ONLINE VOTING SYSTEM

## A PROJECT REPORT

***Submitted by***

## OM PRAKASH M (2303811724321079)

***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**

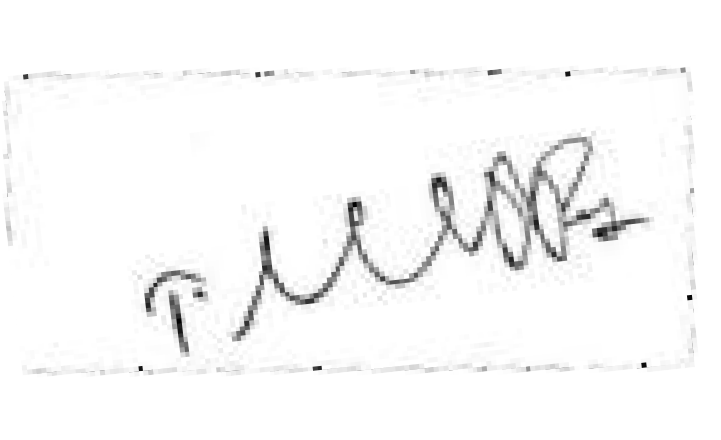
i

# 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 **OM PRAKASH M (ADB23079)** who carried out the project work during the academic year 2024 - 2025 under my supervision.



Signature Signature

**Mrs. S. GEETHA M.E., Dr. T. AVUDAIAPPAN M.E.,Ph.D.,**

**SUPERVISOR, HEAD OF THE DEPARTMENT,**

Department of Artificial Intelligence, Department of Artificial Intelligence,

K. Ramakrishnan College of Technology, K. Ramakrishnan College of Technology, Samayapuram, Trichy -621 112. Samayapuram, Trichy -621 112.

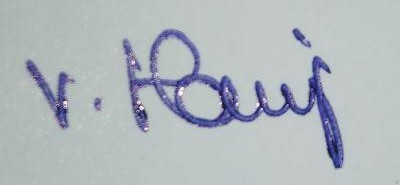
Submitted for the viva-voce examination held on 3.12.24



**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**

## OM PRAKASH M

**Place:** Samayapuram

**Date:** 3/12/2024

## 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.S.GEETHA, M.E.,** Department of **ARTIFICIAL INTELLIGENCE,** 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.

iv

## 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.

v

**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.

vi

1. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
2. **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.
3. **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.
4. **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 is a secure, web-based platform designed to facilitate seamless and transparent elections. The system allows eligible voters to register, authenticate, and cast their votes online, eliminating the need for traditional paper-based voting. Key features include user registration with OTP and email verification, secure login with multi-factor authentication, and vote encryption to ensure data confidentiality and integrity. The system also includes modules for candidate and voter management, real-time result calculation, and notifications to keep users informed. Built using Java and JDBC, the platform leverages modern technologies to handle concurrent operations efficiently, ensuring a smooth and scalable voting process. This project aims to enhance voter accessibility, improve security, and promote trust in the electoral process through digital innovation. Additionally, the system ensures that each voter can cast their vote only once, preventing duplication and fraud. The admin dashboard offers comprehensive control over managing elections, including monitoring user activity and generating reports. Overall, the project promotes transparency, efficiency, and accessibility, contributing to a more reliable and modern electoral process.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **No.** | **TITLE** | **PAGE**  **No.** |
|  | **ABSTRACT** | VIII |
| **1** | **INTRODUCTION** | **1** |
|  | 1.1 INTRODUCTION | **1** |
|  | 1.2 OBJECTIVE | **1** |
| **2** | **PROJECT METHODOLOGY** | **2** |
|  | 2.1 PROPOSED WORK | **2** |
|  | 2.2 BLOCK DIAGRAM | **3** |
| **3** | **JAVA PROGRAMMING CONCEPTS** | **4** |
|  | 3.1 KEY CONCEPT | **4** |
| **4** | **MODULE DESCRIPTION** | **5** |
|  | 4.1 USER REGISTRATION MODULE | **5** |
|  | 4.2 AUTHENTICATION AND LOGIN  MODULE | **5** |
|  | 4.3 VOTING MODULE | **5** |
|  | 4.4 CANDIDATE MANAGEMENT MODULE | **5** |
|  | 4.5 VOTER MANAGEMENT MODULE | **5** |
| **5** | **CONCLUSION** | **6** |
|  | **REFERENCES** | **7** |
|  | **APPENDICES** | **8** |
|  | Appendix A – Source code | **8** |
|  | Appendix B – Screen shots | **17** |

