ONLINE VOTING SYSTEM

A PROJECT REPORT

Submitted by

IRIN MINERWA S (2303811724322038)

in partial fulfillment of requirements for the award of the course

CGB1201 – JAVA PROGRAMMING

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

DECEMBER, 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"ONLINE VOTING SYSTEM"** is the bonafide work of **IRIN MINERWA S(2303811724322038)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

Signature

Dr. T. AVUDAIAPPAN M.E.,Ph.D.,

HEAD OF THE DEPARTMENT,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

Signature

Mrs. JOANY FRANKLIN M.E.,

SUPERVISOR,

Department of Artificial Intelligence,

K. Ramakrishnan College of Technology,

Samayapuram, Trichy -621 112.

S

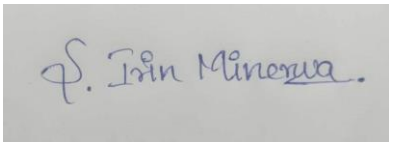
Submitted for the viva-voce examination held on 07.12.24

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on "**ONLINE VOTING SYSTEM**" is the result of original work done by us and best of our knowledge, similar work has not been submitted to "**ANNA UNIVERSITY CHENNAI**" for the requirement of Degree of **BACHELOR OF TECHNOLOGY**. This project report is submitted on the partial fulfillment of the requirement of the award of the **CGB1201 – JAVA PROGRAMMING.**



Signature

IRIN MINERWA S

Your Name

Place: Samayapuram

Date: 07/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and indebtedness to our institution, **"K. Ramakrishnan College of Technology (Autonomous)",** for providing us with the opportunity to do this project.

I extend our sincere acknowledgment and appreciation to the esteemed and honorable Chairman, **Dr. K. RAMAKRISHNAN, B.E.,** for having provided the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director, **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding our project and offering an adequate duration to complete it.

I would like to thank **Dr. N. VASUDEVAN, M.TECH., Ph.D.,** Principal, who gave the opportunity to frame the project to full satisfaction.

I thank **Dr.T.AVUDAIAPPAN, M.E.,Ph.D**., Head of the Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for providing her encouragement in pursuing this project.

I wish to convey our profound and heartfelt gratitude to our esteemed project guide **Mrs. JOANY FRANKLIN M.E**., Department of **ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**, for her incalculable suggestions, creativity, assistance and patience, which motivated us to carry out this project.

I render our sincere thanks to the Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards.

MISSION OF THE INSTITUTION

- Be a centre of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all- round personalities respecting moral and ethical values.

VISION AND MISSION OF THE DEPARTMENT

To excel in education, innovation and research in Artificial Intelligence and Data Science to fulfill industrial demands and societal expectations.

Mission 1: To educate future engineers with solid fundamentals, continually improving teaching methods using modern tools.

Mission 2: To collaborate with industry and offer top-notch facilities in a conductive learning environment.

Mission 3: To foster skilled engineers and ethical innovation in AI and Data Science for global recognition and impactful research.

Mission 4: To tackle the societal challenge of producing capable professionals by instilling employability skills and human values.

PROGRAM EDUCATIONAL OBJECTIVES (PEOS)

PEO 1: Compete on a global scale for a professional career in Artificial Intelligence and Data Science.

PEO 2: Provide industry-specific solutions for the society with effective communication and ethics.

PEO 3: Hone their professional skills through research and lifelong learning initiatives.

PROGRAM OUTCOMES

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs)

- **PSO 1:** Capable of working on data-related methodologies and providing industry-focussed solutions.
- **PSO2:** Capable of analysing and providing a solution to a given real-world problem by designing an effective program.

ABSTRACT

The **Online Voting System** project provides a secure and efficient platform for digital elections using Java Swing. The system allows eligible voters to log in using a unique Voter ID and password, ensuring secure authentication. After logging in, voters can select their preferred candidate from a list of options. The system records and updates vote counts in real-time, ensuring transparency. Voters can view the election results after casting their vote, which displays the total votes for each candidate.

This system is designed to be simple and user-friendly, making it suitable for small-scale elections such as those in educational institutions or organizations. The project highlights the potential of digital voting systems to improve the efficiency and transparency of electoral processes. It serves as a basic prototype, which can be enhanced with features like data encryption, secure storage, and more robust voter verification for larger-scale applications. The system aims to demonstrate the feasibility of online voting, making elections more accessible and transparent for all voters.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Online Voting System is a digital platform designed to streamline the voting process, ensuring security, transparency, and efficiency. Built using Java Swing, the system allows voters to log in with a Voter ID and password, ensuring only eligible participants can vote. After authentication, voters can select their preferred candidate, and the vote is recorded in real-time. The system also displays real-time election results, providing transparency. This project aims to reduce the complexities of traditional voting methods, offering a user-friendly and accessible alternative. It serves as a foundation for future enhancements, such as better security and scalability for larger elections.

