

# ONLINE VOTING PORTAL

## Team Members:

1. Karthik Krishna M (71762133024)
2. Subramanian S (71762133044)

## Abstract:

This project aims to address accessibility, transparency, and convenience issues in voting systems through the development of an online voting platform. Key features include robust user authentication, secure and anonymous voting, and multi-factor authentication for enhanced security. Utilising technologies such as ReactJS for the frontend, Node.js and Express.js for the backend, and MongoDB for database management, the system provides a user-friendly interface for voter registration, login, and casting of votes. Future plans include enhancing result display, implementing biometric authentication, and collaborating with government agencies to ensure regulatory compliance and widespread adoption. Cloud deployment facilitates accessibility, allowing voters to participate from anywhere with internet access. Overall, the project aims to foster trust in democratic processes while promoting inclusivity and convenience for voters.

## a) Problem Identification:

- ❖ **Accessibility:** Many regions struggle with low voter turnout due to accessibility issues. People with disabilities or those living in remote areas find it challenging to access traditional voting booths. Developing an online voting system can help address these challenges by allowing voters to participate from the comfort of their homes using internet-enabled devices.
- ❖ **Transparency and Trust:** Ensuring the integrity of the voting process is crucial for fostering trust in democratic systems. Instances of fraud or tampering can undermine confidence in election outcomes. An online voting system can incorporate features like cryptographic protocols to ensure the security and transparency of the voting process, thus enhancing trust in the system.
- ❖ **Convenience:** Traditional voting methods often require individuals to physically visit a polling station, which can be inconvenient for busy individuals or those unable to leave their homes. By providing an online platform for voting, individuals can cast their votes at their convenience, leading to higher participation rates and overall convenience for voters.

## b) Software Requirements Specification (SRS) & Modules Formation:

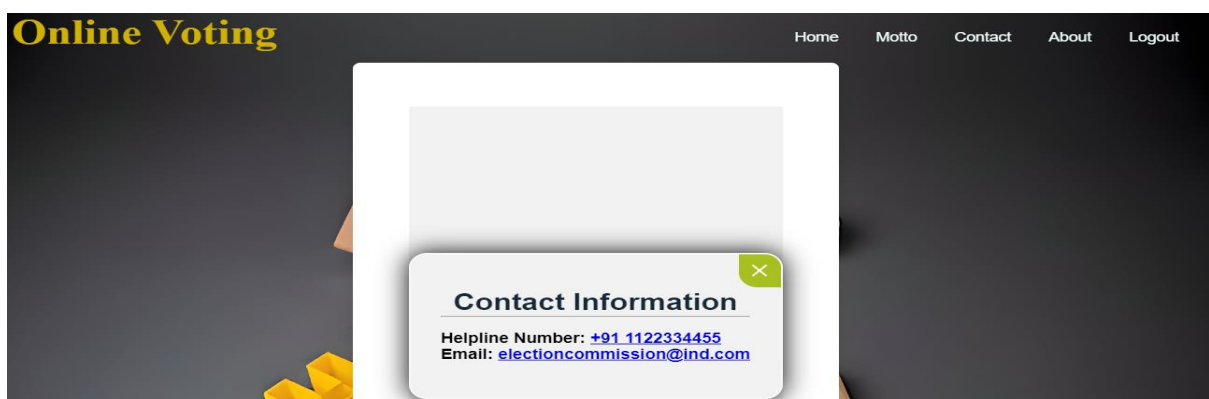
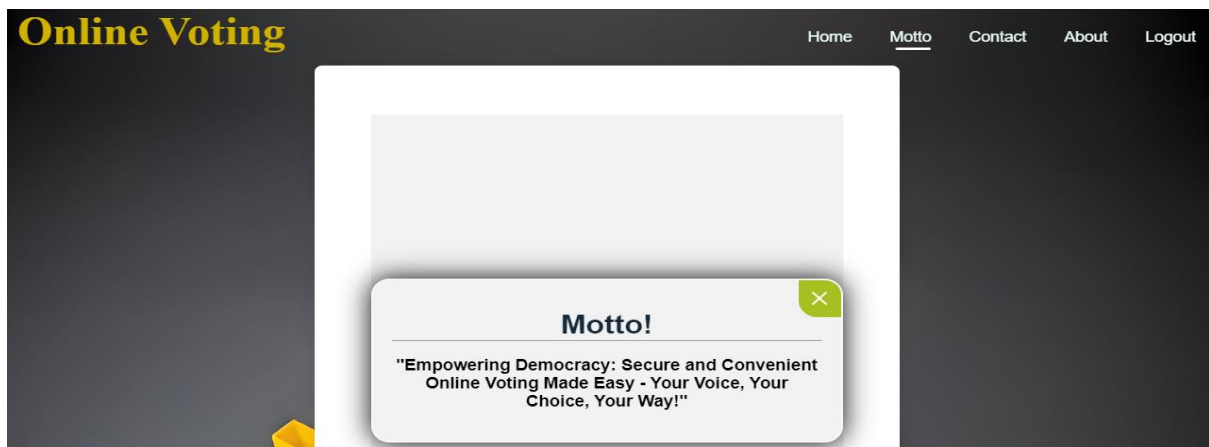
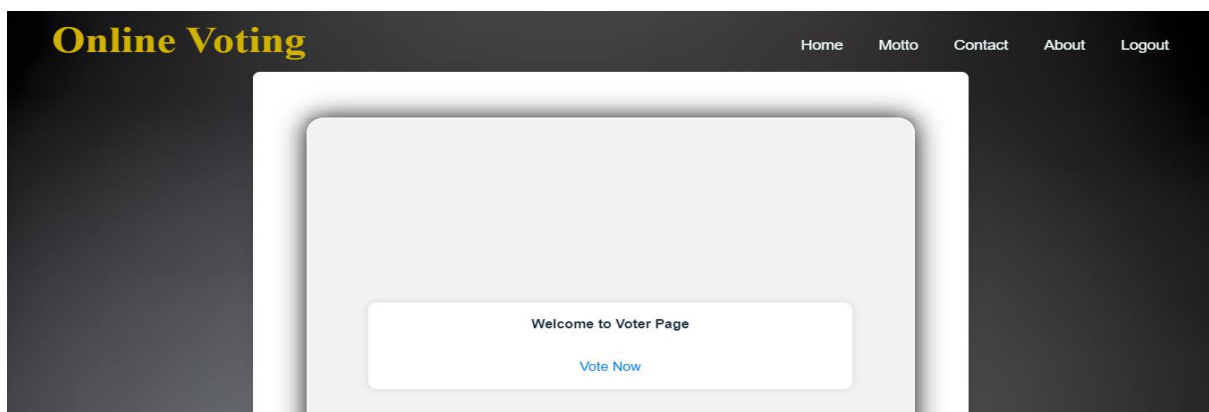
### SRS:

- **User authentication and authorization:** Implement a robust authentication system to verify the identity of voters and authorise access to the voting platform securely.
- **Multi-factor Authentication with OTP:** Implement multi-factor authentication using OTP for enhanced security.
- **Secure and anonymous voting:** Ensure that votes are securely encrypted and anonymized to protect voter privacy and prevent tampering.

### Modules:

- **User Management Module:** Handles user registration, login, and authentication.
- **Voting Module:** Enables registered users to cast their votes securely.
- **Home Module:** Provides the landing page of the online voting system, displaying key information, announcements, and links to other sections of the platform.
- **About Module:** Presents detailed guidelines and rules for the voting process, including eligibility criteria, voting procedures, and code of conduct.
- **Contact Module:** Offers contact information, including helpline numbers or email addresses, for users to reach out for assistance or inquiries related to the online voting system.

### Output Screenshots:



# Online Voting

[Home](#) [Motto](#) [Contact](#) [About](#) [Logout](#)

## About

Online voting systems prioritize security, employing methods like secure login credentials, multi-factor authentication, and encryption to maintain the election process's integrity.

Voter registration typically involves online verification of identity through official documents.

Websites hosting these systems focus on user-friendly interfaces and accessibility features to ensure an intuitive and inclusive voting experience.

Platforms provide comprehensive information about candidates, parties, and election issues, including candidate profiles, party platforms, and policy details.

The voting process involves selecting candidates or indicating preferences on issues, guided by clear instructions to minimize confusion.

localhost:3000/VoterPage

Classes Jobaaj Learnings ... Resume - FlowCV

localhost:3000 says  
VoterID is not in the correct format. Please enter a valid VoterID.

OK

### Vote Now

☐ Indian National Congress - Sonia Gandhi

☐ Bharatiya Janata Party - Narendra Modi

☒ Aam Aadmi Party - Arvind Kejriwal

☐ Bahujan Samaj Party - Mayawati

☐ Communist Party of India (Marxist) - Sitaram Yechury

Submit Vote

localhost:3000/VoterPage

Classes Jobaaj Learnings ... Resume - FlowCV

localhost:3000 says  
Vote submitted successfully

OK

### Vote Now

☐ Indian National Congress - Sonia Gandhi

☐ Bharatiya Janata Party - Narendra Modi

☒ Aam Aadmi Party - Arvind Kejriwal

☐ Bahujan Samaj Party - Mayawati

☐ Communist Party of India (Marxist) - Sitaram Yechury

Submit Vote

## Database Details:

- **User data:** Stores user information such as name, email, and password hash for authentication.
- **Vote records:** Records each vote cast, linking it to the corresponding user and party details.
- **Part Details:** Stores information about system components including part name, description, quantity, supplier details, and acquisition date.
- **Eligible Voter Details:** Contains personal information and eligibility status of voters, such as name, address, date of birth, voter status, registration status, and district/constituency.

