i]['name'];
      else
        echo ", ".$elections[$i]['name'];

    }
    ?>
  </strong>Election(s). <br> If it is not you
      who voted please notify the voting
      officials.
  </div>

  <div class='text-center'>
    <button class="btn btn-success btn-danger"
        data-toggle="modal" data-target="#
        notifyModal" id='mail_button'>
      <span class='glyphicon glyphicon-
          envelope'></span> Notify Voting
          Officials</button>
  </div>

  <hr>

  <div id='notify_done'>
  </div>

</div>
```

ix. voted_footer.php

```
    <!-- <div id="footer">Copyright &copy;
        2016 Cybernetica AS</div> -->
  <footer class="container-fluid text-left
      navbar-default">
    <p>&copy; University of the Philippines
        Manila</p>
  </footer>
  <script src="<?php echo asset_url();?>js/
      jquery-1.12.0.min.js"></script>
  <script src="<?php echo asset_url();?>js/
      bootstrap.js"></script>
  <script type="text/javascript">

    //When page has finished loading...
    $(document).ready(function () {

      $('.content').hide();
      $('.content').delay(200).slideDown(800);

      $('#notify_button').on('click', function
          (e){
        var title = $('#notify_title').val();
        var body = $('#notify_body').val();

        jQuery.ajax({
          type: "POST",
          url: "<?php echo linkbase_url(); ?>"
              + "voter/notify",
          data: {title:title, body:body},
          success: function(res) {
            if (res){
              $('#notify_done').addClass('
                  alert alert-success text-
                  center');
              $('#notify_done').html("SUCCESS:
                  Your notification was sent
                  .");
              $('#mail_button').attr('disabled
                  ',true);
            }else{
              $('#notify_done').addClass('
                  alert alert-danger text-
                  center');
              $('#notify_done').html("ERROR:
                  Your notification was not
                  sent.");
            }
          }
        });

      });

    });

  </script>

</body>
</html>
```

(e) templates

i. footer.php

```
    <!-- <div id="footer">Copyright &copy;
        2016 Cybernetica AS</div> -->
  <footer class="container-fluid text-left
      navbar-default">
    <p>&copy; University of the Philippines
        Manila</p>
  </footer>
  <script src="<?php echo asset_url();?>js/
      jquery-1.12.0.min.js"></script>
  <script src="<?php echo asset_url();?>js/
      bootstrap.js"></script>
  <script type="text/javascript">
    //When page has finished loading...
    $(document).ready(function () {
      $('.content').hide();
      $('.content').delay(200).slideDown(800);
    });
  </script>
</body>
</html>
```

ii. header.php

```
<!DOCTYPE HTML>
<html>
<head>
  <title>Sharemind Secure Voting</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="<?php echo
      asset_url();?>css/bootstrap.min.css"
      type="text/css">
  <link rel="stylesheet" href="<?php echo
      asset_url();?>css/custom.css" type="
      text/css">
  <link rel="stylesheet" href="<?php echo
      asset_url();?>css/dataTables.bootstrap.
      min.css" type="text/css">
  <link rel="stylesheet" href="<?php echo
      asset_url();?>css/rowReorder.bootstrap.
      min.css" type="text/css">
</head>
```

```
<body>
```

4. utility_helper.php

```
<?php if ( ! defined('BASEPATH')) exit('No
     direct script access allowed');

function asset_url(){
   return base_url().'assets/';
}
```

```
function linkbase_url(){
   return base_url().'index.php/';
}
```

5. config

(a) config.php

```
<?php
defined('BASEPATH') OR exit('No direct script
     access allowed');

/*
|--------------------------------------------------------------------------
| Base Site URL
|--------------------------------------------------------------------------
|
| URL to your CodeIgniter root. Typically this
      will be your base URL,
| WITH a trailing slash:
|
|       http://example.com/
|
| WARNING: You MUST set this value!
|
| If it is not set, then CodeIgniter will try
     guess the protocol and path
| your installation, but due to security
     concerns the hostname will be set
| to $_SERVER['SERVER_ADDR'] if available, or
     localhost otherwise.
| The auto-detection mechanism exists only for
      convenience during
| development and MUST NOT be used in
     production!
|
| If you need to allow multiple domains,
     remember that this file is still
| a PHP script and you can easily do that on
     your own.
|
*/
$config['base_url'] = '';

/*
|--------------------------------------------------------------------------
| Index File
|--------------------------------------------------------------------------
|
| Typically this will be your index.php file,
     unless you've renamed it to
| something else. If you are using mod_rewrite
      to remove the page set this
| variable so that it is blank.
|
*/
$config['index_page'] = '';

/*
|--------------------------------------------------------------------------
| URI PROTOCOL
|--------------------------------------------------------------------------
|
| This item determines which server global
     should be used to retrieve the
| URI string.  The default setting of '
     REQUEST_URI' works for most servers.
| If your links do not seem to work, try one
     of the other delicious flavors:
|
| 'REQUEST_URI'     Uses $_SERVER['REQUEST_URI
     ']
| 'QUERY_STRING'    Uses $_SERVER['QUERY_STRING
     ']
| 'PATH_INFO'       Uses $_SERVER['PATH_INFO']
|
| WARNING: If you set this to 'PATH_INFO',
     URIs will always be URL-decoded!
*/
$config['uri_protocol'] = 'REQUEST_URI';
```