ix

# CHAPTER 1 INTRODUCTION

## INTRODUCTION

The Online Voting System is a digital platform designed to simplify and secure the voting process. It allows voters to register, authenticate, and cast their votes online, ensuring convenience and accessibility. The system incorporates advanced security measures like encryption and multi-factor authentication to protect voter data and prevent fraud. By streamlining election management, it enhances transparency, efficiency, and trust in the electoral process.

## OBJECTIVE

The objective of the Online Voting System is to develop a secure, user-friendly platform that facilitates efficient and transparent elections. It aims to streamline voter registration, authentication, and voting processes while ensuring data security and integrity. The project seeks to enhance voter accessibility, prevent fraud, automate vote counting, and deliver accurate, real-time results, fostering trust and confidence in the electoral process.

* + - Secure Registration
    - Data Integrity
    - Convenient Voting
    - Automated Counting
    - Fraud Prevention

1

# CHAPTER 2 PROJECT METHODOLOGY

## PROPOSED WORK

The proposed work involves designing and developing a secure and efficient Online Voting System that facilitates the entire election process digitally. The key steps include:

## User Registration and Verification:

Implement a module to allow eligible voters to register online with proper verification, including OTP and email confirmation for authentication.

## Authentication and Login:

Develop a secure login system with multi-factor authentication to ensure that only registered users can access the voting platform.

## Voting Process Implementation:

Design a user-friendly interface for casting votes. Each voter will be able to select their preferred candidate and submit their vote securely.

## Vote Encryption and Storage:

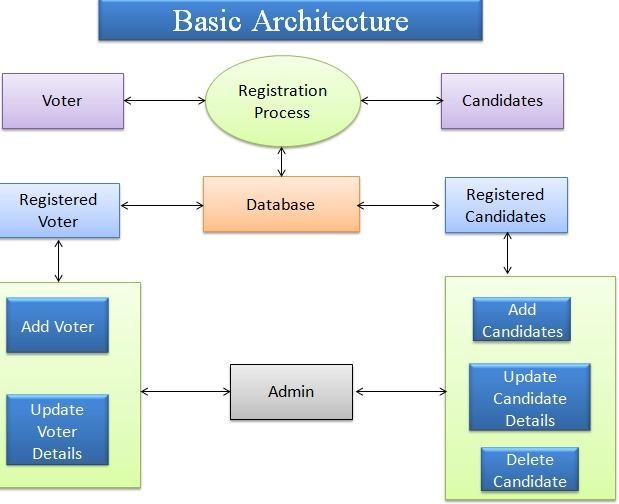
Ensure that votes are encrypted and stored securely in the database, maintaining confidentiality and preventing tampering.

## Result Calculation:

Develop a module that automatically counts votes and displays real-time results once the voting period ends, including percentages and candidate rankings.

## BLOCK DIAGRAM

The diagram illustrates the basic architecture of an online voting system. It involves a **Registration Process** where both voters and candidates register and their details are stored in a central **Database**. The **Admin** manages the system by adding, updating, or deleting voter and candidate information. Registered voters can cast votes, while candidates' details are maintained and updated throughout the process.



# CHAPTER 3

**JAVA PROGRAMMING CONCEPTS**

## Key Java Programming Concepts:

**Object-Oriented Programming (OOP):**

Java is based on OOP principles, which include encapsulation, inheritance, polymorphism, and abstraction. These concepts allow for modular and reusable code.

