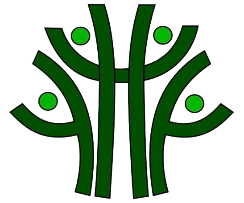
BugSlayers presents VotaMzansi



Contents

[Team Members and their Roles 2](#_Toc167366587)

[Project Summary 2](#_Toc167366588)

[Problem Statement 2](#_Toc167366589)

[Project Aim and Objectives 2](#_Toc167366590)

[Leading factors 2](#_Toc167366591)

[Functional Requirements 3](#_Toc167366592)

[System Design 3](#_Toc167366593)

[Introduction 3](#_Toc167366594)

[Purpose and Scope 3](#_Toc167366595)

[Project executive summary 3](#_Toc167366596)

# Team Members and their Roles

1. MT Chikoti 223090834 - UI/UX developer
2. N Nzimande 223203835 - UI/UX developer and Reviewer
3. S Hokwana 222078158 - Tester and performance insurer
4. MD Lichaba 222196880 - Backend developer and graphic designer
5. M Yusufu 222079860 - Backend developer
6. AS Maduna 223036497 - Analyst, Group Leader, Documenter

|  |  |
| --- | --- |
| Table For LInks and submission Information | |
| Team Name | Bug Slayers |
| team email address | votamzansi@gmail.com |
| team leader info | AS Maduna - Analyst, Group Leader, Documenter |
| GIthub link | <https://github.com/Maxwell-T-Chikoti/Bug_Slayers-DSW02A1_Group_Project> |
| trello link | <https://trello.com/b/dfQWEHc6/vota-mzansi-project> |
|  |  |
| Team member names | MT Chikoti  N Nzimande  S Hokwana  MD Lichaba  M Yusufu  AS Maduna |

# Project Summary

## Problem Statement

We live in the 21st century, where many services have become digital. Which saves people time and money. However, presidential elections have not yet become digital even though the whole country is due to partake in the elections meaning there will be long ques. Which will be time consuming, exhausting, and inconvenient for people that would like to vote but held up by other commitments in their lives.

## Project Aim and Objectives

To address this issue, Bug Slayers is introducing an innovative solution in the form of a user-friendly web application. This digital platform allows citizens to cast their votes for the presidency effortlessly, using just the tips of their fingers. This initiative seeks to modernize the electoral process, making it more accessible and efficient for the entire population. The Bug Slayers' web application ensures security, effectiveness, and inclusivity, catering to voters of all ages, abilities, and levels of technological familiarity. By embracing digital voting, this solution aims to enhance the democratic process, reduce the burden on citizens, and encourage broader participation in elections, ultimately contributing to a more streamlined and accessible electoral system.

## Leading factors

1. Lack of security and inaccuracy: There have been incidents where vehicles transporting voting ballots were hijacked and some burnt. This makes the voting results inaccurate. The traditional way of voting is also not so safe for the livelihood of the country’s citizens who take part in the election event.
2. Time consuming: Some people don’t have the time to stand in long ques for hours. Not only is this time consuming but its also tiresome.
3. Accessibility: Some people travel a lot the traditional voting event is not convenient for them, since the traditional voting event requires you to vote where you registered were registered. VotaMzansi is convenient and accessible.

## Functional Requirements

1. User Registration and Profile Management:

* Users can create accounts with their ID numbers and create passwords for themselves.
* Users can edit their profiles, including their personal information.

1. User Accounts:

* Users can add, edit, and delete their accounts.
* Users have access to Voting ballots, as well as Voting results.
* Users have access to information about the running parties and their candidates.
* Users should receive notifications for the OTP as well as the results for the winning party.

1. Candidates:

* Candidates can create accounts with their License IDs
* Candidates can register their parties at a fee.

1. Admins:

* Login using their ID numbers and Admin key/password.

1. Education and Information:

* Links to the running parties’ websites are provided so that users can have insight of the parties and make informed decisions.

1. System:

* OTP API for authentication.
* ReCAPTCHA API for security.
* PayFast API for monetizing.
* Web application coded using HTML, CSS, JavaScript, PHP, and MySQL.

