LAIKIPIA UNIVERSITY STUDENT ORGANIZATION ONLINE ELECTION SYSTEM.

|  |  |
| --- | --- |
| Name: | Registration No: |
| Ya’kov Erick | N11/31002/13 |
| John Maina | N11/30412/13 |
| Robert Ouko | N11/31010/13 |
| Doris Awino | N11/31007/13 |
| Gregory Ochieng | N11/31029/13 |

A software project submitted in partial fulfillment for the requirement of award of Bachelor of Science Computer Science Degree of Laikipia University.

April, 2015

# **DECLARATION**

This software project is our original work, except where otherwise stated and has not been presented for a degree in any other University or any other award.

………………………………. ………………………

Robert Ouko Date:

# **CERTIFICATION**

The undersigned certify that he has read and hereby recommend for acceptance of Laikipia University a software project entitled “Laikipia University Student Organization Online Election System”.

………………………………. ………………………

Dr. Simon M. Karume Date:

Department of Computer Science and IT

Laikipia University

# **COPYRIGHT**

This software project is copyright material protected under the Berne Convection, the copyright Act 1999 and other international and national enactments in that behalf, on intellectual property. It may not be reproduced by any means in full or in part except for short extracts in fair dealing so for research or private study, critical scholarly review or discourse with acknowledgement, with written permission of the Dean School of Science and Applied Technology on behalf of both the author and Laikipia University.

# **DEDICATION**

This project is dedicated to the Laikipia University for their support for the resources and information required for this project development. We dedicate it to Dr. Karume Computer Science Lecturer, Laikipia University, for his technical guidance in the project and motivation towards the project development. Without forgetting family and friends who also provided the motivation, financial and moral support.

# **ACKNOWLEDGEMENT**

We would like to express our greatest gratitude to our team’s project instructor, Dr. Karume for giving us the guideline for project development. Special thanks to the Laikipia University Information Technology Department for technical support relevant for our project completion. Lastly, team member’scooperation and dedication towards the development and bringing resources together to ensure the completion of the Laikipia University Student Organization Online Voting System.

# **ABSTRACT**

The aim of this project is to design a student electronic electoral system based upon the Laikipia university student organization election process to help in solving some of the problems related to the current existing manual election system by making it more convenient, fast, transparent and an economical process. The Student Electronic Electoral System will be able to register students online as voters for the Laikipia Students Organization Elections. It will also be able to provide for the online application of student candidate and verify if the candidateis registered. Validation and authentication of a candidate is done manually by the Dean of students so as toidentify him or her as a valid candidate to vie for Laikipia University Students Organization post. During voting dates the registered student voters will be able to login and vote their choices. The election statistics implemented by the system, including vote tallying and election relevant statistics are clearly provided by the system. The methods used to gather the requirements for the system included, interviews, research of publications and online materials. The scope of the system is to work within students: registration, voter verification, candidate validation, vote tallying and provision of any relevant election statistics. The system development process is scheduled to be implemented within four months; from 21st January 2015 up to 15th April 2015. The system is to be developed using interactive web development tools such as: PHP, HTML, CSS, JAVA Script, JQuery and PHP and MySQL databases.

# **TABLE OF CONTENT**

Contents

**[DECLARATION 2](#_Toc417378800)**

**[CERTIFICATION 2](#_Toc417378801)**

**[COPYRIGHT 3](#_Toc417378802)**

**[DEDICATION 4](#_Toc417378803)**

**[ACKNOWLEDGEMENT 5](#_Toc417378804)**

**[ABSTRACT 6](#_Toc417378805)**

**[TABLE OF CONTENT 7](#_Toc417378806)**

**[TABLE OF FIGURES 9](#_Toc417378807)**

**[CHAPTER One:](#_Toc417378808)****[INTRODUCTION 9](#_Toc417378808)**

[1.1 Background information 9](#_Toc417378810)

[1.2 Problem definition 10](#_Toc417378811)

[1.3 Description of the current system 10](#_Toc417378812)

[5.2.1 How the current system works 10](#_Toc417378813)

[1.4 Proposed solution 12](#_Toc417378815)

[1.4.1 Justification 12](#_Toc417378816)

[1.4.2 Objectives 12](#_Toc417378817)

[1.5 Project schedule 13](#_Toc417378818)

[1.6 Project budget 14](#_Toc417378819)

**[CHAPTER TWO: LITERATURE REVIEW](#_Toc417378820)** [15](#_Toc417378820)

[2.1 Introduction 15](#_Toc417378822)

[2.2 Convenience of electronic voting system 15](#_Toc417378823)

[2.3 Cost effectiveness 16](#_Toc417378824)

[2.4 Integrity of voting process 17](#_Toc417378825)

[2.5 Time convenience 17](#_Toc417378826)

**[CHAPTER Three: METHODOLOGY](#_Toc417378827)** [18](#_Toc417378827)

[3.1 Introduction 18](#_Toc417378829)

[3.2 Software process models adopted 18](#_Toc417378830)

[3.2.1 Strength 19](#_Toc417378831)

[3.2.2 Weaknesses 19](#_Toc417378832)

[3.3 Requirement gathering tools 19](#_Toc417378833)

[3.4 System requirements 20](#_Toc417378834)

[3.4.1 Hardware requirements 20](#_Toc417378835)

[3.5 Software requirements 21](#_Toc417378836)

**[CHAPTER Four:](#_Toc417378837)****[SYSTEM ANALYSIS AND DESIGN 21](#_Toc417378837)**

**[4.1](#_Toc417378839)** [Introduction](#_Toc417378839) **[21](#_Toc417378839)**

**[4.2](#_Toc417378840)** [Architecture design](#_Toc417378840) **[22](#_Toc417378840)**

**[4.3](#_Toc417378841)** [System analysis](#_Toc417378841) **[23](#_Toc417378841)**

**[4.3.1](#_Toc417378842)** [Context diagram 23](#_Toc417378842)

**[4.3.2](#_Toc417378843)** [Domain analysis 24](#_Toc417378843)

**[4.3.3](#_Toc417378844)** [Use case models 26](#_Toc417378844)

**[4.4.1](#_Toc417378845)** [Entity relationship diagram and class diagram 30](#_Toc417378845)

**[4.4.2](#_Toc417378846)** [Sequence diagrams or data flow diagrams 32](#_Toc417378846)

**[4.4](#_Toc417378847)** [Database design](#_Toc417378847) **[36](#_Toc417378847)**

**[CHAPTER FIVE:](#_Toc417378848)****[SYSTEM IMPLEMENTATION AND TESTING 37](#_Toc417378848)**

**[5.1 Introduction 37](#_Toc417378850)**

**[5.2](#_Toc417378851)** [Summary of modules 37](#_Toc417378851)

**[5.2.1](#_Toc417378852)** [Voter module 38](#_Toc417378852)

**[5.2.2](#_Toc417378853)** [Candidate module 38](#_Toc417378853)

**[5.2.3](#_Toc417378854)** [System administrator module 39](#_Toc417378854)

**[5.2.4](#_Toc417378855)** [Ballot Module 39](#_Toc417378855)

**[5.2.5](#_Toc417378856)** [Real-time vote statistics module 40](#_Toc417378856)

[5.6 Summary of how the system works 40](#_Toc417378857)

[5.6 Test Regime 60](#_Toc417378858)

**[5.6](#_Toc417378859)** [Conclusion 61](#_Toc417378859)

**[5.6](#_Toc417378860)** [Recommendation 61](#_Toc417378860)

**[REFERENCES](#_Toc417378861)** [63](#_Toc417378861)

**[APPENDICES](#_Toc417378862)** [63](#_Toc417378862)

**[Appendix 1: Questionnaire](#_Toc417378863)** [63](#_Toc417378863)

**[Appendix 2: Source codes](#_Toc417378864)** [74](#_Toc417378864)

**[Appendix 3: Test data](#_Toc417378865)** [261](#_Toc417378865)

# **TABLE OF FIGURES**

[Figure 1: system architecture (en.m.wikipedia.org, 2015) 23](#_Toc416760379)

[Figure 2: context diagram (Author) 24](#_Toc416760380)

[Figure 3: Administrator use case (Author) 27](#_Toc416760381)

[Figure 4: Candidate use case (Author) 28](#_Toc416760382)

[Figure 5: Voter use case (Author) 29](#_Toc416760383)

[Figure 6: Entity relationship diagram (Author) 30](#_Toc416760384)

[Figure 7: Class diagram (Author) 31](#_Toc416760385)

[Figure 8: Voter sequence diagram (Author) 32](#_Toc416760386)

[Figure 9: Ballot sequence diagram (Author) 33](#_Toc416760387)

[Figure 10: Candidate sequence diagram (Author) 34](#_Toc416760388)

[Figure 11: Administrator sequence diagram (Author) 35](#_Toc416760389)

[Figure 12: Home page (author) 43](#_Toc416760390)

[Figure 13: Voter registration page (author) 44](#_Toc416760391)

[Figure 14: Candidate Application (Author) 45](#_Toc416760392)

[Figure 15: Voter Login page (Author) 46](#_Toc416760393)

[Figure 16: Ballot paper process 1 (Author) 47](#_Toc416760394)

[Figure 17: Ballot paper process 2 (Author) 48](#_Toc416760395)

[Figure 18: Ballot paper process 3 (Author) 49](#_Toc416760396)

[Figure 19: Ballot paper process 4 (Author) 50](#_Toc416760397)

[Figure 20: Ballot paper process 5 (Author) 51](#_Toc416760398)

[Figure 21: Ballot paper process 6 (Author) 52](#_Toc416760399)

[Figure 22: Ballot paper process 7 (Author) 53](#_Toc416760400)

[Figure 23: Ballot paper process 8 (Author) 54](#_Toc416760401)

[Figure 24: Vote results (Author) 55](#_Toc416760402)

[Figure 25: Administrator registration page (Author) 56](#_Toc416760403)

[Figure 26: Administrator Login (Author) 57](#_Toc416760404)

[Figure 27: Administrator panel (Author) 58](#_Toc416760405)

[Figure 28: Add candidates page (Author) 59](#_Toc416760406)

[Figure 29: Test regime (Author) 60](#_Toc416760407)

# I**NTRODUCTION**

## **Background information**

Every student has a right to vote for their preferred candidates. However, many students do not vote due to the inconvenience of the long queues or perhaps they are not near the university during the Election Day. There has to be a better way to hold university elections. One that will lower the number of rejected votes, reduce the inconveniences and increase voter participation in every election. Our proposed Laikipia University Student Organization Online Voting System lays the framework for holding the elections online.

## **Problem definition**

University elections has always created student -administration conflict with the student perspective that the administration manipulates the tallying process in favor of their candidates of interest. There has been lack of confidentiality with the students concerning the whole process, the fairness and openness in the manual system of elections. The current election system consumes a lot of time in terms of queues and manual counting of the votes and has been quite inconvenient to the most of students because the election date is not set aside in their timetable, leading to low voter turnout.

## **Description of the current system**

### **How the current system works**

Student register at the registrar’s office and validated as valid voters. The administration announces if students are interested in commissioner’s position to oversee the elections to be conducted. Only the registered students are allowed to apply for the positions. The current executives at the Laikipia University Student Organization issue the interested students with application letters; they verify their details and oversee the elections of the commissioners. The elections are only done by the registered students. The current Laikipia University Students Organization seats are declared vacant and interested students for various positions in the Laikipia University Students Organization are asked to apply. Only registered students are allowed to apply. The already elected commissioners form a committee with the University Dean of Students, they issue the interested students with application letters. The interested students apply and give the committee their letters of application. The committee verifies and validates if the applied candidates are valid by checking the C.F counts and disciplinary records. Candidates who go through are shortlisted and given a period to campaign for their positions. The dean of students does the lection process planning and budgeting. On the election dates, the student voters are checked manually from the nominal roll and if valid, he or she is issued with the ballot box. The student voters vote for the persons of interest. After voting, the votes are counted in the public by the commissioners and then the dean of student who acts as the returning officer announces the official results.

### **Weakness of the current system**

The current electoral system has various limitations from the registration process of the students to the actual tallying of the votes: some students are registered illegally and behind the schedule in favor of certain political inventions, lack of confidentiality during candidate validation, insecurity and violence during election process, student inconvenience as students has to vote during class time, lack of confidence with the manual process, time consuming and tiresome as voters have to make long queues, the manual system has high probability of data manipulation especially during tallying of the votes.

