**CHAPTER ONE**

**INTRODUCTION**

This chapter outlines the project and gives some context. Furthermore, this chapter follows the problem description of the project to offer a clear grasp of the project's scope and objectives. This chapter will serve as the foundation for all future phases of development.

**1.1 Background of the Study**

An election is the most effective way for associates to have a voice in the leadership and direction of their organization. When people are able to vote in fair and open elections, they feel a higher feeling of worth, power, and duty. Aside from that, online voting services enable members of associations, people, and other autonomous organizations to vote easily and conveniently. Salleh et al (2021).

Elections are the cornerstones of any constitution, as they allow people to select the officials who will oversee them. From paper ballots to electronic voting machines, the voting system has evolved in terms of convenience and efficiency. Selvarani et al. (2017)

Every session, student elections are held at various Nigerian higher institutions. There are three branches of government in which student representatives or executives are commonly elected: the Students Union Government (SUG), the Departmental, and the Hostel levels. They are elected solely by students. In other words, for each student’s election, students from any specific tertiary institution vote for their peers who have expressed interest in the positions stated above. Voters with this method can vote from anywhere and at any time, eliminating the need to visit polling places. This saves time and money while also avoiding exhaustion and violence. The processes employed are simple, straightforward, and safe. This solution uses no expensive hardware and meets the demand for remote voting.

Higher education institutions are supposed to hold free and fair elections for popularity based on understudy affiliation. Elections provide the people the ability to choose their representatives and express their ideas for how government should be run. Thus, the legitimacy and accuracy of the electoral process are critical to the credibility of democracy itself. (Amit & Abdullahi, 2019).

The primary purpose of e-Voting is to provide voters with an exceptional circumstance in which they may cast their ballots with minimal expense and effort. The use of computers or computerized voting technology to cast ballots in an election is referred to as electronic voting. This phrase is often used more explicitly to refer to voting that takes place through the Internet. Electronic voting machines can be used to register voters, count ballots, and record votes. (Amit & Abdullahi, 2019).

**1.2 Statement of the Problem**

computer science election commission often involves a lot of manual labor, such as printing and distributing ballots, collecting and counting votes, and verifying the accuracy of the results. This can be time-consuming and expensive, and it may not be able to accommodate large numbers of voters. It is also prone to errors and potential fraud. For example, votes may be miscounted, or there may be discrepancies in the results. This can lead to disputes and challenges to the legitimacy of the election. The current voting systems may also pose barriers to certain groups of voters, such as people with disabilities or those who live far from the school axis. This can undermine the integrity and credibility of the election

**1.3 Aim and Objectives of the Study**

The project is aimed at designing a working web-based E-voting application for the computer science department at Kaduna polytechnic.

**Objectives**

The objectives of this research work are as follows:

1. In the front-end development, HTML and CSS will be employed to create an interactive UI and UX as well as Django which is a Python web framework will be employed in developing the back end.
2. Series of vital testing will be carried out in ensuring the efficacy of the research work.
3. In storing and retrieving location data; MySQL, an open-source relational database, will be used as the database technology.
4. The student data set will be extracted from the department based on some criteria ensuring that only the right set of people can vote.

**1.4 Scope of the Study**

This project focuses on the design, implementation, and assessment of a web-based e-voting system that is safe, easy to use, accessible, and complies with web accessibility standards, as well as the procedures of voter registration, authentication, and auditing. The scope of this project will be limited to the computer science department at Kaduna Polytechnic.

**1.5 Limitations of the Study**

This study's scope has been constrained by several core issues, including:

**Time** - The researcher's everyday busy academic pursuits limited the time allotted for research for this study.

**Access to literature** – Access to some material was restricted, although the available material was optimized.

**1.6 Significance of Study**

A study on a web-based e-voting system can have a significant impact on the way elections are conducted in the department. The implementation of a web-based e-voting system can bring many benefits to the electoral process in the department, such as increased efficiency, accessibility, security, and voter participation. But to be able to achieve those benefits, it's vital to ensure that the system is designed and implemented in a secure, transparent, and accessible manner.

**1.7 Project Organization**

The project is divided into five chapters. The outlines are presented below:

**Chapter One: Introduction**

Chapter one introduces this project work, the study's background, the problem statement, the purpose and objectives, the scope of the study, the constraints of the study, the relevance of the study, the project organization, and the definition of terms.

**Chapter Two: Literature review**

This chapter focuses on the literature review, and the contributions of other scholars on the subject matter being discussed.

**Chapter Three: Methodology and Design**

This chapter is concerned with the presentation of the results of system analysis and design. It presents the research methodology used in the development of the system to facilitate an understanding and effective future implementation of the system.

**Chapter Four: System Implementation Evaluation**

This chapter describes the system implementation and documentation, analysis of modules, and system requirements for implementation.

**Chapter Five: Summary, Conclusion, and** **Recommendation**

The chapter provides a summary of major findings, conclusions, and recommendations based on the study conducted.

**1.8 Definition of Terms**

1. **E-voting:** Refers to the use of electronic devices, such as computers or smartphones, to cast and count votes.
2. **Web-based:** Refers to an e-voting system that is accessible via the Internet.
3. **SUG:** SUG is an abbreviation for Student Union Government. It is an organization in higher education that is in charge of safeguarding and defending students' rights on campus. They arrange recreational events, offer social services, and advocate the political interests of students.
4. **Voter registration and authentication:** The process of validating that a voter is eligible to vote and creating a record of their vote.
5. **Auditing and verification:** The process of ensuring the integrity of the election results by auditing and verifying the voting process and results.

**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 Introduction**

This chapter attempts to illustrate how the problem under study links to previous research, current practice, or other fields of knowledge by identifying relevant publications by other researchers who have addressed a related issue. Furthermore, to establish the project subject's importance in the field, this chapter will give a synthesis of existing research on the issue, highlighting areas of agreement, disagreement, and gaps in the literature.

**2.2 Literature Review**

Shanthi et al. (2018). A Survey on Web-Based Application of Secure Online Voting System. For a few years, several approaches, such as punch card systems or the secret ballot method, have been used to continue election procedures in which voters must visit the booth to cast their ballots under the present system. Voting systems have evolved as a result of the rapid expansion of information technology, allowing a voter to be a part of an automated process that is only achievable through Voting Systems. Because the proposed method is online, even those who reside outside of their hometown can vote. The main purpose is to increase the vote percentage. The primary goal of the suggested system is to enable rapid and efficient information retrieval.

Furthermore, the proposed software has been designed and tested to function on Ethernet and to enable online voting. It also produces and manages voting and election details, since all users must log in with their username and password and click on their preferred candidates to register to vote. This will boost the number of people who vote in India. The use of strong security will limit the number of fake votes.

In conclusion this work, we suggested a safe and user-friendly online voting system based on biometric authentication. The prior method was unable to give casted votes and was neither securely nor independently kept. We can give securely casted votes, and only verified individuals may vote. A system ensures online voters' safety and protects them from numerous security threats. It is a dependable technique for voting and recording votes.

Indapwar (2020). E-Voting system using Blockchain technology.

Normal voting is now done with an electronic voting system, which saves each voter's vote in a centralized database. Because these centralized databases keep all of the data in a single area, they are easily hackable and may be changed. As a result, the data may be inconsistent during the vote count and may not give us with the right result. As a result, we employ blockchain technology to construct a decentralized application in which data tampering is practically difficult since Blockchain uses a decentralized method for data storage in which data is saved in a single spot.

Moreso, the system employed the use of Metamask, which allows the user to visit the distributed web of the future in your browser today, as well as NodeJS, an open source, cross-platform environment for developing server-side and networking applications, as well as ReactJS, a JavaScript library that is used for building various reusable UI components, and finally it employs Firebase as a No-SQL database for the voter and manager databases.

To summarize, permissioned, public, shared blockchain is a hybrid system that allows for circumstances where whitelisted access is necessary yet all transactions are available to the public. This will give the transparency that democracies require.

Salleh et al (2021) noted in a study on the Implementation of the Electronic Voting System for Student Representation Council using reCAPTCHA. The students at (Kolej Teknologi Antarabangsa Cybernetics) are now voting on paper ballots, which is the conventional way. Because there are so many students at that college, the present traditional approach is no longer appropriate or convenient. This is not appropriate since the organization must employ people to compute votes or paper ballots during elections. The use of paper ballots is time-consuming, but it causes a lot of work for those involved in ballot counting. This will add time and expense to the institutions' participation in this initiative.

Furthermore, the Waterfall Model is seen to be the best to utilize after gathering data prior to constructing the system, and it is also thought to be the easiest to use. The functional testing was done with thirty students, and the findings indicated that 63% of the respondents agreed that the conventional voting method should be replaced with an electronic voting system, particularly to reduce voting time.

In conclusion, using reCAPTCHA, the built electronic voting system can determine if people have voted for the students' representative candidate twice. Security is critical while creating an E-Voting website. This electronic voting method will allow organizations to improve the efficiency, dependability, and transparency of elections.

Selvarani et al. (2017) carried out a study on a Secure voting System through SMS and using a smartphone application. From paper ballots to electronic voting machines, the voting system has evolved in terms of convenience and efficiency. However, because these systems are time-consuming, the majority of the population finds it difficult to vote because of their busy schedules, and many are also hesitant to stand in long lines. In addition to these, democracies throughout the world face several voting-related issues. As a result, mobile voting allows the primary solutions to emerging to the greatest extent possible.

Moreso, the suggested system is based on the RSA algorithm and consists of two key components: access control, voting, and an election administration server. The first section is for voter validation and identity. The second step is for categorizing the final results for the received encrypted data using decryption. This system's prerequisites include the Election Commission Server (ECS), Election Commission Mobile Phone Databases (ECD), Vote Collecting and Result Phase Server (VCRPS), and Election Commission Office are all examples of election-related services (ECO).

In conclusion, this study suggests using mobile phone voting for commercial reasons. This also prohibits multiple voting because the databases are kept in encrypted form. This remote voting technique is more efficient, helpful, and dependable. This is also cost and time efficient, as it only requires a few resources.

Amit and Abdullahi (2019). Design and Development of a Real-Time E-Voting System with High

Security Features. The primary purpose of this system is to provide voters with an exceptional circumstance in which they may cast their ballots with minimal expense and effort.

