

**ATMIYA UNIVERSITY**

**For the award of the degree of**

**Bachelor of Science in Information Technology**

**Semester – VI (2022-2023)**

**Project Report on**

**LostVote**

**Submitted by**

**Sojitra Mahin Nirmalbhai**

**Submitted to**

**Department of Computer Application & Information Technology**

**Atmiya University**

**LostVote**

**Be bright, vote for the right.**

A PROJECT SUBMITTED TO

**Atmiya University**

**Department of Computer Application & Information Technology**

**RAJKOT**



*Submitted in partial fulfilment of the requirements for the degree of*

**“Bachelor of Science in Information Technology”**

**Sem-6**

**(Year 2022-2023)**

**Submitted By: - Guided By: -**

**Sojitra Mahin Nirmalbhai Dr. Ripal Ranpara**

**Pithva Yagnik Mahendrabhai Mr. Malay Solanki**

Bottom of Form**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Serial No** | **Description** | | | **Page No** |
|  | **Acknowledgement** | | | **5** |
|  | **Abstract** | | | **6** |
| **1.** | **Introduction** | | | **7** |
|  | 1.1 | Project Summary | | 8 |
|  | 1.2 | Purpose | | 9 |
|  | 1.3 | Scope | | 10 |
| **2.** | **Literature Review** | | | **11** |
|  | 2.1 | HTML | | 12 |
|  | 2.2 | CSS | | 15 |
|  | 2.3 | JavaScript | | 18 |
|  | 2.4 | JSP (Java Server Pages) | | 20 |
|  | 2.5 | MySQL | | 22 |
| **3.** | **Project Management** | | | **24** |
|  | 3.1 | Project Planning and Scheduling | | 25 |
|  |  | 3.1.1 | Project Development Approach | 25 |
|  | 3.1.2 | Project Plan | 26 |
|  | 3.1.3 | Schedule Representation | 28 |
| **4.** | **System Requirements Specification** | | | **30** |
|  | 4.1 | User Characteristics | | 31 |
|  | 4.2 | Hardware and Software Requirements | | 32 |
| **5.** | **System Analysis** | | | **33** |
|  | 5.1 | Feasibility Study | | 34 |
|  |  | 5.1.1 | Technical Feasibility | 34 |
|  | 5.1.2 | Economical Feasibility | 35 |
|  | 5.1.3 | Operational Feasibility | 35 |
|  | 5.2 | Functions Of System | | 36 |
|  |  | 5.2.1 | Use Case Diagram | 37 |
|  | 5.3 | Data Modelling | | 41 |
|  |  | 5.3.1 | E-R Diagram | 42 |
|  | 5.3.2 | Activity Diagram | 44 |
|  | 5.4 | Functional and Behavioral Modelling | | 47 |
|  |  | 5.4.1 | Data Flow Diagram | 48 |
| **6.** | **System Design** | | | **52** |
|  | 6.1 | Database Schema Design | | 53 |
| **7.** | **Implementation** | | | **58** |
| **8.** | **Conclusion** | | | **80** |
|  | **References** | | | **82** |

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to all those who have contributed to the completion of this project.

First and foremost, we would like to thank our supervisor Dr. Ripal Ranpara, for her invaluable guidance, support, and encouragement throughout the project. Her insights and suggestions were instrumental in shaping our work and bringing it to completion.

We are also grateful to the staff at Atmiya University, who provided us with the necessary resources, equipment, and facilities to carry out this project. Their prompt assistance and cooperation were essential in ensuring the smooth progress of our work.

We extend our appreciation to our colleagues and friends who supported us during the project, providing valuable feedback and suggestions on our work. Their constructive criticism helped us to improve the quality of our work and contributed to the success of this project.

Finally, we would like to thank our families for their unwavering support and encouragement throughout the project. Their constant love and support were crucial in helping us to overcome the challenges and complete this project.

Thank you all for your contributions and support.

Sincerely,

Mahin Sojitra and Yagnik Pithva.

**ABSTRACT**

E-voting systems are becoming popular with the widespread use of computers and embedded systems. Security is the vital issue should be considered in such systems.

This paper proposes a new e-voting system that fulfils the security requirements of e-voting. The proposed system is implemented on web which serves as a voting machine. It is especially for companies who used to take vote for a particular positioning in company. Employees must visit the company voting booth to cast their vote in the present system. This system is online and hence even people who live out of their hometown can also vote.

In **“LostVote”** an employee login as voter and can use his\her voting right online without any difficulty. It totally webs based system. Admin adds employees of company as voter and se nd credentials as email message on employees registered email address. For casting vote user have to login with their credentials in system once at the same time. This system reduces false vote. And provide a powerful portal for voting from anywhere.

Chapter 1: Introduction

**1.1 Project Summary**

An election system is a complex set of processes and technologies that allow voters to participate in a fair, secure, and transparent democratic election. The main goal of an election system is to ensure that every eligible voter can cast a vote that will be counted accurately, and that the results of the election are accepted as legitimate by all stakeholders.

Developing an election system involves multiple components, such as voter registration, ballot design, voter authentication, vote casting, and vote counting. The election system must also adhere to legal and ethical standards, such as ensuring that all eligible voters have equal access to voting, protecting the privacy of voter information, and preventing fraud and manipulation of votes.

An election system used in companies is a software-based platform that allows employees to vote on various issues, such as electing representatives, making company-wide decisions, or providing feedback on workplace policies. The goal of the system is to facilitate a fair, transparent, and democratic process that empowers employees to participate in shaping the direction and culture of the company.

Election system for companies involves several components, such as designing a user-friendly interface, implementing security measures to protect voter information, and ensuring that the system complies with relevant laws and regulations. The system may also include features such as voter authentication, voter registration, and election result.

In **“LostVote”** an employee login as voter and can use his\her voting right online without any difficulty. It totally webs based system. Admin adds employees of company as voter and send credentials as email message on employees registered email address. For casting vote user have to login with their credentials in system once at the same time. This system reduces false vote. And provide a powerful portal for voting from anywhere.

Overall, an effective election system used in companies can help to foster a culture of participation and engagement among employees. It can also provide valuable feedback and insights that can inform the decision-making process and improve the overall functioning of the company and choose great leaders for a particular position.

**1.2 Purpose**

The purpose of developing an online election system for companies is to provide a secure, efficient, and accessible platform for employees to participate in decision-making processes within the company and choose and effective and great leader for a particular position. An online election system eliminates many of the challenges and barriers associated with traditional paper-based voting systems, such as limited accessibility, time-consuming procedures, and problems in vote counting.

* The key objectives of developing an online election system for companies are as follows:

**Accessibility:** An online election system enables employees to participate in the voting process from anywhere at any time, using any device with an internet connection. This allows more employees to participate in decision-making processes, including those who might not be able to physically attend a meeting or vote in-person due to work commitments, geographic barriers, or health reasons.

**Efficiency:** An online election system eliminates many of the time-consuming and labour-intensive processes associated with paper-based voting, such as printing and distributing ballots, manually counting votes, and resolving discrepancies. This saves time and reduces the likelihood of errors or delays in the voting process.

**Security:** An online election system can provide enhanced security measures, such as encryption, multi-factor authentication, and audit trails, to protect the integrity of the voting process and prevent fraud and manipulation.

**Transparency:** An online election system can provide greater transparency and accountability by allowing employees to track their votes and ensuring that the vote count is accurate and verifiable.

**Cost-effectiveness:** An online election system can be more cost-effective than traditional paper-based voting systems, as it eliminates the need for printing, distributing, and storing paper ballots, and reduces the need for staffing and equipment at voting locations.

**1.3 Scope**

The scope of an online election system for companies is to provide a digital platform for employees to participate in various types of elections, polls, surveys, and other decision-making processes. The system must be designed to ensure that it is secure, efficient, accessible, and easy to use.

* The following are the key components of the scope of an online election system:

**Conducting Elections:** The online election system must be capable of handling various types of elections, such as board member elections, union elections, and referendums. The system must be able to handle complex voting scenarios such as weighted voting, proxy voting, secret ballot, and multiple rounds of voting. The system should also be able to handle other types of polls and surveys to gather feedback from employees on various topics.

**Ensuring Security:** The online election system must ensure the security and integrity of the voting process. The system should have user authentication, encryption, and audit trails to prevent fraud and manipulation.

**Accessibility:** The online election system must be accessible to all eligible voters, regardless of their location or device. The system should be developed with an intuitive and user-friendly interface and should be accessible on various devices such as desktops, laptops, tablets, and smartphones.

**Maintenance and Support:** The online election system requires ongoing maintenance and support to ensure its continued functionality and security.

* The following are some other components of the scope of an online election system:

1. Increasing number of voters as individuals will find it easier and more convenient to vote.
2. Less effort and less labor intensive, as the primary cost and focus primary on creating, managing, and running a secure web voting portal.
3. Vote can vote when admin adds voter in the system by their employee id and send credentials by the email on registered email. So, the user identification is much more secure.
4. There is no need of labor for counting vote. Because the system also provide report generating feature which provide detailed report of election.

* **Scop for future enhancement:**

LostVote can be made more secure by using the following methods.

* Password Changing
* Authentication by Fingerprints

Chapter 2:

Literature Review

**2.1 HTML**

HTML is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables.

HTML, or Hyper Text Markup Language, is the standard language used for creating web pages and web applications. HTML is a markup language, meaning it uses markup tags to define the structure and content of a web page.

HTML documents are made up of various HTML tags, which are enclosed in angle brackets (< >). HTML tags can be used to define headings, paragraphs, images, links, tables, forms, and other elements of a web page.

An HTML document is structured using different HTML tags, with the <html> tag defining the start and end of the document. The <head> section contains metadata about the document, such as the title and character encoding. The <body> section contains the visible content of the web page, including text, images, and other elements.

HTML can be used in conjunction with other web technologies, such as CSS (Cascading Style Sheets) and JavaScript, to create dynamic and interactive web pages. CSS is used for styling and layout, while JavaScript is used for adding interactivity and functionality.

HTML is a widely used and important technology for creating websites and web applications. It is supported by all major web browsers and has a large and active developer community. It is relatively easy to learn and is a great starting point for anyone interested in web development.

* **HTML basic concepts:**

**Tags:**

These are used to define the structure and content of a web page. Some common HTML tags include <html>, <head>, <title>, <body>, <h1> to <h6>, <p>, <img>, <a>, <ul>, <ol>, <li>, and <div>.

HTML documents are made up of tags, which are used to define the structure and content of a web page. Tags are enclosed in angle brackets (< >) and come in pairs: an opening tag and a closing tag. The content of the web page is placed between the opening and closing tags.

For example, the <h1> tag is used to define a heading, and the content of the heading goes between the opening and closing tags: **Example: -** <h1>This is a heading</h1>

**Attributes:**

These are used to provide additional information about HTML elements. Attributes are specified within the opening tag of an element and include values such as "id", "class", "src", "href", "alt", and "style".

HTML tags can have attributes, which are used to provide additional information about the element. Attributes are specified in the opening tag and consist of a name and a value, separated by an equal’s sign.

For example, the <img> tag is used to display an image, and the "src" attribute is used to specify the URL of the image: Example: - <img src="image.jpg" alt="Description of the image">

**Hyperlinks:**

These are used to create clickable links between web pages. Hyperlinks are created using the <a> tag and the "href" attribute. Hyperlinks are used to create clickable links between web pages. They are created using the <a> tag and the "href" attribute. The text or image that is clickable is placed between the opening and closing <a> tags.

**Example:** **-** <a href="https://www.google.com">Click here to visit google.com</a>

**Images:**

These are used to display graphics on a web page. Images are included using the <img> tag and the "src" attribute. Images can be displayed on a web page using the <img> tag. The "src" attribute is used to specify the URL of the image, and the "alt" attribute is used to provide a text description of the image for users who cannot see it.

**Example: -** <img src="image.jpg" alt="Description of the image">

**Lists:**

These are used to organize information into ordered or unordered lists. Lists are created using the <ul>, <ol>, and <li> tags. HTML offers two types of lists: ordered lists and unordered lists. Ordered lists are numbered, and unordered lists use bullet points. Lists are created using the <ol>, <ul>, and <li> tags.

**Example: -**

<ul>

<li>Item 1</li>

<li>Item 2</li>

<li>Item 3</li>

</ul>

This will create an unordered list with three items. The <li> tag is used to define each list item.

**HTML Forms:**

These are used to collect data from users, such as contact information or login credentials. HTML forms can include input fields, checkboxes, radio buttons, dropdown menus, and more.

**2.2 CSS**

CSS stand for **Cascading Style Sheets** is a language used to add style and formatting to HTML documents. CSS can be used to change the color, font, size, layout, and other visual aspects of a web page.

Cascading Style Sheets is a style sheet language used to describe the presentation and layout of HTML or XML documents. CSS is used to apply visual styles to HTML elements, such as fonts, colors, spacing, and positioning, and can be used to create complex layouts and design elements.

CSS works by selecting HTML elements and applying rules to them, which dictate how they should look and behave. CSS rules are made up of a selector, which identifies the HTML element(s) to be styled, and a declaration block, which contains one or more declarations that specify the style properties to be applied.

CSS also allows for cascading and inheritance, meaning that styles can be applied to multiple elements at once and can be overridden or inherited by other elements based on their position in the HTML document.

There are various ways to include CSS in a web page, including using an external style sheet file, embedding styles directly in the HTML document using the ‘style’ tag, or applying styles inline using the ‘style’ attribute.

* **CSS basic concepts:**

**Selectors:**

CSS selectors are used to target specific HTML elements for styling. Selectors can target elements based on their tag name, class, ID, or other attributes. For example, the following CSS rule sets the font color of all paragraphs to red:

These are used to target specific HTML elements based on their attributes, classes, IDs, or other properties.

**Example: -**

p {

color: red;

}

**Properties:**

CSS properties specify the visual styles to be applied to selected elements. Properties can be used to set values for attributes such as font size, color, background, padding, margin, and border. For example, the following CSS rule sets the background color of the body element to light gray:

**Example: -**

body {

background-color: lightgray;

}

**Values:**

CSS values specify the exact styling applied to a selected element. Values can be used to set specific measurements such as pixels, percentages, or ems. They can also be used to set colors, images, and other visual attributes. For example, the following CSS rule sets the font size of all paragraphs to 16 pixels:

**Example: -**

p {

font-size: 16px;

}

**Box model:**

The box model is the foundation of layout in CSS. It describes how every HTML element is a rectangular box with content, padding, border, and margin. The box model determines how an element is sized and positioned on the page.

This refers to the way in which CSS defines the layout and sizing of elements, including the content area, padding, border, and margin.

For example, the following CSS rule sets the padding of all div elements to 10 pixels:

**Example: -**

div {

padding: 10px;

}

**Declarations:**

These specify the style properties to be applied to the selected elements, such as color, font-size, margin, padding, and many others.