## **Proposed solution**

### **Justification**

The Student Electoral Elections system plans to automate every subsystem of the process; candidate verification and validation to bring back the confidence of the candidates and voters that it is fair and free for everyone. The system will allow every student to vote wherever they are to prevent the inconveniency issues and cut off the long queues during elections, this will save on time. The automation of the tallying system will eliminate the fear of human error in counting and manipulation of the results. Wholly, the system reduces the budgeting cost for the whole electoral process in the university.

### **Objectives**

1. General objectives
2. Make voting process convenient so as to increase voter turnout and ensure every student expresses his or her democratic views
3. To reduce on the cost of election planning by the university to carter for the electoral process
4. Reduce time wastage by allowing several voters to vote simultaneously and also release results on timethrough automated calculation.
5. Reduces the manipulation of votes and final results hence increase the integrity of the electoral process.
6. Specific objectives
7. Check for a voter’s record from the list of the nominal roll database.
8. Add the registered voter’s details in the systems database.
9. Allow the voter to login and logout from the system.
10. Provide the application form for the candidates interested in various positions to apply.
11. Add the details of the candidate’s application into the system database.
12. Allow the candidate to login and logout from the system.
13. Register the administrator.
14. Allow the administrator to login and logout into the system.
15. Allow the administrator to add a list of shortlisted candidates into system database.
16. Retrieve shortlisted candidates from database to the homepage.
17. Retrieve candidate records from the shortlisted candidates table to the ballot form automatically.
18. Allow registered voters to vote by providing them with ballot forms and allow them to submit.
19. Calculate the votes and provide election statistic reports in the real time.
20. Allow voters, administrator and candidates to view election results online.

## **Project schedule**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TASK NO: | TASK DESCRIPTION | STARTING DATE | END DATE | %COMPLETION |
| 1 | Preliminary work | 18th December 2014 | 21st January 2015 | Done |
| 2 | Concept and proposal writing | 21st January 2015 | 4th February 2015 | Done |
| 3 | Requirement analysis | 4th February 2015 | 11th February 2015 | Done |
| 4 | System analysis and design | 11th February 2015 | 18th February 2015 | Done |
| 5 | System implementation | 18th February2015 | 1st April2015 | Done |
| 6 | System testing | 2nd April 2015 | 8th April2015 | Done |
| 7 | Documentation | 21st  January 15 | 8thh April 2015 | Done |
| 8 | Printing and binding | 8th April2015 | 15th April 2015 | Done |

## **Project budget**

|  |  |
| --- | --- |
| EXPENSE | COST( KSH) |
| Online research(internet bundles) | 300.00 |
|  | 8000.00 |
| Software purchase | 1500.00 |
| Printing and photocopy | 1000.00 |
| Miscellaneous | 1000.00 |
| Total | 11800.00 |

# **LITERATURE REVIEW**

## **Introduction**

This chapter presents a review of current literature of the

## **Convenience of electronic voting system**

With the surge of mobile devices, online voting is a convenient option for many voters, allowing them to access ballots anytime and anywhere (Peter Westernaus,2012). Online voting has been considered the most convenient and economical way of expressing an individual’s rights as compared to the traditional ways where the administrators have gone digging deeper into the finances setting up the entire polling site and hiring the technical personnel to oversee the elections. It is therefore considered reliable and the most appropriate way to replace the traditional way of voting. Dictson and Ray (2000, p.3) stated that this method allows the voter to log in through secure means, verify their identities, and vote on electronic ballots. This system seems to be more convenient since the voters can vote online using mobile devices from home, office, library, school or any place where the internet is accessible. The online voting systems has some transactional advantages to the voters as it may provide the voter with online help information to guide the voter on the voting procedures rendering the process interactive. Online voting improves the voter awareness about their contesting leaders as the internet provides every information to the voters. (Dictson and Ray, 2000) said that “for potential voters, the internet is an opportunity to access anything they need to know about voting, 24 hours a day, instantly”.

## **Cost effectiveness**

Cost benefit is a very important factor to consider in the election process. There are a lot of financial issues that occur in the traditional ways of elections. This entails, expensive election officials to carry and oversee the election process who in the end have to be paid for the work done in tallying and announcing results from various stations, the expensive and many voting equipment’s purchased for the election process. The Internet based voting system according to Dictson and Ray (2000) would free a location that is known as an absolute requirement. Since online voting utilizes electronic ballots, the cost of production, printing and mailing/transportation of manual ballot paper is eliminated. The transport cost of voters to various voting stations is also eliminated.

## **Integrity of voting process**

Voting is the most powerful way for people to have a voice in leadership and direction of their organization. When allowed to vote in free, fair and open elections, people feel a greater sense of value, ownership, responsibility and confidence in the leadership.

Dealing with security of systems has been one of the major challenge the online voting systems face since there are a lot of risks and malicious minds in the open internet. The most important step of assuring the security system is the verification of individual voters (Dictson and Ray, 2000). The voters are registered by the system, provided with the voter identification numbers and other security credentials. The voters are the restricted to login to the system using the secret identification numbers provided by the system. These credentials should match. The online voting system also offer encryption to the ballot papers the voter submit through the internet so as not to be manipulated by the unauthorized users. The only problem that is faced today is the use of digital signature for voter’s authorization. However, the use of Personal Identification Number probably will answer that problem. Online voting reduces the manipulation of votes and results by election officials as in the traditional system since no one interacts with cast votes physically, the automatic tallying and real time display of makes the counting process very transparent for everyone to see. Online voting puts restrictions to ensure that a voter can vote only once.

## **Time convenience**

Convenience of any system encourages participation. People sometimes are unwilling to participate on election because the set election date is inconvenient since there is no free day set aside for election in many institutions. Traditional system requires people to stand in line for hours to get the chance to vote, while they also have a lot of routine activities to do (Bonetti, 2000). It seems that the traditional one become less efficient regarding its longer waiting time. The internet therefore provides the most convenient way to vote. Dictson and Ray (2000, p.5) argued, “Perhaps the most compelling argument in favor of online voting is the convenience factor. Convenience will obviously encourage participation.”

Traditional system sometimes requires double counting of ballots and double -checking to avoid the human errors this prolongs the counting time and wastes people’s time as they have to wait for election officials to finish counting votes one by one manually. With online voting systems the tallying process is automated hence results are displayed in real time meaning that tallying stops as soon as voting stops and results are announced a few minutes after hence the process is very fast.

# **METHODOLOGY**

## **Introduction**

In this chapter data collection methodologies, software process models adopted, software and hardware requirements are stated.

## **Software process models adopted**

This project uses iterative development lifecycle, where components of the application are developed through a series of hard iteration.

### **Strength**

The choice of the iterative method of development is to ensure the Laikipia University Organization Online Election System is improved step by step. We are interested in attacking the systems defects at early stages and therefore pick on the iterative process model.

### **Weaknesses**

The weakness of the software process model is that there is costly architecture and design issues may arise because not all requirements are gathered up for the entire lifecycle.

## **Requirement gathering tools**

1. Interviews of the departmental representatives

We carried out interviews after identify the participating departments in the school in the election process of the Laikipia University Students Organization. The departments were as follows: The department of the Registrar who participate in student registration and provides the list of registered student, the office of Dean of Students who deals with student’s affairs therefore student elections is a direct responsibility, the IT Department which deals with safekeeping of the student’s database.

1. Review of the Laikipia University voting manual

We carried out the review of the University Voting manual as a group and analyzed the requirements needed for the system.

1. Review of the previous used election forms

We collected the 2014/2015 Laikipia University Student Organization election materials such as ballot papers, which we went through the requirement included in the ballot paper so as to enable us design the ballot paper.

1. Interview of the Laikipia University Students existing Organization

We carried out interviews with the Laikipia University student Organization representatives such as the Students Chairperson to help us acquire relevant information about the election process.

1. Personal observation of election process

Being that we have participated in the Laikipia University student Organization elections, we have observed the process and understood what is required in the process of elections.

## **System requirements**

### **Hardware requirements**

Hardware interface include:Multi-Access ESI requires a dedicated PC configured as follows:

1. Any processor speed for Intel, AMD, Celeron processors
2. 512 Megabytes RAM.
3. 40 Gigabyte hard drive.
4. VGA color monitor (256 color capability and 1024 x 768 Resolution; 4MB Video RAM).
5. Networked access to the server containing the Multi-Access database, 32 bit network adapter card/PCI
6. Net Card.

## Software requirements

1. Operating System: Windows XP with SP2 or Windows Vista or Windows OS.
2. Database: PHP MySQL for the system and MS SQL200 for Student registration database.
3. Technologies: PHP, HTML, CSS, J Query, java script.
4. IDE: Notepad ++
5. Browser: Firefox 35.0.

# **SYSTEM ANALYSIS AND DESIGN**

## **Introduction**

This chapter presents the system analysis and design process .This software design document provides a vivid description of the technical design for the Laikipia University Online Voting System. The goal is to fully describe the technical view for how the system and business requirements will be realized. This document provides an architectural overview of the system to depict different aspects that the system covers. This document also functions as the fundamental reference point for developers involved with the system development.

## **Architecture design**

For this system we have chosen the client server model which is characterized by two distinct interacting processes; the client and the server which in general can run on different computers, exchanging data over the network using protocols such as HTTP, HTTPS and TCP/IP etc. The design of online voting system consists of a client who is a voter, webserver and database server. A registered voter connects to a webserver by using his login identification and password. The client and web server communicate by internet using protocols. The relationship is illustrated in the diagram below

Clients

(Web browser) Http/Https TCP/IP

Web

Server

Database server

 http/https



Figure 1: system architecture (en.m.wikipedia.org, 2015)

Clients and servers exchange messages in request-response messaging pattern: The client sends a request and the server returns a response. This exchange of messages is an example of an inter-process communication. To communicate the computers must have a common language, and they must follow rules so that both the client and the server know what to expect. The language and rules of communication are defined in communication protocol.

In this case, when a user accesses online voting services with a web browser (the client), the client initiates a request to the online voting server. The user’s login credentials is stored in a database and the webserver accesses the database server as a client. An application server interprets the returned data by applying the online voting logic, and provides the output to the web server .The application server – through a database connector-handles all database requests from the clients’ side to the database.

## **System analysis**

### **Context diagram**

VOTER

CANDIDATE

page section

Registration number and password

Filled ballot form

Error report

Ballot form

Registration no. and pswrd

Candidate information

Election result statistics

Registration form

Feedbackabout registration

Registration form

Error report

**LAIKIPIA UNIVERSITY STUDENT ORGANIZATION ONLINE VOTING SYSTEM**.

Objective: To make the electoral process, convenient, fast andeconomical.

Election result statistics

Feedbackabout registration

**`**

Login information

Election date

Registratin details

Student registration details

NOMINAL ROLL DATABASE

ADMINISTRATOR

Registration form

Error reports

Candidate information

Election result statistics

Feedbackabout registration

Figure 2: context diagram (Author)

### **Domain analysis**

This software application is designed as an object-oriented system for an Internet-based architecture using three–layer architecture by factoring application classes into the following layers:

1. **The Presentation layer**- This is the layer where the physical window and widget objects live. It will also contain Controller classes as in classical MVC. Any new user interface widgets developed for this application are put in this layer.
2. **The Domain Mode**- Most objects identified in the OO analysis and design will reside. To a great extent, the objects in this layer can be application-independent. Generic objects may be used in this application to reap the benefits of Object Oriented programming.
3. **The Data layer** - The data is managed by MySQL.

The following entities were identified for the system

* Voter
* Candidate
* Ballot system
* Administrator
* System database
* Registration database
* Vote

The following use cases were identified:

* Voter use case
* Candidate use case
* Administrator use case

### **Use case models**

djfhej

<<INCLUDE>>

<<extends>>

ADMINISTRATOR

Figure 3: Administrator use case (Author)

<<INCLUDE>>

<<Extend>>

<<EXTEND>>

<<EXTEND>>

<<Extend>>

<<INCLUDE>>

<<EXTEND>>

<<Extend>>

<<EXTEND>>

CANDIDATE Extend<<extend>>

<<EXTEND>>

<<Extend>>

Figure 4: Candidate use case(Author)

<<INCLUDE>>

<<EXTEND>>

<<EXTEND>>

<<EXTEND>>

<<INCLUDE>>

<<INCLUDE>>

VOTER <<EXTEND>>

<<INCLUDE>>

Figure 5: Voter use case(Author)

### **Entity relationship diagram and class diagram**

Is added to candidate

Voter

Vote

Casts

Administrator

Candidate

Responds to error message

Receive registration form from candidate.

