VOTING APP : WEB2

Submitted By : VISHNU A G

1. INTRODUCTION

1.1 Overview of the Project

The Voting App is a Web2-Based Application made specifically for holding online elections in a college or other organization. The entire election process will be simplified by this technology, guaranteeing a simple, effective, and safe method to manage voter registration, approvals, and voting. The Voting App makes use of a cutting-edge tech stack that includes Node.js for the backend server, MongoDB for database administration, and React for the front-end interface to provide an easy-to-use experience for all users. The application comprises three main modules: officials (admin), candidates, and voters, each with specific functionalities to manage their roles within the election process.

1.2 Problem Statement

Elections in universities and other institutions are traditionally conducted by manual procedures that are labour-intensive, prone to error, etc. These procedures may make it more difficult to keep track of voter and candidate registration, maintain voting integrity, and rapidly count and announce results. A digital system that can automate these procedures while guaranteeing accuracy, security, and user-friendliness is required.

1.3 Purpose or Objective of the Project

The Voting App's main goal is to offer a digital platform that makes elections in college or other institutional settings easier and more safe. The application seeks to: Simplify the voter and candidate registration and approval procedures.

- Simplify the voter and candidate registration and approval procedures.
- Ensure a transparent and secure voting process.
- Provide a user-friendly interface for managing elections.
- Enable quick and accurate tallying and announcement of results.

2. TECHNOLOGY STACK

Frontend

- **React JS**: A JavaScript library for building user interfaces.
- **React Router**: A library for managing navigation and routing in React applications.
- **Tailwind CSS**: A utility-first CSS framework for styling.

Backend

- **Node.js**: A JavaScript runtime built on Chrome's V8 JavaScript engine.
- **Express.js**: A web application framework for Node.js, used to build the backend server.

Database

- **MongoDB**: A NoSQL database used for storing application data.

3. PROJECT SCOPE

3.1 Scope of the Project

The Voting App is designed to be used by colleges and institutions for conducting their internal elections. The scope includes:

- User authentication and authorization.
- Candidate and voter registration and approval.
- Election management, including creation, start, and stop functionalities.
- Secure and transparent vote casting.
- Real-time results viewing.

3.2 Key Features and Modules

The application is divided into three main modules with distinct functionalities:

1. **Officials (Admin)**
    - Login with default username and password.
    - View and approve/reject registered candidates and voters.
    - Manage election settings: create, start, and stop elections.
    - View election results.
2. **Candidates**
    - Register and log in.
    - Complete profile with personal information.
    - Wait for approval from officials.
    - Once approved, view election status and results within their profile.
3. **Voters**
    - Register and log in.
    - Check approval status from officials.
    - Once approved, cast votes during the election period.
    - View election results within their profile.

4. FUNCTIONAL SPECIFICATIONS

4.1. Officials Module

- **Candidate Management**: Officials can view a list of registered candidates and approve or reject their applications. Approved candidates are eligible to participate in the election, while rejected ones remain in the pending list.
- **Voter Management**: Similar to candidate management, officials can view and approve or reject voter registrations. Approved voters can cast their votes during the election.
- **Election Management**: Officials can create an election by specifying necessary details, start the election when ready, and stop it once the voting period is over.
- **Result Viewing**: Officials can view the results, which are updated in real-time as votes are cast.

4.2. Candidates Module

- **Registration and Profile Completion**: Candidates register by providing essential details and then log in to complete their profiles, which include academic details and a few personal information.
- **Approval Status**: Candidates can view their approval status. Once approved, their names appear on the ballot for voters.
- **Results Viewing**: Candidates can view the election results in their profiles.

4.3. Voters Module

- **Registration and Approval**: Voters register and wait for approval from officials. Approved voters are eligible to participate in the election.
- **Voting**: During the election period, approved voters can log in, view the list of candidates, and cast their votes. The system ensures that only one vote is cast per voter.
- **Results Viewing**: Voters can view the election results in their profiles after the election ends.