## Classes and Objects:

Classes are blueprints for creating objects, and objects are instances of classes that hold data and methods.

## Inheritance:

Enables a class to inherit properties and methods from a parent class, promoting code reuse.

## Polymorphism:

Allows methods to perform different tasks based on the object that invokes them, supporting method overloading and overriding.

## Encapsulation:

Data hiding through access modifiers (private, protected, public) ensures that data manipulation is controlled.

## Abstraction:

Focuses on exposing only essential features and hiding unnecessary details, typically through abstract classes and interfaces.

# CHAPTER 4 MODULE DESCRIPTION

## Module 1 User Registration Module:

* Allows voters and candidates to register by submitting their details, such as name, email, and voter ID. Implements OTP and email verification for authentication to ensure valid registration.

## Module 2 Authentication and Login Module:

Ensures secure user login with credentials (username and password). Uses Features like CAPTCHA and OTP-based login for enhanced security, allowing only registered users to access the system.

## Module 3 Voting Module:

* Facilitates the voting process, enabling authenticated users to view candidates and cast their vote. The system ensures one-time voting and securely stores votes using encryption techniques.

## Module 4 Candidate Management Module:

* Allows the admin to add, update, or delete candidate details. Ensures accurate management of candidates participating in the election.

## Module 5 Voter Management Module:

Enables the admin to manage voter records by adding, updating, or deleting voter details. Maintains a clean and accurate database of eligible voters.

# CHAPTER 5 CONCLUSION

The Online Voting System provides a reliable and secure platform for conducting elections digitally, ensuring transparency and efficiency. Through its various modules, such as user registration, authentication, and voting, it simplifies the election process while maintaining data integrity. The system verifies users using OTP and email authentication, enhancing security and preventing unauthorized access. Votes are encrypted and stored securely, ensuring confidentiality and accuracy in tallying. The result calculation module provides real-time results, making the process faster and more accessible. Additionally, the system’s notifications keep users informed about important updates and deadlines. By leveraging modern technologies and security protocols, the system ensures trust and fairness in the election process. Overall, it promotes wider participation by making voting more convenient and secure. This project is a step toward fostering digital democracy and efficient election management. The Online Voting System is designed to be scalable, handling large numbers of voters and candidates efficiently. It also provides administrators with an intuitive dashboard for managing the election process, ensuring smooth operation. The system’s user-friendly interface makes it easy for both voters and candidates to interact with the platform. By reducing human error and enhancing accessibility, the system contributes to the modernization of electoral systems, ensuring that elections are conducted in a fair and secure environment.

6

## REFERENCES:

1. **Oracle Java Documentation**

Official Java language reference and API documentation. <https://docs.oracle.com/en/java/>

## GeeksforGeeks Java Tutorials

Comprehensive guides and problem-solving examples. https://[www.geeksforgeeks.org/java/](http://www.geeksforgeeks.org/java/)

## W3Schools Java Tutorial

Simple and beginner-friendly tutorials with examples. https://[www.w3schools.com/java/](http://www.w3schools.com/java/)

## TutorialsPoint Java Guide

Detailed Java tutorials and coding examples. https://[www.tutorialspoint.com/java/index.htm](http://www.tutorialspoint.com/java/index.htm)

## JavaTpoint Java Tutorials

Covers basic to advanced Java concepts. https://[www.javatpoint.com/java-tutorial](http://www.javatpoint.com/java-tutorial)

## Stack Overflow

* + Community-driven Q&A platform for coding-related queries.
  + <https://stackoverflow.com/>

# APPENDICES APPENDIX A – SOURCE CODE

package votingSystem;

import javax.swing.\*; import java.awt.\*;

import java.awt.event.ActionEvent; import java.awt.event.ActionListener;

public class MainFrame extends JFrame {

// For login

private JTextField usernameField; private JPasswordField passwordField; private JButton loginButton;

