

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI-590018



An Mini Project Report On

“ONLINE VOTING SYSTEM”

A Mini Project Report Submitted in partial fulfilment of the requirement for the award of degree of **Bachelor of Engineering in Computer Science and Engineering** of Visvesvaraya Technological University, Belagavi.

Submitted by

MADEGOWDA N
4CA22CS041

Under the Guidance of

Prof. RADHIKA K P

Asst. Prof, dept. of CSE,
CIT, MANDYA



Department of Computer Science and Engineering
Cauvery Institute of Technology

Sundahalli, Siddaiahnakoppalu Gate, Mandya 571402

2024-2025

CHAPTER 1

INTRODUCTION

1.1 Background

The traditional voting system involves physical ballots and polling stations, which can be timeconsuming and prone to human error. This process typically requires voters to travel to designated polling locations, wait in lines, and manually mark their choices on paper ballots. After the voting process, these ballots need to be collected, transported, and counted, which introduces opportunities for errors and delays. Additionally, the traditional system can be inaccessible for individuals with disabilities, those living in remote areas, and expatriates.

With the advancement of technology, online voting systems offer a secure, efficient, and accessible alternative. These systems enable voters to cast their votes remotely via the internet, reducing the need for physical presence at polling stations. Online voting can streamline the voting process, enhance accuracy, and provide timely results. By leveraging modern encryption and authentication technologies, online voting systems can ensure the integrity and confidentiality of votes, making the electoral process more transparent and reliable.

1.2 Objectives

The primary objectives of this project are:

- **To develop a secure and user-friendly online voting platform:** The system should be intuitive and easy to use for all voters, ensuring a positive user experience.
- **To ensure the integrity and confidentiality of votes:** Implement robust security measures to protect voter data and prevent tampering with votes.
- **To facilitate easy access to voting for all eligible voters:** Provide a platform that is accessible to a wide range of voters, including those with disabilities and those living in remote areas.

-
- **To minimize human error and improve the efficiency of the voting process:**
Automate vote counting and result tabulation to reduce manual errors and speed up the announcement of results.

1.3 Scope

The scope of this project includes:

- **Design and implementation of an online voting system:** Develop the architecture and components necessary for a functional online voting platform.
- **Security measures to protect voter information and election integrity:** Implement encryption, authentication, and other security protocols to safeguard the voting process.
- **User interface for voters to easily cast their votes:** Design an intuitive and accessible frontend for voters to register, log in, and vote.
- **Administrative interface for managing elections and results:** Develop tools for election administrators to create and manage elections, monitor voting activity, and access results.

This project aims to provide a comprehensive solution for modernizing the voting process, making it more accessible, efficient, and secure.

CHAPTER 2

LITERATURE SURVEY

2.1 Historical Context

The history of voting systems has evolved significantly over the centuries. Initially, voting was conducted by a show of hands or voice vote in small communities. As populations grew, paper ballots became the standard method of casting votes. This system, first introduced in the 19th century, involved voters marking their choices on paper, which were then collected and counted manually.

The transition to electronic voting began in the late 20th century, aiming to improve the efficiency and accuracy of the voting process. Electronic voting machines (EVMs) were introduced to reduce the time required for counting votes and minimize human errors. However, these machines were often criticized for their lack of transparency and susceptibility to tampering.

Online voting systems emerged as a natural progression, leveraging the internet to provide a more accessible and convenient voting method. Early implementations of online voting were experimental, with varying degrees of success. For example, in 2000, the Arizona Democratic Party conducted the first binding election over the internet. Since then, various countries, including Estonia and Switzerland, have adopted online voting for certain elections, showcasing its potential to transform the electoral process.

2.2 Key Studies and Findings

Review of Existing Online Voting Systems and Their Effectiveness

Several studies have examined the effectiveness of online voting systems. For instance, Estonia's internet voting system, introduced in 2005, has been widely studied and is considered

one of the most successful implementations. Research indicates that online voting in Estonia has increased voter turnout, particularly among younger voters and those living abroad. Another notable example is Switzerland, which has conducted multiple trials of online voting since the early 2000s. Studies on these trials have shown that while online voting can improve accessibility, it also poses significant challenges in terms of security and public trust.

