VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 570018, KARNATAKA

Project Report
On

"Online Voting System"

In the partial fulfilment of the requirement for the award of degree

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

D P Kavya	**4VZ22CS009**
Mayank	**4VZ22CS016**
Pratheeksha R M	**4VZ22CS021**
Sharanabasavaraj A D	**4VZ22CS025**

Under the guidance of

Dr. G F Ali Ahammed
Program Coordinator,
Dept. of Computer Science and Engineering
VTU, Regional Office, Mysore.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISVESVARAYA TECHNOLOGICAL UNIVERSITY
Centre for Post Graduate Studies, Sathagalli, Mysuru – 570029.

2024 – 2025

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Centre for Post Graduate Studies, Sathagalli, Mysuru – 570029.

2024 – 2025

CERTIFICATE

This is to certify that the Project work entitled "Online Voting System" is a bonafied work carried out by **D P Kavya, Mayank, Pratheeksha R M, Sharanabasavaraj A D** bearing **USN 4VZ22CS009, 4VZ22CS016, 4VZ23CS021, 4VZ22CS025** at Department of Computer Science and Engineering, Visvesvaraya Technological University, Centre for Post Graduate Studies, Mysuru in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering, Visvesvaraya Technological University, Belagavi during the academic year 2024 - 2025. It is certified that all the corrections/suggestions indicated during Internal Assessment have been incorporated in the report. The Mini Project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Technology degree.

Signature of the Guide

Dr. G. F ALI AHAMMED
Program Coordinator,
Dept. of CS&E,
VTU, CPGS Mysuru – 570029

Signature of Program Coordinator

Dr. G. F ALI AHAMMED
Program Coordinator,
Dept. of CS&E,
VTU, CPGS Mysuru – 570029

Name of the Examiner **Signature with Date**

1. .....
2. .....

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground our efforts with success.

We express our sincere thanks to **Dr. G. F ALI AHAMMED, Program Coordinator, PG Studies, Department of Computer Science & Engineering, VTU Regional Centre, Mysuru** for his kind help and constant encouragement and for providing us necessary facilities for carrying out this work successfully.

In particular, We would like to take this opportunity to express our Honor, Respect, Deepest Gratitude and Genuine Regards to our guide, **Dr. G. F ALI AHAMMED**, **Program Coordinator**, Department of Computer Science and Engineering, VTU Regional Centre, for giving us all guidance required for our project apart from being a constant source of inspiration and motivation.

We owe our special thanks to **Our Parents** for their moral support and warm wishes and finally We would like to express appreciation to all **Our Friends** for their unconditional support which helped us to complete this work successfully.

D P Kavya	**4VZ22CS009**
Mayank	**4VZ22CS016**
Pratheeksha R M	**4VZ22CS021**
Sharanabasavaraj A D	**4VZ22CS025**

DECLARATION

We **D P Kavya, Mayank, Pratheeksha R M, Sharanabasavaraj A D** bearing **USN 4VZ22CS009, 4VZ22CS016, 4VZ23CS021, 4VZ22CS025** hereby declare that this project work entitled **" Online Voting System",** is a bona fide work carried out by us under the guidance and supervision of **Dr. G.F Ali Ahammed** Program Coordinator, Department of Computer Science and Engineering, VTU, CPGS, Mysore. This project work is submitted to **Visvesvaraya Technological University, Belagavi** in partial fulfilment of the requirements for the award to degree of **Bachelor of Technology in Computer Science and Engineering during the academic year 2024 - 2025**.

D P Kavya	**4VZ22CS009**
Mayank	**4VZ22CS016**
Pratheeksha R M	**4VZ22CS021**
Sharanabasavaraj A D	**4VZ22CS025**

ABSTRACT

The online voting system is a revolutionary platform that allows eligible voters to cast their votes securely and conveniently over the internet without visiting physical polling stations. Designed to enhance accessibility, it enables participation even from remote areas, ensuring inclusivity for all voters. By using voter ID and password authentication, the system guarantees security, privacy, and legitimacy in the voting process. With features like real-time monitoring and automated vote counting, it ensures transparency and accuracy while significantly reducing election costs and environmental impact. Despite challenges like the digital divide and cybersecurity concerns, the system holds immense potential to modernize elections, making the democratic process more efficient and accessible.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Aim of the project

To create a robust and secure online voting system that ensures transparent, efficient, and accessible elections by leveraging modern web technologies. The platform will facilitate seamless participation, protect against fraudulent activities, and provide real-time management of election data.

1.2 Motivation

The motivation for developing an online voting system stems from the limitations and inefficiencies of traditional voting methods. Physical polling stations often pose challenges such as long queues, logistical complexities, and restricted accessibility for people in remote areas, those with disabilities, or individuals living abroad. Additionally, manual voting processes are time-consuming, costly, and prone to errors or tampering. By leveraging digital technologies, an online voting system provides a convenient, secure, and cost-effective alternative that empowers voters to participate from anywhere with an internet connection. This project aims to modernize the electoral process, reduce environmental impact by eliminating paper ballots, and enhance inclusivity, thereby promoting a more efficient and democratic voting experience.

1.3 The Problem Statement

The key challenges with traditional voting systems include logistical complexities, limited accessibility, high operational costs, and vulnerabilities to tampering or errors. Developing an online voting system involves addressing the following concerns:

1. Authentication: Ensuring only eligible voters can participate while preventing multiple votes from the same individual.
2. Data Integrity: Protecting against hacking or unauthorized access to voter and candidate data.
3. Scalability: Designing the system to handle high voter traffic efficiently.
4. User Experience: Ensuring a simple, intuitive interface for users across all demographics.

1.4 Scope of the project

The scope of the Voting Management System project is to develop a secure, efficient, and user-friendly platform for managing the voting process in small to medium-scale elections. The system will enable voter registration, authentication, and electronic voting while ensuring data security and voter anonymity. It will provide features like eligibility verification, vote submission, and real-time result generation. The platform aims to simplify election management, eliminate manual errors, and enhance transparency. Role-based access will be implemented, restricting voters, administrators, and candidates to their specific functionalities. It will be scalable, accessible on multiple devices, and designed with robust security protocols. The system does not cater to large-scale national elections but lays the foundation for future enhancements, such as blockchain integration and AI-powered fraud detection. This project is intended to streamline voting processes, save time, and build trust in election outcomes. Future iterations could include multilingual support and advanced analytics for better insights.

CHAPTER 2

LITERATURE SURVEY

2.1 Existing system

Online voting systems have revolutionized the electoral process by enabling eligible voters to conveniently cast their votes through the internet. Neeraj Gutgutia, the founder and CEO of the Right2Vote project, introduced a platform where users can log in using their email ID or phone number. Remarkably, this system permits multiple users to access the same email ID or phone number and even allows voters to change their votes before the poll closes, enhancing flexibility. On the other hand, Neeraj Sawant's Nevon project offers an advanced login mechanism through fingerprint authentication, ensuring a higher level of security.

While these systems enhance convenience, they also pose challenges, such as the potential for hacking candidates' or voters' accounts, raising concerns about data integrity and security. A robust encryption mechanism and frequent system audits can mitigate these vulnerabilities. Moreover, educating users on secure login practices is crucial. Despite these challenges, such systems hold the potential to significantly modernize and democratize the voting process.

Limitations of Existing System

- Using the same email ID and phone number for multiple users in online voting systems increases the risk of security and privacy, Identity Verification Challenges.
- The ability to change votes can lead to loss of vote integrity, challenges in vote counting, confusion and complexity.
- Fake fingerprints can be used to spoof fingerprint systems, enabling unauthorized people to vote. The system may not recognize the voter if their finger pattern is severed or otherwise damaged.
- Hacking can expose sensitive personal data, including voter identification and preferences, compromising voter privacy and security

2.2 Proposed system

The online voting system project aims to create a secure, efficient, and user-friendly platform for conducting elections. This system will streamline the voting process, enhance accessibility, and ensure the integrity of votes cast. Implementing this system can improve voter engagement and streamline electoral processes while addressing challenges related to security and accessibility.

Advantages of the proposed system

- Convenience: Voters can participate from any location with internet access. This flexibility encourages higher voter turnout.
- Efficiency: Reduces waiting times at polling stations and speeds up vote counting. The automated tallying of votes allows for immediate processing and reporting of results.
- Accessibility: Increases voter participation as it eliminates the need to travel to physical polling booths. The platform can offer support in multiple languages, ensuring that non-native speakers can understand and participate in the voting process.
- Security and privacy: Utilizes secure authentication methods to prevent fraud and multiple voting attempts. This technique ensures that votes are encrypted from the moment they are cast until they are counted

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

To support the proposed Agro-culture database system effectively, the following hardware requirements are essential. These requirements ensure that the system operates efficiently, securely, and reliably, meeting the performance needs.

3.1 Hardware Specification

Processor: Intel/AMD

RAM: 512MB or more

Keyboard: RS/32 or USB/normal

Mouse: compatible mouse

3.2 Software Specification

Back-end: MySQL

Font-end: Html, CSS, JavaScript

Server: XAMPP

Code editor: VS code

3.3 Functionality Requirements

1. Data Management:

- Enable users to create, retrieve, update, and delete voter records, candidates, and election details in an online voting system.
- Enable efficient storage and retrieval of data using MySQL as the backend database.

2. User Interface:

- Provide a web-based user interface designed with HTML, CSS, and JavaScript for easy interaction with the system.
- Support user-friendly navigation for accessing different modules like user login and result analysis.

3. Data Security:

- Ensure secure data transactions through encrypted connections.
- Implement user authentication mechanisms to protect access to sensitive data.

