CHAPTER ONE

INTRODUCTION

1.1 Background of Study

Election is the formal process of selecting a person for public office or of accepting or rejecting a political proposition by voting. It is important to distinguish between the form and the substance of elections. In some cases, electoral forms are present but the substance of an election is missing, as when voters do not have a free and genuine choice between at least two alternatives. Most countries hold elections in at least the formal sense, but in many of them the elections are not competitive (e.g., all but one party may be forbidden to contest) or the electoral situation is in other respects highly compromised (Britannica encyclopedia: Paul David Webb, Roger Gubbins, Heinz Eular, 1999).

Although elections were used in ancient Athens, in Rome, and in the selection of popes and Holy Roman emperors, the origins of elections in the contemporary world lie in the gradual emergence of representative government in Europe and North America beginning in the 17th century. At that time, the holistic notion of representation characteristic of the Middle Ages was transformed into a more individualistic conception, one that made the individual the critical unit to be counted. For example, the British Parliament was no longer seen as representing estates, corporations, and vested interests but was rather perceived as standing for actual human beings. The movement abolishing the so-called “rotten boroughs”—electoral districts of small population controlled by a single person or family—that culminated in the Reform Act of 1832 (one of three major Reform Bills in the 19th century in Britain that expanded the size of the electorate) was a direct consequence of this individualistic conception of representation. Once governments were believed to derive their powers from the consent of the governed and expected to seek that consent regularly, it remained to decide precisely who was to be included among the governed whose consent was necessary. Advocates of full democracy favoured the establishment of universal adult suffrage. Across western Europe and North America, adult male suffrage was ensured almost everywhere by 1920, though woman suffrage was not established until somewhat later (e.g., 1928 in Britain, 1944 in France, 1949 in Belgium, and 1971 in Switzerland). (Britannica encyclopedia: Paul David Webb,Roger Gubbins,Heinz Eular, 1999).

Although it is common to equate representative government and elections with democracy, and although competitive elections under universal suffrage are one of democracy's defining characteristics, universal suffrage is not a necessary condition of competitive electoral politics. An electorate may be limited by formal legal requirements—as was the case before universal adult suffrage, or it may be limited by the failure of citizens to exercise their right to vote. In many countries with free elections, large numbers of citizens do not cast ballots. For example, in Switzerland and the United States, fewer than half the electorate votes in most elections. Although legal or self-imposed exclusion can dramatically affect public policy and even undermine the legitimacy of a government, it does not preclude decision making by election, provided that voters are given genuine alternatives among which to choose. (Britannica encyclopedia: Paul David Webb,Roger Gubbins,Heinz Eular, 1999).

During the 18th century, access to the political arena depended largely on membership in an aristocracy, and participation in elections was regulated mainly by local customs and arrangements. Although both the American and French revolutions declared every citizen formally equal to every other, the vote remained an instrument of political power possessed by very few.

Even with the implementation of universal suffrage, the ideal of “one person, one vote” was not achieved in all countries. Systems of plural voting were maintained in some countries, giving certain social groups an electoral advantage. For example, in the United Kingdom, university graduates and owners of businesses in constituencies other than those in which they lived could cast more than one ballot until 1948. Before World War I, both Austria and Prussia had three classes of weighted votes that effectively kept electoral power in the hands of the upper social strata. Until the passage of the Voting Rights Act in 1965 in the United States, legal barriers and intimidation effectively barred most African Americans—especially those in the South—from being able to cast ballots in elections.

During the 19th and 20th centuries, the increased use of competitive mass elections in Western Europe had the purpose and effect of institutionalizing the diversity that had existed in the countries of that region. However, mass elections had quite different purposes and consequences under the one-party communist regimes of Eastern Europe and the Soviet Union during the period from the end of World War II to 1989–90. Indeed, elections in these countries were similar to the 19th-century Napoleonic plebiscites, which were intended to demonstrate the unity rather than the diversity of the people. Dissent in Eastern Europe could be registered by crossing out the name of the candidate on the ballot, as several million citizens in the Soviet Union did in each election before 1989; however, because secret voting did not exist in these countries, this practice invited reprisals. Nonvoting was another form of protest, especially as local communist activists were under extreme pressure to achieve nearly a 100 percent turnout. Not all elections in Eastern Europe followed the Soviet model. For example, in Poland more names appeared on the ballot than there were offices to fill, and some degree of electoral choice was thus provided.

In sub-Saharan Africa, competitive elections based on universal suffrage were introduced in three distinct periods. In the 1950s and ’60s, a number of countries held elections following decolonization. Although many of them reverted to authoritarian forms of rule, there were exceptions (e.g., Botswana and Gambia). In the late 1970s, elections were introduced in a smaller number of countries when some military dictatorships were dissolved (e.g., in Ghana and Nigeria) and other countries in Southern Africa underwent decolonization (e.g., Angola, Mozambique, and Zimbabwe). Beginning in the early 1990s, the end of the Cold War and the reduction of military and economic aid from developed countries brought about democratization and competitive elections in more than a dozen African countries, including Benin, Mali, South Africa, and Zambia.

Competitive elections in Latin America also were introduced in phases. In the century after 1828, for example, elections were held in Argentina, Chile, Colombia, and Uruguay.

Additional countries held elections in the period dating roughly 1943 to 1962. Beginning in the mid-1970s, competitive elections were introduced gradually throughout most of Latin America.

In Asia, competitive elections were held following the end of World War II, in many cases as a result of decolonization (e.g., India, Indonesia, Malaysia, and the Philippines), though once again the restoration of authoritarianism was commonplace. Beginning in the 1970s, competitive elections were reintroduced in a number of countries, including the Philippines and South Korea. With the exception of Turkey and Israel, competitive elections in the Middle East are rare.

Authoritarian regimes often have used elections as a way to achieve a degree of popular legitimacy. Dictatorships may hold elections in cases where no substantive opposition is remotely feasible (e.g., because opposition forces have been repressed) or when economic factors favour the regime. Even when opposition parties are allowed to participate, they may face intimidation by the government and its allies, which thereby precludes the effective mobilization of potential supporters. In other cases, a regime may postpone an election if there is a significant chance that it will lose. In addition, it has been a common practice of authoritarian regimes to intervene once balloting has begun by intimidating voters (e.g., through physical attacks) and by manipulating the count of votes that have been freely cast.

1.2 Functions of Election

Elections make a fundamental contribution to democratic governance. Because direct democracy—a form of government in which political decisions are made directly by the entire body of qualified citizens—is impractical in most modern societies, democratic government must be conducted through representatives. Elections enable voters to select leaders and to hold them accountable for their performance in office. Accountability can be undermined when elected leaders do not care whether they are reelected or when; for historical or other reasons, one party or coalition is so dominant that there is effectively no choice for voters among alternative candidates, parties, or policies. Nevertheless, the possibility of controlling leaders by requiring them to submit to regular and periodic elections helps to solve the problem of succession in leadership and thus contributes to the continuation of democracy. Moreover, where the electoral process is competitive and forces candidates or parties to expose their records and future intentions to popular scrutiny, elections serve as forums for the discussion of public issues and facilitate the expression of public opinion. Elections thus provide political education for citizens and ensure the responsiveness of democratic governments to the will of the people.

They also serve to legitimize the acts of those who wield power, a function that is performed to some extent even by elections that are noncompetitive.

Elections also reinforce the stability and legitimacy of the political community. Like national holidays commemorating common experiences, elections link citizens to each other and thereby confirm the viability of the polity. As a result, elections help to facilitate social and political integration.

Finally, elections serve a self-actualizing purpose by confirming the worth and dignity of individual citizens as human beings. Whatever other needs voters may have, participation in an election serves to reinforce their self-esteem and self-respect. Voting gives people an opportunity to have their say and, through expressing partisanship, to satisfy their need to feel a sense of belonging. Even nonvoting satisfies the need of some people to express their alienation from the political community. For precisely these reasons, the long battle for the right to vote and the demand for equality in electoral participation can be viewed as the manifestation of a profound human craving for personal fulfillment.

Whether held under authoritarian or democratic regimes, elections have a ritualistic aspect. Elections and the campaigns preceding them are dramatic events that are accompanied by rallies, banners, posters, buttons, headlines, and television coverage, all of which call attention to the importance of participation in the event. Candidates, political parties, and interest groups representing diverse objectives invoke the symbols of nationalism or patriotism, reform or revolution, past glory or future promise. Whatever the peculiar national, regional, or local variations, elections are events that, by arousing emotions and channeling them toward collective symbols, break the monotony of daily life and focus attention on the common fate.

1.3 Statement of the Problem

Elections have always been a controversial process and the Nigerian Universities campuses elections are not exempted from these election difficulties. The problem of credibility is a worrisome issue that most students who would normally vote on elections' days would rather stay at their hostels than to go through with the stress attached with elections.

However, this system would eradicate these issues. The problem of credibility would be cancelled since the system would be transparent and none of the parties would have access to manipulate the results. It will also take away the problem of stress since students can vote from anywhere right from their mobile phone thereby motivating many students to decide who they want as their representatives.

1.4 E-Voting System

Online voting is an electronic way of choosing leaders via a web driven application E-voting (also known as electronic voting or EVM) refers to voting using electronic means to either aid or take care of the chores of casting and counting votes.

Depending on the particular implementation, e-voting may use stand-alone electronic machines (also called EVM) or computers connected to the internet. It encompasses a range of Internet services, from basic data transmission to full-function online voting through common connectable household devices. Similarly, the degree of automation may vary from simple chores to a complete solution that includes voter registration & authentication, vote input, local or precinct tallying, vote data encryption and transmission to servers, vote consolidation and tabulation, and election administration. A worthy e-voting system must perform most of these tasks while complying with a set of standards established by regulatory bodies, and must also be capable to deal successfully with strong requirements associated with security, accuracy, integrity, swiftness, privacy, auditability, accessibility, cost-effectiveness, scalability and ecological sustainability.

In general, two main types of E-voting can be identified:

1. E -voting which is physically supervised by representatives of governmental or independent electoral authorities (e.g. electronic voting machines located at polling stations)
2. Remote E-voting via the internet where the voter votes at home or without going to a polling station.

The Online voting system also known as E-voting is a term encompassing several different types of voting embracing both electronic means of counting votes. Electronic voting technology can include punched cards, optical scan voting systems and specialized voting kiosks (including self-contained direct-recording electronic voting systems or DRE). It can also involve transmission of ballots and votes via telephones, private computer networks, or the internet. The advantage of online voting over the common “queue method” is that the voters have the choice of voting at their own free time of the speculated voting period and there is reduced congestion. It also minimizes on errors of vote counting. The individual votes are submitted in a database which can be queried to find out who of the aspirants for a given post has the highest number of votes. This system will gear the voter to turnout has they know there vote is secured and they themselves are safe.

With this system in place also, cases of false votes shall be reduced. With the “E-VOTING SYSTEM”, a voter can use his\her voting right online without any difficulty. He\She has to register as a voter first before being authorized to vote. The registration should be done prior to the voting date to enable data update in the database. However, not just anybody can vote. For one to participate in the elections, he/she must have the requirements. For instance, he/she must be a registered student i.e. must be a student of the school/department. As already stated, the project ‘E-voting' provides means for fast and convenient voting and access to this system is limited only to registered voters. Putting a lot of factors into perspective, E-voting will solve a lot of the problems associated with elections. For one, voting in the confines of one’s home improves credibility by encouraging personal decisions, that is, decision not affected by political cohorts at the polling booths. Another is the issue of security; it is quite often for some electorates to get violent at polling booths if things are not going their way. There is also the situation of stress, queuing for several hours under the sun waiting to vote. And of course accessibility, not everyone can access the voting locations on election days but with the E-voting system, all the aforementioned will be eradicated or reduced to the barest minimum.

1.5 Aims and Objectives of Study

Aims

The aim of this project is to develop an E-voting system that will aid in credible, transparent and stress-free elections where voters would be able to vote from anywhere.

OBJECTIVES

The objectives of the E-voting system are listed below:

1. To takes details of candidates of an election and voters, allow voters to vote from anywhere using their mobile phone and collates the votes to give credible results.

2. To evaluate the performance of the system in order to determine its effectiveness and improve it further if there is need to.