Thus, using data system standards of the Software Development Life Cycle (SDLC) model, Dynamic systems development method (DSDM), and Object-Oriented Analysis and Design, this research set out to build up a utility electronic voting system that, when conveyed on web browsers of a gadget, could potentially enable voting to be done effectively. Following the steps of the Dynamic systems development method (DSDM), a Unified Modeling Language (UML) diagram, entity relational diagram, and dataflow diagram of each component and the entire system were created. Brackets were chosen as the development environment for the implementation part of this research because it provides a complete collection of free modules for the building of web applications. The application built was tested using one of the web testing devices known as Selenium.

In conclusion, the study set out to devise a clever solution for dealing with the intermittent concerns of election conduct by employing an electronic voting system for election purposes. This study highlighted the benefits of electronic voting systems in election administration.

**2.3 Summary of Related Literature Reviews**

|  |  |  |
| --- | --- | --- |
| **Author & Year** | **Title & Description** | **Merit and Demerits** |
| Shanthi et al. (2018). | A Survey on Web-Based Application of Secure Online Voting System.  The proposed technology is a web-based application for online voting, which will make voting much easier and more efficient. | The system provided a securely casted vote, and only verified people may vote.  Difficult to determine the scalability of the system. |
| Indapwar (2020). | E-Voting system using Blockchain technology.  The concept proposes a strategy for increasing trust between voters and governments by safeguarding their data. | The technology provides a continuous decentralized network, with no single body controlling the data.  The system's main drawback is that it can only handle a short string of text that just records a balance transfer between two parties. |
| Salleh et al (2021). | The Implementation of the Electronic Voting System for Student Representation Council using reCAPTCHA.  This study attempts to create an E-Voting System for a private international institution by utilizing the 'reCAPTCHA' security component. | The implementation increases the efficiency, reliability, and transparency of student representative elections.  The implementation is limited to just students at Kolej Teknologi Antarabangsa Cybernetics |
| Selvarani et al. (2017) | Secure voting System through SMS and using a smartphone application.  Voters in this system can vote from anywhere and at any time, without having to attend polling places. This saves time and money while also avoiding fatigue and violence. | The system prohibits multiple voting because the databases are kept in encrypted form.  Internet connectivity is required. |
| Amit and Abdullahi (2019). | Design and Development of a Real-Time E-Voting System with High-Security Features.  The main purpose of this system is to provide voters with exceptional circumstances in which they may cast their ballots with minimal expense and effort. | The system provides a useful, easy, and efficient method of casting a ballot, removing the limitations of the previous method.  The system might not be totally transparent. |

**2.4 Analysis of the Current System**

The current system is a paper-based method of voting, paper voting is a method of casting and counting votes in which voters mark their choices by hand on a paper ballot. This method has been widely used on the school campus and it has several advantages and disadvantages.

The advantages of the current voting system include:

1. It is a simple and straightforward process that is easy for voters (students) to understand and participate in.
2. Paper ballots are durable and can be stored for a long period of time, making them useful for recounts and audits.

The disadvantages of the current voting system include:

1. It is a relatively slow process to count and verify the votes by hand.
2. It is more susceptible to human errors such as mistakes in marking the ballot or miscounting the votes.
3. It can be more expensive than electronic voting systems because of the cost of printing and distributing paper ballots.

Overall, the current voting system is a well-established method of conducting elections, but it does have some limitations. Electronic voting systems have been developed as an alternative, but it is important to weigh the pros and cons of both methods carefully before making a decision on which method to use.

**2.5 Analysis of the Proposed System**

Keeping in mind the aforementioned shortcoming, the suggested approach efficiently addresses the aforementioned issues. Electronic voting (e-voting) is a method of casting and counting votes in which voters use electronic devices, such as computers or smartphones, to mark their choices. This method has been implemented in some elections around the world as a means of improving efficiency and accessibility.

**2.5.1 Advantages of the New Proposed System**

E-voting, or electronic voting, has several advantages that make it an attractive option for conducting elections.

1. **Efficiency**: The proposed system would allow for faster vote counting and tabulation by using electronic devices, which can help speed up the election process and reduce the time required to announce results.
2. **Accessibility**: The proposed system can make voting more accessible for students. It will allow for easy voting from anywhere and in some cases like internet voting students can vote from anywhere.
3. **Reduced potential for human error**: The proposed system can automatically detect and correct errors, which can improve the accuracy of the results and reduce the potential for human error that can occur in the current voting systems.
4. **Better Voter Turnout**: The proposed system can increase voter turnout as it eliminates geographical barriers and makes the voting process more convenient, which in turn can make it more appealing to certain students.

**CHAPTER THREE**

**METHODOLOGY AND DESIGN**

**3.1 Introduction**

A methodology is a process of rigorous study or inquiry, particularly to unearth new facts or information; hence, research methodology should be good enough to enable the achievement of the specified objectives which are achievable using specific components, such as data collection and design procedures and system modeling (use case, activity, and class diagrams). This chapter contains the input/output specifications, and system requirements for the development of a web-based voting application for the computer science department.

**3.2 Methods of Data Collection**

Before developing any system, collecting data and facts about the existing system is critical to understand what is going on. This research was carried out using three methods.

1. Observation of the Work Environment
2. Interview
3. Documentation

**3.2.1 Observation of the Work Environment**

This method was employed to acquire information and data for this study by monitoring how the manual system worked. The most evident flaws in the existing system were discovered via detailed inspection. Using the observational approach, the context in which the observation is made can be modified in a variety of ways.

**3.2.2 Interview**

The main objective of using interviews as a method of data collection is to obtain information thoroughly and rigorously. Based on the questions the researcher provided, the researcher met meet with some election commission members and acquired reliable information.

**3.2.3 Documentation**

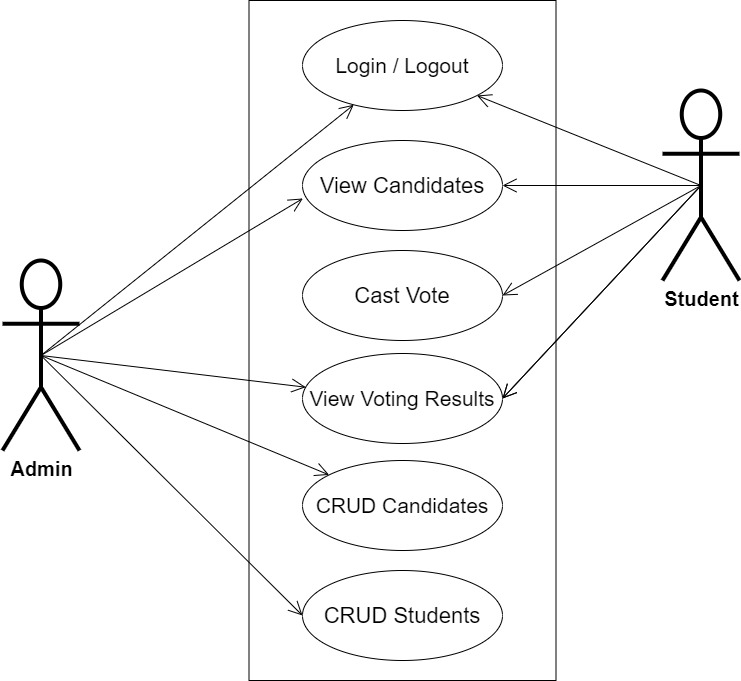
Documentation is a type of secondary data collection. This method makes use of journals, manuals, past work, publications, and other sources. This method of data collection is used because it allows for comparison with past studies. This includes the internet, which is a data collection tool. The internet was used to find information on difficult or ambiguous issues.

**3.3 System Modeling**

A system model is a conceptual model of a system that explains and represents it. A system is any interaction between a set of components that work together to achieve a common purpose. Visual models of object-oriented software-intensive systems may be created utilizing a set of visual notation techniques included in the Unified Modeling Language, which is used in the creation of this contemporary system. UML diagrams utilized in this new design include use case diagrams, class diagrams, and activity diagrams.

**3.3.1 Use Case Diagrams**

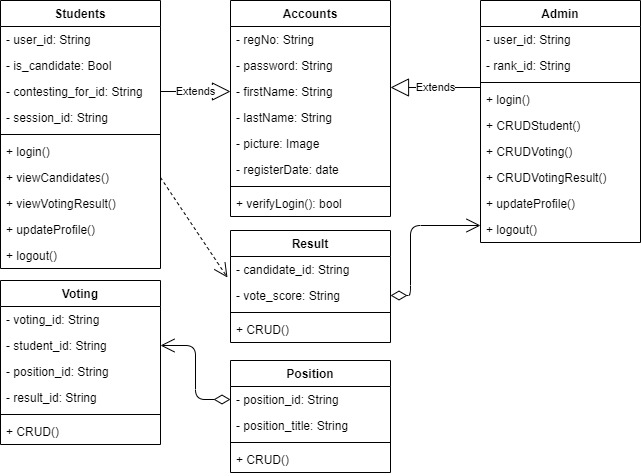
Use cases are collections of system-to-user interactions. Use case diagrams are used to graphically characterize the functionality of a system in terms of its actors, goals (represented as use cases), and dependencies between those use cases.



**Fig 3.1 System Use Case Diagram**

**3.3.2 Class Diagrams**

The Unified Modeling Language (UML) class diagram is an implementation of an independent perspective of how the system interface would look, with each class having its own set of attributes and demonstrating how they interact with one another. Class diagrams employ the Unified Modeling Language standards to visually portray a given system's static structure and composition (UML).



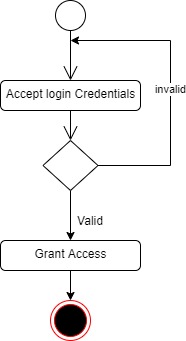
**Fig 3.2 System Class Diagram**

**3.3.3 Activity Diagrams**

An activity diagram, like a flowchart or a data flow diagram, visually illustrates a series of events or the flow of control in a system, but it acts more like an enhanced version of both.