4. Compatibility:

- Ensure the system works with standard input devices like RS/32 or USB keyboards and compatible mice.

5. Server Functionality:

- Use the XAMPP server to manage the web server and database connection seamlessly.
- Support concurrent user access without performance degradation.

6. Code Editing and Maintenance:

- Allow ease of development and code maintenance using VS Code as the primary code editor.

3.4 Non-Functionality Requirements

1. Performance:

- The system should operate efficiently with a minimum of 512MB RAM, ensuring responsiveness even with basic hardware configurations.
- Handle up to a specific number of simultaneous users without significant lag.

2. Reliability:

- The system must be reliable with minimal downtime during regular operations or maintenance.
- Ensure consistent data integrity, even in case of unexpected server crashes or restarts.

3. Usability:

- The interface must be intuitive and accessible to users with minimal training or technical knowledge.
- Provide responsive design for compatibility across different screen sizes.

4. Scalability:

- The system should support future upgrades to handle larger datasets or more complex queries.
- Ensure ease of scaling the database and server configuration as the user base grows.

5. Security:

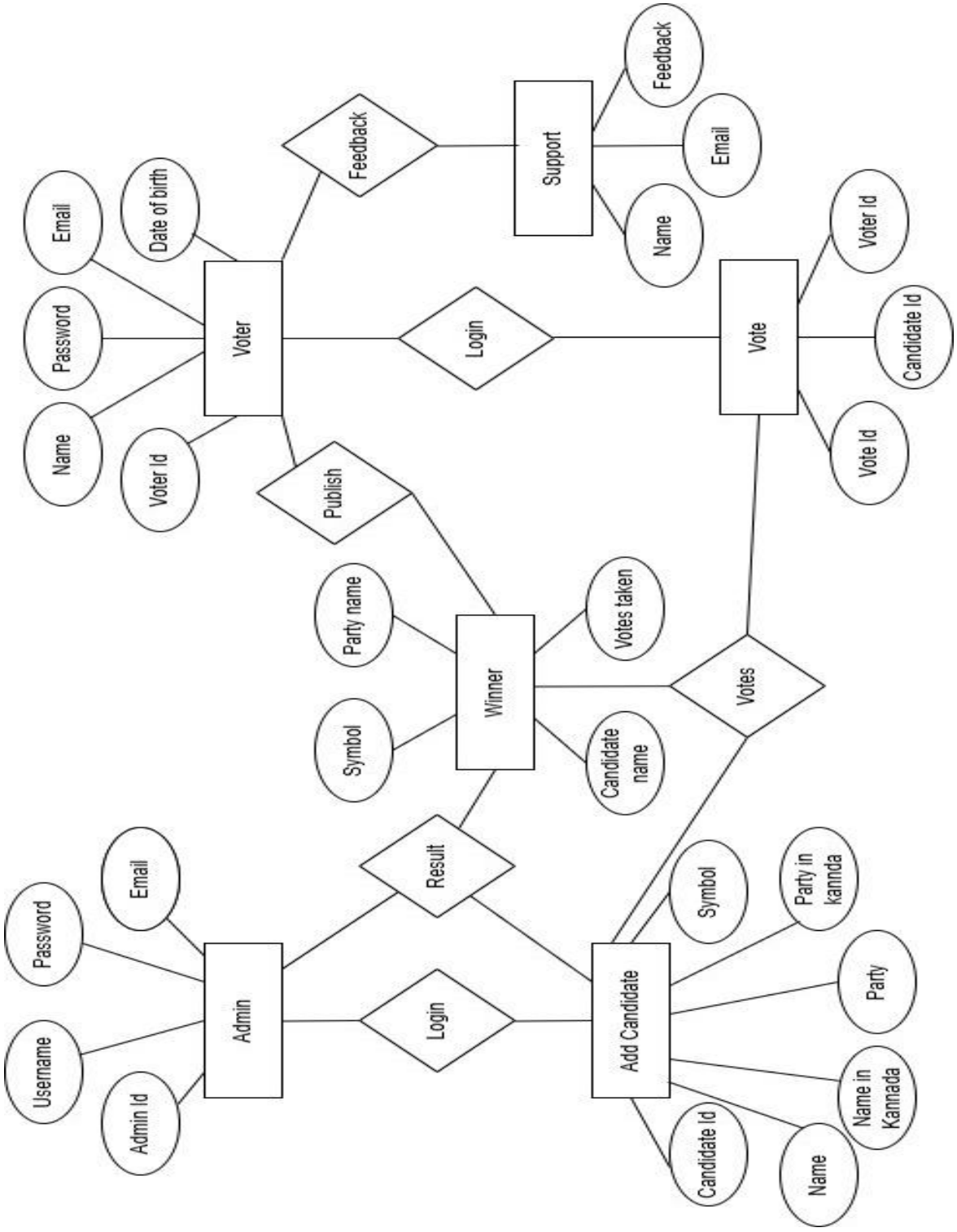
- Protect against common vulnerabilities like SQL injection and cross-site scripting (XSS).
- Ensure regular updates to the server and codebase to address potential security risks.

6. Maintainability:

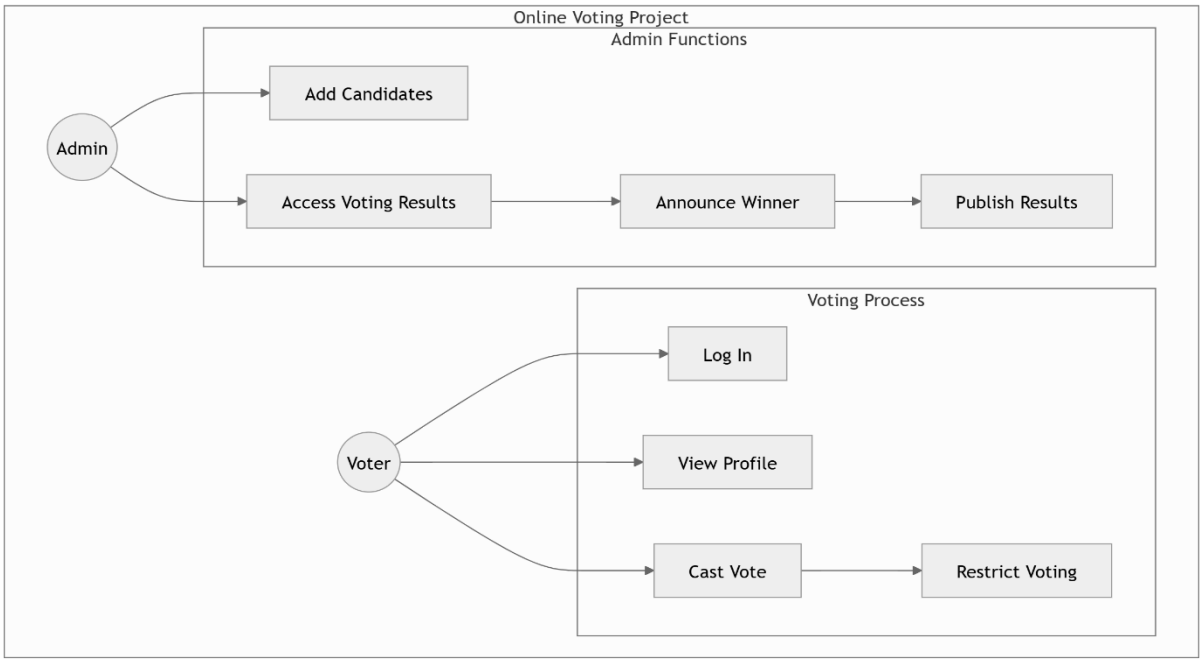
- The codebase should be modular and well-documented for ease of debugging and future enhancements.
- Allow seamless integration of new features without disrupting existing functionality.