```
/*
|--------------------------------------------------------------------------
| URL suffix
|--------------------------------------------------------------------------
|
| This option allows you to add a suffix to
     all URLs generated by CodeIgniter.
| For more information please see the user
     guide:
|
| https://codeigniter.com/user_guide/general/
     urls.html
*/
$config['url_suffix'] = '';

/*
|--------------------------------------------------------------------------
| Default Language
|--------------------------------------------------------------------------
|
| This determines which set of language files
     should be used. Make sure
| there is an available translation if you
     intend to use something other
| than english.
|
*/
$config['language']     = 'english';

/*
|--------------------------------------------------------------------------
| Default Character Set
|--------------------------------------------------------------------------
|
| This determines which character set is used
     by default in various methods
| that require a character set to be provided.
|
| See http://php.net/htmlspecialchars for a
     list of supported charsets.
|
*/
$config['charset'] = 'UTF-8';

/*
|--------------------------------------------------------------------------
| Enable/Disable System Hooks
|--------------------------------------------------------------------------
|
| If you would like to use the 'hooks' feature
     you must enable it by
| setting this variable to TRUE (boolean).
     See the user guide for details.
|
*/
$config['enable_hooks'] = FALSE;

/*
|--------------------------------------------------------------------------
| Class Extension Prefix
|--------------------------------------------------------------------------
|
| This item allows you to set the filename/
     classname prefix when extending
| native libraries.  For more information
     please see the user guide:
|
| https://codeigniter.com/user_guide/general/
     core_classes.html
```

```
| https://codeigniter.com/user_guide/general/
    creating_libraries.html
|
*/
$config['subclass_prefix'] = 'MY_';

/*
|--------------------------------------------------------------------------
| Composer auto-loading
|--------------------------------------------------------------------------
|
| Enabling this setting will tell CodeIgniter
    to look for a Composer
| package auto-loader script in application/
    vendor/autoload.php.
|
|       $config['composer_autoload'] = TRUE;
|
| Or if you have your vendor/ directory
    located somewhere else, you
| can opt to set a specific path as well:
|
|       $config['composer_autoload'] = '/path/
    to/vendor/autoload.php';
|
| For more information about Composer, please
    visit http://getcomposer.org/
|
| Note: This will NOT disable or override the
    CodeIgniter-specific
|       autoloading (application/config/
    autoload.php)
*/
$config['composer_autoload'] = FALSE;

/*
|--------------------------------------------------------------------------
| Allowed URL Characters
|--------------------------------------------------------------------------
|
| This lets you specify which characters are
    permitted within your URLs.
| When someone tries to submit a URL with
    disallowed characters they will
| get a warning message.
|
| As a security measure you are STRONGLY
    encouraged to restrict URLs to
| as few characters as possible.  By default
    only these are allowed: a-z 0-9~%.:_-
|
| Leave blank to allow all characters -- but
    only if you are insane.
|
| The configured value is actually a regular
    expression character group
| and it will be executed as: ! preg_match
    ('/^[<permitted_uri_chars>]+$/i
|
| DO NOT CHANGE THIS UNLESS YOU FULLY
    UNDERSTAND THE REPERCUSSIONS!!
|
*/
$config['permitted_uri_chars'] = 'a-z 0-9~%.:_
    \-';

/*
|--------------------------------------------------------------------------
| Enable Query Strings
|--------------------------------------------------------------------------
|
| By default CodeIgniter uses search-engine
    friendly segment based URLs:
| example.com/who/what/where/
|
| By default CodeIgniter enables access to the
    $_GET array.  If for some
| reason you would like to disable it, set '
    allow_get_array' to FALSE.
|
| You can optionally enable standard query
    string based URLs:
| example.com?who=me&what=something&where=here
|
| Options are: TRUE or FALSE (boolean)
|
| The other items let you set the query string
    'words' that will
| invoke your controllers and its functions:
| example.com/index.php?c=controller&m=
    function
|
| Please note that some of the helpers won't
    work as expected when
| this feature is enabled, since CodeIgniter
    is designed primarily to
| use segment based URLs.
|
*/
$config['allow_get_array'] = TRUE;
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
$config['function_trigger'] = 'm';
$config['directory_trigger'] = 'd';

/*
|--------------------------------------------------------------------------
| Error Logging Threshold
|--------------------------------------------------------------------------
|
| You can enable error logging by setting a
    threshold over zero. The
| threshold determines what gets logged.
    Threshold options are:
|
|       0 = Disables logging, Error logging
    TURNED OFF
|       1 = Error Messages (including PHP
    errors)
|       2 = Debug Messages
|       3 = Informational Messages
|       4 = All Messages
|
| You can also pass an array with threshold
    levels to show individual error types
|
|       array(2) = Debug Messages, without
    Error Messages
|
| For a live site you'll usually only enable
    Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
*/
$config['log_threshold'] = 0;

/*
|--------------------------------------------------------------------------
| Error Logging Directory Path
|--------------------------------------------------------------------------
|
| Leave this BLANK unless you would like to
    set something other than the default
| application/logs/ directory. Use a full
    server path with trailing slash.
|
*/
$config['log_path'] = '';