**Layout and positioning:**

CSS provides various tools for controlling the layout and positioning of elements, such as the display property, the position property, and the float property.

**Responsive design:**

CSS also allows for responsive design, which involves using media queries and other techniques to adjust the layout and styling of a web page based on the size of the viewport or device.

**Cascading and inheritance**:

CSS allows for the cascading and inheritance of styles, meaning that styles can be applied to multiple elements at once and can be overridden or inherited by other elements based on their position in the HTML document. This allows for efficient and flexible styling of web pages.

Overall, these basic concepts form the building blocks of CSS and are essential for creating visually appealing and functional web pages.

**2.3 JAVASCRIPT**

JavaScript is a programming language that is commonly used to create interactive websites and web applications. It is a client-side scripting language, which means that it is executed on the user's web browser rather than on the web server.

JavaScript is a high-level language, which means that it is designed to be easy to read and write for humans. It is also a dynamically typed language, which means that variable types are determined at runtime rather than at compile time.

* **JAVASCRIPT basic concepts:**

**Browser Support:**

JavaScript is supported by all major web browsers such as Chrome, Firefox, Safari, Edge, and Opera. JavaScript is also supported on mobile devices and is used to develop mobile applications.

**Frameworks and Libraries:**

There are several popular JavaScript frameworks and libraries such as React, Angular, and Vue.js that are used to develop complex web applications. These frameworks and libraries provide developers with tools and features to streamline the development process.

**Arrays:**

Arrays are used for storing multiple values in a single variable. They can be used to store strings, numbers, booleans, and other data types.

**Variables:**

In JavaScript, variables are used to store values. You can declare variables using the keywords "var", "let", or "const". Variables declared with "var" are function-scoped, while variables declared with "let" or "const" are block-scoped.

**Data Types:**

JavaScript has several built-in data types including numbers, strings, booleans, arrays, objects, and null/undefined. You can use these data types to store and manipulate different kinds of data.

**Functions:**

Functions are blocks of code that can be called to perform a specific task. They can take parameters as input and return a value. In JavaScript, functions are first-class citizens, which means that they can be passed around and assigned to variables like any other data type.

**Operators:**

JavaScript has several operators including arithmetic operators [+, -, \*, /], assignment operators [=, +=, -=], comparison operators [==, !=, >, <], and logical operators [&&, ||, !].

**Control Structure:**

Control structures are used to control the flow of a program. Examples of control structures in JavaScript include if statements and switch statements.

**Loops:**

Loops are used to repeat code until a certain condition is met. In JavaScript, you can use for loops, while loops, or do/while loops to achieve this.

**Objects:**

Objects are collections of key-value pairs that can be used to represent complex data structures. In JavaScript, objects are created using curly braces ({}) and can be accessed using dot notation (object.property) or bracket notation (object["property"]).

**Events:**

Events are actions that happen in a web page, such as a user clicking a button or scrolling the page. In JavaScript, you can use event listeners to detect and respond to events such as mouse clicks or key presses.

**Error Handling:**

JavaScript provides error handling mechanisms to help developers identify and fix errors in their code. These mechanisms include try/catch blocks, throw statements, and the console object.

**2.4 JSP (Java Server Pages)**

JSP stands for **Java Server Pages.** It is a technology used for creating dynamic web pages that are based on the Java programming language. JSP allows developers to embed Java code directly into the HTML code of a web page, which can then be executed on a web server before the page is sent to a client's web browser.

JSP pages are compiled into Java servlets at runtime by the web server, which means that they can be executed quickly and efficiently. JSP pages can include HTML, XML, and Java code, and can be used to access databases, perform calculations, and generate dynamic content.

JSP is commonly used for building web applications and websites that require dynamic content, such as e-commerce sites, online banking portals, and social media platforms.

* **JSP basic concepts:**

**JSP Syntax:**

JSP syntax is based on HTML and can be easily integrated with HTML code. JSP code is enclosed within **<% and %>** tags. JSP code can include Java code, expressions, and declarations.

**Scriptlets:**

Scriptlets are blocks of Java code that are enclosed within **<% and %>** tags. They are used to perform tasks such as database access, calculations, and other operations.

**Expressions:**

Expressions are used to evaluate and display values on a JSP page. They are enclosed within **<%= and %>** tags and can be used to display text, variables, and the results of calculations.

**Directives:**

JSP directives are used to specify configuration settings for a JSP page. The most used directive is the page directive, which is used to specify the language, content type, and other attributes of a JSP page.

**JSP Actions:**

JSP actions are special tags that are used to perform specific tasks on a JSP page. Examples of JSP actions include the include action, which is used to include the content of another file in a JSP page, and the forward action, which is used to forward the request and responses to other resources like jsp file, html file, pdf file etc.

**Implicit Objects:**

JSP provides several implicit objects that can be used to access information about the request, response, session, and application. Examples of implicit objects include **page, exception, request, response, session, and application.**

**Custom Tages:**

JSP custom tags are user-defined tags that can be used to encapsulate reusable functionality in a JSP page. Custom tags are defined in a separate file and can be easily included in a JSP page using the **<%@ taglib %>** directive.

**Declarations:**

Declarations are used to define variables and methods that can be accessed throughout a JSP page. They are enclosed within **<%! and %>** tags.

**JSP Custom Functions:**

JSP custom functions, also known as custom tags or user-defined tags, are a way of encapsulating reusable functionality in a JSP page. Custom functions are defined using the **<@%taglib %>** directive, which specifies the location of the tag library and the prefix that will be used to reference the tags. Custom functions can be used to perform a wide range of tasks, including generating dynamic content, manipulating data, and interacting with databases.

**JSTL Libraries:**

JSTL stands for **Java Server Pages Standard Tag Library**, which is a collection of custom tags like core tags, sql tags, xml tags, function tags, formatting tags etc. that provide a standard way of accessing and manipulating data within JSP pages.

**2.5 MySQL**

MySQL is an open-source **Relational Database Management System (RDBMS)** that is widely used for building and managing databases. It is one of the most popular databases in the world and is commonly used in web applications, content management systems, and other software applications.

MySQL uses **Structured Query Language (SQL)** to communicate with the database. SQL is a standard language for managing relational databases, and MySQL follows the SQL standard. MySQL is known for its scalability, reliability, and ease of use, and it can handle large amounts of data efficiently.

It is also cross-platform compatible, which means that it can run on a variety of operating systems, including Windows, Linux, and macOS.

* **MySQL basic concepts:**

**Database:**

A database is a collection of tables that store related data. Each database is identified by a unique name and can contain one or more tables.

**Table:**

A table is a collection of related data organized into rows and columns. Each table is identified by a unique name and contains one or more columns that define the data types and properties of the data stored in the table.

**Column:**

A column is a vertical set of data in a table that defines the data type and properties of the data stored in that column. Each column is identified by a unique name and has a data type such as integer, text, or date.

**Row:**

A row is a horizontal set of data in a table that represents a single record in the table. Each row contains a set of values that correspond to the columns in the table.

**Primary Key:**

A primary key is a column or set of columns in a table that uniquely identifies each row in the table. A primary key is used to enforce data integrity and ensure that each row in the table is unique.

**Foreign Key:**

A foreign key is a column or set of columns in a table that refers to the primary key of another table. A foreign key is used to establish a relationship between two tables and enforce referential integrity.

**Query:**

A query is a request to retrieve or manipulate data in a database. Queries are written in SQL and can be used to retrieve data from one or more tables, insert, update, or delete data, and perform other operations on the data in the database.

**INSERT Query:**

An INSERT query is used to insert new data into a table in a MySQL database. The basic syntax for an INSERT query is as follows: INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

**UPDATE Query:**

An UPDATE query is used to modify existing data in a table in a MySQL database. The basic syntax for an UPDATE query is as follows: UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;

**DELETE Query:**

A DELETE query is used to remove one or more rows from a table in a MySQL database. The basic syntax for a DELETE query is as follows: DELETE FROM table\_name WHERE condition;

**SELECT Query:**

A SELECT query is used to retrieve data from one or more tables in a MySQL database. The basic syntax for a SELECT query is as follows: SELECT column1, column2, ... FROM table\_name WHERE condition;

Chapter 3:

Project Management

**3.1 Project Planning and scheduling**

**3.1.1 Project Development Approach**

Developing a voting system requires a well-defined project development approach that covers all stages of the software development life cycle. Here is a general approach that can be followed:

**Define The Project Scope:**

The first step is to define the scope of the voting system. This includes identifying the features and functionalities that are required, as well as any constraints or limitations.

**Gather Requirements:**

Gather requirements from stakeholders, such as election officials, voters, and administrators. This involves understanding their needs and expectations and documenting them in a requirements specification document.

**Design The System:**

Based on the requirements, design the system architecture, user interface, and database schema. This includes identifying the technologies and tools that will be used.

**Develop The Software:**

Once the design is finalized, start developing the software using an iterative approach. This involves breaking down the development process into small, manageable tasks and completing them one by one.

**Test The System:**

Test the system to ensure that it meets the requirements and functions as expected. This includes unit testing, integration testing, and system testing.

**Deploy The System:**

Once the testing is complete, deploy the system on the production environment. This includes configuring the hardware, software, and network components.

**Maintain and Support the System:**

Finally, maintain and support the system after deployment. This includes fixing bugs, adding new features, and providing technical support to users.

Thus, throughout the project development approach, it is important to follow best practices for software development, such as using version control, documenting the code, and ensuring that the system is secure and scalable. It is also important to involve stakeholders throughout the development process to ensure that the final product meets their needs and expectations.

**3.1.2 Project Plan**

**Planning:**

Planning with your client will help achieve site goals and allows for efficient use of time and development resources. Ask your client to define what customers want in the clearest terms possible. Ask questions like:

**1.** What type of website is required?

**2.** Who is the target audience?

**3.** How many web pages are required?

**4.** How often will the content be updated?

**5.** Are there any references worth checking out?

**6.** What’s the web development budget?

**Team discussion:**

The next phase involves discussing your client’s website specification with your project team. This phase is important as it will help you translate the client requirements into a fool-proof website project plan.

Address the following questions:

**1.** What needs to be done for the client?

**2.** How does the project break down into tasks?

**3.** Which team member will perform which task?

4. How much of the client’s budget will the web development project require?

5. Where are the dependencies in the project?

**Branding, infrastructure, and layout:**

A good branding, infrastructure, and layout will not only make the website look attractive but will also help the visitors understand the core message and connect with the brand. Long story short, you will be planning and setting up the foundation of your website development project.

Some popular activities involved within this phase are:

**1.** Define the website name and tagline.

**2.** Conclude logo, colour palette, fonts, and page layouts.

**3.** Buying a domain name or hosting service.

**4.** Build a sitemap to display web pages and their relations.

**Design and development:**

Here comes the most important phase of your web development project. After deciding on branding, infrastructure, layout, and content for your client, start planning, designing, and developing the website pages. The following are some important activities that will happen during this phase:

**1.** Web pages designing based on layout styles and content.

**2.** Designing page elements such as buttons, CTAs, and testimonials.

**3.** HTML, CSS, and JavaScript authentication.

**4.** Developing functionalities like ecommerce store, or content management system.

**5.** Organizing web pages based on the sitemap.

**6.** Reviewing design and making changes based on client’s approval.

**Optimization:**

Website optimization after the designing and development process is important because it can help visitors feel more fortunate with their visits to your client’s website. In other words, people who come to your client’s site hoping to find the answer to a question will find a solution to their problem using different platforms.

From minifying scripts and CSS to improving cross-browser performance and enabling zip compression to optimizing images, this is the phase where your website development team will plan and bring the client’s site up to the highest grade.

**Finalization:**

Finally, maintain and support the system after deployment. This includes fixing bugs, adding new features, and providing technical support to users.

**Initial Testing:** Initial testing is the phase where your web development team will authenticate website functionality and confirm if it matches the client’s demand. Based on his requirements, certain tests may be conducted. Some of the known activities performed during this phase are:

**1.** Making sure if the functionality is working as expected.

**2.** Fixing issues that arise during testing.

**3.** Check if the website design is responsive and work fine on all devices.

**4.** Improving the website loading speed.

**Go Live:** Done with testing and fixing website problems? Great! Now it’s time to make the client’s website and support systems operational. In simple words, it is time to GO LIVE.

**3.1.3 Schedule Representation**

* **Schedule for Implementation of Our (LostVote) Project In 6th Semester Is Described Below:**

**1.** As We decided to make our Project in JSP so For That We started Admin &amp; User Template Finding for Good UI from **12th December 2022.**

**2.** Website Template Finding Schedule Was from **15th December 2022.**

**3.** Then We Started to Find Some of our Competitors to Review the functionalities from **19th December 2022.**

**4.** From **29th December 2022** We started for Defining Our Admin &amp; User Pages in Excel Sheet.

**5.** From **6th January 2023** Our Dependency Finding Phase Was Started for Dependencies Like JSP Mail APIs JSP SQL Connector JAR Files JSTL (Java Standard Tag Library) Files etc.

**6.** As we are using HTML Template for Admin Panel, User Panel &amp; Website, We Started Page Conversion from HTML to JSP from **9th January 2023.**

**7.** From **10th January 2023** We Started Implementing functionalities of Our LostVote Project.

**8.** From **23rd January 2023** We Started to Work on Database and Tables.

**9.** From **15th February 2023** Testing Was on Hand.

**10.** Then Project Documentation was started at **1st March 2023 till 6 March 2023.**

**11.** The Final Submission Was Submitted on **7th March 2023.**

Chapter 4:

System Requirements Specification

**4.1 User Characteristics**

In a computer system, a user is an individual or entity that interacts with the system to perform tasks, access information, or use the system's services. A user can be a human being, such as an employee or customer, or admin. A user typically has a unique identity within the system, which may be represented by a username or an account. Users may be authorized to access certain parts of the system, such as specific files or functions, based on their permissions or roles.

An online election system for a company may have various users with different characteristics. Here are some potential users and their characteristics of LostVote:

**Employee:**

These are the people who work for the company and are eligible to vote in the company's elections. They may have varying levels of technical proficiency, so the online system should be user-friendly and easy to navigate.

**Candidates:**

These are the individuals running for positions in the company's elections. They may need to submit information about themselves, such as their personal information, qualification, and about their self, to the online system. The admin has all rights to register and remove candidates in LostVote.

**Admin:**

These are the people responsible for managing the election process, including setting up the online system, registering the voters, verifying voter eligibility, registering the candidates, circulate appropriate reminders to the voters, and counting votes. They have access to administrative tools within the system to perform these tasks efficiently and accurately.

Overall, the LostVote is designed to accommodate a diverse range of users with varying levels of technical proficiency and roles within the company. The system should be easy to use, secure, and provide access to relevant information and tools for each user.

