



**SRM**  
UNIVERSITY AP  
Andhra Pradesh

# INTER-UNIVERSITY VOTING SYSTEM

Software Engineering

GROUP-12

CSE -306

# INTRODUCTION

Voting is a key part of promoting democracy and leadership in colleges and universities.

- Voting strengthens democracy and leadership in colleges.
- Traditional manual systems face errors, security risks, and delays.
- To address these, a secure, digital, and transparent solution is essential.
- The Inter University Voting System offers a web platform for secure registration, voting, and real-time results — ensuring fairness and efficiency.

# PROBLEM STATEMENT

Manual voting in colleges leads to errors, delays, security issues, and lack of transparency.

A secure, digital voting system is needed to automate elections, ensure fairness, improve participation, and deliver real-time results.

# PROPOSED SOLUTION

- **Secure Digital Voting:**

Encrypted authentication and one-person-one-vote validation prevent tampering and fraud.

- **Real-Time Results and Transparency:**

Instant, automated result computation builds trust and eliminates manual errors or delays.

- **Accessible and User-Friendly Platform:**

Web-based interface allows easy voting from any device, improving participation and reducing logistical challenges.

# OBJECTIVE

- Develop a **secure and reliable online voting platform** for college elections.
- Ensure **transparent, tamper-proof voting** and result computation.
- Simplify the election process for students, candidates, and administrators.
- Enable **real-time access to voting** and results from any device.
- Promote **higher student participation** through an easy and accessible system.

# SYSTEM ANALYSIS

## Existing System

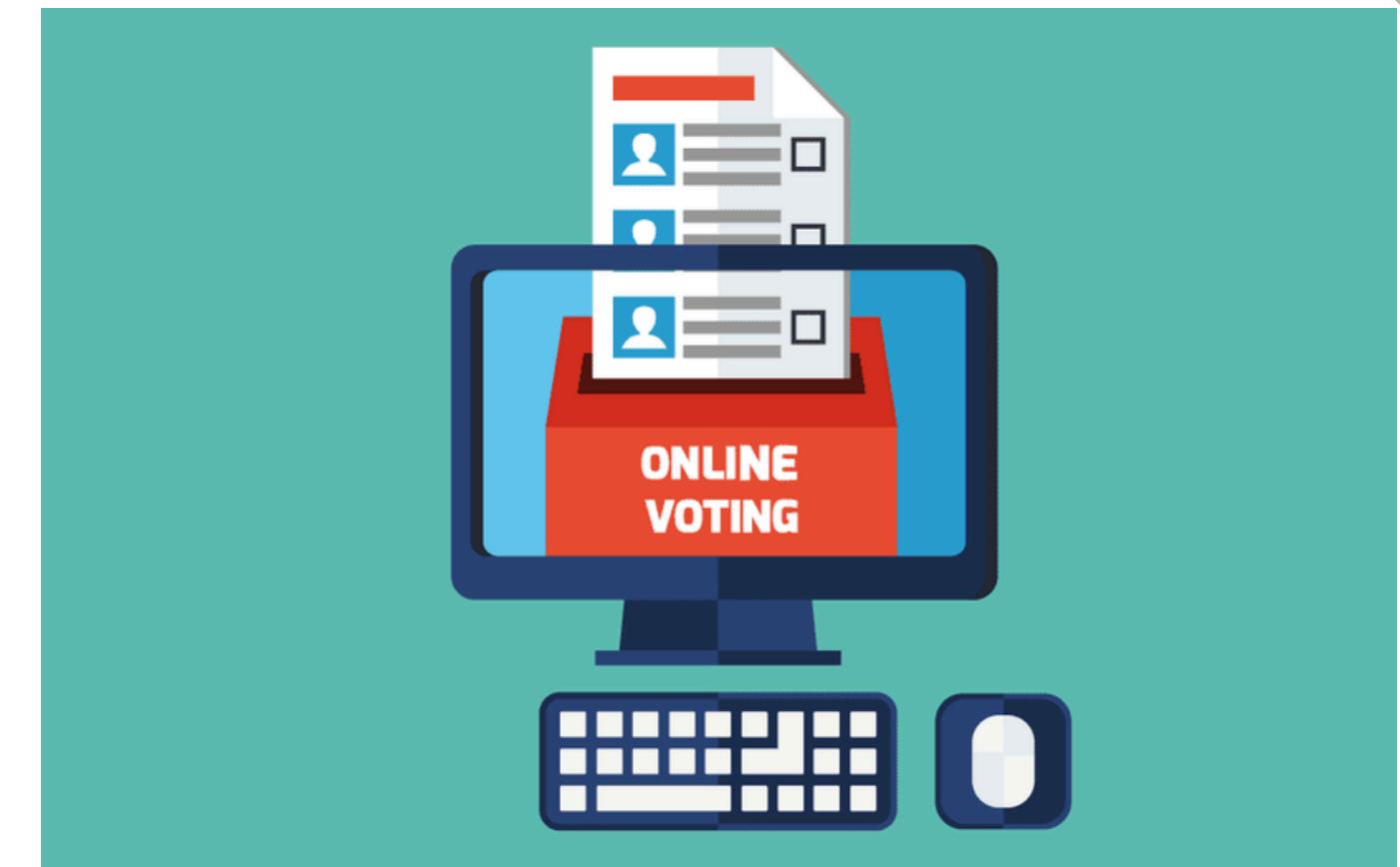
- Manual voting with paper ballots.
- Prone to errors, tampering, and delays.
- Low transparency and accessibility.

