A

MINI PROJECT REPORT

On

ONLINE VOTING SYSTEM

A dissertation submitted in partial fulfillment of the requirement for the award

in the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

B. Supriya (22UP1A0519)

Under the esteemed guidance of

Mr. P. Masoom Basha

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN

AN AUTONOMOUS INSTITUTION

Kondapur (V), Ghatkesar (M), MedchalDist– 501 301.

Affiliated to Jawaharlal Nehru Technological University, Hyderabad.

2022-2026

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
(An Autonomous Institution)
(Sponsored by Lavu Educational Society)
[Affiliated to JNTUH, Hyderabad & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal –Malkajgiri (D)- 501301.Phone:9652910002/3

Department of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled "**Online voting System**" submitted by **B. Shivani (22UP1A0529), B. Supriya (22UP1A0519), CH. Sathvika (22UP1A0534)** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**, **Vignan's Institute of Management and Technology for Women** is a record of bonafide work carried by them under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or institute for the award of any degree.

PROJECT GUIDE

Mr. P. Masoom Basha

(Associate Professor)

Internal Guide

THE HEAD OF DEPARTMENT

Mrs. M. Parimala

(Associate Professor)

External Examiner

VIGNAN'S INSTITUTE OF MANAGEMENT AND TECHNOLOGY FOR WOMEN
(An Autonomous Institution)
(Sponsored by Lavu Educational Society)
[Affiliated to JNTUH, Hyderabad & Approved by AICTE, New Delhi]
Kondapur (V), Ghatkesar (M), Medchal –Malkajgiri (D)- 501301.Phone:9652910002/3

DECLARATION

We hereby declare that the results embodied in the project entitled "**online Voting System**" is carried out by us during the year 2025-2026 in partial fulfillment of the award of Bachelor of Technology in Computer Science and Engineering from Vignan's Institute of Management and Technology for Women is an authentic record of our work under the guidance of DR. G. Rajesh. We have not submitted the same to any other institute or university for the award of any other Degree.

B. Shivani (22UP1A0529)

B. Supriya (22UP1A0519)

CH. Sathvika (22UP1A0534)

ACKOWLEDGEMENT

We would like to express sincere gratitude to **Dr G. APPARAO NAIDU**, **Principal, Vignan's Institute of Management and Technology for Women** for his timely suggestions which helped us to complete the project in time.

We would also like to thank our madam **Mrs. M. Parimala, Head of the Department and Associate Professor, Computer Science and Engineering** for providing us with constant encouragement and resources which helped us to complete the project in time.

We would also like to thank our Project guide **Mr Dr.G. Rajesh,Associate professor, Computer Science and Engineering**, for providing us with constant encouragement and resources which helped us to complete the project in time with his valuable suggestions throughout the project. We are indebted to him for the opportunity given to work under his guidance.

Our sincere thanks to all the teaching and non-teaching staff of Department of Computer Science and Engineering for their support throughout our project work.

B. Shivani (22UP1A0529)
B. Supriya (22UP1A0519)
CH. Sathvika (22UP1A0534)

INDEX

LIST OF FIGURES

1.ABSTRACT

The traditional method of conducting elections, which involves physical ballot papers and manual vote counting, often faces challenges such as long queues, human errors, high operational costs, and in some cases, electoral fraud. With the rapid advancement of digital technologies, there is a growing need for a more efficient, secure, and accessible alternative to conventional voting. This project introduces the design and implementation of an Online Voting System, a web-based platform that enables eligible users to cast their votes securely from any location via the internet.

The primary goal of the system is to streamline the voting process while ensuring key principles of democratic elections such as confidentiality, integrity, and transparency. The system features a secure user authentication mechanism, typically based on unique identifiers such as voter ID and passwords, with the potential to incorporate biometric verification or One-Time Passwords (OTP) for enhanced security. Upon successful authentication, users are granted access to a digital ballot, where they can cast their votes for their preferred candidates.

The backend of the system is designed to securely record and encrypt each vote, ensuring that the data remains tamper-proof and anonymous. Real-time vote counting is implemented to provide instant results once voting concludes, significantly reducing the delay associated with manual tallying.

This Online Voting System is developed using modern web technologies and adheres to essential cybersecurity standards to protect against threats such as vote tampering, multiple voting, and denial-of-service attacks. The system can be adapted for various types of elections, including student body elections, corporate decision-making, and local or national government elections with further scalability and legal validation.

In conclusion, the proposed system not only improves the efficiency and accessibility of the electoral process but also builds public trust in digital voting by ensuring a high level of security, accuracy, and transparency. It represents a significant step toward the digital transformation of electoral systems in the modern age.

CHAPTER 2

2. INTRODUCTION

2.1 OBJECTIVE

In any democratic process, voting is a critical mechanism that allows individuals to express their preferences and participate in decision-making. Traditional voting methods, which involve physical polling stations and paper ballots, are often time-consuming, resource-intensive, and prone to various challenges such as human error, long queues, mismanagement, and even electoral fraud. As societies increasingly rely on digital technologies for efficiency and accessibility, there is a growing interest in implementing online voting systems as a modern alternative.

An Online Voting System is a web-based application that enables voters to cast their votes electronically from any location with internet access. This eliminates the need for physical presence at polling booths and makes the voting process more convenient and accessible, especially for those who may be traveling, physically challenged, or living in remote areas. Such systems also have the potential to increase voter turnout by making the process faster and more user-friendly.

The core functions of an online voting system typically include voter registration, secure authentication, vote casting, vote storage, and result generation. The system must ensure that each vote is confidential, counted accurately, and cast only once per voter. Therefore, data security, user verification, and system integrity are essential components of its design. Encryption, secure databases, and authentication mechanisms (like voter IDs, passwords, OTPs, or biometric data) are employed to ensure a safe and trustworthy process.

portal for voters, and real-time monitoring of results. This system can be deployed for various types of elections, such as student body elections, organizational voting, or local community decision-making.

2.2 Existing System

Existing online voting system refers to the traditional or currently implemented methods for managing elections through digital platforms. Here's a detailed explanation:

1. Traditional (Manual) Voting System

Before online systems, voting was mostly:

- Paper-based ballots
- Physical polling booths
- Manual voter verification
- Slow result compilation

2. Challenges in Existing Systems

- Security risks (hacking, fraud)
- Digital illiteracy in rural areas
- Connectivity issues
- Identity verification loopholes
- Lack of transparency in some platforms

2.2.1 Limitations of Existing System

Despite the increasing adoption of technology in various sectors, many existing voting systems—both traditional and early digital prototypes—still face several limitations that impact their efficiency, reliability, and security. The following are key limitations of the existing systems in the context of online voting:

1. Lack of Robust Security

Most existing systems do not provide adequate protection against cyber threats. Vulnerabilities in authentication methods, data transmission, and storage can lead to potential hacking, vote tampering, or data breaches, undermining the trust in election outcomes.

2. Inadequate Voter Authentication

Current systems often rely on simple login mechanisms like user IDs and passwords, which are prone to theft or misuse.

Without multi-factor or biometric authentication, ensuring that only eligible voters participate becomes difficult.

3. Poor Transparency and Verifiability

Many online voting platforms fail to offer a transparent process where voters can verify that their vote was cast and counted correctly. This lack of auditability creates doubt in the minds of users regarding the credibility of the election.

4. Limited Accessibility

Existing systems often lack user-friendly interfaces and mobile compatibility, making it difficult for people with disabilities, low digital literacy, or limited internet access to participate effectively in the voting process.

2.3 Proposed System

To overcome the limitations of existing voting methods and early online systems, the proposed Online Voting System is designed to provide a secure, efficient, transparent, and user-friendly platform for conducting elections over the internet. This system aims to enhance voter participation, minimize fraud, and ensure the integrity of the electoral process by leveraging modern web technologies and security protocols.

Key Features of the Proposed System

1. Secure User Authentication
    - Voters will be authenticated using unique credentials such as voter ID and password.
    - Additional security layers like OTP (One-Time Password), email verification, or biometric verification (optional) can be added to prevent unauthorized access.
2. User-Friendly Interface
    - The system will feature a clean, intuitive interface for both voters and administrators.
    - It will be accessible on desktops, tablets, and mobile devices to ensure wide usability.

