|  |
| --- |
| **A**  **PROJECT REPORT ON** |
|  |
|  |
| ELECTION COMMISSION OF INDIA |
|  |
|  |
| SUBMITTED IN  PARTIAL FULFILLMENT OF  **DIPLOMA IN ADVANCED COMPUTING (PG-DAC)** |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **Dev Gupta**  **Suraj Kumar**  **Vipul Pal**  **Yash Pawar** |
|  |
|  |
| **UNDER THE GUIDENCE OF**  **Mr.’s POOJA JAISWAL** |
|  |
|  |
|  |
| **AT**  **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE** |

|  |  |
| --- | --- |
|  | |
| **SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,**  **PUNE.** | |
|  | |
|  | |
|  | |
|  | |
| **CERTIFICATE** | |
|  | |
| This is to certify that the project | |
|  | |
| ELECTION COMMISSION OF INDIA | |
|  | |
| Has been submitted by | |
|  | |
| **Dev Gupta**  **Suraj Kumar**  **Vipul Pal**  **Yash Pawar** | |
|  | |
|  | |
| In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC AUG2024)** as prescribed by The **CDAC** ACTS, PUNE. | |
|  | |
|  | |
| Place: Pune | Date: 11-feb-2024 |
|  | |
|  | |
|  | |
|  | |
|  | |
| **Mr.’s Pooja Jaiswal** | **Mr. Yogesh Kolhe** |
| **Project Guide** | **Course Coordinator** |

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to all those who have supported and guided me throughout the development of this project, **"Digital Voting System."**

First and foremost, I am extremely thankful to **SUNBEAM Institute, Pune**, for providing me with this opportunity to work on such an insightful project. I extend my heartfelt appreciation to my project guide **POOJA JAISWAL**, for their invaluable guidance, encouragement, and constructive feedback, which played a crucial role in shaping this project.

I am also grateful to my team members, **DEV GUPTA, SURAJ KUMAR, VIPUL PAL, YASH PAWAR**, for their collaboration, dedication, and effort in successfully implementing this system. Their technical expertise and teamwork made this project a rewarding experience.

Furthermore, I would like to acknowledge my faculty members and mentors for their continuous support and for equipping me with the necessary knowledge and skills to undertake this project. Their insightful suggestions have greatly enhanced the quality of this work.

Lastly, I extend my gratitude to my family and friends for their constant motivation and encouragement throughout this journey.

This project has been a significant learning experience, and I sincerely appreciate everyone's contribution toward its successful completion.

**ABSTRACT**

The Digital Voting System is a secure and efficient online platform designed to modernize the traditional voting process by enabling voters to cast their votes electronically. This system ensures transparency, security, and accuracy while reducing manual errors, paper usage, and logistical challenges associated with conventional voting methods.

The project is developed using React for the front end and Java for the back end, ensuring a seamless and responsive user experience. The system provides role-based access, allowing admin, voters, and candidates to interact with the platform based on their privileges. Secure authentication mechanisms, including encrypted credentials, are implemented to prevent unauthorized access. The system also incorporates database management for storing voter details, candidate information, and election results securely.

This project aims to enhance the electoral process by minimizing fraudulent activities and improving accessibility for voters. The implementation of real-time data updates and user dashboards enables better election monitoring and result tracking. The Digital Voting System serves as a step toward digital transformation in elections, ensuring a fair, reliable, and efficient voting experience.

**INDEX**

|  |  |  |
| --- | --- | --- |
|  | **INTRODUCTION** | 1 |
|  | 1.1 Introduction | 1 |
|  | **Product Overview and Summary** | 2 |
|  | 2.1 Purpose | 2 |
|  | 2.2 Scope | 2 |
|  | 2.3 User Classes and Characteristics | 3 |
|  | 2.4 Design and Implementation Constraints | 3 |
|  | **REQUIREMENTS** | 4 |
|  | 3.1 Functional Requirements | 4 |
|  | 3.1.1 Use case for Administrator. | 4 |
|  | 3.1.2 Use case for Customer. | 4 |
|  | 3.1.3 E-R Diagram | 5 |
|  | 3.2 Non - Functional Requirements | 6 |
|  | 3.2.1 Usability Requirement | 6 |
|  | 3.2.2 Performance Requirement | 6 |
|  | 3.2.3 Reliability Requirement | 6 |
|  | 3.2.4 Portability Requirement | 7 |
|  | 3.2.5 Security Techniques | 7 |
|  | **PROJECT DESIGN** | 8 |
|  | 4.1 Data Model | 8 |
|  | 4.1.1 Database Design | 8 |
|  | **CONCLUSION** | 25 |

**1.INTRODUCTION**

The Digital Voting System is an advanced online platform designed to facilitate secure, transparent, and efficient elections. Traditional voting methods often face challenges such as manual errors, time-consuming processes, logistical difficulties, and security vulnerabilities. This project aims to overcome these limitations by leveraging modern web technologies to create a user-friendly, secure, and scalable voting system.

The system provides role-based access for administrators, candidates, and voters, ensuring that each user can interact with the platform according to their privileges. Security is a major focus, incorporating authentication, data encryption, and real-time monitoring to prevent fraudulent activities. The platform enables seamless voter registration, candidate management, vote casting, and result computation, making the election process more accessible and reliable.

This project is developed using React for the front end and Java for the back end, with a SQL database for secure data storage. The integration of these technologies ensures high performance and scalability, making it suitable for elections of various scales, from small organizations to national-level implementations.

**2.Product Overview and Summary**

**2.1 Purpose**

The purpose of this **Digital Voting System** is to provide a **secure, transparent, and efficient** alternative to traditional voting methods. It aims to:

* **Eliminate paper-based voting** to reduce environmental impact and logistical issues.
* **Ensure vote integrity and security** using encryption and authentication mechanisms.
* **Enhance accessibility** by allowing voters to cast their votes remotely.
* **Improve efficiency** by automating the vote counting process and reducing human errors.
* **Ensure transparency** by providing real-time election monitoring and result announcements.

**2.2 Scope**

The **Digital Voting System** is designed to accommodate different types of elections, whether at the **institutional, corporate, or governmental level**. The system includes the following functionalities:

* **User Registration & Authentication**: Secure login for administrators, voters, and candidates.
* **Candidate Management**: Admins can add, update, or remove candidate details.
* **Voter Verification & Voting**: Voters can securely cast their votes while ensuring their identity is verified.
* **Vote Counting & Result Generation**: Automatic vote tallying with real-time updates.
* **Secure Database Management**: Storage of all election-related data, including voter credentials, candidate details, and results.
* **Audit & Logging**: Every action performed in the system is logged for future reference and transparency.

**2.3 User Classes and Characteristics**

The system is designed for different user roles, each with specific functionalities:

* **Administrator (Admin)**
  + Manages voter and candidate registrations.
  + Configures election rules and monitors the election process.
  + Generates and publishes election results.
* **Voter**
  + Registers and verifies their identity before voting.
  + Casts their vote securely within the election timeline.
  + Views election results once voting is completed.
* **Candidate**
  + Registers as a participant in the election.
  + Views the number of votes received after the election ends.

**2.4 Design and Implementation Constraints**

While designing and implementing the **Digital Voting System**, certain constraints and challenges were considered:

* **Security & Authentication**: Ensuring that only authorized users access the system, preventing multiple voting by the same user, and securing voter identities.
* **Scalability**: The system must be capable of handling a large number of voters without performance degradation.
* **Real-time Updates**: Implementing efficient data handling to ensure real-time vote tracking and result updates.
* **Legal & Compliance Issues**: The system must comply with relevant **election regulations, data privacy laws, and security standards** to be viable for real-world implementation.

**3.REQUIREMENT**

**3.1 Functional Requirements**

3.1.1 Use case for Administrator and Voter.



**E-R Diagram**



**3.2 Non-Functional requirement.**

**3.2.1 Usability Requirement**

* The system should have a **user-friendly and intuitive interface** for voters, administrators, and candidates.
* The **navigation and voting process** should be simple and accessible, even for users with minimal technical knowledge.
* The interface should be **responsive**, ensuring compatibility across various screen sizes (desktop, mobile, and tablet).
* Instructions should be **clear and concise**, guiding users at every step to prevent errors while voting.
* The system should support **multiple languages** to cater to a diverse user base.

**3.2.2 Performance Requirement**

* The system should handle **multiple concurrent users** without performance degradation.
* The voting process, including authentication and result computation, should be **completed within seconds** to ensure a seamless user experience.
* The database should be optimized to handle **large amounts of data efficiently** (voter records, votes, candidate details, and results).
* The system should provide **real-time updates** for vote counts and election progress.
* The response time for any action (e.g., login, vote submission, result generation) should be **less than 3 seconds** under normal conditions.

**3.2.3 Reliability Requirement**

* The system must be available **99.9% of the time** during the election period to ensure smooth voting operations.
* The voting process should be **fail-proof**, meaning no votes should be lost due to system crashes or unexpected failures.
* The system should support **automatic data backup** to prevent data loss in case of server failure.
* The system should ensure **vote integrity**, meaning once a vote is cast, it cannot be altered or deleted.

**3.2.4 Portability Requirement**

* The application should be compatible with **different operating systems** such as Windows, Linux, and macOS.
* The web-based platform should run smoothly on **modern web browsers** like Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
* The system should be **accessible on both desktop and mobile devices**, maintaining full functionality across different screen sizes.

**3.2.5 Security Techniques**

* **User Authentication & Access Control**:
  + Secure login mechanism using **hashed passwords** (e.g., bcrypt, MD5, or SHA encryption).
  + Role-based access control to ensure **admin, voters, and candidates** have appropriate permissions.
* **Data Encryption**:
  + Sensitive data (user credentials, votes) should be **encrypted in transit and at rest** to prevent unauthorized access.
* **Vote Security & Anonymity**:
  + Each vote should be **encrypted and stored securely** to maintain voter confidentiality.
  + A voter must be able to **cast their vote only once**, ensuring fairness.
* **Database Security**:
  + The database should be protected against **SQL Injection, XSS (Cross-Site Scripting), and CSRF (Cross-Site Request Forgery) attacks**.
  + Regular security patches and updates should be applied.

**4.PROJECT DESIGN**

**4.1 DATA MODEL**

# 4.1.1 Database Design

## Admins Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for admin |
| name | VARCHAR(255) | NOT NULL | Admin's full name |
| email | VARCHAR(255) | UNIQUE, NOT NULL | Admin's email |
| password | VARCHAR(255) | NOT NULL | Admin's encrypted password |

## Voters Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for voter |
| first\_name | VARCHAR(255) | NOT NULL | Voter's first name |
| last\_name | VARCHAR(255) | NOT NULL | Voter's last name |
| contact\_number | VARCHAR(10) | UNIQUE, NOT NULL | Voter's contact number |
| dob | DATE | NOT NULL | Voter's date of birth |
| constituency\_id | INT | FOREIGN KEY → constituencies.id, NOT NULL | Voter's constituency |
| has\_voted | BOOLEAN | DEFAULT FALSE | Whether the voter has voted or not |
| adhaar\_number | VARCHAR(12) | UNIQUE, NOT NULL | Voter's Aadhaar number |
| created\_on | DATETIME | DEFAULT CURRENT\_TIMESTAMP | When the voter was registered |
| updated\_on | DATETIME | AUTO-UPDATE ON CHANGE | Last profile update timestamp |

## Candidates Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| candidate\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for candidate |
| voter\_id | INT | FOREIGN KEY → voters.id, NOT NULL | Candidate's voter ID |
| political\_party\_id | INT | FOREIGN KEY → political\_parties.party\_id, NOT NULL | Candidate's party ID |
| constituency\_id | INT | FOREIGN KEY → constituencies.id, NOT NULL | Candidate's constituency |
| is\_valid | INT | DEFAULT 0 | Candidate validation status |
| votes | INT | DEFAULT 0 | Total votes received |
| candidate\_image | LONGBLOB | NULL | Candidate's image file |
| image\_path | VARCHAR(1024) | NULL | Path to candidate's image |

## Constituencies Table

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for constituency |
| constituency\_name | VARCHAR(255) | UNIQUE, NOT NULL | Name of constituency |
| total\_voters | INT | DEFAULT 0 | Total registered voters |
| votes\_casted | INT | DEFAULT 0 | Total votes cast in the constituency |

## Elections Table

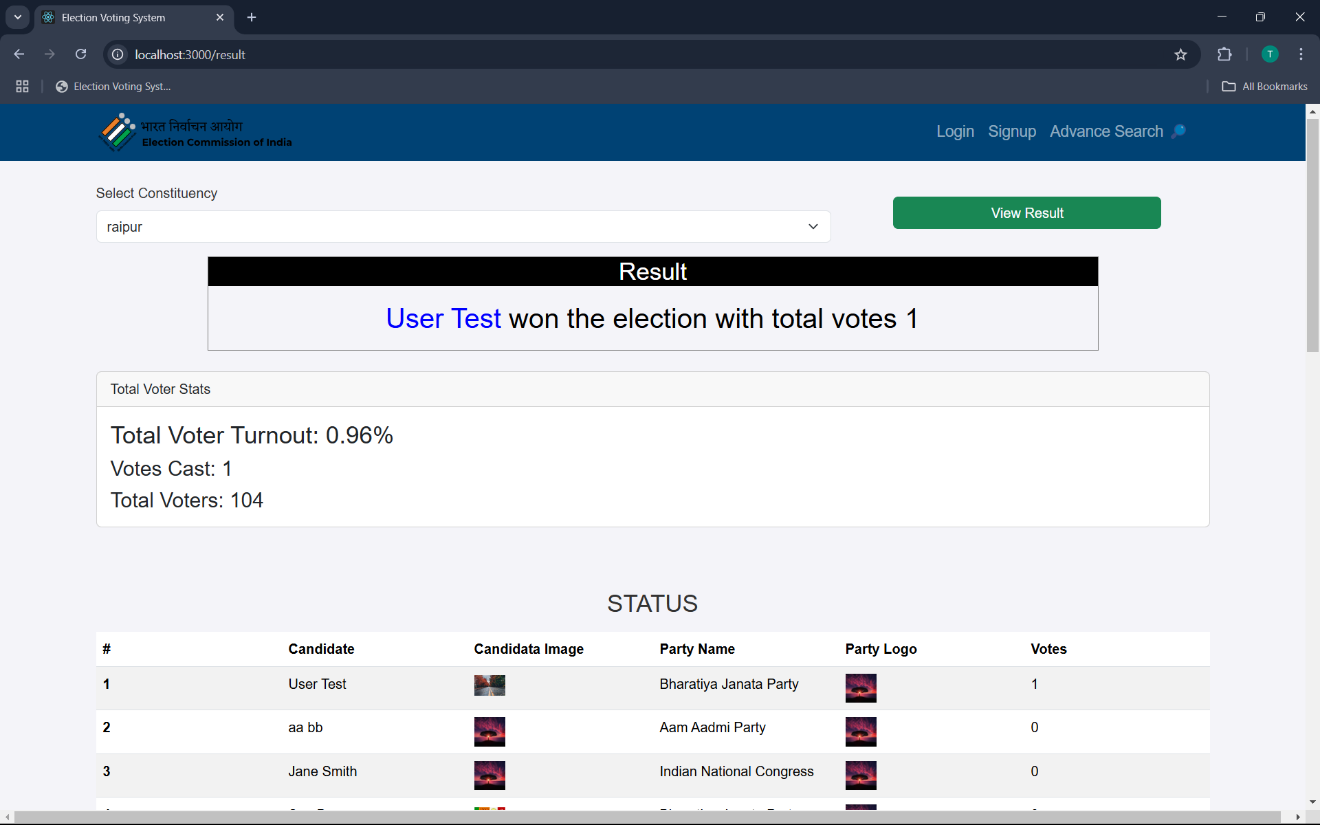
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| election\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for election |
| constituency\_id | INT | FOREIGN KEY → constituencies.id, NOT NULL | Associated constituency |
| election\_start\_time | DATETIME | NOT NULL | Election start time |
| election\_end\_time | DATETIME | NOT NULL | Election end time |