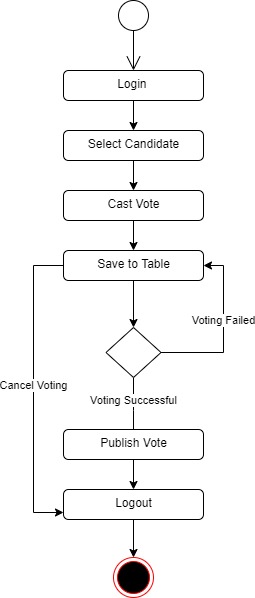
**Login**

The process for gaining access to the system is depicted in the diagram below; the username and password must be accurate to gain access.



**Fig 3.3.1 Login Activity Diagram**

**Voting**

The process for voting for a candidate is depicted below, to vote one has to select from each category of available positions.

**Fig 3.3.2 Voting Activity Diagram**

**3.4 Database Design**

The logical explanation of how data is kept in the computer's memory is called input specification. The freedom experienced in using the system, as well as the convenience of retrieving and reading the data and assuring applicability across the internet, make SQL standards essential for ensuring that structured data is uniform and independent of applications. Some of the input specifications employed in this project work are presented below.

1. Accounts Table: contains basic information about all system users.
2. Voting Table: contains the voting record for the system.

**Table 3.1 Account Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Reg\_no | String | 150 | Reg\_no for login (case insensitive) |
| Password | String | 150 | Access Code (case sensitive) |
| Firstname | String | 150 | User first name |
| Lastname | String | 150 | User last name |
| Picture | Image | - | User profile picture |
| RegisterDate | String | 100 | The date the user was registered |
| acct\_id | String | 64 | A unique string for identifying users |

**Primary key:** acct\_id

**Table 3.2 Voting Table input specification table**

|  |  |  |  |
| --- | --- | --- | --- |
| **FIELD NAME** | **DATA TYPE** | **LENGTH** | **DESCRIPTION** |
| Student\_id | String | 150 | Reference to the student model |
| Position\_id | String | 150 | Reference to the position model |
| Result\_id | String | 150 | Reference to the result of the voting |
| Voting\_id | String | 64 | A unique string for identifying votes |

**Primary key:** voting\_id

**3.5 Output Design**

This declares and displays the outcome of the given input. The automated system's output is dependent on its input. The output specification is listed below.

**Table 3.3 Account Table Output Design**

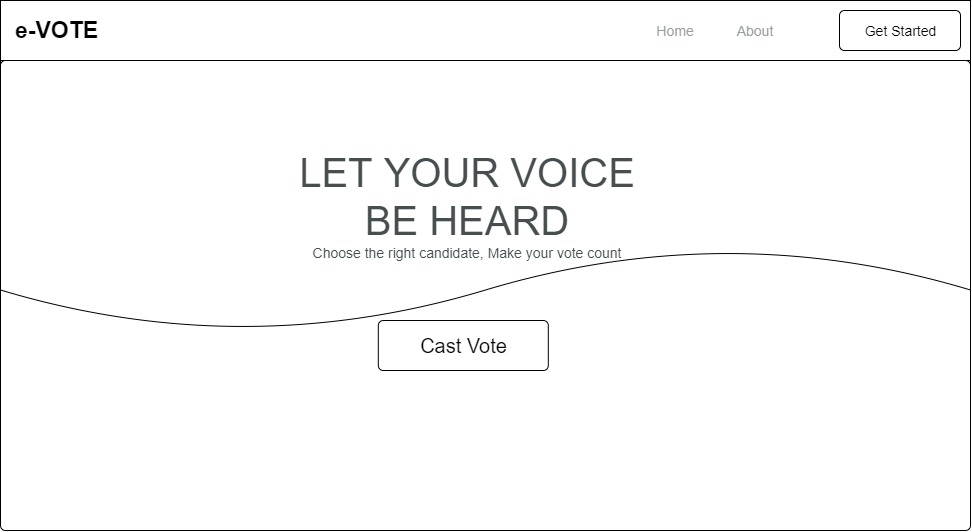
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Reg\_no** | **Password** | **Firstname** | **Lastname** | **RegisterDate** | **Picture** | **Acct\_id** |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX | XXXX | XXXX | XXXX |

**Table 3.4 Voting Table Output Design**

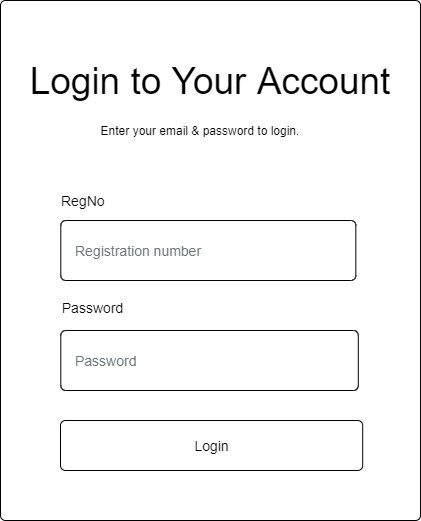
|  |  |  |  |
| --- | --- | --- | --- |
| **Student\_id** | **Postion\_id** | **Result\_id** | **Voting\_id** |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |
| XXXX | XXXX | XXXX | XXXX |

**3.6 Input & User Interface Design**

This is a graphic depiction of the system interface; it will be designed to be user-friendly, responsive, and visually beautiful. Furthermore, it will be fully secured, thus authentication will be required to see various levels of the information. To help with the designs, a mid-fidelity wireframing program called Draw.io is employed.



**Fig 3.6.1 Home Page**



**Fig 3.6.1 User Login Screen**



**Fig 3.6.2 Contestant Form**

**3.7 System Requirement**

Every piece of software-generated has predefined system requirements that it must fulfill in order to function properly. The system requirements, on the other hand, are the bare minimum of hardware and software required for the system's intended operation.

**3.7.1 Hardware Requirement**

System Hardware Requirement Include:

1. Minimum of 2 GB of RAM (Random Access Memory).
2. Minimum of Intel Dual core processor.
3. Minimum of 250GB HDD (Hard Disk Drive).

**3.7.2 Software Requirement**

The software requirements include:

1. At least windows 7 OS (Operating System).
2. Vs. Code IDE installation.
3. Browsers include Chrome and Firefox.

**3.8 Choice of Programming Language**

This research work will be a web-based application and will be implemented on a relational database system (SQLite). HTML (hypertext markup language), CSS (cascading style sheet), and JavaScript will be employed in the front end while Django(python) will be employed for the backend programming. The above are the modern languages used in implementing this system.

# CHAPTER FOUR

# SYSTEM IMPLEMENTATION EVALUATION

# 4.1 Introduction

This section describes in concise detail how the new system is implemented for effective operation. It shows samples of the working (new) system designed and how the system is to be installed.

# System Testing and Evaluation

There are many reasons to conduct the testing for the developed system because is only through testing that we can be able to analyze any problem in the new system and provide solutions to these problems This project employed both unit and integration testing to ensure the effectiveness and efficiency of design and to ensure that the new system meets its required functionalities and is error-free.

**Unit Testing**

In this section, individual units or single components of the system are tested independently to ensure that individual phases are working effectively without errors.

**Integration Testing**

Testing of the program was implemented using integration testing all the units were put together as one so they function as one. The link between the various units was tested to be sure that they are correctly integrated, and also to be sure that the units can function correctly together as one.

# 4.3 System Installation

In order to use the proposed application on any computer system, the following steps need to be taken:

1. Make sure, pip, pipenv, and python3 or greater are installed on the system.
2. Copy your project folder to any location of your choice.
3. Open project folder in Visual Studio Code
4. On the terminal run “pipenv install -r requirements.txt”
5. On the terminal run “python manage.py runserver”
6. Open any browser on the system example Chrome, Microsoft Edge, or Mozilla Firefox.
7. On the address bar, type <http://127.0.0.1> and press the enter key the site should be loaded.

# 4.4 Security Measures

Since the scope of the application is public, some of the information and pages are restricted to either the student or the admin. But some other information and functionalities are restricted to some and not all who use the application. The restrictions are carried out by the use of passwords which gives different levels of access to users.

# 4.5 Sample Outputs

These describe and give the pictorial representation of the program or software; it shows and gives a clear understanding of the design, and displays all the interfaces

**Homepage**

The page serves as a gateway to navigate and explore the various sections of the website

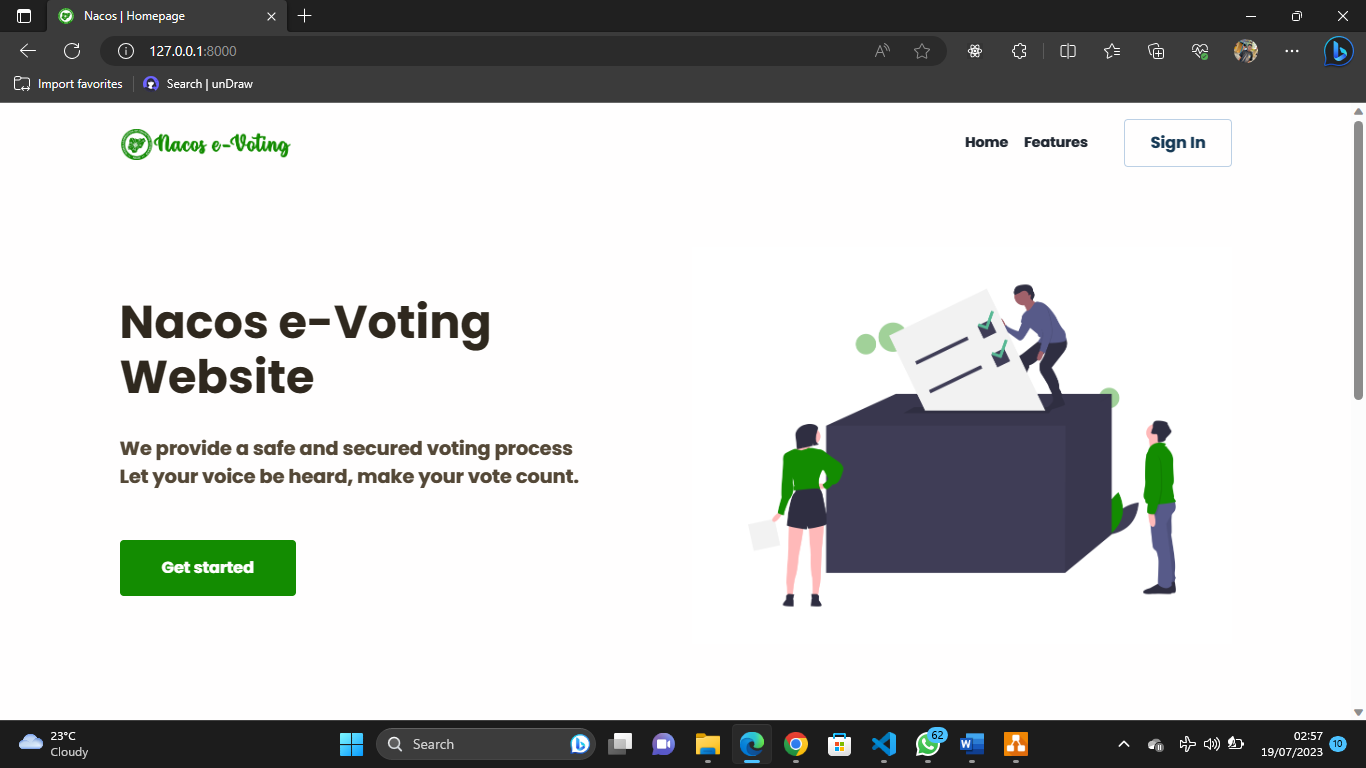
****

Fig 4.1 Homepage

**User Login**

This is a page that grants users (admin, and student) access to the system only if the correct credentials are provided