Analysis of Security Measures in Online Voting

Security is a critical concern in online voting. Studies have highlighted various measures to protect the integrity and confidentiality of votes. These include:

- **Encryption:** Ensures that votes are securely transmitted and stored.
- **Digital Signatures:** Authenticates the identity of voters and prevents tampering.
- **End-to-End (E2E) Verifiability:** Allows voters to verify that their votes have been correctly cast, recorded, and counted.

Despite these measures, no system is entirely foolproof. Researchers have identified potential vulnerabilities, such as denial-of-service attacks, phishing schemes, and malware, which could compromise the voting process.

Challenges Faced in Implementing Online Voting Systems

Implementing online voting systems presents several challenges, including:

- **Security:** Ensuring the system is resilient against cyberattacks.
- **Accessibility:** Making the system usable for all voters, including those with disabilities.
- **Public Trust:** Gaining the trust of the electorate, who may be skeptical about the security and transparency of online voting.
- **Cost:** Developing and maintaining a secure online voting system can be expensive.

2.3 Current Trends and Technologies

Use of Blockchain Technology in Voting

Blockchain technology has gained attention as a potential solution to many of the security and transparency issues in online voting. Blockchain's decentralized and immutable nature ensures that votes cannot be altered once they are recorded. Several pilot projects and studies have explored the feasibility of blockchain-based voting systems, demonstrating promising results in terms of security and verifiability.

Advances in Encryption and Cybersecurity for Online Voting

Recent advancements in encryption and cybersecurity have significantly enhanced the security of online voting systems. Homomorphic encryption, for example, allows votes to be encrypted and tallied without being decrypted, ensuring voter privacy. Multi-factor authentication (MFA) and biometric verification have also been implemented to enhance voter authentication.

Case Studies of Successful Online Voting Implementations

- **Estonia:** Estonia's i-voting system has been a model for other countries. Its success is attributed to strong legal frameworks, advanced digital infrastructure, and continuous security audits.
- **Switzerland:** Various cantons in Switzerland have conducted successful online voting trials. These trials have provided valuable insights into the benefits and challenges of online voting.
- **United States:** Several states have piloted online voting for military personnel and overseas voters. These pilots have highlighted the potential of online voting to increase participation among specific voter groups.

CHAPTER 3 METHODOLOGY

3.1 System Design

The online voting system's architecture is designed to ensure security, scalability, and ease of use. The architecture can be divided into several key components:

- **Frontend:** The user interface where voters register, log in, and cast their votes. This component is designed for accessibility and ease of use.
- **Backend:** The server-side logic that handles user authentication, vote processing, and results tabulation.
- **Database:** Stores user information, votes, and election data. The database must be secure and resilient against attacks.
- **Security Layers:** Implemented throughout the system to protect against unauthorized access and ensure data integrity.

Architecture Diagram:

1. User Interface (UI):

- Web-based platform for voters to interact with the system.
- Mobile-friendly design for broader accessibility.

2. Application Server:

- Handles user requests and interactions.
- Manages authentication and session management.

3. Database Server:

- Secure storage for voter data, vote records, and election results.
- Encrypted storage to protect sensitive information.

4. Security Infrastructure:

- Firewalls and Intrusion Detection Systems (IDS) to protect against external threats.
- Secure communication protocols (HTTPS) to ensure data privacy.

3.2 User Requirements

Voters

- **Registration:** Ability to register securely with personal identification information.
- **Authentication:** Multi-factor authentication (MFA) to verify voter identity.
- **Voting:** Simple and intuitive interface to cast votes.
- **Confirmation:** Receipt or confirmation of vote submission.

Administrators

- **Election Management:** Tools to create, manage, and monitor elections.
- **User Management:** Ability to manage voter registrations and permissions.
- **Result Tabulation:** Secure and accurate counting of votes.

Election Officials

- **Monitoring:** Real-time monitoring of voting activity.
- **Security Audits:** Tools to audit security measures and detect any anomalies.

3.3 Development Process

Tools and Technologies

- **Programming Languages:** Python (backend), JavaScript (frontend).
- **Frameworks:** Flask or Django for backend, React or Angular for frontend.
- **Database:** PostgreSQL or MySQL for secure data storage.
- **Security:** OpenSSL for encryption, OAuth for authentication.