## Feedbacks Table

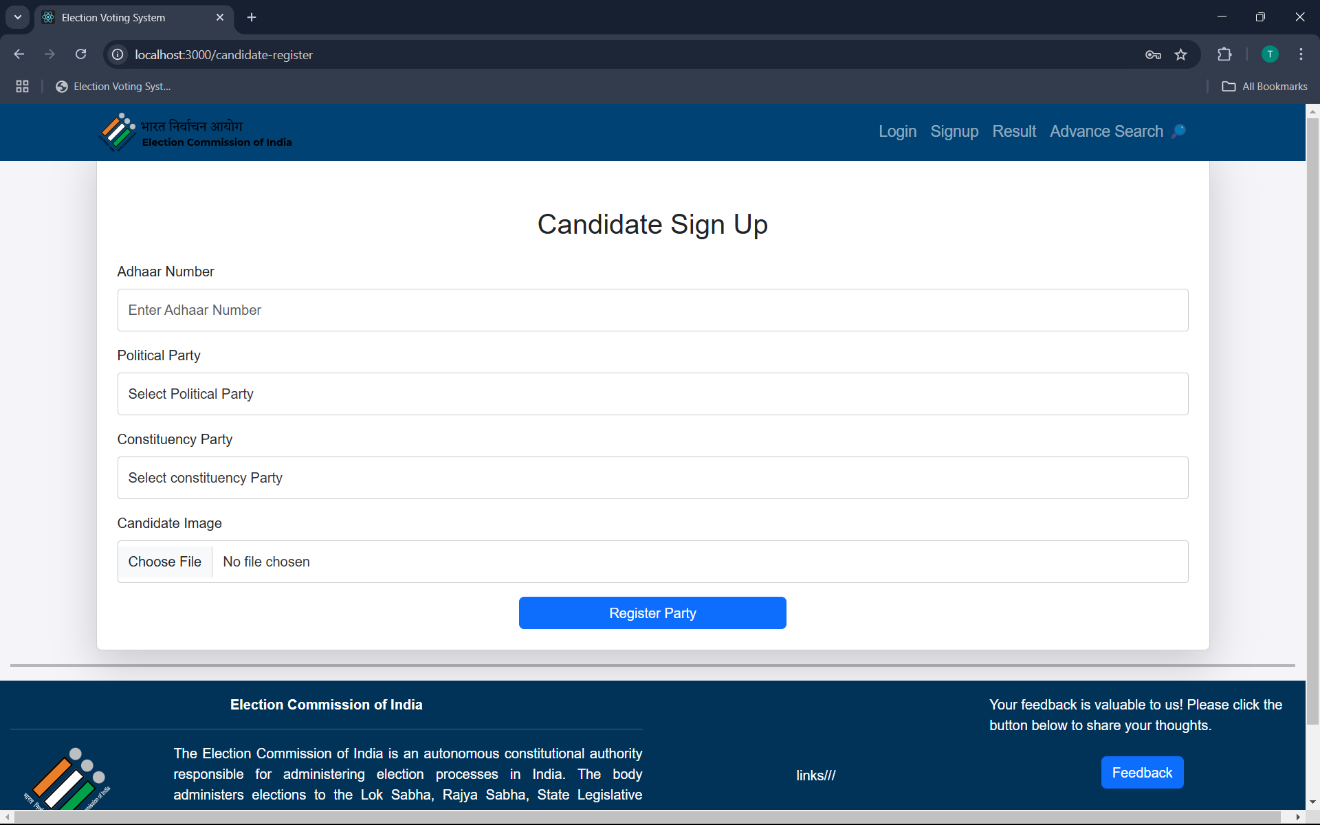
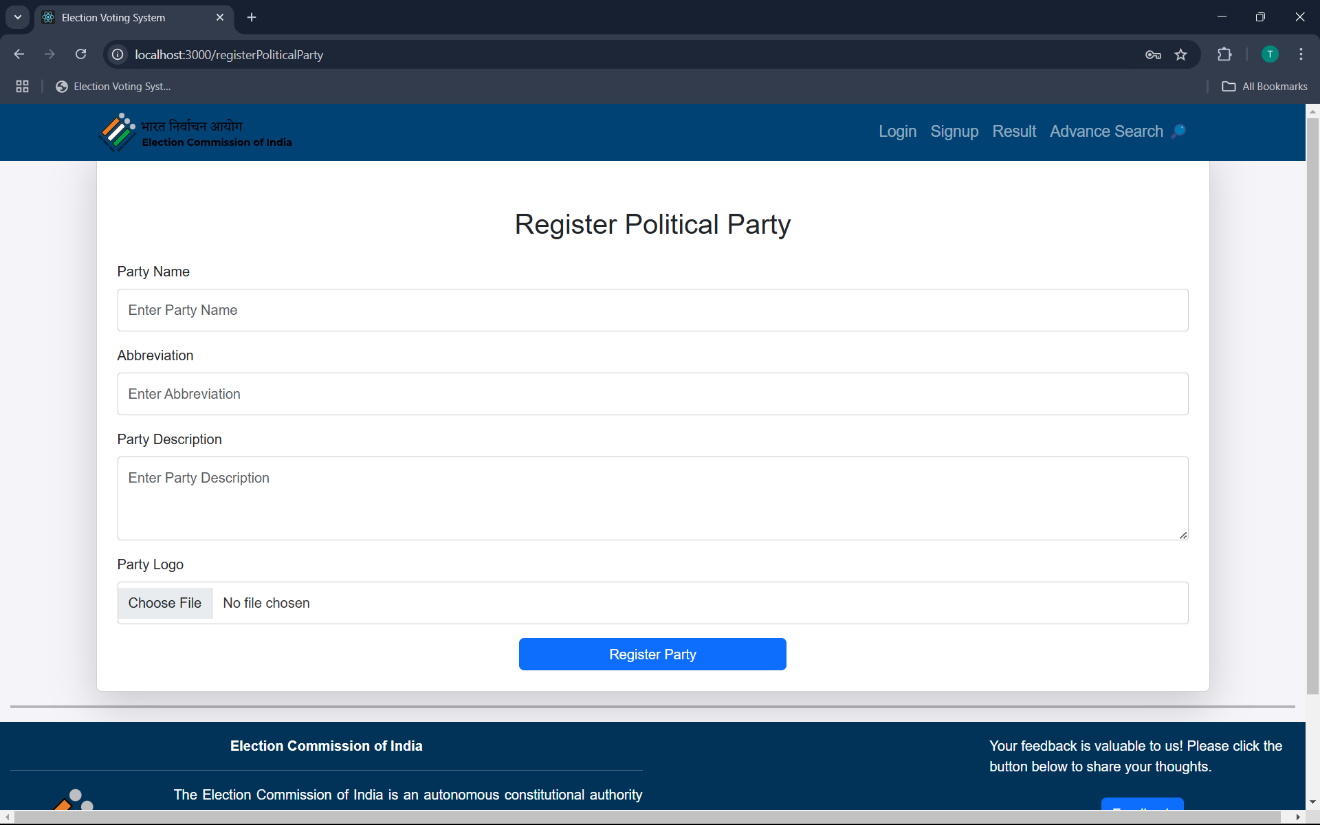
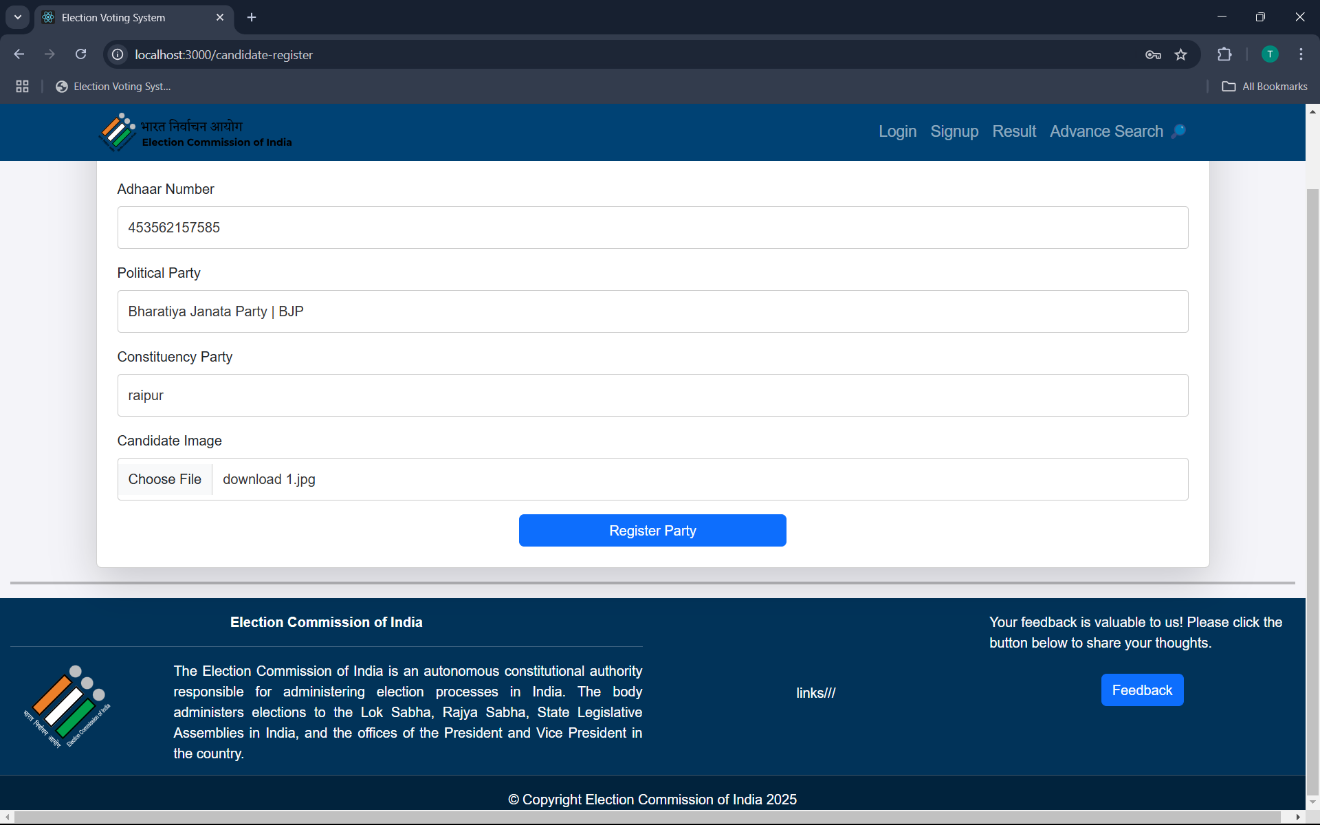
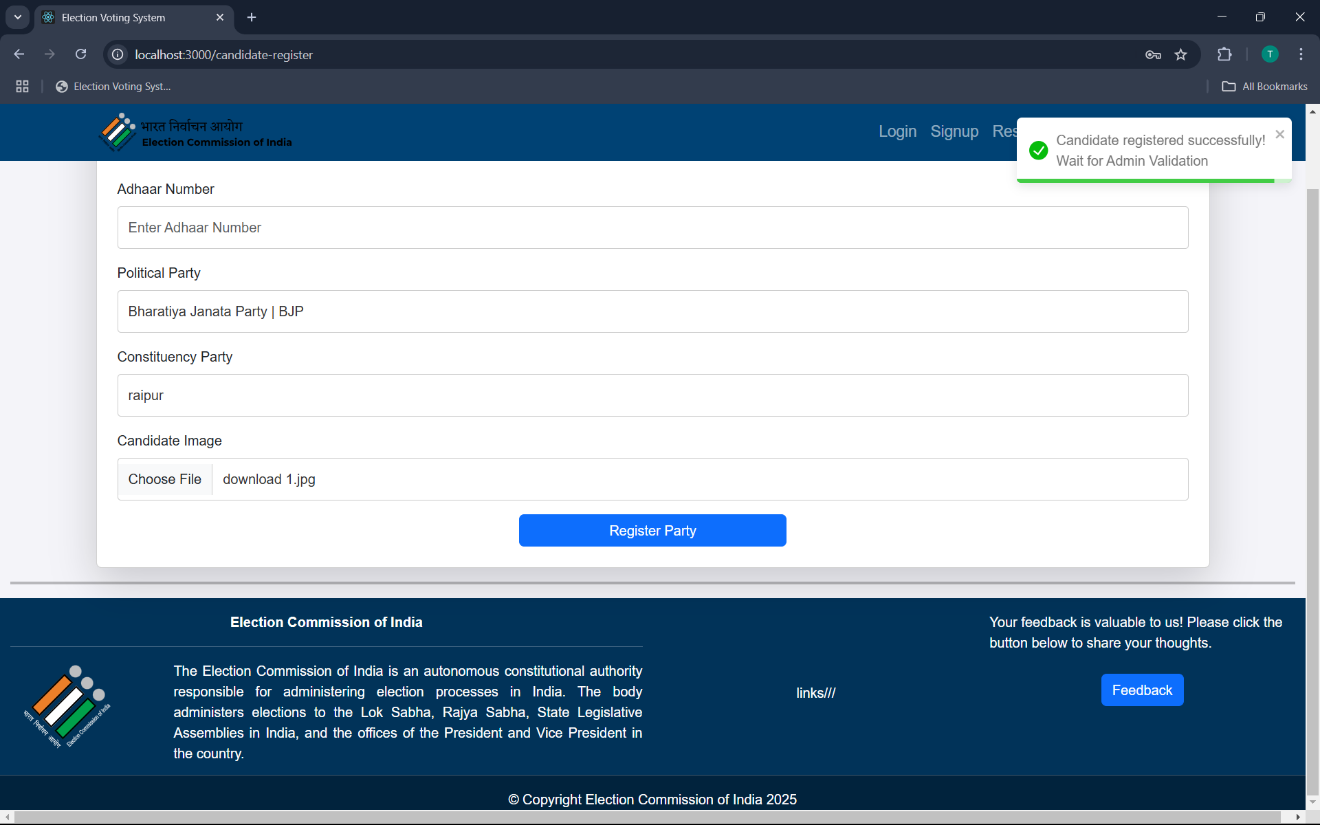
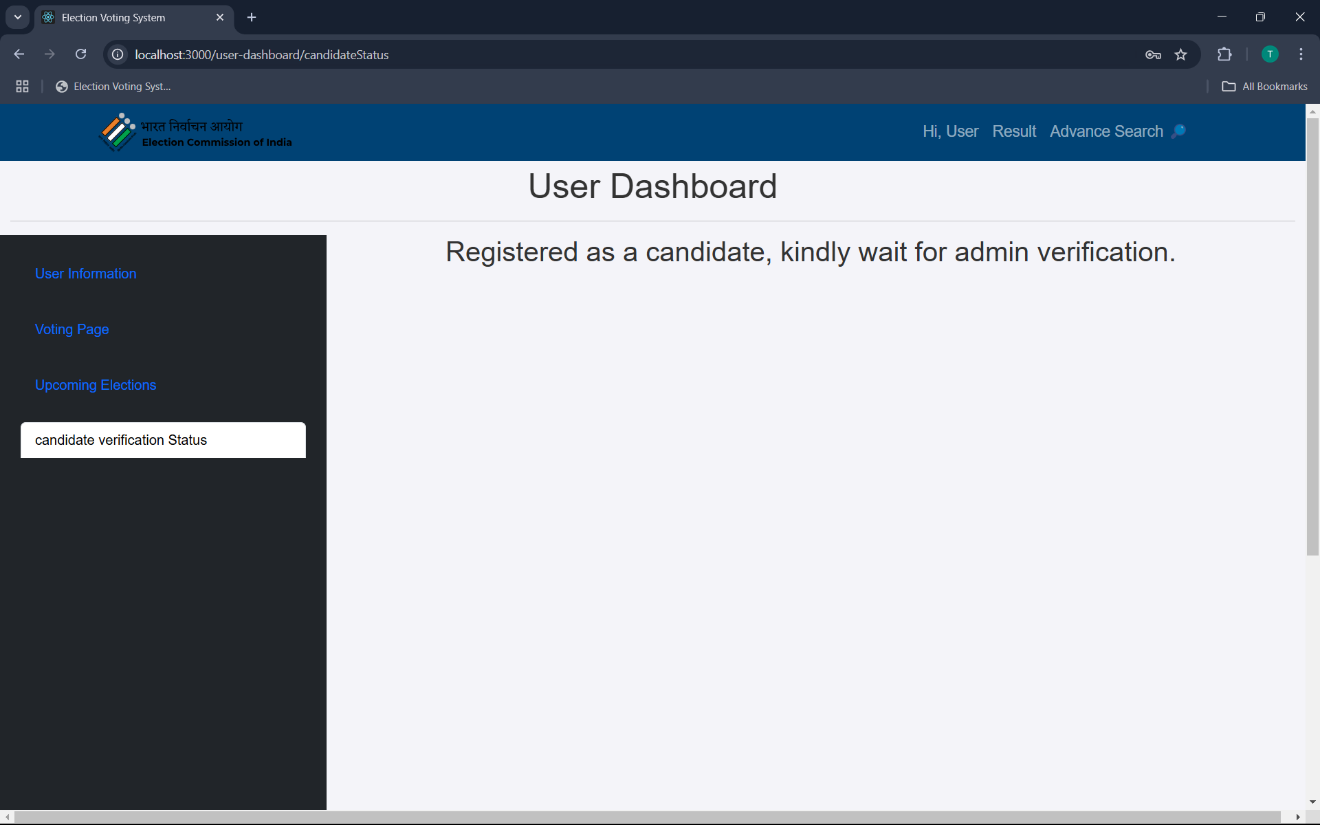
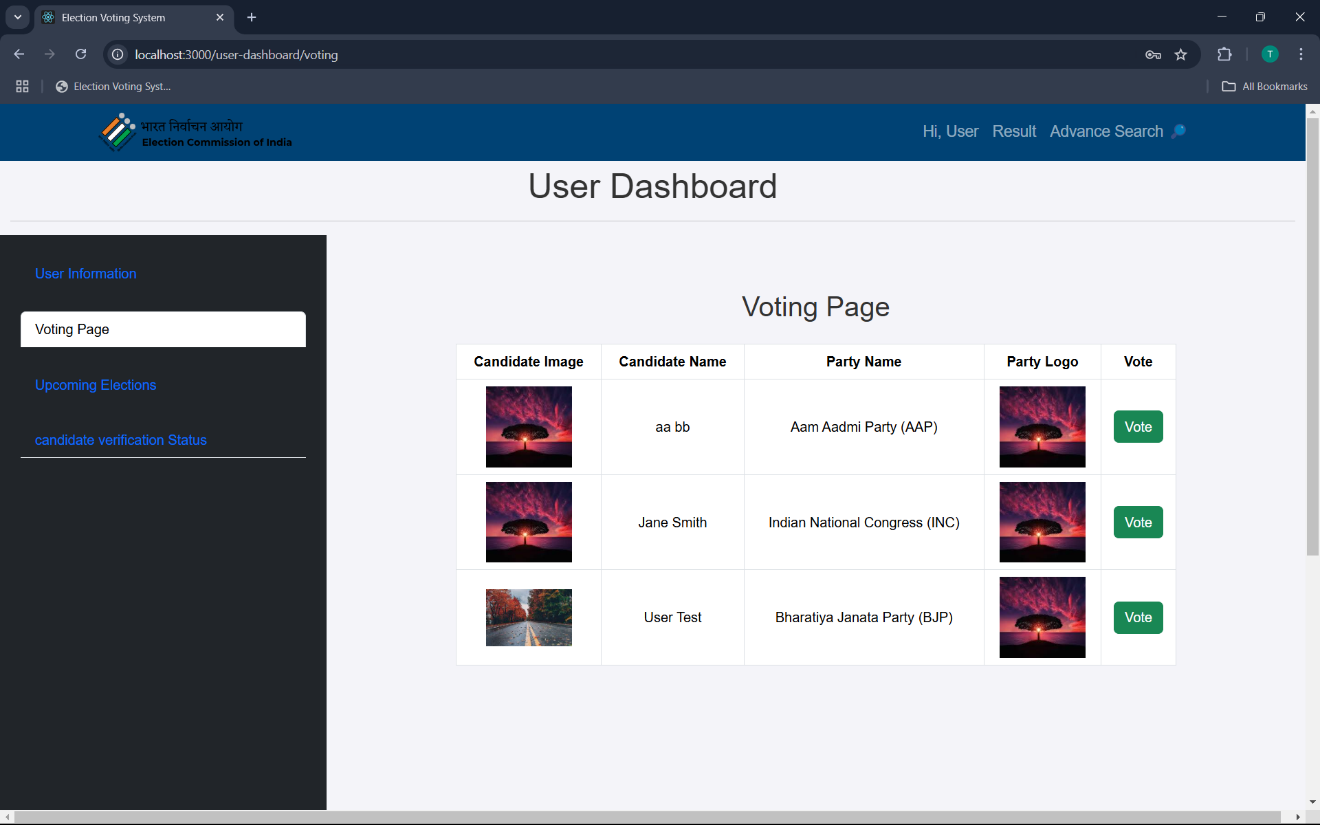
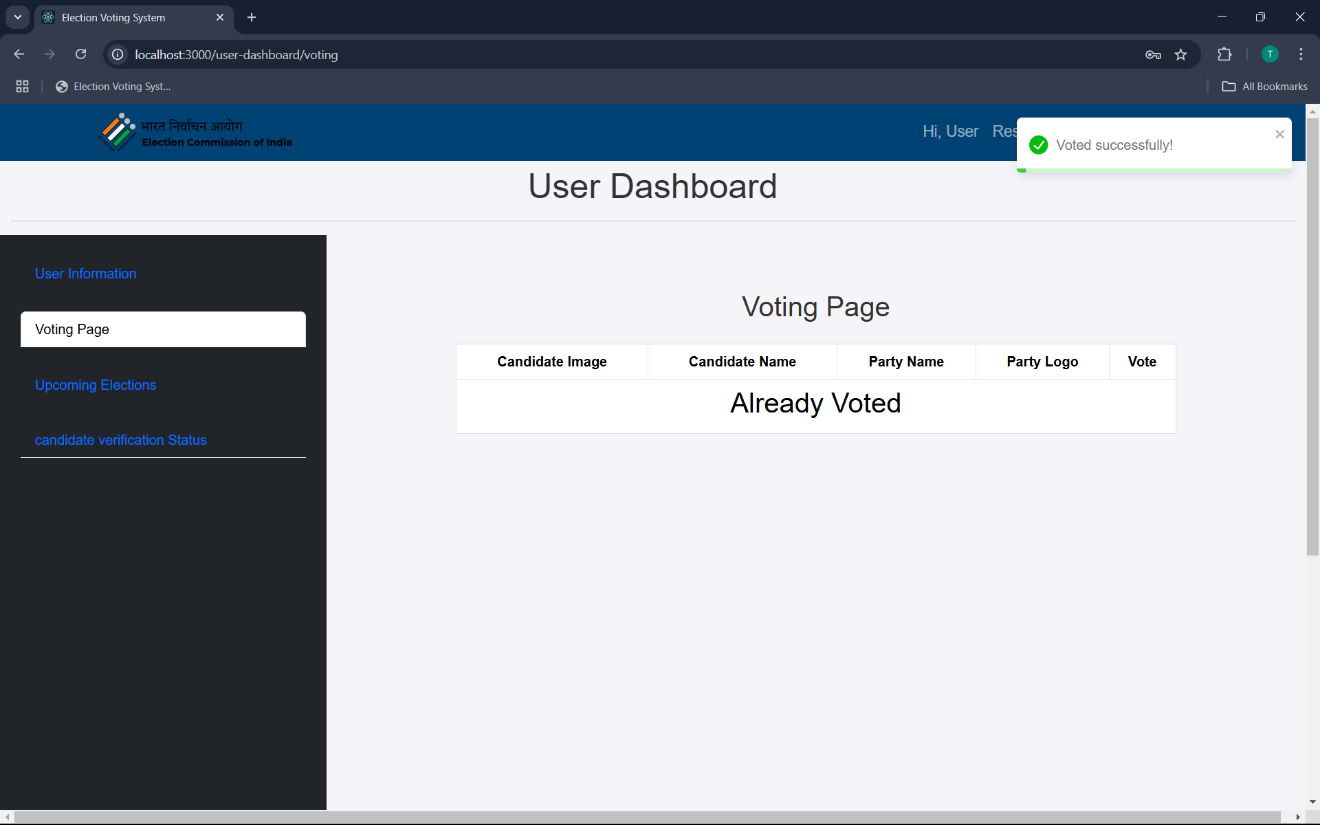
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique feedback ID |
| email | VARCHAR(255) | NOT NULL | User's email |
| full\_name | VARCHAR(255) | NOT NULL | User's full name |
| title | VARCHAR(500) | NOT NULL | Feedback title |
| description | VARCHAR(1000) | NOT NULL | Feedback content |
| is\_reviewed | BOOLEAN | DEFAULT FALSE | Whether feedback is reviewed or not |
| deleted | BOOLEAN | DEFAULT FALSE | Whether feedback is deleted or not |
| created\_at | DATETIME | DEFAULT CURRENT\_TIMESTAMP | When the feedback was submitted |