1.6 Scope of Study

The E-voting system will allow election conductors to input the records of election candidates and voters from their administrative backend and allow accredited voters to vote. Accredited voters will be able to vote by supplying their registered details. The system will also provide statistical results based on voters’ inputs and voters’ behaviors.

CHAPTER TWO

LITERATURE REVIEW

Electronic voting system is a type of voting system which is done through electronic systems. However, electronic systems have spanned beyond just the above. In recent times, electronic voting systems now leverages on the web and other mobile related technologies that rely on the internet.

There are two types of Electronic voting systems which can be identified as: **e-voting** which is supervised physically by independent electoral authorities or governmental representatives like the machines at polling stations and Remote electronic voting is where the vote is not physically supervised by government or independent representatives like voting from a personal computer, mobile phone or television via the internet also known as **i-voting**.

A Voting machine is the combination of electromechanical, mechanical or electronic equipment which includes its software, firmware and the required documentation to program control and support equipment which is used to count and cast votes, defined ballots, to display or report election result and to produce and maintain audit trail information. This machine is able to provide the voter immediate feedback such possible problem as over voting or under voting which will result in a spoiled ballot. This machine has different levels of security, usability, accuracy and efficiency. The first voting machine made was mechanical. Now the most common machine use is electronic. Certain machines may be more or less accessible for voters. The Chartists were a political and social reform movement in the United Kingdom. They were the ones who had made the first proposal for the usage of voting machines in 1838.

Nowadays Electronic voting system has become more popular around the world. Some of the countries which uses electronic and vote on line are: United States, Brazil, Australia, Canada, Belgium, Germany, Romania, France, Venezuela, Philippines, The European Union, Switzerland, Italy, Norway, Romania and United Kingdom. Electronic voting system is very accessible for individuals with disabilities. They have the ability to use joysticks, earphones, Sip and puff technology, foot pedals, etc. This machine has touchscreens which can display the information in several languages and voting choices in audio for visual impaired voters. This feature makes voting easier and comfortable for people with disabilities. The literature tried to list all e-voting system types and channels to make it easier to understand this important application. E-voting can be conducted through two major channels: the first using electronic systems installed in public centers for elections; the second channel using the Internet. The e-voting is to be more flexible and convenient to citizens' needs. On the other hand, when looking into the history of voting systems, the major e-voting system types are the following:

2.1.1 Punch Card Voting System

With punch-card voting systems, the ballot is a card (or cards) and the voters punch holes in it (with a supplied punch device) next to their candidate or choice. After punching the hole(s), the voter may place the ballot in a ballot box, or the voter may feed the ballot into an electronic vote tabulating device at the voting place.

Two common types of punch-card voting systems are the "*Votomatic*" and the "*Datavote*" system. With the Votomatic card, the locations at which holes may be punched to indicate votes are each assigned numbers. The number of the hole is the only information printed on the card. The list of candidates or ballot issue choices and directions for punching the corresponding holes are printed in a separate booklet. With the Datavote card, the name of the candidate or description of the choice is printed on the ballot next to the location of the hole to be punched. The re-count of ballots in Florida during the 2000 presidential election created a debate about the reliability of punch-card voting systems. After 2000, the popularity of punch-card voting systems in the US decreased significantly.

2.1.2 Optical Scan Voting System

These systems use an optical scanner to read and count marked ballot papers. Various systems can be defined as optical scan voting systems including:

i. *Marksense systems* whereby an optical mark (e.g. made with a graphite pencil on the ballot paper) can be recognized by a scanner.

ii. *Electronic ballot markers (EBM)* that can be used to fill out optical scan ballots. The systems look like traditional DREs, but they record votes on paper ballots instead of internal memory. EBM can aid a disabled voter in marking a paper ballot; it can allow for audio interfaces.

iii. *Digital pen:* these systems use ballots on digital paper. A small camera in the pen is able to recognize where the voter marks the digital ballot paper. The ballots are collected in the polling station and the digital pen has to be returned to the elections staff for tabulation.

Optical scan voting systems combine paper with electronic devices. All the systems keep a tangible ballot paper which serves as a tangible record of the voter´s intent. By that, optical scan systems allow for manual recounts of ballots. The big advantage is that the counting process can be done in a central place and that the counting is much faster. The system is easily understandable by the voter: for him/her it doesn´t really change much; they can still mark their preference on a ballot paper. And if – for whatever reason – the scanning system fails to work, ballots can be counted manually.

1. Direct-Recording Electronic (DRE) Voting Machine

With a DRE machine, voting can be done on Election Day or it can be used as an advance voting device in polling stations. It is easily understandable: the voter just pushes a button next to his/her favourite candidate or choice. Or the DRE machines have a touch screen displaying the ballot. After the election or referendum, the DRE machine produces a tabulation of the voting data stored in a removable memory component and/or as printed copy. The system may also allow for transmission of individual ballots or vote totals to a central location. The result can then be consolidated in one central place.

DRE voting machines started to be massively used in 1996 in Brazil. They were also used on a large scale in the US after the Florida 2000 experience. Vision-impaired voters benefit from DRE machines because they can cast their vote without help from another person. DRE machines were also deployed in Europe, e.g. in the Netherlands, where the company NEDAP provided their own DRE machines since 1989. They were used in the Netherlands until 2006. In 2009, the German Constitutional Court found that the DRE-type voting machines used in parliamentary elections in Germany were unconstitutional since they did not allow citizens to examine the determination of the result.

2. Voter-Verified Paper Trail Audit

A voter-verified paper audit trail (VVPAT) or verified paper record (VPR) is not an e-voting system itself, but refers to a component that can be combined with various forms of non-document ballot voting systems. VVPAT means that a paper ballot for each vote is printed by the electronic device that was used to cast the vote. A VVPAT is intended as an independent verification system for voting machines designed to allow voters to verify that their vote was cast correctly, to detect possible election fraud or malfunction, and to provide a means to audit the stored electronic results.

3. Internet Voting System

Internet voting refers to the use of the Internet to cast and/or transmit the vote. Internet voting can take various forms depending on whether it is used in uncontrolled environments (remote Internet voting) or not (Polling Site Internet Voting, Kiosk Voting). With remote Internet voting neither the client machines nor the physical environment are under the control of election officials. Voters can cast their vote at practically any place (at home, at the workplace, at public Internet terminals etc.). The vote is then transmitted over the Internet. This method offers the most advantages to voters, but at the same time it suffers from them most security concerns. They include doubts about the Internet as a means of transmission of confidential information, fear of hacker attacks and anxiety about the possibility of undue influence being exerted on the voter during the voting process (e.g. ‘family voting’).

The other options (polling site Internet voting or kiosk voting) refer to systems where voters cast their ballot from client machines that are physically situated in official polling stations or in public places that are controlled by election officials. In both cases, hardware and software components are controlled by election officials. The difference is that with polling site Internet voting the authentication of the voters may take place by traditional means and with kiosk voting (in public places), the physical environment and voter authentication are not directly under control of election officials. Examples of Internet voting systems include web-based voting systems and Android based voting systems.

E-voting systems' legitimacy is guaranteed through few measures that are summarized from many resources (Anane, 2007; Sandikkaya & Orencik, 2006; Buchsbaum, 2005; Maier, 2010; Kahani, 2005; Smith, 2002; Patil, 2010; Yumeng, Liye, Fanbao & Chong, 2011; Jafari, Karimpour & Bagheri, 2011); following a description of each one:

Democracy*:* E-voting system should guarantee to all legitimate voters the chance to cast their votes on election's day. Such condition is important for the democratic process, where the convenience of election procedures for voters, and especially people with special needs and handicapped persons. Also, it is important to accommodate people with low computer literacy. It is important to guarantee the wide use of electronic system before dropping the traditional system. As seen in the case of Philippines, the government provided all types of backup methods, and especially paper and traditional forms for people who would not or cannot use e-voting machines. Also, it is important at first stages to deploy both methods (traditional and electronic) to gain the critical mass and guarantee the success of e-voting (Bhatnagar, 2004).

Eligibility: Only legitimate voters who are registered and satisfy election conditions can vote (Anane, 2007). Also, this rule controls for repetitive voting.

Accuracy*:* accuracy is not a luxurious condition; it is an essential condition for auditing the total votes against the voters cards used (compared with traditional voting). It is one of the major characteristics of e-voting systems when compared with traditional paper forms. The system should provide rules for checking the total number of casted votes and the total number of voters.

Verifiability: the system should be verifiable against total number of votes casted and the totals of each nominee. All analysis should also be verifiable to guarantee the integrity and image of elections.

Integrity: Votes are not to be changed, forged, or omitted. Also, the system should provide needed means to audit and discover any attempts to tamper in the system. Also, Langer *et al.* (2010) emphasized the fact that voters need to be assured that their votes will not be changed and omitted. Such issue is important so citizens will not lose faith in e-government and electronic systems.

Transparency*:* the voting procedures should be transparent and clear to all parties. All rules and procedures need to be announced and explained to all stakeholders. Also, it is important to train people and conduct pilot elections on voting procedures for two major reasons: to guarantee the needed skills, and second to raise awareness and gain trust.

Privacy: It is essential not to be able to relate voters to their choices (votes). Also, whether it is registered or not, it is important not to reveal such choices to any party. Another aspect of privacy is to prevent other voters from overseeing voters' choices in the election process.

Uniqueness: Some election systems assign one vote for each voter. Other systems assign more. Based on that, it is important to control for the number of choices allowed and the totals resulting from the process when guarding for accuracy and verifiability.

Receipt-free voting: where voters are not supposed to know the results before the end of elections so that voter's choices will not be affected by results.

Flexibility: The e-voting system need to be designed to provide flexibility for voters, with alternative options (electronic vs. paper; compatible systems to all computers and browsers (especially for Internet voting) electrical vs. battery operated; using different languages; utilizing graphical interfaces for illiterate people; provide tools for handicapped).

Scalability: the system's ability to accommodate the change in number of voters, their characteristics and related information.

Gerlach (2009) indicated that keeping e-voting secure and confidential, as paper-based voting, seven design principles need to be incorporated:

1. Proven security: all protocols and techniques must be mathematically proven secure.
2. Trustworthy design responsibility: government security agencies should be responsible for creating secure voting system.
3. Source code must be published and made publicly accessible.
4. Vote verification: it should be possible to verify that all votes have been correctly accounted for in the final election tally.
5. Voters' accessibility: system should be accessible to all, and easy to use.
6. Ensure anonymization: techniques like *onion routing* must be used to ensure anonymization.
7. Expert oversight: team of experts selected and approved by all major parties taking part in election.

2.2 Authentication Methods

Implementing electronic voting systems is faced by several issues which can be the main reason for its success or failure; the most important among all focus on security and privacy especially how to make sure that the one who votes is the legitimate voter not anyone else. There are several security requirements that electronic voting system must satisfy such as: eligibility, authentication, privacy, robustness, and fairness. Eligibility guarantees that only eligible voters can participate and cast their votes during election period. Authentication makes sure that the one who votes is the right one and no one else. Voter's privacy enable’s voters to vote in a highly private way in which their personal information and voting process information are protected and can't be known by others. Robustness means that electronic voting system should be protected against any attacks, fraud and disruption. Finally, fairness in announcing voting results only at the end of allowable voting period (Patil, 2010; Yumeng, Liye, Fanbao & Chong, 2011; Jafari, Karimpour & Bagheri, 2011).

This section will focus on the authentication process, which makes sure that a user has access to the e-voting system and can participate in this e-service or not based on his original identity. Several authentication schemas and methods are available and can be classified into three main types; knowledge-based methods, token based methods and Biometrics based methods (Rao & Patil, 2011; Deep, 2011).

2.2.1 Knowledge-Based Methods

One of its widespread names is password-based methods, which is based on the person’s knowledge of something. An example of what a person knows is a password or a PIN number (Rao & Patil, 2011; Deep, 2011). In this system, only authorized users can access this system, also this system prevents the authorized users to vote or register more than one time through making a check process on the database, which will increase people's trust in E-voting system.