3. Vote Confidentiality and Anonymity
    - Votes are encrypted and stored in a secure database to maintain voter anonymity.
    - The system separates voter identity from the actual vote to comply with secret ballot requirements.
4. One-Voter-One-Vote Logic
    - The system ensures that each registered user can cast only one vote.
    - Once a vote is cast, the user's session is locked for that election, preventing duplication or re-voting.
5. Real-Time Vote Counting and Result Display
    - Votes are counted automatically in real-time and displayed only after the voting period ends.
    - Results are stored securely and can be audited if required.
6. Admin Panel for Election Management
    - An administrative dashboard allows election officials to add candidates, schedule elections, register voters, and monitor voting activity.
    - Logs of all system activities are maintained for transparency.
7. Data Encryption and Security
    - All data transmissions are encrypted using secure protocols (e.g., HTTPS, SSL/TLS).
    - Sensitive information is stored using hashed and encrypted formats to prevent breaches.
8. Scalability and Reliability
    - The system is designed to handle a high number of concurrent users and can be scaled based on the size of the election.
    - Redundant systems and regular backups will be implemented to ensure data integrity and availability.

2.3.1 Advantages of Proposed System

The proposed Online Voting System offers several key advantages over traditional and existing digital voting methods. It enhances the voting process through improved accessibility, security, transparency, and efficiency.

1. Enhanced Security

- Implements secure authentication methods such as voter ID, password, and optional OTP or biometric verification.
- Uses encryption to protect vote data and prevent tampering.
- Ensures only eligible voters can access the system and cast their vote.

2. Improved Accessibility

- Allows voters to participate from any location using a device with internet access.
- Especially beneficial for people in remote areas, persons with disabilities, and those unable to travel.
- Supports cross-platform compatibility (desktop, tablet, mobile).

3. Time and Cost Efficiency

- Reduces the need for physical polling stations, paper ballots, and manual vote counting.
- Minimizes the resources required for staffing, logistics, and materials.
- Results are generated automatically, saving time and reducing human error.

4. Real-Time Monitoring and Instant Results

- Enables real-time tracking of voter turnout and voting activity.
- Votes are counted automatically, and results are available immediately after the voting period ends.

5. Transparency and Trust

- Includes an auditable system log and vote-tracking mechanisms to enhance transparency.
- Voters can be confident that their votes are counted correctly, even if anonymously.

6. Prevention of Fraud and Duplicate Voting

- The system ensures one vote per person using strong voter authentication and vote-locking mechanisms.
- Prevents multiple voting or impersonation.

7. Administrative Efficiency

- Simplifies election management for administrators through a dedicated dashboard.
- Admins can add/remove voters, schedule elections, view statistics, and generate reports easily.

8. Environmental Benefits

- Reduces the use of paper and other physical materials, supporting eco-friendly and sustainable practices.

CHAPTER 3

LITERATURE SURVEY

A literature survey is essential to understand the current landscape of online voting systems, including existing technologies, methodologies, limitations, and innovations. Numerous studies and system implementations have been explored over the past two decades to modernize electoral processes through digital means. Below is a summary of key research and developments in the domain of online voting systems:

1. Traditional vs. Electronic Voting Systems

Early studies focused on the transition from paper-based voting to electronic systems (EVMs). While these systems improved vote counting efficiency, they still required physical presence at polling stations. Researchers emphasized that although electronic voting machines reduced counting time, they lacked transparency and were vulnerable to tampering if physical security was compromised.

2. Web-Based Online Voting Platforms

Web-based systems introduced the idea of remote voting via the internet. Systems such as Estonia's i-Voting platform (in use since 2005) were among the first successful government-level implementations. Studies have shown that while these systems improve convenience, they also raise concerns about vote-buying, malware, and voter coercion.

- *Reference:* Heiberg, S., Martens, T., & Vinkel, P. (2017). "The security architecture of the Estonian Internet voting system."

3. Security and Cryptographic Protocols

Research has been heavily focused on enhancing the security of online voting using cryptographic methods. Protocols such as Homomorphic Encryption, Zero-Knowledge Proofs, vote integrity, confidentiality, and verifiability.

- *Reference:* Adida, B. (2008). "Helios: Web-based open-audit voting." USENIX Security Symposium.

These protocols allow votes to be encrypted and tallied without decrypting individual ballots, preserving voter privacy while ensuring that the outcome is correct and auditable.

4. Blockchain-Based Voting Systems

Recent innovations propose blockchain as a solution to many traditional issues in online voting. Blockchain provides a decentralized, tamper-proof ledger, making it theoretically impossible to alter votes once recorded. However, scalability, anonymity, and voter accessibility remain challenges.

5. Voter Authentication and Biometrics

Authentication remains a significant concern in online voting systems. Studies have explored various techniques like biometric verification, digital signatures, and multi-factor authentication to ensure that only eligible voters can cast their votes.

These techniques, while more secure, raise issues related to privacy, infrastructure cost, and accessibility.

CHAPTER 4

SYSTEMANALYSIS

System analysis is a critical phase in the software development life cycle (SDLC) that involves examining the existing problems, understanding system requirements, and designing effective solutions. For the Online Voting System, the goal is to create a secure, accessible, and user-friendly platform for conducting elections digitally. This section covers the analysis of the current system, user requirements, system requirements, and functional and non-functional aspects.

1. Existing System Overview

Traditional voting systems rely heavily on physical presence, paper ballots, and manual counting. This leads to several challenges, including:

- Time-consuming vote counting
- Risk of human error and manipulation
- High cost of logistics and manpower
- Limited accessibility for remote or disabled voters
- Delayed results and transparency issues

Some institutions have adopted basic electronic voting systems, but many lack robust security, proper authentication, and audit trails, making them vulnerable to fraud or mismanagement.

2. Proposed System Goals

The proposed Online Voting System aims to solve the limitations of the existing system by:

- Enabling secure remote voting over the internet
- Providing real-time vote tracking and result generation
- Preventing duplicate or unauthorized voting
- Ensuring voter anonymity and data confidentiality
- Improving accessibility and user experience

3. Functional Requirements

- User Registration and Login
- Voter Authentication (e.g., Voter ID + OTP)
- Secure Vote Casting Interface
- One-vote-per-user enforcement
- Vote Encryption and Secure Storage
- Real-time Vote Counting and Result Display
- Admin Dashboard for Election Management
- Candidate Management (Add/Edit/Delete)
- Voting Timeline Control (start/end of voting period)
- System Logs and Audit Trails

4. Non-Functional Requirements

- Security: Data encryption (SSL, hash-based password storage), secure session handling
- Scalability: Ability to handle thousands of concurrent users
- Reliability: 24/7 uptime during the election period
- Usability: Intuitive UI with responsive design
- Accessibility: Support for users with disabilities (e.g., screen readers, keyboard navigation)
- Performance: Fast response times even during peak traffic
- Compatibility: Cross-platform (desktop, tablet, mobile)

5. System Modules

- User Module: Registration, login, authentication, vote casting
- Admin Module: Candidate/voter management, result generation, analytics
- Voting Module: Ballot display, vote encryption, vote submission
- Result Module: Real-time vote tallying and secure result publication
- Security Module: Data encryption, session control, login security
- Notification Module: Email/SMS alerts for registration, voting status, etc.

6. System Architecture (High-Level Overview)

- Frontend: User interface (HTML, CSS, JavaScript, React/Angular)
- Backend: Business logic and APIs (Python/PHP/Node.js)
- Database: Relational or NoSQL database (e.g., MySQL, MongoDB)
- Security Layer: HTTPS, SSL, JWT tokens or session management
- Optional: Blockchain integration for immutable vote records

4.1 Purpose

The purpose of the Online Voting System is to provide a secure, transparent, and efficient digital platform that allows eligible users to cast their votes remotely.through the internet. This system is designed to overcome the limitations of traditional paper-based and manual voting processes, such as long queues, human error, delayed results, and limited accessibility.

The system aims to:

- Enhance accessibility by allowing voters to participate from any location, including remote or rural areas.
- Improve security by using encryption and authentication methods to protect voter data and prevent fraud.
- Ensure transparency and trust by providing verifiable and tamper-proof vote counting.
- Increase efficiency by automating vote collection, counting, and result generation.
- Support scalability for elections of varying sizes—from small institutions (like schools or organizations) to large-scale public elections.

By fulfilling these goals, the system contributes to a more inclusive, reliable, and future-ready electoral process.

4.2 Scope

The scope of the **Online Voting System** defines the boundaries and functionalities of the system to be developed. This system is intended to serve as a secure, user-friendly platform for conducting elections over the internet, allowing registered users to vote from any location with internet access.