## Output Screenshots:

The screenshot shows the MongoDB Atlas web interface. The top navigation bar includes the Atlas logo, a user profile for 'Karthik's Or...', and links for 'Access Manager' and 'Billing'. The main interface is divided into a left sidebar with navigation options like 'Overview', 'DEPLOYMENT', 'Database', 'Data Lake', 'SERVICES', and 'SECURITY'. The 'Database' section is expanded, showing a tree view of the 'Voting\_portal' database with collections: 'Eligibletovote', 'Parties', 'Users', 'VoteDetails', 'sample\_mflix', and 'test'. The 'Users' collection is selected, and its details are shown on the right. The 'Voting\_portal.Users' collection has a storage size of 36KB, logical data size of 916B, total documents of 5, and index total size of 36KB. A search bar is present with the text 'Type a query: { field: 'value' }'. Below the search bar, the query results are displayed, showing a single document with fields: '\_id', 'username', 'email', 'mobile', 'password', 'confirmPassword', and 'voterId'.

**Voting\_portal.Users**

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 916B TOTAL DOCUMENTS: 5 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

Filter Type a query: { field: 'value' } Reset Apply Options

QUERY RESULTS: 1-5 OF 5

```
{
  "_id": ObjectId("6602e02126f705e4ae30fbc9"),
  "username": "Subramanian",
  "email": "subramanian160104@gmail.com",
  "mobile": "9872744511",
  "password": "mani@441",
  "confirmPassword": "mani@441",
  "voterId": "wjb4446901"
}
```

The screenshot shows the MongoDB Atlas web interface for the 'Voting\_portal.VoteDetails' collection. The left sidebar is the same as the previous screenshot, with the 'VoteDetails' collection selected. The 'Voting\_portal.VoteDetails' collection has a storage size of 36KB, logical data size of 819B, total documents of 13, and index total size of 36KB. A search bar is present with the text 'Type a query: { field: 'value' }'. Below the search bar, the query results are displayed, showing two documents with fields: '\_id', 'voterId', and 'partyName'.

**Voting\_portal.VoteDetails**

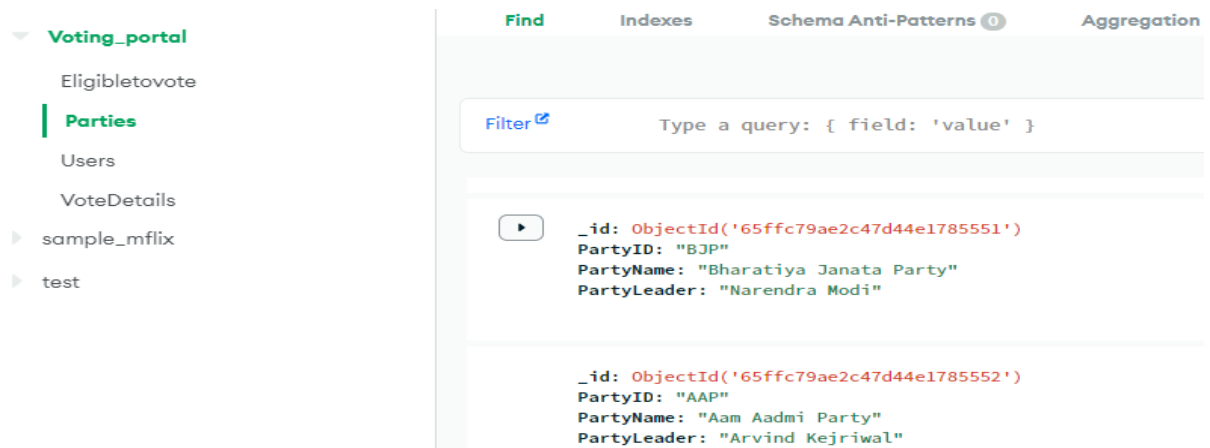
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 819B TOTAL DOCUMENTS: 13 INDEXES

Find Indexes Schema Anti-Patterns Aggregation

Filter Type a query: { field: 'value' } Reset Apply Options

```
{
  "_id": ObjectId("6602eb8f530ea455a66e7714"),
  "voterId": "wjb1234567",
  "partyName": "AAP"
}
```