Using this simple system, depending on an ID and password, for authentication will weaken the security of e-voting systems against hackers especially that IDs and passwords can be easily forgotten, lost, attacked, known and used by others. Using additional methods, which will increase cost, will increase the security of ID and password and prevent unauthorized users from using the system (Rao & Patil, 2011). The complexity or simplicity of passwords can weaken or strengthen the security of the e-voting system; using simple passwords which will be easily remembered will weaken the security of the system because these passwords will be easily known by others, easily attacked and easily shared. Although using complex passwords will increase the system security, it makes remembering such passwords more difficult (Kumar *et al.*, 2011). Using complex IDs and passwords will increase the chances of typing errors and as a result rejection of the system, which makes system usage more irritating and inconvenient.

To overcome the low level of security associated with using IDs and passwords alone in authentication, Sodiya *et al.* (2011) presented an enhanced electronic voting architecture that uses usernames and PIN numbers that are generated randomly by a computer program during the registration phase in which people provide several personal information that will be used for authentication purpose during the voting period. Using a system that depends only on a user name and PIN number is not enough to guarantee that the one who votes is the same legitimate person supposed to be. On the other hand, more costly methods combined with username and PIN number will make the e-voting system more effective. Such methods can depend on using biometrics as fingerprint, iris or face recognition or other methods like token based.

Another enhanced and secure e-voting system enables citizens to vote from their personal computers, where their computers are connected to the Internet. In the identification phase of this system, the citizen will request a voting certificate (PIN) that makes such citizen authorized to access the voting system after saving his personal information in the government election server (this request is done by using the citizen national ID as a bases for such process), this PIN number will be used for authentication purposes in the future and in the e-voting system for which citizens will use it to have a public key that will enable them to vote (this public key will be received in a message from the e-voting authority using mobile phones after checking personal information and making sure they comply with what is saved on system database). This system enables citizens to vote from their personal computers without any additional cost or effort (Al-Anie *et al*., 2011).

There are several authentication schemas that are suitable to be used to enhance the security of remote voting which is one of the main types of electronic voting that enables citizens to vote from their Internet connected personal computer. Sahu and Choudhray (2011) introduced an electronic voting schema that integrates electronic voting system with GSM infrastructure which enables people to vote through their mobile terminals anywhere within the election period. Anyone who wants to vote using mobile terminals will have a unique mobile ID which was given by the election committee, this E-voting system "uses 89S52 microcontroller, which has dual serial communication facility, 16X2 dual line LCD for massage display, and line converter MAX 232, GSM modem" (Sahu & Choudhray, 2011, p.5644). This microcontroller will receive a message from GSM modem that contains voter's unique mobile ID and candidate ID, then it will make authentication process based on the received voter's mobile ID and the information stored on its database and then send a notification message to voter's mobile if the voting process done successfully or not. This E-voting system armed with this feature (voting through mobile terminals using unique mobile ID in authentication process) will make the voting process more secured and will increase the percentage of people participating in the e-voting activities. On the other hand, remote voting systems still need to deal with several security problems like the denial of service attacks which will weaken people's trust and usage of this remote electronic voting system.

2.2.2 Token-Based Methods

Using token based methods seems to be stronger and more effective than using knowledge based methods (Khan, 2010) especially that these methods of authentication are based on the person's knowledge particularly what the person has. (Rao & Patil, 2011; Deep, 2011; Khan, 2010). Even though using token based methods has several advantages that come from its ability to achieve more secure process than using passwords especially if token is stolen; still there is a need to know the user's password, which results in having two levels of security. On the other hand, thinking of tokens attacks issues and the high cost required for purchasing high quality token readers is needed (Rao & Patil, 2011).

2.2.3 Biometrics-Based Methods

Biometrics-based methods are based on the physical characteristics of the person that are different from one person to another (Rao & Patil, 2011; Deep, 2011). Using biometrics like face recognition, fingerprint, iris and vein has several advantages based on its accuracy, uniqueness and complexity especially that it can't be stolen, altered or used by someone else, so systems that use biometrics in authentication process will increase the system's level of security and encourage people to use it (Sarkar, Alisherov, Kim & Bhattacharyya , 2010). Also the performance of biometrics-based authentication methods can be evaluated based on two main types of errors: matching errors, which occur during the voting period; and acquisition errors, which occur during the preparation and registration stages (Deep, 2011).

Another proposed schema for biometric e-voting systems used citizen's fingerprint that is stored in the system database, when a fingerprint scanner scans the voter's fingerprint, the voter ID and other personal information will appear on the screen if the scanned fingerprint matched the fingerprint template stored in the database. The uniqueness feature of a person's fingerprint will increase the level of people's trust in this system (Altun & Bilgin, 2011). Using fingerprint in the authentication process has several drawbacks especially if the fingerprint templates that exist in the database are attacked or stolen. To avoid fingerprint templates attack, an urgent need for enhancing the security and authentication system that uses fingerprint is required; this enhancement can be done through the use of fragile image watermarking techniques that embed additional information to fingerprint image like an identification number that will be used with fingerprint in the authentication process (Gothwal *et al*., 2011). On the other hand, skin status (especially dry or wet skin and skin injuries) have a significant impact on the quality of scanning and matching process, which will increase the percentage of matching errors. Such issue will result in a system that will treat eligible voters as ineligible voters and prevent them from casting their votes and participating in the e-voting stage (Kumar & Walia, 2011b).

In general, using biometrics as fingerprint, iris and face recognition in the authentication process has several pros and cons based on the extent to which people accept them. The uniqueness and consistency of biometrics will result in a robust authentication process that will leverage the security levels of e-voting systems, and will compensate for the additional cost resulting from the need to purchase special scanners that need special maintenance and support (Rao & Patil, 2011).

Even though biometrics can be used as a unique identifier for any person, several risks are associated with using such technique in the authentication process. Risks come from many resources like: the ability to scan or capture anyone’s iris using hidden cameras; people fingerprints are left on any object that a person touches; if biometrics templates are attacked or stolen, a disaster will happen especially that these templates cannot be replaced by another new one. A person cannot change his fingerprint, iris or other biometric identifier, so a password can be replaced or updated, but physical features of a human body cannot (Deep, 2011; Zahed & Sakhi, 2011).

Roa and Patil (2011) introduced an enhanced authentication method utilizing a three dimensional environment that can be used in several systems for authentication purposes. The security and reliability of such 3-D environment are increased by giving users the option to choose an authentication method according to their preferences that blend a knowledge-based method, token-based method or a biometrics-based method; if users don't prefer to carry cards, they can choose a authentication method that they prefer as Id\passwords or biometrics as fingerprint or iris.

CHAPTER THREE

RESEARCH METHODOLOGY

In this chapter, the source of data methods of collection, the evaluation of the existing system and the organization structure of the system problem are presented. It includes specific methods which were used in order to achieve the objectives of the project, particular requirements for implementation of the project and a brief explanation of why such methods were used for implementing the proposed system, also included is a brief description of the current system of Overall System Design Objectives

System Implementation Technologies

The web-based Student Electronic Voting System (SEVS) was developed as an online information system to offer users convenient access to the voter register. Several tools used during implementation include the following:

Technology Overview

i. **MYSQL DBMS**-it allows combination, extraction, manipulation and organization of data in the voters' database. It is platform independent and therefore can be implemented and used across several such as Windows, Linux server and is compatible with various hardware mainframes. It is fast in performance, stable and provides business value at a low cost.

ii. **HTML -Hypertext Markup Language**-This is currently the core of the web world, it is a language used to makeup web page. It is the glue that holds everything together. Although HTLM was used for the implementation of the OVS, it is highly compatible with extensible HTML (XHTML) which is designed to be a replacement of HTML made to handle data and is also portable between different browsers and platforms with little or no alterations in code. Brackets is a prefer tool for designing HTML pages and that is the tool used in coming up with this OVS system.

iii. **PHP-** PHP is a general-purpose scripting language that is especially suited to server side web development where PHP generally runs on a web server. PHP code is embedded into the HTML source document. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content. It can also be used for command-line scripting and client-side GUI applications. PHP can be deployed on many web servers and operating systems, and can be used with many relational database management systems (RDBMS). It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

iv. **Apache-**The Apache HTTP Server is a web server software notable for playing a key role in the initial growth of the World Wide Web. In 2009 it became the first web server software to surpass the 100 million web site milestone. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Since April 1996 Apache has been the most popular HTTP server software in use. As of November 2010 Apache served over 59.36% of all websites and over 66.56% of the first one million busiest websites.

v. **WAMP-**WAMP is a small and light Apache distribution containing the most common web development technologies in a single package. Its contents, small size, and portability make it the ideal tool for students developing and testing applications in PHP and MySQL. WAMP is available as a free download in two specific packages: full and lite. While the full package download provides a wide array of development tools, WAMPP Lite contains the necessary technologies that meet the Ontario Skills Competition standards. The light version is a small package containing Apache HTTP Server, PHP, MySQL, phpMyAdmin, Openssl, and SQLite.

vi. **Obtaining and Installing WAMP-**As previously mentioned, WAMP is a free package available for download and use for various web development tasks. All WAMP packages and add-ons are distributed through the Apache Friends website at the address: http://www.apachefriends.org/. Once on the website, navigate and find the Windows version of WAMPP and download the self-extracting ZIP archive. After downloading the archive, run and extract its contents into the root path of a hard disk or USB drive. For example, the extract path for a local Windows installation would simply be C:\. If extracted properly we will notice a new wamp directory in the root of your installation disk. In order to test that everything has been installed correctly, first start the Apache HTTP Server by navigating to the wamp directory and clicking on the apache_start.bat batch file.

Next we will test if the server is running correctly by opening an internet browser and typing http://localhost/ into the address bar. If configured correctly, we will be presented with a screen similar to that of the one below. WAMP splash screen. In order to stop all Apache processes we do not close the running terminal application, but instead run another batch file in the wampplite directory called apache_stop.bat.

vii. **Creating a Database and Inserting Data-**Now that we have run and tested Apache and PHP, the next step is running MySQL and creating a database and table which will hold information to be used by our website. In order to start MySQL, navigate to the wamp directory and run the mysql_start.bat batch file. The WAMP package contains an application called phpMyAdmin which allows developers to administer and maintain MySQL databases. We will be using phpMyAdmin to create a database and table, and enter test data. Before testing phpMyAdmin, make sure that both Apache and MySQL are running by opening their respective batch files: apache_start.bat and mysql_start.bat. Along with Apache and MySQL running in the background, we type http://localhost/phpMyAdmin/ into our web browser. If successful we will be presented with a phpMyAdmin start page similar to the one shown below. phpMyAdmin start page

The first step with phpMyAdmin running is creating a new database. We create a new database by directly executing SQL statements as shown below. The successful execution of the sql querry creates a database 'student' with two tables in it. The tables are admin_login and student_identification. We also inserted values in the admin table. The screenshot below shows the successful execution of the query thus creation of a database named student. Creation of database in mysql using phpMyadmin, thus we have learned to create a database in MYSQL by executing sql statements. After creating the database and tables we are now ready to use them in our website "STUDENT ELECTRONIC VOTING SYSTEM".

PROJECT DESCRIPTION

Design of Student Electronic Voting System can be used by education institutes to maintain a proper free and peaceful election in the institution. Achieving this objective is difficult using the old system of voting by which rigging of the election is most unavoided and also most of the election result to violence, collection of result might not be genuine. All these problems are solved using this project. The directory structure of the project is as follows:

Description of root directory contents

- **User_Images Directory:** This directory contains the images uploaded by the students during registration process. Supported formats are the .jpg and .gif files.
- **JS Directory:** This Directory contains all JavaScript.
- **Include Directory:** This Directory contains all configuration files.
- **Fonts Directory:** This Directory contains all font use for the development of the project.
- **Database Directory:** This Directory contains database backup.
- **CSS Directory:** This Directory contains all CSS file.
- **Activepassword.php**: Page for password connectivity. It is used whenever password values are required to access a page.
- **Activepasswordck.php**: Page for password connectivity check. It is used whenever Activepassword.php is passing values are required to access a page.
- **Change-password.php**: Page for changing password. It is used whenever user want to change a password values.
- **Change-passwordck.php**: Page responsible for password change check. It is used whenever user want to change their password.
- **Edit_Student.php**: Admin page for editing information of a student. The administrator can change details of a student in this page. Though facility of changing the image and resume are not yet provided but will be provided in future versions of the project.
- **Admin_Header.php** : Header file for pages accessible to administrator only
- **Admin_Home.php** : Home page for administrator after logging in process.