Voter votes for candidate

Applies for seat

Seat

Included in vote

Vote

Contains

1 1

1...\*

1...\* 1…\*

1 1...\* 1

1 1…\*

1…\*

1…\*

1 1

1

1…\*

1

Figure 6: Entity relationship diagram(Author)

|  |
| --- |
| Voter |
| +voterName  -voterReg  #password |
| #Registration()  +saveInSystemDatabase()  #login()  -voterVerification()  -feedbackMessage()  #castVote()  submitVote()  viewResults()  Logout() |

|  |
| --- |
| User |
| +username  #userId  #password |
| -login()  +viewPage()  +viewResults()  -logout() |

|  |
| --- |
| Candidate |
| +candidate Name  #candReg  #password  +seat  +profile |
| Registration()  saveInSystemDatabase()  login()  updateProfile()  castVote()  viewResults()  Logout() |

|  |
| --- |
| Post |
| +position  +positionId  +candidateName |
| fillposition()  countCandidates() |

|  |
| --- |
| Ballot System |
| +noVoted  +votePerCandidate  +votePerposition  +winnersPerPosition |
| countVote()  analyzeVote()  displayWinner() |

|  |
| --- |
| Vote |
| -voteId  +seat  +candidateName |
| submitVote()  verifyVote() |

|  |
| --- |
| Administrator |
| +adminName  #adminNumber  #password |
| updateElectionDates()  respond toErrors()  viewResults  announceWinner() |

Figure 7: Class diagram (Author)

### **Sequence diagrams or data flow diagrams**

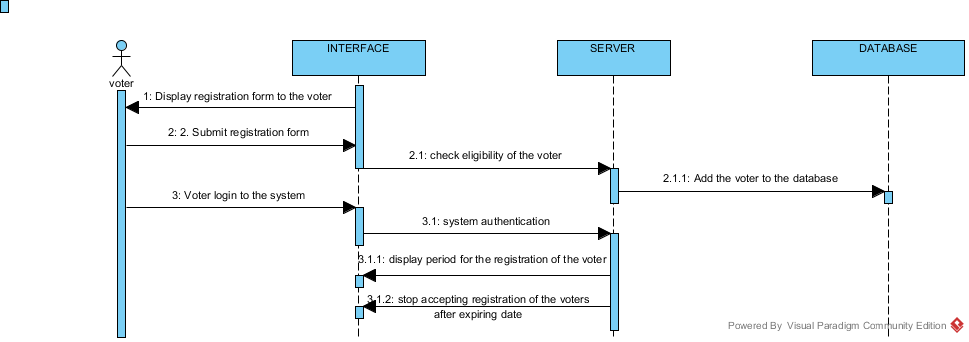


Figure 8: Voter sequence diagram(Author)

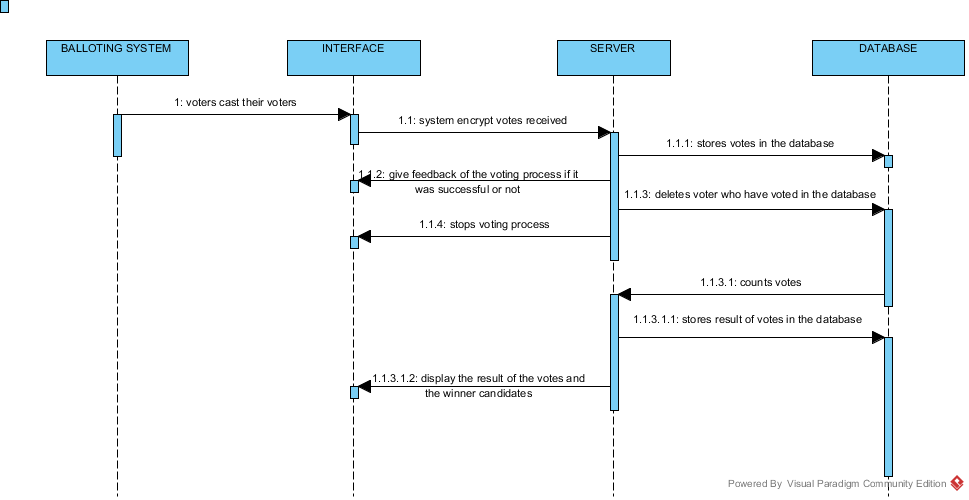


Figure 9: Ballot sequence diagram(Author)

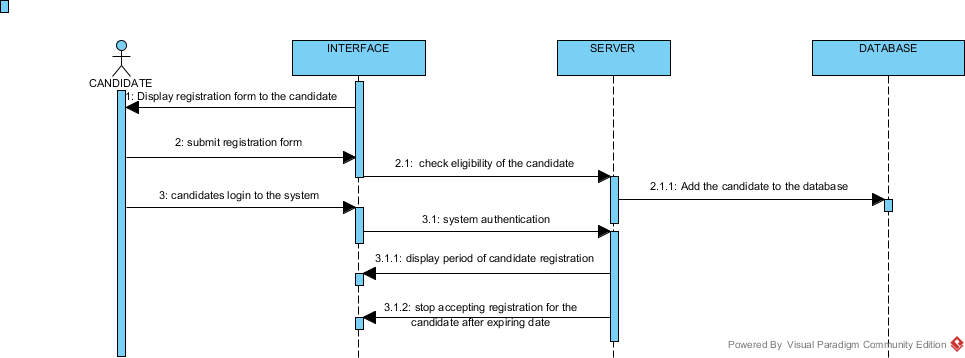


Figure 10: Candidate sequence diagram(Author)

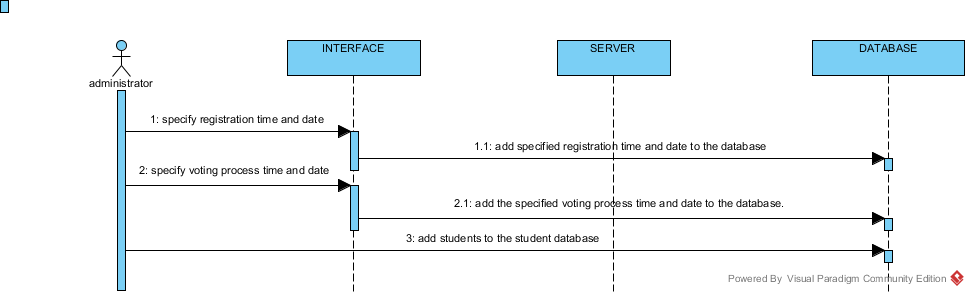


Figure 11: Administrator sequence diagram (Author)

## **Database design**

|  |  |  |
| --- | --- | --- |
| **VOTER** |  |  |
| Voter\_Name | Varchar |  |
| Voter\_Reg | Int | Primary key |
| Voter\_Password | Varchar | Unique |
| Voter\_Programme | Varchar |  |
| Voter\_Year | Int |  |
| Voter \_Sem | Varchar |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **CANDIDATE** |  |  |
| Cand\_name | Varchar |  |
| Cand\_regnumber | Int | Primary key |
| Cand\_password | Int | Unique |
| Cand\_profile | Varchar |  |
| Year | Int |  |
| Course | Varchar |  |
| Seat | Varchar |  |
| Cand\_profile | Varchar |  |

|  |  |  |
| --- | --- | --- |
| **ADMINISTRATOR** |  |  |
| Admin\_ id | Varchar | Primary key |
| Admin\_name | Varchar | Unique |
| Admin\_password | Varchar | Unique |
| Age | Int |  |
| Admin\_salary | currency |  |
| Phone | Int |  |
| Date/time s | Date/time | s |

|  |  |  |
| --- | --- | --- |
| **VOTE** |  |  |
| Vote\_ID | Int | Primary key |
| \*Voter \_registration number | Int |  |
| Date/time | Date/time |  |
| Seat | Varchar | Unique |
| Cand\_name | Varchar |  |
| Cand\_profile | Varchar |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Voter | Candidate | Vote | Administrator |
| Administrator | 1:M | 1:M | 1:M |  |
| Candidate | M:N |  | 1:1 | M:1 |
| Voter |  | M:N | 1:1 | M:1 |
| Vote | 1:1 | 1:1 |  | M:1 |

Relationships

# **CHAPTER FIVE**

# **SYSTEM IMPLEMENTATION AND TESTING**

# **5.1 Introduction**

System implementation and testing are two vital phases without which the system cannot be released for use. System testing is important because it ensures that all processes are according to the specifications. The logical and physical design is continuously examined to ensure that the test data are verified for correctness and accuracy. Duringtesting two kind of errors are likely to occur: syntactic errors and logical errors. Syntactic errors occur when a program statement violates one or more rules of the language in which the programs are written in, thiserrors have to be corrected before the program is finally execute. The compilation of the program does not show any errors if the syntax errors are corrected. But when the system is run and the desired output does not appear this id due to logical errors. The errors can be corrected by asking the program to display a number of message statements. Different modules would be tested individually and are made error free. The implementation and tests on the proposed LUSO online election system are presented in this chapter

## **Summary of modules**

The main modules of Laikipia University Student Union Online Voting System are:

* Voter module
* Candidate module
* System administrator module
* Ballot Module
* Real-time vote statistics module

### **Voter module**

This module is for the voter where he or she must first register his/her details first into the registration form fulfilling all the required specification. The required fields are: voter\_name,voter\_ reg, course, year and faculty. Before the voter is registered he/she must be verified as a bonified student, this is done by searching the nominal roll database for her record if he/she exists in the nominal roll database .If all validations are valid then her/his information is registered into the registration database of the system.

### **Candidate module**

In this module for a candidate to apply for candidacy, they must first register. They are provided with registration feature where they can enter their details in the registration form fulfilling all the specifications .The fields required include: cand\_name, cand\_reg, Course, year, faculty, position and cand\_profile. Authentication and validation of the candidate is done next where searching of her records in the nominal roll database is done. If her records exists her information is saved in the candidates table in the registration database. After a candidate is registered a login feature is made available to him/her where she/he can log into the system using the cand\_reg and password which allows him access into the system if all her inputs are valid.

The shortlisting of approved candidates is done by the system administrator.

### **System administrator module**

This module provides features for administrator/admin registration, login, and shortlisting of approved candidates. The system administrator must first register by filling her details in the registration form. Her information is the saved in the admin table in the registration database. If registration is successful the system administrator is provided with a login feature where she/he can access the applications from candidates. The admin forwards the lists of potential candidates to the University administration for approval after which he/she shortlists the viable candidates by entering their details and names in the registration database.

The admin has the privilege of viewing registered voters or searching for a voter entering his details, he is also provided with the feature.

### **Ballot Module**

In this module the ballot forms are predesigned and the administrator is provided with a feature for inserting candidate names into the ballot for respective categories (electoral position).

The registered voter can access the ballot form by logging in using their registration number and password. The voter can then click on the categories to select a candidate for each position. The candidate can reselect again to edit her choice before submitting her vote.

### **Real-time vote statistics module**

This module provides a feature for automatic calculation of votes cast by voters in real time. The votes are summarized according to the number of votes each candidate gets and reports about the winning candidate for each position is provided.

The candidate, voters and system administrator are given the privilege of viewing the vote statistics by allowing them to login using their valid login details.

## **Summary of how the system works**

Laikipia university student union online voting system works in the following way: The first event is the registration of the system administrator, he/she is presented by a registration form where he fills in his/her registration details according to the specifications in the form after successful registration he is provided with a platform to create his/her login account to allow him/her to access the system parts specified for him.The system administrator updates information about the start and end date for voter registration, start and end date for campaign and the actual voting date on the system’s website after which the registration process of the voter and candidates commences,among the activities that take place during registration include: the voters type the LUSO online voting system address into the browser, the voter clicks the registration page to open an online registration form him/her to register. The voter enters his or her details into the form according to the fields specification in the form and clicks submit, the voters details are authenticated and verified to see if he or she exists in the nominal roll database of the university if her records are present a message is sent back to him informing him that the registration is successful her/his details are saved in the system database as a voter. The voter is then provided with a platform for a voter to create a login and logout account using a username and a secret password, if the registration process is not successful a message is displayed informing him that he should check if he has signed the nominal roll.