**4.2 Hardware and Software Requirements**

**Hardware Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Client Side** | | **Server Side** | |
| RAM | 1 GB | RAM | 4 GB |
| HDD / SSD | 10 GB | HDD / SSD | 50 GB |
| Processer | 2.0 GHz | Processer | 2.0 GHz |
| Bandwidth | 10 Mbps | Bandwidth | 10 Mbps |

**Software Requirements:**

|  |  |
| --- | --- |
| **Client Side** | |
| Web Browser | Any latest and stable version of browser e.g., chrome, firefox, safari, opera, edge. |
| Operating System | Any latest and stable version windows or any equivalent OS. |

|  |  |
| --- | --- |
| **Server Side** | |
| Web Server | Glassfish server 4.x or above versions or you can use apache tomcat 8.x or above versions. |
| Web Browser | Any latest and stable version of browser e.g., chrome, firefox, safari, opera, edge. |
| Server-Side Language | Java 8 SDK or above versions. |
| IDE | Any latest and stable version of java supported IDEs e.g., Apache NetBeans, Eclipse, IntelliJ IDEA. |
| Database Server | Any latest and stable version of MySQL database. |
| Operating System | Any latest and stable version windows or any equivalent OS. |

Chapter 5:

System Analysis

**5.1 Feasibility Study**

Feasibility study is a process of evaluating the viability of a proposed project or system by assessing its technical, economic, and operational feasibility. It is a comprehensive analysis of the potential benefits and risks of a project, which helps decision-makers determine whether to proceed with the project, modify its scope, or abandon it altogether.

In a feasibility study, a proposed plan or project is evaluated for its practicality. As part of a feasibility study, a project or venture is evaluated for its viability to determine whether it will be successful. As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable.

The feasibility study results can also be used to create a realistic project plan and budget. Without a feasibility study, it cannot be easy to know whether a proposed project is worth pursuing.

**5.1.1 Technical Feasibility**

An online voting system would consist of an online voting Portal, a front-end system, and a back-end system. This software is very much technically feasible. Some of the technical tools and technologies are listed below.

* HTML
* JavaScript
* CSS
* JSP (Java Server Pages)
* Bootstrap
* Java Email API
* MySQL

Most of the tools and technologies mentioned above are freely available. This system uses Java web technology with Sql which is open source and portable to any operating system. And the stability of java for future expects is increase day by day. And this portal uses MySQL for communication with database which is supported by almost all languages. So, this Secure E-voting Portal technically feasible.

**5.1.2 Economical Feasibility**

Development of this application is highly economically feasible. The only thing to be done is making an environment with an effective supervision. It is cost effective in the sense that has eliminated the setup of voting booth and paperwork completely. The system is also time effective because the voting is cast by voter at anywhere in the world. So, this Secure E-voting Portal Economically feasible.

**5.1.3 Operational Feasibility**

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance includes issues such as determining whether the system can provide the right information for the voter personal details, and whether the system can be organized so that it always delivers this information at the right place and on time using internet services. And it’s totally web based so, user don’t need to install any external application or software. So, this Secure E-voting Portal operationally feasible.

**5.2 Functions of System**

The functions of a system refer to the various tasks or operations that the system is designed to perform. The functions of a system can vary depending on its type and purpose. But generally, they can be categorized into the following types:

**Input:**

This function involves collecting data or information from various sources and entering it into the system. The input function can involve various methods such as manual data entry, scanning, or electronic transfer.

**Processing:**

This function involves manipulating and transforming the input data to produce meaningful output. Processing can involve various operations such as calculations, sorting, filtering, or searching.

**Output:**

This function involves presenting the processed data or information in a useful and meaningful format for users. Output can be in various forms such as reports, graphs, charts, or audio-visual presentations.

**Storage:**

This function involves maintaining and managing the data or information within the system. Storage can involve various operations such as data backup, retrieval, deletion, or archiving.

**Control:**

This function involves monitoring and managing the system's operations to ensure that it performs efficiently and effectively. Control can involve various activities such as error handling, security management, or performance monitoring.

Overall, the functions of a system are designed to support its purpose and objectives by providing the necessary capabilities and operations to meet the user's needs. The functions of a system may be performed by hardware, software, or a combination of both, and they are typically designed to work together in a coordinated and integrated manner.

**5.2.1 Use Case Diagram**

A use case diagram is a type of behavioural diagram in Unified Modelling Language (UML) that represents the interaction between actors (users) and a system in terms of use cases (functional requirements) of the system. The symbols used in a use case diagram include:

**Actor:**

Actors are external entities that interact with the system to achieve a specific goal. They are represented by stick figures and are placed outside the system boundary. Examples of actors include voter, customers, employee, and administrators.

A picture containing diagram

Description automatically generated

**Use Cases:**

Use cases represent the functional requirements of the system and describe the interactions between the actors and the system to achieve a specific goal. They are represented by ovals and are placed inside the system boundary. Use cases are named with a verb-noun phrase that describes the function they perform. Examples of use cases include "login," "view candidates," and "view result".

Diagram

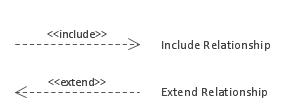
Description automatically generated

**Relationship:**

This function involves maintaining and managing the data or information within the system. Storage can involve various operations such as data backup, retrieval, deletion, or archiving. Relationships: Relationships describe the interactions between actors and use cases in terms of the functionality they provide. There are two types of relationships: "include" and "extend."

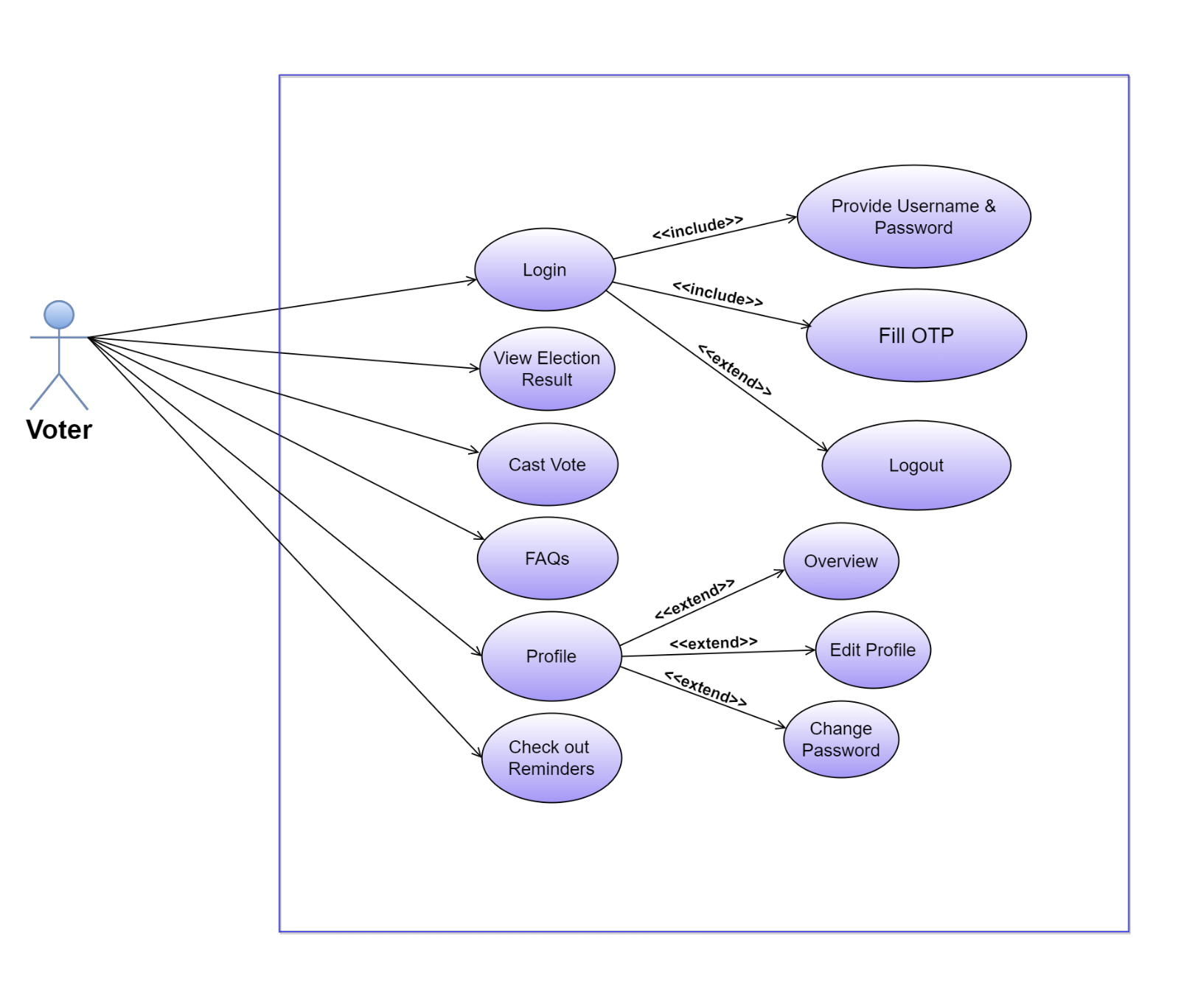
**"Include" relationship:** This represents a common functionality that is shared by multiple use cases. It is represented by a dashed arrow from the including use case to the included use case.

**"Extend" relationship:** This represents a functionality that is optional or alternative to the primary use case. It is represented by a dotted arrow from the extending use case to the extended use case.

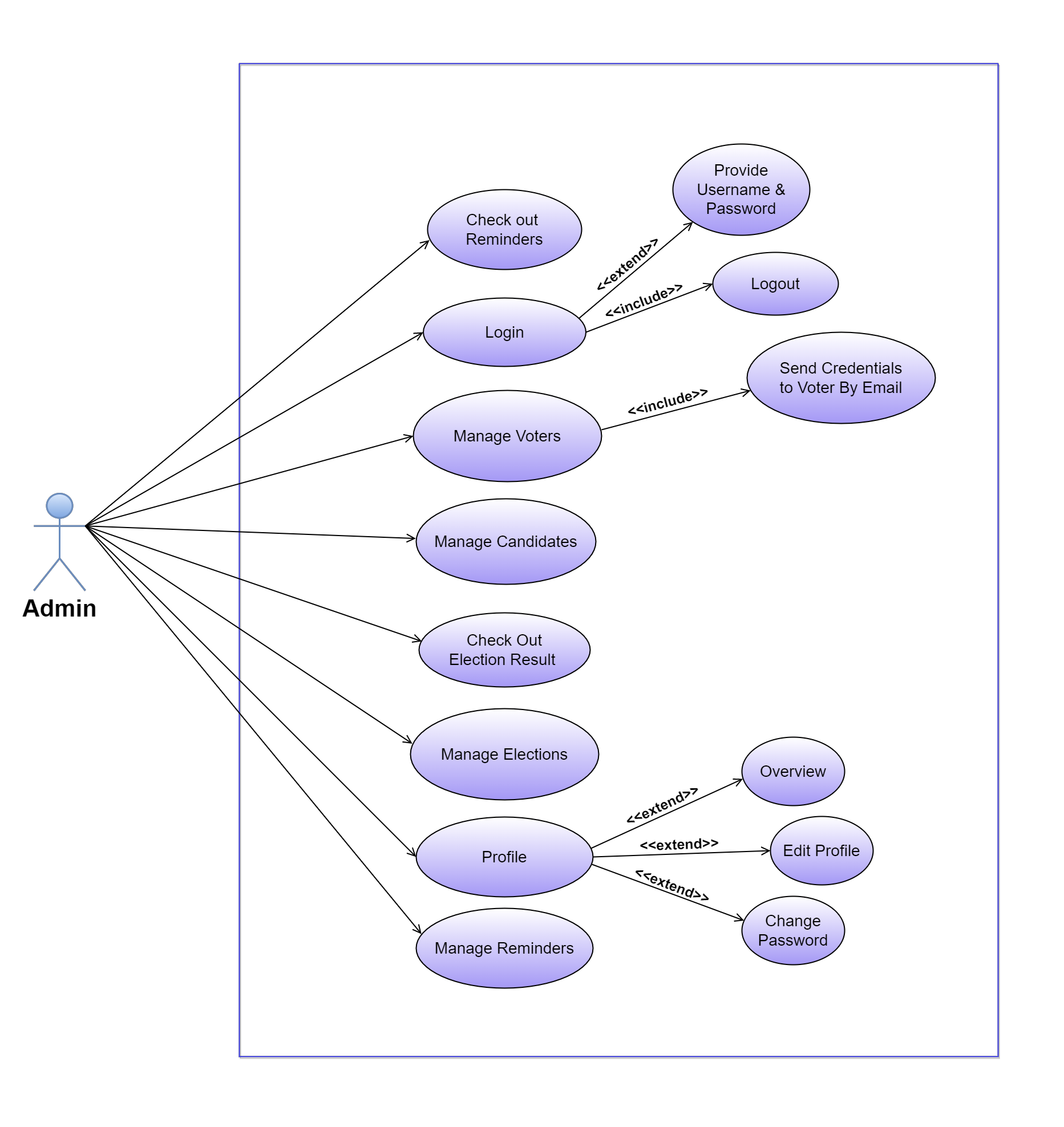


Overall, these symbols provide a graphical representation of the functional requirements of a system and help to identify the different actors and their interactions with the system. They also help to identify any shared or alternative functionalities between use cases and provide a high-level overview of the system's functionality.

**Voter’s Use case:**



**Admin’s Use case:**



**5.3 Data Modelling**

Data modelling is an important aspect of a project report that involves creating a conceptual representation of the data that is being used or generated by the project. A good data model should provide a clear understanding of the relationships between the various data elements, their attributes, and how they are stored and accessed.

Data modelling is the process of creating a conceptual representation of data and information that is used to support various business processes, applications, and systems. It involves identifying the entities, attributes, and relationships that exist within a system and developing a model that represents them. Data modelling can be divided into two main types: conceptual and logical.

**Conceptual Data Modelling:**

This type of data modelling focuses on identifying the high-level business concepts and their relationships in a system. It is used to create a common understanding between the business stakeholders and the technical team, and it helps to identify the scope and boundaries of the system.

**Logical Data Modelling:**

This type of data modelling focuses on translating the conceptual data model into a more detailed and structured representation of the data. It involves identifying the entities, attributes, relationships, and constraints that are required to support the business processes and requirements. Logical data modelling helps to ensure the accuracy, consistency, and completeness of the data and information within the system.