// For voting

private ButtonGroup candidateGroup; private VotingSystem votingSystem; private JPanel votingPanel;

public MainFrame() {

votingSystem = new VotingSystem(); setTitle("Online Voting System"); setSize(500, 350);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

// Initially show Login Panel showLoginPanel();

}

// Method to show Login UI private void showLoginPanel() {

JPanel loginPanel = new JPanel();

loginPanel.setLayout(new BoxLayout(loginPanel, BoxLayout.Y\_AXIS)); loginPanel.setBackground(new Color(173, 216, 230)); // Light blue background loginPanel.setBorder(BorderFactory.createEmptyBorder(50, 50, 50, 50)); //

Padding for the content

JLabel usernameLabel = new JLabel("Username:"); usernameField = new JTextField(15);

usernameField.setMaximumSize(usernameField.getPreferredSize()); // To avoid stretching

JLabel passwordLabel = new JLabel("Password:"); passwordField = new JPasswordField(15);

passwordField.setMaximumSize(passwordField.getPreferredSize()); // To avoid stretching

loginButton = new JButton("Login");

// Styling buttons

loginButton.setBackground(new Color(34, 139, 34)); // Green button loginButton.setForeground(Color.WHITE); loginButton.setFont(new Font("Arial", Font.BOLD, 14));

// Add components to the login panel loginPanel.add(usernameLabel); loginPanel.add(usernameField); loginPanel.add(passwordLabel); loginPanel.add(passwordField);

loginPanel.add(Box.createVerticalStrut(20)); // Space between components loginPanel.add(loginButton);

// Set the login panel as the main content setContentPane(loginPanel);

loginButton.addActionListener(new ActionListener() { @Override

public void actionPerformed(ActionEvent e) { String username = usernameField.getText();

String password = new String(passwordField.getPassword()); if (isValidUser(username, password)) {

// On successful login, show voting screen showVotingPanel();

} else {

JOptionPane.showMessageDialog(null, "Invalid Username or

Password");

}

}

});

}

// Method to validate user credentials (hardcoded for simplicity)

private boolean isValidUser(String username, String password) {

return username.equals("user") && password.equals("password"); // Simple validation

}

// Method to show Voting Panel private void showVotingPanel() {

// Remove login panel and set up voting screen getContentPane().removeAll();

setLayout(new BoxLayout(getContentPane(), BoxLayout.Y\_AXIS));

// Add instruction label

JLabel instructionLabel = new JLabel("Please select your favorite candidate and click 'Vote'.");

instructionLabel.setFont(new Font("Arial", Font.PLAIN, 14)); instructionLabel.setForeground(new Color(0, 102, 204)); instructionLabel.setAlignmentX(CENTER\_ALIGNMENT); add(instructionLabel);

candidateGroup = new ButtonGroup(); votingPanel = new JPanel();

votingPanel.setLayout(new BoxLayout(votingPanel, BoxLayout.Y\_AXIS));

// Display 5 candidates with different colors as radio buttons

String[] colors = {"#FF6347", "#90EE90", "#ADD8E6", "#FFD700", "#DDA0DD"};

for (int i = 0; i < votingSystem.getCandidates().size(); i++) { Candidate candidate = votingSystem.getCandidates().get(i);

11

JRadioButton radioButton = new JRadioButton(candidate.getName()); radioButton.setFont(new Font("Arial", Font.PLAIN, 14)); radioButton.setForeground(Color.BLACK); radioButton.setBackground(Color.decode(colors[i])); // Set each candidate's

button color

candidateGroup.add(radioButton); votingPanel.add(radioButton);

}

// Vote and Show Results buttons with colors JButton voteButton = new JButton("Vote");

JButton resultsButton = new JButton("Show Results");

// Styling Vote Button

voteButton.setBackground(new Color(34, 139, 34)); // Green button voteButton.setForeground(Color.WHITE);

voteButton.setFont(new Font("Arial", Font.BOLD, 14)); voteButton.setAlignmentX(CENTER\_ALIGNMENT);