1.2 OBJECTIVE

The objective of the Online Voting System is to create a secure, easy-to-use platform for digital voting. The main goals are:

- ✓ **Secure Login:** Ensure only authorized voters can vote using a Voter ID and password.
- ✓ **Easy Voting:** Allow voters to easily select and vote for their chosen candidate.
- ✓ **Real-Time Vote Counting:** Track and update the vote count instantly.
- ✓ **Show Results:** Display the election results clearly after voting.
- ✓ **User-Friendly:** Make the system simple and accessible for all voters.
- ✓ **Future Improvements:** Provide a foundation for adding more security and features for larger elections.

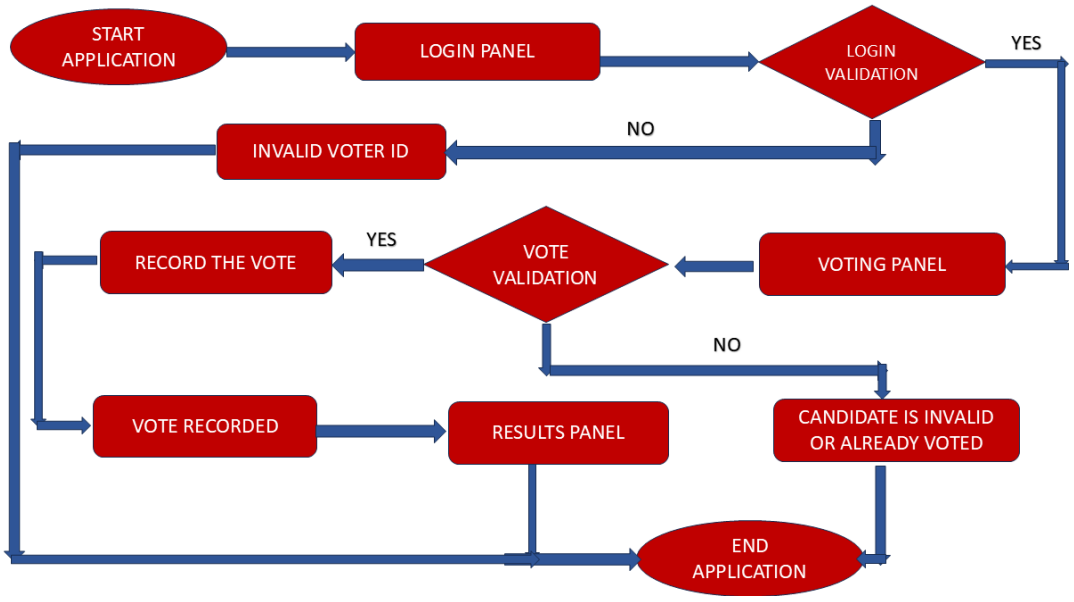
CHAPTER 2
PROJECT METHODOLOGY

2.1 PROPOSED WORK

The proposed work focuses on developing a secure and transparent online voting system for organizational or educational elections. The system will allow eligible voters to cast their votes electronically while maintaining the integrity of the voting process. Key features include:

1. **Login System:** Voters authenticate using a secure Voter ID and password, with passwords hashed using SHA-256.
2. **Vote Casting:** Voters can select a candidate, and the system ensures they can vote only once.
3. **Election Results:** After voting, results are displayed with the total votes for each candidate.
4. **Security:** Ensures the confidentiality of votes and prevents multiple votes from the same user.
5. **User Interface:** A user-friendly interface is built using Java Swing, with separate panels for login, voting, and results.

2.2 BLOCK DIAGRAM



CHAPTER 3

JAVA PROGRAMMING CONCEPTS

3.1 Object-Oriented Programming (OOP)

The Online Voting System follows Object-Oriented Programming (OOP) principles. This includes the use of classes and objects to represent real-world entities like voters, candidates, and the voting process. For instance:

- The Voter can be represented as a class with attributes such as voterId and password.
- The VotingSystem class can handle the overall logic of the voting system, including managing voters, storing votes, and displaying results.
- OOP allows for modular and scalable code, where each module (login, voting, results) can be treated as separate objects or methods within the system.