5. WORKFLOW

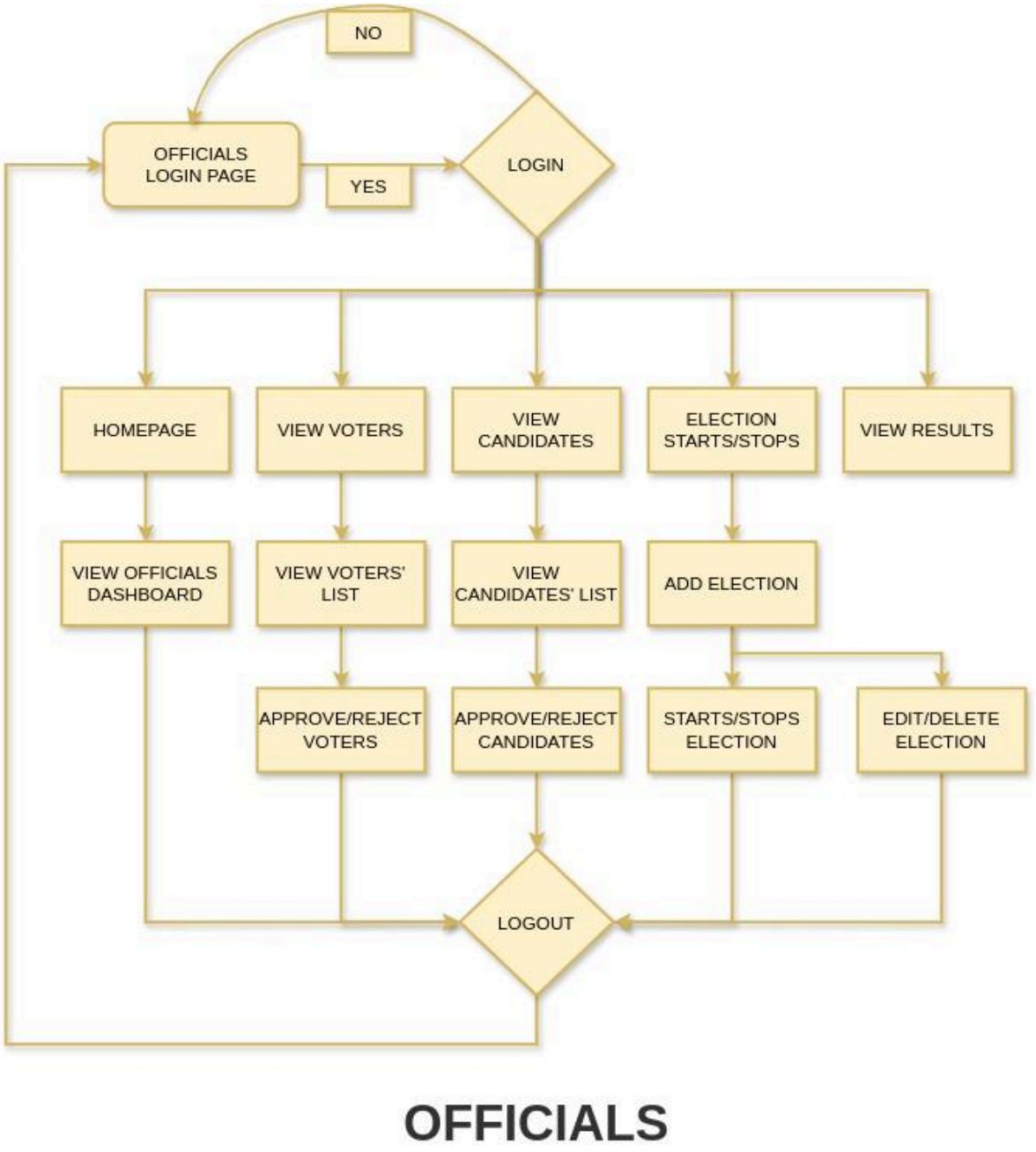