# System Design

## Introduction

In today’s fast-paced world, time is crucial, and efficiency is key. VotaMzansi is a well-designed system that ensures the safety and confidentiality of user’s sensitive information and their votes. Our website not only allows users to create secure and safe accounts, but it also gives users access to information that is valuable enough for users to make an informed decision when voting. Our voting website aims to save time and ensure integrity within the voting event and process.

## Purpose

The purpose of the website is to make the voting process easier and convenient for the citizens of the country. It helps users save money and time as they can vote at the comfort of their home with just the tips of their fingers.

The primary objective of this system are as follows:

* The design is user friendly.
* The system is secure and reliable.
* The system is well informative, and it provides a seamless and intuitive user experience.

## Scope

The scope of this system design encompasses the key components and functionalities of a voting web application. The following are the major aspects included within the scope of this system design:

Voter Registration: The design of the website clearly states the process of user registration, authentication as well as profile management.

Storage and safety: The system is well-designed to ensure user information is safe and confidential. Users are also able to Create account, modify their accounts as well as delete their accounts.

### UML class diagram

A screenshot of a computer

Description automatically generated



This UML diagram represents an online voting system, detailing how different classes interact within the system. Each class represents a component or actor in the system and includes its attributes (data fields) and methods (functions or operations it can perform).

* User/Voter

Represents a person who participates in the voting process.

Attributes:

ID: Unique identifier for the user.

Username: The user's username.

UserSurname: The user's surname.

Email: The user's email address.

Password: The user's password.

Address: The user's physical address.

Methods:

Login(Type): Allows the user to log into the system.

Register(Type, Type): Allows the user to register for the system.

Vote(Type): Allows the user to cast a vote.

* Admin

Represents a system administrator who manages the voting events.

Attributes:

ID: Unique identifier for the admin.

UserName: The admin's username.

Email: The admin's email address.

Password: The admin's password.

Role: The admin's role in the system.

Methods:

Login(): Allows the admin to log into the system.

CreateVotingEvent(): Allows the admin to create a new voting event.

UpdateVotingEvent(): Allows the admin to update an existing voting event.

CloseVoting(): Allows the admin to close a voting event.

* VotingEvent

Represents a specific event during which voting takes place.

Attributes:

ID: Unique identifier for the voting event.

Title: The title of the voting event.

Description: A description of the voting event.

StartDate: The date the voting event starts.

EndDate: The date the voting event ends.

Status: The current status of the voting event (e.g., open, closed).

Candidate: A list of candidates participating in the event.

Methods:

Create(): Creates a new voting event.

Update(): Updates the details of the voting event.

Close(): Closes the voting event.

* Candidate

Represents a person or option that can be voted for in a voting event.

Attributes:

ID: Unique identifier for the candidate.

Name: The name of the candidate.

Description: A description of the candidate.

Methods:

addVote(): Method to record a vote for the candidate.

* Ballot

Represents the actual ballot cast by a voter in a voting event.

Attributes:

BallotID: Unique identifier for the ballot.

VotingEvent: The voting event associated with the ballot.

Parties: List of candidates involved in the ballot.

TimeStamp: The date and time when the vote were cast.

* Vote

Represents an individual vote cast by a user for a candidate.

Attributes:

ID: Unique identifier for the vote.

User: The user who cast the vote.

Candidate: The candidate who received the vote.

Timestamp: The date and time when the vote was cast.

* Message

Represents a message sent within the system.

Attributes:

ID: Unique identifier for the message.

Recipient: The user who receives the message.

Message: The content of the message.

TimeStamp: The date and time when the message were sent.

Status: The status of the message (e.g., sent, received).

* Role

Represents a role in the system, defining a set of permissions.

Attributes:

ID: Unique identifier for the role.

Name: The name of the role.

Permissions: A list of permissions associated with the role.

Relationships

User/Voter and Vote:

A user can cast multiple votes, and each vote is associated with one user.

Vote and Candidate:

Each vote is for one candidate, and a candidate can receive multiple votes.

VotingEvent and Candidate:

A voting event can have multiple candidates, and each candidate can participate in multiple voting events.

Ballot and VotingEvent:

Each ballot is associated with one voting event, and a voting event can have multiple ballots.

Ballot and Candidate:

Each ballot can involve multiple candidates.