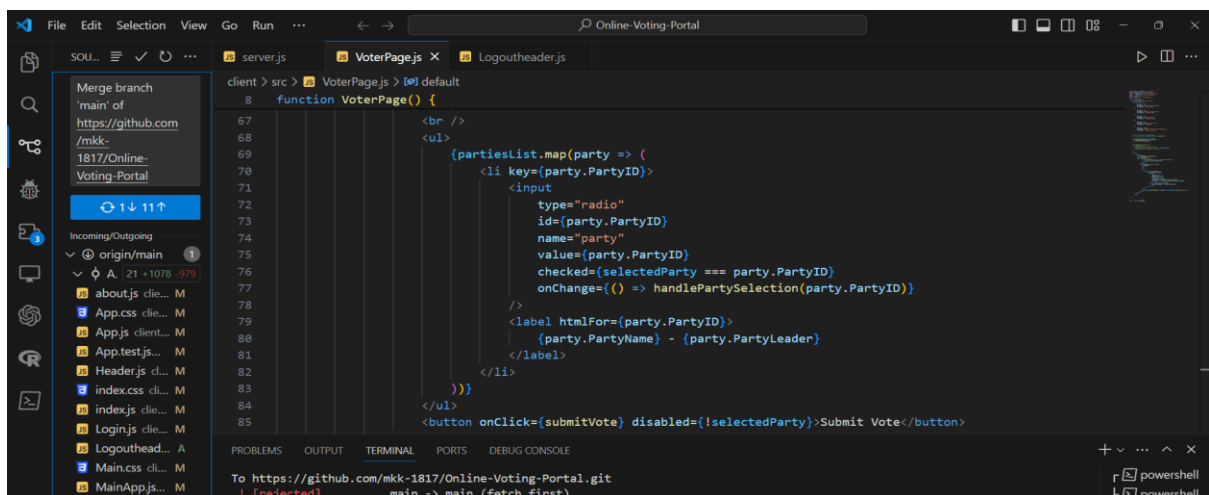
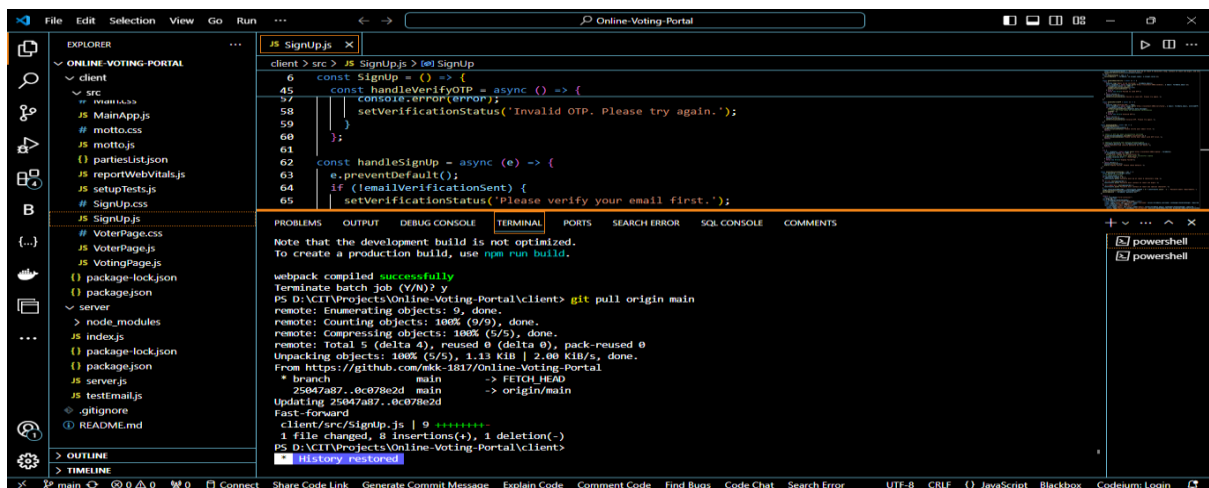
```
{
  "_id": ObjectId("66030b46cf1a4728c95d321f"),
  "voterId": "wjb4221411",
  "partyName": "AAP"
}
```



### c) GitHub Collaborative Development:

- **Repository setup:** Create a Git repository with separate branches for development, testing, and production environments to facilitate collaborative development.
- **Active participation:** Encourage regular commits, pull requests, code reviews, and issue tracking to ensure efficient collaboration and code quality.