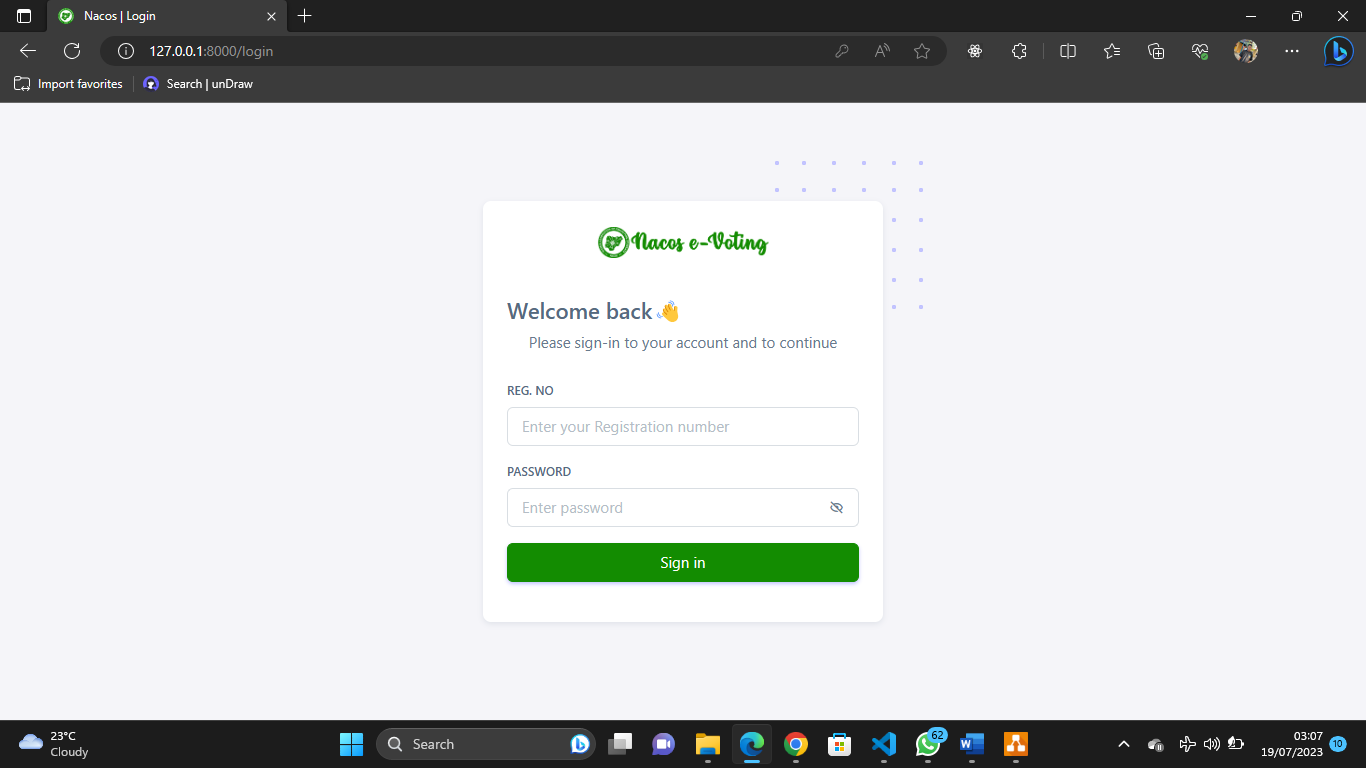
****

Fig 4.2 User Login

**Admin Dashboard**

This is the clearance admin dashboard, the sidebar shows the available functionality for the admin

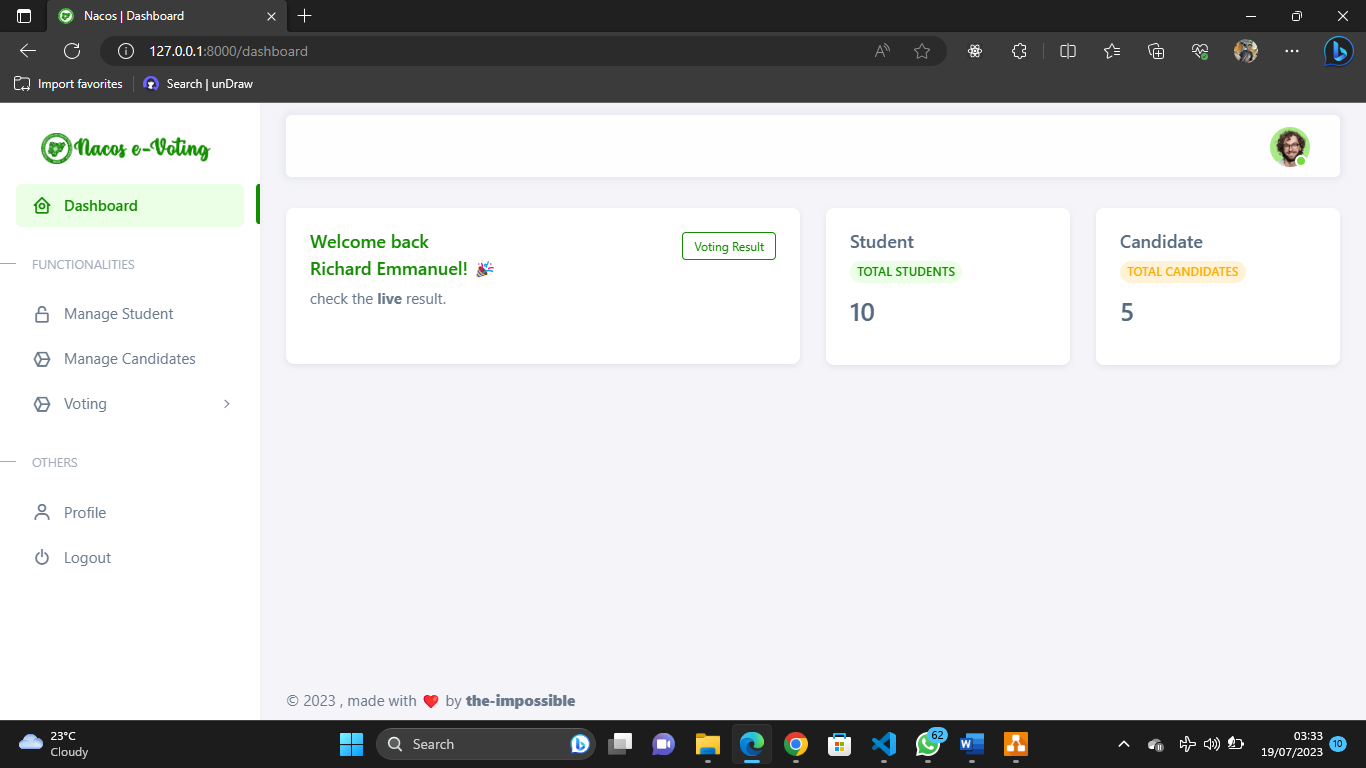
****

Fig 4.3 Admin Dashboard

**Manage Student’s Account**

This is the page where the admin can effectively create and manage each student account