- **Admin_Login.php** : Login page for administrator access. Shows appropriate message for wrong matric no and/or password.
- **Admin_Login_handler.php** : Page handler for Admin_Login.php page. It checks the values provided with the values in the database.
- **config.php** : Page for database connectivity. It is used whenever database values are required on the page.
- **Manage_Student.php**: Page to display all registered students to the administrator. This facility is only available to the administrator.
- **leftbar.php** : left file for all pages.
- **topbar.php** : top file for login page and homepage of the site.
- **Index.php** : Homepage of the website.
- **Logout.php** : Logout handler page. It Destroys all session variables thus ending user session.
- **Search.php** : Search page to search students. It can only be used by administrator. Students can be searched using different fields such as user id, account status etc.
- **Search_Result.php** : Page to display search results to the administrator.
- **Search_View_Result.php** : Page to display student information for any selected search result.
- **Student_Edit.php** : Page to edit student information.It can be accessed by students.
- **Student_Edit_Handler.php** : Page handler for Student_Edit.php.
- **Student_Header.php** : Header file for student pages.
- **Student_Home.php** : Home page for students after they log into their respective accounts.

- **Student_login.php** : Login Page for student login. Appropriate message is displayed if the login is unsuccessful.
- **Student_login_handler.php** : Page handler for Student_Login.php. It check the values provided with that in the database.
- **Student_Registration.php** : Student Registration page. The student enters various details here for registration.
- **Student_Registration_handler.php** : Page Handler for handling the file Student registration. php.It adds value to the student information table thus creating a new user.
- **Student_Reset_Password.php** : Page for resetting password. It can be used only by students. Administrator password can be changed only by changing the values in the table directly. **_Reset_Password_Handler.php** : Page handler for handling page. Student_Reset_ Password.php.
- **Student_View.php** : Page to display student profile with all the details of the student.
- **Style.css** : Stylesheet for the whole site design.
- **Validation.js** : Javascript validations used for validation of form values. Various form entries are validated at the client side using this file only.

Description of database tables

- **admin_login :**
  - user_id : Stores user id of administrator(s).
  - password : Stores password of the administrator(s).
  - last_login_date : Stores the last login date of the administrator(s).
- **Student_information :**
  - student_id : Stores user id of the student(s)

- student_password : Stores password of the student(s)
- first_name : Stores first name of the student(s)
- last_name : Stores last name of the student(s)
- registration_date : Stores the registration date of the student(s).
- gender : Stores the gender of the student(s).
- date_of_birth : Stores the date of birth of the student(s).
- student_status : Stores the current status of the student account(s).
- contact_no : Stores the contact number of the student(s).
- qualification : Stores student(s) qualification.
- city : Stores the city in which the student(s) lives.
- email1 : Stores primary email of the student(s).
- email2 : Stores secondary email of the student(s).
- address : Stores the address of the student(s).
- description : Stores description of the student(s).
- image : Stores image of the student(s).
- last_login_date : Strores last login date of the student(s).

Features

The Website provides following functionalities to the users :

- **Administrator** :
    - Login/Logout
    - View student information
    - Edit Student Information
    - Enable/disable student accounts

  - Search students

- **Student** :
  - Login/Logout
  - View profile
  - Edit profile
  - Change password
  - Register new profile

Source Code

```
Sudent login.php: <?php
/**
 * Created by PhpStorm.
 * User: faintedbrain63
 * Date: 11/07/2017
 * Time: 8:11 AM
 */
class StudentLogin
{
    private $_stud_id;
    public function __construct($c_stud_id) {
        $this->_stud_id = $c_stud_id;
    }
```

```
public function StudLogin() {
    global $db;
    //Start session
    session_start();

    //Array to store error message
    $error_msg_array = array();
    //Error messages
    $error_msg = FALSE;
    if($this->_stud_id == "") {
        $error_msg_array[] = "Please input your ID number.";
        $error_msg = TRUE;
    }
    if($error_msg) {
        $_SESSION['ERROR_MSG_ARRAY'] = $error_msg_array;
        header("location: http://localhost/voting_system/index.php");
        exit();
    }
    $sql = "SELECT * FROM voters WHERE stud_id = ? LIMIT 1";
    if(!$stmt = $db->prepare($sql)) {
        echo $stmt->error;
    } else {
        $stmt->bind_param("s", $this->_stud_id);
        $stmt->execute();
        $result = $stmt->get_result();
    }
    if($result->num_rows > 0) {
        $row = $result->fetch_assoc();
        //Create session
        session_regenerate_id();
        $_SESSION['ID']      = $row['id'];
        $_SESSION['NAME']    = $row['name'];
        $_SESSION['DEPARTMENT']  = $row['department'];
        $_SESSION['YEAR']    = $row['year'];
        $_SESSION['STUD_ID'] = $row['stud_id'];
        session_write_close();
        header("location: http://localhost/voting_system/stud_page.php");
    } else {
        $error_msg_array[] = "Sorry the ID number you entered is not in the database.";
        $error_msg = TRUE;
        if($error_msg) {
            $_SESSION['ERROR_MSG_ARRAY'] = $error_msg_array;
            header("location: http://localhost/voting_system/index.php");
            exit();
        }
        $stmt->free_result();
```

```
        }
        $result->free();
        return $result;
    }
}
Voting.php: <?php
/**
 * Created by PhpStorm.
 * User: faintedbrain63
 * Date: 12/07/2017
 * Time: 2:24 PM
 */
class Voting
{
    public function READ_ORG() {
        global $db;
        $user_id = $_SESSION['ID'];
        $sql = "SELECT * FROM voters  WHERE id = '$user_id'";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->execute();
            $result = $stmt->get_result();
    }
    return $result;
}
public function READ_POSITION($org) {
    global $db;
    $sql = "SELECT *
            FROM positions
            WHERE org = ?";
    if(!$stmt = $db->prepare($sql)) {
        echo $stmt->error;
    } else {
        $stmt->bind_param("s", $org);
        $stmt->execute();
        $result = $stmt->get_result();
    }
    $stmt->free_result();
    return $result;
}
public function READ_NOMINEES($org, $pos) {
    global $db;
    $sql = "SELECT *
            FROM nominees
            WHERE org = ?
            AND pos = ?";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->bind_param("ss", $org, $pos);
            $stmt->execute();
            $result = $stmt->get_result();
        }
        $stmt->free_result();
        return $result;
    }
    public function VALIDATE_VOTE($org, $pos, $voters_id) {
        global $db;
        //Check to see if the voter votes already
        $sql = "SELECT *
            FROM votes
            WHERE org = ?
            AND pos = ?
            AND voters_id = ?
            LIMIT 1";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
```

```
        $stmt->bind_param("ssi", $org, $pos, $voters_id);
        $stmt->execute();
        $result = $stmt->get_result();
    }
    $stmt->free_result();
    return $result;
}
public function VOTE_NOMINEE($org, $pos, $candidate_id, $voters_id) {
    global $db;
    //Check to see if the voter votes already
    $sql = "SELECT *
            FROM votes
            WHERE org = ?
            AND pos = ?
            AND voters_id = ?
            LIMIT 1";
    if(!$stmt = $db->prepare($sql)) {
        echo $stmt->error;
    } else {
        $stmt->bind_param("ssi", $org, $pos, $voters_id);
        $stmt->execute();
        $result = $stmt->get_result();
    }
```

```
        if($result->num_rows > 0) {
            echo "<div class='alert alert-danger'>Sorry you voted in that position already.</div>";
        } else {
            //Vote successful.
            $sql = "INSERT INTO votes(org, pos, candidate_id, voters_id)VALUES(?, ?, ?, ?)";
            if(!$stmt = $db->prepare($sql)) {
                echo $stmt->error;
            } else {
                $stmt->bind_param("ssii", $org, $pos, $candidate_id, $voters_id);
            }
            if($stmt->execute()) {
                echo "<div class='alert alert-success'>Vote successful.</div>";
            }
            $stmt->free_result();
        }
        return $stmt;
    }
}
```