Application for candidacy for various positions commences according to the specified dates right after voter registration. All interested candidates open the registration form for candidates and fill their details in the form according to the forms specifications and submits their application. If they exists in the nominal roll database of the university, a message is sent back to them informing them that the registration is successful. Their details are saved in the system database as candidates. The list of all candidates who applied for various position are displayed. After registration the candidates are provided with a platform to create their personal login in accounts which they are to use to access the specified pages of the website. Candidate validation ,verification and shortlisting is done manually by Dean of students who then provides a list of shortlisted candidates to the system administrator who then enters the list of shortlisted candidates in the system database and displays the names of the shortlisted candidates and the position they are vying for.

When the election dates are announced, the registered voters can remotely login into the website using their username and passwords as long as they have internet access and access the ballot forms. To vote the voter clicks the radio button against the name of the preferred candidate for the specified position e.g. President. The voter does this for all available positions and after successfully filling her preference for every position he or she can change her choice as many times as possible before submitting his or her vote. One a voter has voter has voted her login details are deleted from the system database to prevent her from voting twice.

The counting of votes starts as soon as the voting process begins. Thereis automatic calculation of votes cast by voters in real time. The calculation of the votes stops when the voting dates and hour are over. The votes are summarized according to the number of votes each candidate gets and reports about the winning candidate for each position is provided. The candidate, voters and system administrator are given the privilege of viewing the vote statistics by allowing them to login using their valid login details.

**The home page**

This is the home page that is accessible to the Voters and the candidates. The voters are provided with the option to register to be a voter and login to vote their choices. The candidates can also apply online and view the shortlisted candidates.

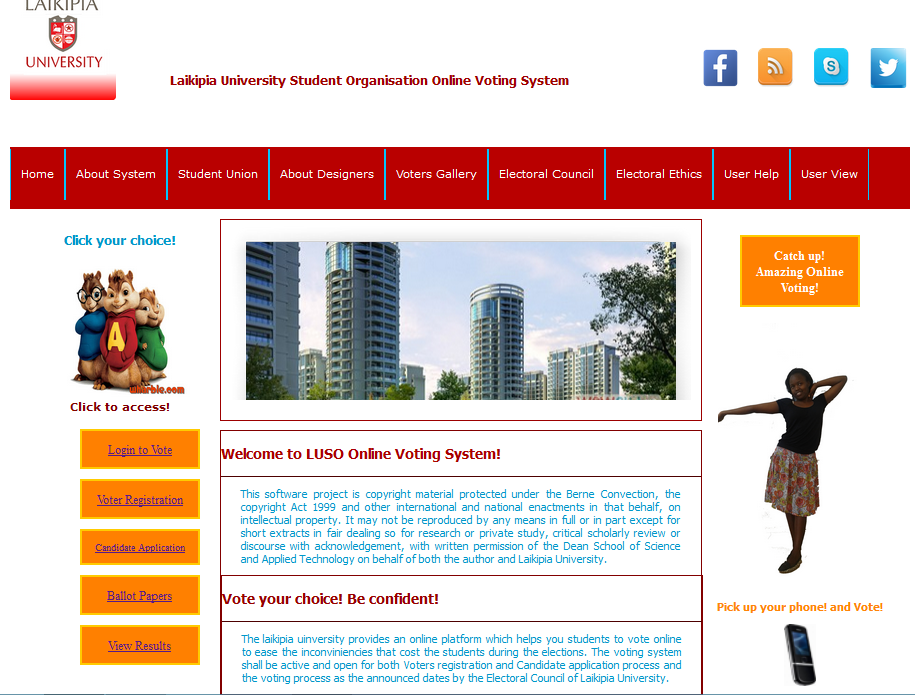


Figure 12: Home page (author)

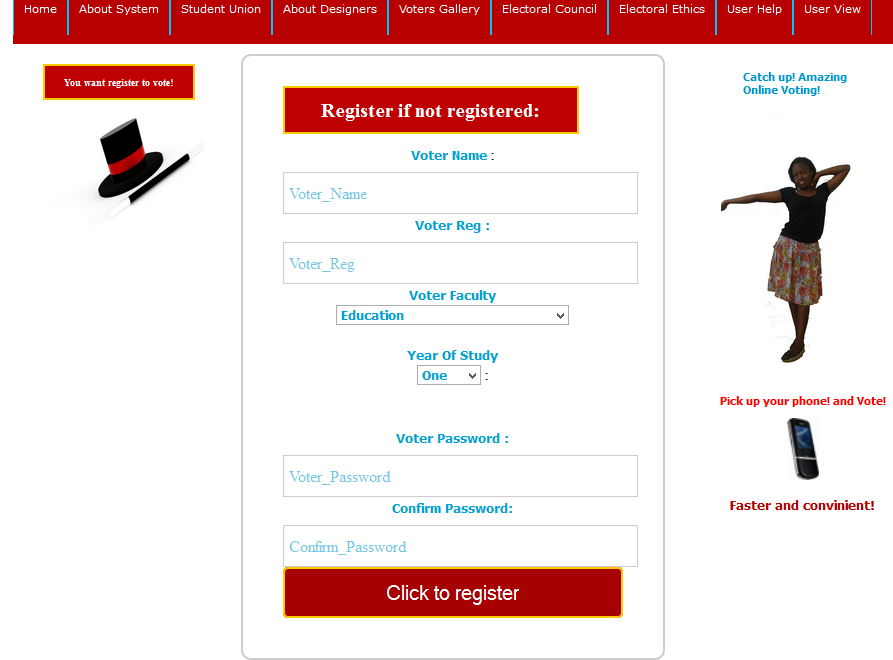
Voter registration form: This is the form provided for the bornified students of laikipia university to register to be able to vote online. 

Figure 13: Voter registration page (author)

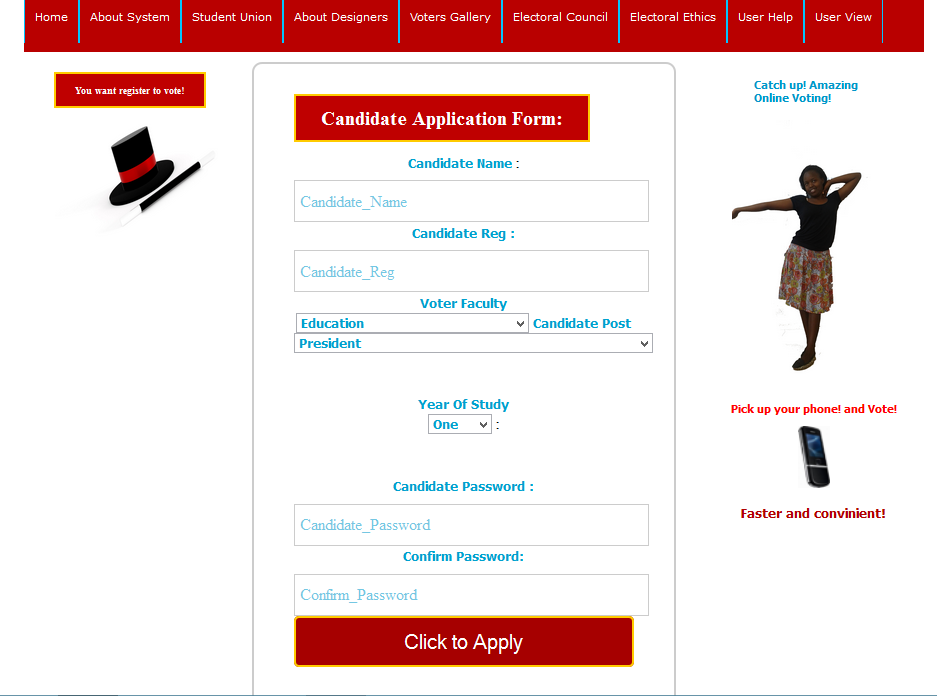
Candidate application form: This is the form provided for candidates to apply for the various posts in the Laikipia University Student Organization.

Figure 14: Candidate Application (Author)

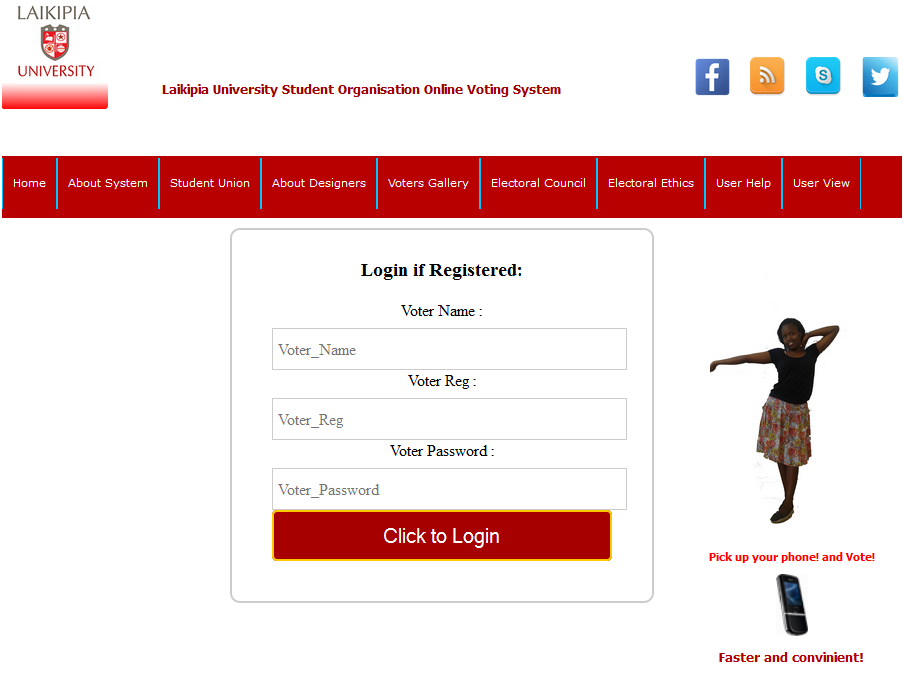
Voter Login form: This restricts a voter to provide for registered password so as to access the ballot paper. 

Figure 15: Voter Login page (Author)

Ballot papers: This provides the voter with the options to select their desired candidates and submit the votes. The ballot papers also provides the voters with the list of candidates verified and shortlisted. The ballot papers are in an ordered step as follow:

Process 1: Presidential election ballot: Which provides for options for the presidential candidates.

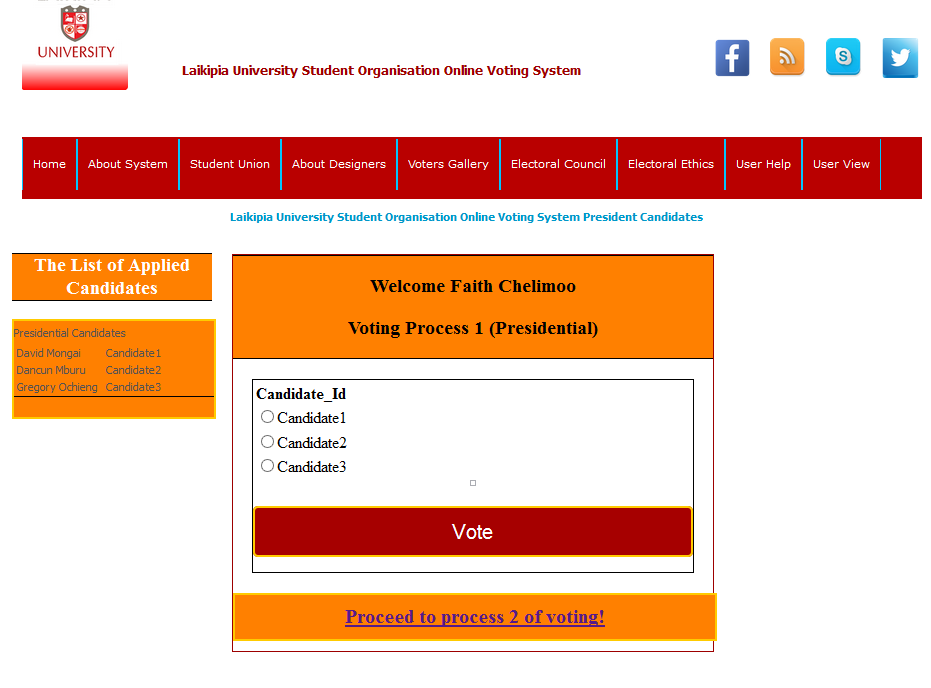


Figure 16: Ballot paper process 1 (Author)

Process 2: Vice President elections: Provides for Vice President options.