/*
|--------------------------------------------------------------------------
| Log File Extension
|--------------------------------------------------------------------------
|
| The default filename extension for log files
    . The default 'php' allows for
| protecting the log files via basic scripting
    , when they are to be stored
| under a publicly accessible directory.
|
| Note: Leaving it blank will default to 'php
    '.
|
*/
$config['log_file_extension'] = '';

/*
|--------------------------------------------------------------------------
| Log File Permissions
|--------------------------------------------------------------------------
|
| The file system permissions to be applied on
     newly created log files.
|
| IMPORTANT: This MUST be an integer (no
    quotes) and you MUST use octal
|            integer notation (i.e. 0700,
    0644, etc.)
*/
$config['log_file_permissions'] = 0644;

/*
|--------------------------------------------------------------------------
```

```
| Date Format for Logs
|--------------------------------------------------------------------------
|
| Each item that is logged has an associated
    date. You can use PHP date
| codes to set your own date formatting
|
*/
$config['log_date_format'] = 'Y-m-d H:i:s';

/*
|--------------------------------------------------------------------------
| Error Views Directory Path
|--------------------------------------------------------------------------
|
| Leave this BLANK unless you would like to
    set something other than the default
| application/views/errors/ directory.  Use a
    full server path with trailing slash.
|
*/
$config['error_views_path'] = '';

/*
|--------------------------------------------------------------------------
| Cache Directory Path
|--------------------------------------------------------------------------
|
| Leave this BLANK unless you would like to
    set something other than the default
| application/cache/ directory.  Use a full
    server path with trailing slash.
|
*/
$config['cache_path'] = '';

/*
|--------------------------------------------------------------------------
| Cache Include Query String
|--------------------------------------------------------------------------
|
| Whether to take the URL query string into
    consideration when generating
| output cache files. Valid options are:
|
|      FALSE      = Disabled
|      TRUE       = Enabled, take all query
    parameters into account.
|                   Please be aware that this
     may result in numerous cache
|                   files generated for the
    same page over and over again.
|      array('q') = Enabled, but only take
    into account the specified list
|                   of query parameters.
|
*/
$config['cache_query_string'] = FALSE;

/*
|--------------------------------------------------------------------------
| Encryption Key
|--------------------------------------------------------------------------
|
| If you use the Encryption class, you must
    set an encryption key.
| See the user guide for more info.
|
| https://codeigniter.com/user_guide/libraries
    /encryption.html
|
*/
$config['encryption_key'] = '
    FafMuh2vdex9618e3dVSX6dCpoBIFJ5X';

/*
|--------------------------------------------------------------------------
| Session Variables
|--------------------------------------------------------------------------
|
| 'sess_driver'
|
|      The storage driver to use: files,
    database, redis, memcached
|
| 'sess_cookie_name'
|
|      The session cookie name, must contain
    only [0-9a-z_-] characters
|
| 'sess_expiration'
|
|      The number of SECONDS you want the
    session to last.
|      Setting to 0 (zero) means expire when
    the browser is closed.
|
| 'sess_save_path'
|
|      The location to save sessions to,
    driver dependent.
|
|      For the 'files' driver, it's a path to
     a writable directory.
|      WARNING: Only absolute paths are
    supported!
|
|      For the 'database' driver, it's a
    table name.
|      Please read up the manual for the
    format with other session drivers.
|
|      IMPORTANT: You are REQUIRED to set a
    valid save path!
|
| 'sess_match_ip'
|
|      Whether to match the user's IP address
     when reading the session data.
|
|      WARNING: If you're using the database
    driver, don't forget to update
|               your session table's PRIMARY
    KEY when changing this setting.
|
| 'sess_time_to_update'
|
|      How many seconds between CI
    regenerating the session ID.
|
| 'sess_regenerate_destroy'
|
|      Whether to destroy session data
    associated with the old session ID
|      when auto-regenerating the session ID.
     When set to FALSE, the data
|      will be later deleted by the garbage
    collector.
|
| Other session cookie settings are shared
    with the rest of the application,
| except for 'cookie_prefix' and '
    cookie_httponly', which are ignored here.
|
*/
$config['sess_driver'] = 'files';
$config['sess_cookie_name'] = 'ci_session';
$config['sess_expiration'] = 7200;
$config['sess_save_path'] = NULL;
$config['sess_match_ip'] = FALSE;
$config['sess_time_to_update'] = 300;
$config['sess_regenerate_destroy'] = FALSE;

/*
|--------------------------------------------------------------------------
| Cookie Related Variables
|--------------------------------------------------------------------------
|
| 'cookie_prefix'   = Set a cookie name prefix
     if you need to avoid collisions
| 'cookie_domain'   = Set to .your-domain.com
    for site-wide cookies
| 'cookie_path'     = Typically will be a
    forward slash
| 'cookie_secure'   = Cookie will only be set
    if a secure HTTPS connection exists.
| 'cookie_httponly' = Cookie will only be
    accessible via HTTP(S) (no javascript)
|
| Note: These settings (with the exception of
    'cookie_prefix' and
|       'cookie_httponly') will also affect
    sessions.
|
*/
$config['cookie_prefix']        = '';
$config['cookie_domain']        = '';
$config['cookie_path']          = '/';
$config['cookie_secure']        = FALSE;
$config['cookie_httponly']      = FALSE;