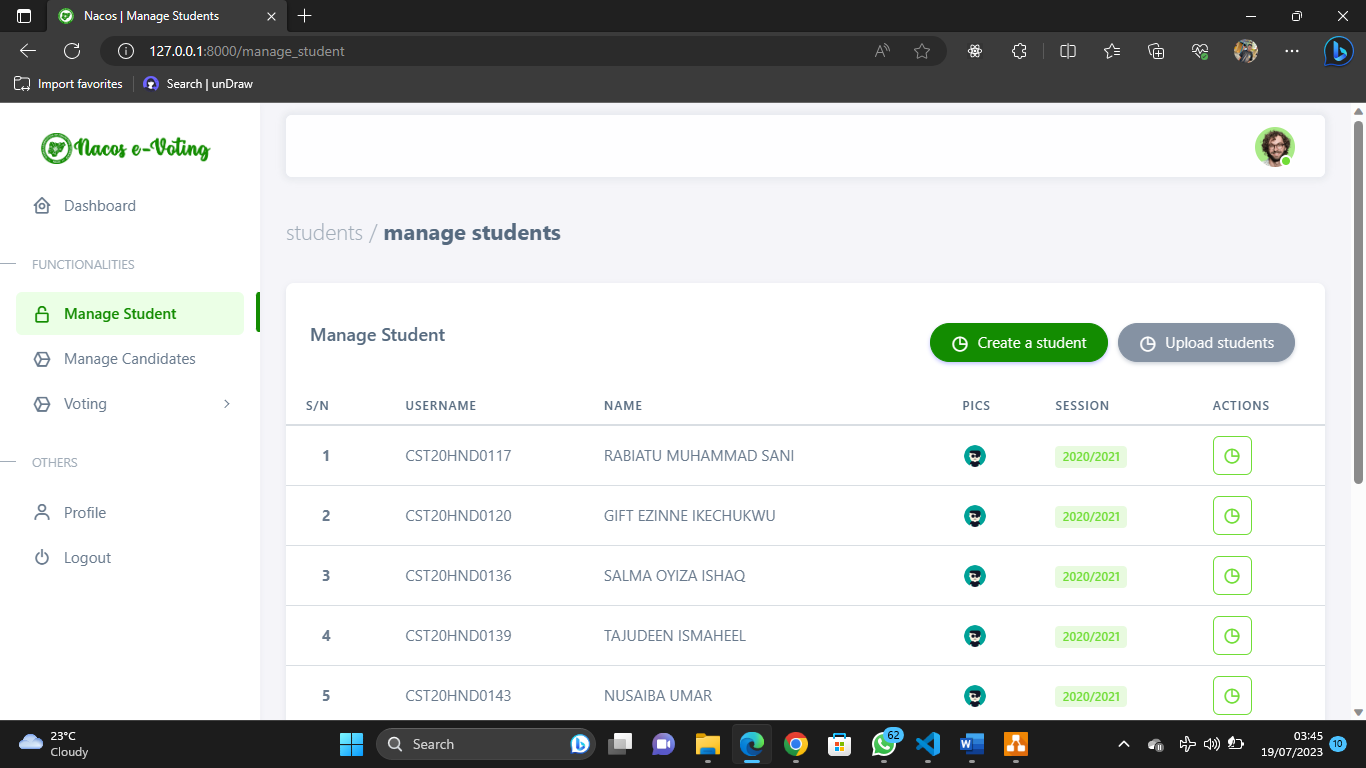


Fig 4.4 Manage Student’s Account

**Manage Candidates**

This is the page where the admin can effectively create and manage each candidate and positions

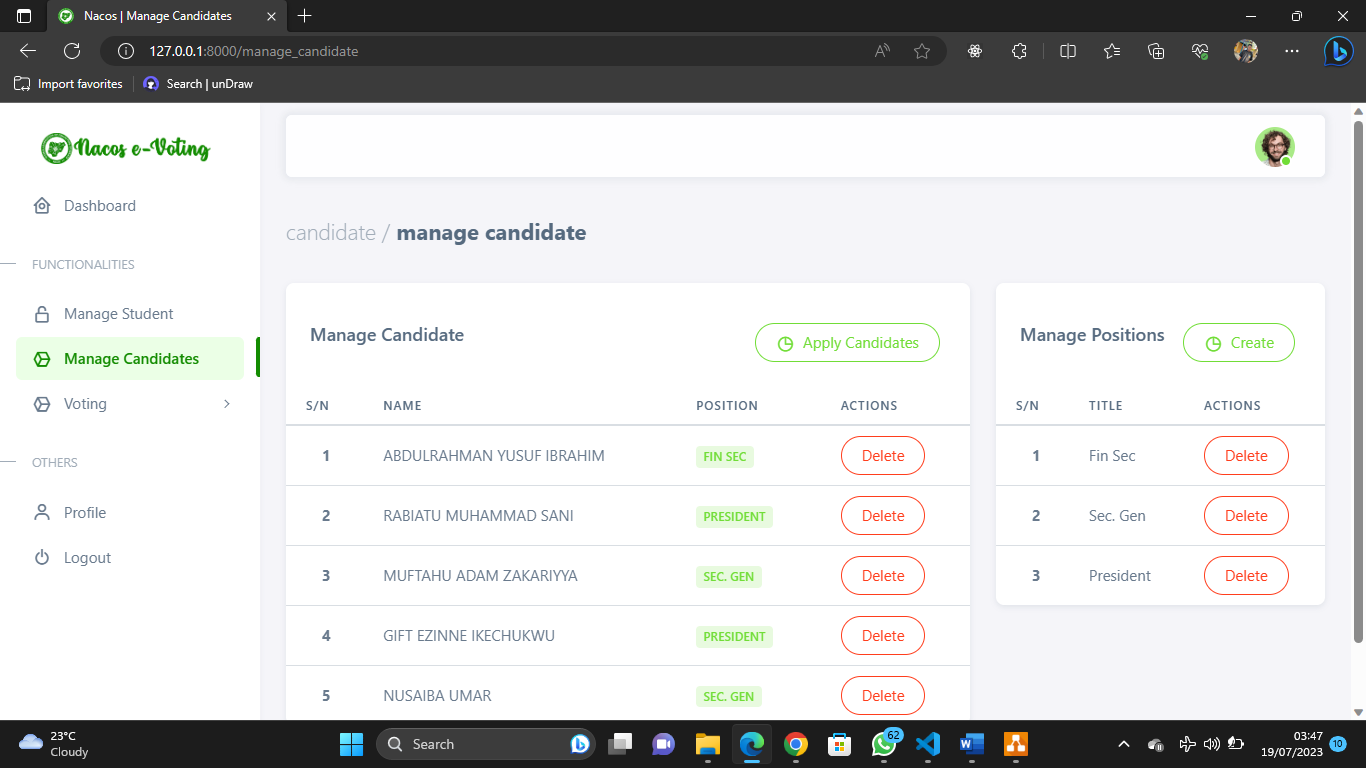


Fig 4.5 Manage Candidates

**View Candidates**

This is the page where the user can view contesting students and their positions

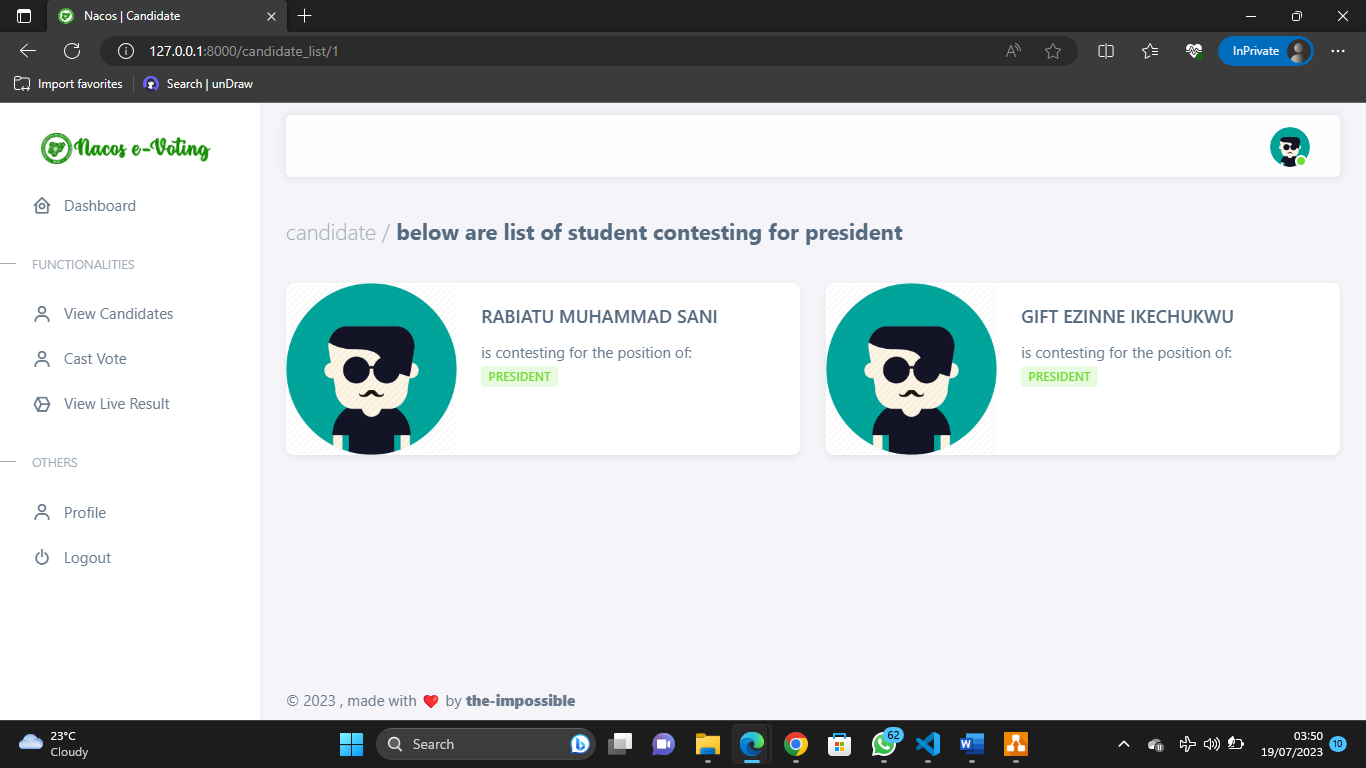


Fig 4.6 View Candidates

**Select Position**

This is the page where the user can select the category, they want to cast their vote for