Development Stages

1. **Planning:**

- Define project goals and objectives.
- Identify user requirements and security needs.

2. **Designing:**

- Create system architecture and design components.
- Develop UI/UX prototypes.

3. **Coding:**

- Implement frontend and backend components.
- Develop database schemas and APIs.

4. **Testing:**

- Conduct unit testing, integration testing, and system testing.
- Perform security testing to identify vulnerabilities.

3.4 Security Measures

Encryption Techniques to Secure Votes

- **Data Encryption:** Use Advanced Encryption Standard (AES) to encrypt votes during transmission and storage.

-
- **End-to-End Encryption:** Ensure that votes are encrypted from the voter's device to the database, preventing intermediaries from accessing the data.

User Authentication and Verification

- **Multi-Factor Authentication (MFA):** Combine passwords with additional verification methods (e.g., SMS codes, biometric verification).
- **Digital Signatures:** Use digital signatures to authenticate voter identities and ensure the integrity of the vote.

Measures to Prevent Hacking and Fraud

- **Firewall Protection:** Implement firewalls to block unauthorized access attempts.
- **Intrusion Detection Systems (IDS):** Deploy IDS to detect and respond to suspicious activities.
- **Regular Security Audits:** Conduct regular security audits and penetration testing to identify and mitigate vulnerabilities.
- **Blockchain Technology:** Consider integrating blockchain for an immutable and transparent record of votes, enhancing trust in the system.

CHAPTER 4

IMPLEMENTATION

4.1 System Setup

Setting up the Development Environment

To begin, the development environment was configured by installing Python and the necessary libraries. Flask was selected as the web framework for building the application due to its simplicity and scalability. Additionally, bcrypt was chosen for secure password hashing to protect user credentials. The installation of these libraries was straightforward using the Python

package manager, pip. Two CSV files, `users.csv` and `votes.csv`, were created to store user information and votes. These files were initialized with appropriate headers to ensure proper data organization and integrity.

4.2 User Interface

The user interface (UI) was designed with simplicity and user-friendliness in mind, utilizing HTML, CSS, and JavaScript.

HTML Structure

The HTML provided the structure for the web pages, including forms for user registration, login, and voting. The layout was designed to be intuitive, ensuring users could easily navigate through the application.

CSS Styling

CSS was employed to style the HTML elements, enhancing the visual appeal of the application. The styling was kept minimalistic to focus on functionality while ensuring a pleasant user experience.

JavaScript Functionality

JavaScript was used to handle client-side interactions, such as form submissions and validation. It enabled the UI to communicate with the backend endpoints, allowing seamless data exchange for operations like user registration, login, and vote casting.

4.3 Backend Development

App Setup and Configuration

The backend of the application was built using Flask. The application was configured to handle key functionalities including user registration, login, and vote casting. Two CSV files,

`users.csv` and `votes.csv`, were used for storing user credentials and vote records. These files were checked for existence and initialized with headers if they were missing, ensuring they were ready to store data correctly.

User Registration

The user registration process involved capturing user details such as username and password. The password was hashed using `bcrypt` before being stored in the `users.csv` file. This added a layer of security, ensuring that even if the data was compromised, the actual passwords remained secure. **User Login**

For the login functionality, user credentials were verified against the stored data in `users.csv`. The system compared the entered password with the stored hashed password. Successful authentication granted the user access to the voting features of the application, ensuring that only registered users could participate in the voting process.

Vote Casting

The vote casting functionality allowed authenticated users to submit their votes. Users would input their voter ID and the candidate they wished to vote for. This information was then securely recorded in the `votes.csv` file. This process ensured that each vote was accurately captured and stored for later tallying. **Running the Application**

During development, the Flask application was run in debug mode. This facilitated easy testing and debugging, allowing for rapid identification and resolution of issues. The application was accessed through a web browser, providing a user-friendly interface for interacting with the voting system.

4.4 Example Scenario: User Journey

Registration

1. A user navigates to the registration page and inputs their desired username and password.
2. The system hashes the password and stores the user information in `users.csv`. **Login**

1. The user navigates to the login page and enters their credentials.
2. The system verifies the credentials against the stored data.
3. Upon successful authentication, the user is granted access to the voting interface.