### Output Screenshots:



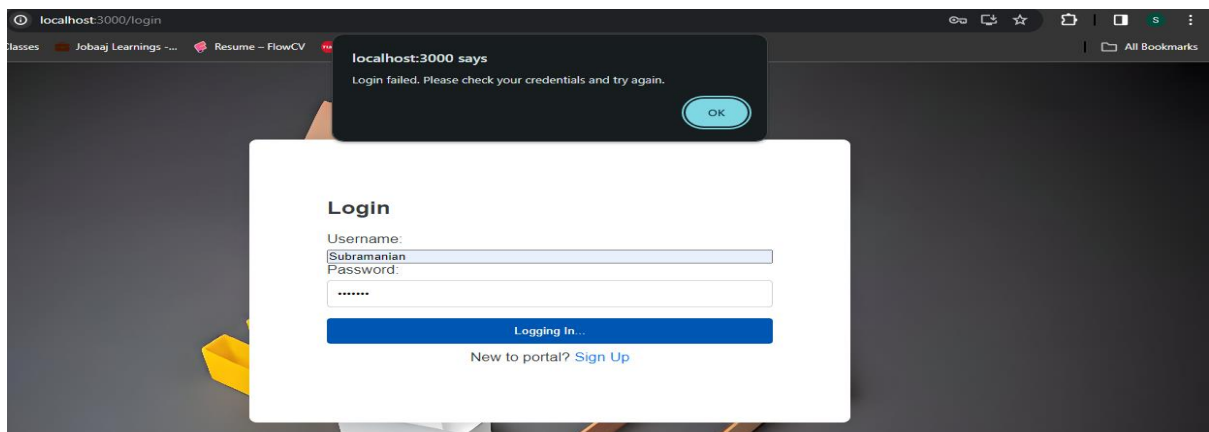
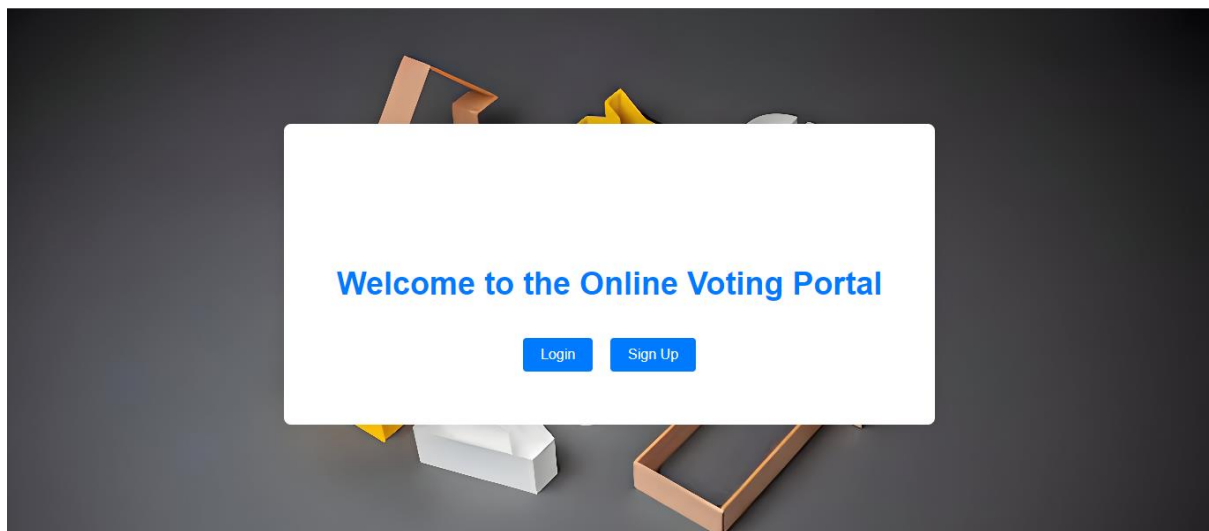
#### d) Functionalities and Users of the System:

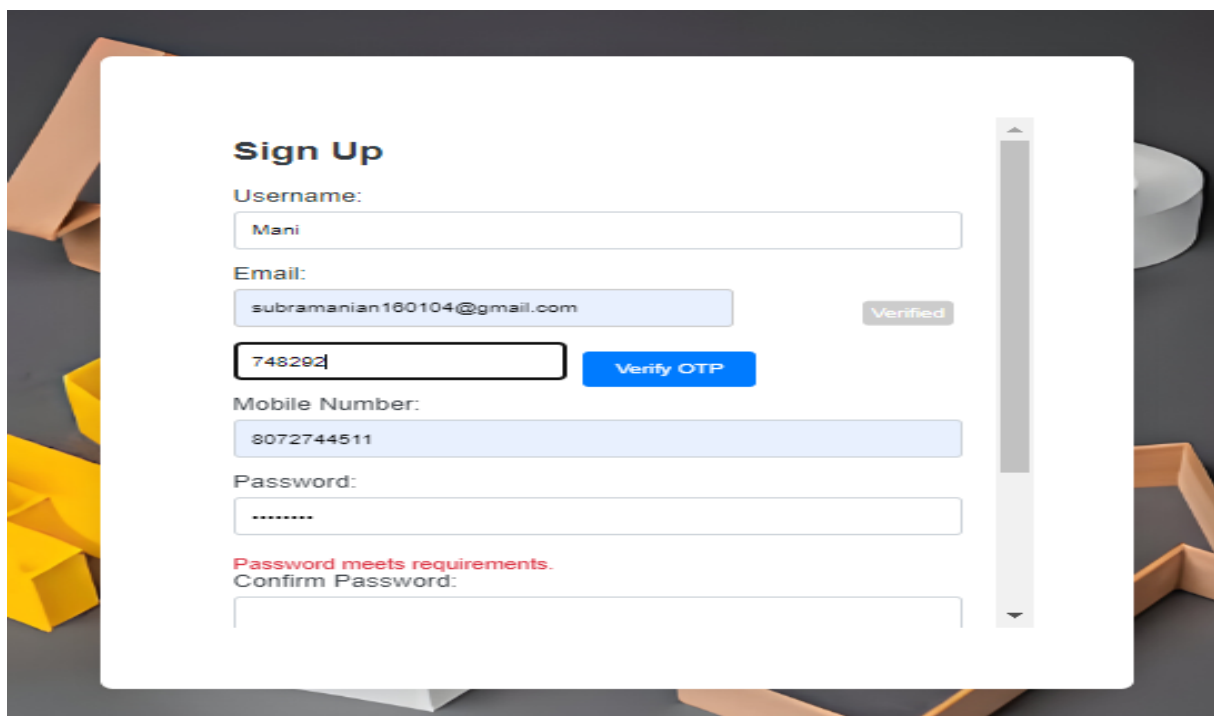
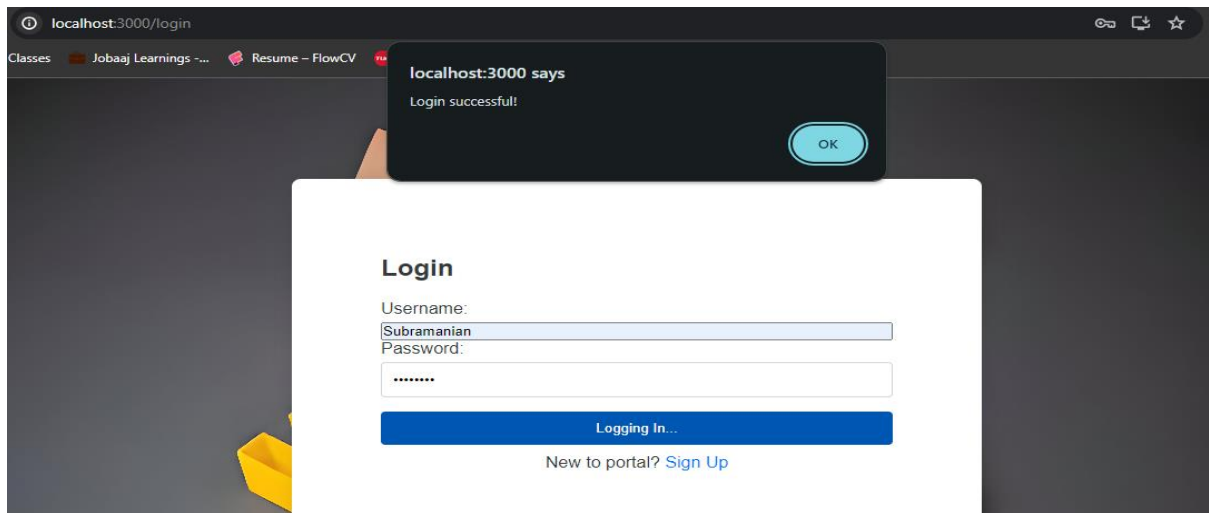
- **Sign Up:** Allows users to create an account by providing their username, email, mobile number, password, confirmed password, and voter ID number. It includes email verification through OTP (one-time password).
- **Login:** Enables registered users to log in using their email and password.
- **Voter Page:** Provides a page where authenticated users can view information about political parties and leaders and submit their votes for a selected party.
- **Contact Information:** Displays contact information, including a helpline number and email address for contacting the election commission.

#### Users:

- **General Users:** Individuals who want to participate in voting by creating an account, logging in, and submitting their votes.
- **Admin (Not explicitly implemented):** Hypothetical users who might manage the system, such as monitoring user activities, managing parties, or overseeing the voting process.