In-Scope (What the system will do):

- Enable secure user registration and login for eligible voters.
- Authenticate voters using credentials such as Voter ID and OTP.
- Allow each voter to cast **only one vote** during a defined voting period.
- Encrypt and securely store all votes to ensure privacy and integrity.
- Provide an intuitive web interface for voters and administrators.
- Allow administrators to:
    - Add/manage candidates and voters.
    - Schedule election and end times.
    - Monitor voting activity in real-time.
    - View and publish election results immediately after voting closes.
- Maintain logs and audit trails for transparency and accountability.

Out-of-Scope (What the system will not do):

- Integration with national-level ID or biometric databases (unless extended).
- Detection or prevention of external malware on the voter's device.
- Offline voting or voting without an internet connection.
- Legal or regulatory enforcement of election laws.
- Election administrators or committee members.

4.3 Feasibility Study

A feasibility study is conducted to evaluate the viability of the proposed Online Voting System from various perspectives. It helps determine whether the system can be successfully developed, deployed, and maintained within given constraints such as time, cost, and technology.

The feasibility study covers the following aspects:

1. Technical Feasibility

The system is technically feasible with the use of widely available technologies such as:

- Frontend: HTML, CSS, JavaScript, React/Angular
- Backend: PHP, Python, Node.js
- Database: MySQL, MongoDB
- Security: HTTPS, SSL, encrypted login, and vote encryption

Cloud-based infrastructure and modern web development frameworks make implementation possible with scalable performance and secure architecture.

2. Operational Feasibility

The system is easy to use for both voters and administrators:

- Simple and intuitive user interface
- Accessible on mobile and desktop devices
- Minimal training required for end users
- Reduces administrative effort in managing elections

Operational tasks such as voter registration, candidate setup, and result management can be done efficiently through the admin panel.

3. Economic Feasibility

The development cost is moderate and mainly includes:

- Developer time or third-party service (if outsourced)
- Hosting and domain charges
- Minimal maintenance costs

Compared to the recurring costs of traditional voting (paper, logistics, personnel), the online system is cost-effective in the long run.

4. Legal Feasibility

For private organizations, schools, and clubs, online voting is generally legally acceptable. However, for governmental or public elections, legal frameworks and data protection laws must be considered.

4.3.1 Economic Feasibility

Economic feasibility evaluates whether the proposed Online Voting System is financially viable—meaning that the benefits gained from the system justify the investment and ongoing costs involved in its development, deployment, and maintenance.

1. Development Costs

The development of the Online Voting System may include:

- Human resources: Salaries or fees for software developers, UI/UX designers, and testers.
- Software tools: Most tools and frameworks used (e.g., HTML, CSS, JavaScript, PHP, Python) are open-source or free, minimizing licensing costs.
- Time investment: Estimated 4–8 weeks of development depending on system complexity.

2. Deployment and Operational Costs

- Hosting and Domain: Web hosting and domain registration
- Server Infrastructure: Cloud hosting (like AWS, Digital Ocean, or Firebase) depending on traffic volume.
- Maintenance: Minor costs for updates, backups, and system monitoring.
- Support Staff: If required, minimal administrative training and user support costs.

3. Cost Savings Compared to Traditional Voting

- Eliminates printing, transportation, ballot box handling, and storage.
- Reduces the need for physical polling stations and on-ground staff.
- Speeds up result processing, saving hours/days of manpower.

4. Long-Term Benefits

- Scalability: Once developed, the system can be reused across multiple elections with minimal additional cost.
- Efficiency: Saves time and effort for both voters and administrators.
- Accessibility: Increases voter participation, especially in remote or underserved areas.

4.3.2 Technical Feasibility

Technical feasibility assesses whether the proposed Online Voting System can be successfully developed and implemented using current technologies, tools, and available technical resources. It examines whether the system requirements align with modern computing environments and whether the necessary skills and infrastructure are accessible.

1. Availability of Technology

The Online Voting System can be developed using widely available, well-supported technologies:

- **Frontend:** HTML, CSS, JavaScript, Bootstrap, React or Angular (for responsive user interface)
- **Backend:** PHP, Python (Flask/Django), or Node.js (Express) for server-side logic
- **Database:** MySQL, PostgreSQL, or MongoDB for storing user data and votes
- **Security Tools:** SSL/TLS for secure data transmission, hashing algorithms for password storage, and encryption for vote data

2. System Compatibility

- Works across platforms (Windows, macOS, Linux)
- Compatible with all modern browsers (Chrome, Firefox, Safari, Edge)
- Mobile-responsive interface for smartphones and tablets
- Can be hosted on cloud platforms like AWS, Heroku, or Firebase

3. Security Implementation

The system's design supports essential security features:

- **User authentication** using Voter ID and OTP/email verification
- **Data protection** using HTTPS and backend encryption
- **Secure database** to prevent SQL injection, XSS, CSRF, etc.
- **Audit logs** to monitor and detect suspicious activity

4. Developer Skills and Resource Availability

The required skills to build the system (web development, database management, API development, cybersecurity) are commonly available among student developers or IT professionals. Open-source tools and documentation reduce learning and implementation time.

4.3.3 Social Feasibility

Social feasibility assesses how well the proposed Online Voting System will be accepted by its intended users and the broader community. It considers the cultural, social, and aspects that could influence the system's successful adoption.

1. User Acceptance

- Most users today are familiar with using web and mobile applications.
- Voters, especially students, employees, or organization members, are likely to adopt the system due to its convenience and ease of use.
- Online voting eliminates the need to travel or stand in long queues, which enhances participation.

2. Trust and Transparency

- The system ensures transparency through secure vote recording and real-time result viewing.
- Features such as one-person-one-vote and data encryption help build public trust.
- Proper awareness campaigns and clear instructions can reduce resistance.

3. Social Impact

- Encourages greater participation in democratic processes, especially among tech-savvy and younger populations.
- Promotes inclusivity by allowing people in remote or underserved areas to vote.
- Reduces environmental impact by eliminating paper ballots and physical logistics.

4.4 Requirement Analysis

Requirement analysis identifies and documents the essential functions, features, and constraints of the Online Voting System. It ensures the system meets user needs and complies with technical and business expectations. The analysis is divided into functional and non-functional requirements.

4.4.1 Functional requirements

Functional requirements define what the system should do. These are the core actions the system must perform:

a. User Module

- User registration with unique Voter ID

- Secure login using username/password and optional OTP
- Cast one vote per user during the election period
- View confirmation of vote submission

b. Admin Module

- Admin login with elevated privileges
- Create/manage elections (start and end dates)
- Add, update, or delete candidates and voter records
- Monitor voting progress and view voter activity logs
- View and publish real-time election results

c. Voting Module

- Display list of eligible candidates
- Allow voters to select and cast their vote
- Store votes securely in an encrypted format
- Prevent double voting or unauthorized access

4.4.2 Non-functional requirements

Non-functional requirements define how the system should perform and support functional.

a. Security

- End-to-end data encryption (SSL/HTTPS)
- Secure password storage.
- Session handling to prevent unauthorized access
- Protection from SQL injection, XSS, and CSRF attacks

b. Performance

- Fast response times under normal and peak loads
- Scalable to support hundreds or thousands of users simultaneously

c. Usability

- Intuitive and clean user interface
- Mobile and desktop responsive design
- Multi-language support (optional)

d. Reliability

- Minimal downtime, especially during voting period
- Accurate and consistent vote recording

e. Maintainability

- Modular codebase for easy updates and debugging
- Clear documentation and error logging

4.5 Requirements Specification

1. Introduction

This document outlines the software requirements for the development of a secure, web-based Online Voting System. The system is designed to enable eligible users to vote remotely and electronically, while ensuring integrity, privacy, and transparency.

2. Functional Requirements

ID	Requirement Description
1	The system shall allow users to register using a unique Voter ID.
2	The system shall authenticate users via login credentials and optional OTP verification.
3	The system shall allow each authenticated user to cast one vote per election.
4	The system shall store votes in an encrypted format.
5	The system shall allow administrators to manage voter and candidate records.
6	The system shall allow administrators to schedule and manage elections.
7	The system shall automatically tally votes and generate results.
8	The system shall provide a dashboard for admins to view statistics and logs.
9	The system shall display confirmation after a vote is successfully cast.
10	The system shall restrict access to unauthorized users.

3. Non-Functional Requirements

1. The system shall use HTTPS for all data transmission.
2. The system shall store all passwords in hashed format using secure algorithms
3. The system shall be responsive and usable on mobile, tablet, and desktop devices.
4. The system shall handle at least 1000 concurrent users without performance degradation.
5. The system shall provide 99.5% uptime during active election periods.
6. The system shall log all admin and voting actions for auditing purposes.