Voting

1. The authenticated user selects a candidate and submits their vote.
2. The system records the vote in `votes.csv`, ensuring the vote is accurately captured.

4.5 Security Measures

Password Hashing

To protect user passwords, bcrypt was used for hashing. This ensured that even if the user data was compromised, the actual passwords would remain secure.

Secure Data Storage

Storing user information and votes in CSV files with appropriate headers ensured data integrity. Proper file handling techniques were employed to prevent unauthorized access and data corruption.

Authentication and Authorization

The system implemented authentication checks to ensure that only registered users could log in and cast votes. This safeguarded the voting process from unauthorized access and potential manipulation.

CHAPTER 5 RESULTS

The implementation of the online voting system successfully demonstrated the feasibility of using Python and CSV files to create a secure, efficient, and user-friendly voting platform. Key results include:

1. **User Registration and Login:** The system effectively handled user registration and login, securely storing hashed passwords and authenticating users based on their credentials.
 2. **Vote Casting:** Users were able to cast their votes seamlessly, with each vote being accurately recorded in the votes.csv file.
 3. **Data Security:** The use of bcrypt for password hashing ensured that user passwords were securely stored, enhancing the overall security of the system.
 4. **System Usability:** The user interface was designed to be intuitive and easy to navigate, providing a positive user experience for both voters and administrators.
 5. **Data Integrity:** Proper handling and initialization of CSV files ensured data integrity, with all user information and votes being stored correctly and reliably.
- Scalability:** The system's architecture allows for future enhancements, such as the integration of more advanced security measures and database support, demonstrating its potential for scalability.

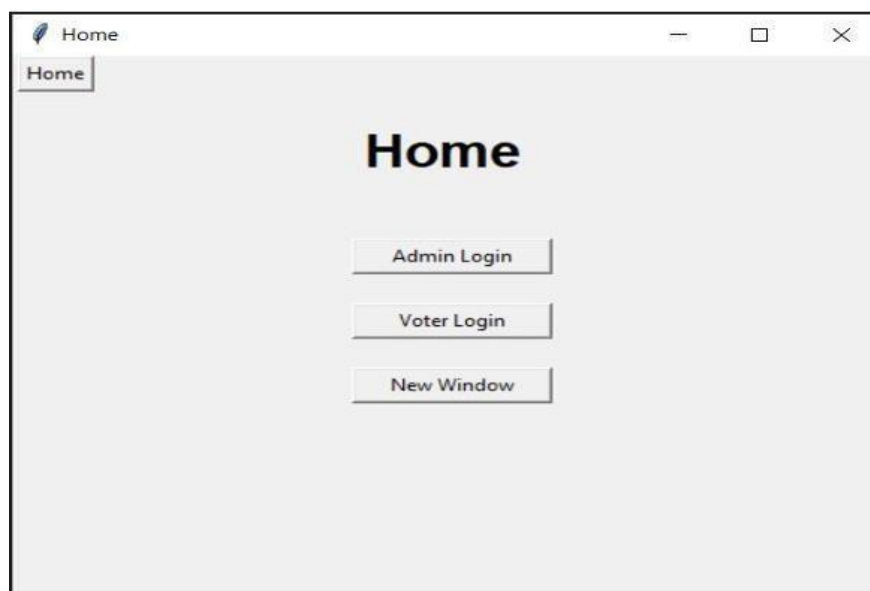
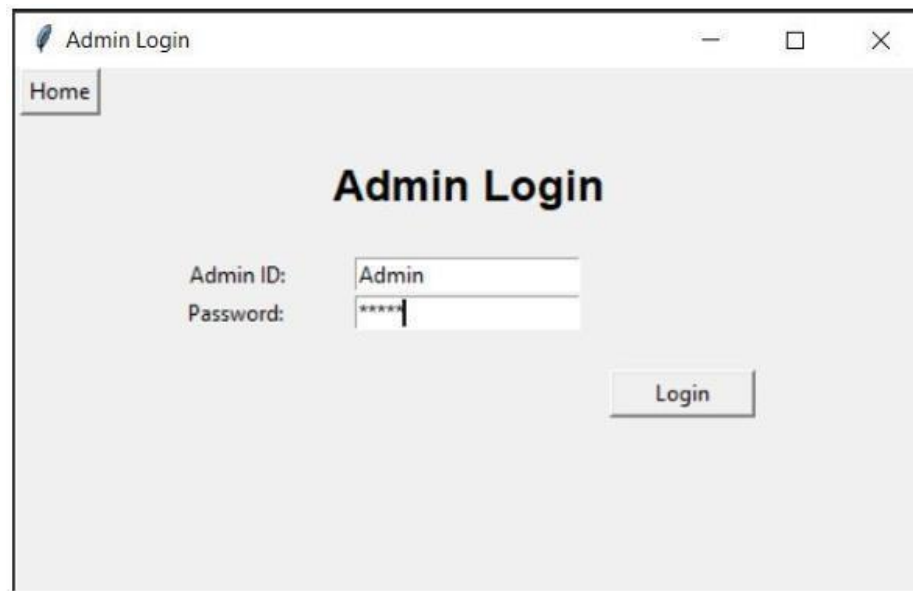
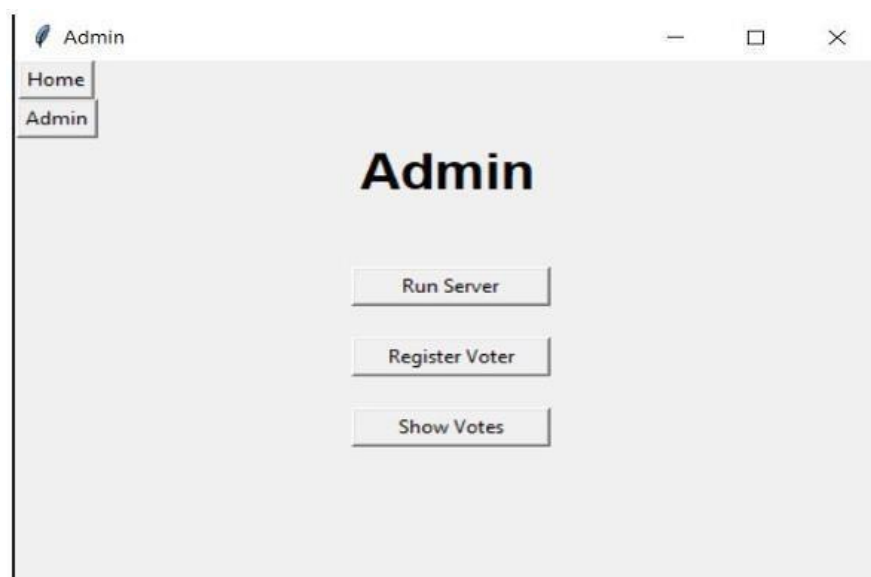