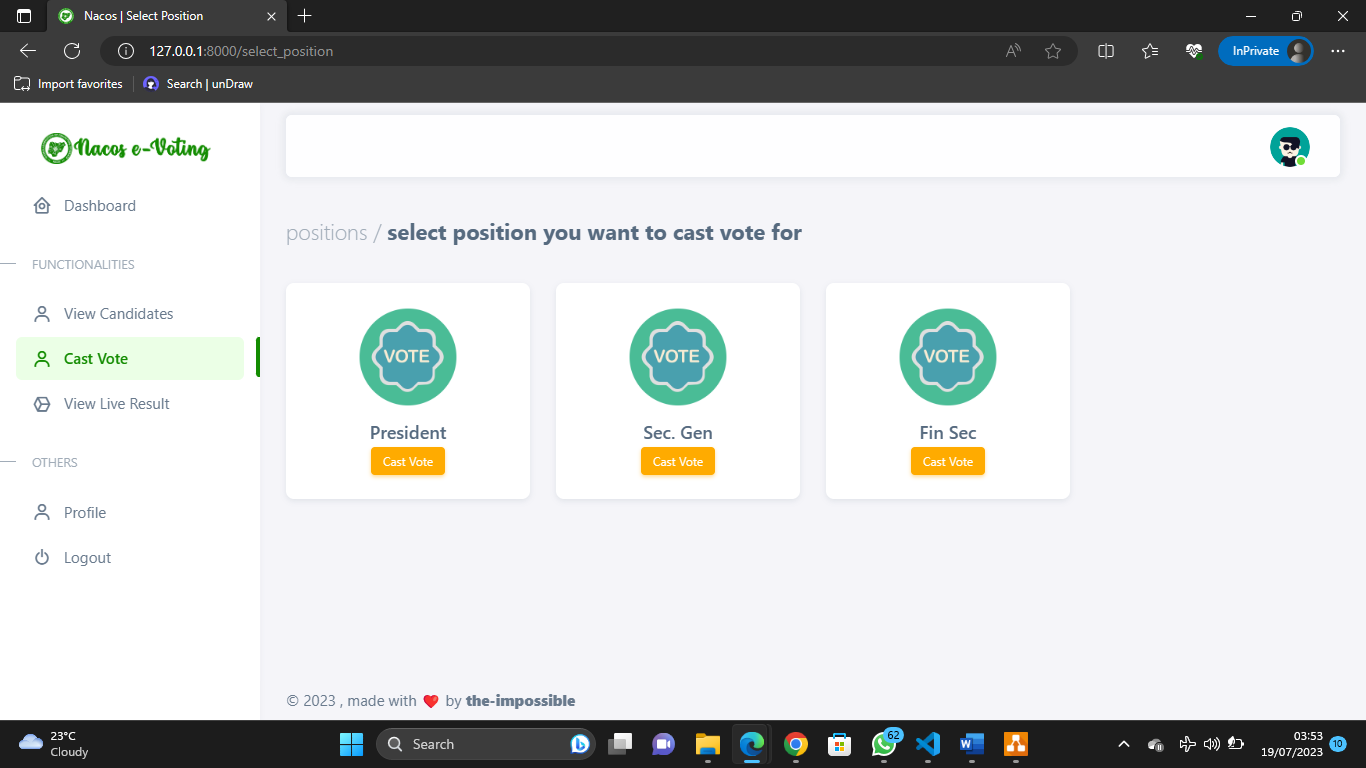


Fig 4.7 Select Position

**Cast Vote**

According to the selected position students can cast their votes

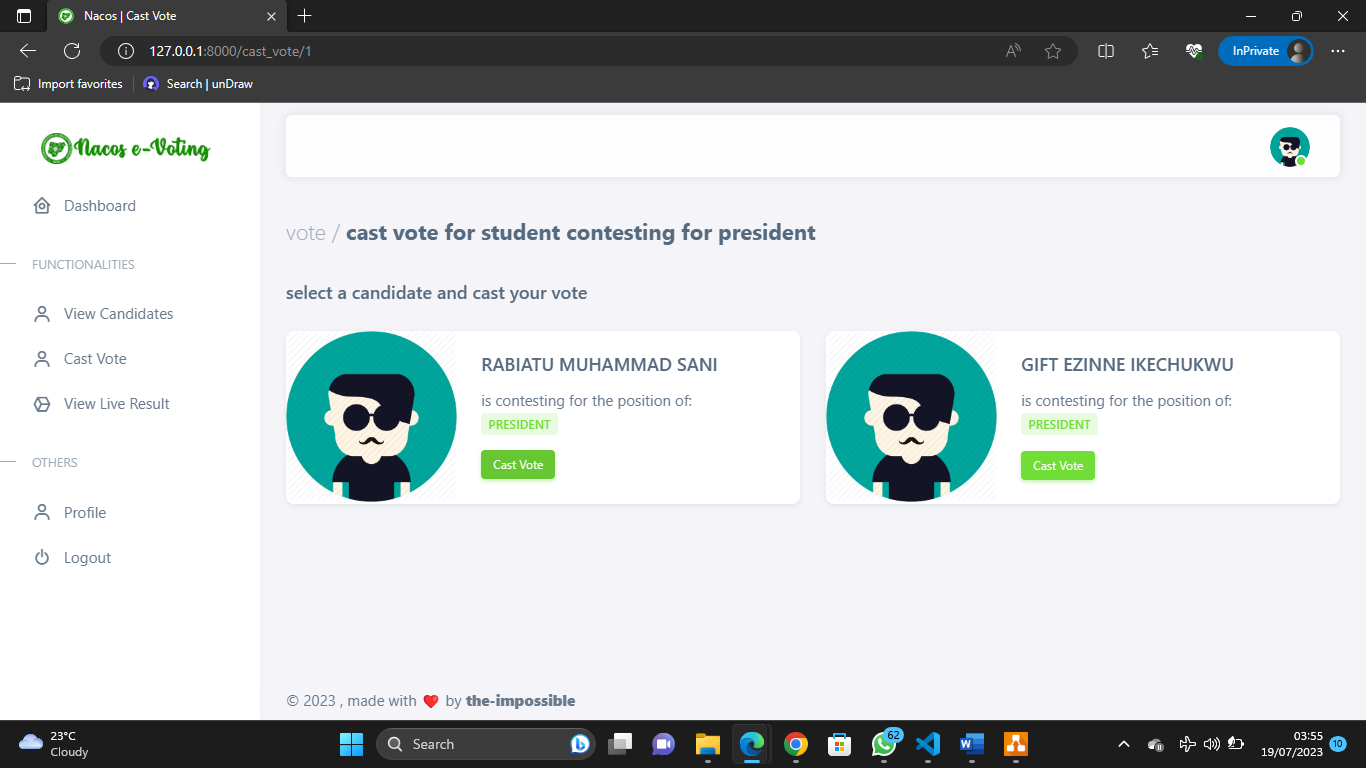


Fig 4.8 Cast Vote

**View Result Statistics**

Students can view live result categorized by the candidate leading in a particular category as it is a real-time system

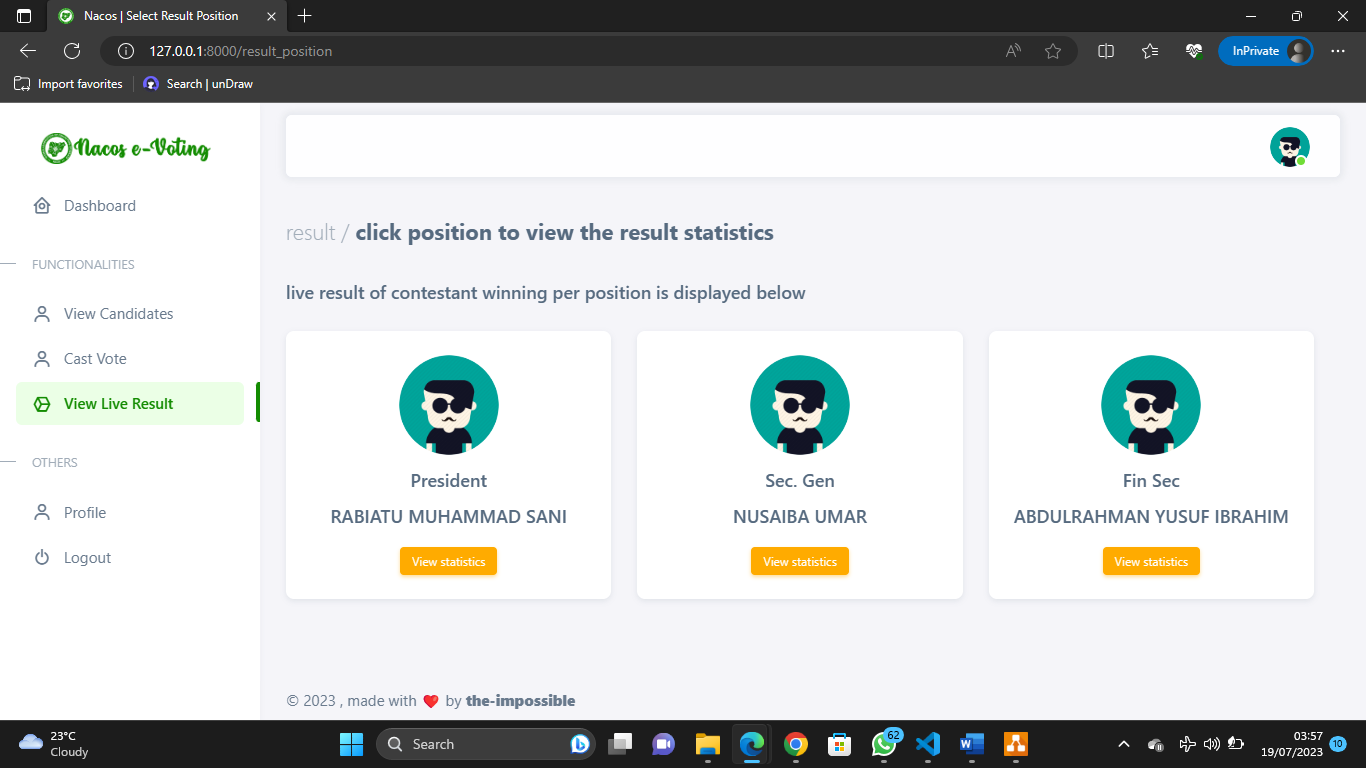


Fig 4.9 View Result Statistics

**Result Details**

Students can view live result details, which show the number of vote a candidate has