7 The system shall be compatible with Chrome, Firefox, Safari, and Edge.

8 The system shall display error messages for failed login or voting attempts.

9 The system shall be easy to maintain and support future feature upgrades.

4. User Requirements

User	Requirements
Voter	Register, log in, view candidates, cast vote, receive confirmation.
Administrator	Log in securely, manage elections, add/update/delete users and view results.
System Admin	Perform system maintenance, security updates, data backups, and error monitoring.

4.5.1 Hardware Requirements

Client-Side (Voter Device)

- Device: Desktop, laptop, tablet, or smartphone
- Processor: Minimum dual-core CPU (2.0 GHz or higher)
- RAM: Minimum 2 GB (4 GB or more recommended for smooth performance)

- Display: Compatible with modern web browsers (Chrome, Firefox, Safari, Edge)
- Internet: Stable internet connection (Wi-Fi or mobile data)

Server-Side (Hosting & Backend)

- Processor: Quad-core 2.5+ GHz or better
- RAM: Minimum 8 GB (16 GB recommended for high traffic)
- Storage: SSD with at least 250 GB for data storage and logs
- Network: High-speed internet with static IP
- Server Uptime: Minimum 99.9% (Cloud/VPS recommended)

4.5.2 Software Requirements

Client-Side

- Web Browser: Latest versions of Chrome, Firefox, Edge, Safari
- Operating System: Windows, macOS, Android, iOS, or Linux
- Optional: Mobile app (if the voting system supports it)

Server-Side

- Operating System: Linux (Ubuntu/CentOS) or Windows Server
- Web Server: Apache, Nginx, or IIS
- Backend Language: PHP, Python, Node.js, or Java
- Database: MySQL, PostgreSQL, or MongoDB
- Frameworks: Laravel, Django, Express.js, or Spring Boot
- Security Tools: SSL certificate, firewalls, intrusion detection systems

6. Constraints

- Requires stable internet connection
- One user can vote only during the active election window
- Users must have valid credentials (e.g., Voter ID) to access the system

7. Assumptions

- All voters have internet-enabled devices
- Admins have technical knowledge to operate the backend portal
- Users are familiar with basic login and web navigation

4.5.3 language specification

The Language Specification section outlines the programming languages, scripting languages, and query languages used in developing the Online Voting System. The

selection is based on ease of development, compatibility, community support, and security features.

1. Programming and Scripting Languages

Language	Purpose
HTML	Used to structure the web pages (frontend layout)
CSS	Used to style the web pages for responsive and clean UI
JavaScript	Adds interactivity to the frontend (form validation, events)
PHP / Python	Used for backend development to handle logic, sessions, and server-side operations
SQL (MySQL/PostgreSQL)	Used for managing and querying the relational database that stores user, vote, and result data

2. Justification of Language Choice

- PHP/Python/Node.js are widely used, open-source, and well-supported for building secure web applications.
- HTML, CSS, JavaScript are the standard languages for modern web development.
- SQL ensures robust and efficient management of structured data like user profiles and voting records.
- These languages provide scalability, security, and fast development cycles suited for a voting system.

5.SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, and interfaces of a software system. In the case of the Online Voting System, the design ensures that the system is secure, modular, scalable, and easy to use.

5.1 System Architecture

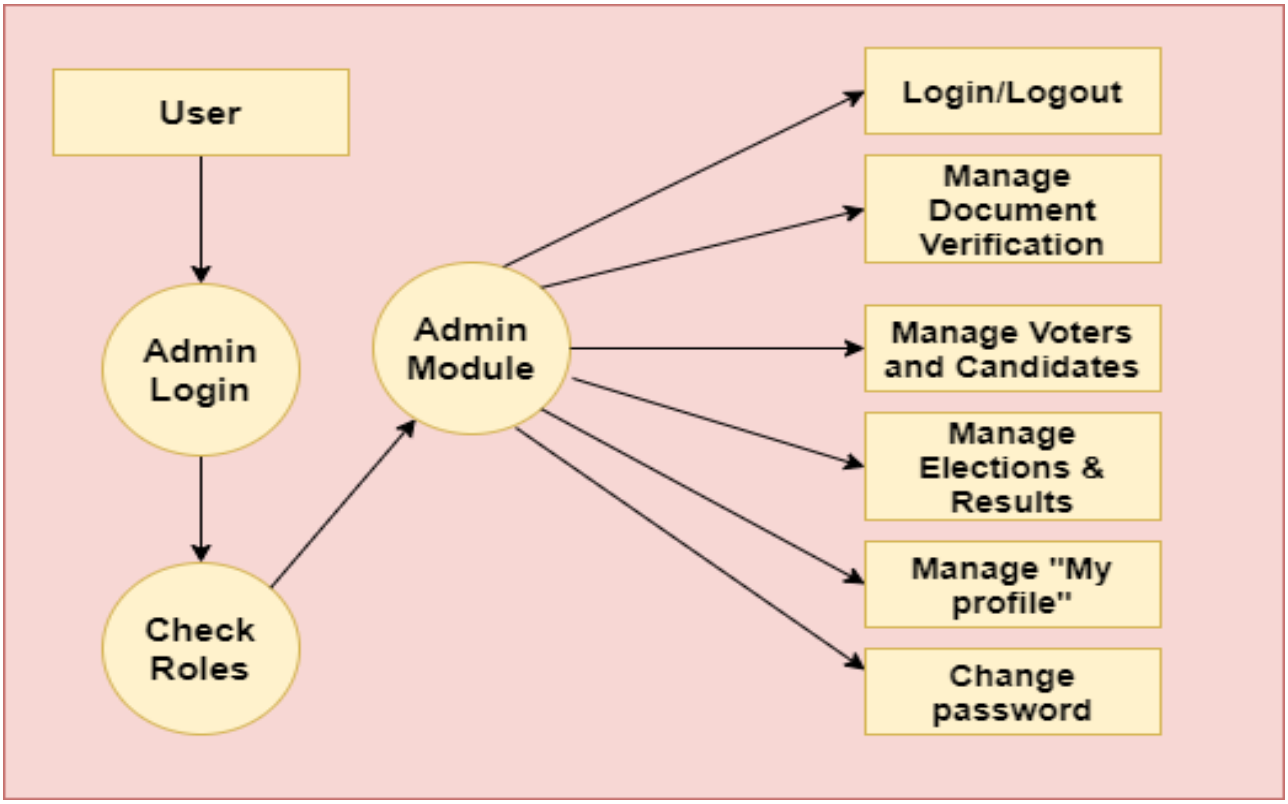


Fig no: 5.1

5.2 UML Diagrams

Unified Modeling Language (UML) diagrams are used to visually represent the structure, behavior, and interactions of the components in a software system. They serve as a blueprint during the design phase, making it easier to understand system functionality, identify key components, and ensure a clear communication flow between developers and stakeholders.

For the Online Voting system, three essential UML diagrams are used:

- Use Case Diagram
- Activity Diagram
- Sequence Diagram

Each diagram captures different aspects of the system:

- The Use Case Diagram shows the interaction between the user and the system.
- The Activity Diagram illustrates the flow of actions and decision points in the vulnerability scanning process.

These diagrams help ensure that the system meets its functional requirements and that all components are working in sync.

They also make future enhancements easier by providing a visual representation of the system’s flow.Below, each diagram will be explained in detail and structured based on the scanner's features, modules, and use-case scenarios.

5.2.1 Use Case Diagram

This image is a **Use Case Diagram** for an **online voting system**. It shows interactions between two types of actors—**User** and **Admin**—and the system functions (use cases) they can perform.

Actors:

1. **User** – A regular user of the voting system.
2. **Admin** – The system administrator with additional control capabilities.

Use Cases:

1. **Login** – Both User and Admin can log in.
2. **Registration** – Only the User can register.
3. **Vote** – Only the User can vote in an election.
4. **Result** – Both User and Admin can view the results.
5. **Adds candidate** – Only Admin can add a candidate.
6. **Remove candidates** – Only Admin can remove candidates.

Diagram Interpretation:

Arrows from an actor to a use case represent interactions or permissions.