## Proposed System

- Secure online voting via web app.
- Real-time result computation.
- Easy access for students and admins.

## Feasibility Study

- Technical: MERN stack ensures scalability and security.
- Operational: Simple UI increases user adoption.
- Economic: Cost-effective using open-source tools.
- Security: JWT authentication protects user data.



# FUNCTIONAL REQUIREMENTS



user authentication



voting system



candidate  
management



Election  
management



Election  
management

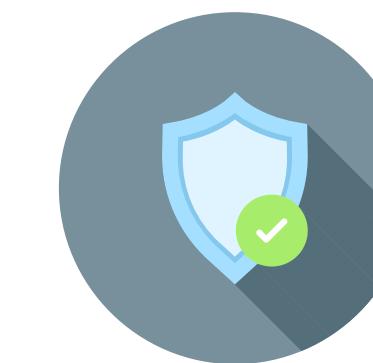
# NON - FUNCTIONAL REQUIREMENTS



security



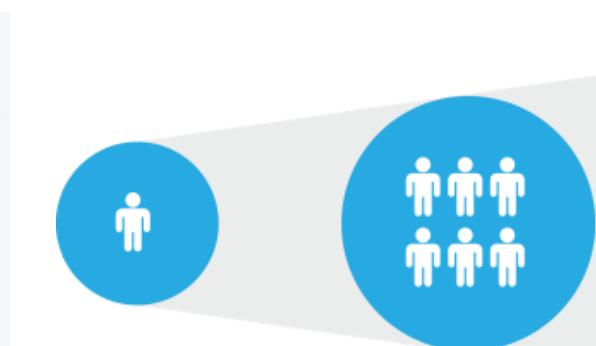
Performance



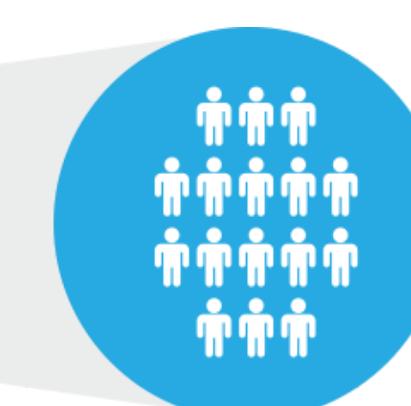
Reliability



usability



Scalability



# OVERVIEW

To design, develop, and deliver a secure, efficient, and transparent digital voting system for colleges, improving over traditional manual voting systems.

## Key Features:

- Secure student login and voting (JWT authentication).
- Admins manage elections and candidates.
- Real-time result display and updates.
- Responsive design for all devices.