## Political Parties Table

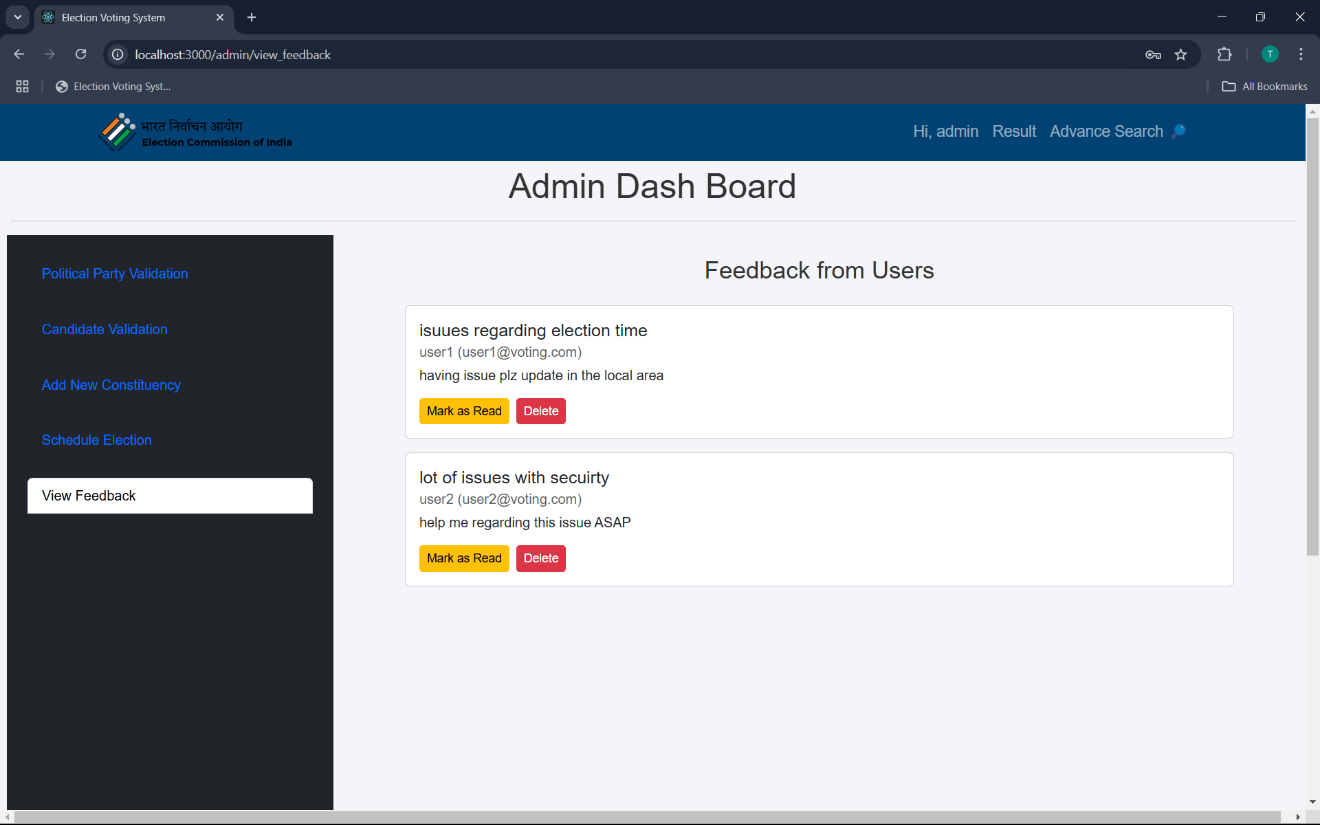
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Constraints | Description |
| party\_id | INT | PRIMARY KEY, AUTO\_INCREMENT | Unique ID for political party |
| party\_name | VARCHAR(255) | UNIQUE, NOT NULL | Name of the political party |
| abbreviation | VARCHAR(6) | UNIQUE, NOT NULL | Party abbreviation (e.g., BJP, INC) |
| party\_description | VARCHAR(500) | NULL | Description of the party |
| is\_valid | INT | DEFAULT 0 | Party validation status |
| party\_logo | LONGBLOB | NULL | Party logo file |
| logo\_path | VARCHAR(1024) | NULL | Path to party logo |