/*
|--------------------------------------------------------------------------
| Standardize newlines
```

```
|--------------------------------------------------------------------------
|
| Determines whether to standardize newline
    characters in input data,
| meaning to replace \r\n, \r, \n occurrences
    with the PHP_EOL value.
|
| This is particularly useful for portability
    between UNIX-based OSes,
| (usually \n) and Windows (\r\n).
|
*/
$config['standardize_newlines'] = FALSE;

/*
|--------------------------------------------------------------------------
| Global XSS Filtering
|--------------------------------------------------------------------------
|
| Determines whether the XSS filter is always
    active when GET, POST or
| COOKIE data is encountered
|
| WARNING: This feature is DEPRECATED and
    currently available only
|          for backwards compatibility
    purposes!
|
*/
$config['global_xss_filtering'] = FALSE;

/*
|--------------------------------------------------------------------------
| Cross Site Request Forgery
|--------------------------------------------------------------------------
| Enables a CSRF cookie token to be set. When
    set to TRUE, token will be
| checked on a submitted form. If you are
    accepting user data, it is strongly
| recommended CSRF protection be enabled.
|
| 'csrf_token_name' = The token name
| 'csrf_cookie_name' = The cookie name
| 'csrf_expire' = The number in seconds the
    token should expire.
| 'csrf_regenerate' = Regenerate token on
    every submission
| 'csrf_exclude_uris' = Array of URIs which
    ignore CSRF checks
*/
$config['csrf_protection'] = FALSE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] = '
    csrf_cookie_name';
$config['csrf_expire'] = 7200;
$config['csrf_regenerate'] = TRUE;
$config['csrf_exclude_uris'] = array();

/*
|--------------------------------------------------------------------------
| Output Compression
|--------------------------------------------------------------------------
|
| Enables Gzip output compression for faster
    page loads. When enabled,
| the output class will test whether your
    server supports Gzip.
| Even if it does, however, not all browsers
    support compression
| so enable only if you are reasonably sure
    your visitors can handle it.
|
| Only used if zlib.output_compression is
    turned off in your php.ini.
| Please do not use it together with httpd-
    level output compression.
|--------------------------------------------------------------------------
| VERY IMPORTANT: If you are getting a blank
    page when compression is enabled it
| means you are prematurely outputting
    something to your browser. It could
| even be a line of whitespace at the end of
    one of your scripts. For
| compression to work, nothing can be sent
    before the output buffer is called
| by the output class. Do not 'echo' any
    values with compression enabled.
|
*/
$config['compress_output'] = FALSE;

/*
|--------------------------------------------------------------------------
| Master Time Reference
|--------------------------------------------------------------------------
|
| Options are 'local' or any PHP supported
    timezone. This preference tells
| the system whether to use your server's
    local time as the master 'now'
| reference, or convert it to the configured
    one timezone. See the 'date
| helper' page of the user guide for
    information regarding date handling.
|
*/
$config['time_reference'] = 'local';

/*
|--------------------------------------------------------------------------
| Rewrite PHP Short Tags
|--------------------------------------------------------------------------
|
| If your PHP installation does not have short
     tag support enabled CI
| can rewrite the tags on-the-fly, enabling
    you to utilize that syntax
| in your view files. Options are TRUE or
    FALSE (boolean)
|
| Note: You need to have eval() enabled for
    this to work.
|
*/
$config['rewrite_short_tags'] = FALSE;

/*
|--------------------------------------------------------------------------
| Reverse Proxy IPs
|--------------------------------------------------------------------------
|
| If your server is behind a reverse proxy,
    you must whitelist the proxy
| IP addresses from which CodeIgniter should
    trust headers such as
| HTTP_X_FORWARDED_FOR and HTTP_CLIENT_IP in
    order to properly identify
| the visitor's IP address.
|
| You can use both an array or a comma-
    separated list of proxy addresses,
| as well as specifying whole subnets. Here
    are a few examples:
|
| Comma-separated:
    '10.0.1.200,192.168.5.0/24'
| Array:                array('10.0.1.200',
    '192.168.5.0/24')
*/
$config['proxy_ips'] = '';