- **User Capabilities**:
    - Login
    - Register
    - Vote
    - View Results

- **Admin Capabilities**:
    - Login
    - View Results
    - Add Candidates
    - Remove Candidates

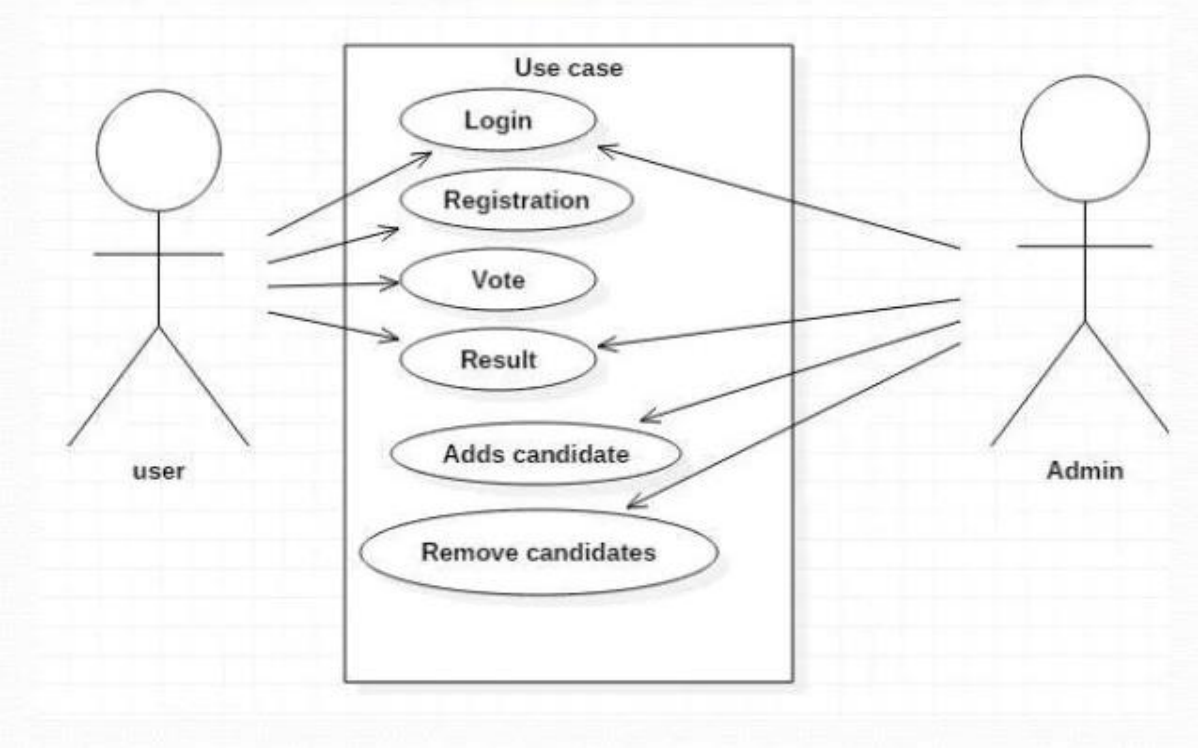


Fig no:5.2.1: Use case diagram

This diagram provides a clear overview of how each type of user interacts with the system and helps in defining system requirements for development.

5.2.2 Activity Diagram

Activity Diagram—a type of UML diagram that represents the dynamic flow of actions and decisions within a system. It visually maps out how a process proceeds step-by-step, including decision points and alternate paths.

This is an **Activity Diagram** for an **Online Voting System**, depicting the sequence of actions based on different user roles (Student and Admin).

Key Components and Flow:

1. **Start Node** (black circle at the top):
    - Represents the beginning of the process.
2. **Homepage → Login**:
    - The system begins at the homepage and proceeds to the login activity.
3. **Decision Node** (horizontal bar after login):
    - Splits the process based on the type of user: **Student** or **Admin**.

Student Path:

- **Choose the candidate and votes him**:
  Student selects and votes for a candidate.
- **Checks the vote**:
  After voting, the student can check the voting status.
- **Logout → Stop**:
  The student logs out, and the session ends.

Admin Path:

- **Adds and removes the candidates**:
  Admin manages candidate entries.
- **Checks the result**:
  Admin views the voting results.
- **Logout → Stop**:
  The admin logs out, ending the session