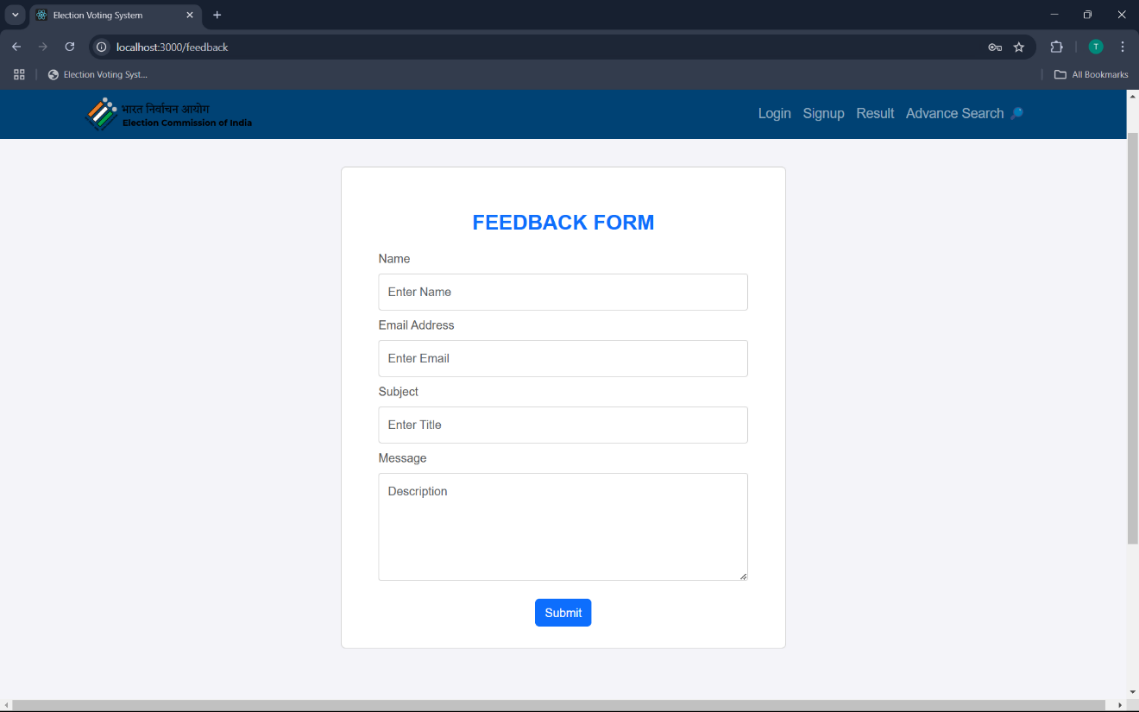
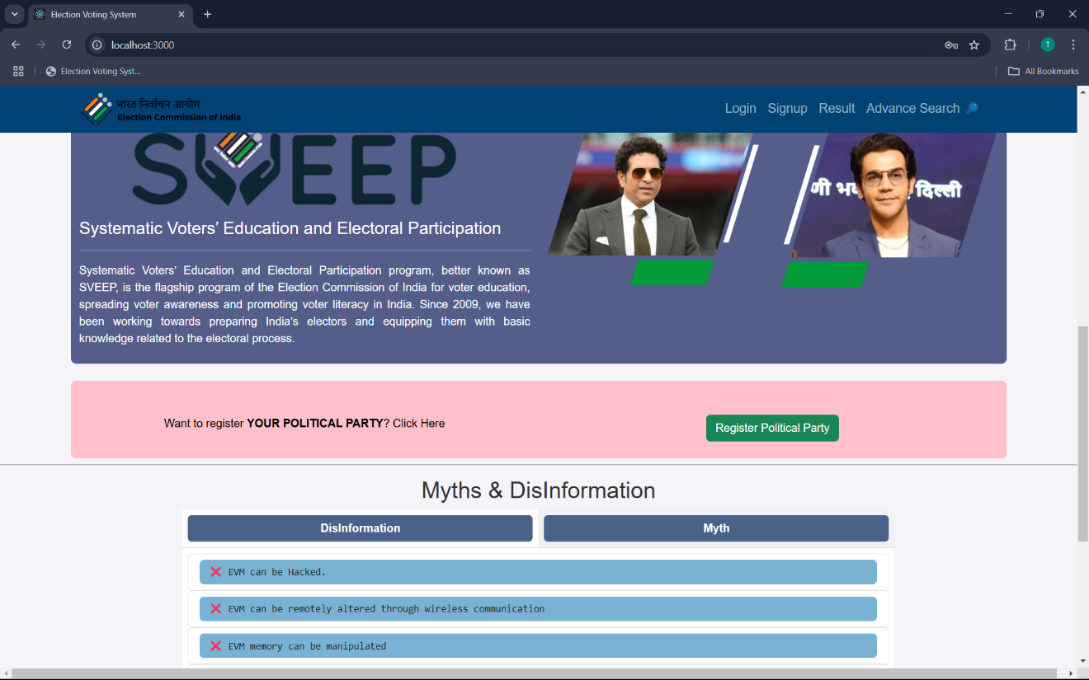
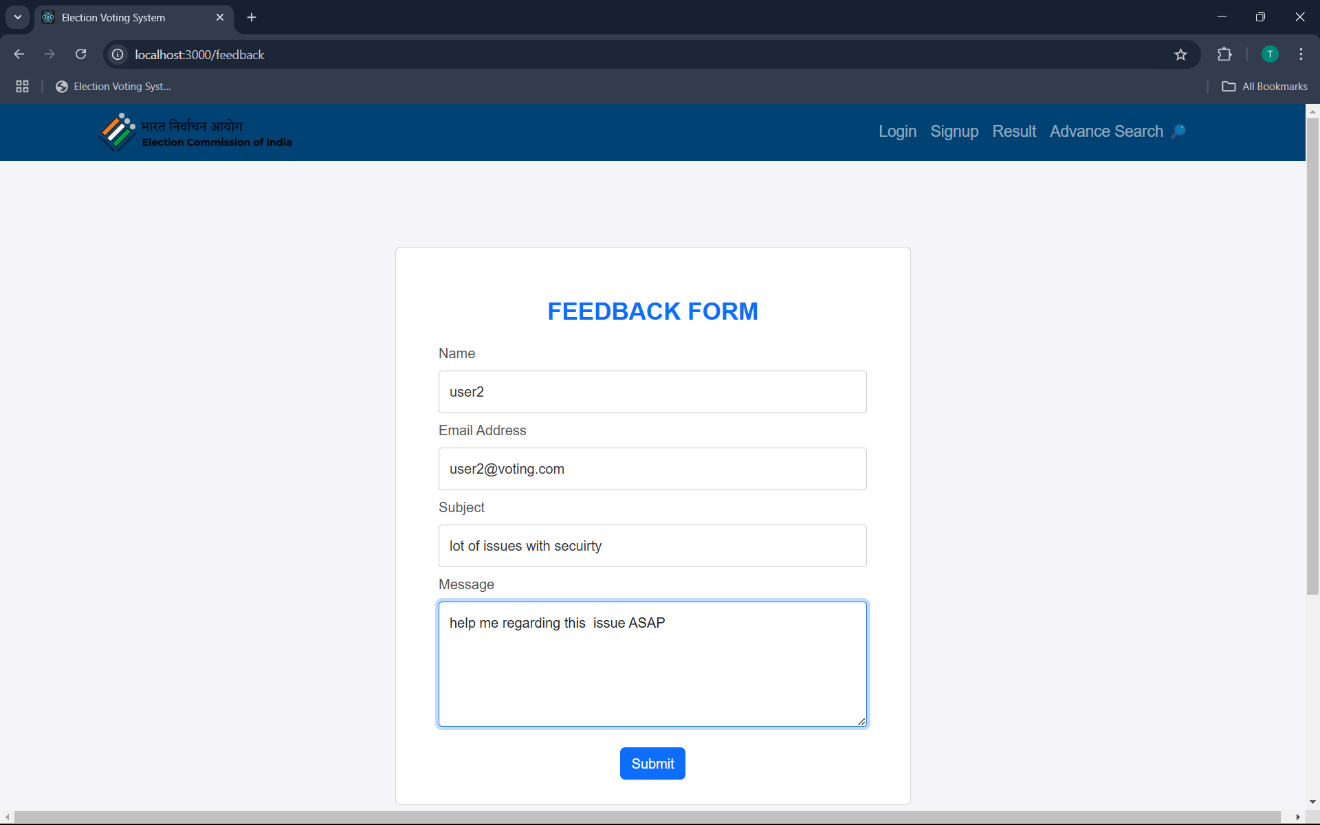
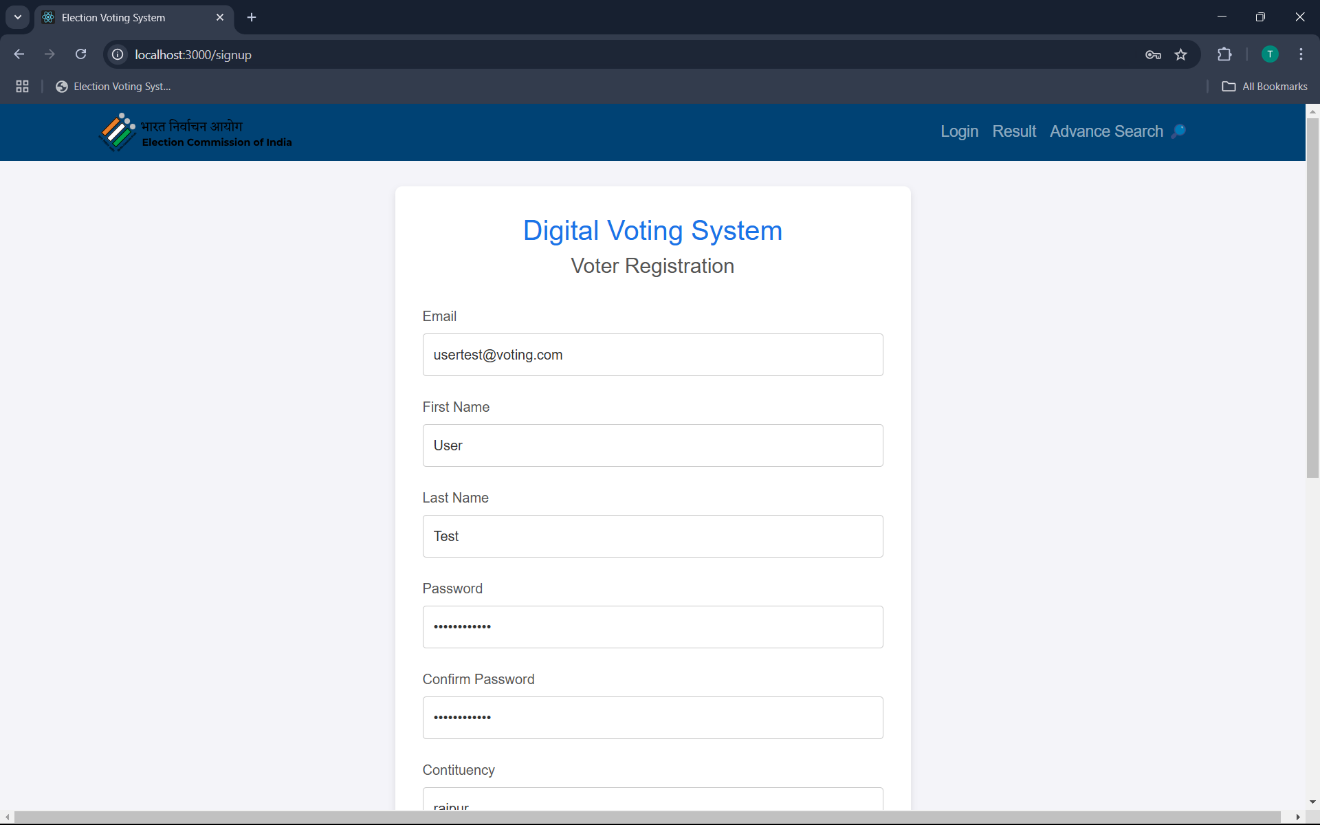
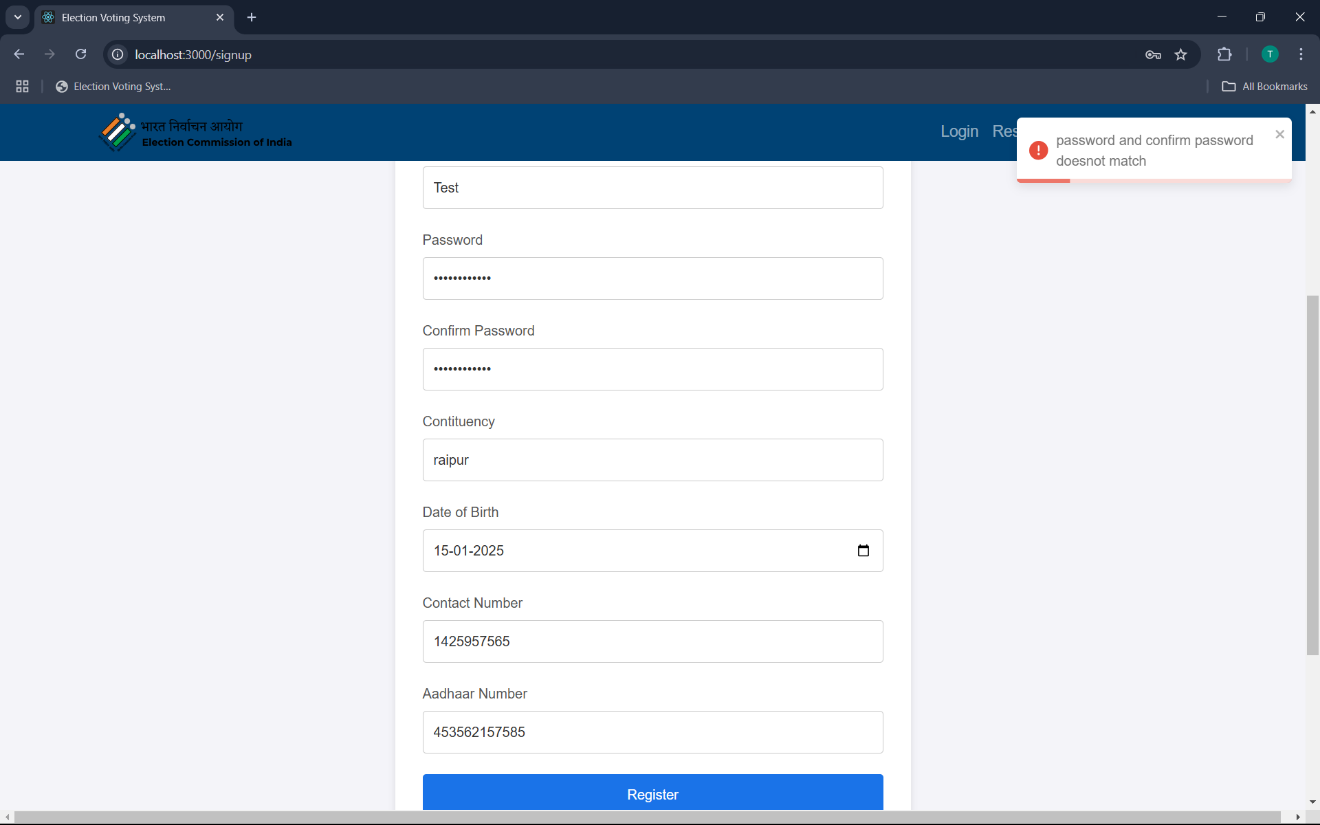