## Technology Stack:

- Frontend: React.js, React Router, Context API.
- Backend: Node.js, Express.js.
- Database: MongoDB.
- Authentication: JWT tokens for secure sessions.

## System Architecture:

User → React Frontend → Express API → MongoDB Database.

# HARDWARE AND SOFTWARE SPECIFICATIONS

## Hardware Requirements

- Processor: Intel i5/Ryzen 5 or higher
- RAM: 8 GB minimum
- Storage: 100 GB (SSD recommended)
- Internet: Stable broadband connection

## Other Libraries/Packages:

- JWT for authentication
- Bcrypt.js for password encryption
- Axios for HTTP communication

## Software Requirements

- Frontend: React.js
- Backend: Node.js + Express.js
- Database: MongoDB
- Tools:
  - Visual Studio Code (IDE)
  - Postman (API Testing)
  - GitHub (Version Control)

# METHODOLOGY

Requirement → Design → Development → Testing → Deployment

## Requirement Analysis:

- Gathered functional and non-functional requirements for a secure, scalable voting platform.

## System Design:

- Created Data Flow Diagrams (DFDs), ER diagrams, and architectural designs to model system processes.

## Technology Stack Selection:

- Adopted the MERN stack (MongoDB, Express.js, React.js, Node.js) for seamless full-stack development.

## Development and Integration:

- Developed frontend interfaces, backend APIs, and database connections with secure authentication (JWT).

## Testing and Validation:

- Conducted unit testing, integration testing, and security testing to ensure reliability and data protection.

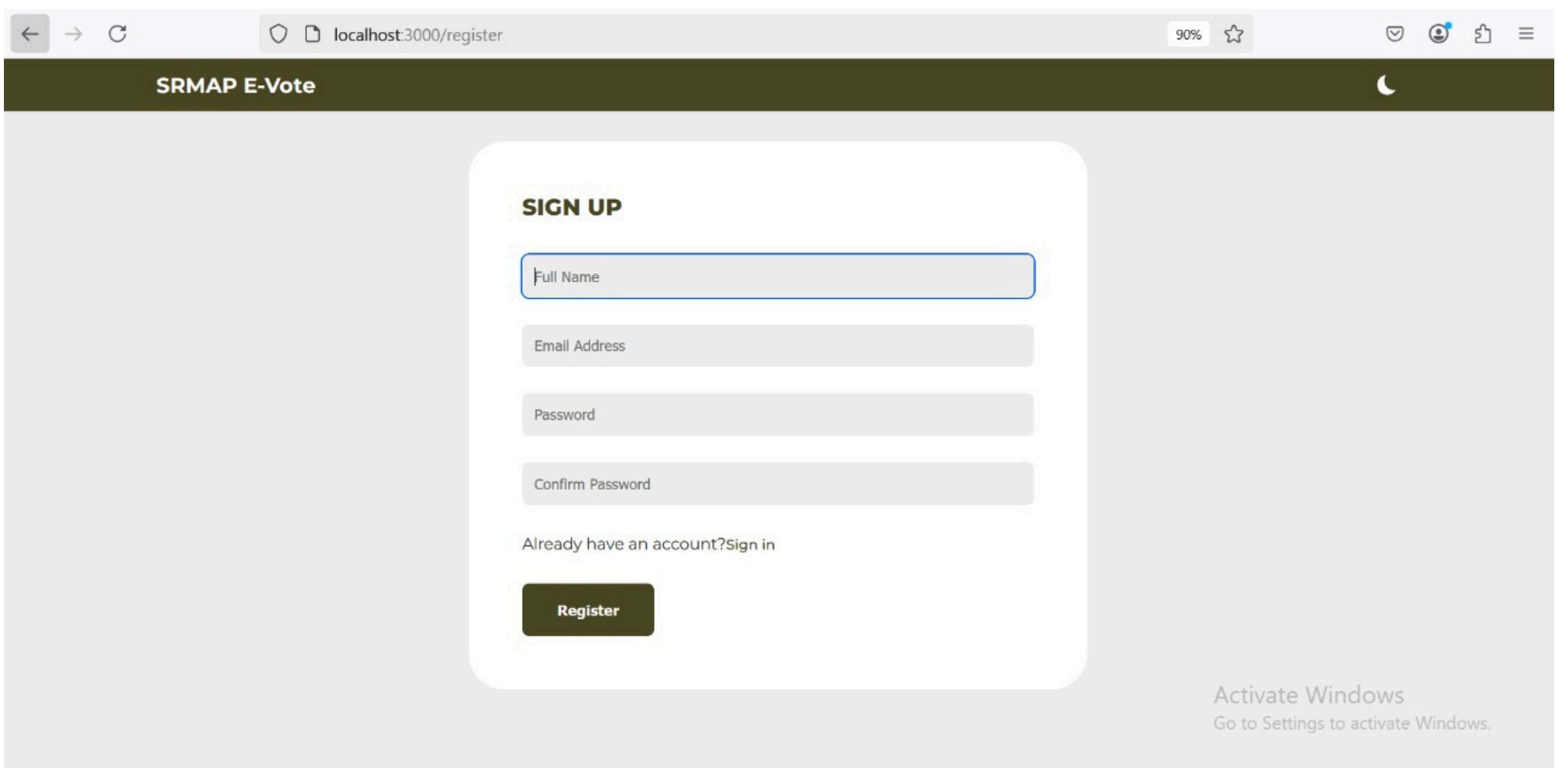
## Deployment and Access:

- Hosted the application on a cloud platform to provide easy, real-time access for students and administrators.



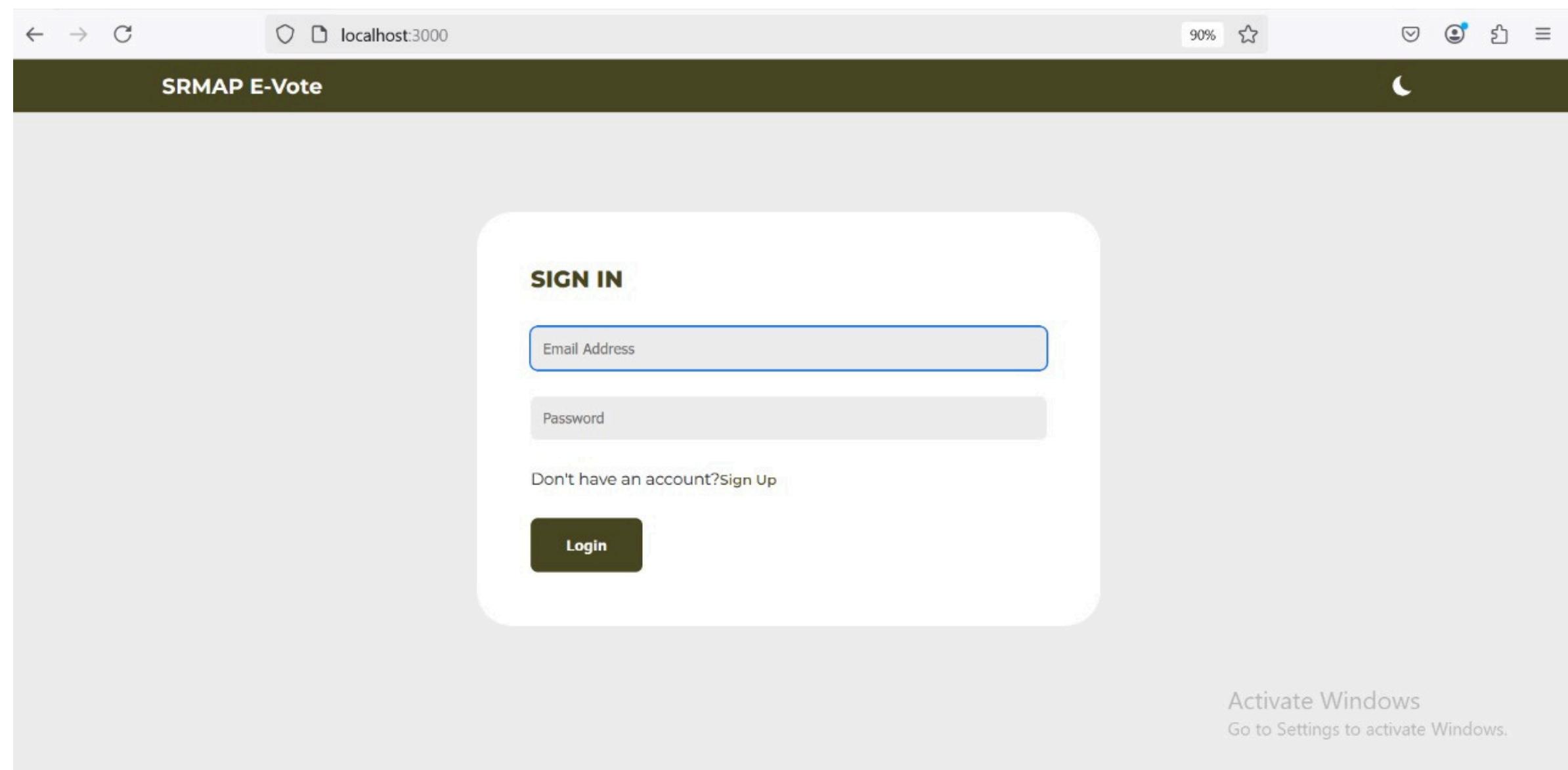
# SYSTEM OUTPUTS:

Registration page :



# SYSTEM OUTPUTS:

Login page :



# SYSTEM OUTPUTS:

Elections page :

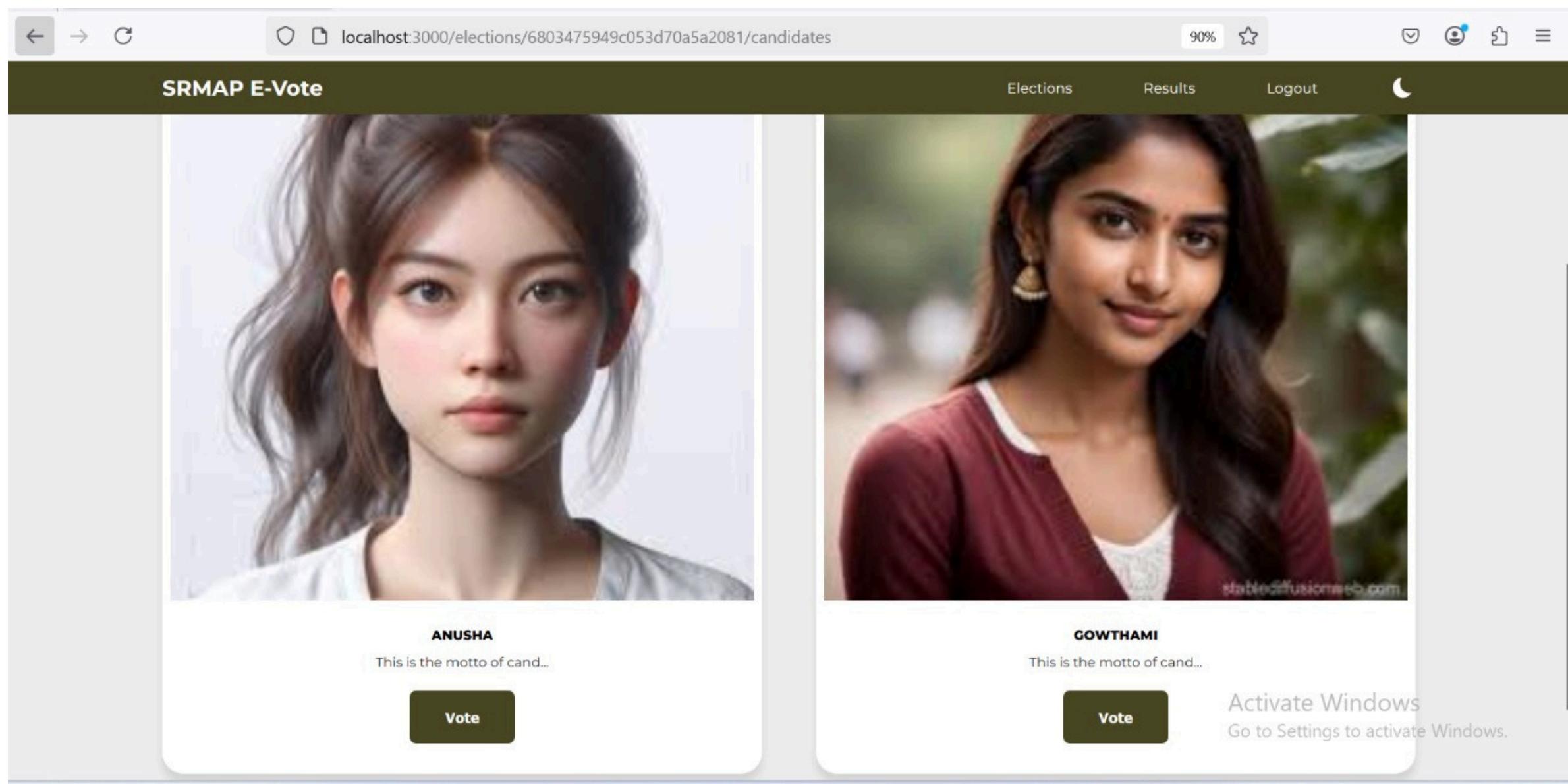
The screenshot shows a web browser window for the "SRMAP E-Vote" application at the URL [localhost:3000/elections](http://localhost:3000/elections). The page has a dark header bar with the title "SRMAP E-Vote" and navigation links for "Elections", "Results", and "Logout". A "Create New Election" button is visible in the top right corner. The main content area is titled "ONGOING ELECTIONS" and displays two entries:

- DANCE CLUB**: A thumbnail image shows several people dancing. The description text reads: "The dance club is a vibrant hub of creativity and passion, empowering dancers to excel on national platforms and within the university community. From its inception, the club has been driven by a mission to provide comprehensive training and opportunities...". There are "View" and "Edit" buttons below the description.
- CUBING CLUB**: A thumbnail image shows a person's hands solving a Rubik's Cube. The description text reads: "The world of cubing, where the twist and turn of a Rubik's Cube isn't just a pastime—it's a journey into the realm of mental acuity. It offers relaxation, social interaction, and learning opportunities, making it a versatile and rewarding pursuit for indi...". There are "View" and "Edit" buttons below the description.

In the bottom right corner of the page, there is a watermark-like message: "Activate Windows Go to Settings to activate Windows."

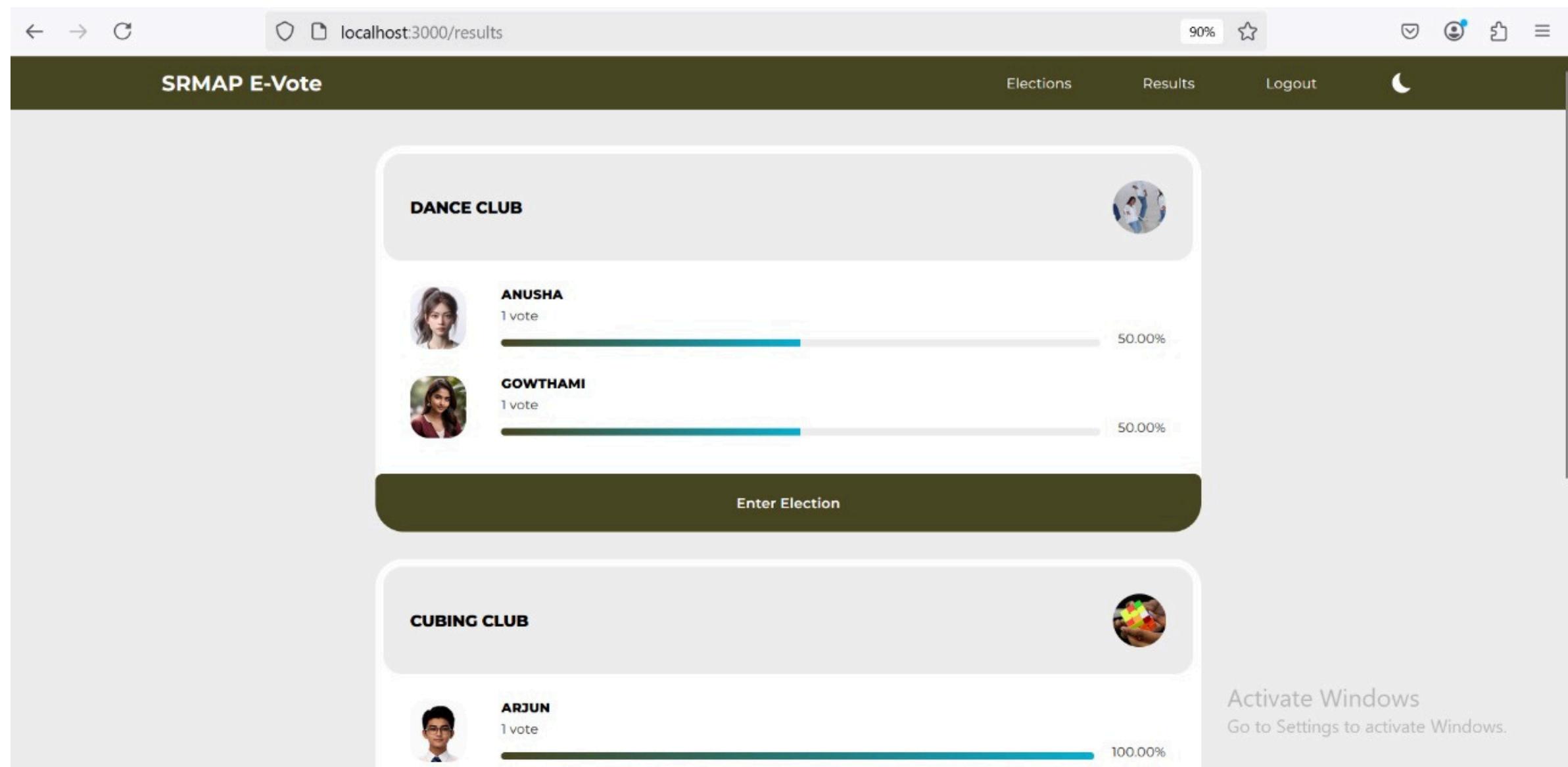
# SYSTEM OUTPUTS:

Voting page :



# SYSTEM OUTPUTS:

Results page :



# ADVANTAGES

- Reduces administrative workload by automating voting and result computation.
- Eliminates manual counting errors and improves election transparency.
- User-friendly interface for both students and admins.
- Real-time results increase engagement and trust.
- Easily extendable for mobile apps, analytics, or blockchain-based security.

# LIMITATIONS

- Requires stable internet connectivity for users to vote.
- Limited to college/university elections (not large-scale public voting).
- No offline voting mode for areas with poor internet access.
- Currently basic role control (Admin and Student only; no multi-level roles like Club Heads).

# FUTURE ENHANCEMENTS

- Integrate Two-Factor Authentication (2FA) for more secure logins.
- Implement Blockchain Technology to ensure tamper-proof voting records.
- Develop Mobile Applications for Android and iOS platforms.
- Add Role-Based Access Control (RBAC) to manage different admin levels (college head, club coordinator, etc.).

# CONCLUSION

- This project successfully automates and secures the college election process,
- providing a transparent, fast, and reliable voting platform.
- Students can easily vote online, and admins can efficiently manage elections and candidates.
- With future enhancements, the system can evolve into a complete, scalable solution for educational institutions' democratic processes.

# Thank You

This project has been successfully developed and presented by:

Sreeram reddy-AP22110010552

Gowthami reddy-AP22110010541

Manoj Kumar-AP22110010529

Mohith Varun-AP22110010573