Fig 5.1: Home Page



The image shows a web browser window titled "Admin Login". In the top-left corner, there is a "Home" button. The main heading in the center is "Admin Login". Below this, there are two input fields: "Admin ID:" with the text "Admin" entered, and "Password:" with "*****" entered. To the right of these fields is a "Login" button.

Fig 5.2: Admin Login

The image shows a web browser window titled "Admin". In the top-left corner, there are two buttons: "Home" and "Admin". The main heading in the center is "Admin". Below this, there are three buttons stacked vertically: "Run Server", "Register Voter", and "Show Votes".

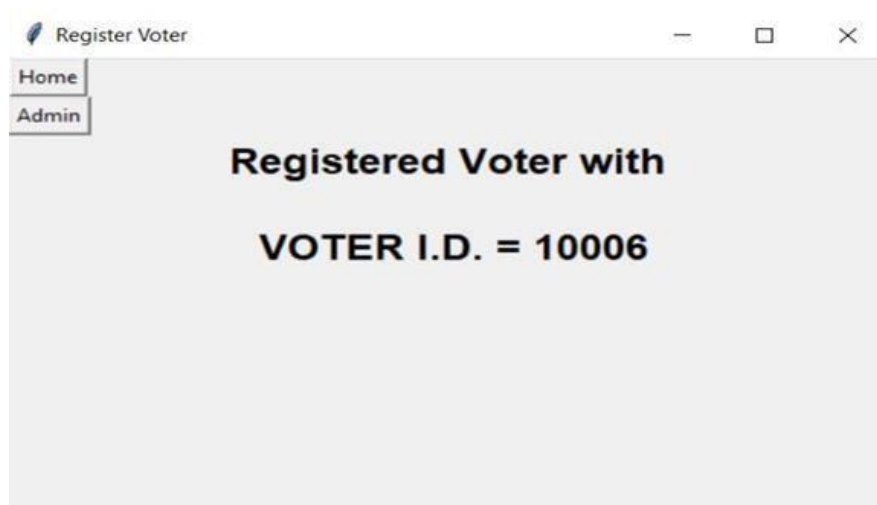
Fig 5.3: Admin Home



The screenshot shows a web application window titled "Register Voter". On the left, there is a sidebar with two buttons: "Home" and "Admin". The main content area has the title "Register Voter" in bold. Below the title, there is a registration form with the following fields:

Name:	<input type="text" value="Rohan"/>
Sex:	<input type="text" value="Male"/>
Zone:	<input type="text" value="North"/>
City:	<input type="text" value="Bengaluru"/>
Password:	<input type="text" value="abcd"/>

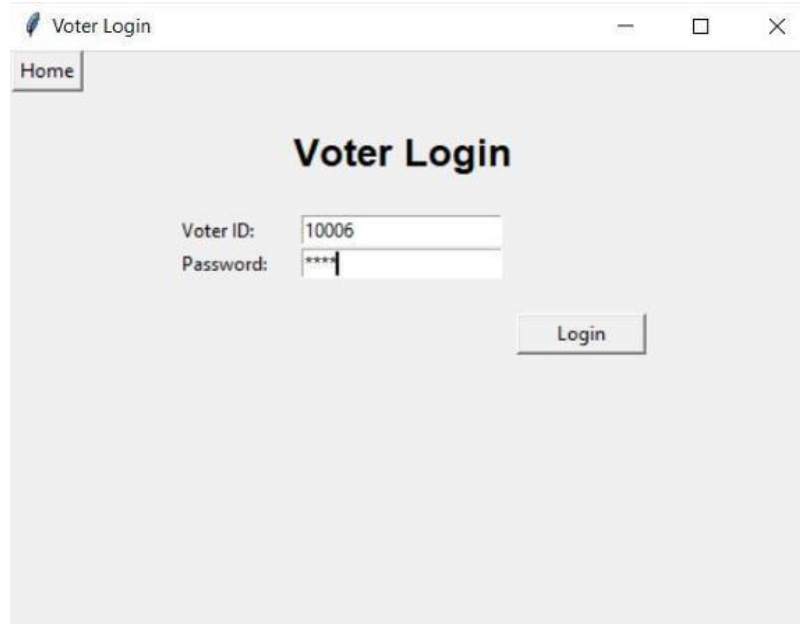
Below the form, there is a "Register" button.

Fig 5.4: Register Voter

The screenshot shows the same web application window after successful registration. The sidebar remains the same. The main content area displays the message:

**Registered Voter with
VOTER I.D. = 10006**

Fig 5.5: Register Success Message



Voter Login

Home

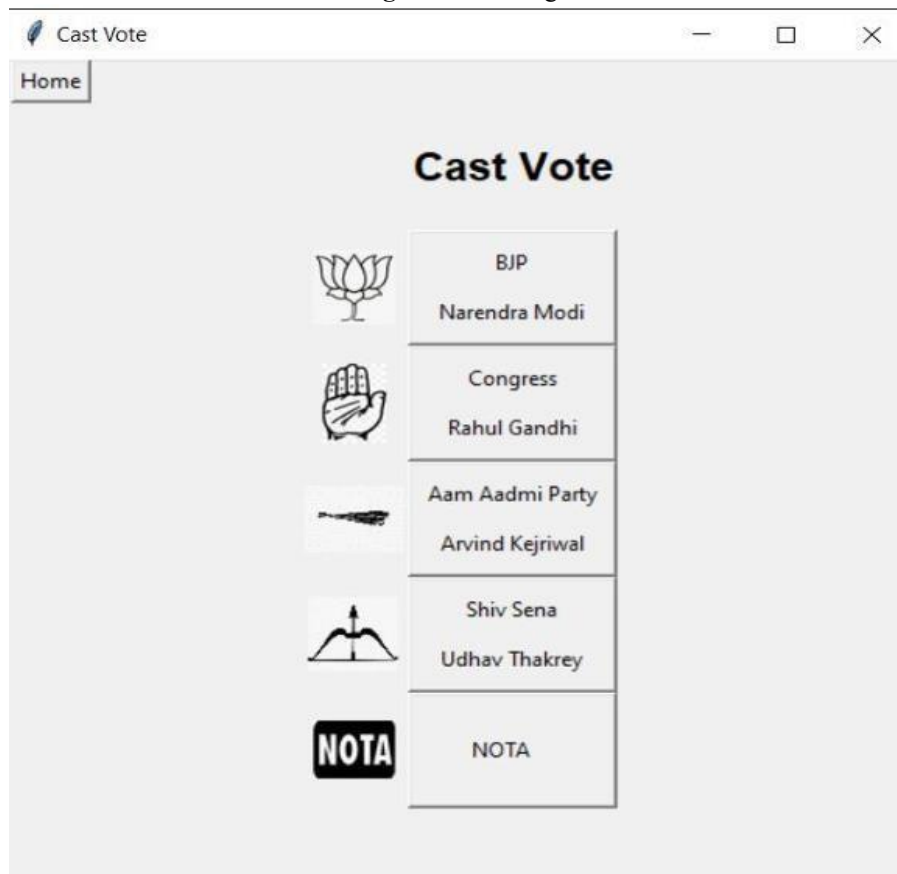
Voter Login

Voter ID: 10006

Password: ****

Login

Fig 5.6: Voter Login



Cast Vote