Database:

```
<?php
/**
 * Created by PhpStorm.
 * User: Gizmo
 * Date: 7/2/2017
 * Time: 6:40 PM

 */

//This line of code connects to mysql database

define("HOST_NAME", "localhost");

define("HOST_USER", "root");

define("HOST_PASS", "");

define("HOST_DB", "voting");

$db = new mysqli(HOST_NAME, HOST_USER, HOST_PASS, HOST_DB);

/**

* This line of code checks if connection error exists.

if($db->connect_error) {

    echo $db->connect_errno . " " . $db->connect_error;

} else {

    echo "Connection successful.";

}

 */

Index.php: <?php

//Start session

session_start();

unset($_SESSION['ID']);

unset($_SESSION['NAME']);

unset($_SESSION['COURSE']);

unset($_SESSION['YEAR']);
unset($_SESSION['STUD_ID']);
?>
<!DOCTYPE HTML>
<html lang="en-US">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Voting System</title>
    <link rel="stylesheet" href="assets/css/bootstrap.min.css">
    <link rel="stylesheet" href="assets/css/style_voter.css">
</head>
<body>
<!-- Header -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="index.php">Voting Sytem</a>
        </div>
    </div><!-- /.container-fluid -->
</nav>
<!-- End Header -->
<div class="container">
    <div class="row">
        <div class="col-md-4 col-md-offset-4">
            <div class="login-con">
                <h3>Student Log-in</h3><hr>
                <?php
                if(isset($_SESSION['ERROR_MSG_ARRAY']) &&
is_array($_SESSION['ERROR_MSG_ARRAY']) &&
COUNT($_SESSION['ERROR_MSG_ARRAY']) > 0) {
                    foreach($_SESSION['ERROR_MSG_ARRAY'] as $msg) {
                        echo "<div class='alert alert-danger'>";
                        echo $msg;
                        echo "</div>";
                    }
                    unset($_SESSION['ERROR_MSG_ARRAY']);
                }
                ?>
        <form method="post" action="process/login.php" role="form">
          <div class="form-group has-warning has-feedback">
            <label for="stud_id">Student ID</label>
            <input type="text" name="stud_id" id="stud_id" class="form-control"
autocomplete="off"><br>
            <label for="stud_id">Password</label>
            <input type="password" name="password" id="stud_id" class="form-control"
autocomplete="off">
            <span class="glyphicon glyphicon-lock form-control-feedback"></span>
          </div>
            <button type="submit" name="submit" class="btn btn-info">Submit</button>
        </form>
      </div>
    </div>
  </div>
</div>
<!-- Footer -->
<nav class="navbar navbar-inverse navbar-fixed-bottom" role="navigation">
  <div class="container">
    <div class="navbar-text pull-left">
      Copyright 2018
    </div>
  </div>
```

```
</nav>
<!-- End Footer -->
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>
Student page.php: <?php
//Include authentication
require("process/auth.php");
//Include database connection
require("config/db.php");
//Include class Voting
require("classes/Voting.php");
?>
<!DOCTYPE HTML>
<html lang="en-US">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Voting System</title>
    <link rel="stylesheet" href="assets/css/bootstrap.min.css">
    <link rel="stylesheet" href="assets/css/style_voter.css">
</head>
```

```
<body>
<!-- Header -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="stud_page.php"><span class="glyphicon glyphicon-home"></span> Voting Sytem</a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="stud_page.php"><span class="glyphicon glyphicon-home"></span></a></li>
                <li class="dropdown">
```

```
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-user"></span> <span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="process/logout.php">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>
<!-- End Header -->
<?php
$readOrganization = new Voting();
$rtnReadOrg = $readOrganization->READ_ORG();
?>
<div class="container">
    <div class="row">
        <div class="col-md-4"><br><br>
            <h3 style="text-align: center;">Select Depertment Election</h3><hr />
            <p style="color: white">,</p>
            <h4>Welcome <?php echo $_SESSION['NAME']; ?></h4>
            <?php if($rtnReadOrg) { ?>
            <form action="voting_page.php" method="GET" role="form">
                <div class="form-group">
                    <label for="organization">Depertment</label>
                        <?php while($rowOrg = $rtnReadOrg->fetch_assoc()) { ?>
                </div>
                <div>
                    <input readonly class="form-control" type="text" name="organization" value="<?php echo $rowOrg['department']; ?>">
                </div><br>
                <div class="form-group">
                    <input type="submit" name="submit" value="Select" class="btn btn-info">
                </div>
            </form>
             <?php } //End while ?>
                <?php $rtnReadOrg->free(); ?>
            <?php } //End if ?>
        </div>
<?php
$readOrganization = new Voting();
$rtnReadOrg = $readOrganization->READ_ORG();
?>
<div class="container">
    <div class="row">
    <div class="col-md-4"><br><br>
      <h3 style="text-align: center;">Select Faculty Election</h3><hr />
      <?php if($rtnReadOrg) { ?>
      <form action="voting_page.php" method="GET" role="form">
        <div class="form-group"><br><br><br>
          <label for="organization">Facuty</label>
            <?php while($rowOrg = $rtnReadOrg->fetch_assoc()) { ?>
        </div>
        <div>
          <input readonly class="form-control" type="text" name="organization"
value="<?php echo $rowOrg['faculty']; ?>">
        </div><br>
        <div class="form-group">
          <input type="submit" name="submit" value="Select" class="btn btn-info">
        </div>
      </form>
       <?php } //End while ?>
        <?php $rtnReadOrg->free(); ?>
      <?php } //End if ?>
    </div>
<?php
$readOrganization = new Voting();
$rtnReadOrg = $readOrganization->READ_ORG();
```

```
?>
<div class="container">
    <div class="row">
        <div class="col-md-4"><br><br>
            <h3 style="text-align: center;">Select Faculty Election</h3><hr />
            <?php if($rtnReadOrg) { ?>
            <form action="voting_page.php" method="GET" role="form">
                <div class="form-group"><br><br><br>
                    <label for="organization">Facuty</label>
                        <?php while($rowOrg = $rtnReadOrg->fetch_assoc()) { ?>
                </div>
                <div>
                    <input readonly class="form-control" type="text" name="organization" value="<?php echo $rowOrg['institution']; ?>">
                </div><br>
                <div class="form-group">
                    <input type="submit" name="submit" value="Select" class="btn btn-info">
                </div>
            </form>
             <?php } //End while ?>
                <?php $rtnReadOrg->free(); ?>
            <?php } //End if ?>
        </div>
    </div>
</div>
<!-- Footer -->
<nav class="navbar navbar-inverse navbar-fixed-bottom" role="navigation">
    <div class="container">
        <div class="navbar-text pull-left">
            Copyright 2018
        </div>
    </div>
</nav>
<!-- End Footer -->
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>
Voting_page.php: <?php
//Include authentication
require("process/auth.php");
//Include database connection
require("config/db.php");
//Include class Voting
require("classes/Voting.php");
?>
```

```html
<!DOCTYPE HTML>
<html lang="en-US">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Voting System</title>
    <link rel="stylesheet" href="assets/css/bootstrap.min.css">
    <link rel="stylesheet" href="assets/css/style_voter.css">
</head>
<body>
<!-- Header -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="stud_page.php"><span class="glyphicon glyphicon-
home"></span> Voting Sytem</a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="stud_page.php"><span class="glyphicon glyphicon-
home"></span></a></li>
                <li class="dropdown">
                    <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-
haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-user"></span>
<span class="caret"></span></a>
                    <ul class="dropdown-menu">
                        <li><a href="process/logout.php">Logout</a></li>
                    </ul>
                </li>
            </ul>
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>
<!-- End Header -->
<?php
if(isset($_GET['organization'])) {
    $org = $_GET['organization'];
}
?>
<?php
$readPos = new Voting();
$rtnReadPos = $readPos->READ_POSITION($org);
?>
<div class="container">
    <div class="row">
        <?php if($rtnReadPos) { ?><br><br>
        <div class="col-md-6 col-md-offset-3">
            <?php
            if(isset($_POST['vote'])) {
                $org            = trim($_POST['org']);
                $pos            = trim($_POST['pos']);
                $candidate_id   = trim($_POST['nominee']);
                $voters_id      = trim($_POST['voter_id']);
                $insertVote = new Voting();
                $rtnInsertVote = $insertVote->VOTE_NOMINEE($org, $pos, $candidate_id,
$voters_id);
            }
            ?>
            <div class="voting-con">
        <h4 style="text-align: center;"><?php echo $org; ?> Voting Page</h4><hr />
        <?php while($rowPos = $rtnReadPos->fetch_assoc()) { ?>
        <form action="<?php $_SERVER['PHP_SELF']; ?>" method="POST" role="form">
          <p class="help-block"><b><?php echo $rowPos['pos']; ?></b></p>
            <?php
            $readNominee = new Voting();
            $rtnReadNominee = $readNominee->READ_NOMINEES($org, $rowPos['pos']);
            ?>
            <?php if($rtnReadNominee) { ?>
              <div class="form-group">
                <select name="nominee" class="form-control">
                  <option value="">*****Select Nominee*****</option>
                  <?php while($rowNominee = $rtnReadNominee->fetch_assoc()) { ?>
                  <option value="<?php echo $rowNominee['id']; ?>"><?php echo
$rowNominee['name']; ?></option>
                  <?php } //End while ?>
                </select>
              </div>
            <?php } //End if ?>
            <input type="hidden" name="org" value="<?php echo $org; ?>">
            <input type="hidden" name="pos" value="<?php echo $rowPos['pos']; ?>">
            <input type="hidden" name="voter_id" value="<?php echo $_SESSION['ID'];
?>">
```

```
                    <?php
                    $validateVote = new Voting();
                    $rtnValVote = $validateVote->VALIDATE_VOTE($org, $rowPos['pos'], $_SESSION['ID'])
                    ?>
                        <button type="submit" name="vote"
                                <?php if($rtnValVote->num_rows > 0) { ?>
                                <?php echo "class='btn btn-default disabled'>"; ?>
                                <?php } else { ?>
                                <?php echo "class='btn btn-info'>"; ?>
                                <?php } //End if ?>
                            Vote
                        </button>
                </form><hr />
                <?php } //End while ?>
            </div>
        </div>
        <?php } //End if ?>
    </div>
</div>
<!-- Footer -->
<nav class="navbar navbar-inverse navbar-fixed-bottom" role="navigation">
    <div class="container">
```

```
        <div class="navbar-text pull-left">
            Copyright 2018
        </div>
    </div>
</nav>
<!-- End Footer -->
<script src="assets/js/jquery.js"></script>
<script src="assets/js/bootstrap.min.js"></script>
</body>
</html>
Faculty.php: <?php
/**
 * Created by PhpStorm.
 * User: faintedbrain63
 * Date: 03/07/2017
 * Time: 1:41 PM
 */
class Organization2
{
    public function INSERT_ORG($organization2) {
        global $db;
        //Check if the organization already exists in the database
        $sql = "SELECT *
```

```
                FROM organization
                WHERE org = ?
                LIMIT 1";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->bind_param("s", $organization);
            $stmt->execute();
            $result = $stmt->get_result();
        }
        if($result->num_rows > 0) {
            echo "<div class='alert alert-danger'>Sorry the organization you are trying to insert already exists in the database.</div>";
        } else {
            //Successfully inserted
            $sql = "INSERT INTO organization(org)VALUES(?)";
            if(!$stmt = $db->prepare($sql)) {
                echo $stmt->error;
            } else {
                $stmt->bind_param("s", $organization);
            }
            if($stmt->execute()) {
                echo "<div class='alert alert-success'>Organization was inserted successfully.</div>";
            }
            $stmt->free_result();
        }
        $result->free();
        return $stmt;
    }
    public function READ_ORG() {
        global $db;
        $sql = "SELECT * FROM organization";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->execute();
            $result = $stmt->get_result();
        }
        return $result;
    }
    public function EDIT_ORG($org_id) {
        global $db;
        $sql = "SELECT *
                FROM organization
                WHERE id = ?
                LIMIT 1";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->bind_param("i", $org_id);
            $stmt->execute();
            $result = $stmt->get_result();
        }
        $stmt->free_result();
        return $result;
    }
    public function UPDATE_ORG($org, $org_id) {
        global $db;
        $sql = "UPDATE organization
                SET org = ?
                WHERE id = ?
                LIMIT 1";
        if(!$stmt = $db->prepare($sql)) {
            echo $stmt->error;
        } else {
            $stmt->bind_param("si", $org, $org_id);
        }
        if($stmt->execute()) {
```

```
      echo "<div class='alert alert-success'>Update successful <a
href='http://localhost/voting_system/sandbox/add_org.php'><span class='glyphicon glyphicon-
backward'></span> </a></div>";
    }
    $stmt->free_result();
    return $stmt;
  }
  public function DELETE_ORG($org_id) {
    global $db;
    //Delete organization
    $sql = "DELETE FROM organization
        WHERE id = ?
        LIMIT 1";
    if(!$stmt = $db->prepare($sql)) {
      echo $stmt->error;
    } else {
      $stmt->bind_param("i", $org_id);
    }
    if($stmt->execute()) {
      header("location: http://localhost/voting_system/sandbox/add_org.php");
      exit();
    }
    $stmt->free_result();
```

```
        return $stmt;
    }
}
Admin login.php: <?php
/**
 * Created by PhpStorm.
 * User: Gizmo
 * Date: 7/2/2017
 * Time: 6:59 PM
 */
class Admin_Login
{
    private $_username;
    private $_password;
    public function __construct($c_username, $c_password) {
        $this->_username = $c_username;
        $this->_password = $c_password;
    }
    public function AdminLogin() {
        global $db;
        //Start session
        session_start();
        //Array to validate errors
```

```
$error_msg_array = array();
//Error messages
$error_msg = FALSE;
if($this->_username == "") {
    $error_msg_array[] = "Please input your username";
    $error_msg = TRUE;
}
if($this->_password == "") {
    $error_msg_array[] = "Please input your password";
    $error_msg = TRUE;
}
if($error_msg) {
    $_SESSION['ERROR_MSG_ARR'] = $error_msg_array;
    header("location: http://localhost/voting_system/sandbox/index.php");
    exit();
}
$sql = "SELECT * FROM admin WHERE username = ? AND password = ? LIMIT 1";
if(!$stmt = $db->prepare($sql)) {
    echo $stmt->error;
} else {
    $stmt->bind_param("ss", $this->_username, $this->_password);
    $stmt->execute();
    $result = $stmt->get_result();
}
if($result->num_rows > 0) {
    //Login successful
    $row = $result->fetch_assoc();
    //Create session
    session_regenerate_id();
    $_SESSION['ADMIN_ID']   = $row["id"];
    $_SESSION['ADMIN_NAME'] = $row["name"];
    session_write_close();
    header("location: http://localhost/voting_system/sandbox/admin_page.php");
} else {
    //Login failed
    $error_msg_array[] = "The username and password you entered is incorrect.";
    $error_msg = TRUE;
    if($error_msg) {
        $_SESSION['ERROR_MSG_ARR'] = $error_msg_array;
        header("location: http://localhost/voting_system/sandbox/index.php");
        exit();
    }
    $stmt->free_result();
}
$result->free();
return $result;
```

```
    }
}
Voter result.php: <?php
//Include authentication
require("process/auth.php");
//Include database connection
require("../config/db.php");
//Include class Organization
require("classes/Organization.php");
//Include class Position
require("classes/Position.php");
//Include class Nominees
require("classes/Nominees.php");
?>
<!DOCTYPE HTML>
<html lang="en-US">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Administrator Login</title>
    <link rel="stylesheet" href="../assets/css/bootstrap.min.css">
    <link rel="stylesheet" href="../assets/css/style_admin.css">
</head>
```

```
<body>
<!-- Header -->
<nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
    <div class="container">
        <!-- Brand and toggle get grouped for better mobile display -->
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-expanded="false">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <a class="navbar-brand" href="index.php">Voting Sytem</a>
        </div>
        <!-- Collect the nav links, forms, and other content for toggling -->
        <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
            <ul class="nav navbar-nav navbar-right">
                <li><a href="admin_page.php"><span class="glyphicon glyphicon-home"></span></a></li>
                <li><a href="add_org.php"><span class="glyphicon glyphicon-plus-sign"></span>Add Organization</a></li>
                <li><a href="add_pos.php"><span class="glyphicon glyphicon-plus-sign"></span>Add Position</a></li>
                <li><a href="add_nominees.php"><span class="glyphicon glyphicon-plus-sign"></span>Add Nominees</a></li>
                <li><a href="add_voters.php"><span class="glyphicon glyphicon-plus-sign"></span>Add Voters</a></li>
                <li class="active"><a href="vote_result.php"><span class="glyphicon glyphicon-plus-sign"></span>Vote Result</a></li>
                <li class="dropdown">
                  <a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false"><span class="glyphicon glyphicon-user"></span> <span class="caret"></span></a>
                  <ul class="dropdown-menu">
                    <li><a href="process/logout.php">Logout</a></li>
                  </ul>
                </li>
            </ul>
        </div><!-- /.navbar-collapse -->
    </div><!-- /.container-fluid -->
</nav>
<!-- End Header -->
<?php
$readOrg = new Organization();
```

```
$rtnReadOrg = $readOrg->READ_ORG();
?>
<div class="container">
    <div class="row">
        <?php if($rtnReadOrg) { ?>
        <div class="col-md-3">
            <h3>Select Organization</h3><hr>
            <form action="<?php $_SERVER['PHP_SELF']; ?>" method="GET" role="form">
                <div class="form-group">
                    <label for="organization">Organization</label>
                    <select name="organization" class="form-control">
                        <option value="">*****Select Organization*****</option>
                        <?php while($rowOrg = $rtnReadOrg->fetch_assoc()) { ?>
                        <option value="<?php echo $rowOrg['org']; ?>"><?php echo $rowOrg['org'];
?></option>
                        <?php } //End while ?>
                    </select>
                </div>
                <button  type="submit" name="submit" class="btn btn-info">Submit</button>
            </form>
        </div>
            <?php $rtnReadOrg->free(); ?>
        <?php } //End if ?>
    <div class="col-md-9">
      <?php
      if(!isset($_GET['organization'])) {
        echo "<div class='alert alert-warning'>Please select organization and click submit to show vote result.</div>";
      } else {
      $org = trim($_GET['organization']);
      ?>
        <a  href="http://localhost/voting_system/sandbox/print_result.php?organization=<?php echo $org; ?>"><h3><span class="glyphicon glyphicon-print pull-right"></h3></span> </a>
        <h4><?php echo $org; ?> Result</h4>
        <hr>
<?php
//$org=('LAUTECH');
$qry = "SELECT COUNT(org)  AS count
    FROM votes WHERE org='$org'" ;
$res = $db->query($qry);
$total = 0;
$rec = $row = $res->fetch_assoc();
$total = $rec['count'];
echo "<div   class='col-md-3'>Total Vote:" . $total . "\n";
echo "</div>";
?>
```

```
<?php

//$org=('LAUTECH');

$qry = "SELECT COUNT(department)  AS count

        FROM voters WHERE department     = '$org'" ;

$res = $db->query($qry);

$total = 0;

$rec = $row = $res->fetch_assoc();

$total = $rec['count'];

echo "<div class='col-md-3'>Department Total voter: " . $total . "\n";

echo "</div>";

?>

<?php

//$org=('LAUTECH');

$qry = "SELECT COUNT(faculty)  AS count

        FROM voters WHERE faculty     = '$org'" ;

$res = $db->query($qry);

$total = 0;

$rec = $row = $res->fetch_assoc();

$total = $rec['count'];

echo "<div class='col-md-3'>Faculty Total voter: </tr>" . $total . "\n";

echo "</div>";

?>

<?php
```

```
//$org=('LAUTECH');
$qry = "SELECT COUNT(institution)  AS count
    FROM voters WHERE institution     = '$org'" ;
$res = $db->query($qry);
$total = 0;
$rec = $row = $res->fetch_assoc();
$total = $rec['count'];
echo "<div class='col-md-3'>Institution Total voter:</tr> " . $total . "\n";
echo "</div>";
?>
<br><BR><BR>
        <?php
        $readPos = new Position();
        $rtnReadPos = $readPos->READ_POS_BY_ORG($org);
        ?>
        <?php if($rtnReadPos) { ?>
          <?php while($rowPos = $rtnReadPos->fetch_assoc()) { ?>
          <h5><?php echo $rowPos['pos']; ?></h5>
            <?php
            $readNomOrgPos = new Nominees();
            $rtnReadNomOrgPos  =  $readNomOrgPos->READ_NOM_BY_ORG_POS($org,
$rowPos['pos']);
            ?>
```

```
<div  class="table-responsive">
  <?php if($rtnReadNomOrgPos) { ?>
  <table class="table table-condensed table-striped">
    <tr>
     <!-- <th>ID</th>-->
      <th>Name</th>
      <th>Votes</th>
    </tr>
    <?php while($rowCountVotes = $rtnReadNomOrgPos->fetch_assoc()) { ?>
      <?php
      $countVotes = new Nominees();
      $rtnCountVotes = $countVotes->COUNT_VOTES($rowCountVotes['id'])
      ?>
      <tr>
        <!--<td style="width: 20%;"><?php// echo $rowCountVotes['id']; ?></td>-->
        <td style="width: 70%;"><?php echo $rowCountVotes['name']; ?></td>
        <td style="width: 10%;"><?php echo $rtnCountVotes->num_rows; ?></td>
      </tr>
    <?php } //End while ?>
  </table>
  <?php $rtnReadNomOrgPos->free(); } //End if ?>
```

```
                        </div>
                    <?php } //End while ?>
                <?php $rtnReadPos->free(); } //End if ?>
            <?php } //End if ?>
        </div>
    </div>
</div>
<!-- Footer -->
<nav class="navbar navbar-inverse navbar-fixed-bottom" role="navigation">
    <div class="container">
        <div class="navbar-text pull-left">
            Copyright 2017
        </div>
    </div>
</nav>
<!-- End Footer -->
<script src="../assets/js/jquery.js"></script>
<script src="../assets/js/bootstrap.min.js"></script>
</body>
</html>
```