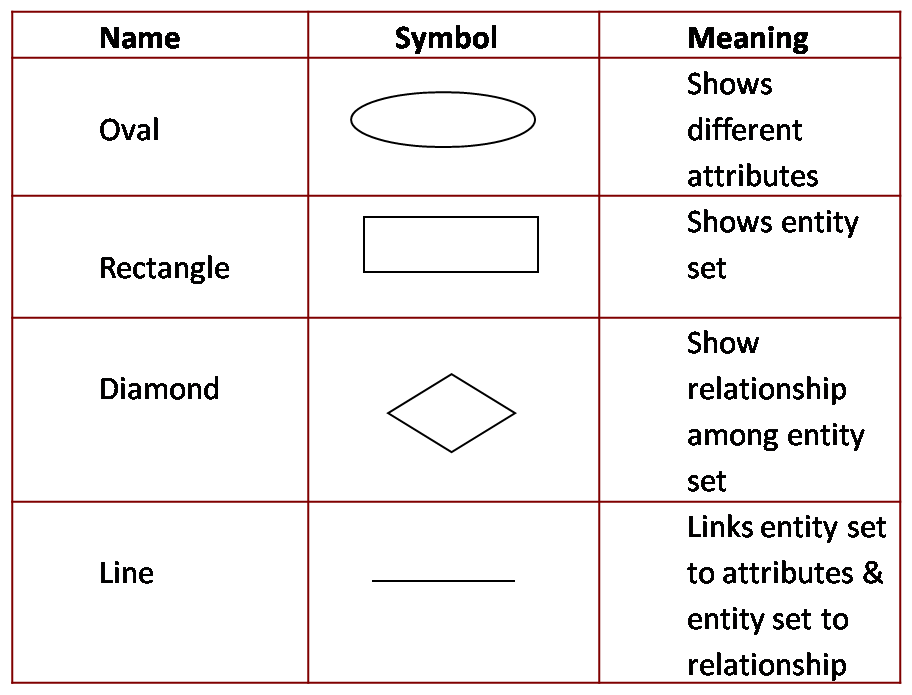
Overall, data modelling can be done using various techniques and tools, including Entity-Relationship Diagrams (ERD), Class Diagram, and Activity Diagram. The data model serves as a blueprint for designing and developing the database, and it helps to ensure that the database is structured, organized, and optimized to support the system's requirements.

**5.3.1 E-R Diagram**

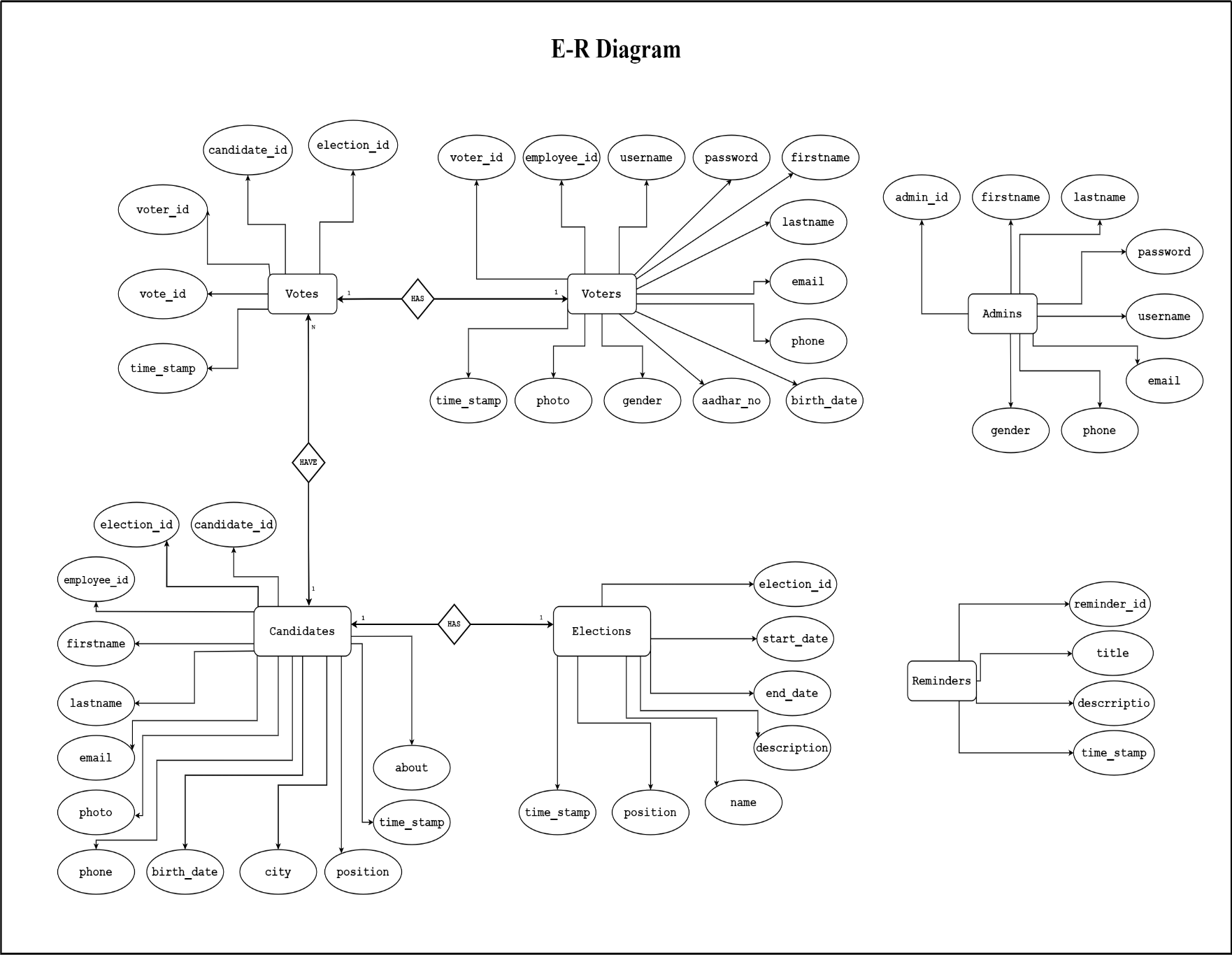
An E-R (Entity-Relationship) diagram is a type of diagram used in database design to represent the relationships between entities in a system. It shows the entities as rectangles, the attributes of the entities as ovals, and the relationships between entities as diamonds.

In an E-R diagram, an entity is a real-world object or concept that has its own set of attributes. The attributes describe the properties of the entity. For example, in a university database, the "student" entity might have attributes like "student ID," "name," "email address," and "major."

The relationships between entities in an E-R diagram are represented by special types of symbols which are describe below:



**E-R Diagram:**

****

**5.3.2 Activity Diagram**

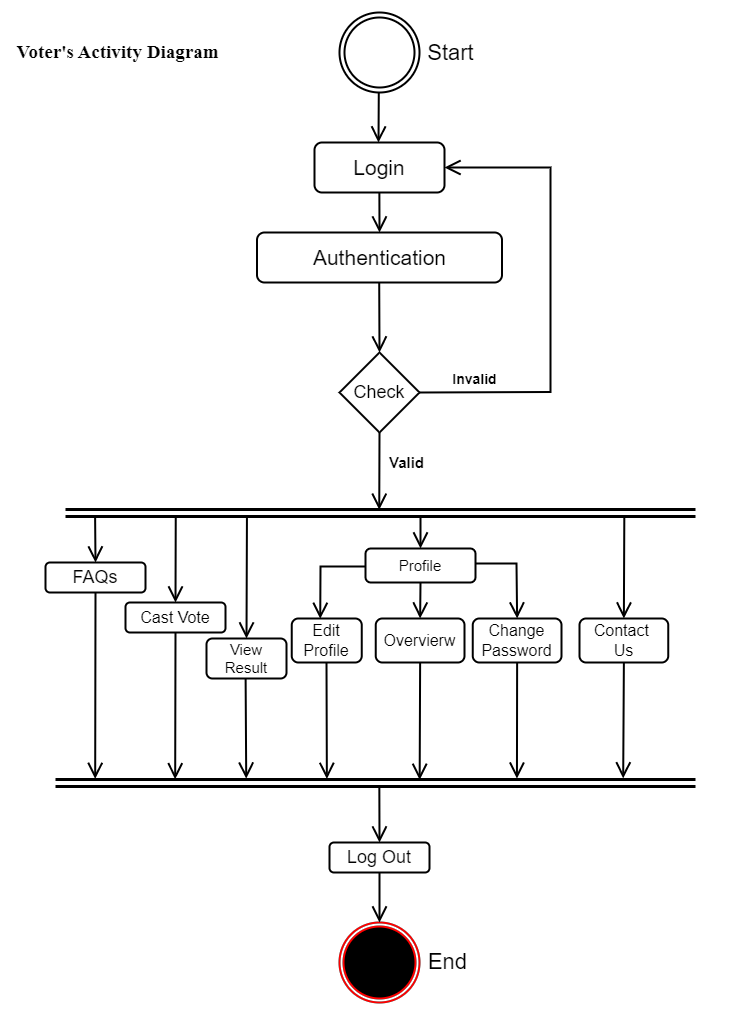
An activity diagram is a type of **UML (Unified Modelling Language)** diagram that is commonly used to model business processes or workflows. It shows the flow of activities involved in a particular system, process, or use case. Activity diagrams are represented by a series of connected shapes that represent actions, decisions, and flows between them. Each shape represents an activity or action, and arrows show the flow of control or the sequence of steps.

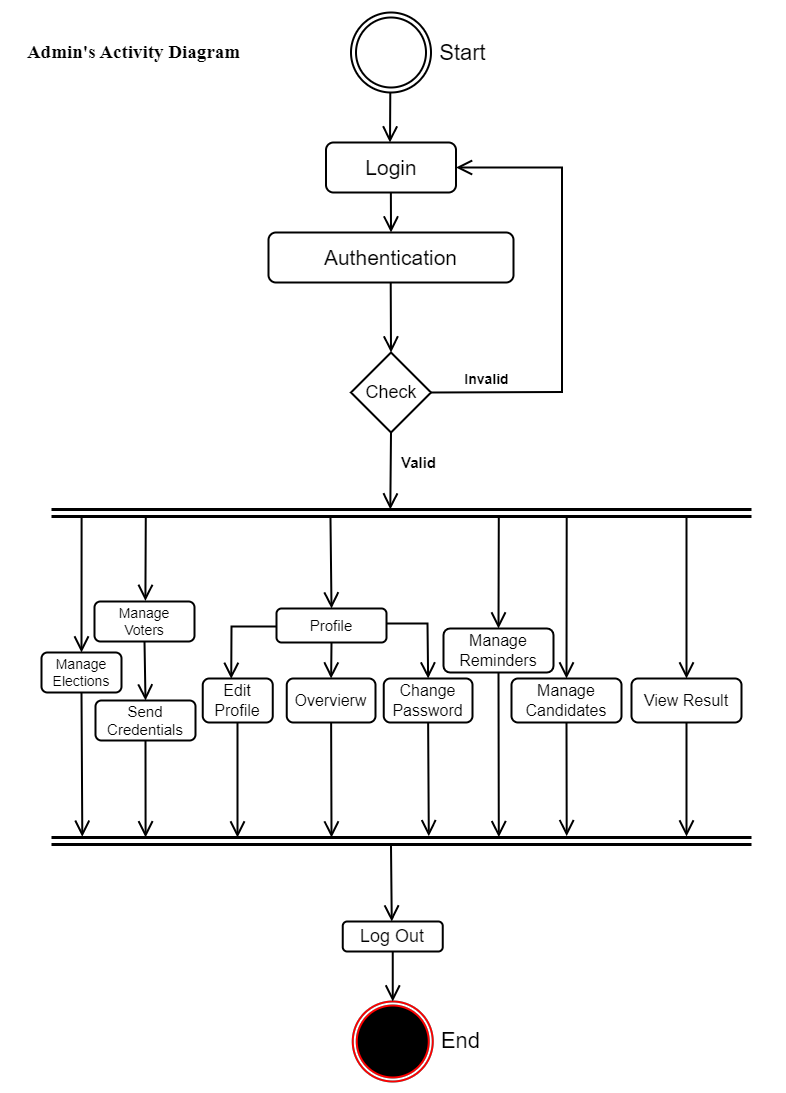
The relationships actions in an activity diagram are represented by special types of symbols which are describe below:

A picture containing shape

Description automatically generated

**Activity Diagram:**





**5.4 Functional and Behavioural Modelling**

Functional and behavioural modelling are two important techniques used in project reports to describe the characteristics and functionality of a system or product. These techniques help project teams to communicate and understand the system requirements and design more effectively.

Functional modelling refers to the process of breaking down the system into smaller components or functions, each of which represents a specific task or activity. This technique is useful for identifying the different functions of a system and their relationships with one another. Functional modelling can be achieved using various tools such as flowcharts, and data flow diagrams.

Behavioural modelling, on the other hand, focuses on the behaviour of the system under various conditions or scenarios. This technique is used to analyse how the system responds to different inputs and events, and to identify any potential issues or problems that may arise.

Overall, both functional and behavioural modelling techniques are important for project reports as they help to ensure that all project stakeholders have a common understanding of the system's requirements and design. This can help to minimize misunderstandings and reduce the risk of errors or delays during the development process. Additionally, these techniques can also help to identify areas where improvements can be made to the system design or functionality, leading to a more effective and efficient final product.

**5.4.1 Data Flow Diagram**

A data flow diagram (DFD) is a graphical representation of a system that shows how data flows through different processes and entities. It is used to model the processes, data stores, and external entities of a system and to depict the flow of data between them.

In a DFD, each process is represented by a rectangle, and the data flows between processes, data stores, and external entities are represented by arrows. Data stores are represented by parallel lines, while external entities are represented by rectangles with rounded corners.

DFDs can be used to describe a system at different levels of detail, with the highest level showing the system as a whole and the lower levels showing the details of individual processes. They can be used to identify data flow and potential data redundancies, to analyse system requirements, to document system design, and to communicate system functionality to stakeholders.

DFDs can be useful in a variety of contexts, including software development, functional and behavioural modelling, and system analysis and design. Followings are different types of symbols used in data flow diagram to define entity, flow of data, data store, process etc.