Figure 17: Ballot paper process 2 (Author)

Process 3: Finance election ballot: Provides for finance candidates options.

Figure 18: Ballot paper process 3 (Author)

Process 4: Wefare election ballot: Provides for welfare candidate options

Figure 19: Ballot paper process 4 (Author)

Process 5: Academics election ballot: Provides for academics candidates options.

Figure 20: Ballot paper process 5 (Author)

Process 6: Sports and Entertainment election ballot: Provides for the names candidates options.

Figure 21: Ballot paper process 6 (Author)

Process 7: Secretary General elecetion ballot: Provide for secretary general candidates options.

Figure 22: Ballot paper process 7 (Author)

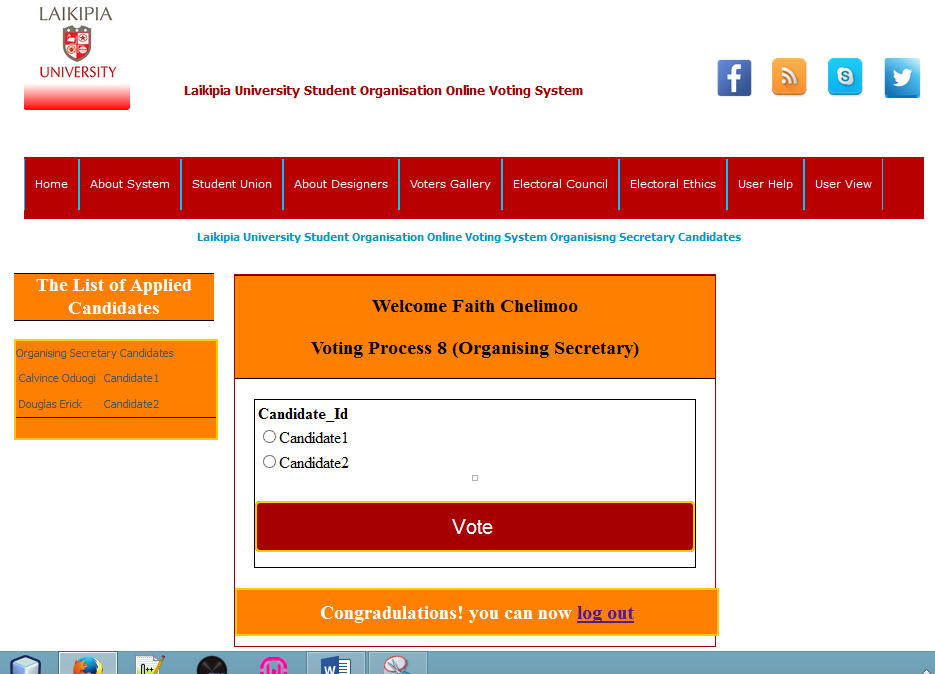
Process 8: Organising Secretary election ballot: Provide for the secretarygeneral candidates options.

Figure 23: Ballot paper process 8 (Author)

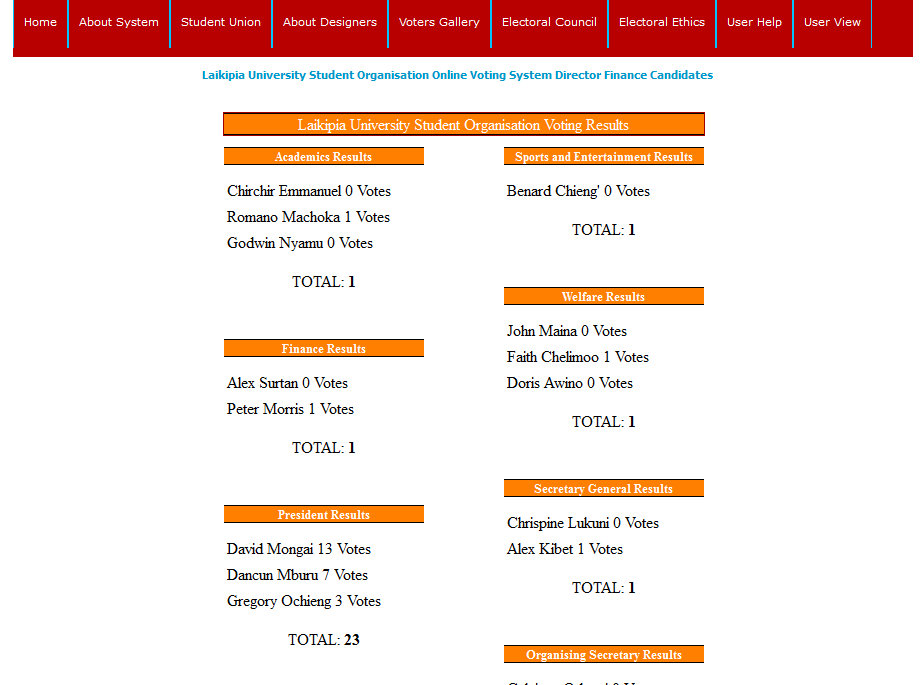
Voting statistics: This is the page that both the voters and the candidates can view the voting results.

Figure 24: Vote results (Author)

Administrator registration: This is the form provided for the administrator to register to be given the access authority to the admiinistrator panel.

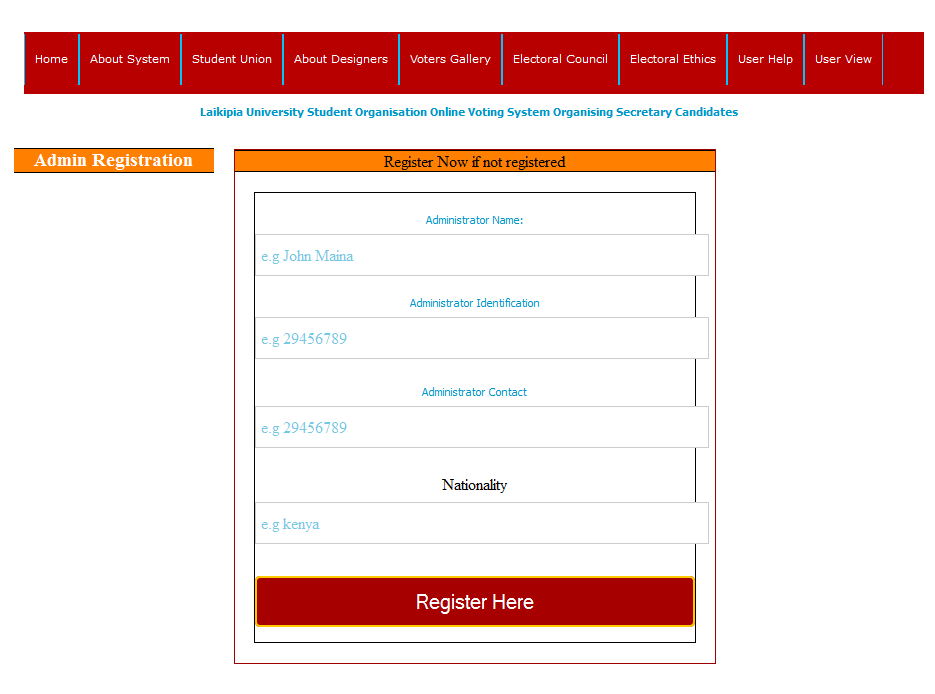


Figure 25: Administrator registration page (Author)

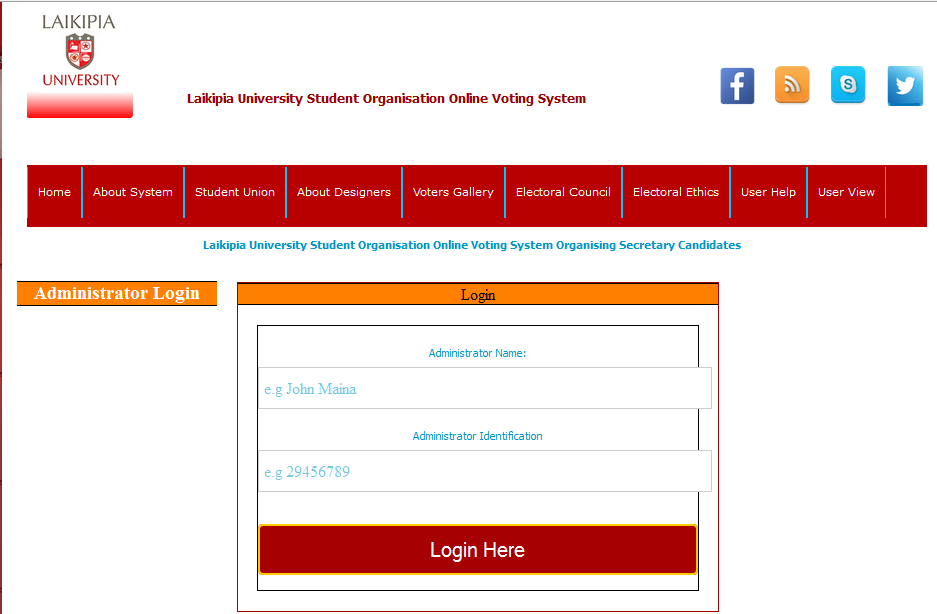
Administrator login: This is the page that the admin is restricted to access the administrator panel after providing their registration details i.e Administrator Name and Administrator Identification. 

Figure 26: Administrator Login (Author)

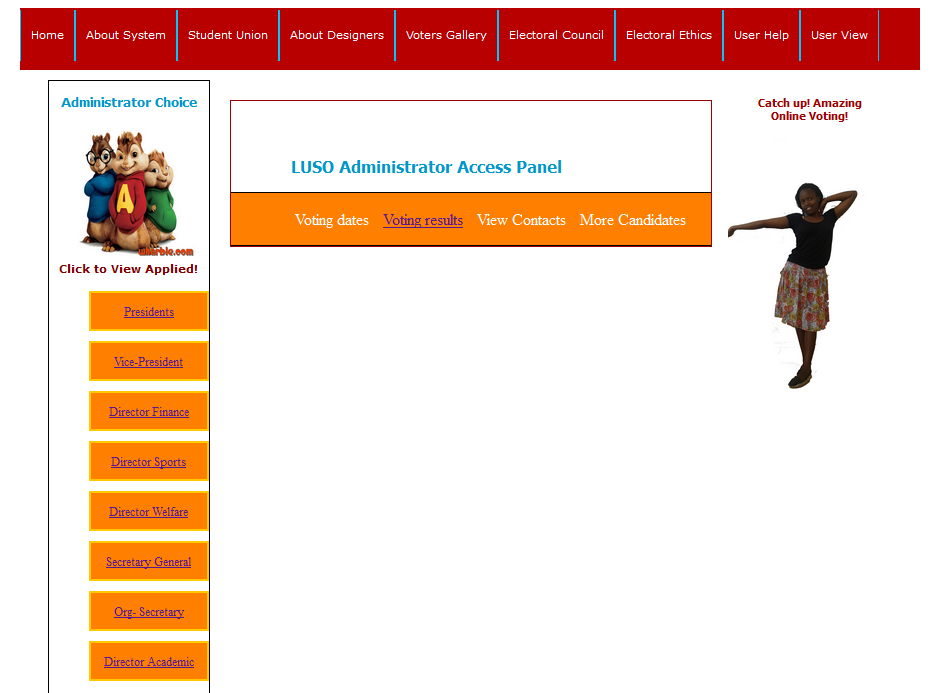
Administrator panel: This is the administrator panel that the administrator uses to access his or her duties to add candidates who have been verified by the Electoral commission.