#### Output Screenshots:





## OTP Verification Inbox x

44

71762133044@cit.edu.in

Your OTP for verification is: 225823

3

44

71762133044@cit.edu.in

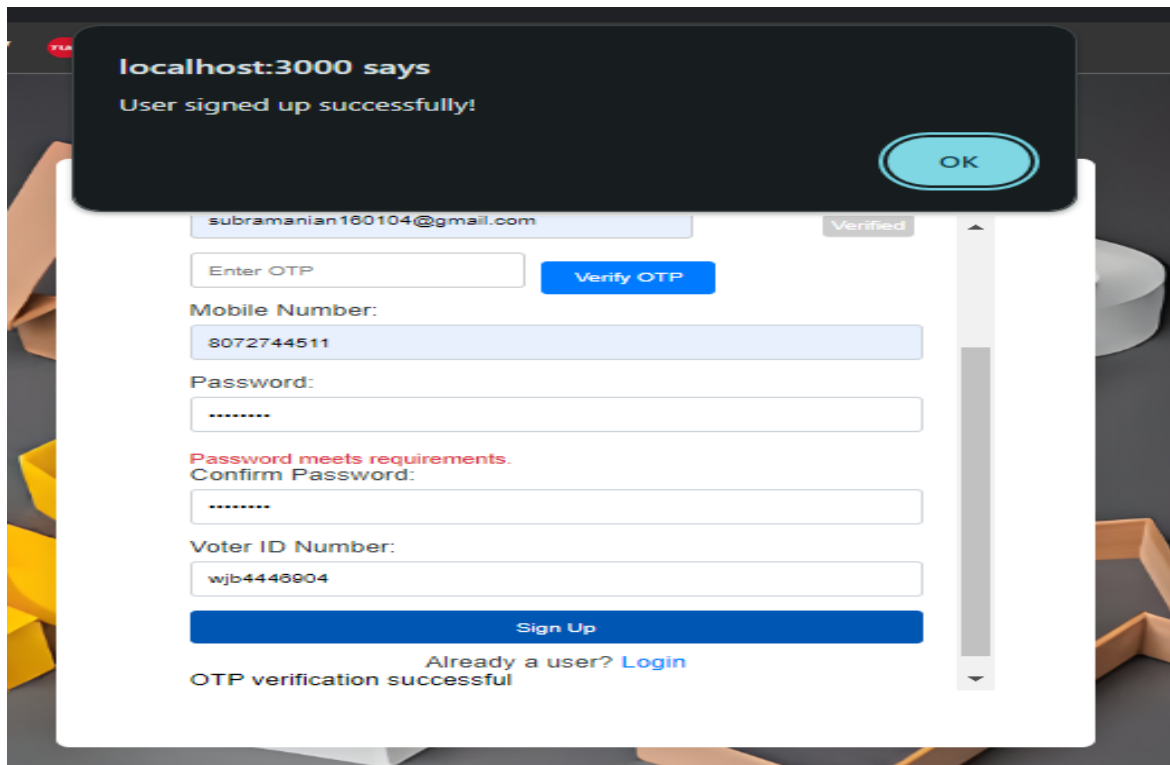
Your OTP for verification is: 355882

44

71762133044@cit.edu.in

to me ▾

Your OTP for verification is: 748292



#### e) Concepts Used:

##### ReactJS:

- **Functional Components:** All components (SignUp, Login, VoterPage, Contact) are functional components defined using React Hooks.
- **State Management:** Utilises useState hook to manage component state, such as form data, email verification status, OTP, login status, selected party, etc.
- **Routing:** Utilises react-router-dom for client-side routing, enabling navigation between different views or pages.
- **Conditional Rendering:** Conditional rendering is used to display elements conditionally based on certain states, like whether email verification is sent, login status, etc.
- **Event Handling:** Handles events like form submission, button clicks, and input changes using event handlers.