AND MANY MORE.

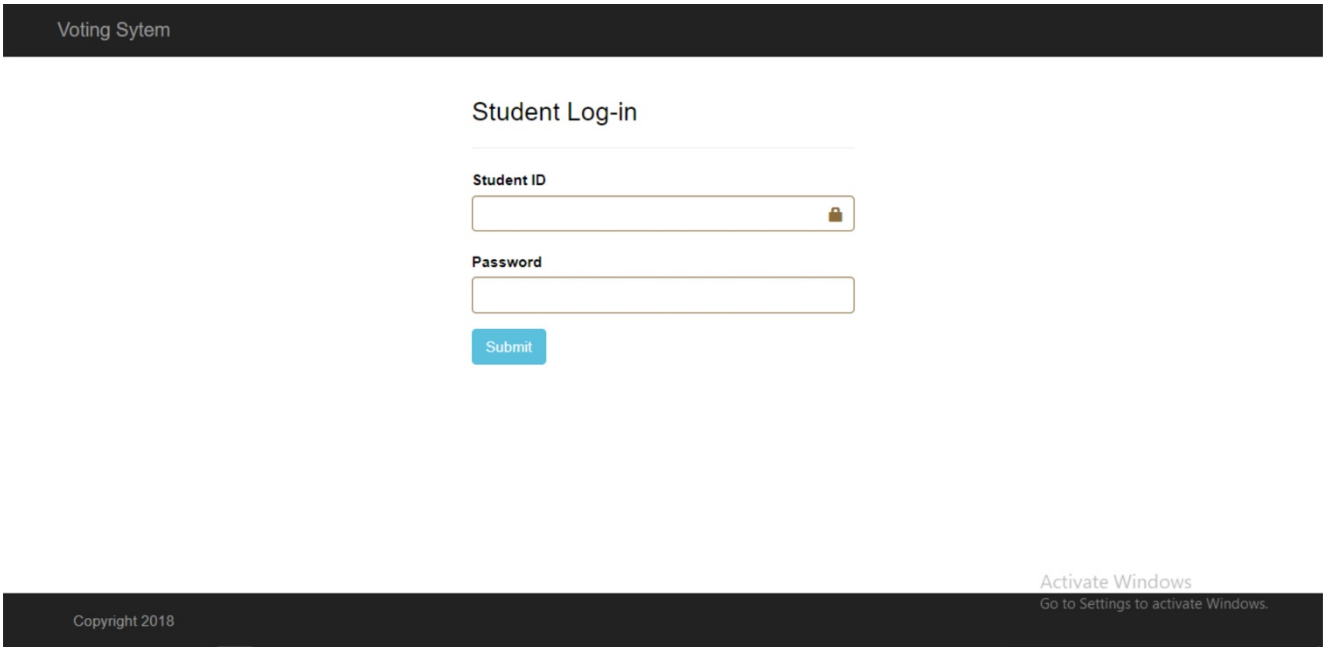
CHAPTER FOUR

RESULT AND DISCUSSION OF THE SYSTEM FUNCTIONS

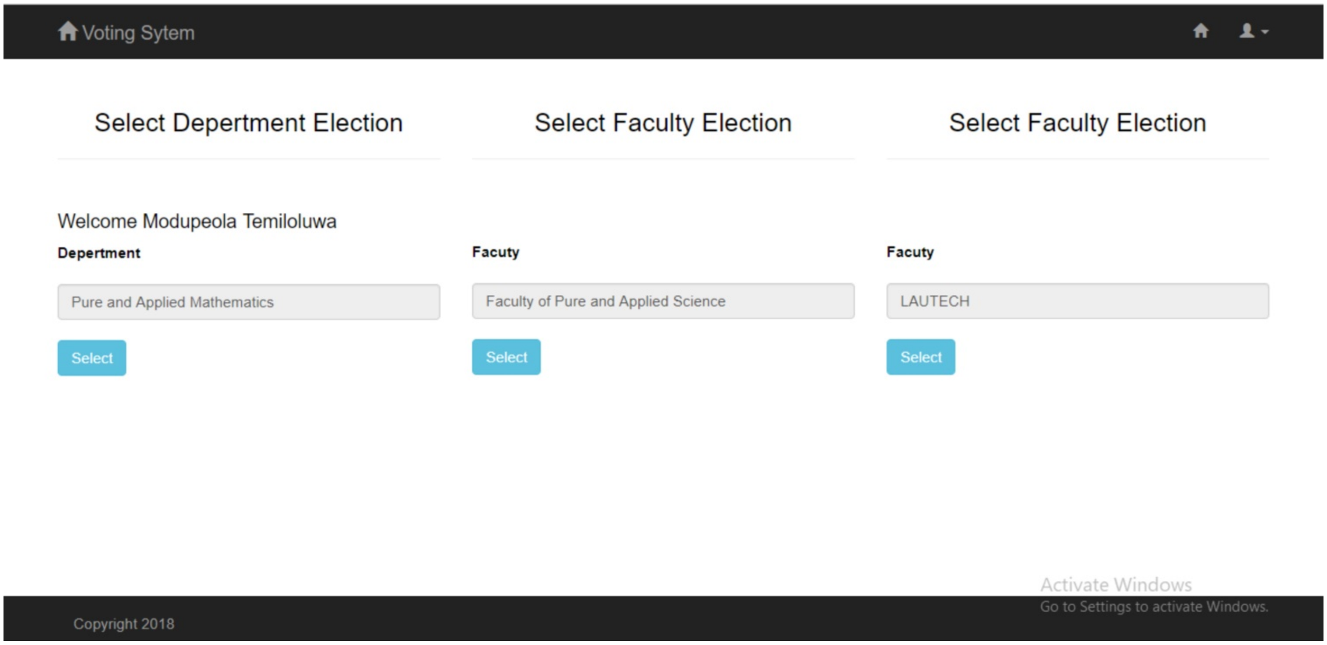
4.1 Result

In this section snapshots of various input/output interfaces are:

This is the student login page where student that have been registered will be able to login and vote



This is the student home page after logging in there student ID (which is there matriculation number) and there password