Figure 27: Administrator panel (Author)

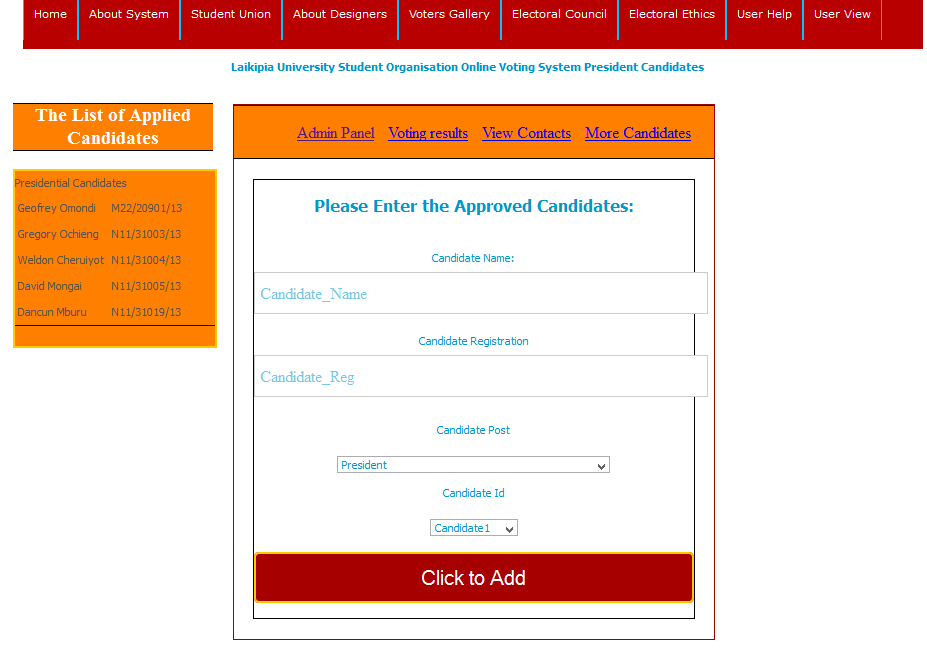


Figure 28: Add candidates page (Author)

Add candidates form: These forms include eight pages for adding candidates in the eight post. Each page provides the list of the candidates who have applied for vcarious posts.

Exapmle below shows a page to add the candidates who have applied for presidential post.

## **Test Regime**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test id | Description | Test data | Expected results | Actual results |
| 1 | Voter registration |  | Registration successful. Voter records updated in voter table in the registration database | Registration successful |
| 2 | Login |  | Allow access into the system | As expected |
| 3 | Candidate registration |  | Registration successful. Candidate records updated in candidate table in the registration database. | As expected |
| 5 | Candidate shortlisting |  | List of shortlisted candidate is displayed | As expected |
| 5 | Voter validation |  | Voter is allowed to register | Registration successful |
| 6 | Voting |  | Voter selects one candidate per position and submit vote | Voting successful |
| 7 | Vote statistics |  | Display real time vote count per candidate | As expected |

Figure 29: Test regime (Author)

## **Conclusion**

Overwhelming evidence and research has shown than electronic/online voting systems can guarantee a credible and reliable elections (I.J.Computer Networks and Information Security, 2013), with real time result production without any possibility of interference with election results. This proposed online voting system will place our democracy on the success path. This system also aims to polish the image of Laikipia University Student Union Election system by making it more integral and trustworthy .It is observed that a lot of problems and inconsistency that arise from manual voting system can result into serious manipulation of votes, which in turn can cause conflict, disagreement and mistrust in the leadership in place.

In conclusion careful planning and implementation of LUSO online election system and careful engineering of the system hardware and software; the aspect of free and fair election and in result good leadership can be achieved by adoption of this system.

## **Recommendation**

After a lot of research many challenges associated with the manual voting system used in Laikipia University currently came to our attention but we are certain that these challenges could be solved if the online voting system is adopted. We therefore recommend the following: There should be adoption of distributed encryption techniques for the purpose of secure data transmission, there should be adequate and proper user or public enlightenment on how to use the system, precautions to take when using the system and the benefits of using online voting system before the system is implemented. We also recommend that more research should be carried out on this area to realize more improvements especially in areas such as voter authentication and database security that can be made on our proposed system. Finally we recommend that Laikipia University should take up this project as it will be greatly beneficial to the institution in many ways as stated in the earlier chapters of this document and students and the University administration should not shy away from trying to interact with it.

# **REFERENCES**

1. *Client-Server model*(n.d), retrieved 16th, April, 2015, ell.m.wikipedia.org.

2*.* Dictson&Ray (2000), *Non-technical Risk of Remote Electronic Voting*.

p.3, p.4, and p.g5

3. I.J (2013) *Computer Network and Information Security*, Retrieved on 16th, April, 2015

Published online (April, 2013), in MECS.

http://wwmecs-press.og

4. Westernaus (23rd, July, 2012*). Survey and Ballot systems*. Retrieved on 14th, April, 2015

www.surveyandballotsystems.com.

# **APPENDICES**

# **Appendix 1: Questionnaire**

Dear respondent.

We would wish to ease one of the most vital activities in the university that involves students, admin, security and other relevant departments in the school, Laikipia University Student Organization Elections by computerizing the process having a design of an Electronic Electoral System. The completion and success of the design e-polling web application for the Laikipia Student Organization Election would be a success of our Team Work Project Course a Degree requirement in Bask. Computer Science major. We would therefore ask to source information from you to enable us acquire enough requirements to aid in the design of a satisfactory system.

**Student voter:**

**(*Please give a tick for you selection*)**

**Personal Details:**

1. Are you a student of Laikipia University?

Yes

No

If yes, what level of study?

Certificate

Diploma

Undergraduate

Masters

1. Please specify the year of study (1) (2) (3) (4)

**Technological Details:**

1. Do you have an access to the internet?

Yes

No

If yes, how often?

Every day

Three times a week

Twice a week

Once a week

1. What gadget do you use to access the internet?

Mobile phones

Personal computer

Cyber computers

1. How reliable is the internet?

**System Details:**

1. Have you ever participated in Laikipia University Student Organization Elections?

Yes

No

If no, explain why

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

If yes, how many times?

4

3

2

1

1. What kind of system do you use for elections?

Manual system

E-polling system

If other please list

…………………………………………………………………………………………………………………………………………………………………

1. Please provide the rating of the system efficiency.

Excellent

Very good

Good

Poor

Very poor

1. Are there challenges you have experienced with the current electoral system?

Yes

No

If yes, please state a few challenges.

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………...

1. Suggest any improvement that you may wish to be implemented

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

1. Would you prefer an online electoral system to be implemented for LUSO elections?

Yes

No

If no, explain

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

1. Are you familiar with an online electoral system?

Yes

No

If yes, state example…………………………………………………………………….

1. What would you like to be included and improved with the online electoral system?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Student Candidate**

1. Have you ever applied for LUSO position?

Yes

No

If yes, how?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. Did you experience any difficulty in the application process?

Yes

No

If yes, how?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

1. What difficulties did you experience with the current electoral system?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

1. What improvements would you like to be implemented?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

1. Do you prefer an electronic electoral system?

Yes

No

1. What would you wish to be solve day the system?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..

(Thank you for the support and cooperation)

**Office of Student Registrar**

1. What kind of database do you use managing the registered student information upon student signing the nominal role?

Traditional database(paper work)

Access data bases

Oracles databases

My SQL databases

1. During student registration, please tick to confirm the details required for the student registration database.

studentName

studentRegno

studentProgramme

studentContacts

studentIdno

studentCounty

if any other details please include

........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Are you involved in the Laikipia University Student Organization electoral process?

Yes

No

If yes, how?

...........................................................................................................................................................

...........................................................................................................................................................

...........................................................................................................................................................

...........................................................................................................................................................

1. Do you experience any difficulties with the exercise?

Yes

No

If yes, explain

...........................................................................................................................................................

...........................................................................................................................................................

...........................................................................................................................................................

...........................................................................................................................................................

1. Do you prefer an electronic electoral system for luso elections?

Yes

No

If no, why?

...........................................................................................................................................................

...........................................................................................................................................................

1. What would you wish to be improved with the electronic electoral system?

...........................................................................................................................................................

..............................................................................................................................................

(thank you for your cooperation and support)

**Ofice of Dean of Students**

1. Are you involved in LUSO elections?

Yes

No

If yes, how?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. What are you roles in the LUSO elections?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. What is involved in the candidate verifaction and validations?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. If any, what are the references involved in acquiring these inforamtion?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Do you have any form of recod keeping for candidate verificatiion?

Yes

No

If any, what details are involved?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. What improvements would you like to be made on the current electoral system?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

1. Are you satisfied with the current electoral system?

Yes

No

1. Would you prefer an electronic electoral system for the LUSO elections?

Yes

No

If yes, what integrity issues would you wish to be adhered to?

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

(thank you for yournsupport and cooperation)

**Security Department:**

1. Are you involved in the Laikipia University Student Organization Elections?

Yes

No

If yes, how?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………….

1. Do you have any difficulties performing your roles during LUSO elections?

Yes

No

If yes, explain

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

1. In terms of security, what would wish to be implemented?

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

(Thank you for your support and cooperation)

## **Appendix 2: Source codes**

**Codes for voter registration form:**

<?php

session\_start();

?>

<!DOCTYPE html >

<html xmlns="http://www.w3.org/1999/xhtml">

<head><title>i love you fayee</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="style/stylevoter.css" rel="stylesheet" type="text/css" />

<link href="style/style1.css" rel="stylesheet" type="text/css" />

<link href="style/style2.css" rel="stylesheet" type="text/css" />

<link href="style/styleslider.css" rel="stylesheet" type="text/css" />

<!--[if IE]>

<link href="style/style-ie.css" rel="stylesheet" type="text/css" />

<![endif]-->

<style type="text/css">

<!--.style3 {color: #9D0000}-->

</style>

<style type="text/css" media="all">

<!--@import url("\*");

.style5 {font-size: 10px}

.style16 {font-size: 12}

.style18 {

color: #FF0000;

font-weight: bold;

}

.style19 {color: #B00000}

.style20 {font-size: 12px}

-->

</style>

</head>

<body>

<center>

<div class="outer">

<div class="wrapper">

<div class="logo"><strong></strong>

<div class="picture1">

<img src="images/Logo.png" width="106" height="106">

</div>

<div class="laikipia">

<div class="name">

<h3 align="center" class="gallery-name style3">Laikipia University Student Organization Online Voting System</h3>

</div>

<div class="social">

<ul>

<li><img src="icons/Facebook-icon.png" width="37" height="40"/></li>

<li><img src="icons/Rss-icon.png" width="37" height="40"/></li>

<li><img src="icons/Skype-icon.png" width="37" height="40"/></li>

<li><img src="icons/Twitter-icon.png" width="37" height="40"/></li>

</ul>

</div>

</div>

<div class="menu">

<ul class="solidblockmenu">

<li><a href="indexvoter.php">Home</a></li>

<li><a href="#">About System </a></li>

<li><a href="#">Student Union </a></li>

<li><a href="#">About Designers </a></li>

<li><a href="#">Voters Gallery </a></li>

<li><a href="#">Electoral Council </a></li>

<li><a href="#">Electoral Ethics </a></li>

<li><a href="#">User Help </a></li>

<li><a href="#">User View </a></li>

</ul>

<div class="clear"></div>

</div>

</div>

<div class="body-wrapper">

<div class="body-right">

<blockquote>

<h3 align="left" class="product-price">Catch up! Amazing Online Voting!</h3>

</blockquote>

<img src="images/fayee2.jpg" width="163" height="266"/>

<h3 class="product-compare style18">Pick up your phone! and Vote!</h3>

<img src="images/mpbile.jpg" width="32" height="62"/>

<h3 class="product-title style19">Faster and convinient!</h3>

</div>

<div class="body-middle">

<div id="login">

<form action="voterfinal.php" method="post">

<h3 class="sitail">Register if not registered: </h3>

<label><span class="gallery-name">Voter Name </span>:</label>

<input name="Voter\_Name" type="text" class="product-price" id="Voter\_Name" placeholder="Voter\_Name">

<label class="gallery-name">Voter Reg :</label>

<input name="Voter\_Reg" type="text" class="product-price" id="Voter\_Reg" placeholder="Voter\_Reg">

<label for="label"><span class="gallery-name">Voter Faculty</span><br>

</label>

<select name="Voter\_Faculty" class="gallery-name" id="select">

<option>Education</option>

<option>Science and Applied Technology</option>

<option>Humanity Development Studies</option>

<option>Business and Economics</option>

</select>

<label></label>

<label for="select"><br><br>

<span class="gallery-name">Year Of Study</span><br>

</label>

<select name="Voter\_Year" class="gallery-name" id="select">

<option>One</option>

<option>Two</option>

<option>Three</option>

<option>Four</option>

</select>

<label>:</label>

<p class="gallery-name">&nbsp; </p>

<span class="gallery-name">

<label>Voter Password :</label>

</span>

<input name="Voter\_Password" type="password" class="product-price" id="Voter\_Password" placeholder="Voter\_Password">