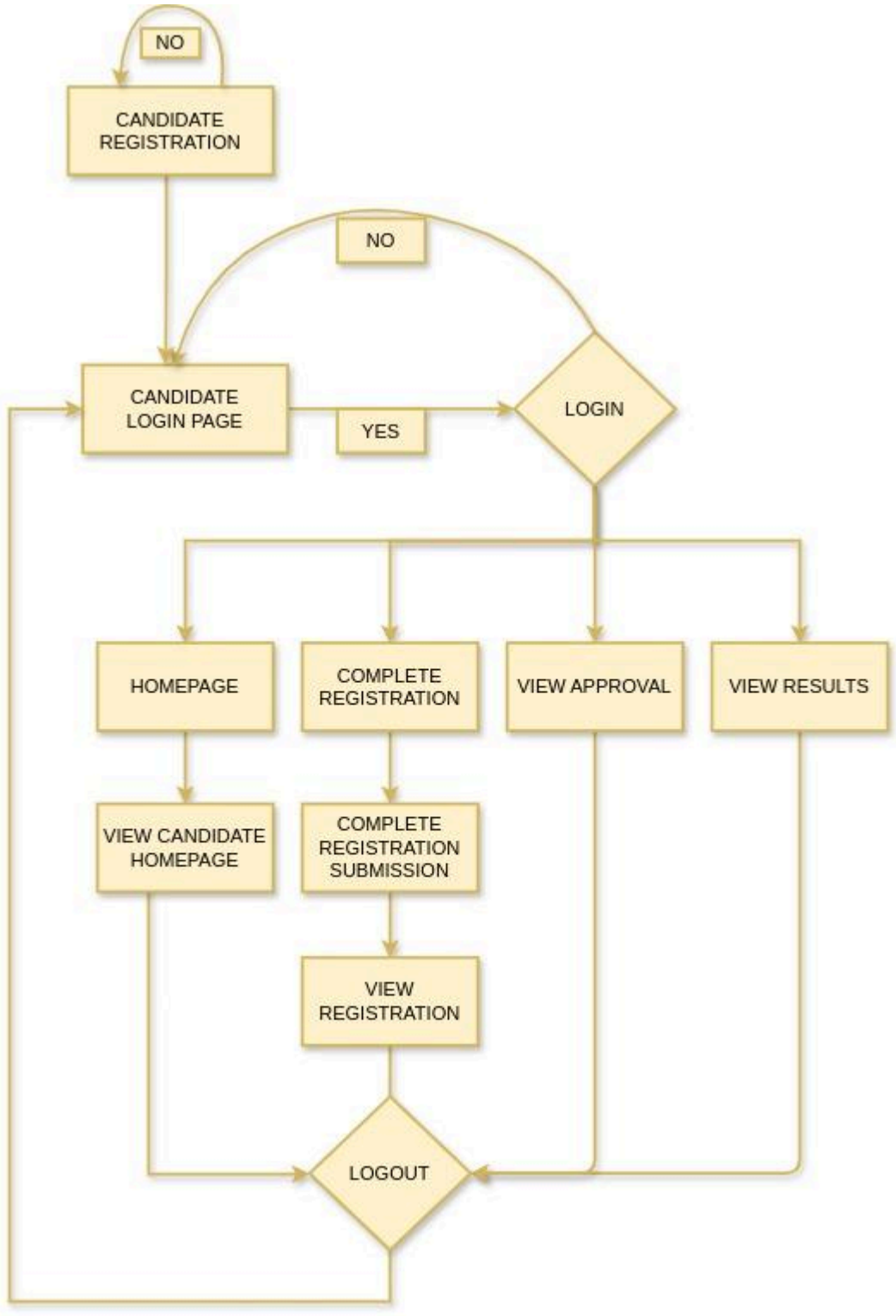