This concept is essential to create organized, maintainable, and reusable code in the Online Voting System.

3.2 Data Structures (HashMap)

The Online Voting System utilizes HashMap, a core data structure in Java, to store and manage key data efficiently.

- HashMap is used to store voter credentials (e.g., voters.put("voter1", "hashedPassword")) and map each candidate to their vote count (e.g., candidatesVotes.put("Candidate A", 0)).
- HashMap allows for fast lookups, insertions, and updates, which is crucial for managing the voting process and ensuring that votes are counted accurately and quickly.

CHAPTER 4
MODULE DESCRIPTION

4.1 Login Module

Authenticates voters by verifying their voter ID and password against stored credentials. Only valid users can proceed to vote, ensuring secure access. If the voter has already cast their vote, they are prevented from voting again. The login module utilizes password hashing to enhance security and protect user credentials.

4.2 Voting Module

Allows authenticated voters to cast their vote for a candidate. Ensures voters can only vote once and prevents invalid votes. The system updates vote counts for candidates and prevents multiple submissions. Voters are informed of successful votes or errors in candidate selection.

4.3 Results Module

Displays the aggregated results of the election after votes are cast. It shows the vote count for each candidate in a clear and user-friendly format. The module ensures transparency and provides voters with a view of the election outcome after voting.

4.4 Security Module

Secures voter credentials through password hashing to prevent unauthorized access. It tracks whether a voter has already voted to ensure one vote per person. Protects the election process from fraud and unauthorized manipulation by maintaining data integrity.

4.5 User Interface (UI) Module

Provides a user-friendly graphical interface for voters. The module uses panels for login, voting, and results with clear navigation. It ensures smooth interaction with the system, guiding users step by step through the election process with intuitive design.

CHAPTER 5

CONCLUSION

The Online Voting System project demonstrates a secure, transparent, and efficient way of conducting elections using Java programming. By leveraging **GUI components** (Swing) and secure password handling (SHA-256 hashing), the system ensures that only authenticated and eligible voters can cast their votes. The voting process is protected from multiple submissions, and the results are displayed accurately after the voting period ends.

Key features like **user authentication**,**vote tracking**, and **results presentation** are effectively implemented, providing an intuitive and secure platform for conducting elections. The modular design helps in maintaining and enhancing the system easily, ensuring it can cater to various types of elections, such as governmental, organizational, or educational. The project is a practical demonstration of how technology can improve the integrity and accessibility of the election process.

REFERENCES

1) Oracle. (n.d.). *Java Platform, Standard Edition Documentation*. Retrieved from https://docs.oracle.com/javase/8/docs/

2) *Swing GUI Widgets in Java* - TutorialsPoint. (n.d.). Retrieved from https://www.tutorialspoint.com/swing/index.htm

3) *Java Collections Framework Overview*. Oracle. (n.d.). Retrieved from https://docs.oracle.com/javase/8/docs/api/java/util/package-summary.html

APPENDICES

APPENDIX A – SOURCE CODE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.security.MessageDigest;
import java.util.HashMap;
import java.util.Map;

public class OnlineVotingSystem {

    // Store candidate votes in a map
    private static Map<String, Integer> candidatesVotes = new HashMap<>();
    private static final String[] candidates = {"Candidate C", "Candidate B",
"Candidate A"};

    // Store valid voter credentials (hashed passwords)
    private static final Map<String, String> voters = new HashMap<>();
    private static final Map<String, Boolean> hasVoted = new HashMap<>(); //
Track if a voter has already voted

    static {
        // Initializing sample voters with Voter ID and hashed password
        voters.put("voter1", hashPassword("password1"));
        voters.put("voter2", hashPassword("password2"));
        voters.put("voter3", hashPassword("password3"));
```

```
    // Initialize candidates' votes to 0
    for (String candidate : candidates) {
        candidatesVotes.put(candidate, 0);
    }
}

public static void main(String[] args) {
    // Setting up the main frame
    JFrame frame = new JFrame("Online Voting System");
    frame.setSize(400, 300);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // Create a card layout to switch between different panels
    CardLayout cardLayout = new CardLayout();
    JPanel mainPanel = new JPanel(cardLayout);

    // Create login panel
    JPanel loginPanel = createLoginPanel(cardLayout, mainPanel);
    mainPanel.add(loginPanel, "Login");

    // Create voting panel
    JPanel votingPanel = createVotingPanel(cardLayout, mainPanel);
    mainPanel.add(votingPanel, "Vote");