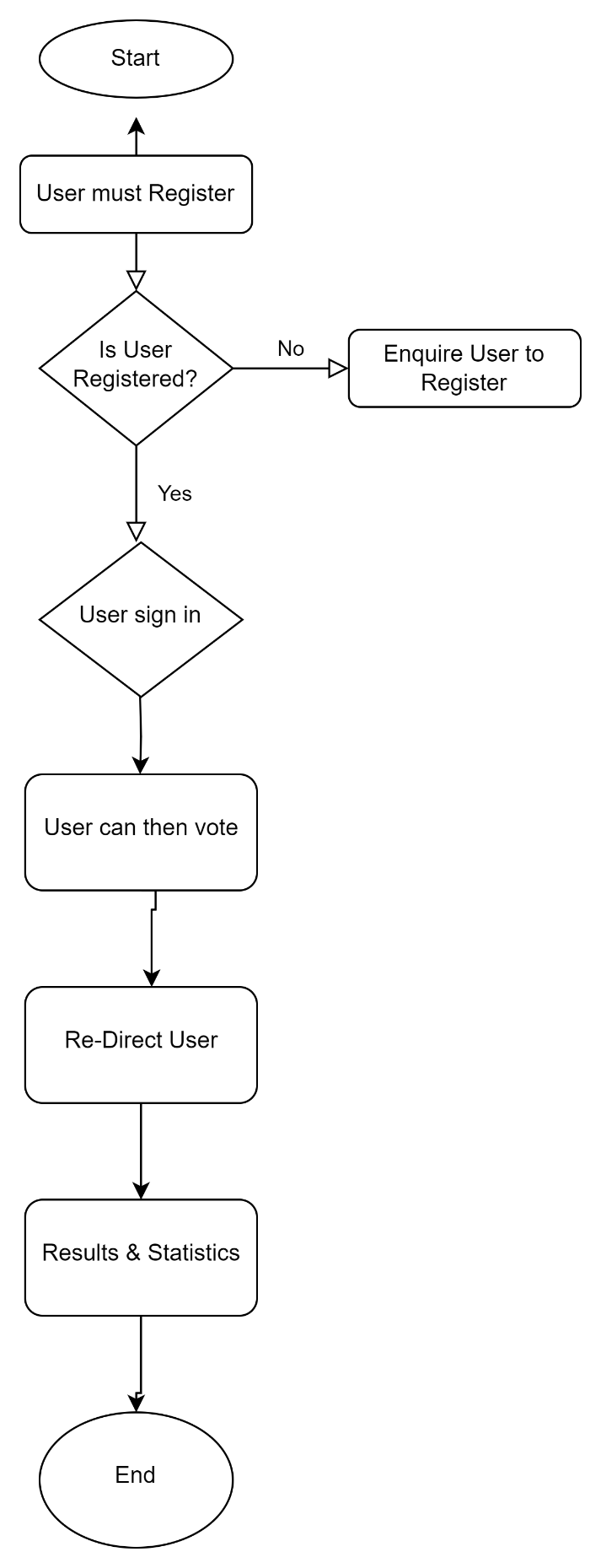
Message and User:

Each message is sent to one user, and a user can receive multiple messages.

This diagram outlines the structure and interactions within the online voting system, helping to understand the roles and processes involved.