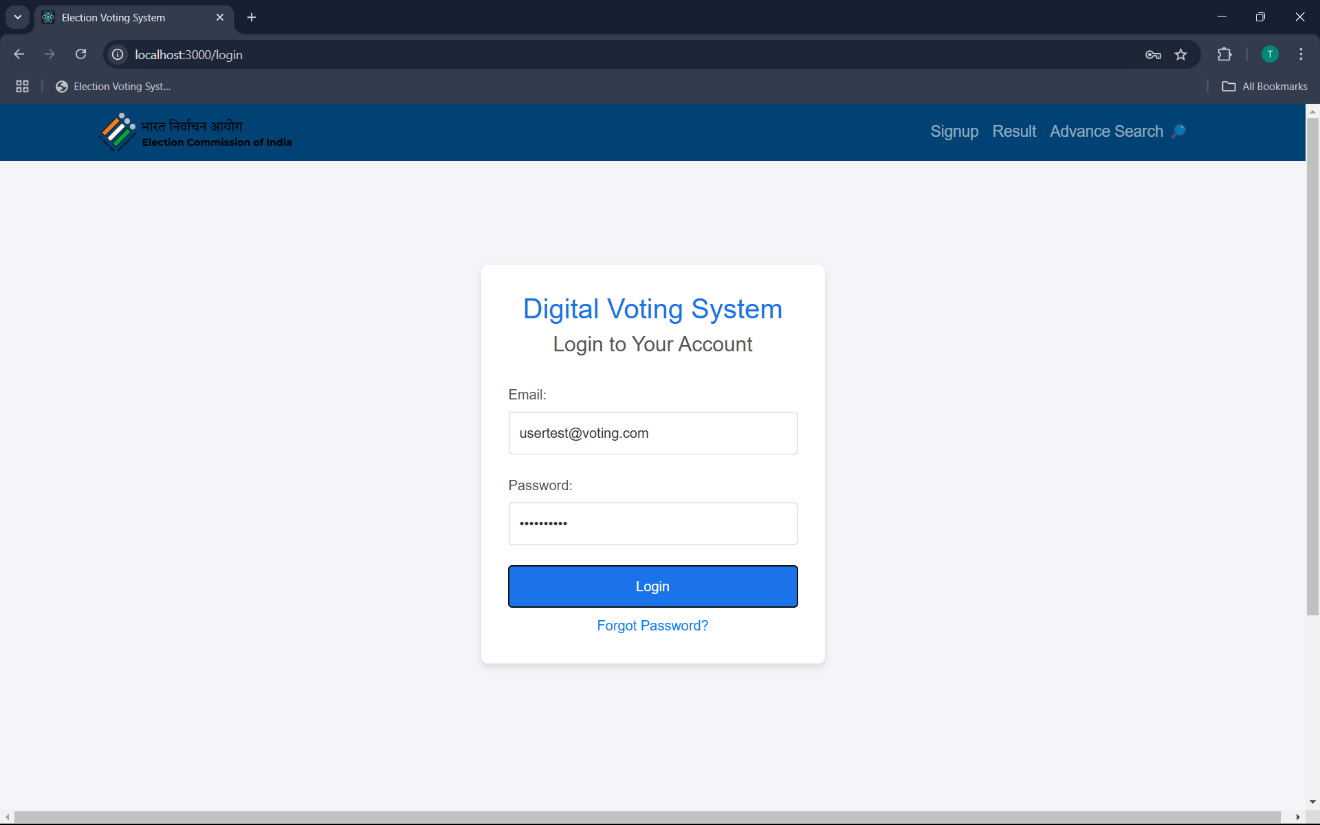
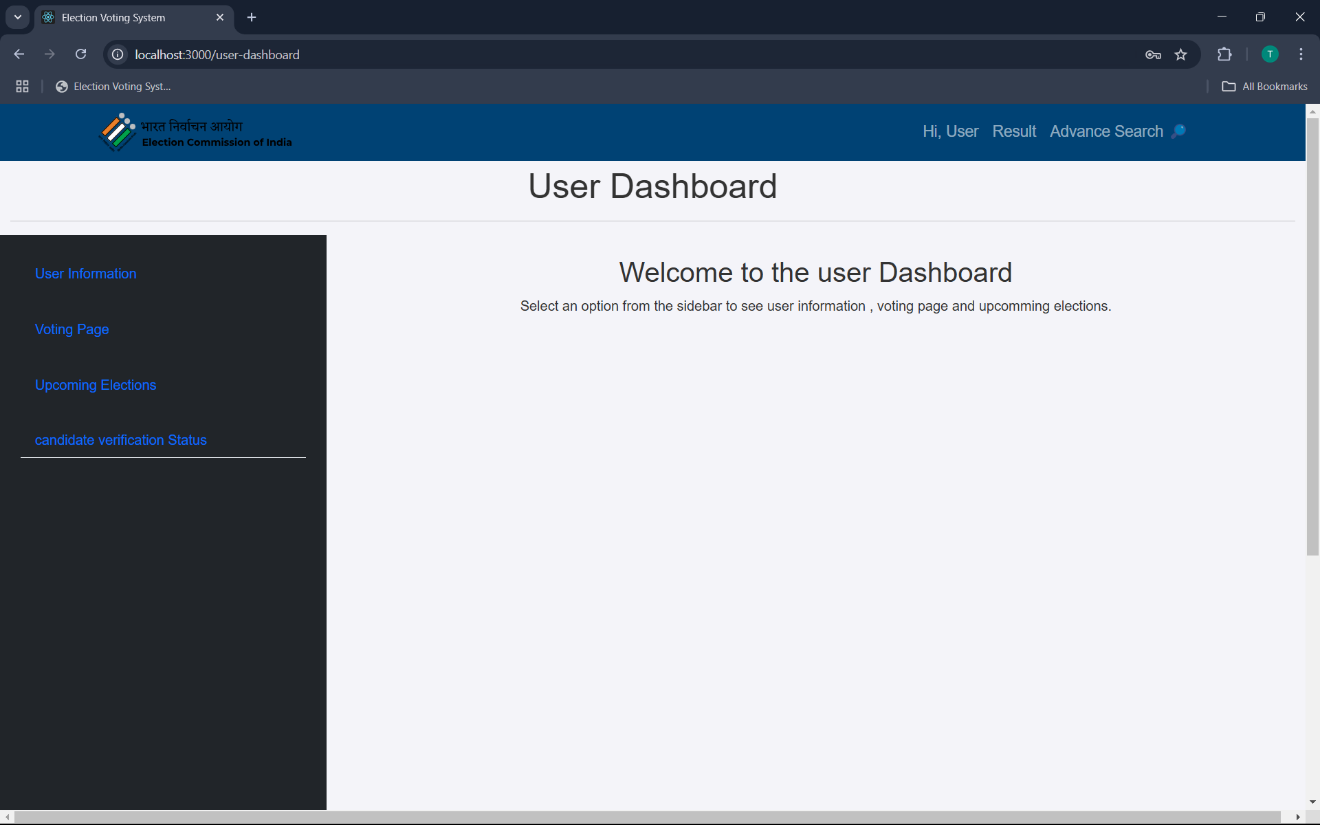
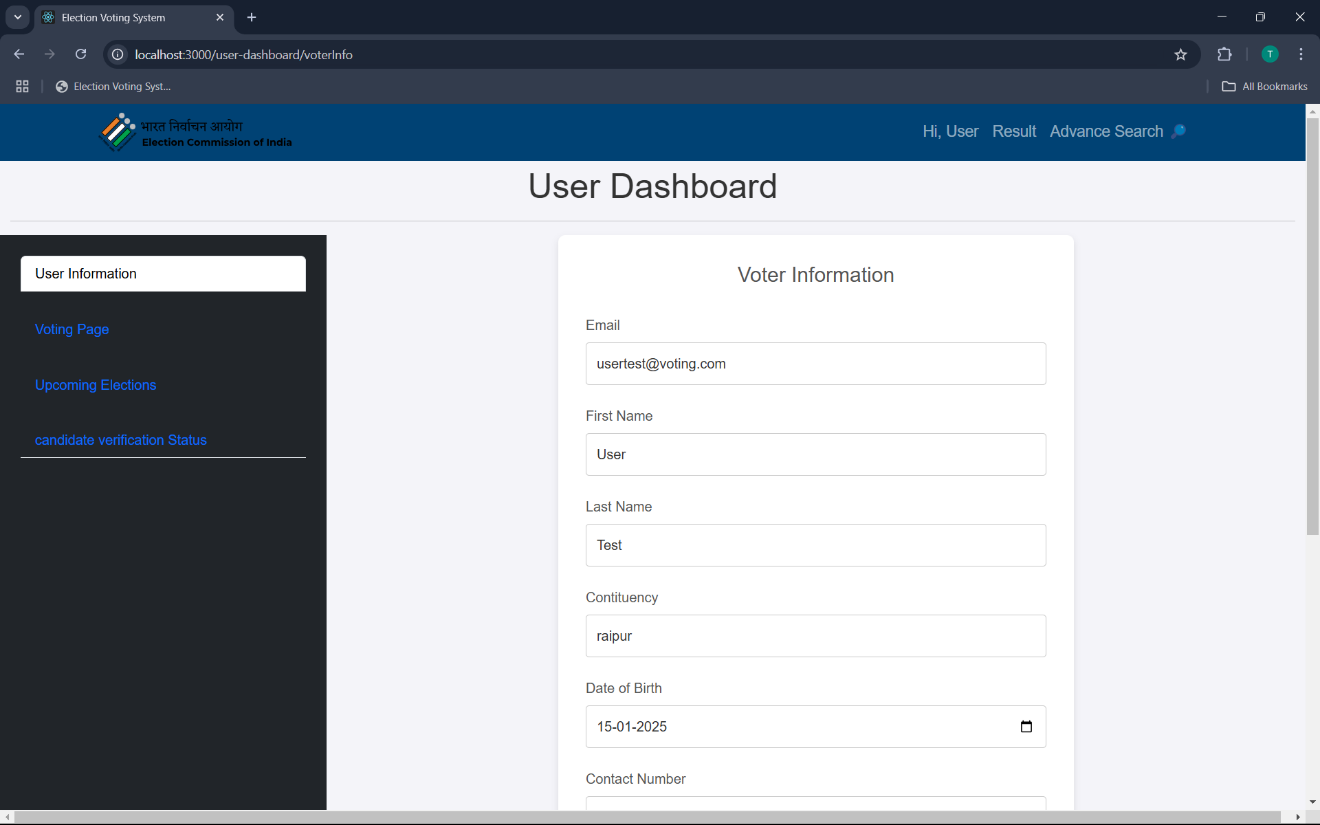
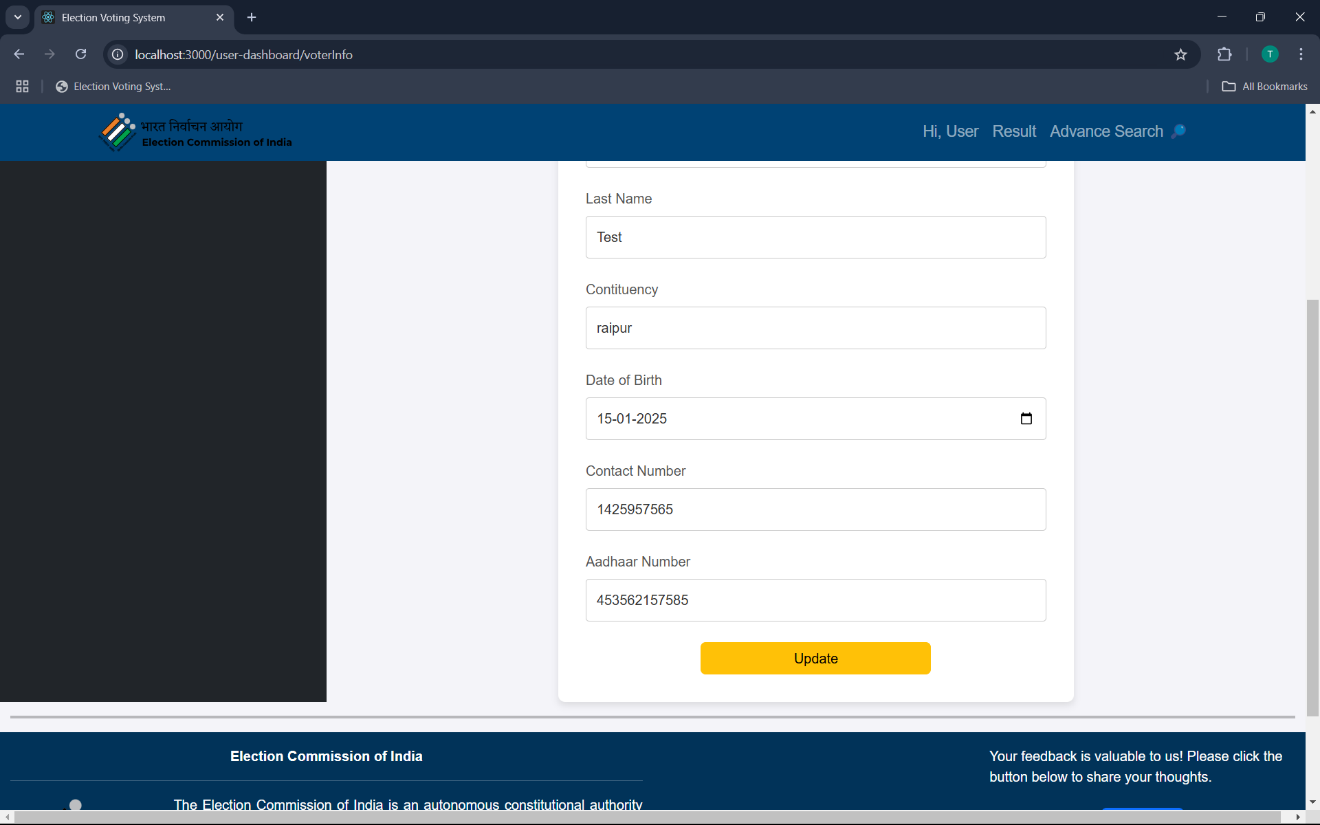