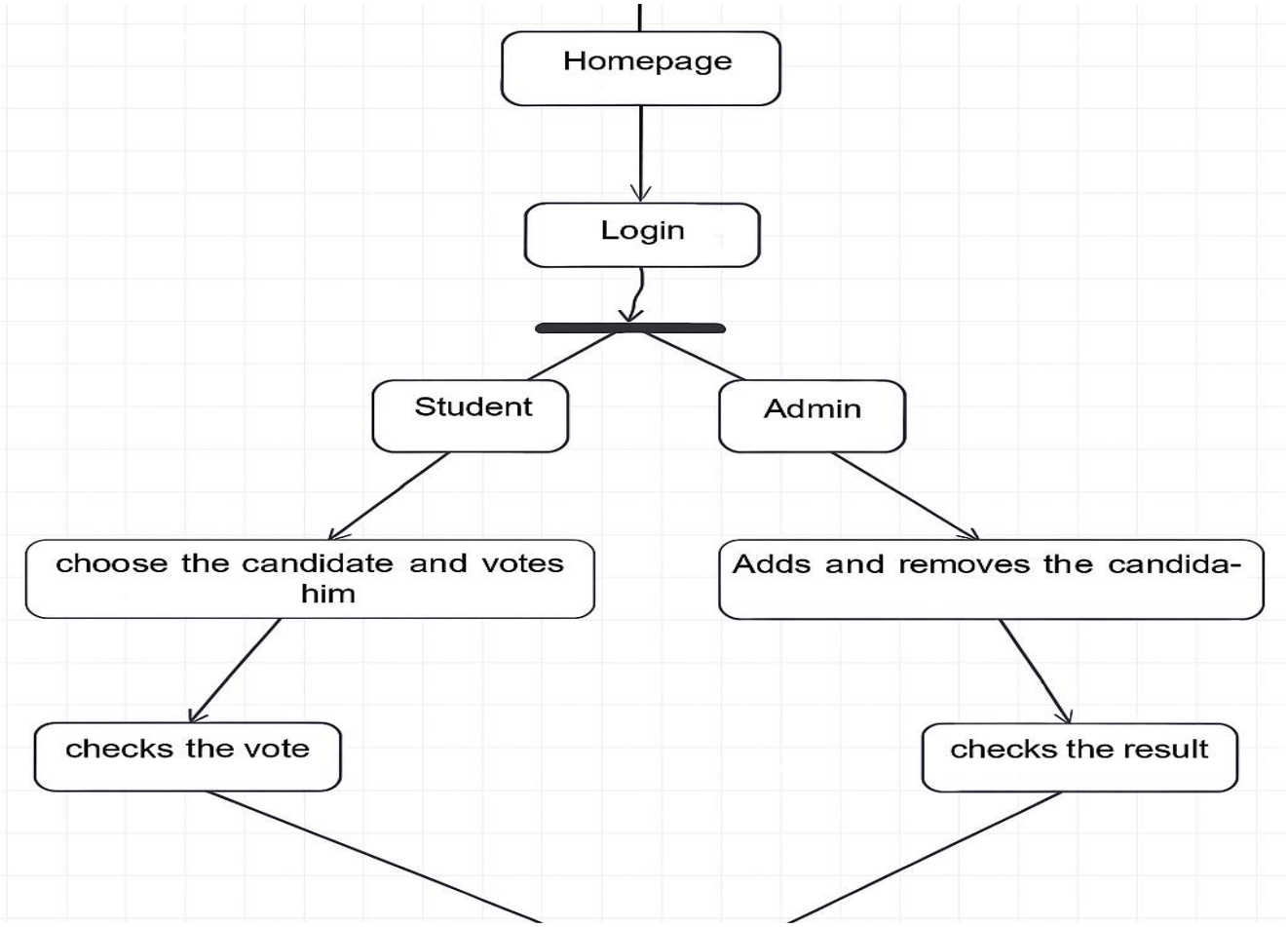


Fig no: 5.2.2 Activity Diagram

5.2.3 Sequence Diagram

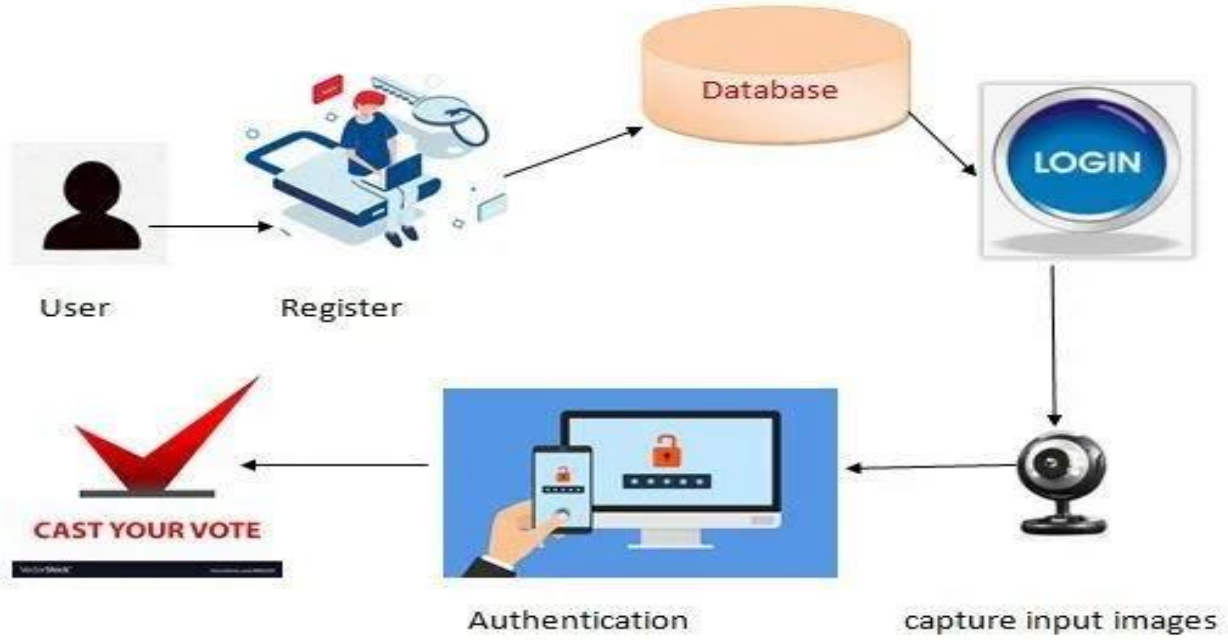


Fig no:5.2.3: Sequence Diagram

This image illustrates a process flow diagram for an online voting system with biometric or image-based authentication. Here's a step-by-step explanation of each component:

1. User

- The process starts with a user who wants to participate in the voting system.

2. Register

- The user performs registration, likely involving entering details and submitting identity proofs (possibly an image or document).
- Registration data is then stored in the database.

3. Database

- Stores user credentials and biometric/image data for future verification.
- Interacts with both the registration and login processes.

4. Login

- The user attempts to log in using credentials (e.g., username/password or biometrics).
- This action retrieves user data from the database for verification.

5. Capture Input Images

- During login or voting, the system captures input images (likely using a webcam) for face verification or ID validation.

6. Authentication

- Compares the live image input with stored data in the database.
- Ensures that the user is genuine and prevents impersonation or fraud.

7. Cast Your Vote

- Once authenticated successfully, the user is allowed to cast their vote.

6.IMPLEMENTATION AND RESULTS

6.1 Algorithm used

Registration Phase:

1. Voter Registration: Voters register on the platform with their details
2. Verification: Voters' details are verified through email or other means.
3. Unique ID Generation: A unique ID is generated for each voter.

Voting Phase:

1. Voter Login: Voters log in to the platform using their credentials.
2. Candidate Selection: Voters select their preferred candidate(s).
3. Vote Encryption: The vote is encrypted to ensure secrecy.
4. Vote Casting: The encrypted vote is cast and stored in a secure database.

Vote Counting Phase:

1. Vote Decryption: Votes are decrypted after the voting period ends.
2. Vote Counting: Votes are counted and results are generated.
3. Result Announcement: Results are announced publicly.

Security Measures:

1. Authentication: Voters are authenticated to ensure only eligible voters can vote.
2. Authorization: Voters can only vote once.
3. Encryption: Votes are encrypted to ensure secrecy.
4. Secure Storage: Votes are stored in a secure database.
5. Audit Trail: All activities are logged for auditing purposes.

6.2 Sample Code:

Main.py

```
from users import register  _user,  login _user
from voters import Voting System
def main():
voting _system = Voting System()
while True:
print("\n1. Register")
print("2. Login")
print("3. Vote")
print("4. View Results")
print("5. Exit")
choice = input("Enter your choice: ")
if choice == "1":
register _user()
elif choice == "2":
login _user()
elif choice == "3":
voting_system.vote()
elif choice == "4":
voting_system.view_results()
elif choice == "5":
print("Exiting the system.")
break
else:
print("Invalid choice. Please try again.")
if __name__ == "__main__":
main()
```

Users.py:

```
import getpass
class User:
users_db = {}
def __init__(self, username, password):
self.username = username
self.password = password
self.has_voted = False
@classmethod
def add_user(cls, username, password):
if username in cls.users_db:
print("Username already exists. Please try a different username.")
return False
cls.users_db[username] = User(username, password)
return True
@classmethod
def authenticate(cls, username, password):
user = cls.users_db.get(username)
if user and user.password == password:
return user
return None
# These must be outside the class
def register_user():
username = input("Enter a username: ")
password = getpass.getpass("Enter a password: ")
if User.add_user(username, password):
```

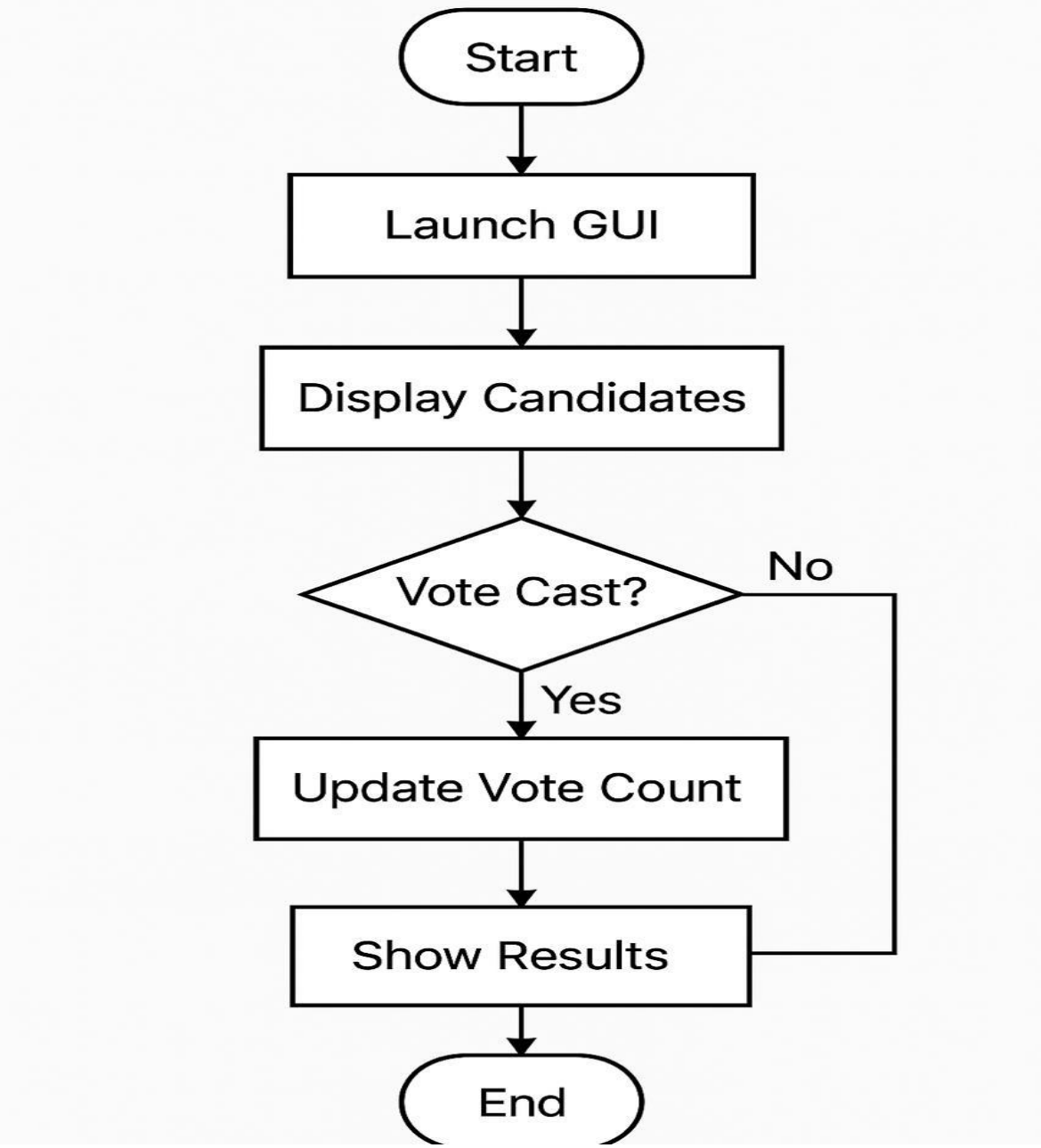
```
print("User registered successfully.")
else:
print("Registration failed.")
def login_user():
username = input("Enter a username: ")
password = getpass.getpass("Enter a password: ")
user = User.authenticate(username, password)
if user:
print("Login successful.")
else:
print("Invalid username or password.")
```