CHAPTER 4
SYSTEM ARCHITECTURE

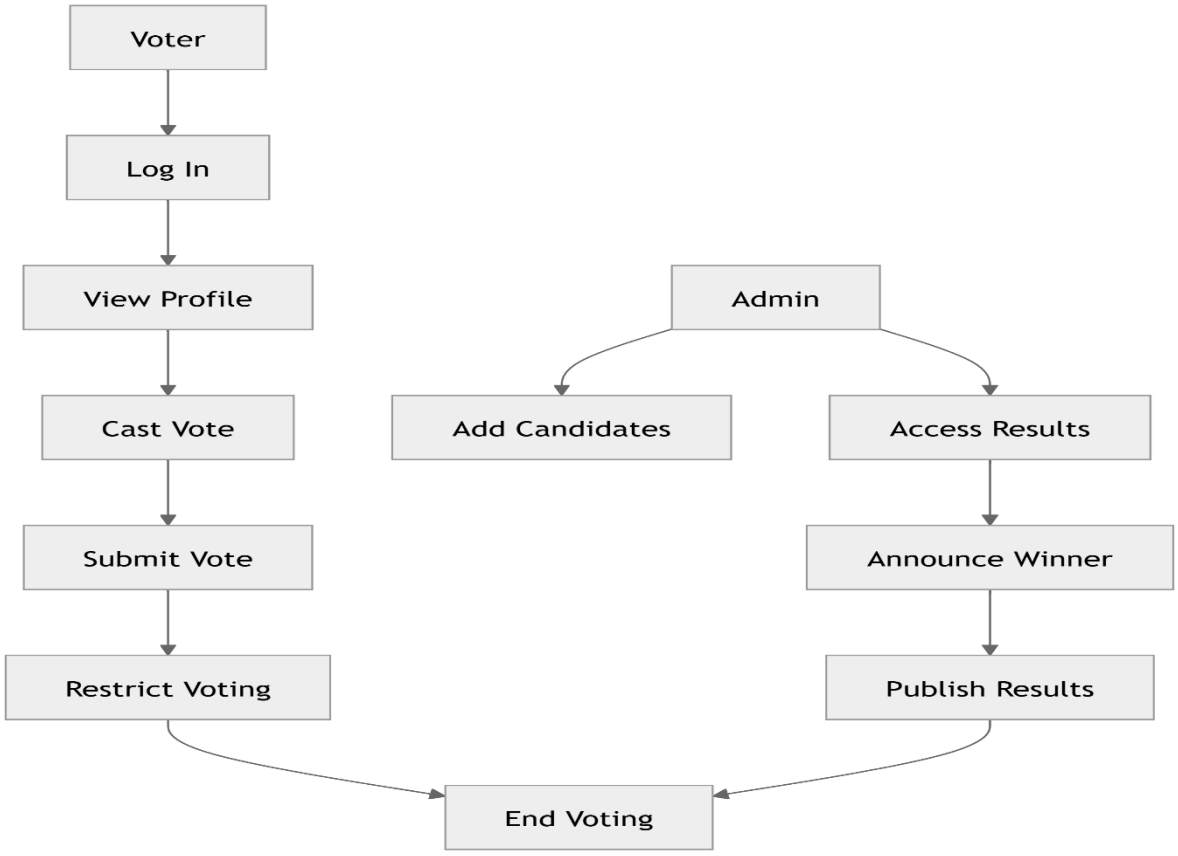
4.1 ER Diagram



4.2 Use case Diagram



4.3 Activity Diagram



CHAPTER 5

SYSTEM DESIGN

5.1 Database Table Structure

admin_id	username	password	email	created_at
1	Admin	password	admin@gmail.com	2024-11-14 00:00:00

Table 5.1

candidate_id	name	name_in_kannada	party	party_in_kannada	symbol_path	bio
9001	Narendra	ನರೇಂದ್ರ	Bharatiya Janata Party	ಭಾರತೀಯ ಜನತಾ ಪಕ್ಷ	6c93c7acd9dd15c41e97790eb93cd524	Ab ki Baar 400 Paar
9002	Aravind Kejrwal	ಅರವಿಂದ್ ಕೇಜ್ರಿವಾಲ	Aam Aadmi Party	ಆಮ್ ಆದ್ಮಿ ಪಕ್ಷ	d2de0700511a97cf42b8c6b9450ef463	bustling, 24x7 hospitable city
9003	Kapil Sharma	ಕಪೇಲ ಶರ್ಮ	Samaj Seva Party	ಸಮಾಜ ಸೇವಾ ಪಾರ್ಟಿ	2a1e9bde72cc7c0b4b68069af10bab43	Vote for Future
9004	Suresh Patil	ಸುರೇಶ ಪಾಟೀಲ	Liberty Party	ಲಿಬರ್ಟಿ ಪಾರ್ಟಿ	91b276d660a17a15908f0c5be2649f84	Vote For Liberty
9005	Smrithi Jadhav	ಸ್ಮೃತಿ ಜಾದ್ವ	Samajavadi Party	ಸಮಾಜವಾದ್ಮಿ ಪಕ್ಷ	cef74389fd72e4ec9ff1f09b28d7f1ee	Vote for Equality
9007	Chandrababu	ಚಿಂದ್ರಬಾಬು	Swachatha Party	ಸವಚಚತಾ ಪಕ್ಷ	cea36aafd7d14d83fc3c23b7cb5ca699	Vote to control Pollution
9008	Shivakumar Singh	ಶಿವಕುಮಾರ್ ಸ್ಮೃಿಂಗ್	Jan Jeevan Party	ಜನ ಜೀವನ್ ಪಾರ್ಟಿ	19206dd7defb0d8c0a24fcb080b5b007	Vote

Table 5.2

Name	Email	message
Raju	raju@gmail.com	It was nice
Mayank	mayankbhandari647@gmail.com	How to vote

Table 5.3

vote_id	Voter name	Password	email	registration_date	adhar_no	fingerprint
10001	Arjun	1234	arjun@example.com	1995-07-15	415738861233	NULL
10002	Priya	1234	priyaexample.com	1990-09-25	203948573829	NULL
10003	Rahul	1234	rahul@example.com	1998-12-05	640937462810	NULL
10004	Meera	1234	meera@example.com	1993-03-18	402944534929	NULL
10005	Vikram	1234	vikram@example.com	1992-11-22	203948273892	NULL
10006	Anjali	1234	anjali@example.com	1997-08-30	847203927182	NULL
10007	Rohan	1234	rohan@example.com	1994-01-10	647110292832	NULL
10008	Sohan	1234	sohan@example.com	1996-06-15	192847369502	NULL
10009	Aditya	1234	aditya@example.com	1991-04-20	39029383928	NULL
10010	karthik	1234	karthik@example.com	1999-02-14	098736482738	NULL
10011	Mayank	1234	mayank@example.com	1999-02-14	214729375839	NULL
10012	Ravi	1234	ravi@example.com	1999-02-04	298346485032	NULL
10013	Vikas	1234	Vikas@example.com	1999-12-14	381020293009	NULL
10014	Varun	1234	varun@example.com	1999-05-14	987402936483	NULL
10015	Pavan	1234	pavan@example.com	1994-02-14	102938291247	NULL
10016	Ramesh	1234	ramesh@gmail.com	1997-10-13	920382754678	NULL
10017	Vishvas	1234	vishwas@gmail.com	1998-07-13	478392678353	NULL
10018	Rishabh	1234	rishi@gmail.com	1994-07-12	283728467383	NULL
10020	Pavan	1234	pavan@gmail.com	2004-06-07	209485789643	NULL
10021	Rani	1234	vikas@gmail.com	2002-09-24	456785545562	NULL
10022	Ravi	1234	ravi@email.com	2003-11-05	993930039282	NULL

Table 5.4

vote_id	voter_id	candidate_id	voted_at
30	10002	9001	2024-11-30 16:26:33
31	10003	9001	2024-12-01 10:09:48
32	10006	9001	2024-12-03 19:53:19
33	10007	9001	2024-12-03 19:54:04
34	10022	9001	2024-12-12 19:51:23
35	10001	9005	2024-12-17 18:43:03
36	10010	9003	2024-12-17 18:43:51
37	10012	9008	2024-12-17 18:44:12
38	10013	9002	2024-12-17 18:44:32
39	10016	9008	2024-12-17 18:44:54

Table 5.5

CHAPTER 6

IMPLEMENTATION

6.1 Packages

1. Express

- Description: Express is a web application framework for Node.js that simplifies the process of building web applications and APIs. It provides robust routing and middleware options.
- Usage:
    1. Define application routes (e.g., GET, POST, etc.).
    2. Handle HTTP requests and responses.
    3. Serve static files like images, CSS, and JavaScript.

2. EJS (Embedded JavaScript)

- Description: EJS is a template engine for rendering HTML pages dynamically. It allows you to embed JavaScript logic into HTML templates for dynamic content generation.
- Usage:
    1. Render dynamic HTML pages with data passed from the server.
    2. Use loops and conditions directly in the template for customization.

3. Express-Session

- Description: Express-Session is a middleware for managing user sessions in an Express app. It stores session data on the server-side and provides session management for user authentication and state persistence.
- Usage:
    1. Track logged-in users across multiple requests.
    2. Store temporary user-specific data like cart details or preferences.

4. Multer

- Description: Multer is a middleware for handling file uploads in Node.js. It is used to process multipart/form-data, which is primarily used for file uploading.
- Usage:
    1. Handle file uploads from forms.
    2. Save uploaded files to a specific directory or database.

5. MySQL

- Description: MySQL is a Node.js package that allows interaction with a MySQL database.
- Usage:
    1. Connect to the database and execute queries.
    2. Perform CRUD operations on tables.

6. Body-Parser

- Description: Body-Parser is middleware that parses incoming request bodies in a middleware before your handlers. It parses JSON, URL-encoded, and other data formats.
- Usage:
    1. Parse JSON and form data from HTTP POST requests.
    2. Extract data from the req.body object in Express.

6.2 Functions

1. use

Purpose: Attaches middleware to the Express app. Middleware functions are executed sequentially to process requests and responses.

2. set

Purpose: Used to set application settings or configurations in an Express app.

3. get

Purpose: Defines a route to handle HTTP GET requests.

4. post

Purpose: Defines a route to handle HTTP POST requests, often used for form submissions or API calls.

5. listen

Purpose: Starts the server and listens for incoming connections on a specified port.

6. require

Purpose: Imports external modules or files into your Node.js application.