Select Depertment Election

Select Faculty Election

Select Faculty Election

Welcome Modupeola Temiloluwa

Depertment: Pure and Applied Mathematics

Facuty: Faculty of Pure and Applied Science

Facuty: LAUTECH

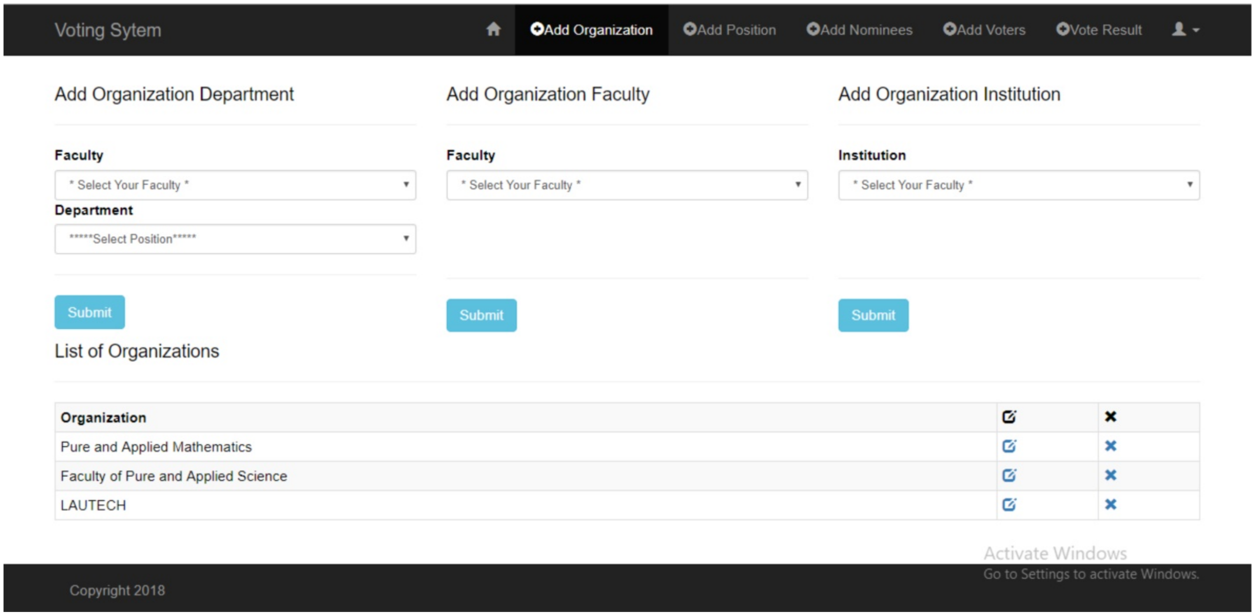
Copyright 2018

This is the voting page where the student votes for candidate of their own choice for their respective post.

This is the administrator login page where the administrator will login his/her username and the password

This is the administrator home page

This is where the administrator add the organization conducting an election



Add Organization Department

Faculty: * Select Your Faculty *

Department: *****Select Position*****

Add Organization Faculty

Faculty: * Select Your Faculty *

Add Organization Institution

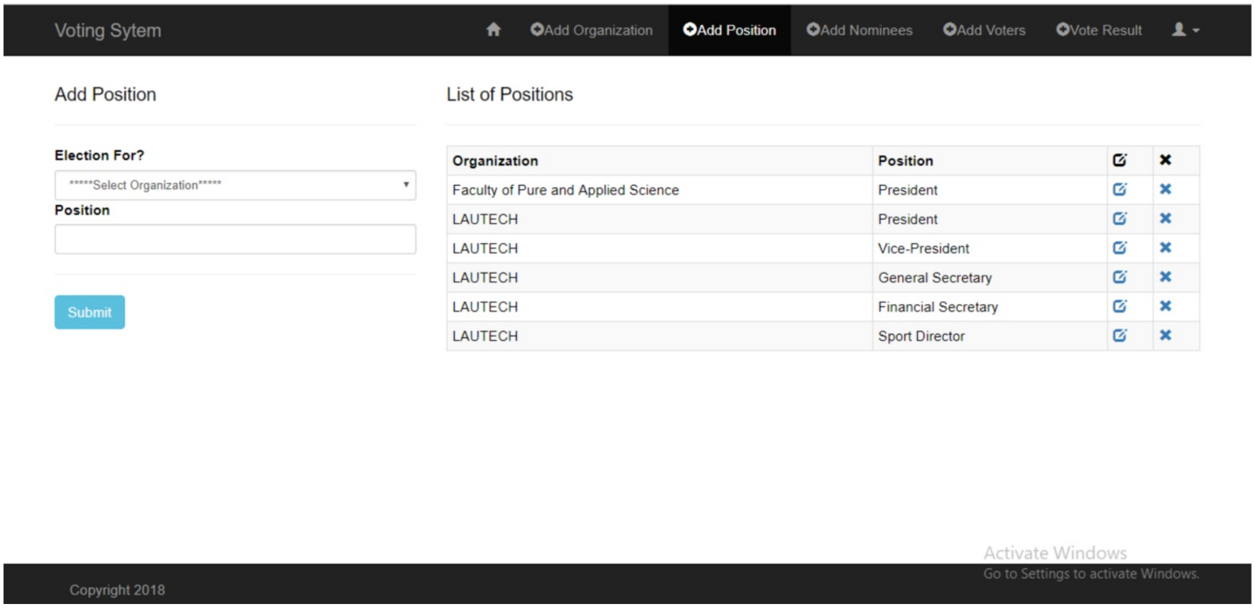
Institution: * Select Your Faculty *

List of Organizations

Organization	✎	✖
Pure and Applied Mathematics	✎	✖
Faculty of Pure and Applied Science	✎	✖
LAUTECH	✎	✖

Copyright 2018

This is where the administrator add the positions available for vote



Add Position

Election For?: *****Select Organization*****

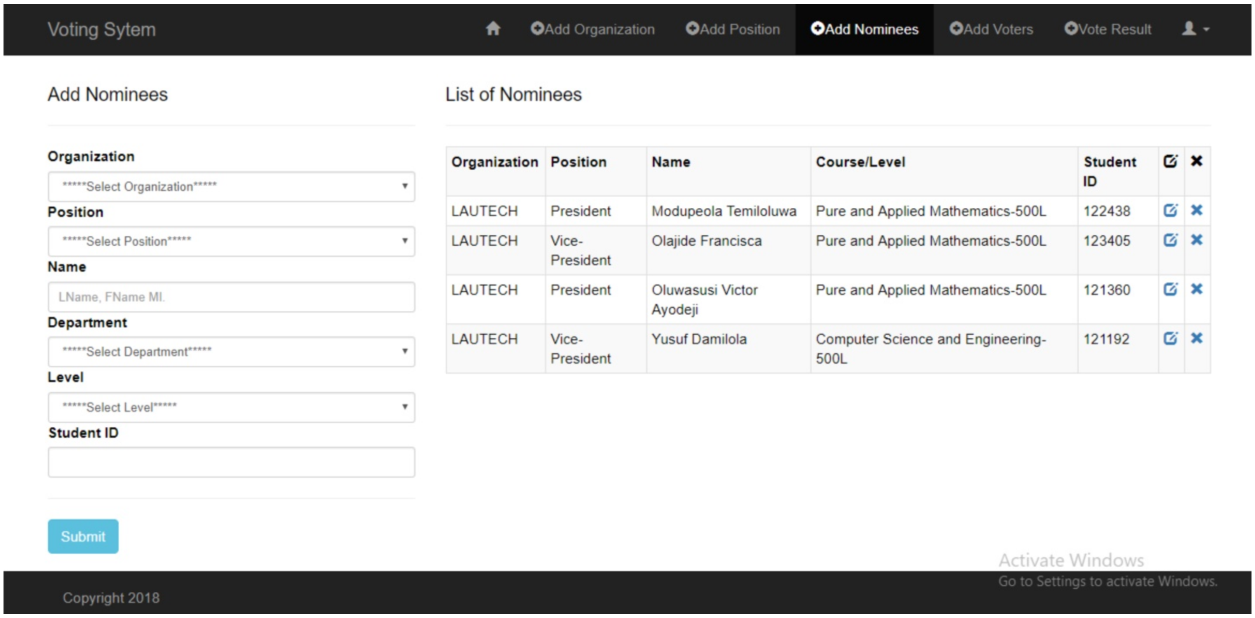
Position:

List of Positions

Organization	Position	✎	✖
Faculty of Pure and Applied Science	President	✎	✖
LAUTECH	President	✎	✖
LAUTECH	Vice-President	✎	✖
LAUTECH	General Secretary	✎	✖
LAUTECH	Financial Secretary	✎	✖
LAUTECH	Sport Director	✎	✖

Copyright 2018

This is where the administrator add the candidate or nominee to be voted for



Add Nominees

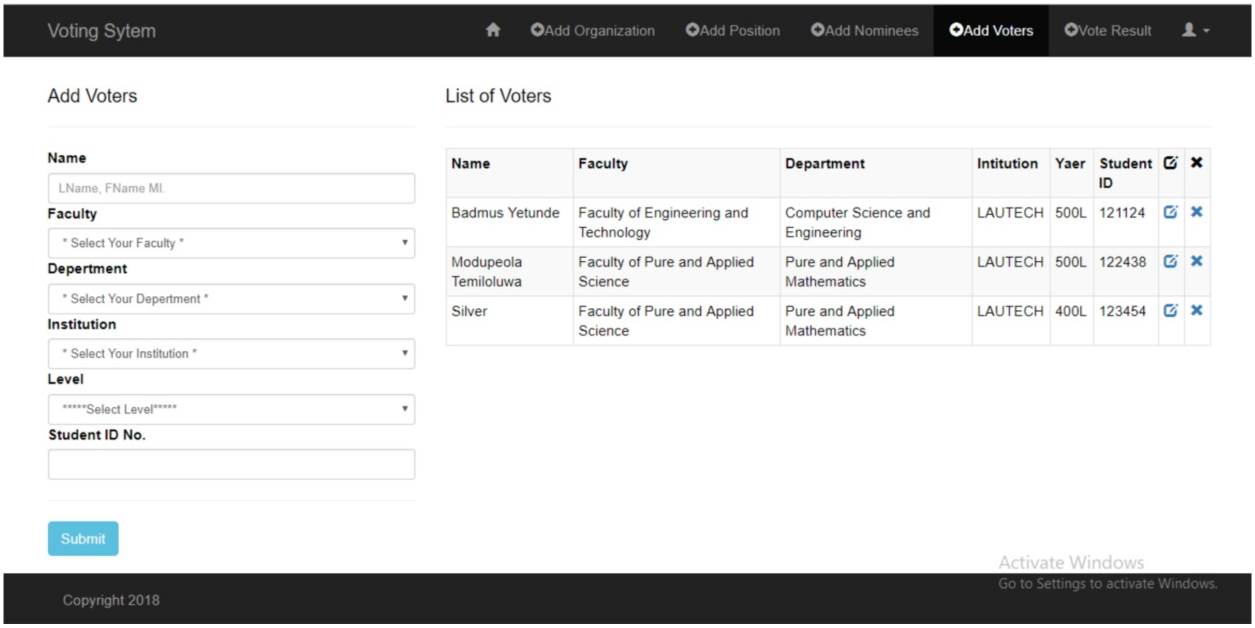
Organization
*****Select Organization*****
Position
*****Select Position*****
Name
LName, FName MI.
Department
*****Select Department*****
Level
*****Select Level*****
Student ID

Submit

List of Nominees

Organization	Position	Name	Course/Level	Student ID	✎	✖
LAUTECH	President	Modupeola Temiloluwa	Pure and Applied Mathematics-500L	122438	✎	✖
LAUTECH	Vice-President	Olajide Francisca	Pure and Applied Mathematics-500L	123405	✎	✖
LAUTECH	President	Oluwasusi Victor Ayodeji	Pure and Applied Mathematics-500L	121360	✎	✖
LAUTECH	Vice-President	Yusuf Damilola	Computer Science and Engineering-500L	121192	✎	✖

This is where voters registration is done whereby the administrator adding all eligible voters