### Screen flow and Use-Case

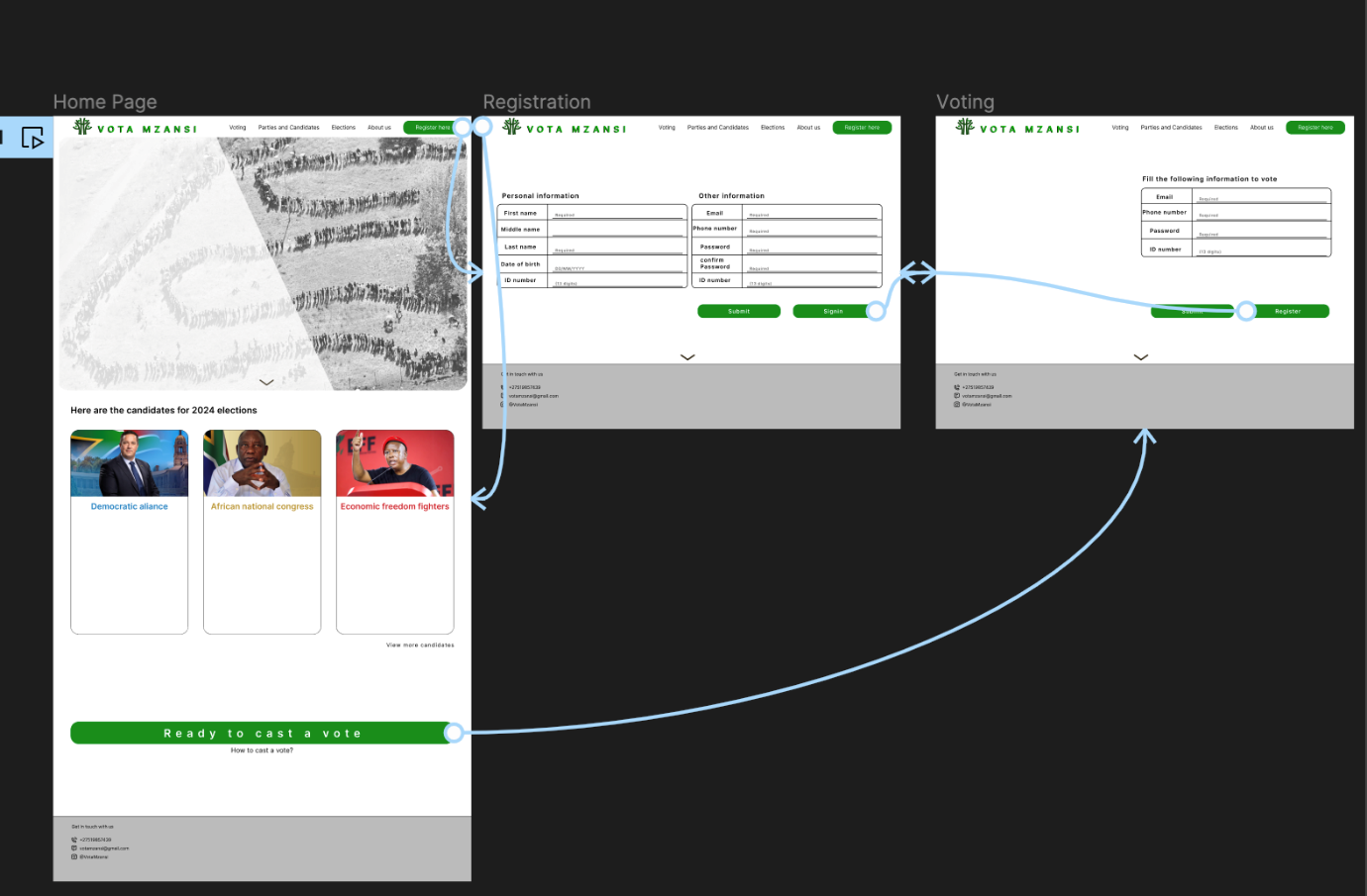


This flowchart explains the steps a user needs to follow to vote on a website. Here’s a simple explanation:

1. Start: This is where the process begins.
2. User must Register: The user needs to sign up on the website first.
3. Is User Registered? The system checks if the user is already registered.
   1. If No, the user is asked to register.
   2. If yes, the user can sign in.
4. User sign in: The user logs into the website.
5. User can then vote: The user is then allowed to vote.
6. Re-Direct User: After voting, the user is redirected to another page.
7. Results & Statistics: The user can see the voting results and statistics.
8. End: The process is complete

# Web layout mock-ups

Registration Process:



User first gets into the web application’s home page which has access to the Sign up (Registration), Log In, and Candidates’ pages. As well as the Contacts for VotaMzansi.

Registration: User must give the required information of themselves for them to create an account.