$config['installed'] = 0;
```

(b) database.php

```
<?php  if ( ! defined('BASEPATH')) exit('No
    direct script access allowed');
/*
|
    --------------------------------------------------------------------------
| DATABASE CONNECTIVITY SETTINGS
|
    --------------------------------------------------------------------------
| This file will contain the settings needed
    to access your database.
|
| For complete instructions please consult the
     ”Database Connection”
| page of the User Guide.
|
|
    --------------------------------------------------------------------------
| EXPLANATION OF VARIABLES
|
    ------------------------------------------------
|
|       ['hostname'] The hostname of your
    database server.
|       ['username'] The username used to
    connect to the database
|       ['password'] The password used to
    connect to the database
|       ['database'] The name of the database
    you want to connect to
|       ['dbdriver'] The database type. ie:
    mysql.  Currently supported:
                                    mysql, mysqli
                                        ,
                                        postgre,
                                         odbc,
                                        mssql,
                                        sqlite,
                                        oci8
|       ['dbprefix'] You can add an optional
    prefix, which will be added
|                                   to the table
    name when using the  Active Record class
|       ['pconnect'] TRUE/FALSE - Whether to
    use a persistent connection
|       ['db_debug'] TRUE/FALSE - Whether
    database errors should be displayed.
|       ['cache_on'] TRUE/FALSE - Enables/
    disables query caching
|       ['cachedir'] The path to the folder
    where cache files should be stored
|       ['char_set'] The character set used in
     communicating with the database
|       ['dbcollat'] The character collation
    used in communicating with the database
|
| The $active_group variable lets you choose
    which connection group to
| make active.  By default there is only one
    group (the "default" group).
|
| The $active_record variables lets you
    determine whether or not to load
| the active record class
*/

// The following values will probably need to
    be changed.
$db['default']['username'] = "root";
$db['default']['password'] = "537608234";
$db['default']['database'] = "Voting";

// The following values can probably stay the
    same.
$db['default']['hostname'] = "localhost";
$db['default']['dbdriver'] = "mysqli"; //
    Updated to latest driver.
$db['default']['dbprefix'] = "";
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = FALSE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = "";
$db['default']['char_set'] = "utf8";
$db['default']['dbcollat'] = "utf8_general_ci
    ";

$active_group = "default";
$active_record = TRUE;

/* End of file database.php */
/* Location: ./application/config/database.php
     */
```

(c) email.php

```
<?php defined('BASEPATH') OR exit('No direct
    script access allowed.');

$config['useragent']        = 'PHPMailer';
                // Mail engine switcher: '
    CodeIgniter' or 'PHPMailer'
$config['protocol']         = 'smtp';
                    // 'mail', 'sendmail',
    or 'smtp'
$config['mailpath']         = '/usr/sbin/
    sendmail';
$config['smtp_host']        = 'smtp.gmail.com
    ';
$config['smtp_user']        = 'votemind@gmail.
    com';
$config['smtp_pass']        = 'votemind123';
$config['smtp_port']        = 587;
$config['smtp_timeout']     = 30;
                            // (in seconds)
$config['smtp_crypto']      = 'tls';
                            // '' or 'tls' or '
    ssl'
$config['smtp_debug']       = 0;
                            // PHPMailer's
    SMTP debug info level: 0 = off, 1 =
    commands, 2 = commands and data, 3 = as 2
     plus connection status, 4 = low level
    data output.
$config['smtp_auto_tls']    = true;
                        // Whether to enable
    TLS encryption automatically if a server
    supports it, even if `smtp_crypto` is not
     set to 'tls'.
$config['smtp_conn_options'] = array();
                    // SMTP connection
    options, an array passed to the function
    stream_context_create() when connecting
    via SMTP.
$config['wordwrap']         = true;
$config['wrapchars']        = 76;
$config['mailtype']         = 'html';
                    // 'text' or 'html'
$config['charset']          = null;
                        // 'UTF-8', 'ISO
    -8859-15', ...; NULL (preferable) means
    config_item('charset'), i.e. the
    character set of the site.
$config['validate']         = true;
$config['priority']         = 3;
                            // 1, 2, 3, 4, 5;
    on PHPMailer useragent NULL is a possible
     option, it means that X-priority header
    is not set at all, see https://github.com
    /PHPMailer/PHPMailer/issues/449
$config['crlf']             = "\n";
                            // "\r\n" or "\n" or
    "\r"
$config['newline']          = "\n";
                            // "\r\n" or "\n" or
    "\r"
$config['bcc_batch_mode']   = false;
$config['bcc_batch_size']   = 200;
$config['encoding']         = '8bit';
                        // The body encoding.
    For CodeIgniter: '8bit' or '7bit'. For
    PHPMailer: '8bit', '7bit', 'binary', '
    base64', or 'quoted-printable'.

// DKIM Signing
// See https://yomotherboard.com/how-to-setup-
    email-server-dkim-keys/
// See http://stackoverflow.com/questions
    /24463425/send-mail-in-phpmailer-using-
    dkim-keys
// See https://github.com/PHPMailer/PHPMailer/
    blob/v5.2.14/test/phpmailerTest.php#L1708
$config['dkim_domain']      = '';
                            // DKIM signing
    domain name, for exmple 'example.com'.
$config['dkim_private']     = '';
                            // DKIM private key
     file path.
$config['dkim_selector']    = '';
                            // DKIM selector.
$config['dkim_passphrase']  = '';
                            // DKIM passphrase,
     used if your key is encrypted.
$config['dkim_identity']    = '';
                            // DKIM Identity,
    usually the email address used as the
    source of the email.
```

(d) routes.php

```
<?php
defined('BASEPATH') OR exit('No direct script
    access allowed');

/*
|
    ------------------------------------------------
| URI ROUTING
|
    ------------------------------------------------
```

