Software Requirements Specification (SRS)

Election Management System

Group - 8

Adamya Sharma - AM.AI.U4AID23024

Harita Aneesh - AM.AI.U4AID23042

Santhanalakshmi - AM.AI.U4AID23057

Mounish Senisetty - AM.AI.U4AID23058

---

1. Project Summary

1.1 Project Overview

The Election Management System (EMS) is a full-stack web application designed to digitally manage the electoral process in a constituency. The system provides an integrated platform for managing voter registration, officer verification, candidate details, vote casting, and result declaration. By digitizing these critical aspects of the electoral process, the EMS aims to improve efficiency, accuracy, transparency, and security.

Built using modern technologies such as React.js (frontend), Node.js with Express (backend), PostgreSQL with Sequelize ORM (database interaction), and styled using Bootstrap and Tailwind CSS, this system is responsive, secure, and scalable. It supports multiple user roles—Admin, Officer, and Elector—each with their specific functionalities. Secure authentication is enforced using JSON Web Tokens (JWT), and access to pages and operations is restricted based on user roles. The EMS offers a real-time dashboard for each user, a mobile-friendly UI, and an interactive result display using charts.

1.2 Project Scope

The EMS covers all major aspects of a constituency-level election, including:

- A complete user authentication and authorization system (signup, login, JWT tokens, protected routes).
- Role-based dashboards:
    - Admins can manage polling stations, constituencies, officers, electors, parties, and candidates.
    - Officers can verify electors assigned to their polling station and monitor polling day progress.
    - Electors can view their profile, verify their voting eligibility, cast a vote (only once), and view live results.
- Candidate and party registration handled by the Admin.
- Vote casting interface for electors, with real-time feedback.

- Result visualization with dynamic charts (bar and pie) and vote count announcements.
- Fully responsive UI for desktop and mobile devices.
- Backend API endpoints for all CRUD operations.
- Hosting-ready configurations for deployment to platforms like Vercel (frontend) and Render or Heroku (backend).

---

2. General Description

2.1 Product Functions Overview

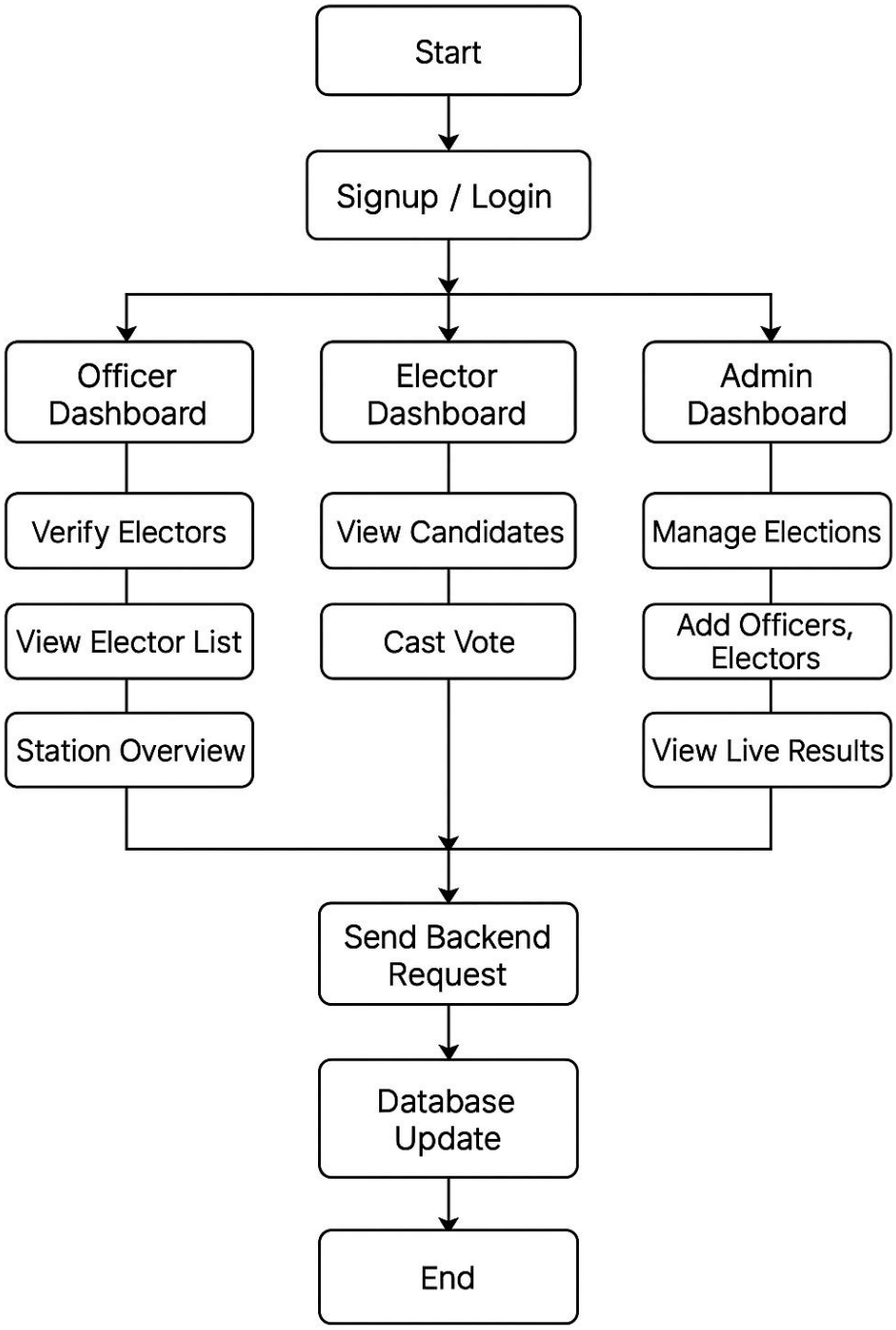
The EMS application supports a wide range of functionalities:

- **User Authentication**: Signup and login for Admin, Officer, and Elector roles. Upon login, a JWT token is issued, and routing is role-specific.
- **Admin Functions**:
    - Add, edit, and delete constituencies and polling stations.
    - Manage political parties and register candidates.
    - Assign officers to polling stations.
    - View live voting results.
- **Officer Functions**:
    - View their assigned polling station and its details (ward, area).
    - Access the list of electors assigned to their station.
    - Verify electors and update verification status.
- **Elector Functions**:
    - View personal profile (name, serial number, polling station).
    - View candidate list for their constituency.
    - Cast a vote securely.
    - Check voting status and result announcement page.

2.2 Types of Users / Actors

- **Admin**: Full access to system management features. Can add/update/delete constituencies, polling stations, officers, parties, candidates, and electors. Also has access to live voting results and winner declaration.
- **Officer**: Assigned to a specific polling station. Responsible for verifying electors, ensuring correct voter data, and monitoring voter activity on polling day.
- **Elector**: Registered voter with access to their profile, the ability to cast one vote, and view results.

2.3 Flowchart



3. Technical Requirements

Backend:

- Node.js with Express.js for building RESTful APIs
- Sequelize ORM for database modeling and interaction
- JWT-based authentication and role-based authorization middleware
- PostgreSQL as the database
- Middleware for CORS and JSON parsing

Frontend:

- React.js for building the user interface
- React Router v6 for client-side routing
- Axios for making API requests, with interceptors for attaching tokens
- React Hooks for component state and lifecycle management
- React Toastify for user notifications and error alerts
- Tailwind CSS and Bootstrap for styling and responsive layout
- Lazy loading for optimizing component rendering and speed

Other Configurations:

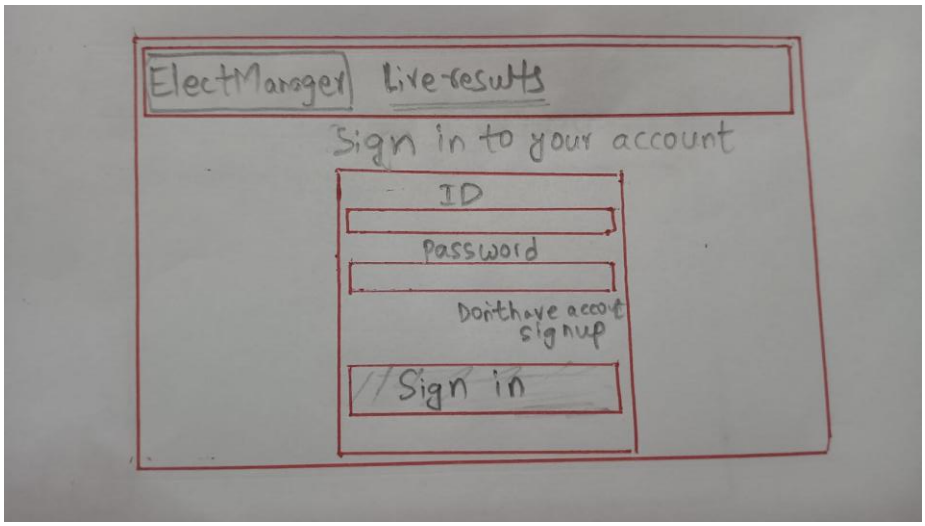
- Environment variables for securing API base URLs, JWT secrets, and DB credentials
- Backend migrations for managing database schema
- Vite for frontend build and fast development server