Add Voters

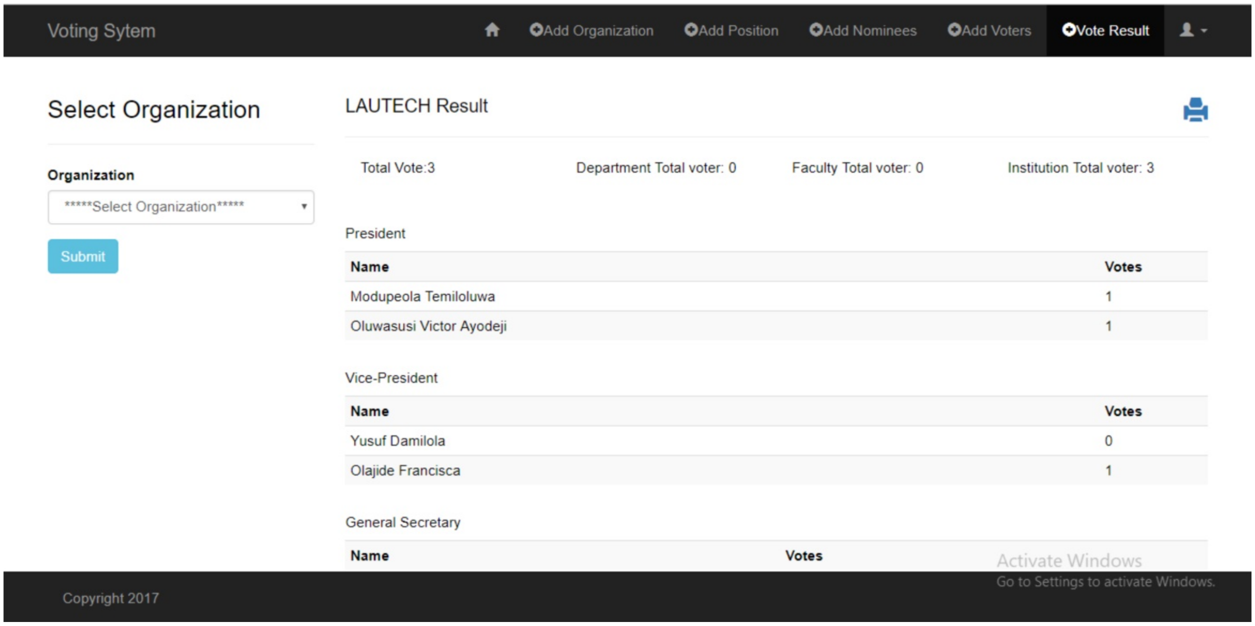
Name
LName, FName MI.
Faculty
* Select Your Faculty *
Depertment
* Select Your Depertment *
Institution
* Select Your Institution *
Level
*****Select Level*****
Student ID No.

Submit

List of Voters

Name	Faculty	Department	Intitution	Yaer	Student ID	✎	✖
Badmus Yetunde	Faculty of Engineering and Technology	Computer Science and Engineering	LAUTECH	500L	121124	✎	✖
Modupeola Temiloluwa	Faculty of Pure and Applied Science	Pure and Applied Mathematics	LAUTECH	500L	122438	✎	✖
Silver	Faculty of Pure and Applied Science	Pure and Applied Mathematics	LAUTECH	400L	123454	✎	✖

This is the vote result which shows the total number of voters and the total number votes



Select Organization

Organization: *****Select Organization*****

LAUTECH Result

Total Vote:3 Department Total voter: 0 Faculty Total voter: 0 Institution Total voter: 3

President

Name	Votes
Modupeola Temiloluwa	1
Oluwasusi Victor Ayodeji	1

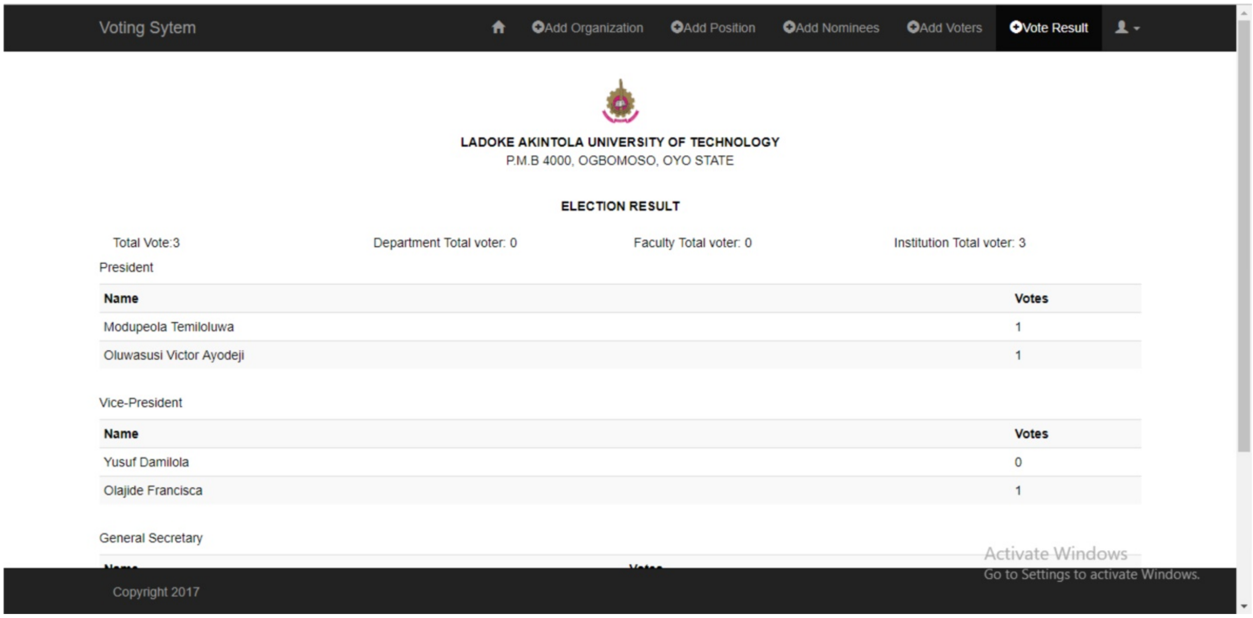
Vice-President

Name	Votes
Yusuf Damilola	0
Olajide Francisca	1

General Secretary

Name	Votes

Copyright 2017



LADOKE AKINTOLA UNIVERSITY OF TECHNOLOGY
P.M.B 4000, OGBOMOSO, OYO STATE

ELECTION RESULT

Total Vote:3 Department Total voter: 0 Faculty Total voter: 0 Institution Total voter: 3

President

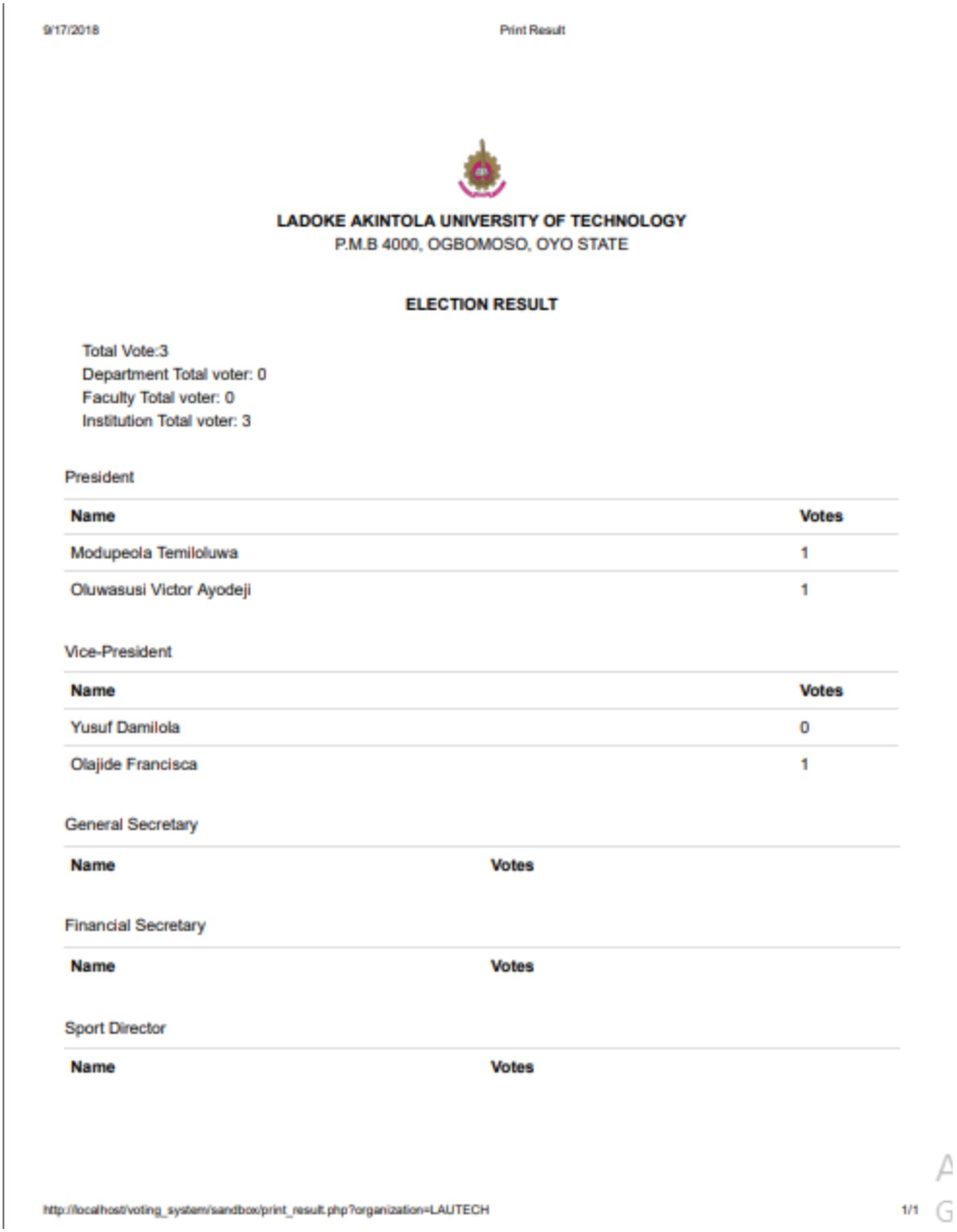
Name	Votes
Modupeola Temiloluwa	1
Oluwasusi Victor Ayodeji	1

Vice-President

Name	Votes
Yusuf Damilola	0
Olajide Francisca	1

General Secretary

Copyright 2017



LADOKE AKINTOLA UNIVERSITY OF TECHNOLOGY
P.M.B 4000, OGBOMOSO, OYO STATE

ELECTION RESULT

Total Vote:3
Department Total voter: 0
Faculty Total voter: 0
Institution Total voter: 3

President

Name	**Votes**
Modupeola Temiloluwa	1
Oluwasusi Victor Ayodeji	1

Vice-President

Name	**Votes**
Yusuf Damilola	0
Olajide Francisca	1

General Secretary

Name	**Votes**

Financial Secretary

Name	**Votes**

Sport Director

Name	**Votes**

CHAPTER FIVE

CONCLUSION

The main aspect behind Student Electronic Voting System (SEVS) is that it enabled us to bring out the new ideas that were sustained within us for many for many days. This project offers the voters to cast easily through internet. Vote counting is also made easy by the Student Electronic Voting System (SEVS) since it's just a matter of querying the database. Student Electronic Voting System (SEVS) is used by a number of countries today. Developing a good system is critical to the success of the system to prevent system failures and to gain wide acceptance as the best method available. A good Student Electronic Voting System (SEVS) requires ten characteristics which this system already has. These are:

1. Accuracy
2. Convenience
3. Reliability
4. Verifiability
5. Flexibility
6. Consistency
7. Democracy
8. Mobility
9. Social Acceptance Privacy

In analyzing, designing, implementing, and maintaining standards, we considered these characteristics as the foundation. Student Electronic Voting System (SEVS) will be an inexpensive and less time consuming method.

Recommendation

After my research and my finalization of this project, I highly recommend that the Student Electronic Voting System (SEVS) serves to be the best to be put in use in the institutions which will encourage transparency in the election and violence free election.