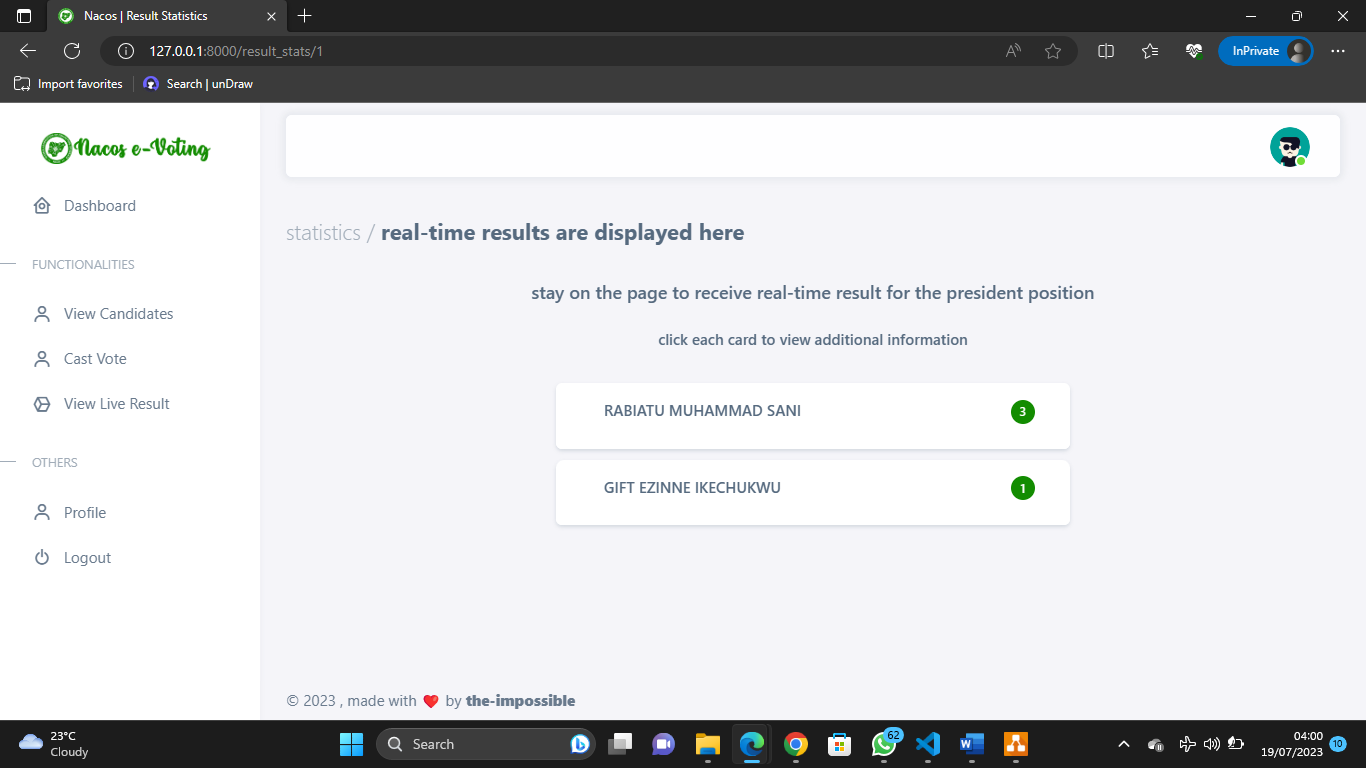


Fig 4.10 Result Details

**Account Profile**

This is the page performs updates on the profile of every authenticated user

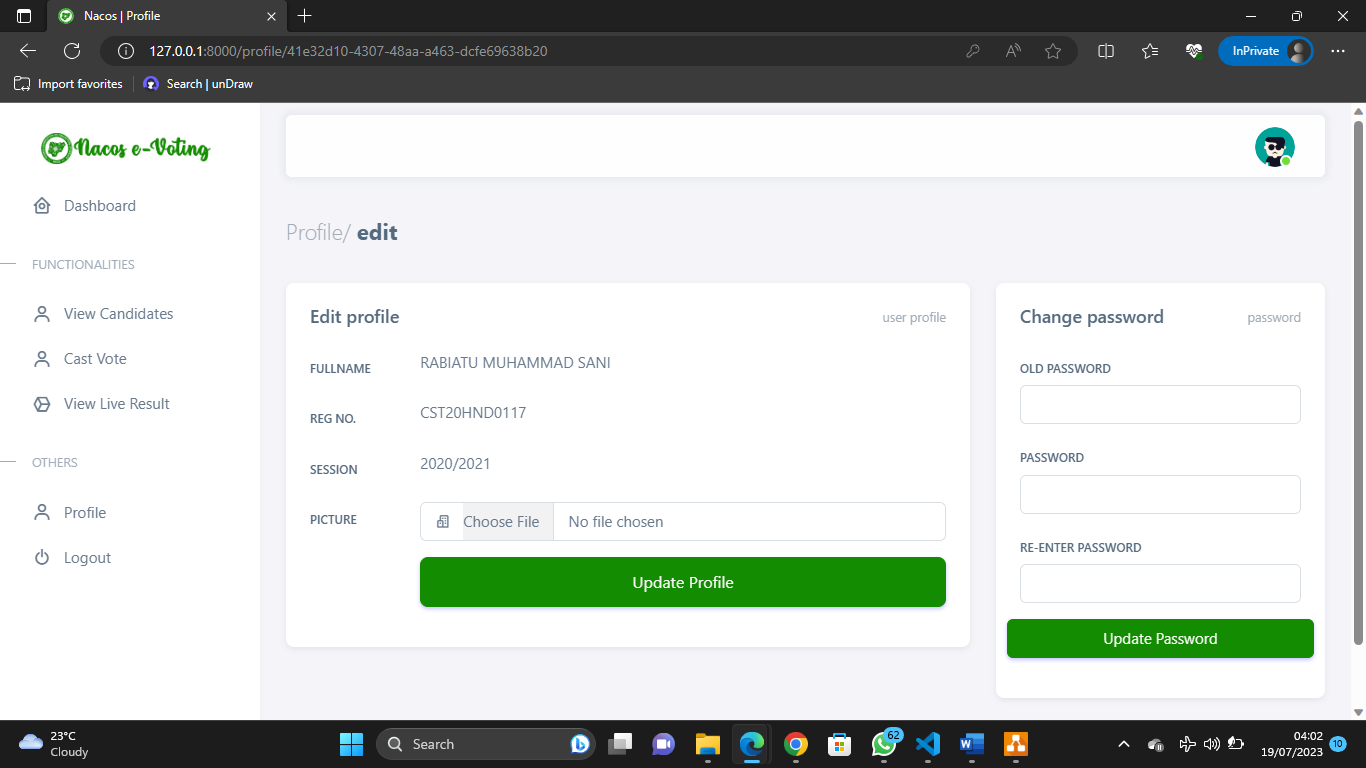


Fig 4.11 Account Profile

**CHAPTER FIVE**

**SUMMARY CONCLUSION AND RECOMMENDATION**

**5.1 Summary**

This study aims to design and implement a comprehensive web-based e-voting system for the computer science department at Kaduna Polytechnic. The current manual voting process poses challenges such as inefficiency, potential errors, and limited accessibility. By developing an electronic voting system, the study seeks to overcome these issues and enhance the overall electoral process. The proposed system will focus on ensuring user-friendliness, security, and accuracy through an intuitive interface, robust testing procedures, and effective data management. By adopting this web-based e-voting solution, the department can streamline voting procedures, increase participation, and foster transparency, ultimately promoting a more democratic and efficient electoral system. Additionally, the study will consider factors such as scalability, adaptability, and adherence to privacy regulations, enabling the system to cater to future needs and uphold the integrity of the voting process.

**5.2 Conclusion**

In conclusion, this study highlights the significance of developing a web-based e-voting system for the computer science department at Kaduna Polytechnic. The current manual voting process presents challenges in terms of efficiency, accuracy, and accessibility. By implementing an electronic voting system, the department can overcome these challenges and improve the overall electoral process. The proposed system focuses on user-friendliness, security, and accuracy through an intuitive interface, rigorous testing, and effective data management. It is anticipated that the adoption of this web-based e-voting solution will streamline voting procedures, increase participation, and foster transparency. Additionally, considerations for scalability, adaptability, and compliance with privacy regulations will ensure the long-term effectiveness and integrity of the voting system. By embracing this technological advancement, the computer science department can establish a more democratic, efficient, and reliable electoral system for the benefit of all stakeholders involved.

**5.2 Recommendation**

Based on the findings of this study, the following recommendations are proposed:

1. Implementation of the Web-based E-voting System: It is recommended that the computer science department at Kaduna Polytechnic proceeds with the implementation of the designed web-based e-voting system. This system will provide numerous benefits, including increased efficiency, accuracy, and accessibility in the electoral process.
2. User-Friendly Interface: The system should prioritize the development of a user-friendly interface that is intuitive and easy to navigate. This will ensure that voters, regardless of their technological expertise, can easily understand and participate in the e-voting process.
3. Rigorous Testing and Validation: It is crucial to conduct thorough testing and validation of the web-based e-voting system before its deployment. This includes testing for security vulnerabilities, system functionality, and usability. Comprehensive testing will help identify and rectify any issues or shortcomings, ensuring a smooth and reliable voting experience.
4. Data Security and Privacy: The system should adhere to stringent data security and privacy measures. This includes implementing encryption protocols, access controls, and secure storage of voter information. Compliance with relevant data protection regulations will enhance trust and confidence in the e-voting system.

By implementing these recommendations, the computer science department at Kaduna Polytechnic can establish a robust and efficient web-based e-voting system. This will contribute to a more democratic, transparent, and inclusive electoral process, fostering trust and confidence among voters.