    // Create results panel
    JPanel resultsPanel = createResultsPanel();
    mainPanel.add(resultsPanel, "Results");

    // Set the initial view to the login panel
```

```
        frame.add(mainPanel);
        frame.setVisible(true);
    }

    // Creates the login panel
    private static JPanel createLoginPanel(CardLayout cardLayout, JPanel
mainPanel) {
        JPanel loginPanel = new JPanel();
        loginPanel.setLayout(new GridLayout(3, 2));

        JLabel voterIdLabel = new JLabel("Enter Voter ID: ");
        JTextField voterIdField = new JTextField();
        JLabel passwordLabel = new JLabel("Enter Password: ");
        JPasswordField passwordField = new JPasswordField();

        JButton loginButton = new JButton("Login");

        // Action listener for login button
        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String voterId = voterIdField.getText();
                String password = new String(passwordField.getPassword());

                if (voters.containsKey(voterId) &&
voters.get(voterId).equals(hashPassword(password))) {
                    if (hasVoted.getOrDefault(voterId, false)) {
                        JOptionPane.showMessageDialog(loginPanel, "You have already
voted.");
```

```
                } else {
                    JOptionPane.showMessageDialog(loginPanel, "Login
successful!");
                    cardLayout.show(mainPanel, "Vote");
                }
            } else {
                JOptionPane.showMessageDialog(loginPanel, "Invalid Voter ID or
Password.");
            }
        }
    });

    loginPanel.add(voterIdLabel);
    loginPanel.add(voterIdField);
    loginPanel.add(passwordLabel);
    loginPanel.add(passwordField);
    loginPanel.add(loginButton);

    return loginPanel;
}

// Creates the voting panel
private static JPanel createVotingPanel(CardLayout cardLayout, JPanel
mainPanel) {
    JPanel votingPanel = new JPanel();
    votingPanel.setLayout(new BoxLayout(votingPanel, BoxLayout.Y_AXIS));

    JLabel candidatesLabel = new JLabel("Candidates: ");
    for (String candidate : candidates) {
```

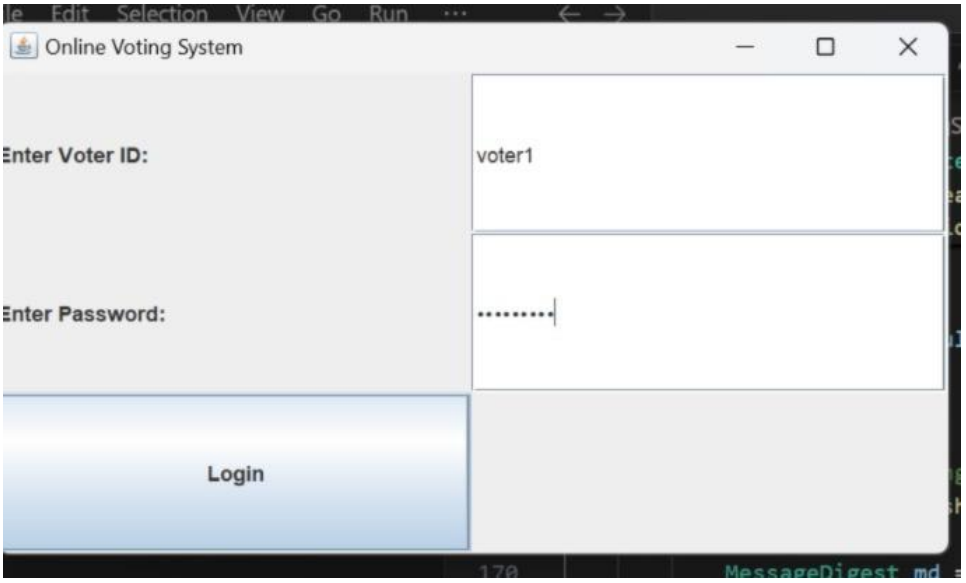
```
            candidatesLabel.setText(candidatesLabel.getText() + candidate + ", ");
        }

        JLabel voteLabel = new JLabel("Enter your vote (Candidate Name): ");
        JTextField voteField = new JTextField();

        JButton voteButton = new JButton("Cast Vote");

        // Action listener for vote button
        voteButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String vote = voteField.getText().trim();
                String voterId = voteField.getText().trim();  // assuming voterId is
unique
                if (candidatesVotes.containsKey(vote) &&
!hasVoted.getOrDefault(voterId, false)) {
                    candidatesVotes.put(vote, candidatesVotes.get(vote) + 1);
                    hasVoted.put(voterId, true);  // Mark as voted
                    JOptionPane.showMessageDialog(votingPanel, "Your vote has been
cast successfully!");