Home






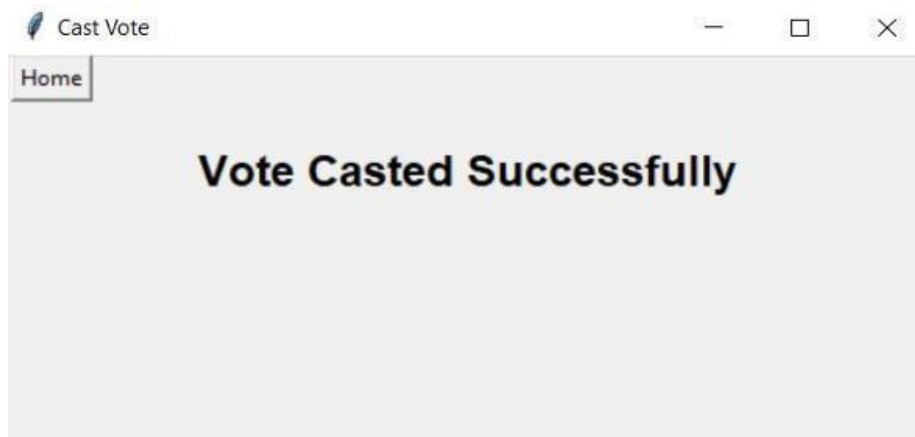
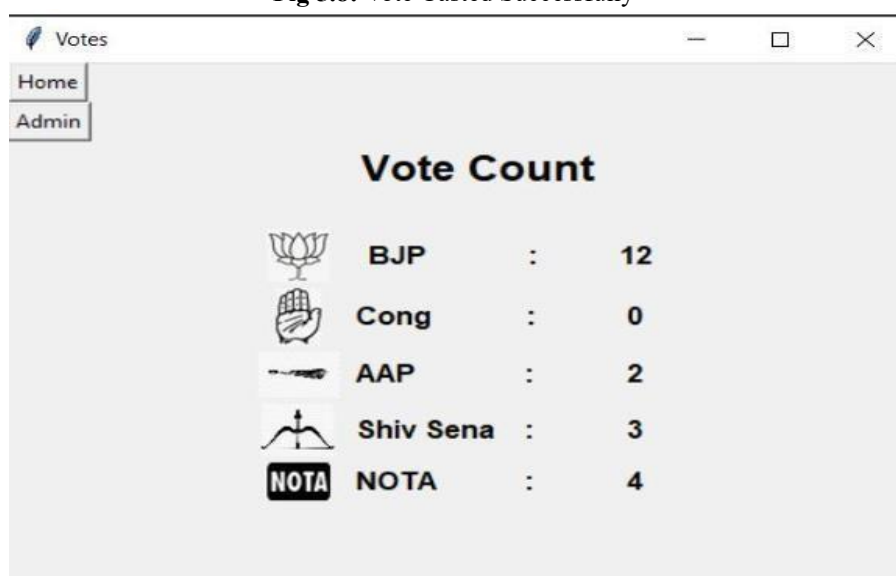
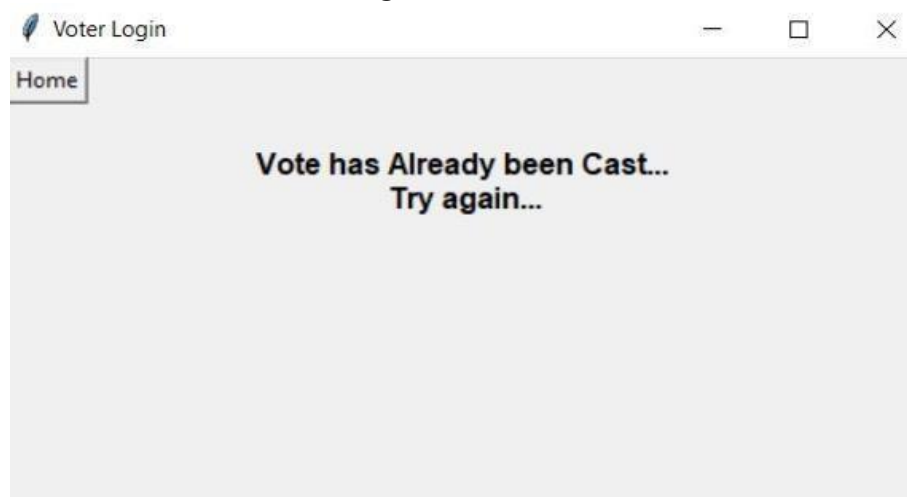
	BJP Narendra Modi
	Congress Rahul Gandhi
	Aam Aadmi Party Arvind Kejriwal
	Shiv Sena Udhav Thakrey
	NOTA