6.3 Pseudocode for Backend Implementation

1. Use method

```
app.use(cors());

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.use(express.static("public"));
```

2. Set method

```
// Set EJS as the view engine

app.set("view engine", "ejs");

app.set("views", "./views");
```

3. Get method

```
app.get("/", (req, res) => {
res.render("user_login"); // Render login page
});
```

4. Post method

```
app.post("/user_login" , (req, res) => {
const { voter_id, voter_name, email, dob, aadhar_no, password, fingerprint_data } = req.body;
const query = `
INSERT INTO users (voter_id, votername, email, registration_date, aadhar_no, password, fingerprint)
VALUES (?, ?, ?, ?, ?, ?, ?) `;
conn.query(query, [voter_id, voter_name, email, dob, aadhar_no, password, fingerprint_data], (error, result) => {
if (error) {
console.error("Error inserting data:", error);
res.status(500).send("Failed to insert data.Email Exists");
} else {
console.log("Data successfully inserted");
res.send(`<script>
alert("You are Successfully registered");
window.location.href = "/";
</script>`);
      }
   });
});
```

5. Listen method

```
// Start the server
const PORT = 3000;
app.listen(PORT, () => {
console.log(Server is running on http://localhost:${PORT});
});
```

6. CreateConnection method

createConnection method

```
const conn = mysql.createConnection({
host: "localhost",
user: "root",
password: "",
database: "voting1",
port: "3311",
});
```

CHAPTER 7

TESTING

7.1 Test Cases for Functional Requirements

Test Case ID	Description	Input	Expected Result	Status
TC001	Verify login with valid credentials	Email: test@example.com Password: 123456	User is successfully logged in	Pass/Fail
TC002	Verify login with invalid credentials	Email: test@example.com Password: wrongpass	Error message: "Invalid email or password"	Pass/Fail
TC003	Ensure only one vote per voter	Cast vote twice using same account	Second attempt is rejected, error message displayed	Pass/Fail
TC004	Verify logout functionality	Click logout button	User is redirected to the login page	Pass/Fail

7.2 Data Integrity

Test Case ID	Description	Input	Expected Result	Status
TC005	Verify data is encrypted during transmission	Monitor network traffic	Data packets should be encrypted	Pass/Fail
TC006	Ensure database integrity after a vote is cast	Cast a vote	Vote is recorded correctly in the database	Pass/Fail
TC007	Test for SQL injection vulnerability	Enter malicious input: ';DROP TABLE votes;--	Input is sanitized, system handles safely	Pass/Fail

7.3 Scalability

Test Case ID	Description	Input	Expected Result	Status
TC008	Test system performance under high traffic	Simulate 1000 concurrent users	System remains responsive, no crashes	Pass/Fail
TC009	Verify database can handle multiple votes concurrently	Cast votes from multiple users simultaneously	Votes are recorded correctly	Pass/Fail

7.4 User Experience

Test Case ID	Description	Input	Expected Result	Status
TC010	Verify responsiveness of the UI	Open on different devices (desktop, tablet, mobile)	UI adjusts appropriately to screen size	Pass/Fail
TC011	Ensure error messages are user-friendly	Enter invalid input in any form	User-friendly error message displayed	Pass/Fail
TC012	Test navigation between pages	Click on various links/buttons	Pages load without errors	Pass/Fail

7.5 Hardware Requirements

Test Case ID	Description	Input	Expected Result	Status
TC013	Test system functionality on minimum hardware	System with 512MB RAM	System operates without performance degradation	Pass/Fail

7.6 Software Requirements

Test Case ID	Description	Input	Expected Result	Status
TC014	Verify compatibility with the specified server	Deploy on XAMPP server	Application works correctly	Pass/Fail
TC015	Verify database connection	Start XAMPP, query MySQL DB	Database connects successfully	Pass/Fail

7.7 Test Cases for Voting Functionality

Test Case ID	Description	Input	Expected Result	Status
TC016	Verify the AI tool assists users with queries about online voting system	User asks, "how to register for the online voting system?" and some others	The AI tool provides clear and accurate registration instructions	Pass/Fail
TC017	Ensure vote count updates correctly	Cast a vote	Candidate's vote count increases by 1	Pass/Fail

These test cases ensure the system is robust, user-friendly, secure, and meets both functional and non-functional requirements.