This technology stack supports a robust full-stack JavaScript architecture, offering scalability, performance, and maintainability.

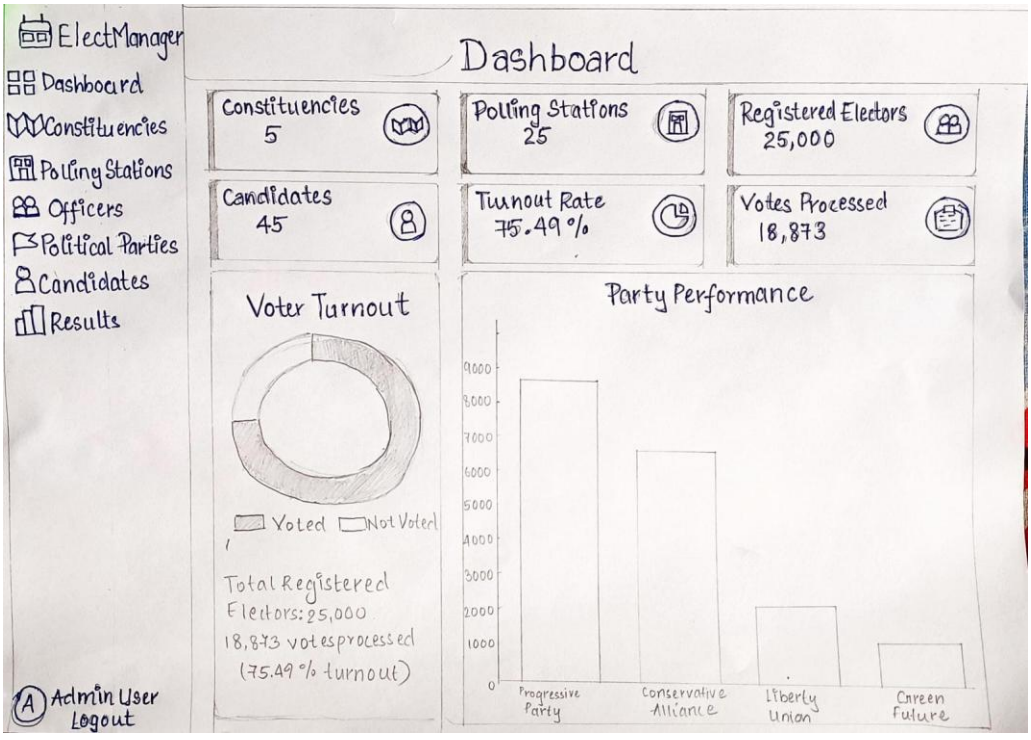
4. User Interface Design

Each page of the EMS follows a clean and modern design:

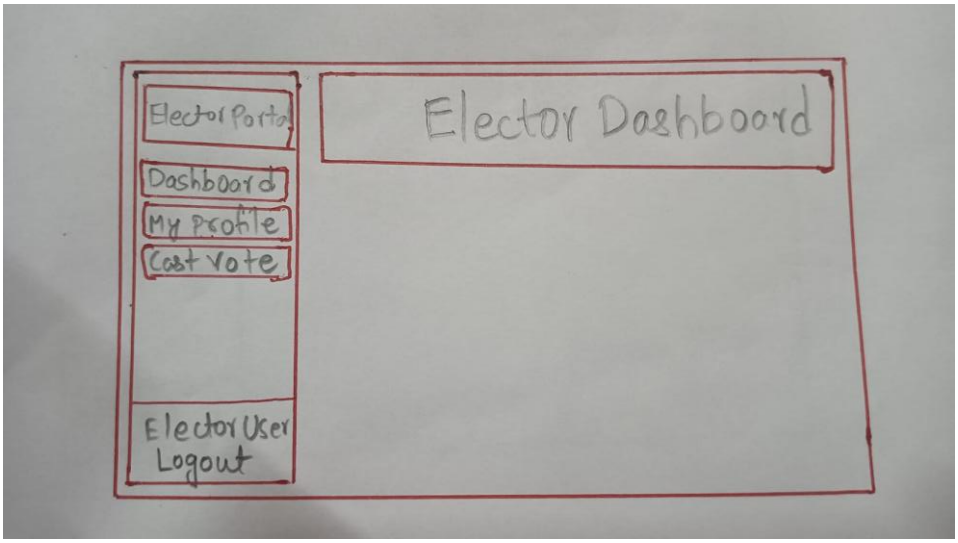
- **Signup/Login Pages**: Minimal design using Bootstrap forms. Role selection in signup. Error handling for invalid input.



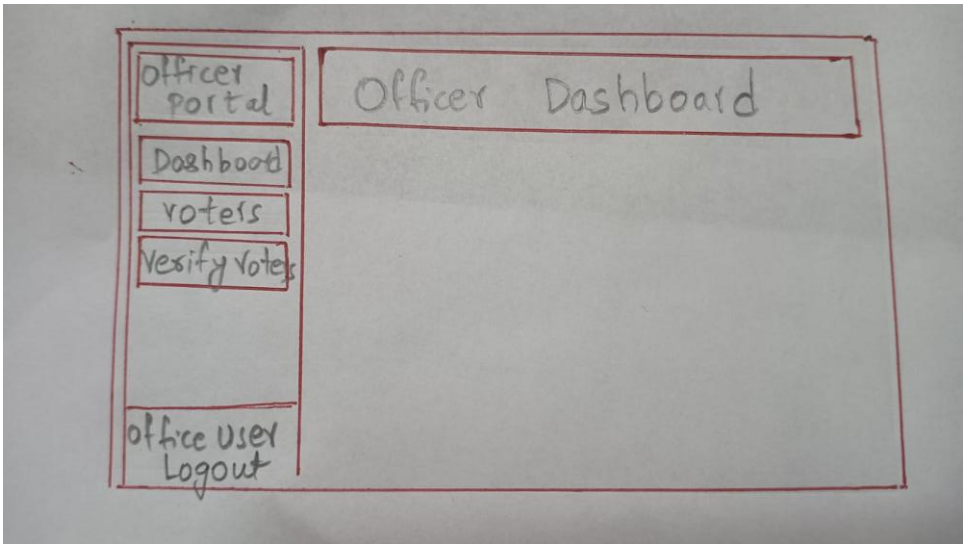
- **Admin Dashboard**: Sidebar layout with cards showing stats. Access to pages for managing entities. Uses modals for CRUD actions.



- **Elector Dashboard**: Shows profile and voting status. Link to voting page



- **Officer Dashboard**: Lists of electors by polling station. Option to mark as verified. Real-time elector status indicators.



- **Voting Page**: Grid layout of candidate cards with names, parties, and party logos. Radio buttons for selection. Submit button disabled until a candidate is selected.
- **Results Page**: Displays vote count in real time using bar and pie charts. Winner highlighted.
- **Navigation**: Role-based top navbar and side drawer for navigation. Logout functionality.
- **Mobile Optimization**: Responsive layout for all devices.

5. Team Members and Contributions

Team Member	Feature Ownership
Member 1	Officer Dashboard, Elector Verification
Member 2	Elector Dashboard, Voting Page
Member 3	Results Visualization, Profile Pages
Member 4	Admin Dashboard, Backend Integration

---

6. Database Models Overview

Entity	Attributes & Relationships
Constituency	id (PK), name
PollingStation	id (PK), name, area, ward, constituencyId (FK)
Officer	id (PK), name, role (ENUM), pollingStationId (FK)
Elector	id (PK), serialNumber (unique), name, pollingStationId (FK)
Party	id (PK), name, symbol
Candidate	id (PK), name, partyId (FK), constituencyId (FK)
Vote	id (PK), electorId (FK), candidateId (FK), pollingStationId (FK)