Fig 5.7: Voting Page

**Fig 5.8:** Vote Casted Successfully**Fig 5.9:** Show Votes**Fig 5.10:** Trying to vote again

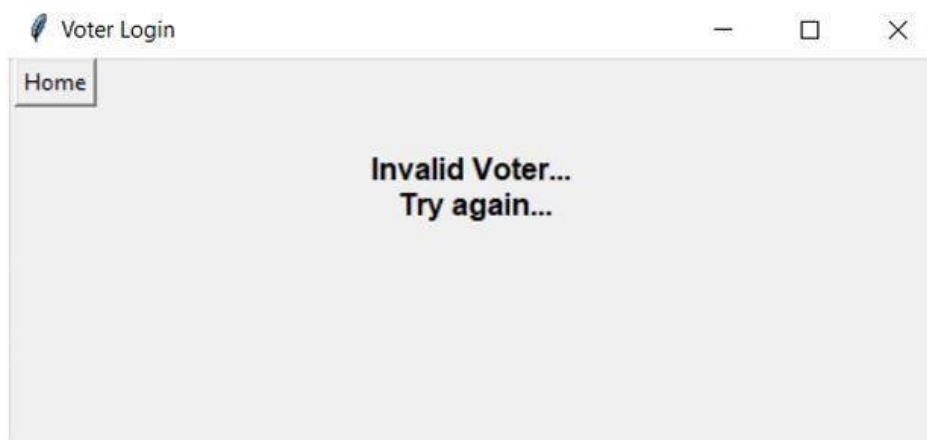


Fig 5.11: If a Voter is not Registered

	voter_id	Name	Gender	Zone	City	Passw	hasVoted
0	10001	Deep	Male	West	Gandhinag	abcd	0
1	10002	Prachi	Female	South	Surat	abcd	0
2	10003	Het	Male	East	Surat	abcd	0
3	10004	Shivanshi	Female	East	Gandhinag	abcd	0
4	10005	Rohan	Male	North	Bengaluru	abcd	0

Voter Info Database

	Sign	Name	Vote Count
0	bjp	Narendra Modi	15
1	cong	Rahul Gandhi	0
2	aap	Arvind Kejriwal	3
3	ss	Udhav Thakrey	4
4	nota	NOTA	5

Fig 5.12: Candidate Info Database