CHAPTER 8

RESULT

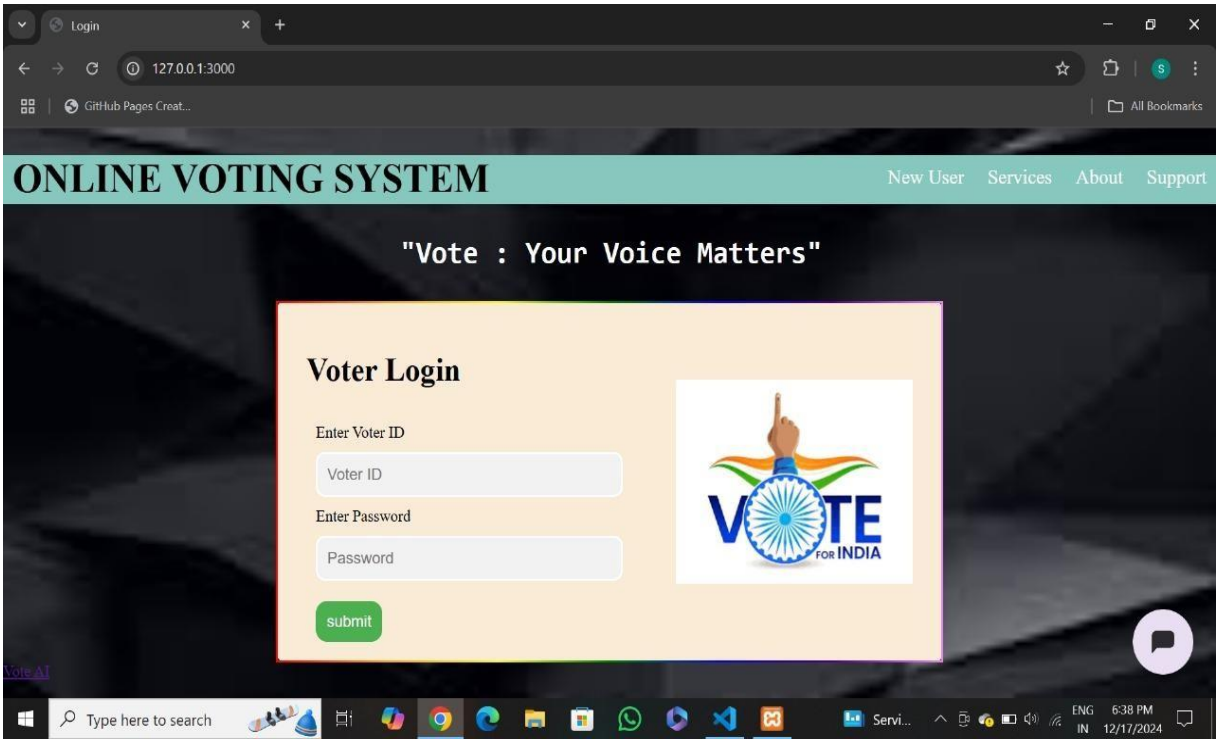


Figure 8.1

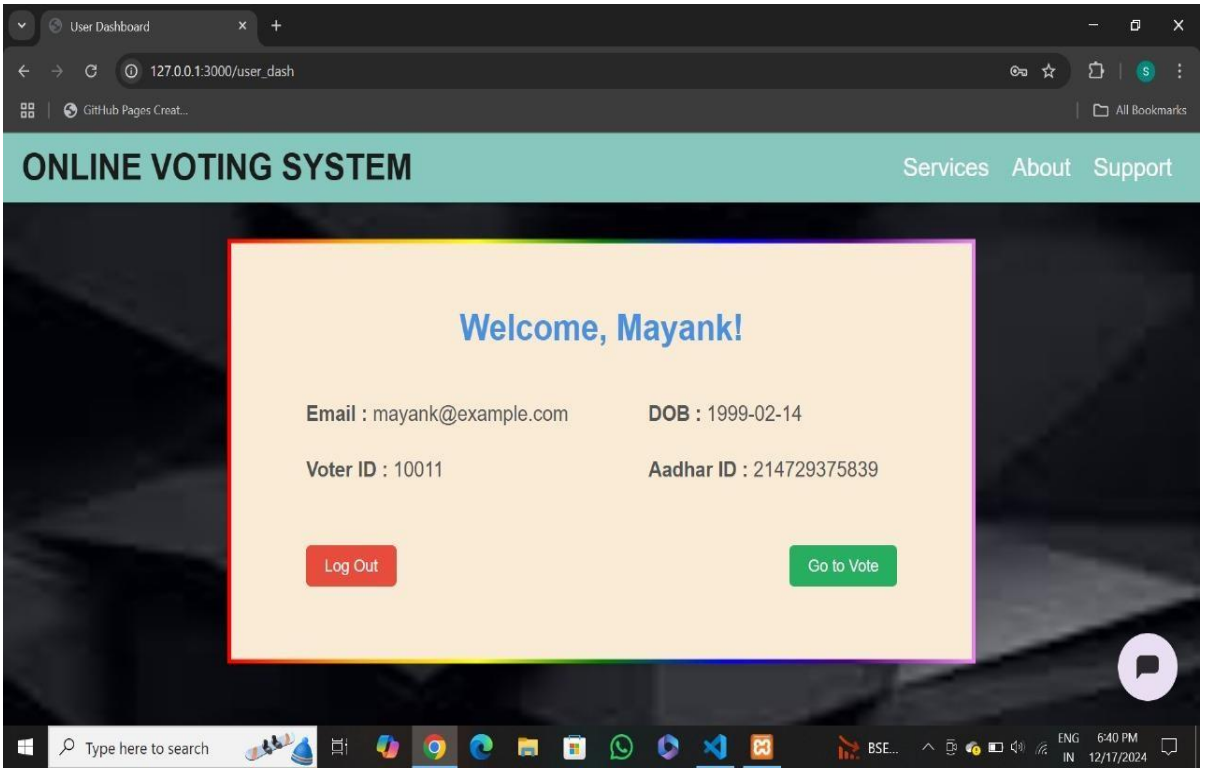


Figure 8.2

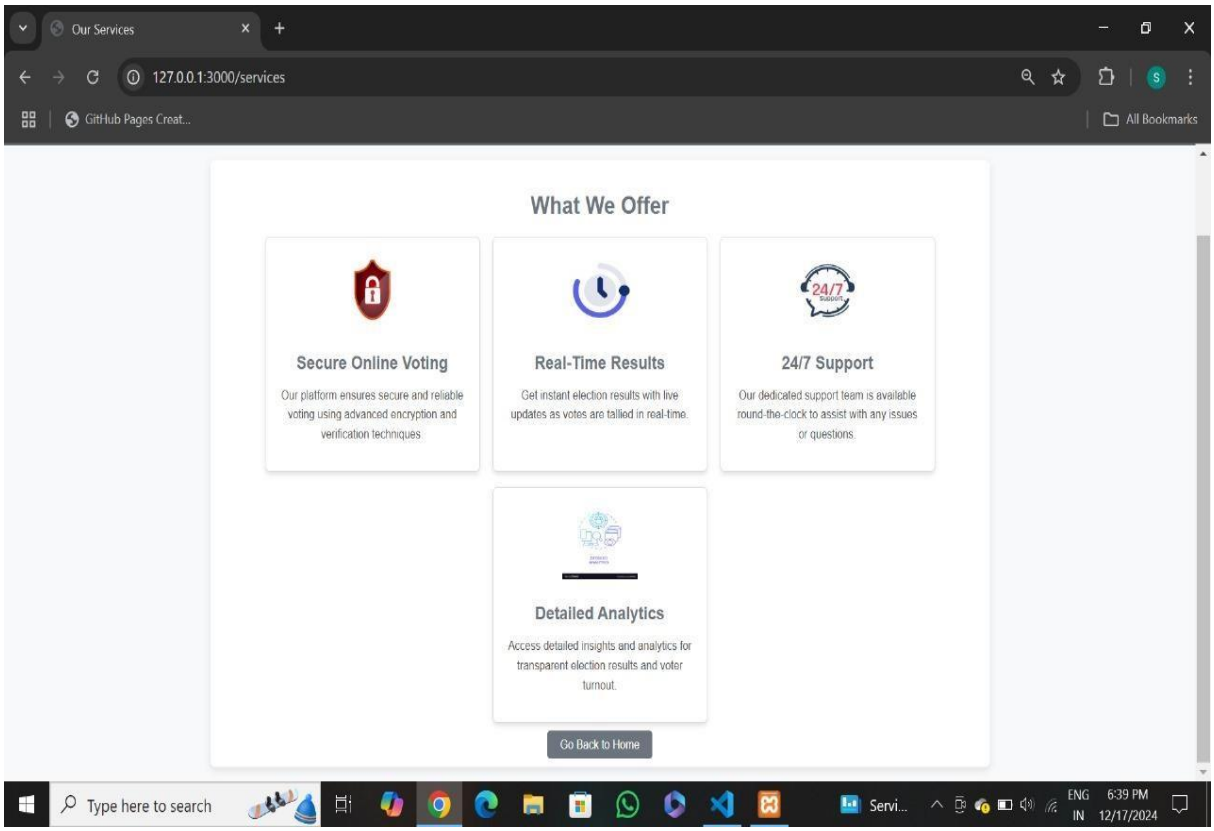


Figure 8.3

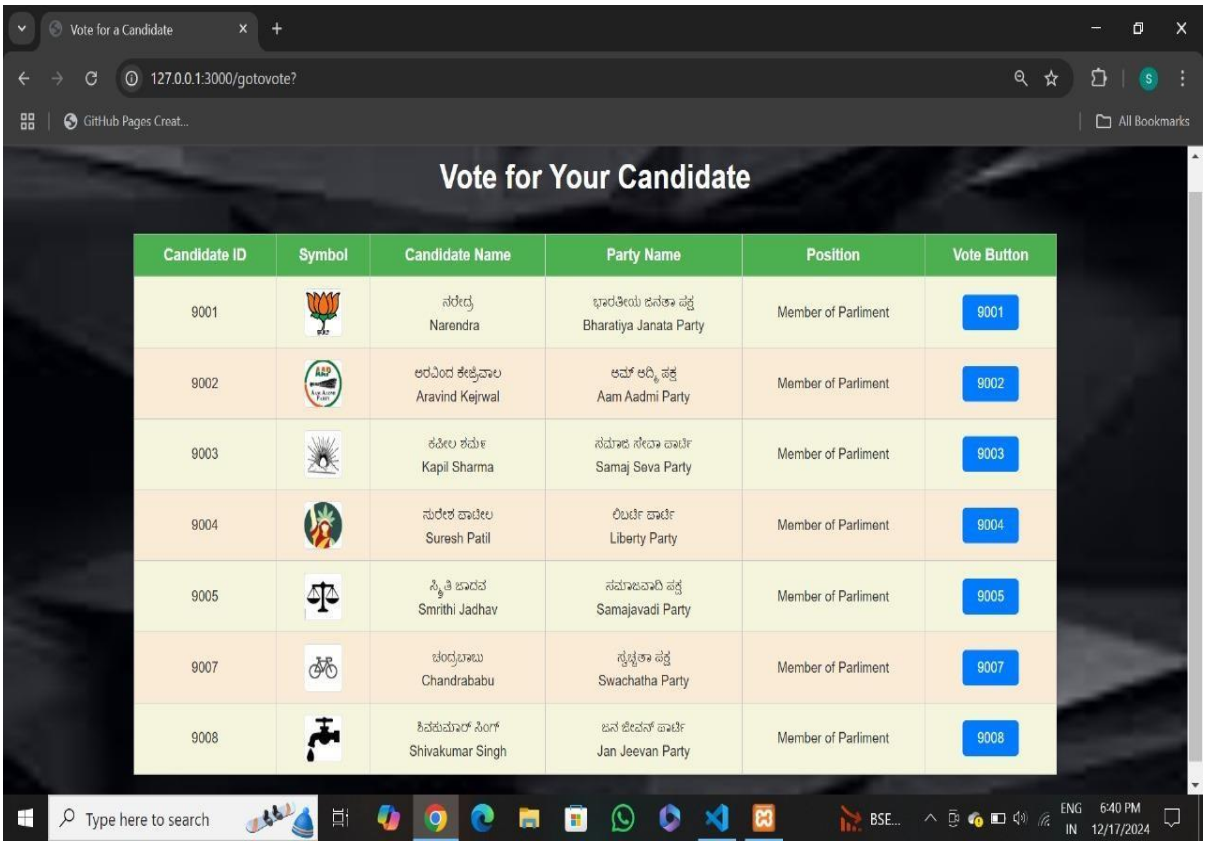


Figure 8.4

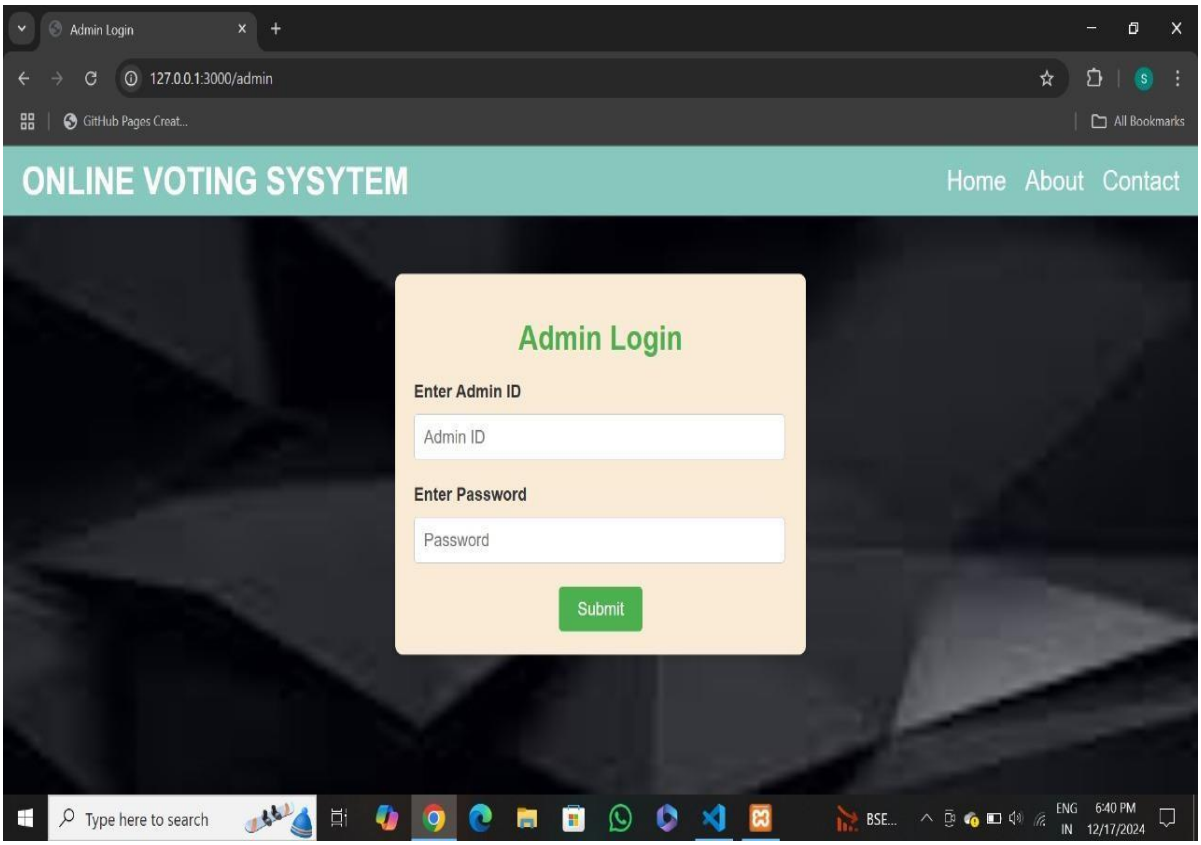


Figure 8.5

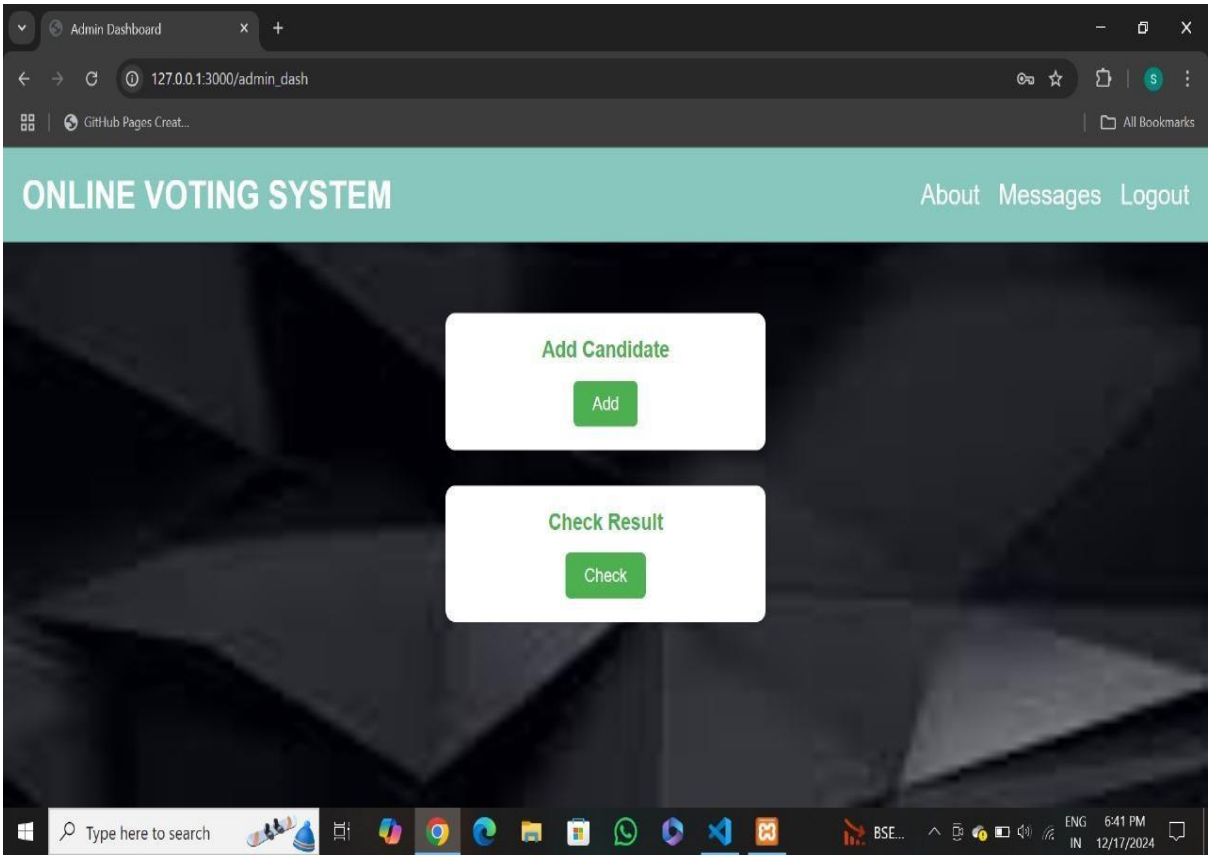


Figure 8.6

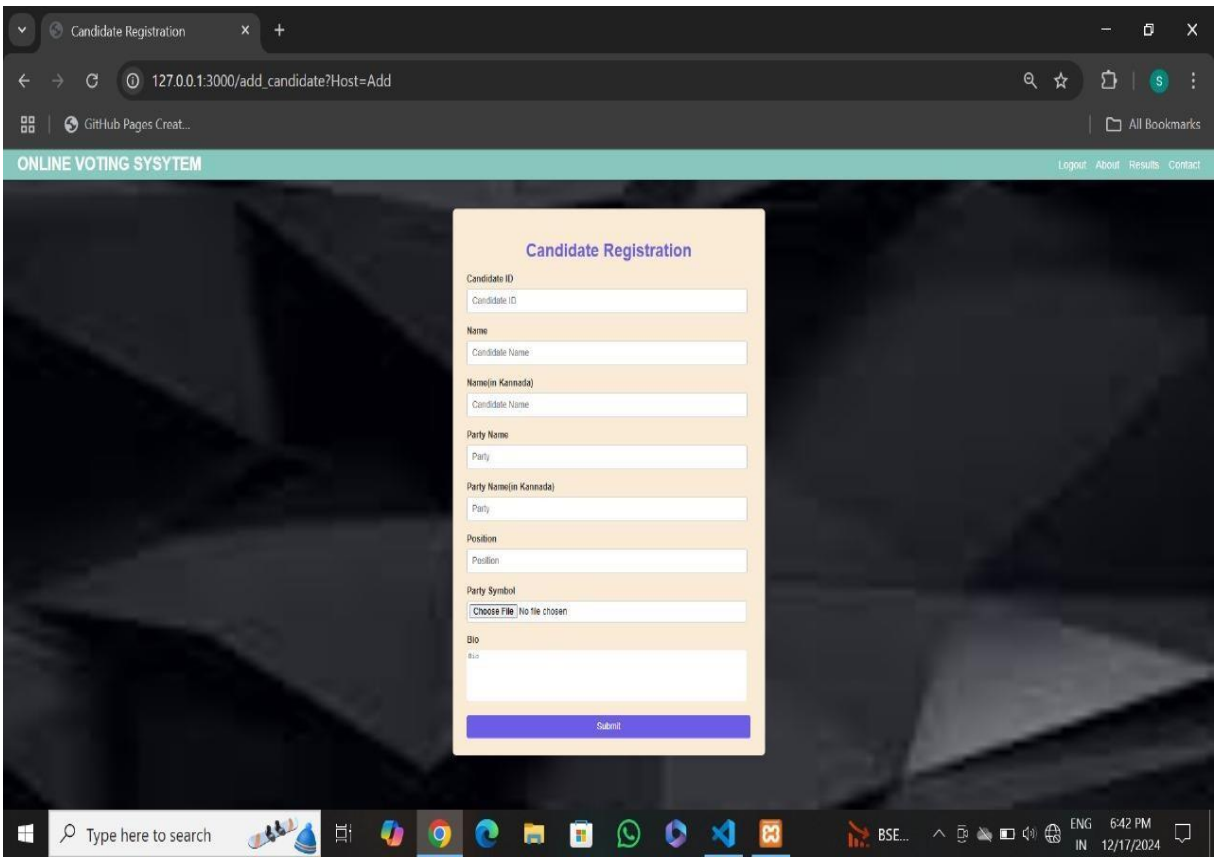


Figure 8.7

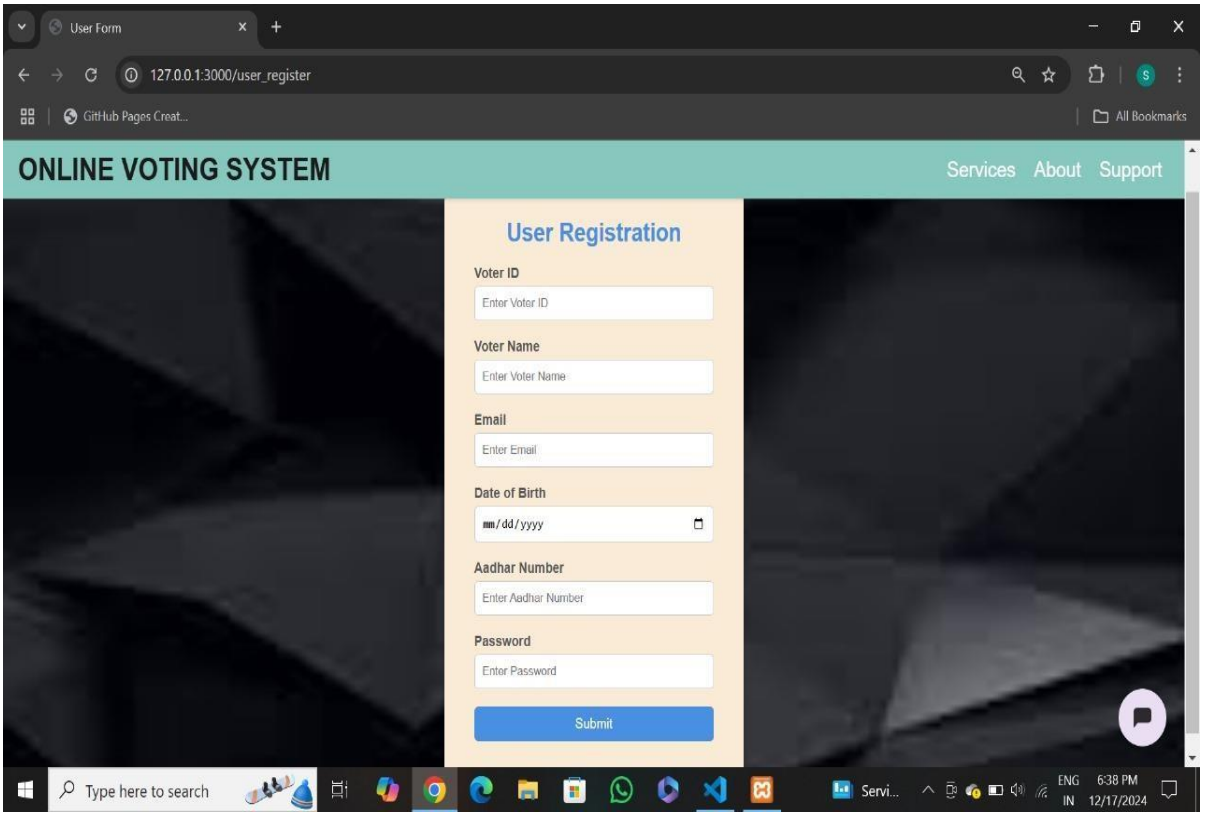


Figure 8.8

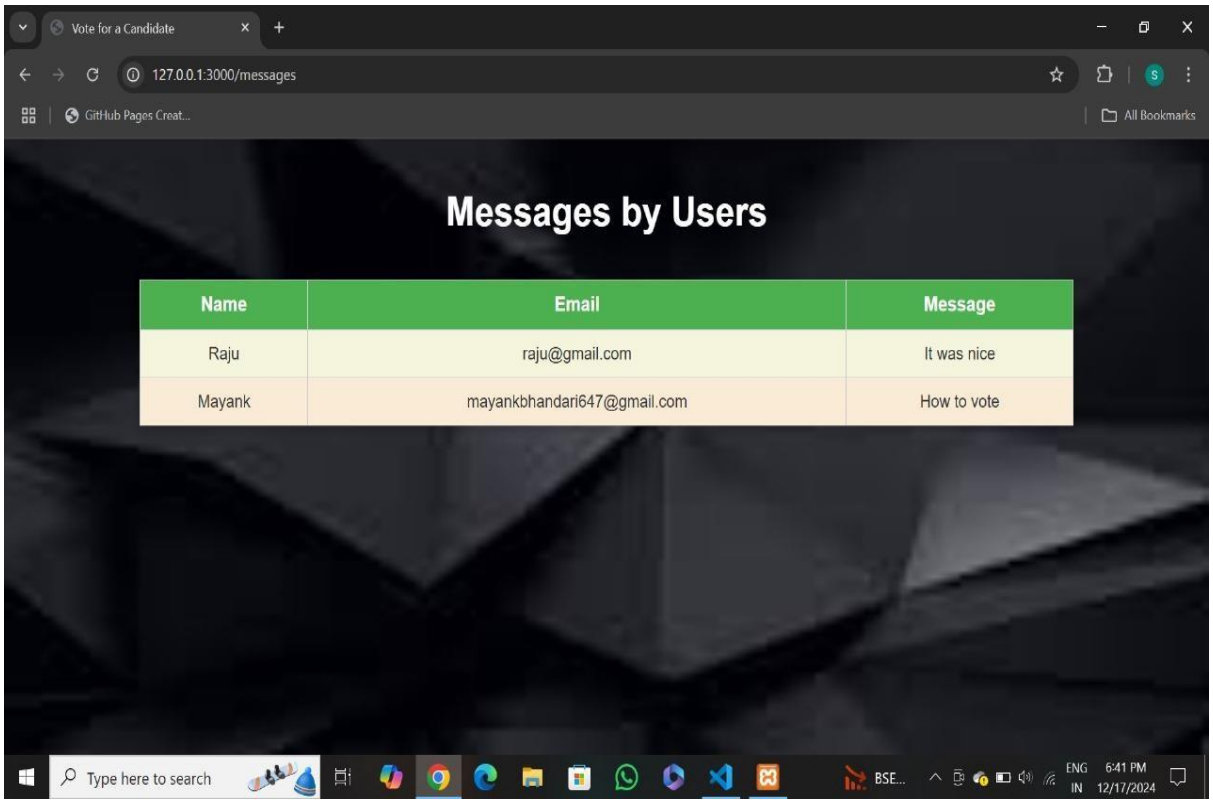


Figure 8.9

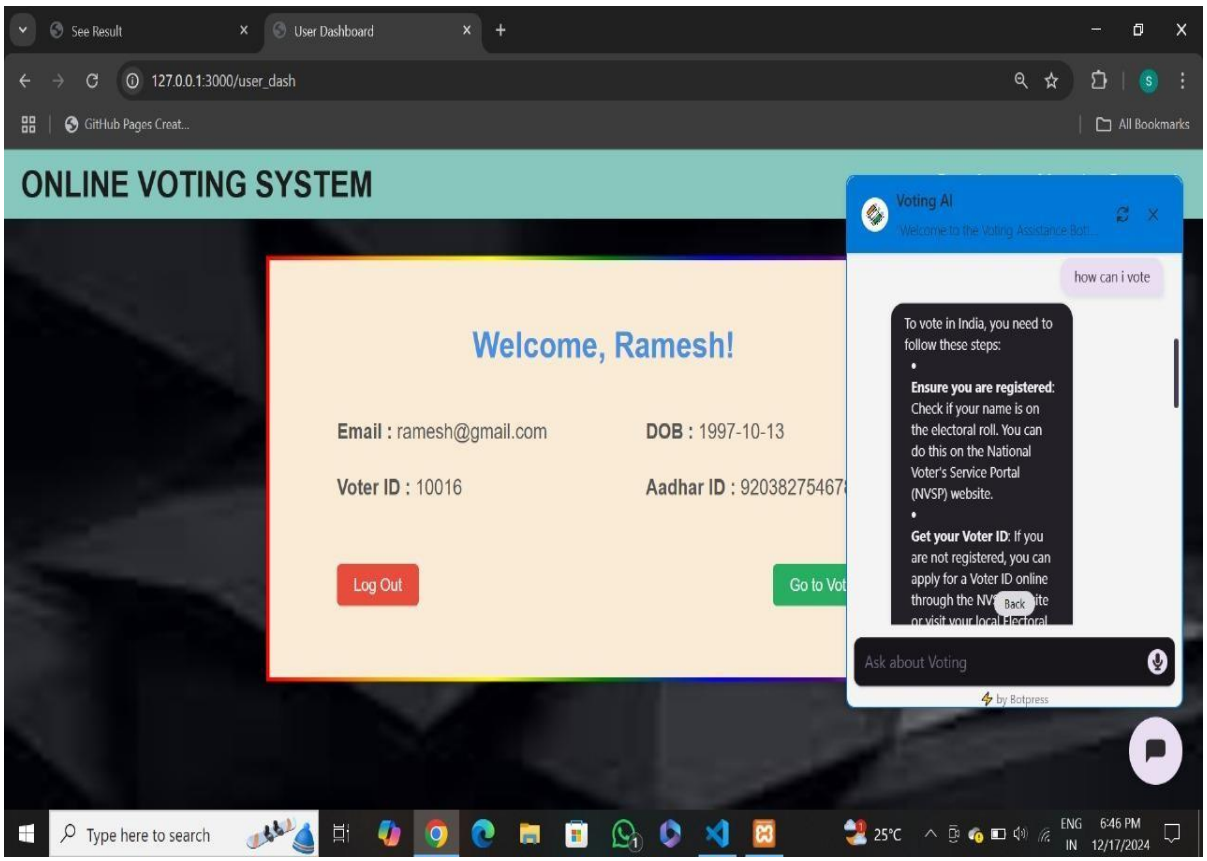


Figure 8.10

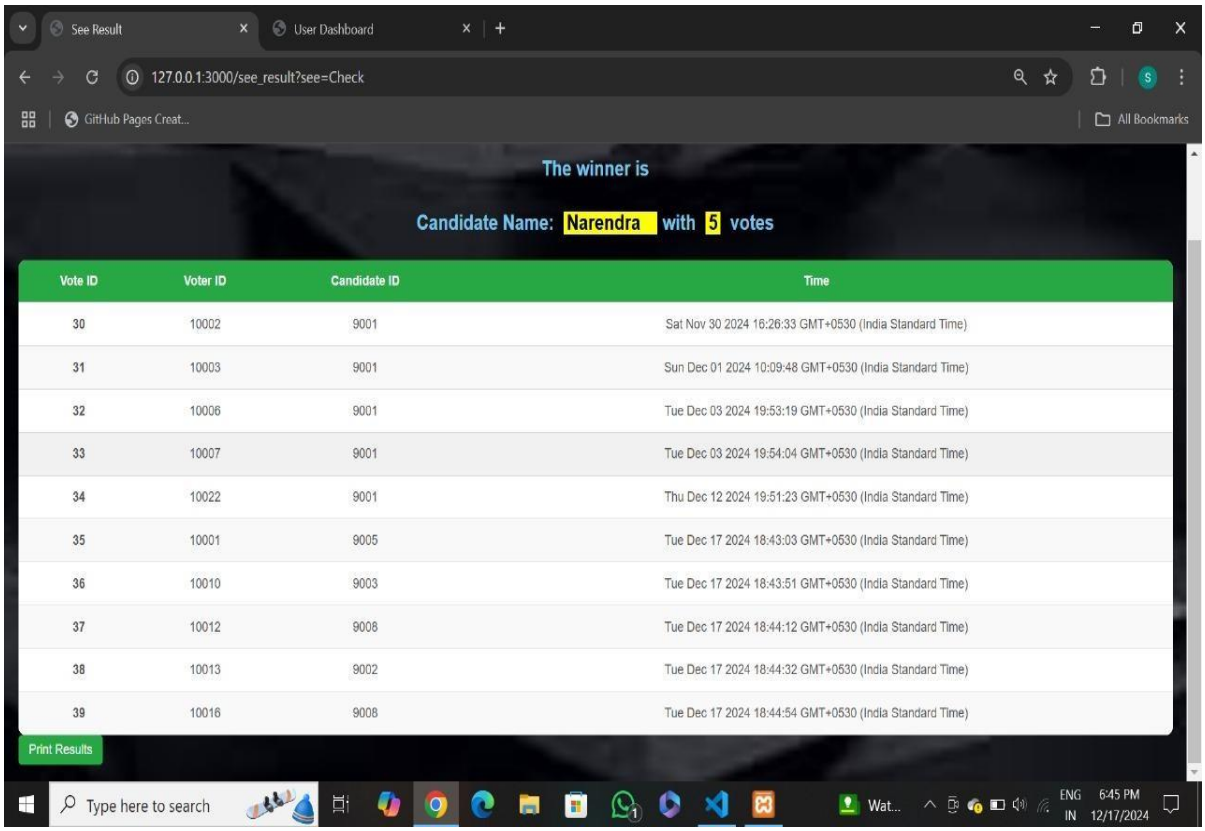


Figure 8.11

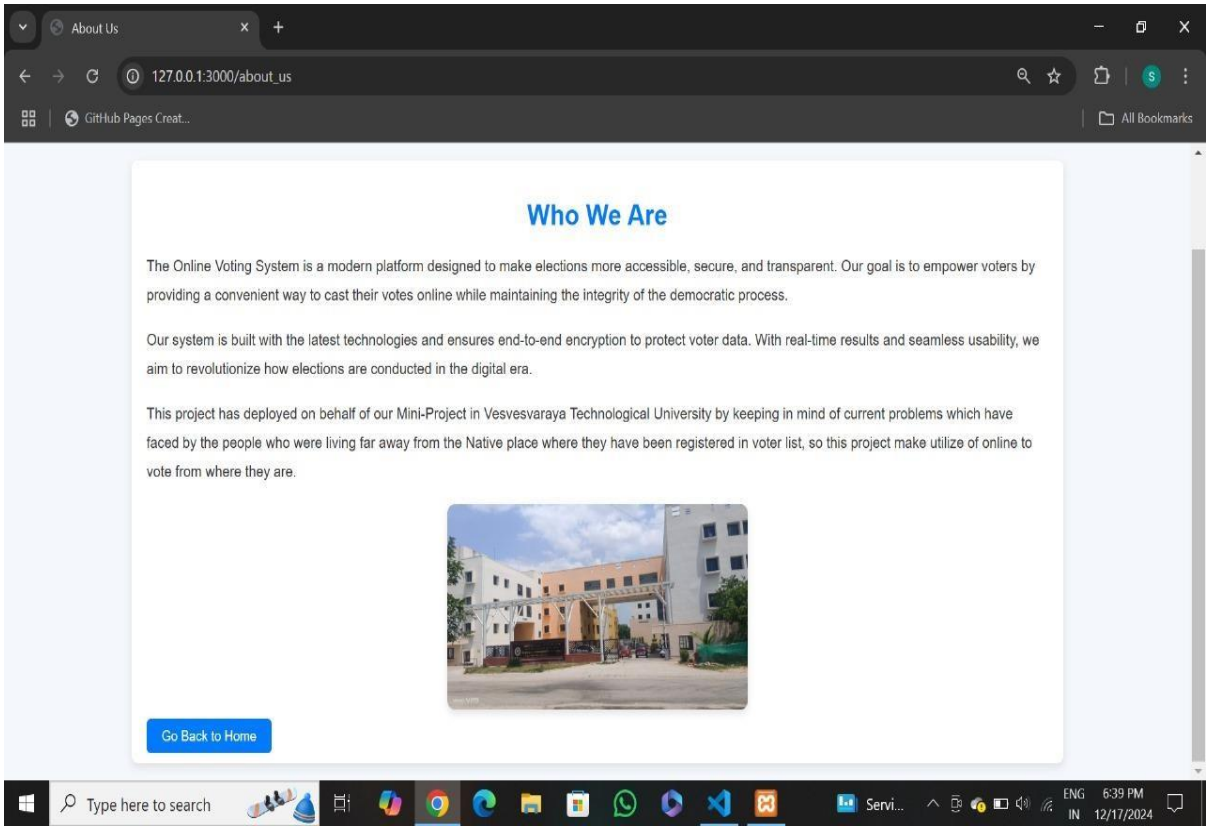


Figure 8.12

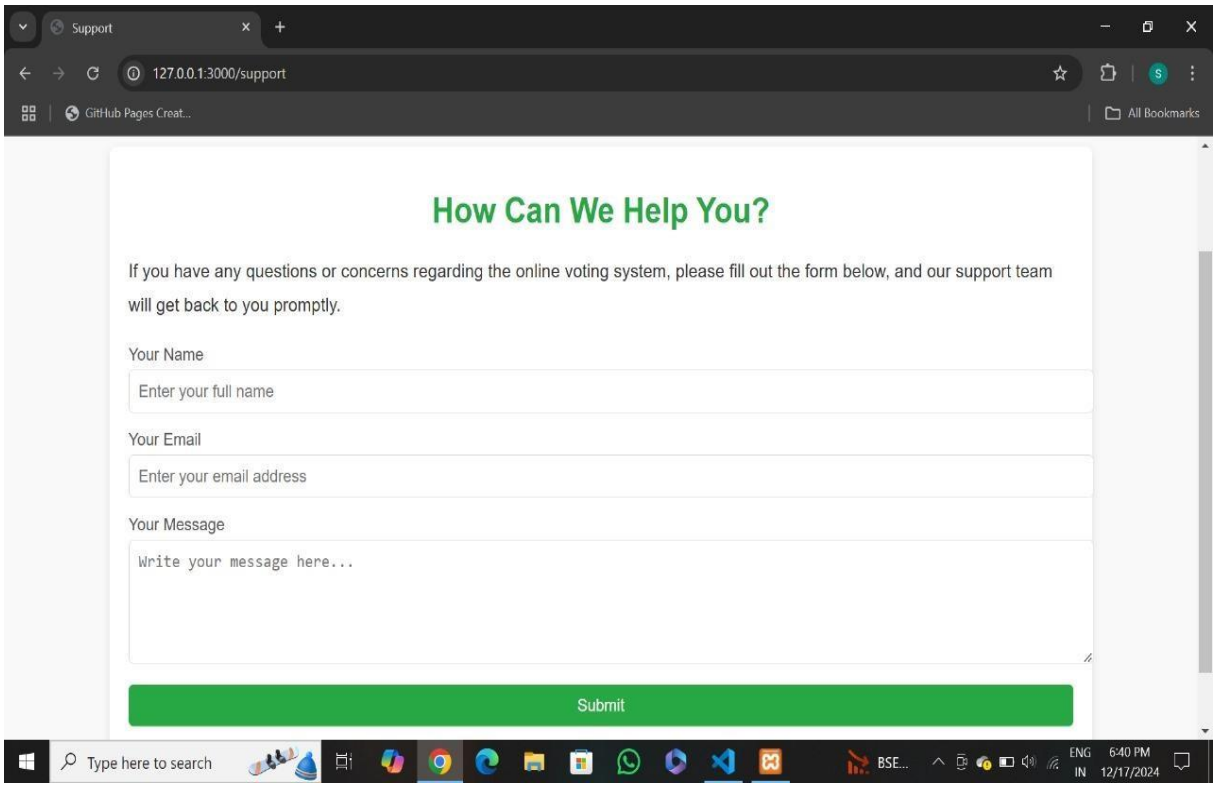

Support