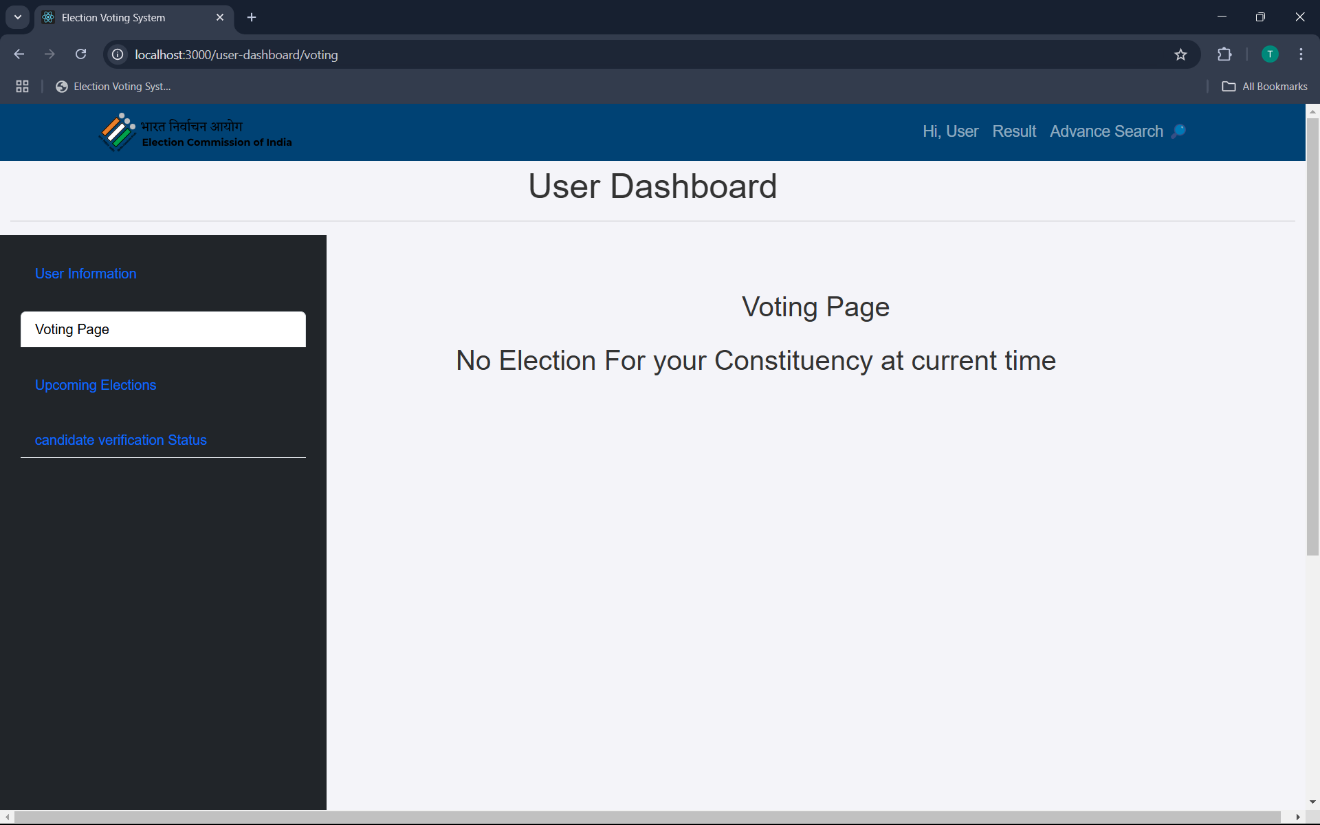
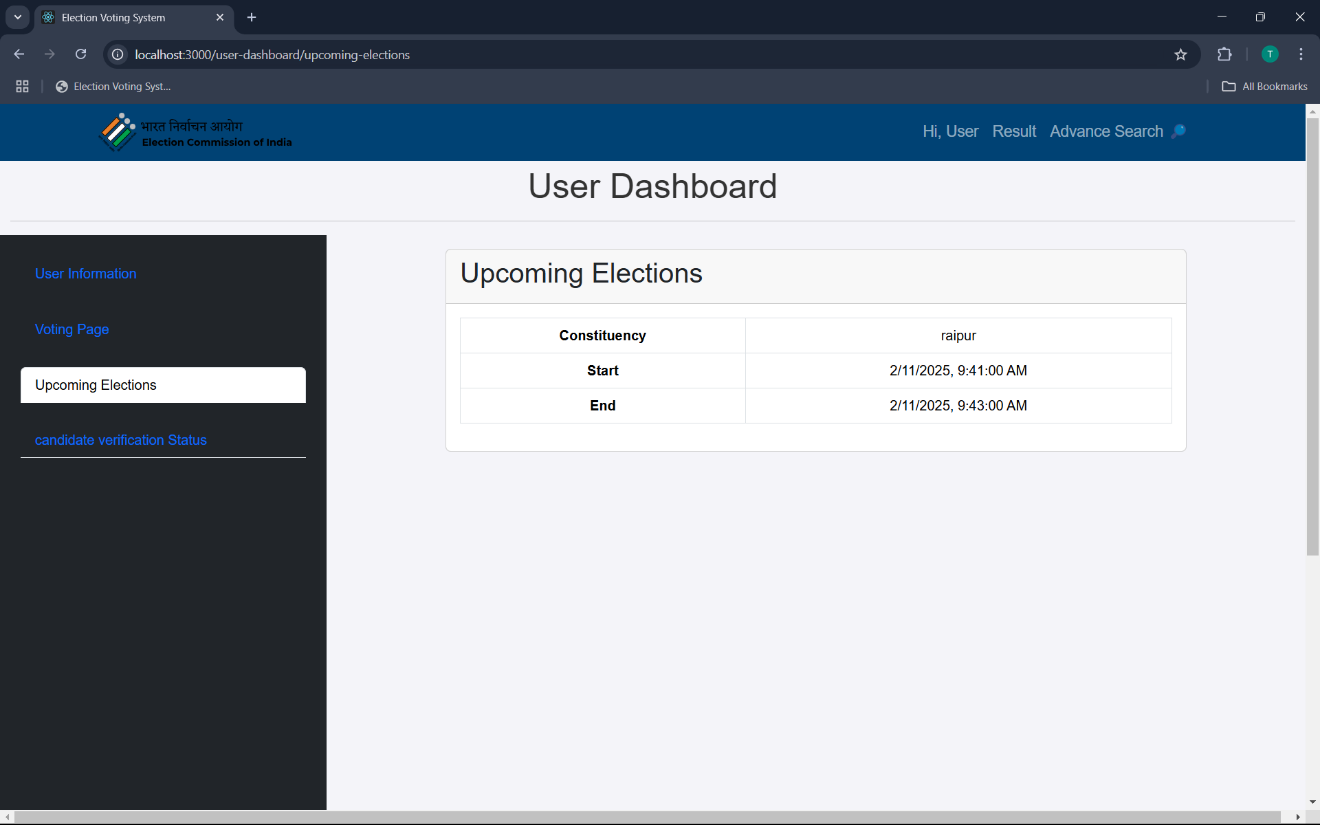
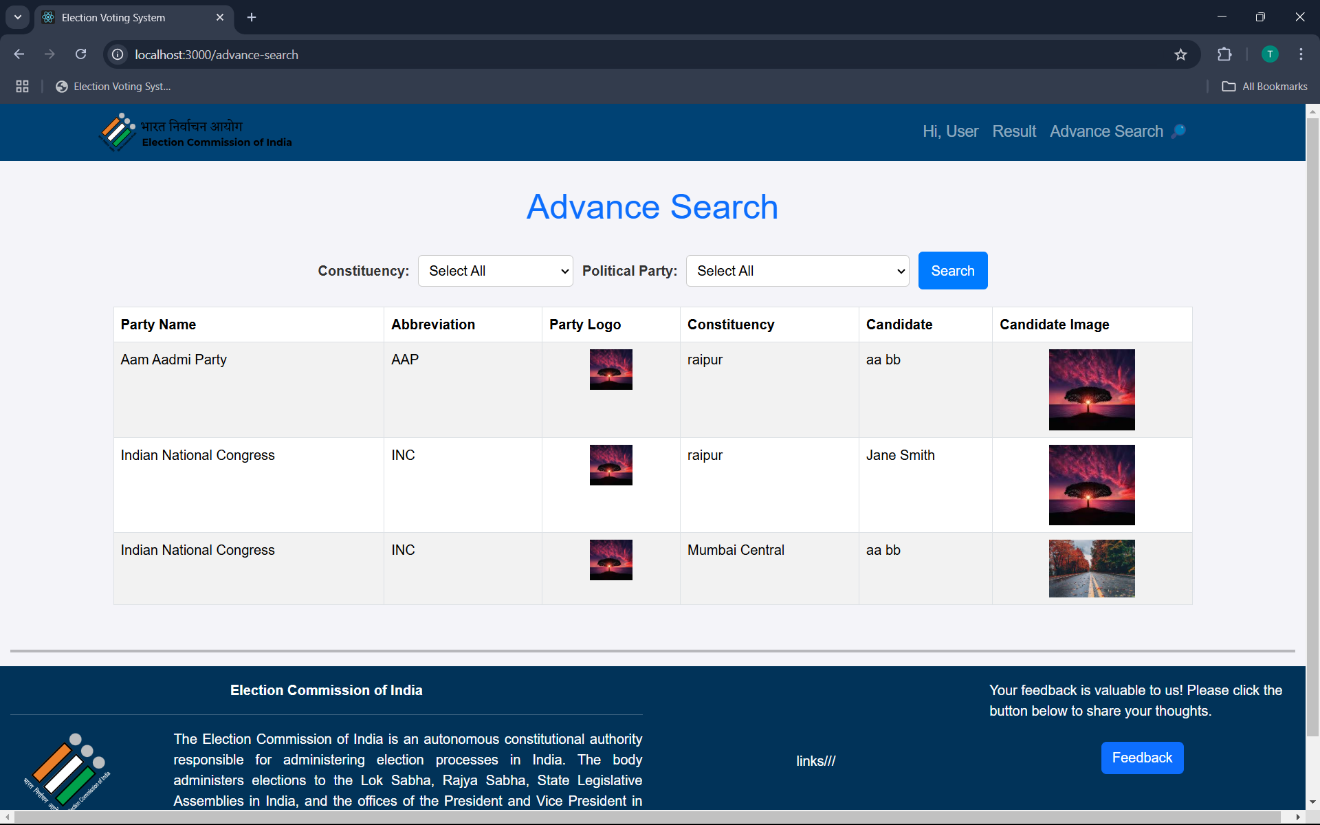
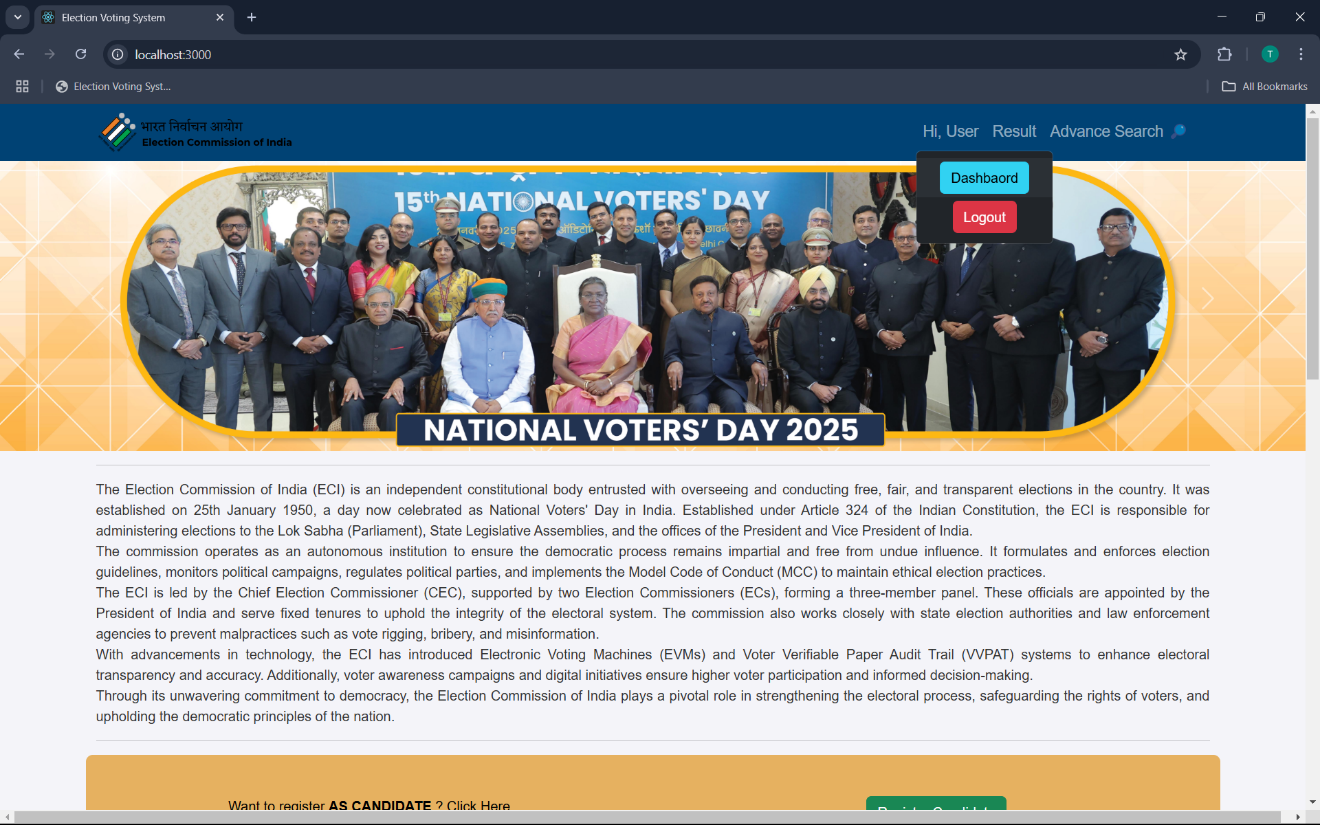
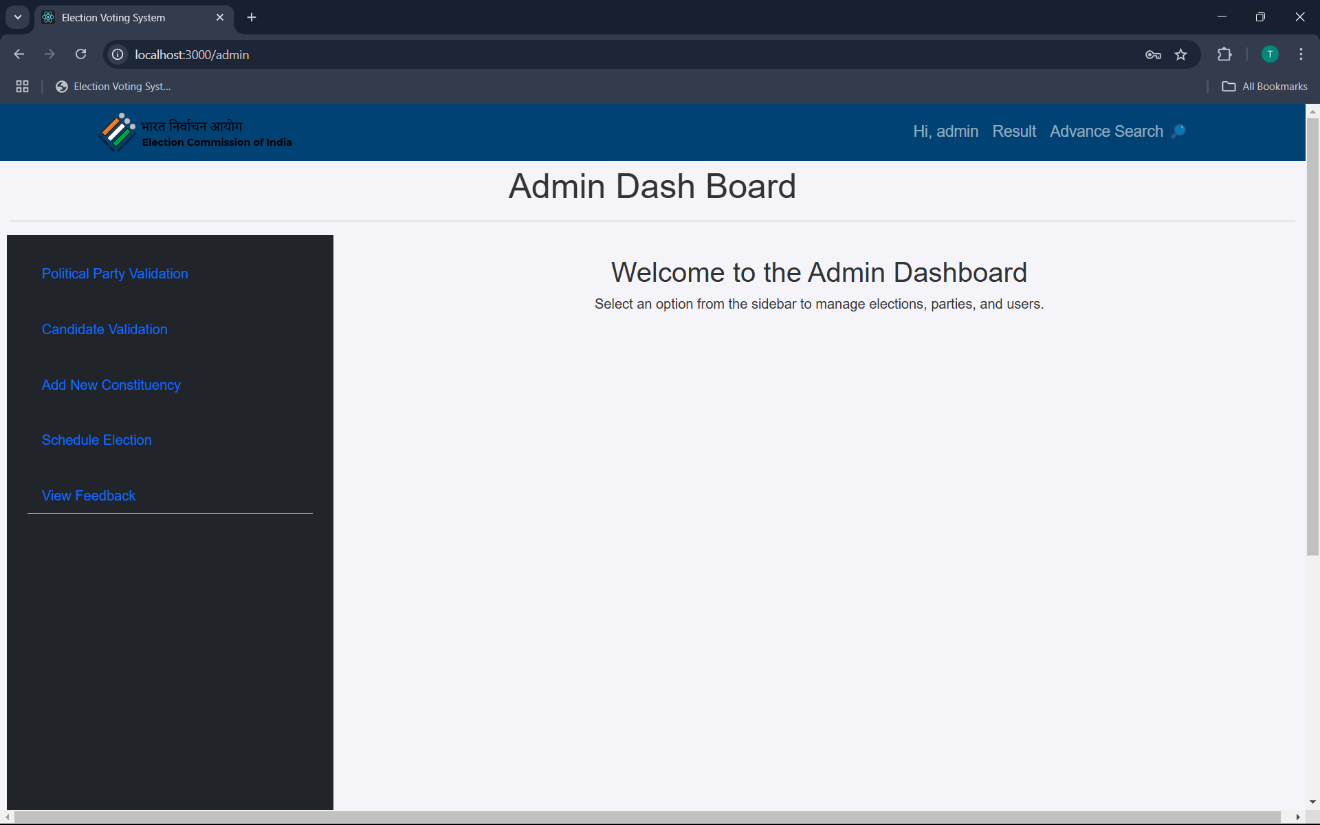