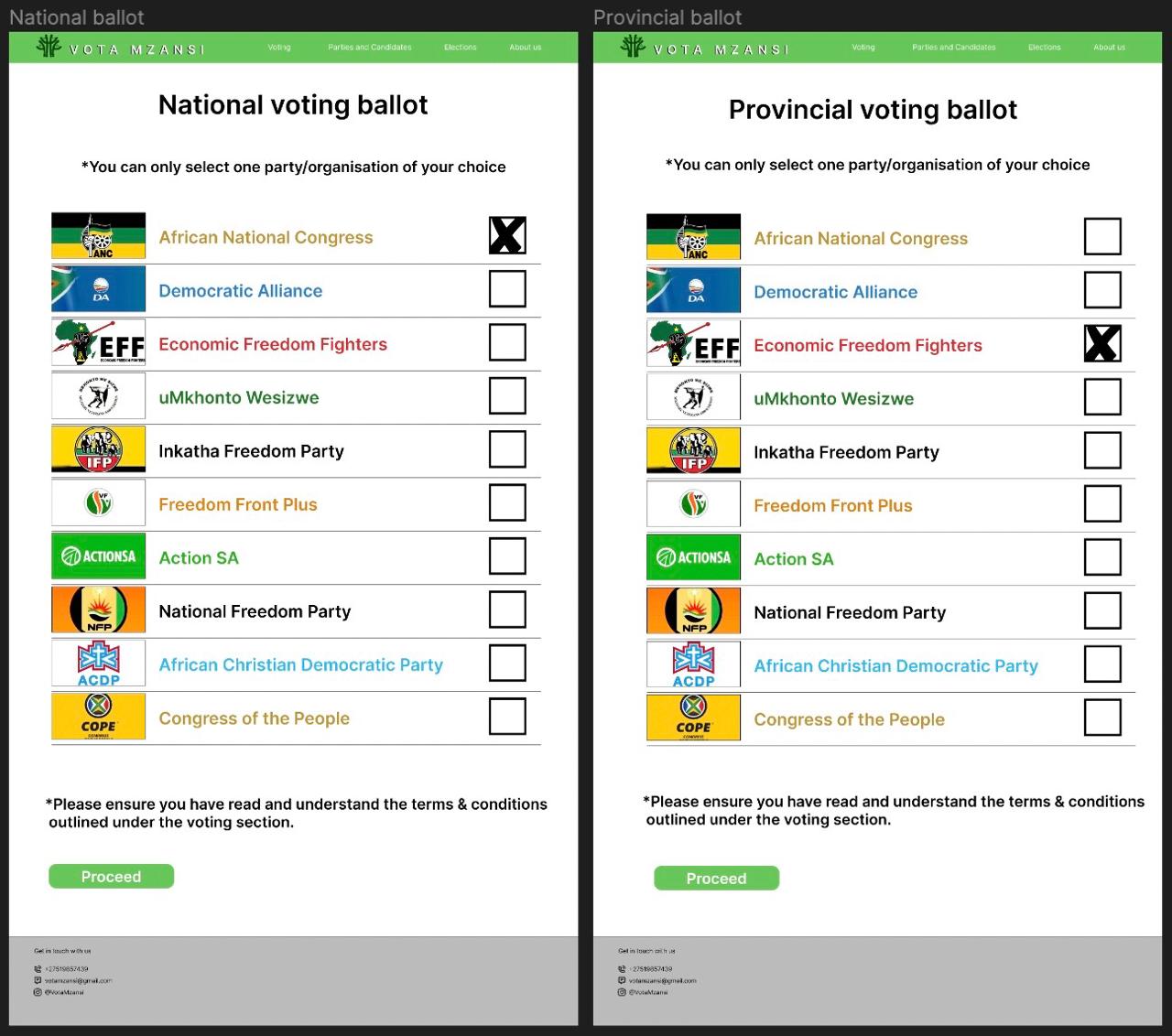
Log In:

A screenshot of a login form

Description automatically generated

Login: User will either get the Voting Ballots, on the Election period, or an alternative page that informs them of the details of the Elections.

Voting:



Voting Process: A Registered User will get a National Voting ballot and a Provincial Voting Ballot and after the user submits their votes they will be redirected to a page. Where users will be informed about the results.