                    // Ask if the voter wants to see the results
                    int response = JOptionPane.showConfirmDialog(votingPanel,
"Would you like to see the results?",
                            "Election Results", JOptionPane.YES_NO_OPTION);
                    if (response == JOptionPane.YES_OPTION) {
                        cardLayout.show(mainPanel, "Results");
                    }
```

```
            } else {
                JOptionPane.showMessageDialog(votingPanel, "Invalid candidate
name or you have already voted.");
            }
        }
    });

    votingPanel.add(candidatesLabel);
    votingPanel.add(voteLabel);
    votingPanel.add(voteField);
    votingPanel.add(voteButton);

    return votingPanel;
}

// Creates the results panel
private static JPanel createResultsPanel() {
    JPanel resultsPanel = new JPanel();
    resultsPanel.setLayout(new BoxLayout(resultsPanel, BoxLayout.Y_AXIS));

    // Displaying election results
    JButton resultsButton = new JButton("Show Results");
    resultsButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            StringBuilder results = new StringBuilder("<html><h3>Election
Results:</h3>");
            for (String candidate : candidates) {
                results.append(candidate).append(":
```

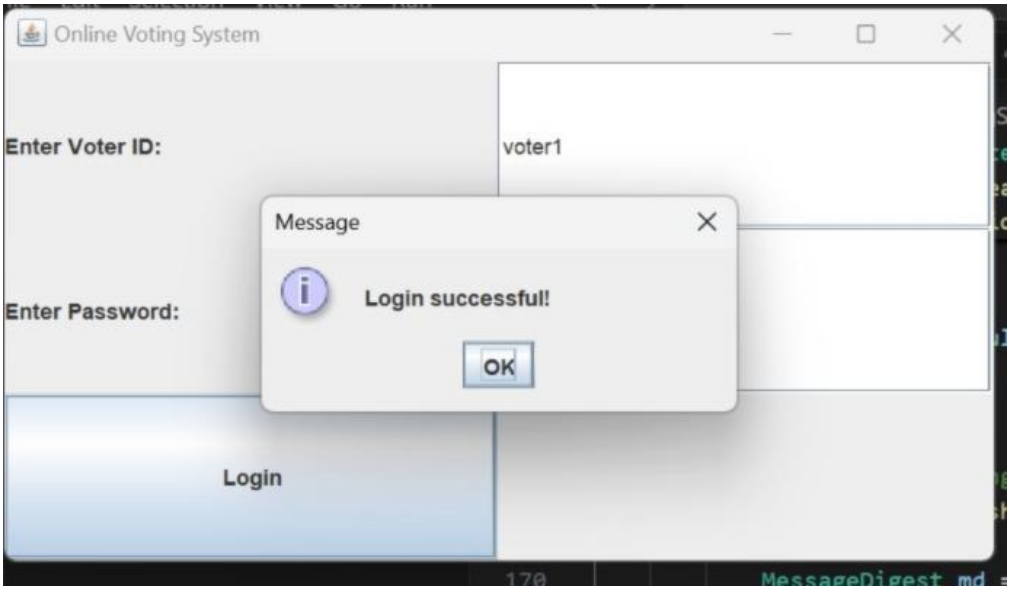
```
").append(candidatesVotes.get(candidate)).append(" votes<br>");
                }
                results.append("</html>");
                JOptionPane.showMessageDialog(resultsPanel, results.toString());
            }
        });