**REFERENCES**

Indapwar, A. (2020). E-Voting system using Blockchain technology. *International Journal of Advanced Trends in Computer Science and Engineering, 9*(3), 2775–2779. <https://doi.org/10.30534/ijatcse/2020/45932020>

Mishra, A., & Ofujeh Ahmad, A. (2019). Design and Development of Real-Time E-Voting System with High Security Features. *International Journal of Education and Management Engineering*, *9*(3), 37–50. https://doi.org/10.5815/ijeme.2019.03.04

Salleh, A. A. K., Kadir, H. A., Ahmad, A., & Yusof, M. A. M. (2021). “The Implementation of Electronic Voting System for Student Representation Council using reCAPTCHA.” *In IOP Conference Series: Materials Science and Engineering (Vol. 1062). IOP Publishing* Ltd. <https://doi.org/10.1088/1757-899X/1062/1/012043>

Shanthi S., Ilakkiyavani R., & Amsaveni P. (2018). “A Survey on Web Based Application of Secure Online Voting System”. I*nternational Journal of Trend in Scientific Research and Development, 2*(3), 1546–1551. https://doi.org/10.31142/ijtsrd11483

Selvarani, X. I., Shruthi, M., Geethanjali, R., Syamala, R., & Pavithra, S. (2017). Secure voting

System through SMS and using smart phone application.

https://doi.org/10.1109/ICAMMAET.2017.8186724

**APPENDIX**

**Views.py**

from django.shortcuts import render, redirect, reverse

from django.http import HttpResponseRedirect, Http404, HttpResponse

from django.contrib.auth import authenticate, login, logout

from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView, TemplateView

from django.contrib import messages

from django.contrib.messages.views import SuccessMessageMixin

from django.contrib.auth.mixins import LoginRequiredMixin

from django.utils.decorators import method\_decorator

from django.contrib.auth.hashers import make\_password, check\_password

from django.views import View

import csv

import io

import codecs

from django.urls import reverse\_lazy

from eAuth.models import \*

from eAuth.forms import \*

from eAuth.decorators import \*

PASSWORD = '12345678'

# Create your views here.

class HomePageView(TemplateView):

    template\_name = "frontend/index.html"

class LoginPageView(View):

    def get(self, request):

        return render(request, 'backend/auth/login.html')

    def post(self, request):

        username = request.POST.get('username').upper().strip()

        password = request.POST.get('password').strip()

        if username and password:

            user = authenticate(request, username=username, password=password)

            if user:

                if user.is\_active:

                    login(request, user)

                    messages.success(request, f"You are now signed in {user}")

                    nxt = request.GET.get('next', None)

                    if nxt is None:

                        return redirect('auth:dashboard')

                    return redirect(self.request.GET.get('next', None))

                else:

                    messages.warning(

                        request, 'Account not active contact the administrator')

            else:

                messages.error(request, 'Invalid login credentials')

        else:

            messages.error(request, 'All fields are required!!')

        return redirect('auth:login')

class LogoutView(LoginRequiredMixin, View):

    def post(self, request):

        logout(request)

        messages.success(

            request, 'You are successfully logged out, to continue login again')

        return redirect('auth:login')

class ManageStudentView(LoginRequiredMixin, ListView):

    template\_name = "backend/admin/manage\_student.html"

    form\_class = CreateSingleStudentForm

    second\_form\_class = CreateMultipleStudentForm

    def get\_queryset(self):

        return User.objects.filter(is\_superuser=False, is\_staff=False)

    def get\_context\_data(self, \*\*kwargs):

        context = super().get\_context\_data(\*\*kwargs)

        context["form1"] = self.form\_class

        context["form2"] = self.second\_form\_class

        return context

    def post(self, request):

        form1 = self.form\_class(request.POST, request.FILES)

        form2 = self.second\_form\_class(request.POST, request.FILES)

        if 'multiple' in request.POST:

            if form2.is\_valid():

                csv\_obj = csv.reader(codecs.iterdecode(

                    request.FILES['file'], 'utf-8'))

                objs = []

                sub\_objs = []

                session = form2.cleaned\_data.get('session')

                for row in csv\_obj:

                    objs.append(User(

                        username=row[0].upper(), name=row[1], session=session, password=make\_password(PASSWORD)))

                created\_users = User.objects.bulk\_create(objs)

                messages.success(request, "Students has been created")

            else:

                messages.error(request, form2.errors.as\_text())

                return render(request, 'backend/admin/manage\_student.html',

                              context={

                                  'form1': self.form\_class,

                                  'form2': form2,

                                  'object\_list': self.get\_queryset()

                              })

            return HttpResponseRedirect(self.get\_success\_url())

        if 'single' in request.POST:

            if form1.is\_valid():

                instance = form1.save(commit=False)

                instance.password = make\_password(PASSWORD)

                instance.save()

                messages.success(request, "Student created successfully!")

            else:

                messages.error(request, form1.errors.as\_text())

                return render(request, 'backend/admin/manage\_student.html',

                              context={

                                  'form1': form1,

                                  'form2': self.second\_form\_class,

                                  'object\_list': self.get\_queryset()

                              })

            return HttpResponseRedirect(self.get\_success\_url())

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

class UpdateProfileView(LoginRequiredMixin, SuccessMessageMixin, UpdateView):

    model = User

    template\_name = "backend/admin/edit\_delete\_student.html"

    form\_class = EditCreateSingleStudentForm

    success\_message = 'Updated Successfully!'

    def get\_success\_url(self):

        return reverse("auth:manage\_student")

class DeleteStudentView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = User

    success\_message = 'Deleted Successfully!'

    success\_url = reverse\_lazy('auth:manage\_student')

    def post(self, request):

        form1 = self.form\_class(request.POST)

        form2 = self.second\_form\_class(request.POST)

        if 'create\_position' in request.POST:

            if form2.is\_valid():

                form2.save()

                messages.success(request, "Position has been created")

            else:

                messages.error(request, form2.errors.as\_text())

                return render(request, 'backend/admin/manage\_candidate.html',

                              context={

                                  'form1': self.form\_class,

                                  'form2': form2,

                                  'object\_list': self.get\_queryset(),

                                  'object\_list2': self.position\_queryset()

                              })

            return HttpResponseRedirect(self.get\_success\_url())

        if 'apply\_position' in request.POST:

            if form1.is\_valid():

                form1.save()

                messages.success(request, "Student applied successfully!")

            else:

                messages.error(request, form1.errors.as\_text())

                return render(request, 'backend/admin/manage\_candidate.html',

                              context={

                                  'form1': form1,

                                  'form2': self.second\_form\_class,

                                  'object\_list': self.get\_queryset(),

                                  'object\_list2': self.position\_queryset(),

                              })

            return HttpResponseRedirect(self.get\_success\_url())

    def get\_success\_url(self):

        return reverse("auth:manage\_candidate")

class DeleteCandidateView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = Candidates

    success\_message = 'Deleted Successfully!'

    success\_url = reverse\_lazy('auth:manage\_candidate')

class DeletePositionView(LoginRequiredMixin, SuccessMessageMixin, DeleteView):

    model = Positions

    success\_message = 'Deleted Successfully!'

    success\_url = reverse\_lazy('auth:manage\_candidate')

**Homepage**

{% extends 'base.html' %}

{% load static %}

{% block title %}Homepage{% endblock title %}

{% block head %}{% include 'partials/head.html' %}{% endblock head %}

  {% block body %}

    <!-- ===============================================-->

    <!--    Main Content-->

    <!-- ===============================================-->

    <main class="main" id="home">

      {% block nav %} {% include 'partials/nav.html' %} {% endblock %}

      <section class="pt-7">

        <div class="container">

          <div class="row align-items-center">

            <div class="col-md-6 text-md-start text-center py-6">

              <h1 class="mb-4 fs-9 fw-bold">Nacos e-Voting Website</h1>

              <p class="mb-6 lead text-secondary">We provide a safe and secured voting process<br class="d-none d-xl-block" />Let your voice be heard, make your vote count.</p>

              <div class="text-center text-md-start">

                <a class="btn btn-warning me-3 btn-lg" href="{% url 'auth:login' %}" role="button">Get started</a>

              </div>

            </div>

            <div class="col-md-6 text-end"><img class="pt-7 pt-md-0 img-fluid" src="{% static 'frontend/assets/img/hero/voting.png' %}" alt="" /></div>

          </div>

        </div>

      </section>

      <!-- ============================================-->

      <!-- <section> begin ============================-->

      <section class="pt-5 pt-md-9 mb-6" id="feature">

        <div class="bg-holder z-index--1 bottom-0 d-none d-lg-block" style="background-image:url({% static 'frontend/assets/img/category/shape.png' %});opacity:.5;">

        </div>

        <!--/.bg-holder-->

        <div class="container">

          <h1 class="fs-9 fw-bold mb-4 text-center"> Powered Features</h1>

          <div class="row">

            <div class="col-lg-3 col-sm-6 mb-2"> <img class="mb-3 ms-n3" src="{% static 'frontend/assets/img/category/icon1.png' %}" width="75" alt="Feature" />

              <h4 class="mb-3">Great UI/UX experience</h4>

              <p class="mb-0 fw-medium text-secondary">Good user interface and user experience,</p>

            </div>

            <div class="col-lg-3 col-sm-6 mb-2"> <img class="mb-3 ms-n3" src="{% static 'frontend/assets/img/category/icon2.png' %}" width="75" alt="Feature" />

              <h4 class="mb-3">Highly Secured</h4>

              <p class="mb-0 fw-medium text-secondary">Your informations are safe with us</p>

            </div>

            <div class="col-lg-3 col-sm-6 mb-2"> <img class="mb-3 ms-n3" src="{% static 'frontend/assets/img/category/icon3.png' %}" width="75" alt="Feature" />

              <h4 class="mb-3">High-end Performance</h4>

              <p class="mb-0 fw-medium text-secondary">Great in terms of speed.</p>

            </div>

            <div class="col-lg-3 col-sm-6 mb-2"> <img class="mb-3 ms-n3" src="{% static 'frontend/assets/img/category/icon4.png' %}" width="75" alt="Feature" />

              <h4 class="mb-3">Transparency</h4>

              <p class="mb-0 fw-medium text-secondary">Real-time voting scores.</p>

            </div>

          </div>

          <div class="text-center"><a class="btn btn-warning mt-5" href="{% url 'auth:login' %}" role="button">Sign In</a></div>

        </div><!-- end of .container-->

      </section>

      <!-- <section> close ============================-->

      <!-- ============================================-->

      <!-- ============================================-->

      <!-- <section> begin ============================-->

      {% comment %} <section class="py-md-11 py-8" id="result">

        <div class="bg-holder z-index--1 bottom-0 d-none d-lg-block background-position-top" style="background-image:url({% static 'frontend/assets/img/superhero/oval.png' %});opacity:.5; background-position: top !important ;">

        </div>

        <!--/.bg-holder-->

        <div class="container">

          <div class="row justify-content-center">

            <div class="col-lg-6 text-center">

              <h1 class="fw-bold mb-4 fs-7">Voting Result</h1>

              <p class="mb-5 text-info fw-medium">Live voting result will be displayed here<br /></p>

              <a href="{% url 'auth:login' %}" class="btn btn-warning btn-md">Sign In</a>

            </div>

          </div>

        </div><!-- end of .container-->

      </section> {% endcomment %}

      <!-- <section> close ============================-->

      <!-- ============================================-->

    {% block footer %}{% include 'partials/footer.html' %}{% endblock footer %}

    </main>

    <!-- ===============================================-->

    <!--    End of Main Content-->

    <!-- ===============================================-->

    {% block script %}{% include 'partials/script.html' %}{% endblock script %}

  {% endblock body %}