```
| This file lets you re-map URI requests to
    specific controller functions.
|
| Typically there is a one-to-one relationship
     between a URL string
| and its corresponding controller class/
    method. The segments in a
| URL normally follow this pattern:
|
|       example.com/class/method/id/
|
| In some instances, however, you may want to
    remap this relationship
| so that a different class/function is called
     than the one
| corresponding to the URL.
|
| Please see the user guide for complete
    details:
|
|       https://codeigniter.com/user_guide/
    general/routing.html
|
|
    -------------------------------------------------------------------------
| RESERVED ROUTES
|
    -------------------------------------------------------------------------
|
| There are three reserved routes:
|
|       $route['default_controller'] = '
    welcome';
|
| This route indicates which controller class
    should be loaded if the
| URI contains no data. In the above example,
    the "welcome" class
| would be loaded.
|
|       $route['404_override'] = 'errors/
    page_missing';
|
| This route will tell the Router which
    controller/method to use if those
| provided in the URL cannot be matched to a
    valid route.
|
|       $route['translate_uri_dashes'] = FALSE
    ;
|
| This is not exactly a route, but allows you
    to automatically route
| controller and method names that contain
    dashes. '-' isn't a valid
| class or method name character, so it
    requires translation.
| When you set this option to TRUE, it will
    replace ALL dashes in the
| controller and method URI segments.
|
| Examples:   my-controller/index      ->
    my_controller/index
|             my-controller/my-method ->
    my_controller/my_method
*/
// $route['news/create'] = 'news/create';
// $route['news/(:any)'] = 'news/view/$1';
// $route['news'] = 'news';
// $route['(:any)'] = 'pages/view/$1';

$route['default_controller'] = "login";
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

6. custom.css

```
/* Remove the navbar's default margin-bottom
     and rounded borders */
    @import url(//fonts.googleapis.com/css?
        family=Quicksand);
    @import url(//fonts.googleapis.com/css?
        family=Lato);

    body{
      background-color: RGB(240,240,240);
      padding-top: 3%;
      padding-bottom: 3%;
      padding-left: 3%;
      padding-right: 3%;
      font-family: 'Quicksand', sans-serif;
      font-size: 17px;
    }

    .content{
      padding-top: 0px;
      padding-bottom: 30px;
      padding-left: 5%;
      padding-right: 5%;
      background-color: white;
      border-style: solid;
      border-top-style: none;
      border-color: RGB(231,231,231);
      border-width: 1px;
    }

    input[type="text"] {
      font-size: 17px;
    }

    input[type="search"] {
      font-size: 17px;
    }

    option{
      font-size: 17px;
    }

    .a-btn{
      padding: 10px;
    }

    img{
      width:110px;
      height:110px;
      padding: 10px;
      float:left;
      margin-top: -25px;
      margin-left: -25px
    }

    .navbar {
      margin-bottom: 0;
      border-radius: 0;
      padding: 30px;
      padding-bottom: 10px;
      background-color: white;
      border-bottom-style: none;
    }

    /* Set height of the grid so .sidenav can
        be 100% (adjust as needed) */
    .row.content {
      height: 500px;
    }

    /* Set black background color, white text
        and some padding */
    footer.navbar-default{
      border-top-style: solid;
      border-width: 1px;
      padding: 10px;
      padding-bottom: 0px;
      background-color: RGB(240,240,240);
    }

    /* On small screens, set height to 'auto'
        for sidenav and grid */
    @media screen and (max-width: 767px) {
      .sidenav {
        height: auto;
        padding: 15px;
      }
      .row.content {height:auto;}
    }

    .form{
      width: 100%;
      align: center;
      margin: 0 auto;
    }

    .modal-form-control{
      width: 70%;
      margin: 0 auto;
    }

    .modal-form{
      width: 100%;
      align: center;
      margin: 0 auto;
    }

    .modal-header, .close {
      background-color: #5cb85c;
      color:white !important;
      text-align: center;
```

```
  font-size: 30px;
}

.modal-header-red {
  background-color: RGB(217,83,79);
}

.modal-header-blue {
  background-color: RGB(51,122,183);
}

.modal-header-orange {
  background-color: RGB(245,160,40);
}

.modal-footer {
    background-color: #f9f9f9;
}

.file{
  margin: 0 auto;
}

.dropdown{
  padding: 10px;
}

table.dataTable thead > tr > th.sorting,
table.dataTable thead > tr > th.
    sorting_asc,
table.dataTable thead > tr > th.
    sorting_desc{
  padding-left: 30px !important;
  padding-right: initial !important;
}

table.dataTable thead .sorting:after,
table.dataTable thead .sorting_asc:after,
table.dataTable thead .sorting_desc:after
    {
    left: 8px !important;
    right: auto !important;
}
```

7. miners.js

```
(function(app, sm, $){
  app.hosts = [
    "http://321.312.321.315:8081",
    "http://321.312.321.316:8082",
    "http://321.312.321.317:8083"
  ];
})(this.app = this.app === undefined ? {} :
    this.app, sm, jQuery);
```