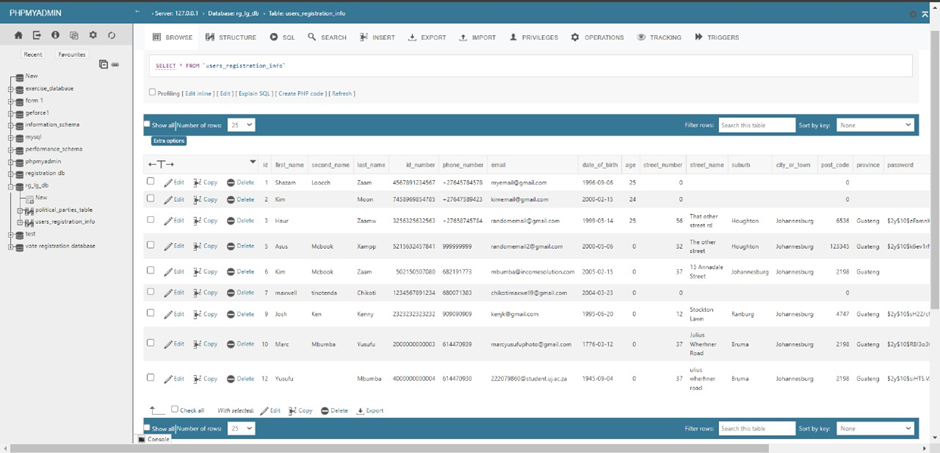
Results:

A screenshot of a computer

Description automatically generated

Election results: The system will count the voting ballots which are stored in a database. Then the total number of voters will be displayed for each party.

Database:



A screenshot of a computer

Description automatically generated

Database and storage:

* The database is only accessed by the admin and user’s information is confidential, even to the administrator. The database has all user accounts, and users can update their accounts and even delete them.
* The voting ballots then gets stored in the database and they get retrieved by the system which then counts the votes.

Admin Portal:A screenshot of a computer

Description automatically generatedA screenshot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generated

The admin portal is an online platform designed to give administrators exclusive access to the web application. It allows for efficient management of user registrations, statistical data, parties, and candidates information. With a user-friendly interface, the portal streamlines administrative tasks and enhances operational control. This ensures smooth application performance and informed decision-making.

The portal features a sidebar with essential links for easy navigation. Administrators can access the Dashboard for an overview of key metrics, the Statistics page for detailed data analysis, and the Parties page for managing party and candidate information. The sidebar also includes a logout button, ensuring secure sign-out from the portal.

The Dashboard site.

● The Dashboard provides a comprehensive overview of the application's performance. It includes a statistics column offering a snapshot of the data available on the Statistics page, helping administrators quickly assess key metrics. Additionally, a parties column lists all registered parties and candidates, enabling easy management and updates.

The Statistics site.

● The Statistics page is dedicated to in-depth data analysis. It features a detailed graph that shows the number of registered voters in each province, allowing administrators to monitor registration trends and make data-driven decisions. This page is crucial for understanding voter distribution and planning targeted outreach efforts.

The Parties site.

The Parties page lists all the registered parties and candidates, providing a central location for managing political entities within the application. Administrators can add, update, or delete party and candidate information, ensuring that the data remains accurate and up to date. This page is essential for maintaining a comprehensive and organised record of all political participants.

Candidate Portal:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Candidate portal

• Sign-up and Information Submission: Users provide their address, cellphone number, party name, ID Number, and candidacy name on the first form. On the next page, once you click Proceed, they provide you more information about the political party like the party address, town/city, and province the offices are located in.

• After filling out form #2 user will click the Pay Now button, allowing users to pay the party registration fee, following payment confirmation, the administrative staff checks the data and the party will be added to the ballots .

# Explanation of Tools Integrated

* Figma:

we used figma to come up and design concepts we then further developed for our homepage , login portals , voting ballots and admin portal.

* Visual studio code:

we used visual studio to bring our ideas into life by using the software to code our concepts to see how they would look when brought to life we used JavaScript to add functionality CSS to add colour and design and HTML to give structure.

* GitHub facilitated collaboration:

code sharing among team members, enabling efficient project management, bug tracking, and continuous integration as a group project.

* Trello:

Keep track of tasks and progress of the project and assign projects t team members.

* MySQL:

Designed the database and tables.

* Хаmpp:

Used it as a local host for the website anc a local server for the sql database.

* Bootstrap:

Used CSS code libraries to style the website.

* Drawio:

For the UML, Database Schema and Flow chart diagrams.