Voters.py:

```
from users import User
class VotingSystem:
parties = {"A": 0, "B": 0, "C": 0}
def vote(self):
username = input("Enter your username to vote: ")
password = input("Enter your password to vote: ")
user = User.authenticate(username, password)
if user:
if user.has_voted:
print("You have already voted.")
return
print("Parties: A, B, C")
choice = input("Enter the party name you want to vote for: ").upper()
if choice in self.parties:
```

```
self.parties[choice] += 1
user.has_voted = True
print("Vote cast successfully.")
else:
print("Invalid party choice.")
else:
print("Invalid username or password.")
def view _results(self):
print("\n Voting Results:")
for party, votes in self. parties.items():
print(f"Party {party}: {votes} votes")
```

7.SYSTEM TESTING



8.Screenshots

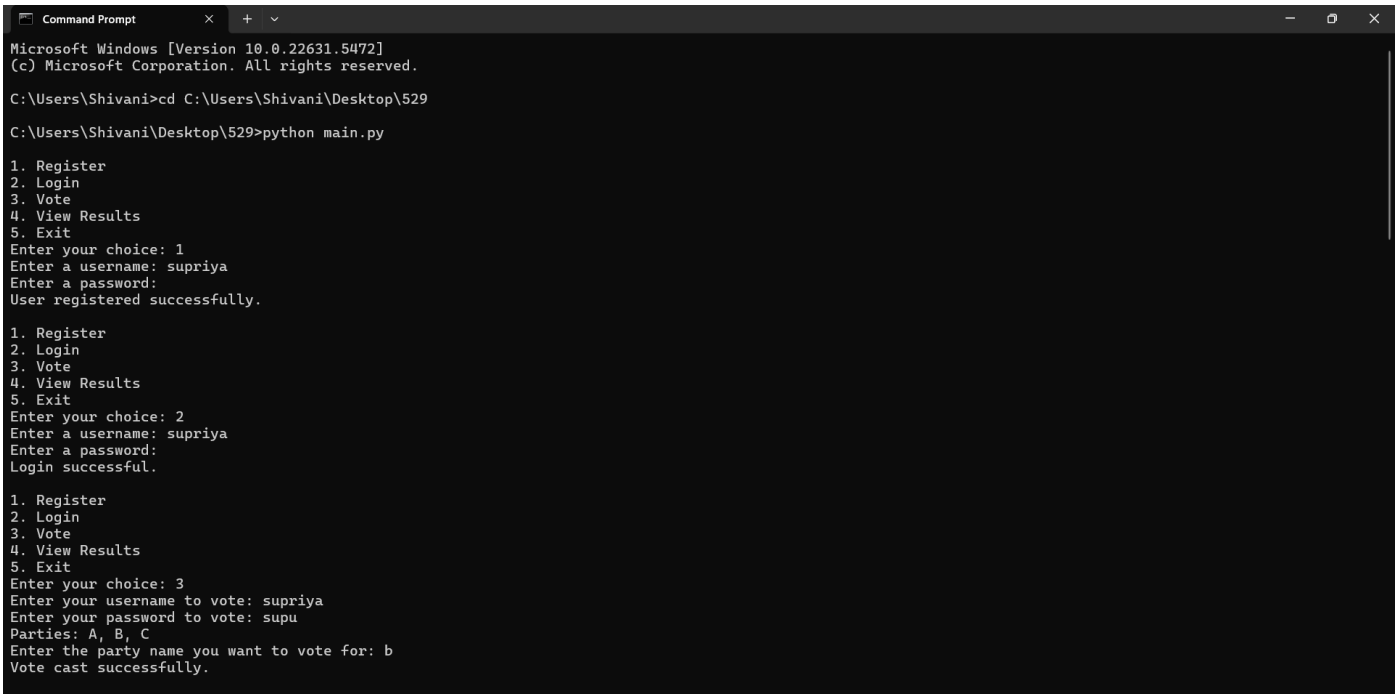


Fig no : 8.1

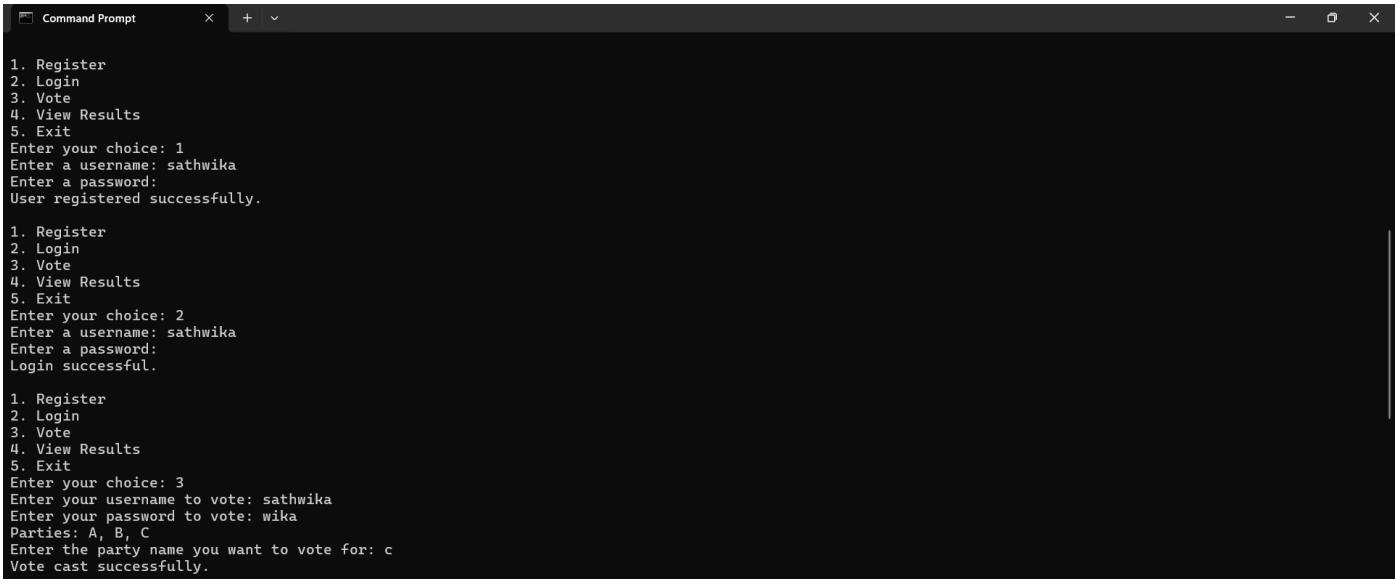


Fig no : 8.2

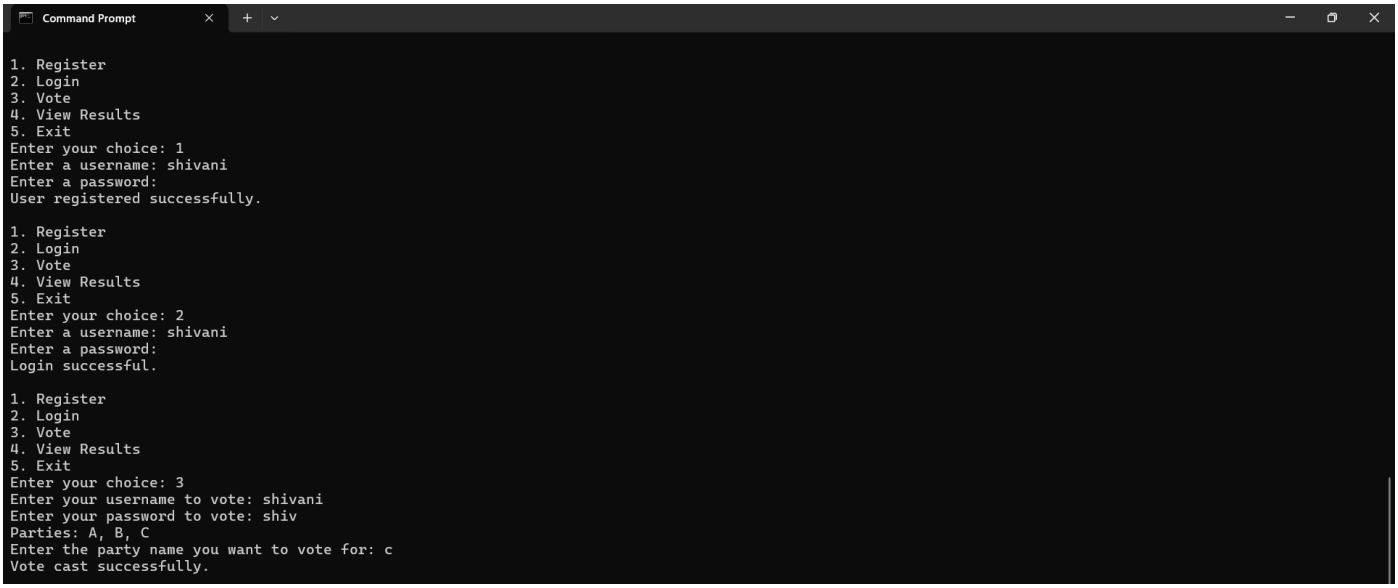


Fig no: 8.3

Fig no : 8.4

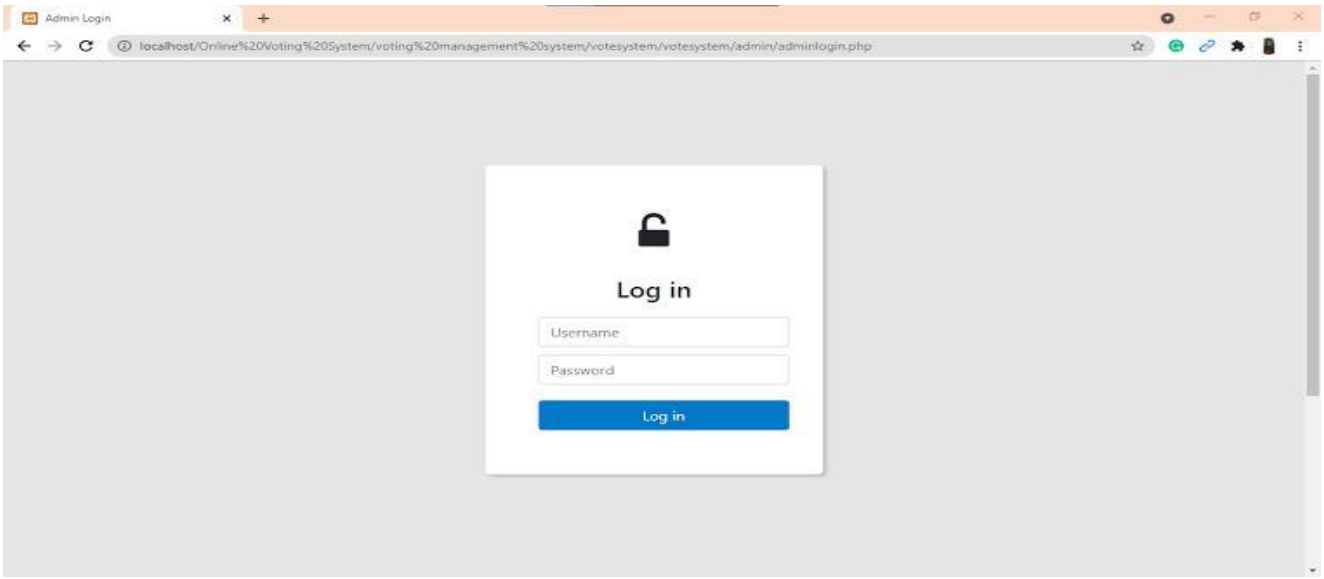


Fig No: 8.5 output 1

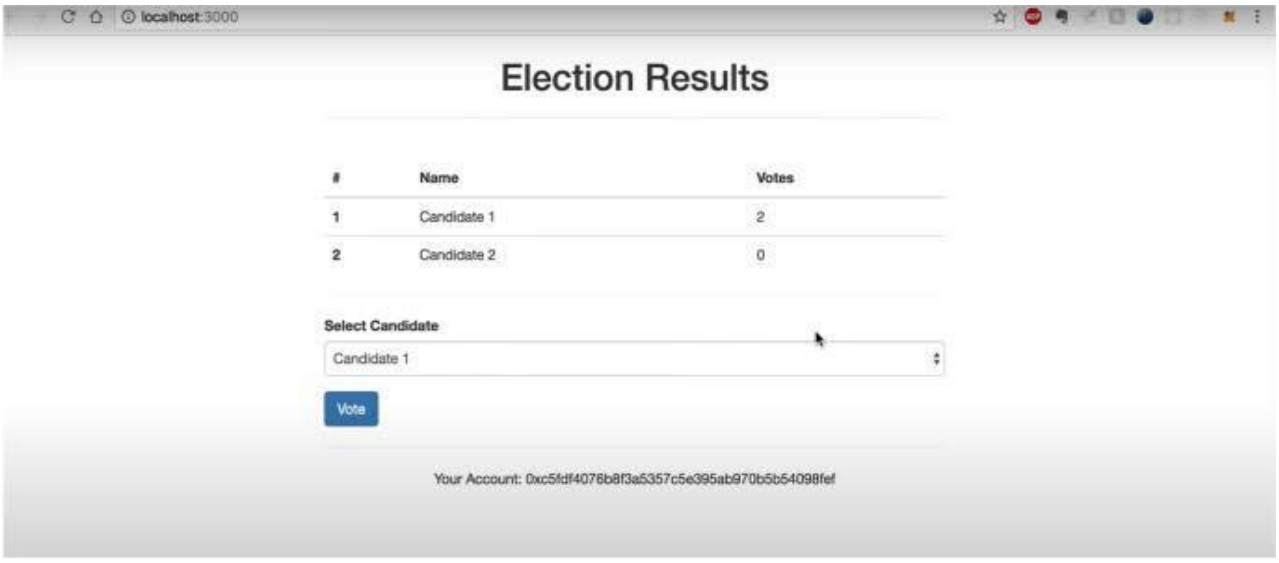


Fig No: 8.6 output 2

9.CONCLUSION

In conclusion, Online Voting System is a highly innovative and technological solution to many of the challenges faced in traditional voting systems. It not only simplifies voting process but also saves time and resources. With secure authentication and verification measures in place, the system offers transparency and accountability in the electoral process.

However, the implementation of the online voting system still raises concerns regarding security, privacy, and accessibility. It is important to address these issues before making the system available to the public. The online voting system has tremendous potential to revolutionize the electoral process and ensure greater participation and representation for all. The way forward is to focus on developing a comprehensive security infrastructure and addressing concerns to make the system accessible to all.

By doing so, we can create a more inclusive and democratic electoral process for the future. Overall, this system streamlines the election process, encourages higher voter turnout, and builds trust through accurate and fair results. It is a practical and scalable model suitable for schools, universities, and organizations.