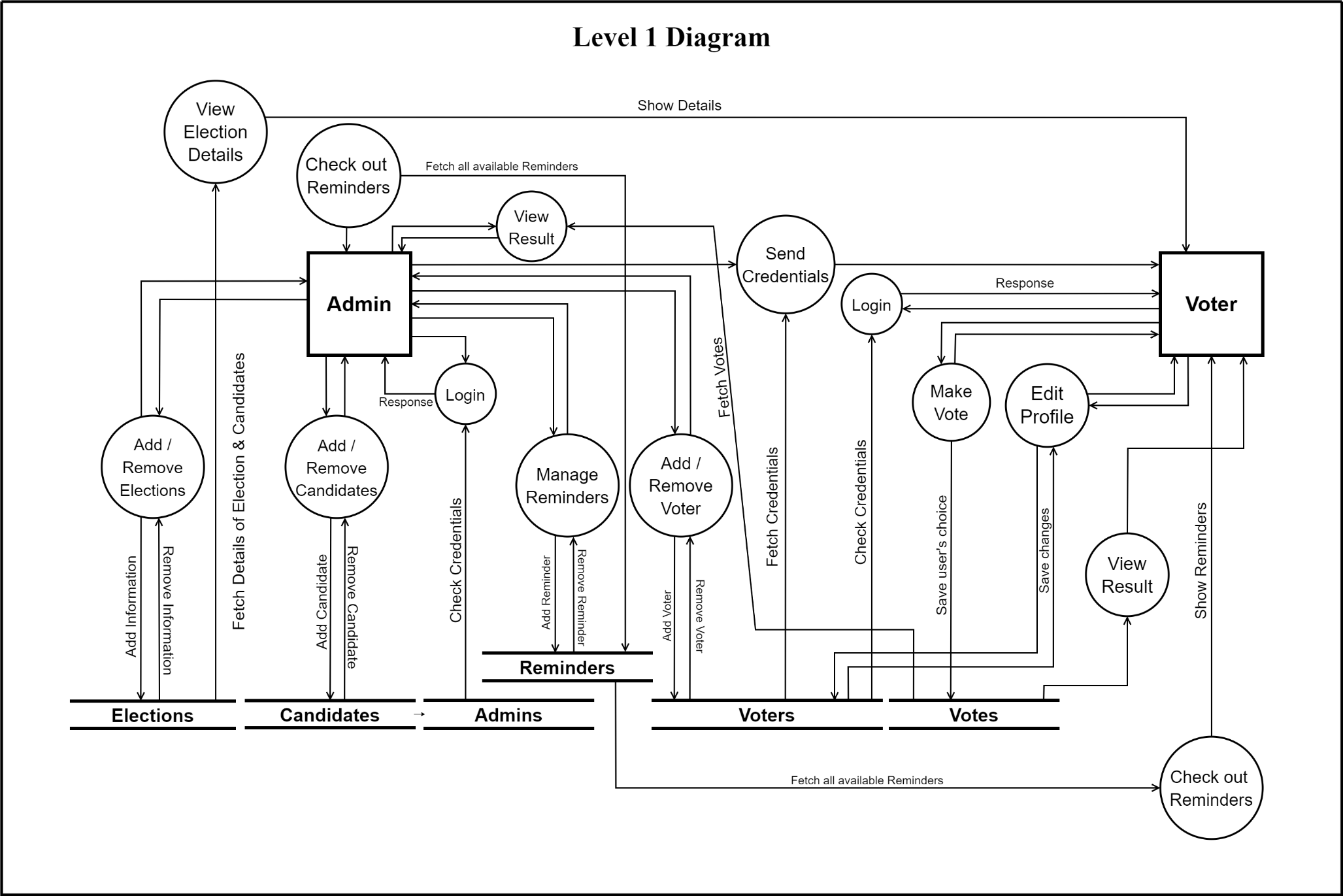
Graphical user interface, application, table

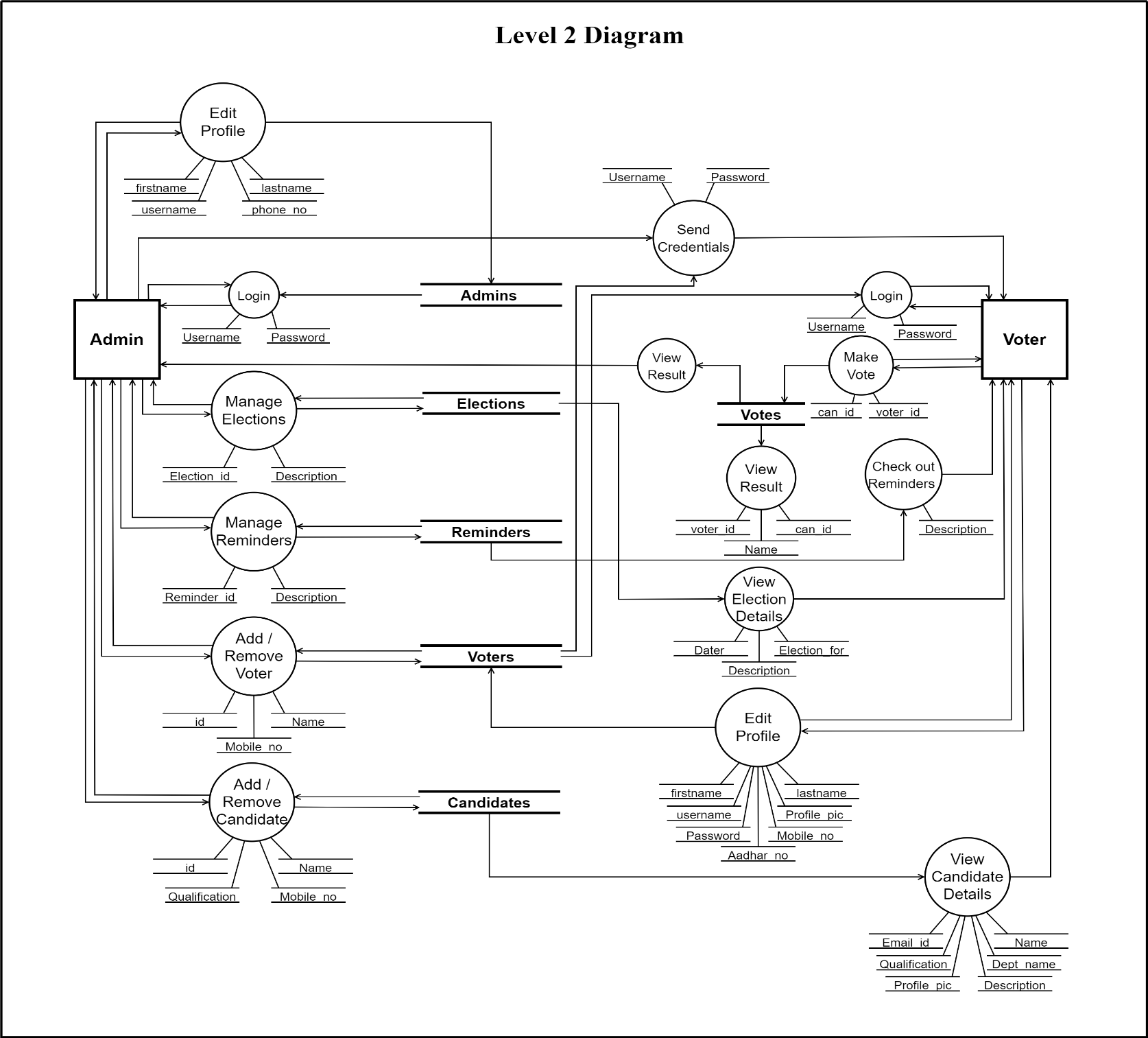
Description automatically generated

**Data Flow Diagram:**

**Graphical user interface, application

Description automatically generated**



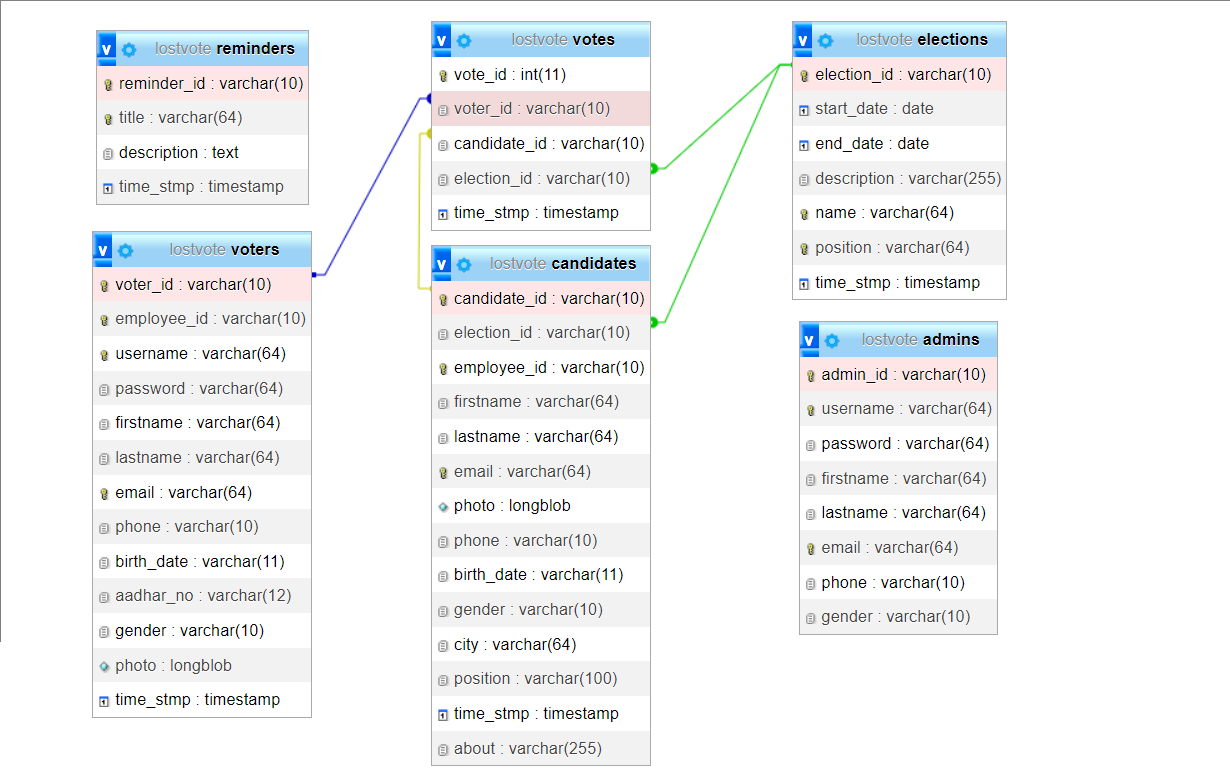


Chapter 6:

System Design

**6.1 Database Schema Design**

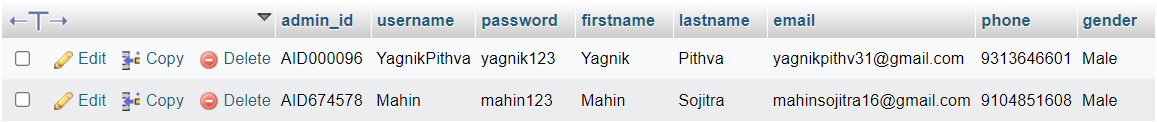
**Database Structure:**

****

**Admins Table:**

**Graphical user interface, text, application

Description automatically generated**

****

**Reminders Table:**

**Graphical user interface, text, application

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

**Elections Table:**

**Graphical user interface, text, application

Description automatically generated**



**Votes Table:**

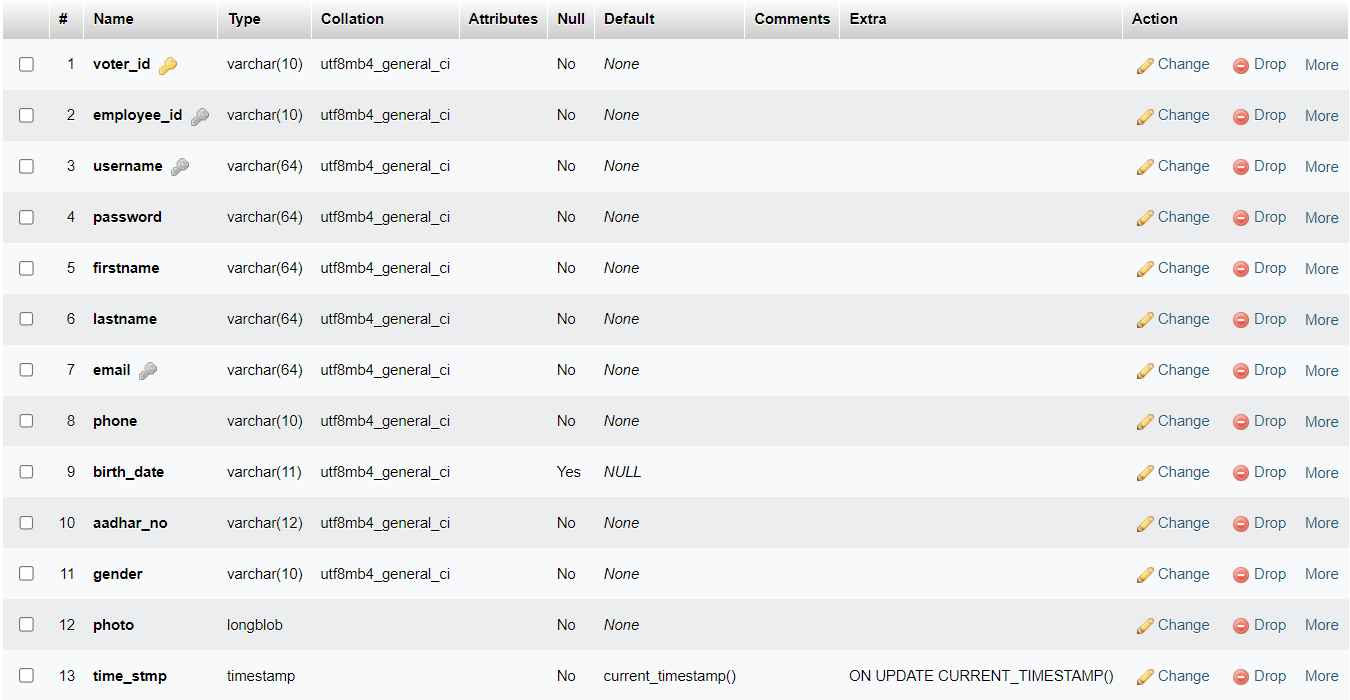
**Graphical user interface, text, application