##### Node.js and Express.js:

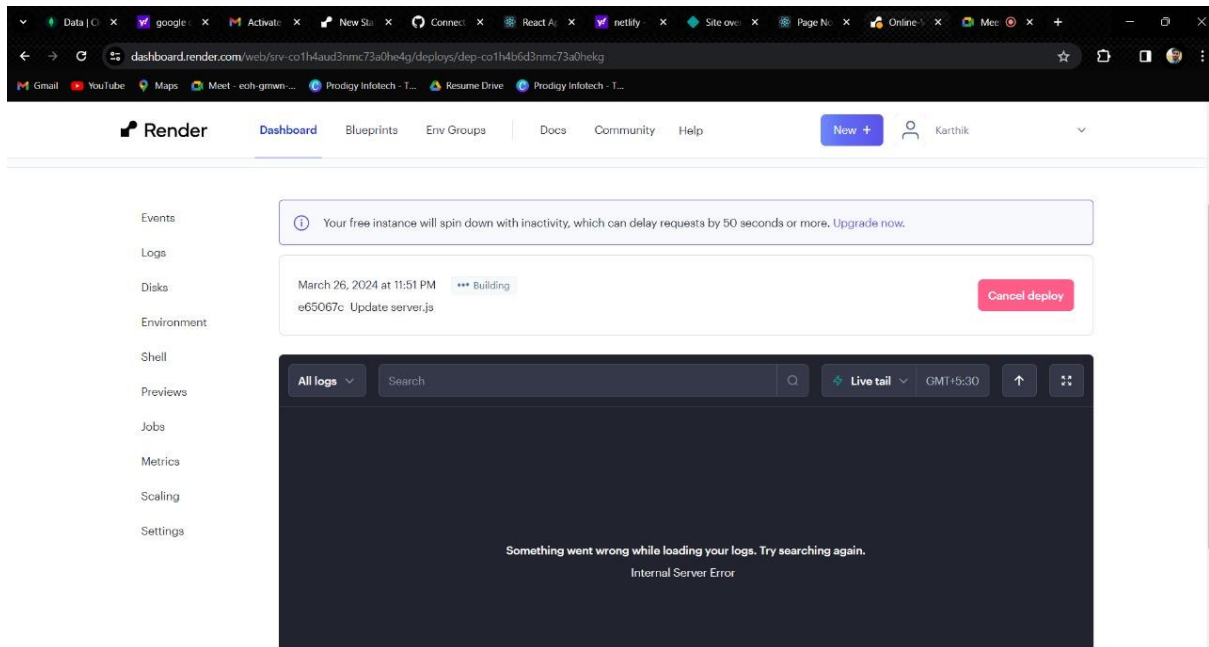
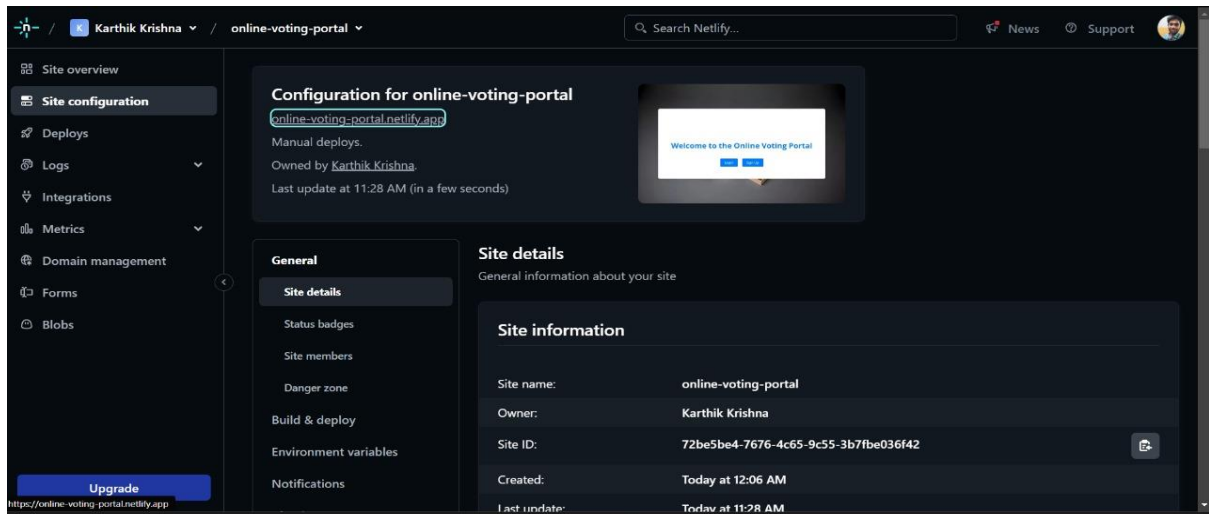
- **Server-Side Routing:** Defines routes using Express.js to handle various HTTP requests like POST requests for sign up, login, sending OTP, verifying OTP, submitting votes, etc.
- **Middleware:** Utilises middleware like body-parser for parsing request bodies and cors for handling Cross-Origin Resource Sharing.
- **Error Handling:** Implements error handling for asynchronous operations using try-catch blocks and appropriate status codes for responses.
- **Database Interaction:** Connects to a MongoDB database using mongoose and performs operations like inserting user data, adding vote details, etc.
- **Sending Emails:** Integrates with node mailer to send emails for email verification using SMTP service (Gmail in this case).
- **Security:** Although not explicitly implemented in the provided code, security measures like password hashing, input validation, and authentication/authorization mechanisms should be considered in a production environment.



### e) Cloud Deployment:

- **Backend:** <https://online-voting-portal.onrender.com/>
- **Frontend:** <https://online-voting-portal.netlify.app/>

### Output Screenshots:



### c) Future Plan:

- **Result Page Enhancement:** Develop a dedicated result page to display voting outcomes, providing transparency and accountability to users and stakeholders.
- **Biometric authentication:** Implement biometric authentication methods such as fingerprint or facial recognition for added security and accessibility.
- **Collaborate with government agencies:** Work with government agencies to ensure regulatory compliance and promote wider adoption of the online voting system.