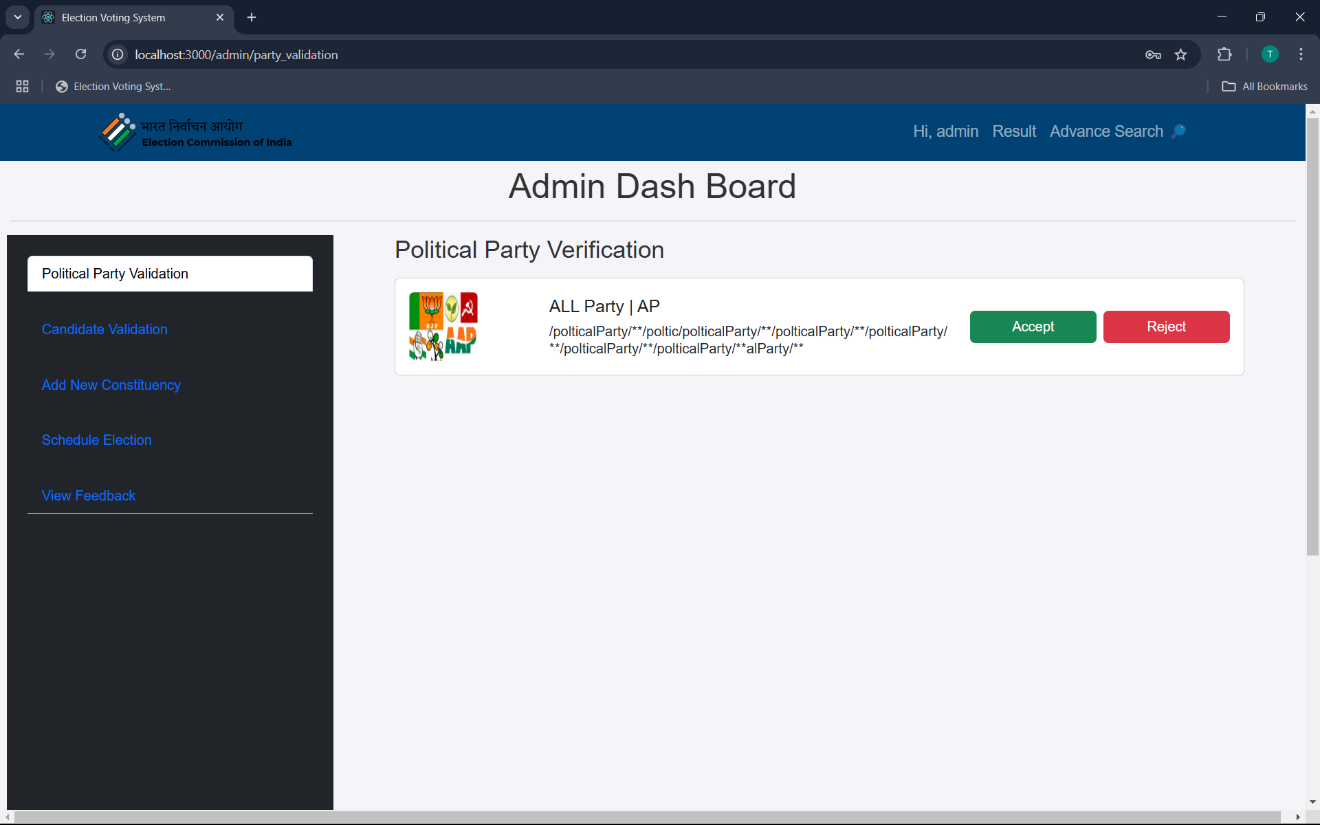
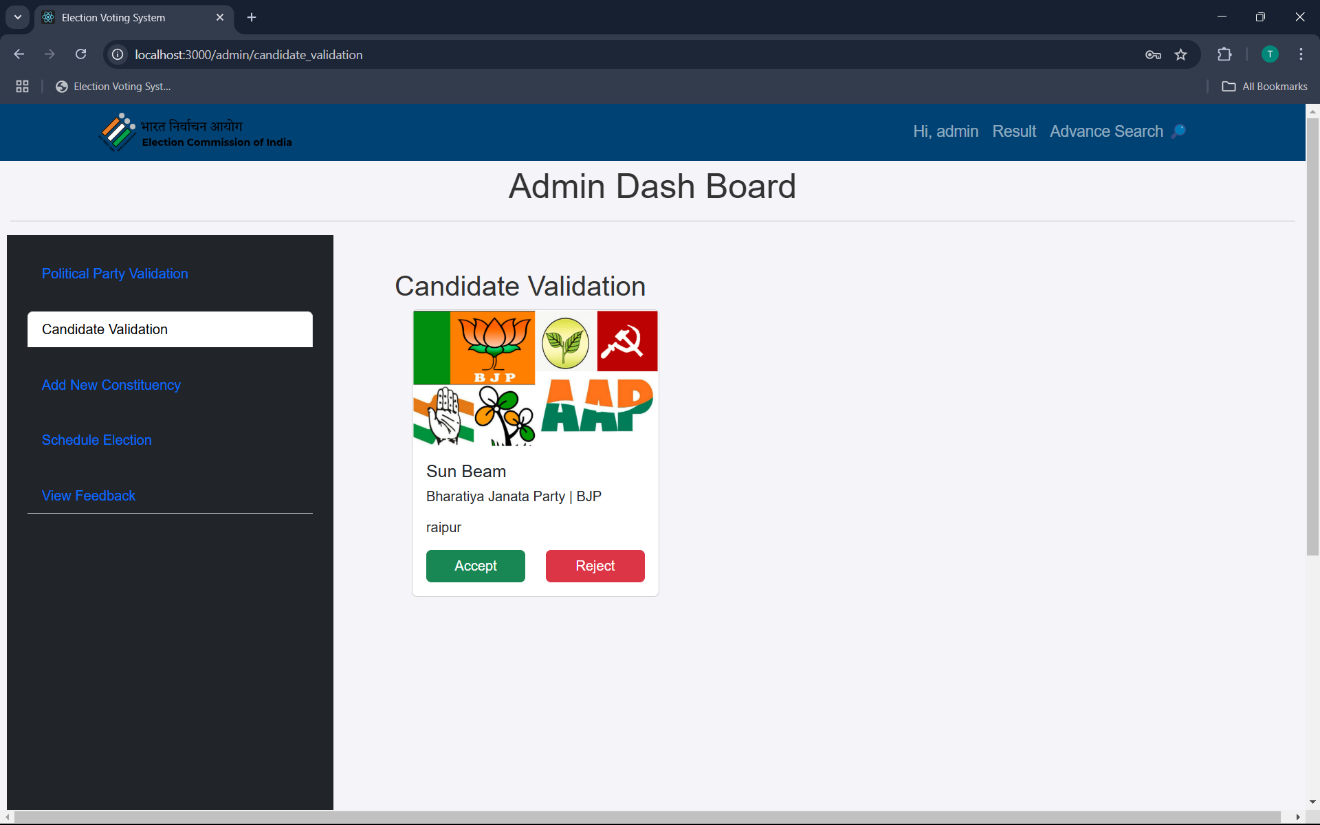
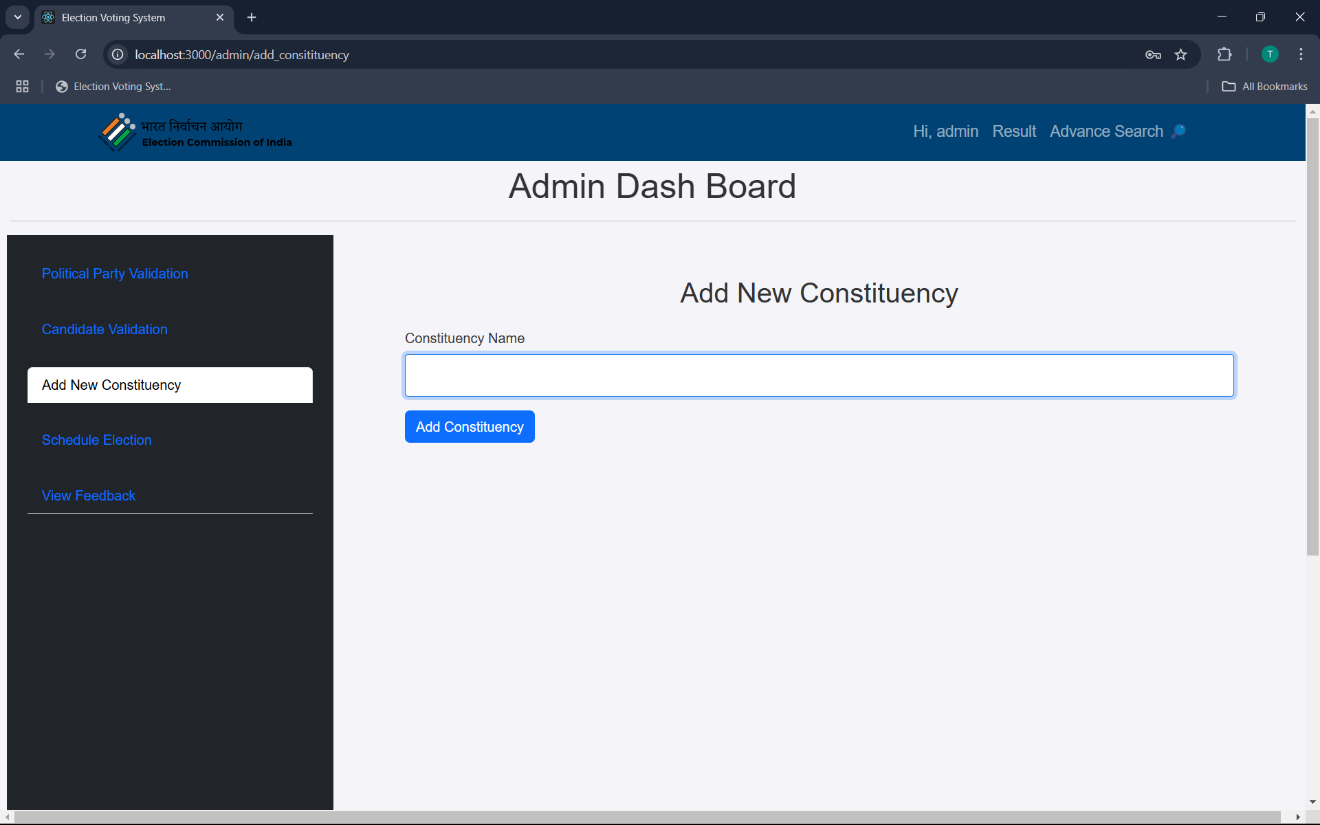
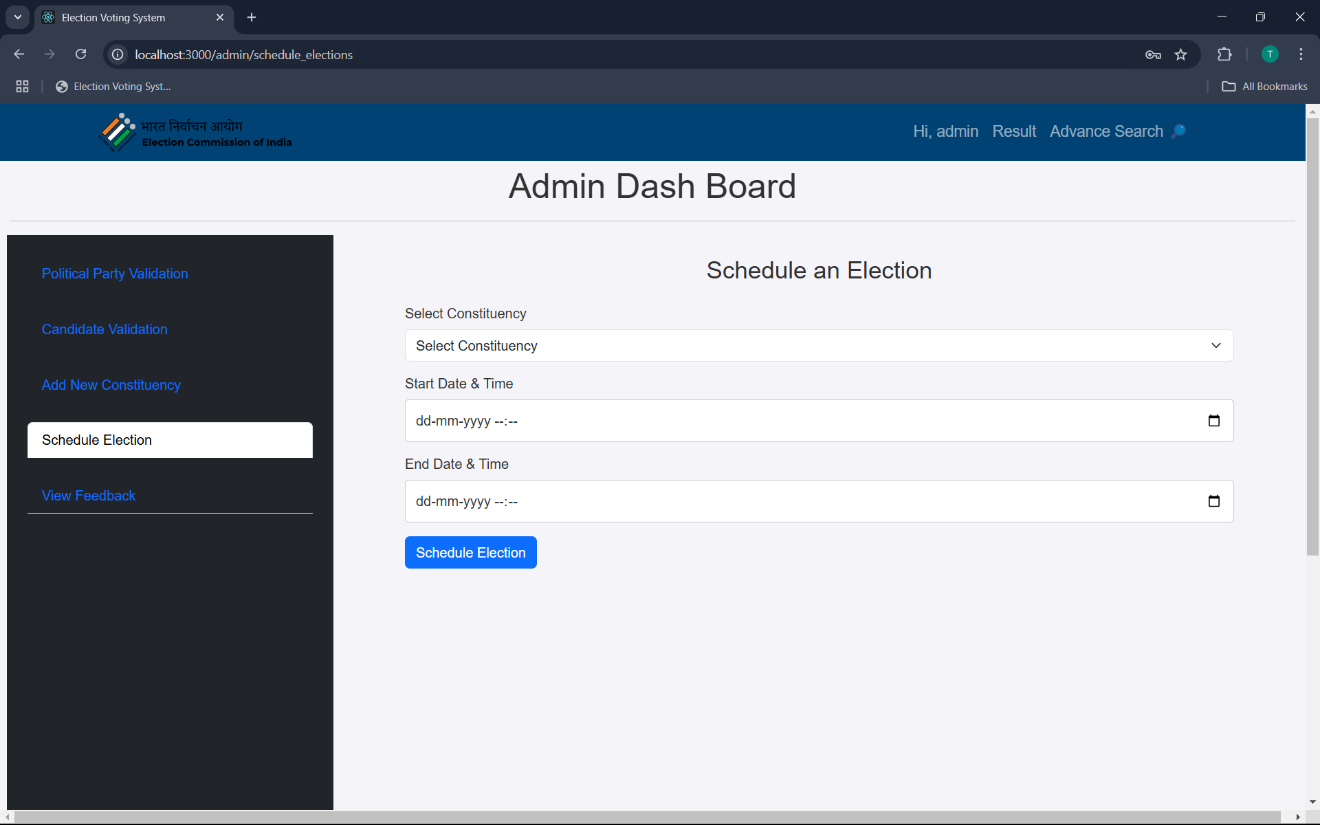