        resultsPanel.add(resultsButton);
        return resultsPanel;
    }

    // Hash the password using SHA-256
    private static String hashPassword(String password) {
        try {
            MessageDigest md = MessageDigest.getInstance("SHA-256");
            byte[] hashBytes = md.digest(password.getBytes());
            StringBuilder sb = new StringBuilder();
            for (byte b : hashBytes) {
                sb.append(String.format("%02x", b));
            }
            return sb.toString();
        } catch (Exception e) {
            throw new RuntimeException("Error hashing password", e);
        }
    }
}
```

APPENDIX B - SCREENSHOTS

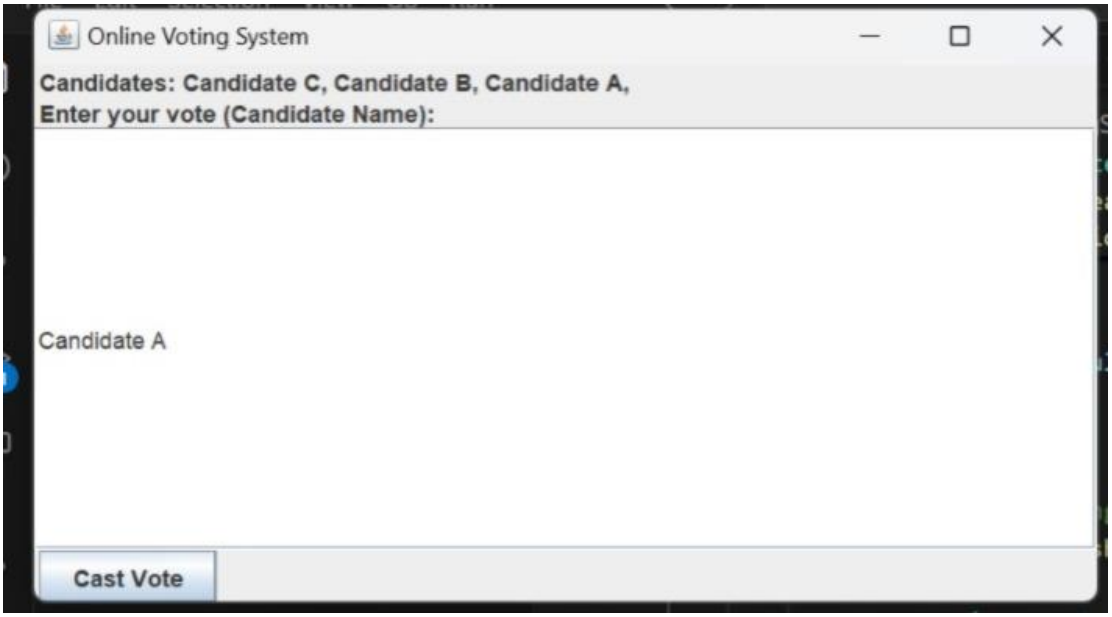
1)Welcome to the Online Voting System



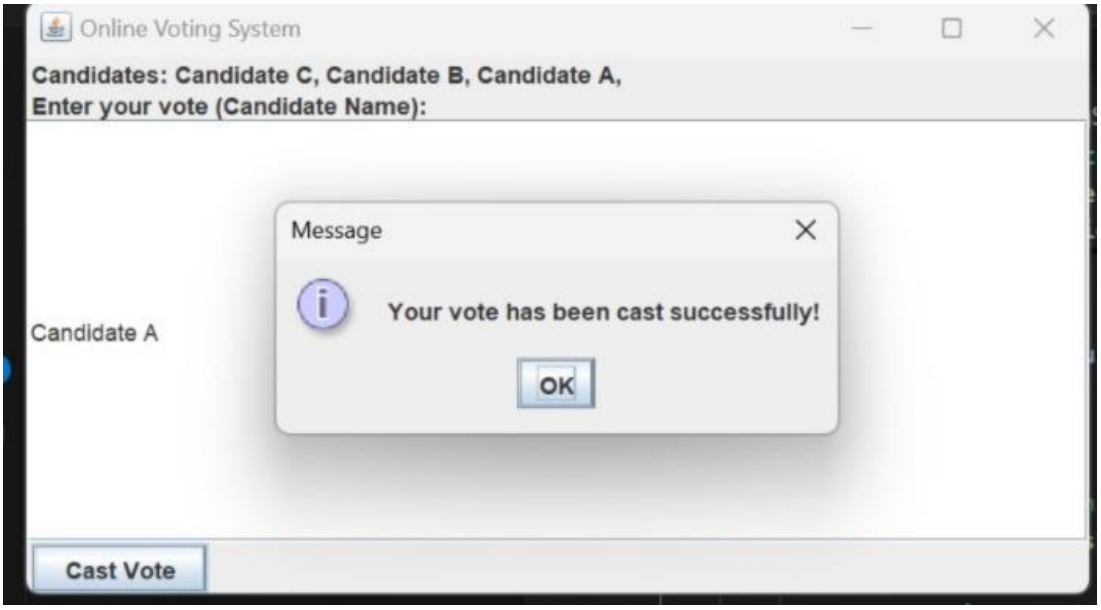
2) Login to Your Account



3)Vote a Candidate



4)Vote Has Been Cast Successfully



5)Would You Like To View Your Results

6)Click Show Results

7)View Your Votes