Figure 5.1 Flow Diagram of Officials



CANDIDATES

Figure 5.2 Flow Diagram of Candidates

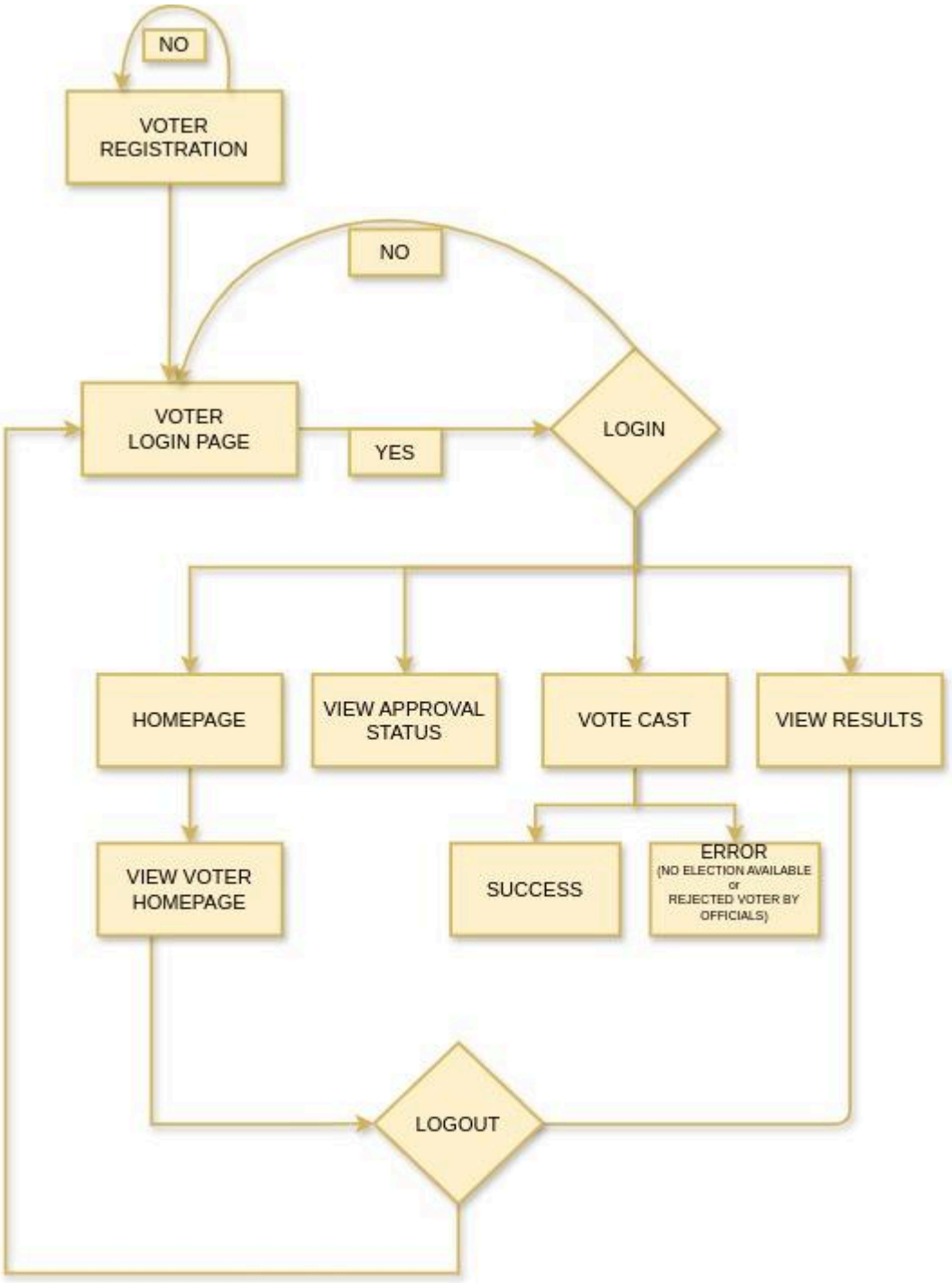


Figure 5.1 Flow Diagram of Voters

6. FUTURE ENHANCEMENT

- Integration of Blockchain Technology: Implementing blockchain technology as part of a Web3 infrastructure to ensure greater security, transparency, and immutability of the voting process. This enhancement will help in creating a decentralized and tamper-proof election system where every vote is recorded on the blockchain, ensuring that the data is immutable and transparent.

- Enhanced Security Features: Adding advanced security features such as two-factor authentication (2FA) for login, end-to-end encryption of all data, and biometric authentication options to further secure the application and protect user data.

- Idea Pitching Platform for Candidates: Introducing a feature where candidates can pitch their ideas and campaigns to the voters. This will include a dedicated section for candidates to upload videos, documents, and other media related to their campaign promises and initiatives.

- Automated notification system for election-related alerts.

- Continuous improvement of the user interface for a better user experience.

7. CONCLUSION

The Voting App addresses the challenges of conducting elections in colleges and institutions by providing a robust, secure, and user-friendly digital platform. By automating the registration, approval, and voting processes, the app ensures a seamless election experience for candidates, voters, and officials. This modern approach to election management not only saves time and reduces errors but also enhances transparency and trust in the election outcomes.

8. REFERENCES

1. Express.js Guide, https://expressjs.com/
2. React Documentation, https://react.dev/learn
3. MongoDB Documentation, https://www.mongodb.com/
4. Tailwind CSS Documentation, https://tailwindcss.com/
5. Node.js Documentation, https://nodejs.org/
6. Docker Documentation, https://www.docker.com/
7. Npmjs, https://www.npmjs.com/