In conclusion, the Online Voting System is an innovative application that has the potential to revolutionize how elections are conducted. With further enhancements such as biometric authentication, email/SMS verification, and blockchain integration, it can be made even more secure and scalable for use in large-scale government or institutional elections

10.FUTURE SCOPE

The future scope of online voting systems is highly promising, with potential to revolutionize the way democratic processes are conducted. As technology continues to advance, online voting is expected to become more secure, accessible, and efficient. Future developments may include integration with blockchain for enhanced transparency and security, biometric authentication for reliable voter verification, and mobile-based platforms to increase participation, especially in remote areas. Governments may increasingly adopt online voting for national and local elections, enabling greater inclusivity for citizens such as expatriates and military personnel. Additionally, online voting is likely to find broader use in private organizations, educational institutions, and corporate governance. While challenges like cybersecurity and digital accessibility remain, the overall trajectory points toward widespread adoption supported by strong legal, technical, and ethical frameworks.

11.BIBLIOGRAPHY

11.1 References

1. Downey, Allen B.
   Think Python: How to Think Like a Computer Scientist, 2nd Edition, O'Reilly Media, 2015.
2. Zelle, John M.
   Python Programming: An Introduction to Computer Science, 3rd Edition, Franklin, Beedle & Associates Inc., 2016.

11.2 Websites

1. Python Official Documentation
   https://docs.python.org/3/
2. W3Schools Python Tutorial
   https://www.w3schools.com/python/
3. Geeks for Geeks – Python Programming
   https://www.geeksforgeeks.org/python-programming-language/
4. Real Python – Tutorials and Examples
   https://realpython.com/
5. Stack Overflow – Community Support and Debugging Help
   https://stackoverflow.com/