8. app.js

```
/*
 * This file is a part of the Sharemind
     framework.
 * Copyright (C) Cybernetica AS
 *
 * All rights are reserved. Reproduction in
     whole or part is prohibited
 * without the written consent of the
     copyright owner. The usage of this
 * code is subject to the appropriate license
     agreement.
 */

(function(app, sm, $){
  // var hosts = [
  //   "http://172.16.126.162:8081",
  //   "http://172.16.126.114:8082",
  //   "http://172.16.126.62:8083"
  // ];

  var servers = null;
  var pub = sm.types.base;
  var priv = sm.types.shared3p;

  app.init = function() {
    $(window).on("allDone", function () {
      app.debug("All done!");
      serversDisconnect();
    });
  };

  function serversConnect(callback) {
    if (servers === null) {
      app.debug("Sharemind not yet connected
          .");

      try {
        app.log("Connecting to Sharemind.");

        servers = sm.ctrl.connect(
          app.hosts,
          function() {  // success callback
            app.log("Connected to Sharemind.")
                ;

            var bytes = new sm.types.base.
                Uint32Array(1);
            bytes.set(0, 32);  // 32 random
                bytes needed.

            var args = {};

            args["proxyParams"] = {"codefile":
                 "get_random.sb"}

            var smParams = {};
            smParams["bytes"] = bytes;
            args["smParams"] = smParams;

            servers.emit(
              "run_code",
              args,
              function (data) {
                app.log("Got randomness for
                    PRNG.");

                sm.types.prng = new PRNG(data.
                    random);

                if (sm.types.prng) {
                  app.log("Ready to send data
                      .");
                } else {
                  app.log("Failed to create
                      PRNG!");
                  serversDisconnect();
                  return;
                }

                // Successfully connected.
                    Execute callback.
                callback();
              });
          },
          function(error) {  // on error
              function
            app.error(error);
            callback(error);
            $(window).trigger("allDone");
          }
        );
      } catch (err) {
        app.debug("Failed to connect to
            Sharemind: " + err);
        callback("Failed to connect to
            Sharemind: " + err);
        serversDisconnect();
        return;
      }
    } else {
      // Was already connected. Execute
          callback.
      callback();
    }
  }

  function serversDisconnect() {
    if (servers !== null) { // If it's not
        currently disconnected.
      app.log("Disconnecting from Sharemind.")
          ;
      servers.disconnect();
      delete servers;
      servers = null;
    }
  }

  app.vote = function (electionId, voterId,
      positionId, candidateId, callback) {

    serversConnect(function (err_msg) {
```

```
  if (err_msg) {
    // Trigger callback:
    if (callback) callback(err_msg);
    return;
  }

  var pub_value_e = new pub.Uint32Array(
       electionId.length);
  for(var i=0; i<electionId.length; i++){
    pub_value_e.set(i, new BigInteger(''+
         electionId[i], 10));
  }
  var private_value_e = new priv.Uint32Array
       (pub_value_e);

  var pub_value_v = new pub.Uint32Array(1);
  pub_value_v.set(0, new BigInteger(''+
       voterId, 10))
  var private_value_v = new priv.Uint32Array
       (pub_value_v);

  var pub_value_p = new pub.Uint32Array(
       positionId.length);
  for(var i=0; i<positionId.length; i++){
    pub_value_p.set(i, new BigInteger(''+
         positionId[i], 10));
  }
  var private_value_p = new priv.Uint32Array
       (pub_value_p);

  var pub_value_c = new pub.Uint32Array(
       candidateId.length);
  for(var i=0; i<candidateId.length; i++){
    pub_value_c.set(i, new BigInteger(''+
         candidateId[i], 10));
  }
  var private_value_c = new priv.Uint32Array
       (pub_value_c);

  var args = {};
  args["proxyParams"] = {"codefile": "
       app_save_vote.sb"}
  args["smParams"] = {"electionId":
       private_value_e, "voterId":
       private_value_v, "candidateId":
       private_value_c, "positionId":
       private_value_p};

  app.log("Saving vote.");

  servers.emit("run_code",
    args,
    function (data) {
        app.log("Successfully saved vote.");

        $(window).trigger("allDone");

        // Trigger callback:
        if (callback) callback('ok',data);
    });
  });
};

app.getHashes = function (electionId,
     callback) {

  serversConnect(function (err_msg) {

  if (err_msg) {
    // Trigger callback:
    if (callback) callback(err_msg);
    return;
  }

  // Add parameters:
  var pub_value_e = new pub.Uint32Array(1);
  pub_value_e.set(0, new BigInteger(''+
       electionId, 10));

  var args = {};

  args["proxyParams"] = {"codefile": "
       app_get_hashes.sb"}
  args["smParams"] = {"electionId":
       pub_value_e};

  app.log("Getting hashes.");

  servers.emit("run_code",
    args,
    function (data) {
      app.log("Successfully retrieved
           results.");
      console.debug(data);

      $(window).trigger("allDone");

      // Trigger callback:
      if (callback) callback('ok', data);
      });
    });
};

app.getResults = function (electionId,
     positionId, candidateId, callback) {

  serversConnect(function (err_msg) {

  if (err_msg) {
    // Trigger callback:
    if (callback) callback(err_msg);
    return;
  }

  // Add parameters:
  var pub_value_e = new pub.Uint32Array(1);
  pub_value_e.set(0, new BigInteger(''+
       electionId, 10))

  var pub_value_p = new pub.Uint32Array(
       positionId.length);
  for(var i=0; i<positionId.length; i++){
    pub_value_p.set(i, new BigInteger(''+
         positionId[i], 10));
  }

  var pub_value_c = new pub.Uint32Array(
       candidateId.length);
  for(var i=0; i<candidateId.length; i++){
    pub_value_c.set(i, new BigInteger(''+
         candidateId[i], 10));
  }

  var args = {};

  args["proxyParams"] = {"codefile": "
       app_get_histogram.sb"}
  args["smParams"] = {"electionId":
       pub_value_e, "positionId":
       pub_value_p, "candidateId":
       pub_value_c};

  app.log("Computing results.");

  servers.emit("run_code",
    args,
    function (data) {
      app.log("Successfully retrieved
           results from servers.");
      console.debug(data);

      $(window).trigger("allDone");

      // Trigger callback:
      if (callback) callback('ok', data);
    });
  });
};

app.deleteElection = function (electionId,
     callback) {
  app.log('Deleting election');

  serversConnect(function (err_msg) {

  if (err_msg) {
    // Trigger callback:
    if (callback) callback(err_msg);
    return;
  }

  // Add parameters:
  var pub_value = new pub.Uint32Array(1);
  pub_value.set(0, new BigInteger(''+
       electionId, 10))

  var args = {};

  args["proxyParams"] = {"codefile": "
       app_delete_table_vm.sb"}
  args["smParams"] = {"electionId":
       pub_value};

  app.log("Deleting table.");

  servers.emit("run_code",
    args,
    function (data) {
      app.log("Successfully deleted table.")
           ;

      $(window).trigger("allDone");

      // Trigger callback:
      if (callback) callback();
    });
  });
};
```

```
  app.debug = function(txt) {
    console.debug(”[DEBUG]| ” + txt);
  };

  app.log = function(txt) {
    console.log(”[INFO] | ” + txt);
  };

  app.error = function(txt) {
    bootbox.dialog({
      message: txt,
      title: ”<span class=’glyphicon glyphicon
          −alert’></span> Communication Error
          ”,
      buttons: {
        main: {
          label: ”Ok”,
          className: ”btn−default”
        }
      }
    });
    $(’body’).css({’cursor’:’default’});
    console.error(”[ERROR]| ” + txt);
  };

})(this.app = this.app === undefined ? {} :
    this.app, sm, jQuery);
```