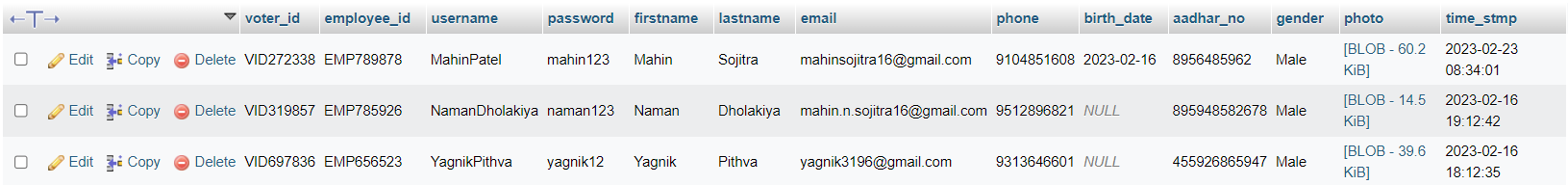
Description automatically generated**

Text

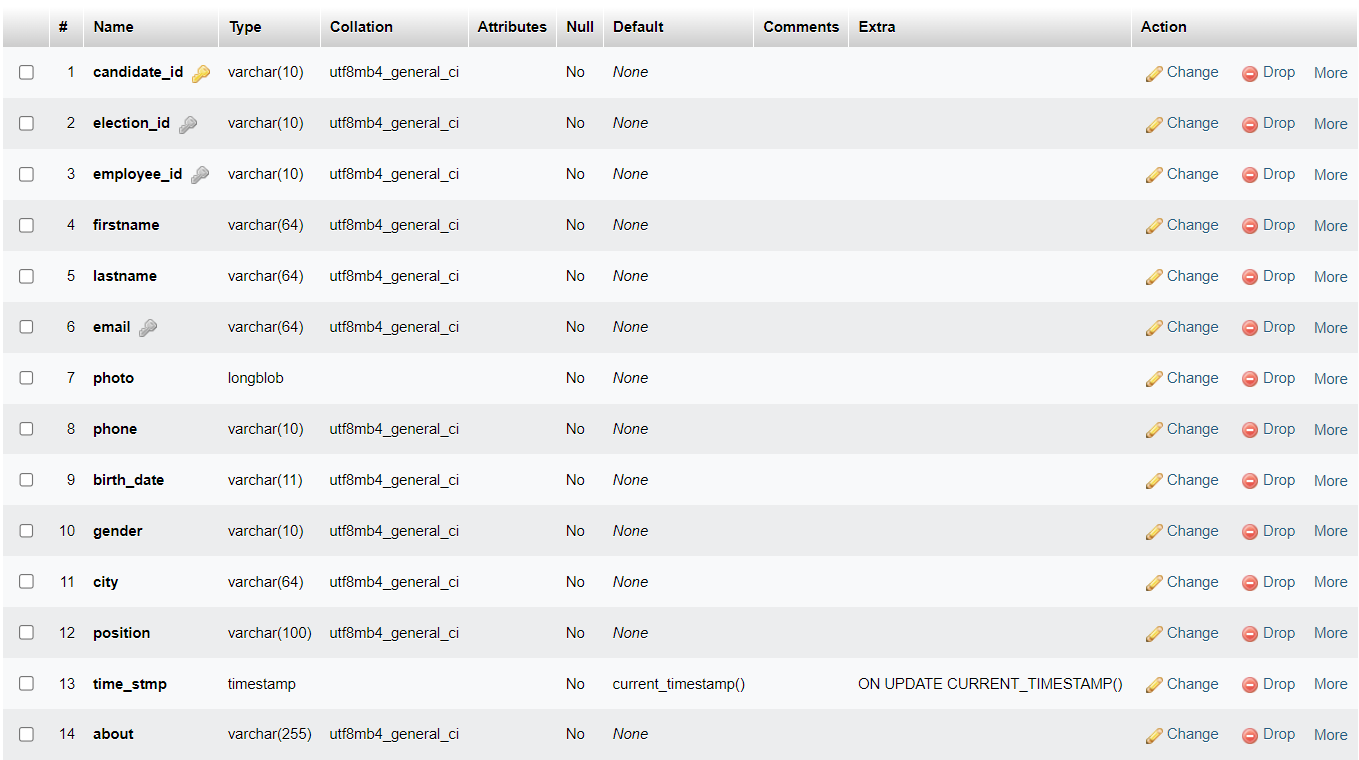
Description automatically generated with medium confidence

**Voters Table:**

****

****

**Candidates Table:**

****

**Graphical user interface, application

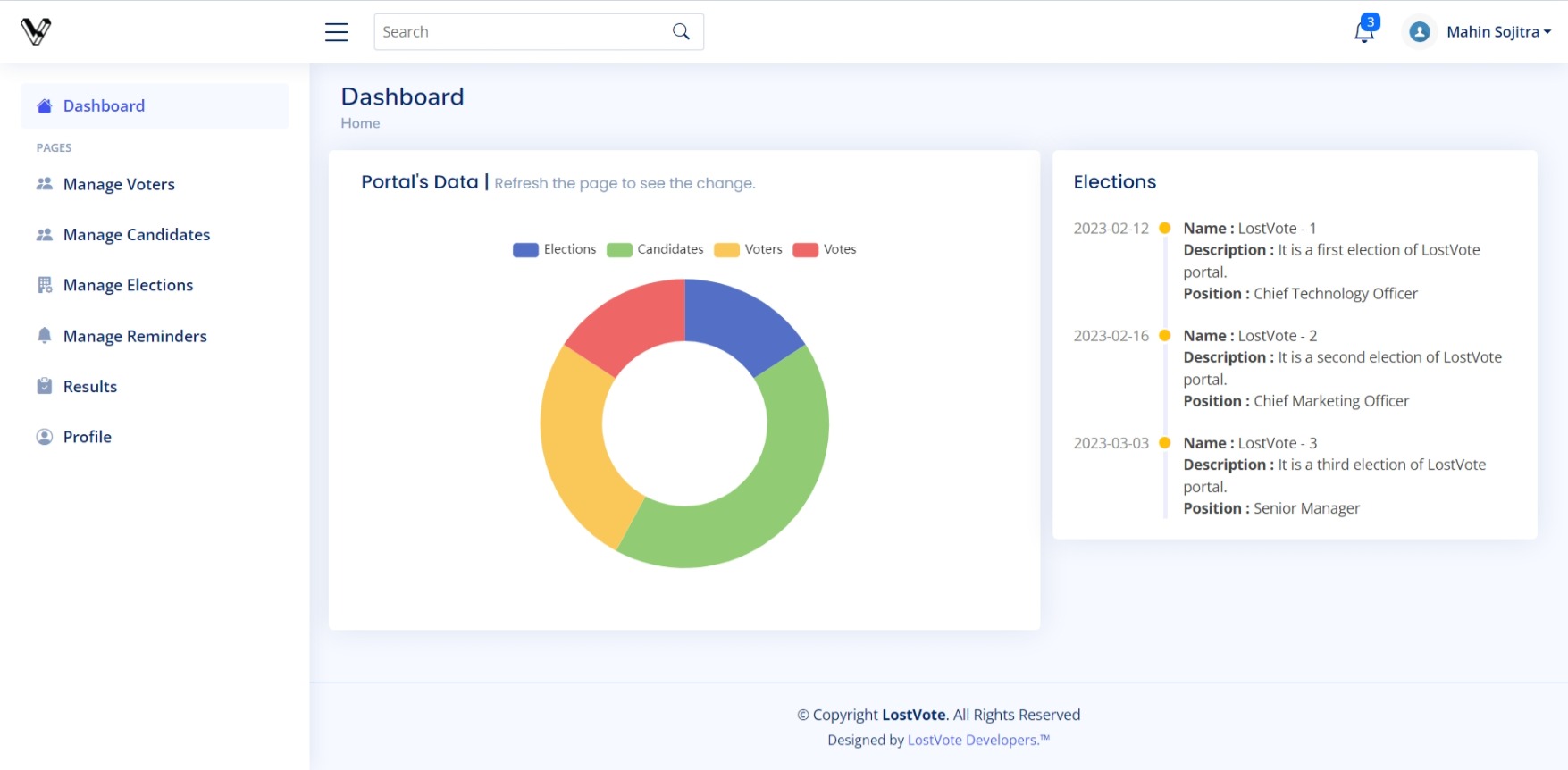
Description automatically generated**

Chapter 7:

Implementation

**Admin’s Panel:**

**Index Pages:**

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated**

**Manage Voters Page:**

**Graphical user interface, text, application

Description automatically generated**

**Manage Candidates Page:**

**Graphical user interface, table

Description automatically generated**

**Manage Elections Page:**

**Graphical user interface, text, email

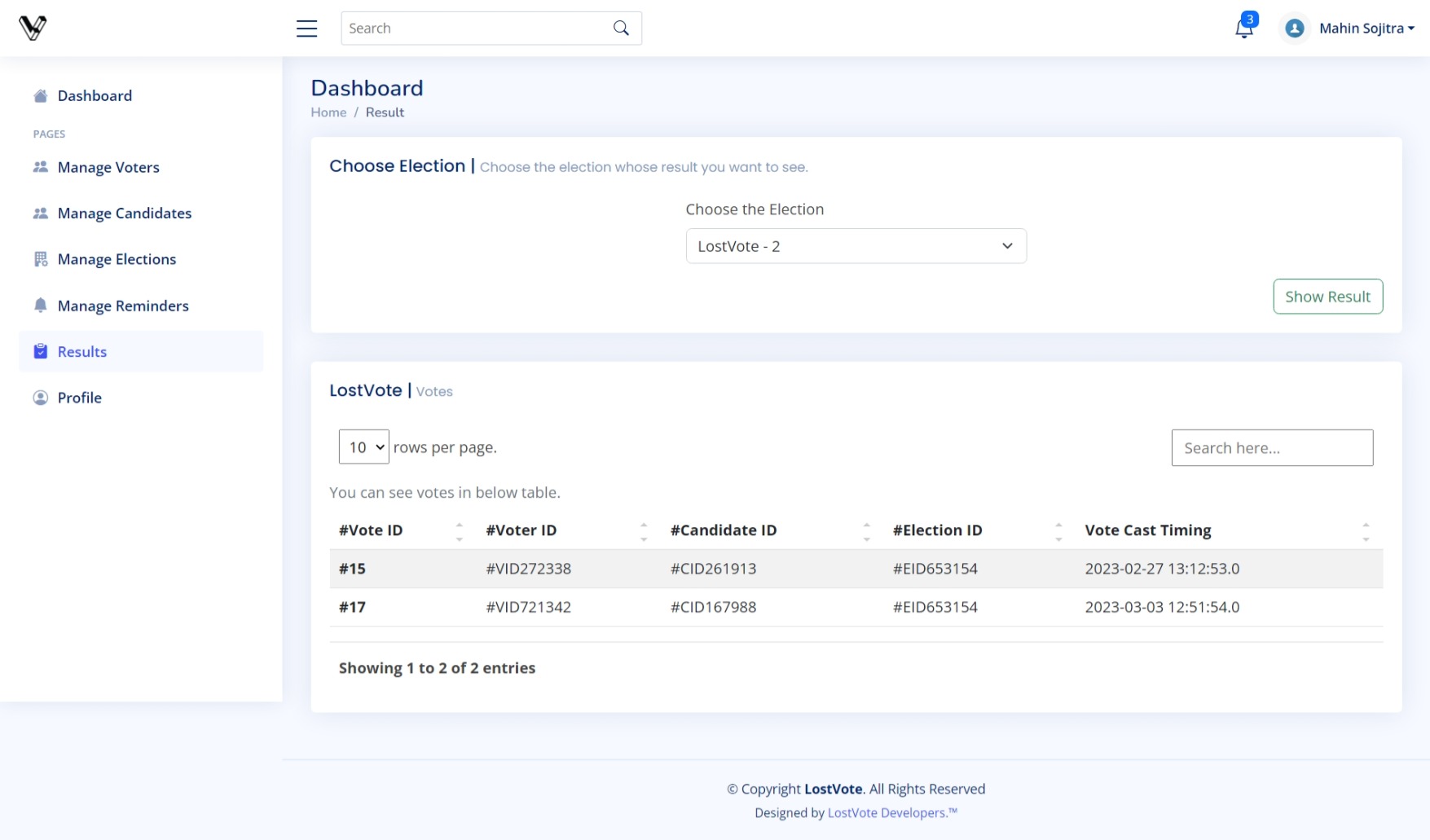
Description automatically generated**

**Manage Reminders Page:**

**Graphical user interface, text, application, email

Description automatically generated**

**Result Page:**

****

**Profile Pages:**

**Graphical user interface, text, application, email

Description automatically generated**

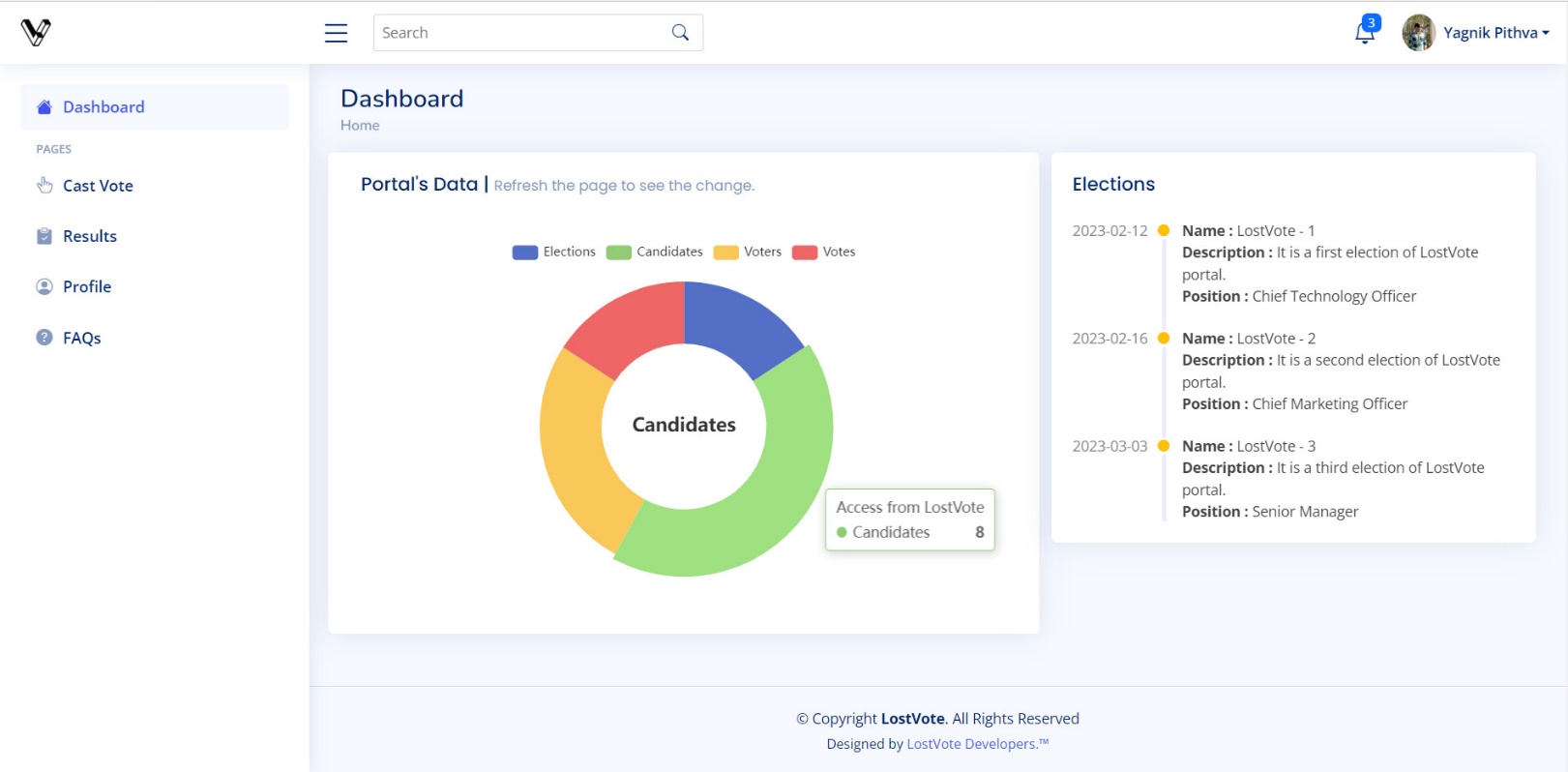
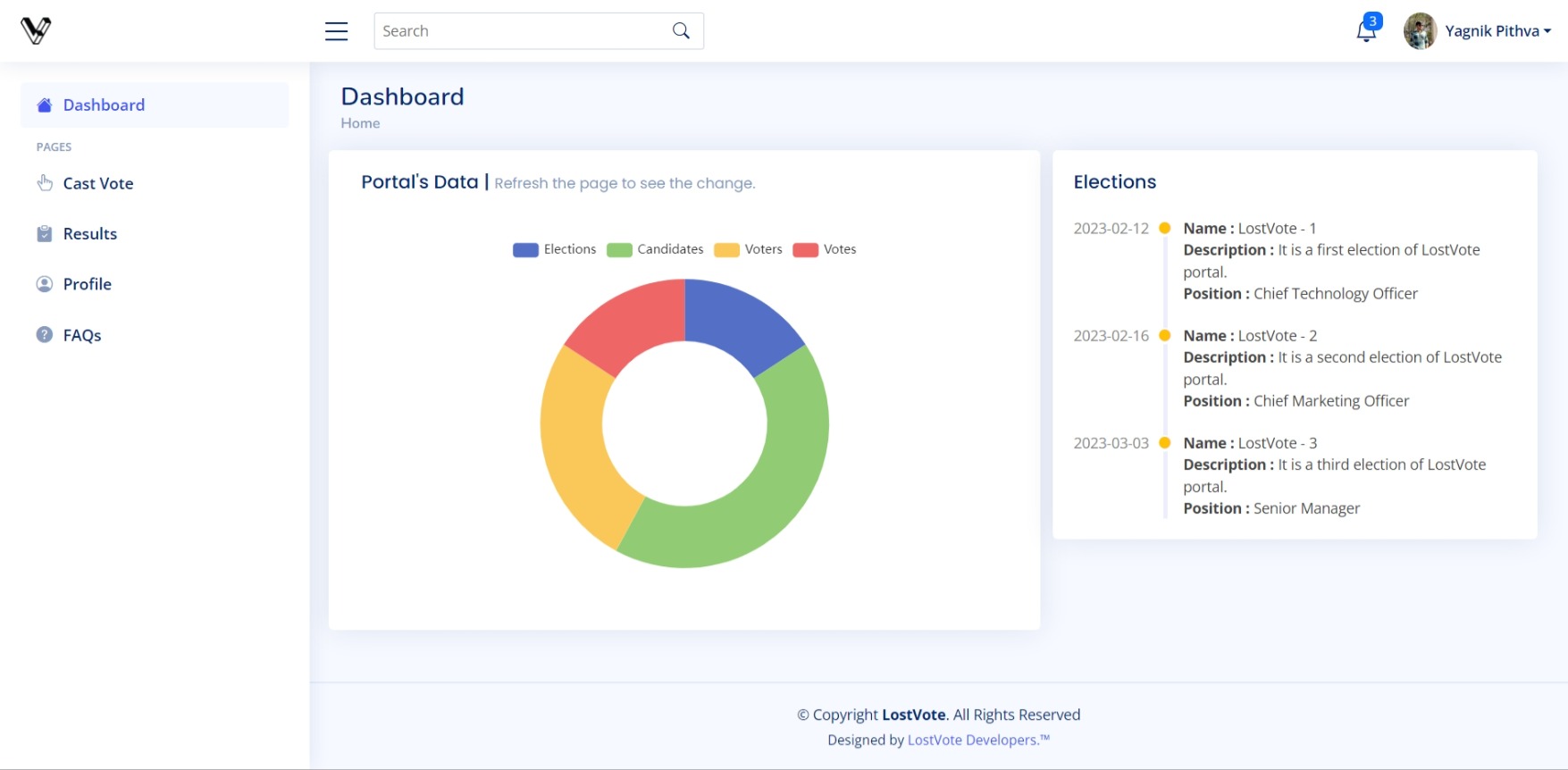
**Graphical user interface, text, application, email

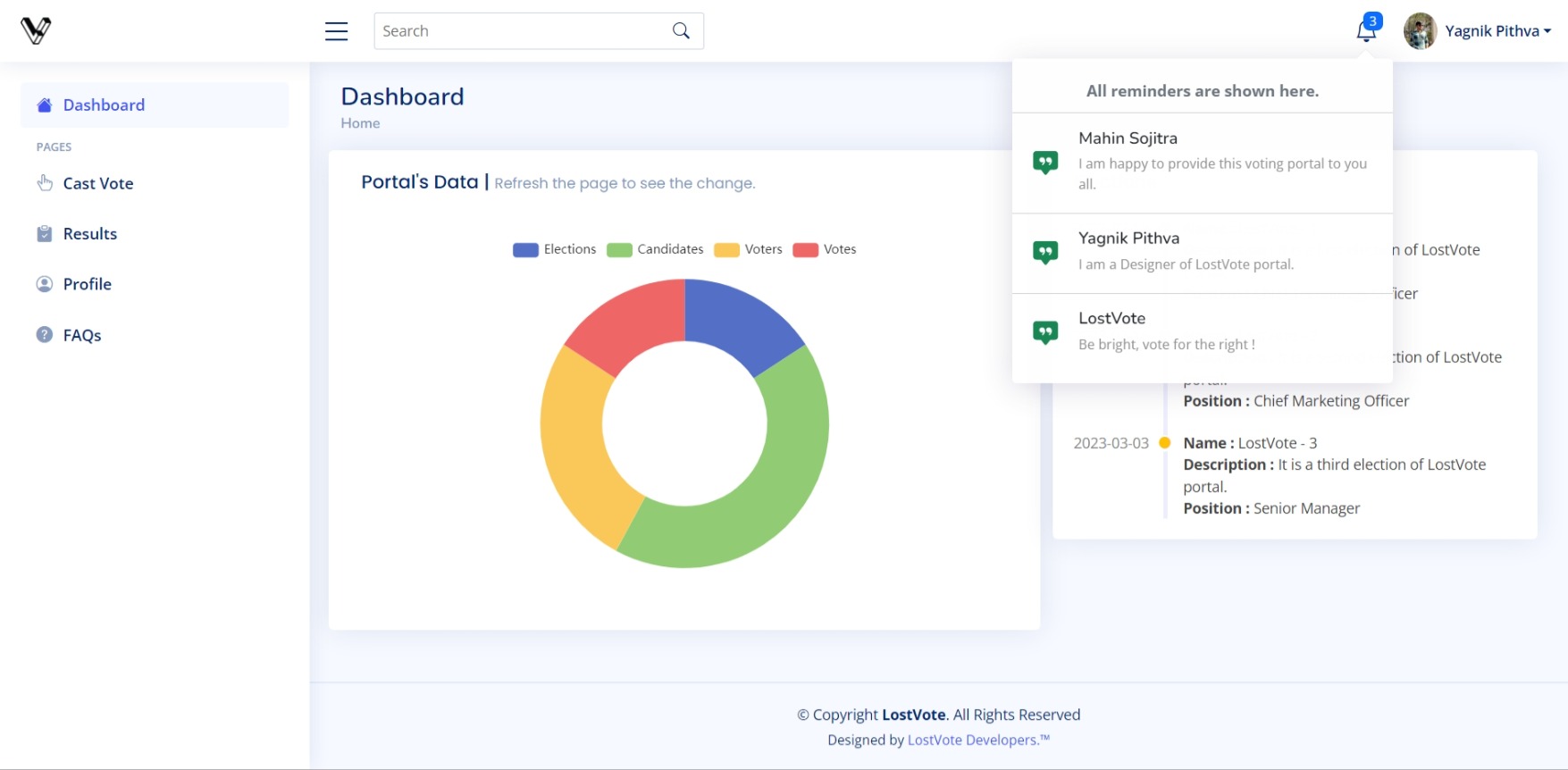
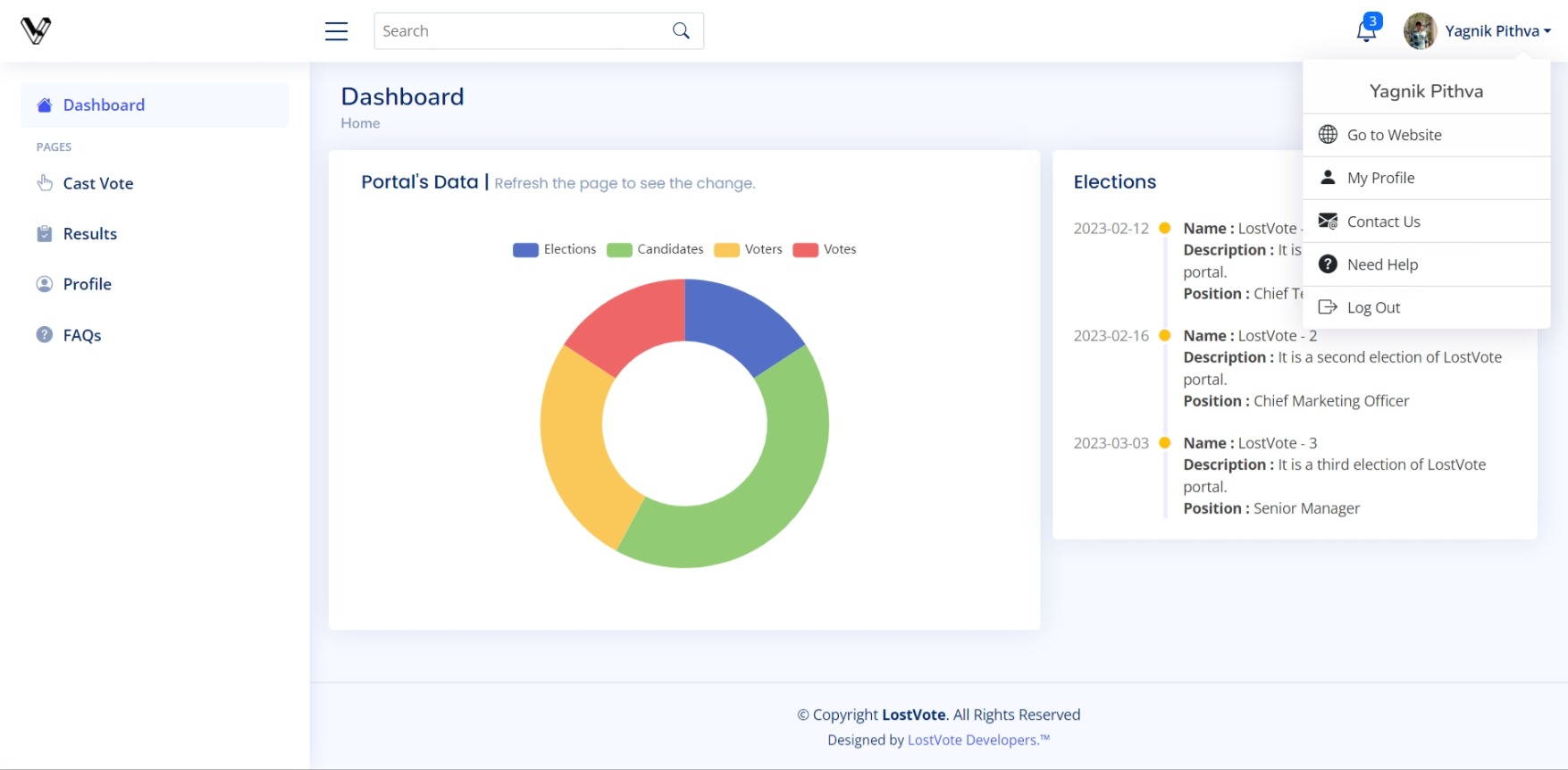
Description automatically generatedGraphical user interface, text, application, email