127.0.0.1:3000/support
GitHub Pages Creat...
All Bookmarks
How Can We Help You?
If you have any questions or concerns regarding the online voting system, please fill out the form below, and our support team will get back to you promptly.
Your Name
Enter your full name
Your Email
Enter your email address
Your Message
Write your message here...
Submit
Type here to search
BSE...
ENG
IN
6:40 PM
12/17/2024

Figure 8.13

CHAPTER 9

CONCLUSION

The development of an Online Voting System represents a significant advancement in the electoral process, providing a secure, efficient, and accessible method for conducting elections. This project aims to address the limitations of traditional voting systems by integrating modern technology to enhance voter participation and ensure the integrity of the voting process. It offers a modern solution to electoral challenges by combining security, accessibility, and efficiency. It not only facilitates a more inclusive voting experience but also enhances the reliability of election outcomes. As technology continues to evolve, further improvements and adaptations can be made to this system to address ongoing concerns about security and voter confidence. The future of elections may very well depend on embracing these digital innovations to foster greater democratic participation and integrity in the electoral process.

REFERENCES

1. **An Electronic Voting System: A Case Study of St. Mary's Heritage College**
   https://www.researchgate.net/publication/375962885_AN_ELECTRONIC_VOTING_SYSTEMA_CASE_STUDY_OF_ST_MARYS_HERITAGE_COLLEGE-_KAWEMPE

2. **Online Voting System Using Cloud**
   https://www.researchgate.net/publication/340972420_Online_Voting_System_using_Cloud

3. **Web and Mobile Platforms for Managing Elections Based on IoT and Machine Learning Algorithms**
   https://arxiv.org/abs/2303.09045

4. **Electronic Voting - Overview and Implementations**
   https://en.wikipedia.org/wiki/Electronic_voting

5. **Politics and Technology - Blockchain Voting Platforms**
   https://en.wikipedia.org/wiki/Politics_and_technology

6. **Design and Development of Secure E-Voting Systems**
   https://www.academia.edu/41492694/Design_and_Development_of_Secure_E_Voting_Systems