**Software Requirements Specification (SRS) for Voting System**

**1. Introduction**

The Voting System (VS) is a web application designed to facilitate the election process by allowing users to view a list of candidates, vote for their preferred candidate, and monitor the election results. The system is built using React with Redux Toolkit and Vite, and utilizes Axios for API handling. Ant Design (antd) is used for the user interface components.

**2. Purpose**

The purpose of the VS is to provide a user-friendly and efficient platform for conducting elections. It aims to streamline the voting process, enhance transparency, and provide real-time updates on the election results.

**3. Scope**

The VS includes the following features:

- User authentication for voters and candidates.

- Display of a list of candidates with their profiles.

- Voting interface for users to cast their votes.

- Candidate dashboard for candidates to edit their profiles, provide campaign updates, and monitor election results.

- Real-time updates on the election results for all users.

1. **Overall Description**

**4.1 Product Perspective**

The VS is a standalone application that interacts with a backend API to fetch and update data. It is designed to be scalable and maintainable, with modular components that can be easily extended or modified.

**4.2 User Classes and Characteristics**

- Voters: Users who can view the list of candidates and cast their votes.

- Candidates: Users who can edit their candidate profiles, provide campaign updates, and monitor the election results.

**4.3 Operating Environment**

The VS is a web-based application that can be accessed from any modern web browser.

**4.4 Design and Implementation Constraints**

- The VS is implemented using React with Redux Toolkit and Vite, and uses Axios for API handling.

- Ant Design (antd) is used for the user interface components.

**5. Functional Requirements**

**5.1 User Authentication**

- Voters and candidates must be able to authenticate using their credentials.

- Only authenticated users can access the voting and candidate features.

**5.2 Candidate Features**

- Candidates must be able to edit their candidate profiles, including their name, email, and other relevant information.

- Candidates must be able to provide campaign updates and information about their campaign.

- Candidates must be able to monitor the election results as they are updated.

**5.3 Voting Interface**

- Users must be able to view a list of candidates and their profiles.

- Users must be able to cast their votes for their preferred candidate.

**5.4 Real-time Updates**

- The system must provide real-time updates on the election results for all users.

**6. Non-Functional Requirements**

**6.1 Performance**

- The system must be able to handle a large number of concurrent users without significant performance degradation.

**6.2 Usability**

- The user interface must be intuitive and easy to use, with clear navigation and informative feedback messages.

**6.3 Security**

- User data must be protected and secure, with authentication and authorization mechanisms in place to prevent unauthorized access.

**7. Conclusion**

The Voting System (VS) is a comprehensive web application designed to streamline the election process. By providing user-friendly interfaces for voters and candidates, real-time updates on the election results, and robust security measures, the VS aims to enhance the overall election experience for all stakeholders.