<label class="gallery-name">Confirm Password:</label>

<input name="Confirm\_Password" type="password" class="product-price" id="Confirm\_Password" placeholder="Confirm\_Password">

<input name="submit" type="submit" class="h3" value=" Click to register ">

</form>

</div>

</div>

<div class="body-left">

<h3 class="sitail style5">You want register to vote! </h3>

<img src="images/cute\_magician-1503646-1920x1080.jpg"width="170" height="137"/></div>

<div class="footer">

<h3 align="center" class="product-compare">Copyright @2015 LimooDev Software creators</h3>

<h3 align="center" class="product-compare">Designed by: Ya'kov </h3>

</div>

</div>

</body>

</html>

<?php

//define the connection credentials

$user="root";

$host="localhost";

$dbname1="studentregistrationdb";

$dbname2="voterregistration";

$votername=$\_POST['Voter\_Name'];

$voterreg=$\_POST['Voter\_Reg'];

$voterfaculty=$\_POST['Voter\_Faculty'];

$voteryear=$\_POST['Voter\_Year'];

$voterpassword=$\_POST['Voter\_Password'];

$confirmpassword=$\_POST['Confirm\_Password'];

//db connection

try

{

$dbname1= new PDO("mysql:host=$host;dbname=$dbname1", $user);

$count=$dbname1->prepare("select Student\_Reg from studentsnominal where Student\_Reg=:Student\_Reg");

$count->bindParam(":Student\_Reg",$voterreg);

$count->execute();

$no=$count->rowCount();

if($no >0 )

{

try

{

$dbname2= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$count=$dbname2->prepare("select Voter\_Reg from voterregistration where Voter\_Reg=:Voter\_Reg");

$count->bindParam(":Voter\_Reg",$voterreg);

$count->execute();

$no=$count->rowCount();

if($no==0 )

{

$sql="INSERT INTO voterregistration (Voter\_Name, Voter\_Reg, Voter\_Faculty, Voter\_Year, Voter\_Password, Confirm\_Password) VALUES(:Voter\_Name, :Voter\_Reg, :Voter\_Faculty, :Voter\_Year, :Voter\_Password, :Confirm\_Password)";

$query=$dbname2->prepare($sql );

$query->execute(array(':Voter\_Name'=>$votername, ':Voter\_Reg'=>$voterreg,':Voter\_Faculty'=>$voterfaculty, ':Voter\_Year'=>$voteryear, ':Voter\_Password'=>md5($voterpassword), ':Confirm\_Password'=>md5($confirmpassword) ));

echo'<script type="text/javascript">window.alert("You are successfully registered!")

</script>';

}

else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("You have already registered as a voter!")

</script>';

}

}

catch(PDOException $e)

{

die('Could not connect to the database:' . $e);

}

}

else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("Check if you have signed the nominal roll!")

</script>';

}

}

catch(PDOException $e)

{

die('Could not connect to the database:' . $e);

}

?>

**Candidate application form:**

<?php

session\_start();

?>

<!DOCTYPE html >

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>LUSO ONLINE VOTING SYTEM</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="style/stylevoter.css" rel="stylesheet" type="text/css" />

<link href="style/style1.css" rel="stylesheet" type="text/css" />

<link href="style/style2.css" rel="stylesheet" type="text/css" />

<link href="style/styleslider.css" rel="stylesheet" type="text/css" />

<!--[if IE]>

<link href="style/style-ie.css" rel="stylesheet" type="text/css" />

<![endif]-->

<style type="text/css">

<!--

.style3 {color: #9D0000}

-->

</style>

<style type="text/css" media="all">

<!--

@import url("\*");

.style5 {font-size: 10px}

.style16 {font-size: 12}

.style18 {

color: #FF0000;

font-weight: bold;

}

.style19 {color: #B00000}

.style20 {font-size: 12px}

-->

</style>

</head>

<body>

<center>

<div class="outer">

<div class="wrapper">

<div class="logo"><strong></strong>

<div class="picture1">

<img src="images/Logo.png" width="106" height="106"></div>

<div class="laikipia">

<div class="name">

<h3 align="center" class="gallery-name style3">Laikipia University Student Organization Online Voting System</h3>

</div>

<div class="social">

<ul>

<li><img src="icons/Facebook-icon.png" width="37" height="40"/></li>

<li><img src="icons/Rss-icon.png" width="37" height="40"/></li>

<li><img src="icons/Skype-icon.png" width="37" height="40"/></li>

<li><img src="icons/Twitter-icon.png" width="37" height="40"/></li>

</ul>

</div>

</div>

<div class="menu">

<ul class="solidblockmenu">

<li><a href="indexvoter.phpl">Home</a></li>

<li><a href="#">About System </a></li>

<li><a href="#">Student Union </a></li>

<li><a href="#">About Designers </a></li>

<li><a href="#">Voters Gallery </a></li>

<li><a href="#">Electoral Council </a></li>

<li><a href="#">Electoral Ethics </a></li>

<li><a href="#">User Help </a></li>

<li><a href="#">User View </a></li>

</ul>

<div class="clear"></div>

</div>

</div>

<div class="body-wrapper">

<div class="body-right">

<blockquote>

<h3 align="left" class="product-price">Catch up! Amazing Online Voting!</h3>

</blockquote>

<img src="images/fayee2.jpg" width="163" height="266"/>

<h3 class="product-compare style18">Pick up your phone! and Vote!</h3>

<img src="images/mpbile.jpg" width="32" height="62"/>

<h3 class="product-title style19">Faster and convinient!</h3>

</div>

<div class="body-middle">

<div id="login">

<form action="candidatefayee.php" method="post">

<h3 class="sitail">Candidate Application Form: </h3>

<label><span class="gallery-name">Candidate Name </span>:</label>

<input name="Candidate\_Name" type="text" class="product-price" id="Candidate\_Name" placeholder="Candidate\_Name">

<label class="gallery-name">Candidate Reg :</label>

<input name="Candidate\_Reg" type="text" class="product-price" id="Candidate\_Reg" placeholder="Candidate\_Reg">

<label for="label"><span class="gallery-name">Voter Faculty</span><br>

</label>

<select name="Candidate\_Faculty" class="gallery-name" id="select">

<option>Education</option>

<option>Science and Applied Technology</option>

<option>Humanity Development Studies</option>

<option>Business and Economics</option>

</select>

<label></label>

<label for="label"><span class="gallery-name">Candidate Post</span><br>

</label>

<select name="Candidate\_Post" class="gallery-name" id="select">

<option>President</option>

<option>Vice President</option>

<option>Director Finance</option>

<option>Director Academics</option>

<option>Director Welfare</option>

<option>Director Sports and Entertainment</option>

<option>Secretary General</option>

<option>Organising Secretary</option>

<option>Executive Secretary</option>

<option>School of Science Rep/option>

<option>School of Education Rep</option>

<option>School of Business and Economics Rep</option>

<option>School of Humanities and Development Studies Rep</option>

<option>First years Rep</option>

<option>Second Years Rep</option>

<option>Third Years Rep</option>

<option>Fourth Years Rep</option>

<option>Non-Residence Rep</option>

</select>

<label for="select"><br>

<br>

<span class="gallery-name">Year Of Study</span><br>

</label>

<select name="Candidate\_Year" class="gallery-name" id="select">

<option>One</option>

<option>Two</option>

<option>Three</option>

<option>Four</option>

</select>

<label>:</label>

<p class="gallery-name">&nbsp; </p>

<span class="gallery-name">

<label>Candidate Password :</label>

</span>

<input name="Candidate\_Password" type="password" class="product-price" id="Candidate\_Password" placeholder="Candidate\_Password">

<label class="gallery-name">Confirm Password:</label>

<input name="Confirm\_Password" type="password" class="product-price" id="Confirm\_Password" placeholder="Confirm\_Password">

<input name="submit" type="submit" class="h3" value=" Click to Apply ">

</form>

</div>

</div>

<div class="body-left">

<h3 class="sitail style5">You want register to vote! </h3>

<img src="images/cute\_magician-1503646-1920x1080.jpg"width="170" height="137"/></div>

<div class="footer">

<h3 align="center" class="product-compare">Copyright @2015 LimooDev Software creators</h3>

<h3 align="center" class="product-compare">Designed by: Ya'kov </h3>

</div>

</div>

</body>

</html>

<?php

//define the connection credentials

$user="root";

$host="localhost";

$dbname1="studentregistrationdb";

$dbname2="voterregistration";

$candidatename=$\_POST['Candidate\_Name'];

$candidatereg=$\_POST['Candidate\_Reg'];

$candidatefaculty=$\_POST['Candidate\_Faculty'];

$candidateyear=$\_POST['Candidate\_Year'];

$candidatepost=$\_POST['Candidate\_Post'];

$candidatepassword=$\_POST['Candidate\_Password'];

$confirmpassword=$\_POST['Confirm\_Password'];

//db connection

try

{

$dbname1= new PDO("mysql:host=$host;dbname=$dbname1", $user);

$count=$dbname1->prepare("select Student\_Reg from studentsnominal where Student\_Reg=:Student\_Reg");

$count->bindParam(":Student\_Reg",$candidatereg);

$count->execute();

$no=$count->rowCount();

if($no >0 )

{

try

{

$dbname2= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$count=$dbname2->prepare("select Candidate\_Reg from candidatesfinal where Candidate\_Reg=:Candidate\_Reg");

$count->bindParam(":Candidate\_Reg",$candidatereg);

$count->execute();

$no=$count->rowCount();

if($no==0 )

{

$sql="INSERT INTO candidatesfinal (Candidate\_Name, Candidate\_Reg, Candidate\_Faculty, Candidate\_Post, Candidate\_Year, Candidate\_Password, Confirm\_Password) VALUES(:Candidate\_Name, :Candidate\_Reg, :Candidate\_Faculty, :Candidate\_Post, :Candidate\_Year, :Candidate\_Password, :Confirm\_Password)";

$query=$dbname2->prepare($sql);

$query->execute(array(':Candidate\_Name'=>$candidatename, ':Candidate\_Reg'=>$candidatereg,':Candidate\_Faculty'=>$candidatefaculty, ':Candidate\_Post'=>$candidatepost, ':Candidate\_Year'=>$candidateyear, ':Candidate\_Password'=>md5($candidatepassword), ':Confirm\_Password'=>md5($confirmpassword) ));

echo'<script type="text/javascript">window.alert("You have susccesfully Applied!")

</script>';

}

Else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("You have already Applied for LUSO Post!")</script>';

}

}

catch(PDOException $e)

{

die('Could not connect to the database:' . $e);

}

}

else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("Check if you have signed the nominal roll!")</script>';

}

}

catch(PDOException $e)

{

die('Could not connect to the database:' . $e);

}

?>

**Presidential voting process codes:**

<?php

$user="root";

$host="localhost";

$dbname2="voterregistration";

?>

<!DOCTYPE html PUBLIC>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>LUSO Online Voting</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="style/stylevoter.css" rel="stylesheet" type="text/css" />

<link href="style/style1.css" rel="stylesheet" type="text/css" />

<link href="style/style2.css" rel="stylesheet" type="text/css" />

<link href="style/styleslider.css" rel="stylesheet" type="text/css" />

<!--[if IE]>

<link href="style/style-ie.css" rel="stylesheet" type="text/css" />

<![endif]-->

<style type="text/css">

<!--

.style3 {color: #9D0000}

-->

</style>

<style type="text/css" media="all">

<!--

@import url("\*");

.style20 {color: #FFFFFF}

.style21 {color: #000000}

.style23 {font-size: 16px}

-->

</style>

<?php

session\_start();

extract($\_SESSION);

//Initialize sessions

$voterreg=$\_SESSION['Voter\_Reg'];

$votername=$\_SESSION['Voter\_Name'];

?>

</head>

<body>

<center>

<div class="outer">

<div class="wrapper">

<div class="logo"><strong></strong>

<div class="picture1">

<img src="images/Logo.png" width="106" height="106"> </div>

<div class="laikipia">

<div class="name">

<h3 align="center" class="gallery-name style3">Laikipia University Student Organization Online Voting System</h3>