Description automatically generated**

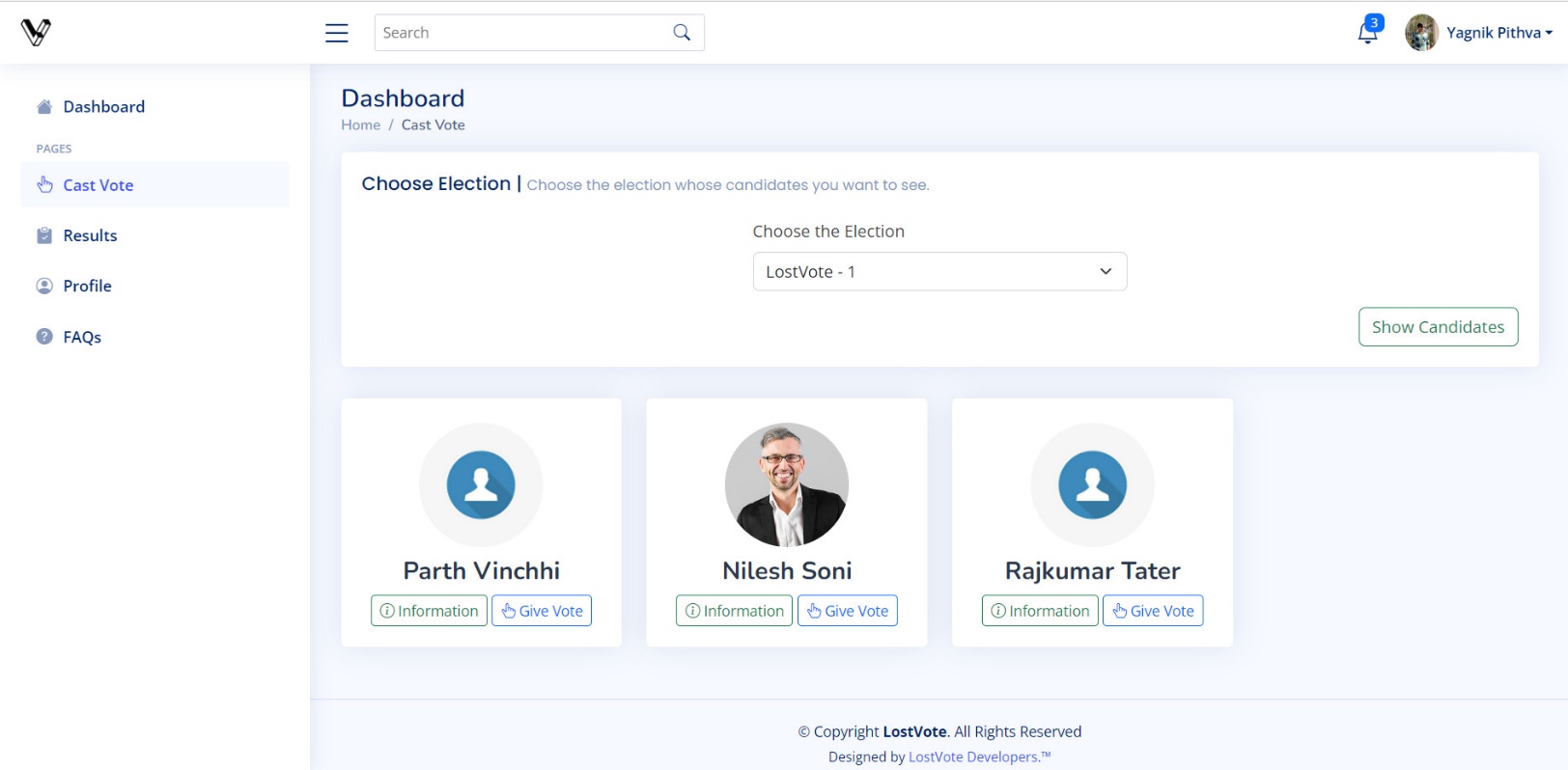
**Voter’s Panel:**

**Index Pages:**

****

****

**Cast Vote Page:**

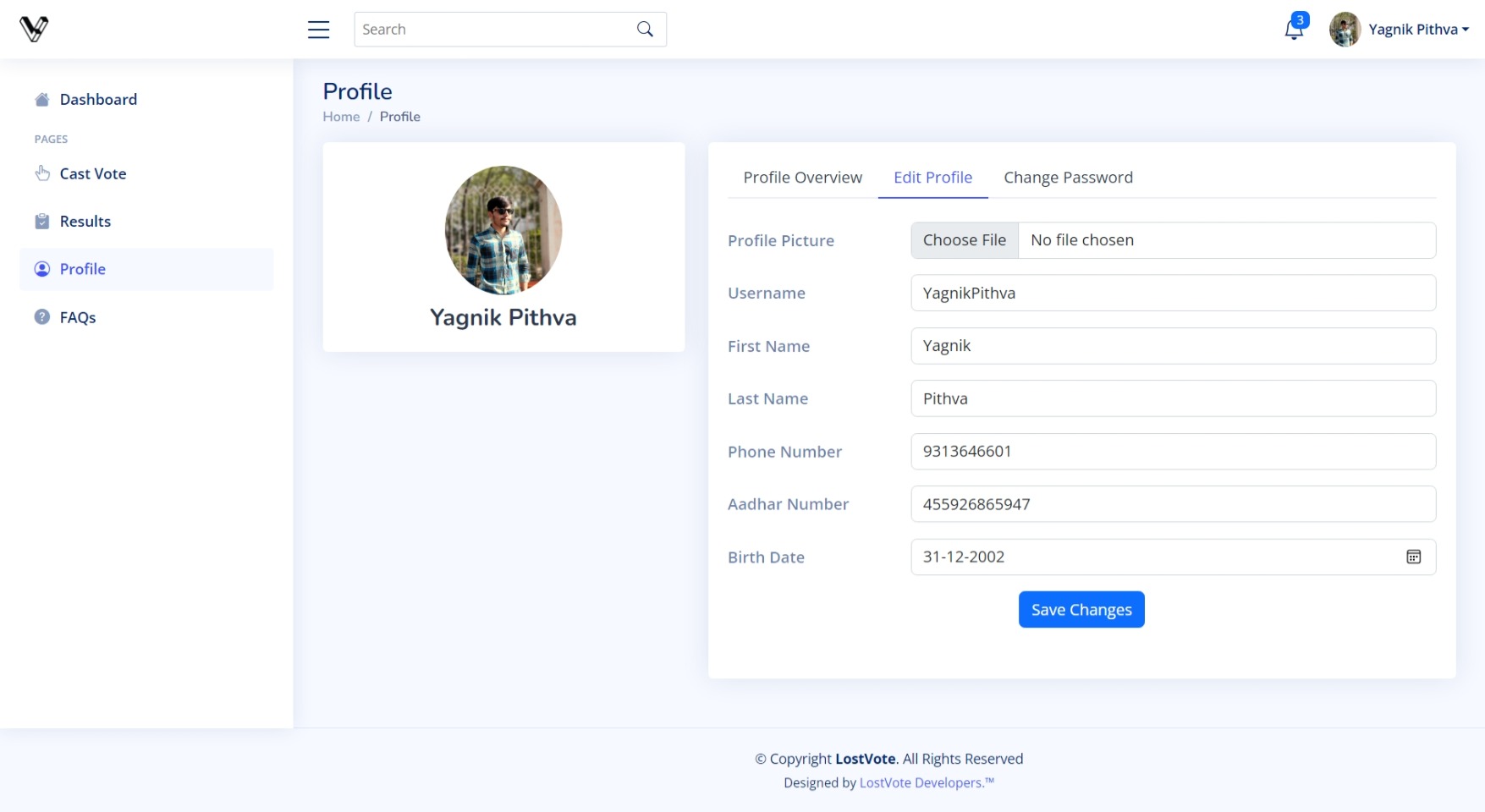
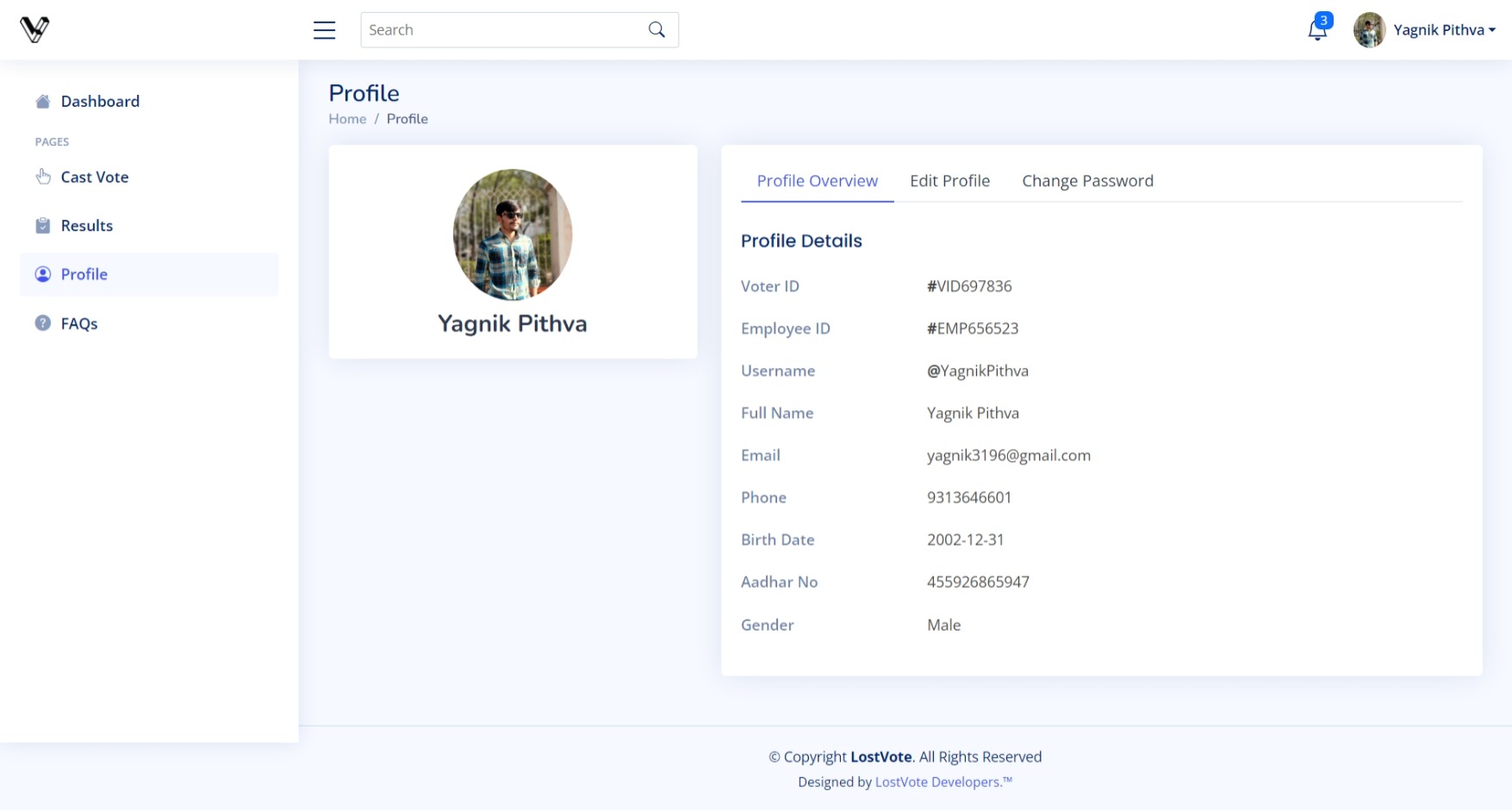
****

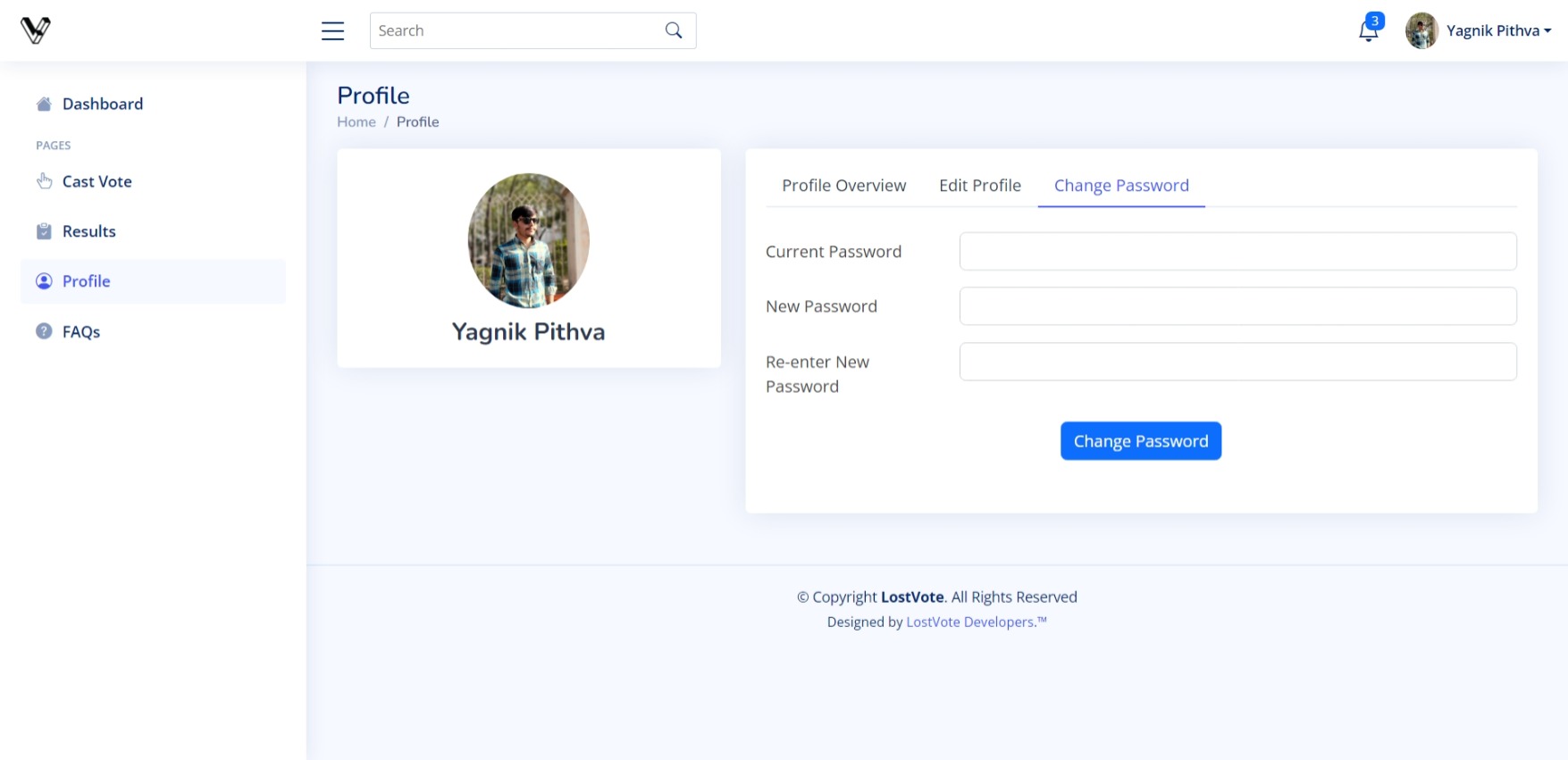
**Result Page:**

**Graphical user interface, application, Teams

Description automatically generated**

**Profile Pages:**

****

****

**Website’s Pages:**

**Index Page Without Login:**

**Graphical user interface, application, Teams

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Index Page with Login:**

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, application

Description automatically generated**

**About Us Page:**

**Graphical user interface, text, application

Description automatically generated**

**Contact Us Page:**

**Graphical user interface, application, table

Description automatically generated**

**FAQs Page:**

**Graphical user interface, application

Description automatically generated**

**Login Page for All Users of Portal:**

**Graphical user interface, application

Description automatically generated**

Chapter 8:

Conclusion

**Conclusion:**

In conclusion, the development of an election system for companies offers a reliable and efficient way to conduct internal elections and decision-making processes. This project has highlighted the importance of implementing a system that is secure, transparent, and user-friendly.

Through this project, we have discussed various security measures that can be implemented to safeguard the integrity of the election process. It is also essential to educate users on how to use the system effectively and address any concerns they may have regarding the system's security and reliability.

Overall, an election system for companies can streamline the decision-making process, increase employee engagement and participation, and enhance transparency in the organization. It is important to continually evaluate and improve the system to ensure that it meets the changing needs of the company and its employees. With proper planning and execution, an election system for companies can contribute to the success of the organization and its stakeholders.

**REFERENCES**

1. [https://demo.themefisher.com/wallet](https://demo.themefisher.com/wallet/) (Live preview of website template).
2. <https://github.com/themefisher/wallet> (Source of template for website).
3. [https://bootstrapmade.com/demo/NiceAdmin](https://bootstrapmade.com/demo/NiceAdmin/) (Source of template for admin & voter panels).
4. <https://bootstrapmade.com/nice-admin-bootstrap-admin-html-template> (Live preview of above template).
5. <https://app.diagrams.net> (For drawing diagrams of project).
6. <https://docs.oracle.com/en/java> (Reference for Java related queries).
7. <https://javaee.github.io/javaee-spec/javadocs> (Reference for Java EE 8 API).
8. <https://javaee.github.io/javamail/docs/api> (Reference for Java Mail API).
9. <https://stackoverflow.com> (For solving runtime errors and other suggestions).