# XI. Acknowledgement

First of all, to my heavenly Father, Friend and Guide, I do not know what I did to deserve Your grace, thank you, without You I am nothing. Soli Deo gloria.

I would like to thank my family: Daddy, Mommy, Ading Kari and Manong Danzen. To my Daddy, thank you for all the advice and support you give me. To Mommy, thank you going out of your way to help me, for doing the things I should have done, thank you for your sacrifice. To Manong, my big bro, thank you so much for being there and for the empathy. To Ading, thank you for the heartwarming hugs and detoxifying moments. Without the support of all of you I never would have made it. You have always been there. Thank you for your love. To my grandparents: Lolo Sim and Lola Tina. Thank you so much for taking care of me. Sa paghahatid at pagsusundo. Sa pagbibigay po ng baon. Hehe. Maraming salamat po.

To my hardworking adviser, Sir Bryann, you know very well that we won't be able to graduate on time if it weren't for your weekly consultations. We know that you too have your fair share of things to take care of, but you have always found time for us. Thank you sir. To my co-advisees: Marvin and Aky. Grabe, di ko akalaing gagraduate tayo lahat on time! Congrats sa atin!

To sir Bogdanov and sir Rebane, without your generosity we would have been able to finish our thesis. Thank you for sharing Sharemind and your wisdom with us!

To my homie, brother from another mother, and bestfriend, Joseph, grabe, we made it. Thanks for always being there. Love you bro ;) To the optimist Monique, thank you for your encouragement and corny jokes. They help more than you know :) And thank you for the prayers! To ate Frances, thank you for being there to listen to my problems, for the care and understanding. Salamat po ate :p Nurse ka nga talaga ate. Haha. To my BNSEC family, ate Mathy and Christian, maraming salamat po! For the prayers, encouragement and kind words. Thank you so much. You peeps give me hope :)

To Bella, salamat sa huug that made me not cry. You're one hellufa friend. Thenks. To Krizia, aka homewrecker, hahaha. Thanks for all the fun! Pakabait ka po. To Eva, thanks for all the favors bro, I owe ya ;) To Earl, thanks for everything and for being such a good support! LoL. Godspeed! To Brendan, salamat sa pagtuturo samin sa ibang subjects lalo na sa 150. Ikaw na ang true master Hacker. Thank you! To Mark, the brainiest batchmate I know, thank you for being such an awesome friend :) To Alee, how to be you po. Thanks for patiently answering all my academic questions in life! To Reena, one of the most artistic people I've met, thank you for sharing your awesome talent with us. And for always being so encouraging :) To Christine, thanks for always staying classy and for helping in group works :) To Cristine, thank you sa pangugulit at pag"go Ruah" lagi haha. Thanks :) To Rain, salamat sa pagsama sakin sa mga lakad sa Accenture! And for the cool name!

To the CS Batch 2016, everyone, thank you for being one heck of a group. Thank you for the allowing me to experience fun and hard times with you. You peeps inspire me.

Laus Deo Semper.