</div>

<div class="social">

<ul>

<li><img src="icons/Facebook-icon.png" width="37" height="40"/></li>

<li><img src="icons/Rss-icon.png" width="37" height="40"/></li>

<li><img src="icons/Skype-icon.png" width="37" height="40"/></li>

<li><img src="icons/Twitter-icon.png" width="37" height="40"/></li>

</ul>

</div>

</div>

<div class="menu">

<ul class="solidblockmenu">

<li><a href="indexvoter.php">Home</a></li>

<li><a href="about.html">About System </a></li>

<li><a href="studentunion.html">Student Union </a></li>

<li><a href="designer.html">About Designers </a></li>

<li><a href="gallery.html">Voters Gallery </a></li>

<li><a href="electoralcouncil.html">Electoral Council </a></li>

<li><a href="ethics.html">Electoral Ethics </a></li>

<li><a href="userhelp.php">User Help </a></li>

<li><a href="userview.php">User View </a></li>

</ul>

<div class="clear"></div>

</div>

</div>

<div class="pagetittle">

<h3 align="center" class="product-price">Laikipia University Student Organization Online Voting System President Candidates</h3>

</div>

<div class="body-wrapperlist">

<div class="body-candidates1">

<div class="body-middle-boxcandidates">

<div class="adminpanel">

<h3>Welcome <?php echo $votername; ?></h3>

<h3>Voting Process 1 (Presidential)</h3>

</div>

<div class="formcandidates">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Id FROM presidentfinal"; //first pass just gets the columnnames print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <tr> \n";

foreach ($row as $field =>$value)

{

print " <th>$field</th> \n";

} // end foreach

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "<td><input type=radio name=$name value=$value>$value<td>

</form>\n";

} // end field loop

print "</tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try ?>

<input style=" width:0px; text-color:white; height:0px; "class"="sitail2" "type="hidden" value="<?php echo $voterreg; ?>" name="Voter\_Reg" /><BR /><BR />

<input name="submit" type="submit" class="sitail2" value=" Vote ">

</form>

</div>

<div class="continue">

<h3 class="sitail3"><a href="vicepresidentvote.php">Proceed to process 2 of voting!</a></h3>

</div>

</div>

</div>

<div class="body-leftcandidates">

<div class="approved">

<h3 class="adminpanel style20">The List of Applied Candidates</h3>

</div>

<div class="sitaillist">

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT President\_Elect, Candidate\_Id FROM presidentfinal"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Presidential Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

</div>

</div>

<div class="footer">

<h3 align="center" class="product-compare">Copyright @2015 LimooDev Software creators</h3>

<h3 align="center" class="product-compare">Designed by: Ya'kov </h3>

</div>

</div>

</body>

</html>

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate4'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate4'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate5'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate5'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate6'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate6'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate7'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate7'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate8'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate8'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate9'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate9'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat WHERE Candidate\_Id='Candidate10'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Vicepresident\_Elect FROM vicepresidentfinal WHERE Candidate\_Id='Candidate10'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM vicepresidentstat";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?>

<p>TOTAL: <?php echo " <b> $num\_supp\_name </b> "; ?></p><br/>

</div>

</div>

<div class="rightresults">

<div class="result-one">

<h3 align="center" class="adminpanel style20 style21">Sports and Entertainment Results</h3>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate1'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate1'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate2'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate2'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate3'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate3'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate4'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate4'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate5'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate5'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate6'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate6'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate7'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate7'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate8'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate8'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate9'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate9'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat WHERE Candidate\_Id='Candidate10'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Sport\_Elect FROM sportsfinal WHERE Candidate\_Id='Candidate10'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM sportsstat";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?>

<p>TOTAL: <?php echo " <b> $num\_supp\_name </b> "; ?></p><br/>

</div>

<div class="result-two">

<h3 align="center" class="adminpanel style20 style21">Welfare Results</h3>

<?php

//db connection

$link=mysqli\_connect('localhost','root','') or die (mysqli\_error());

mysqli\_select\_db($link, "voterregistration") or die (mysqli\_error());

$query = "SELECT Candidate\_Id FROM welfarestat WHERE Candidate\_Id='Candidate1'";

$result = mysqli\_query($link, $query);

$num\_supp\_name = mysqli\_num\_rows($result);

?><?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Welfare\_Elect FROM welfarefinal WHERE Candidate\_Id='Candidate1'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td>$value $num\_supp\_name Votes<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

<td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Vice President'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Vice President Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Director Finance'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Director Finance Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Director Sports and Entertainment'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Director Sports Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Director Welfare'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Director Welfare Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Director Academics'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Director Academics Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Secretary General'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Secretary General Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

<div class="candidateapplied">

<?php

try {

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$con->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$query = "SELECT Candidate\_Name, Candidate\_Reg FROM candidatesfinal WHERE Candidate\_Post='Organising Secretary'"; //first pass just gets the columnnames

print "<table> \n";

$result = $con->query($query); //return only the first row (we only need field names)

$row =$result->fetch(PDO::FETCH\_ASSOC);

print " <p align=justify class=product-compare style20 style22>Organising Secretary Candidates</p> \n";

print " </tr> \n"; //second query gets the data

$data = $con->query($query);

$data->setFetchMode(PDO::FETCH\_ASSOC);

foreach($data as $row)

{

print " <tr> \n";

foreach ($row as $name=>$value)

{

print "

<td><p align=justify class=product-compare style20 style22>$value</p><td>

</form>\n";

} // end field loop

print " </tr> \n";

} // end record loop

print "</table> \n";

}

catch(PDOException $e)

{

echo 'ERROR: ' .$e->getMessage();

} // end try

?>

</div>

</div>

</div>

</div>

<div class="footer">

<h3 align="center" class="product-compare">Copyright @2015 LimooDev Software creators</h3>

<h3 align="center" class="product-compare">Designed by: Ya'kov </h3>

</div>

</div>

</body>

</html>

<?php

//define the variables

$candidatename=$\_POST['President\_Elect'];

$candidatereg=$\_POST['Candidate\_Reg'];

$candidatepost=$\_POST['Candidate\_Post'];

//check if already confirm registration

try{

$con= new PDO("mysql:host=$host;dbname=$dbname2", $user);

$count=$con->prepare("SELECT Candidate\_Reg, Candidate\_Name, Candidate\_Post FROM candidatesfinal WHERE Candidate\_Reg=:Candidate\_Reg && Candidate\_Name=:Candidate\_Name && Candidate\_Post=:Candidate\_Post");

$count->bindParam(":Candidate\_Reg",$candidatereg);

$count->bindParam(":Candidate\_Name",$candidatename);

$count->bindParam(":Candidate\_Post",$candidatepost);

$count->execute();

$no=$count->rowCount();

if($no>0 )

{

$count=$con->prepare("SELECT Candidate\_Reg, Candidate\_Name FROM presidentfinal WHERE Candidate\_Reg=:Candidate\_Reg && Candidate\_Name=:Candidate\_Name");

$count->bindParam(":Candidate\_Reg",$candidatereg);

$count->bindParam(":President\_Elect",$candidatename);

$count->execute();

$no=$count->rowCount();

if($no==0 )

{

$sql="INSERT INTO presidentfinal (President\_Elect, Candidate\_Reg, Candidate\_Post ) VALUES(:President\_Elect, :Candidate\_Reg, :Candidate\_Post)";

$query=$con->prepare($sql);

$query->execute(array(':President\_Elect'=>$candidatename, ':Candidate\_Reg'=>$candidatereg, ':Candidate\_Post'=>$candidatepost));

echo'<script type="text/javascript">

window.alert("You have successfully registered the candidates!")

</script>';

}

else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("The candidates have already registered")

</script>';

}

}

else

{

echo'<title>LUSO Online Voting System</title><script type="text/javascript">

window.alert("Please check from the list of Candidates who have applied")

</script>';

}

}

catch(PDOException $e)

{

die('Could not connect to the database:' . $e);

}

?>

**Admin panel**:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>LUSO Online Voting</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<link href="style/stylevoter.css" rel="stylesheet" type="text/css" />

<link href="style/style1.css" rel="stylesheet" type="text/css" />

<link href="style/style2.css" rel="stylesheet" type="text/css" />

<link href="style/styleslider.css" rel="stylesheet" type="text/css" />

<!--[if IE]>

<link href="style/style-ie.css" rel="stylesheet" type="text/css" />

<![endif]-->

<style type="text/css">

<!--

.style3 {color: #9D0000}

-->

</style>

<style type="text/css" media="all">

<!--

@import url("\*");

.style5 {font-size: 12px}

.style13 {color: #9F0000}

.style14 {color: #800000}

.style19 {font-size: 16px}

.style20 {color: #FFFFFF}

-->

</style>

</head>

<body>

<center>

<div class="outer">

<div class="wrapper">

<div class="logo"><strong></strong>

<div class="picture1">

<img src="images/Logo.png" width="106" height="106"> </div>

<div class="laikipia">

<div class="name">

<h3 align="center" class="gallery-name style3">Laikipia University Student Organization Online Voting System</h3>

</div>

<div class="social">

<ul>

<li><img src="icons/Facebook-icon.png" width="37" height="40"/></li>

<li><img src="icons/Rss-icon.png" width="37" height="40"/></li>

<li><img src="icons/Skype-icon.png" width="37" height="40"/></li>

<li><img src="icons/Twitter-icon.png" width="37" height="40"/></li>

</ul>

</div>

</div>

<div class="menu">

<ul class="solidblockmenu">

<li><a href="#">Home</a></li>

<li><a href="#">About System </a></li>

<li><a href="#">Student Union </a></li>

<li><a href="#">About Designers </a></li>

<li><a href="#">Voters Gallery </a></li>

<li><a href="#">Electoral Council </a></li>

<li><a href="#">Electoral Ethics </a></li>

<li><a href="#">User Help </a></li>

<li><a href="#">User View </a></li>

</ul>

<div class="clear"></div>

</div>

</div>

<div class="body-wrapper">

<div class="body-right">

<blockquote>

<h3 align="center" class="product-price style13">Catch up! Amazing Online Voting!</h3>

</blockquote>

<img src="images/fayee2.jpg" width="163" height="266"/></div>

<div class="body-candidates2">

<div class="body-middle-boxadmin1">

<div class="sale-title">

<h3 class="sale-title style19">LUSO Administrator Access Panel </h3>

</div>

<div class="adminpanel">

<ul>

<li class="style20">Voting dates</li>

<li class="style20"><a href="statisticsfinal1.php">Voting results</a></li>

<li class="style20">View Contacts</li>

<li class="style20">More Candidates</li>

</ul>

</div>

</div>

</div>

<div class="body-leftadmin">

<h3 class="product-title">Administrator Choice </h3>

<img src="images/Modern\_Alvin\_and\_The\_Chipmunks.jpg" width="147" height="139"/>

<h3 class="solidblockmenu style14">Click to View Applied!</h3>

<ul>

<li class="sitail2 style5"><a href="president.php">Presidents</a></li>

<li class="sitail2 style5"><a href="vicepresident.php">Vice-President</a></li>

<li class="sitail2 style5"><a href="finance.php">Director Finance</a></li>

<li class="sitail2 style5"><a href="sports.php">Director Sports</a></li>

<li class="sitail2 style5"><a href="welfare.php">Director Welfare</a></li>

<li class="sitail2 style5"><a href="secgen.php">Secretary General</a></li>

<li class="sitail2 style5"><a href="orgsec.php">Org- Secretary</a></li>

<li class="sitail2 style5"><a href="academics.php">Director Academic</a></li>

</ul>

</div>

</div>

</div>

</div>

<div class="footer">

<h3 align="center" class="product-compare">Copyright @2015 LimooDev Software creators</h3>

<h3 align="center" class="product-compare">Designed by: Ya'kov </h3>

</div>

</div>

</body>

</html>

## **Appendix 3: Test data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test id | Description | Test data | Expected results | Actual results |
| 1 | Voter registration | Ya’kov  Erick  Gregory Ochieng | Registration successful. Voter records updated in voter table in the registration database | Registration successful |
| 2 | Login | Ya’kov Erick  Gregory Ochieng | Allow access into the system | As expected |
| 3 | Candidate registration | Geofrey Omondi  Calvince Oduogi  Faith Chelimoo | Registration successful. Candidate records updated in candidate table in the registration database. | As expected |
| 5 | Candidate shortlisting |  | List of shortlisted candidate is displayed | As expected |
| 5 | Voter validation |  | Voter is allowed to register | Registration successful |
| 6 | Voting |  | Voter selects one candidate per position and submit vote | Voting successful |
| 7 | Vote statistics | Robert Ouko  Ya'kov Erick | Display real time vote count per candidate | As expected |