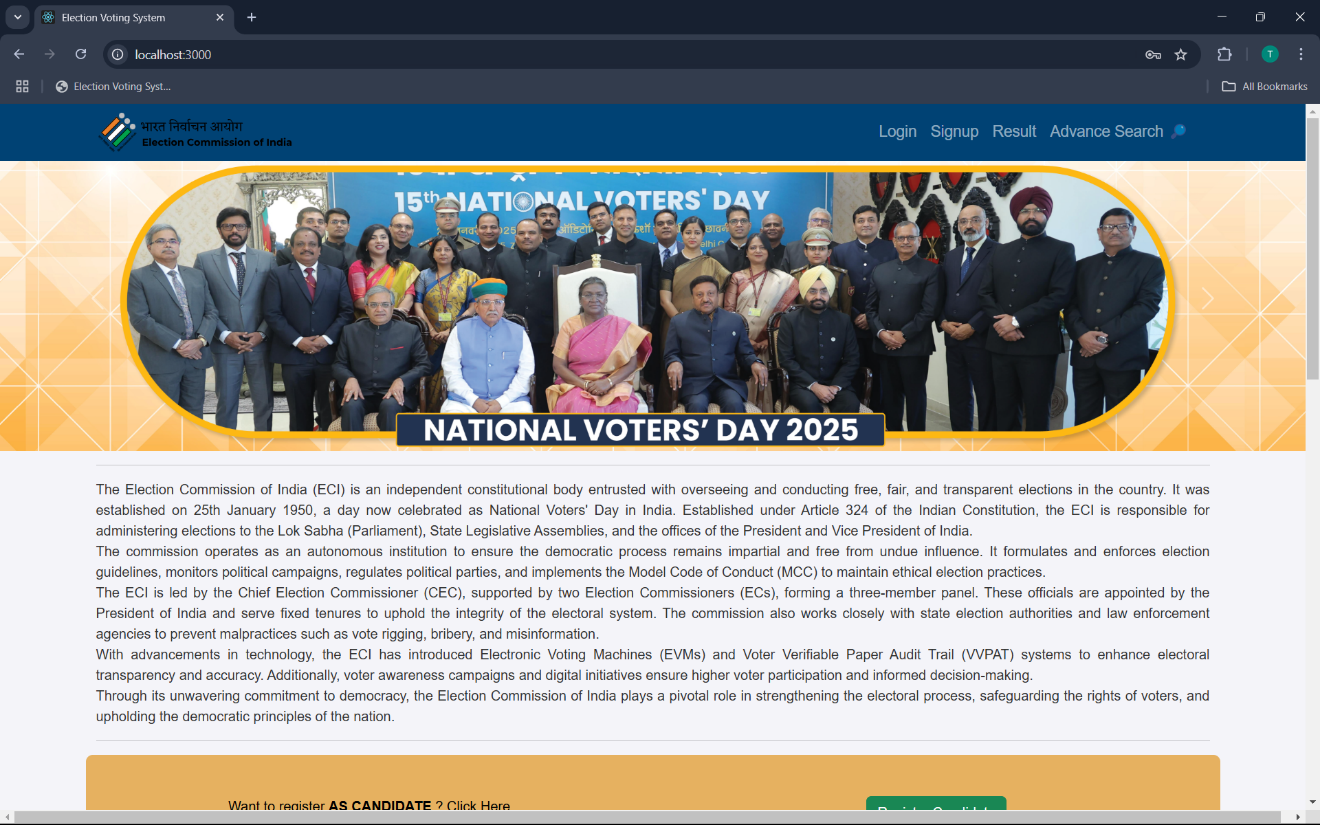
**User dashboard**



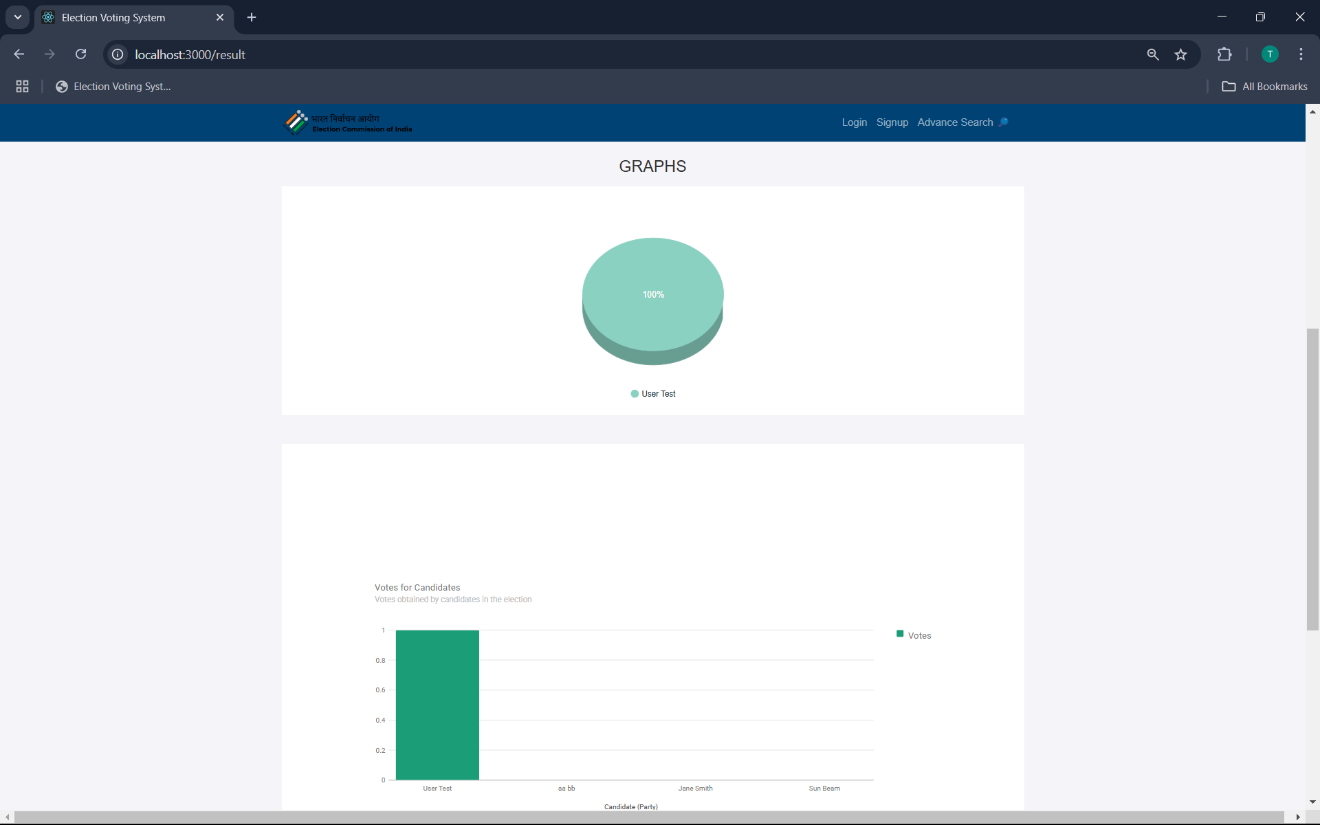
**Admin dashboard**







**Fig. Home page**



**CONCLUSION**

The Digital Voting System project successfully addresses the challenges associated with traditional voting methods by providing a secure, transparent, and efficient online voting platform. This system ensures that only eligible voters can cast their votes, prevents duplicate voting, and maintains the integrity of the electoral process.

With role-based access for admins, voters, and candidates, the system ensures smooth election management, secure vote casting, and real-time vote counting. The use of authentication mechanisms, database encryption, and security techniques makes the system reliable and tamper-proof.

By implementing this system using React for the frontend and Java for the backend, we have developed a modern, scalable, and user-friendly application that can be adapted for various election scenarios. This project demonstrates how technology can be leveraged to improve democracy and voting accessibility while maintaining transparency and security.

Overall, the Digital Voting System is a step forward in modernizing elections and can serve as a foundation for future enhancements, including blockchain-based voting, biometric authentication, and AI-driven voter fraud detection