// Styling Results Button

resultsButton.setBackground(new Color(30, 144, 255)); // Blue button resultsButton.setForeground(Color.WHITE); resultsButton.setFont(new Font("Arial", Font.BOLD, 14)); resultsButton.setAlignmentX(CENTER\_ALIGNMENT);

voteButton.addActionListener(new ActionListener() { @Override

public void actionPerformed(ActionEvent e) {

for (int i = 0; i < votingSystem.getCandidates().size(); i++) { JRadioButton radioButton = (JRadioButton)

candidateGroup.getElements().nextElement(); if (radioButton.isSelected()) {

votingSystem.voteForCandidate(i); JOptionPane.showMessageDialog(null, "Vote Cast Successfully!"); showVotingPanel(); // Refresh the voting screen to show updated vote

count

return;

}

}

}

});

JOptionPane.showMessageDialog(null, "Please select a candidate!");

resultsButton.addActionListener(new ActionListener() { @Override

public void actionPerformed(ActionEvent e) { JOptionPane.showMessageDialog(null, votingSystem.getResults());

}

});

add(votingPanel); add(voteButton); add(resultsButton);

// Revalidate and repaint the layout revalidate();

repaint();

}

public static void main(String[] args) { SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new MainFrame().setVisible(true); // Show login screen

}

});

}

}

## VOTING SYSTEM

package votingSystem; import java.util.ArrayList;

public class VotingSystem {

private ArrayList<Candidate> candidates;

public VotingSystem() { candidates = new ArrayList<>();

// Adding 5 candidates with different names candidates.add(new Candidate("Alice")); candidates.add(new Candidate("Bob")); candidates.add(new Candidate("Charlie")); candidates.add(new Candidate("David"));

candidates.add(new Candidate("Eva"));

}

public ArrayList<Candidate> getCandidates() { return candidates;

}

public void voteForCandidate(int index) {

if (index >= 0 && index < candidates.size()) { candidates.get(index).addVote();

}

}

public String getResults() {

StringBuilder results = new StringBuilder("Voting Results:\n\n"); for (Candidate candidate : candidates) {

results.append(candidate.getName()).append(": ").append(candidate.getVotes()).append(" votes\n");

}

return results.toString();

}

}

class Candidate { private String name; private int votes;

public Candidate(String name) {

this.name = name; this.votes = 0;

}

public String getName() { return name;

}

public int getVotes() { return votes;

}

public void addVote() { this.votes++;

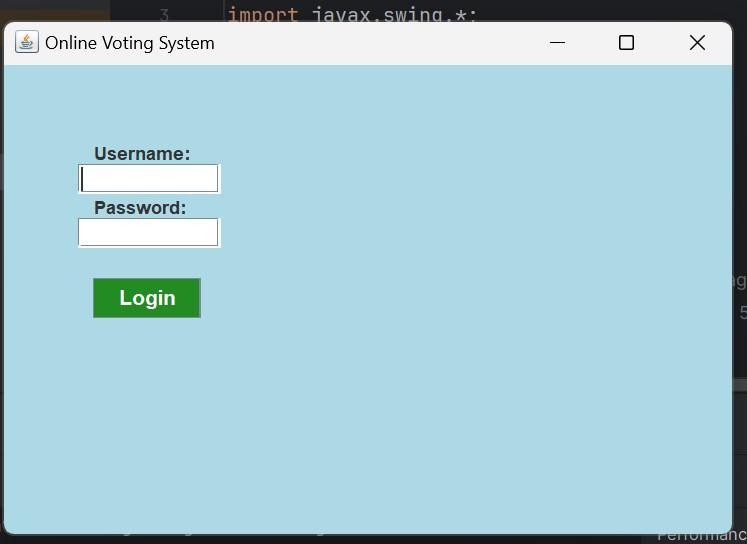
}

}

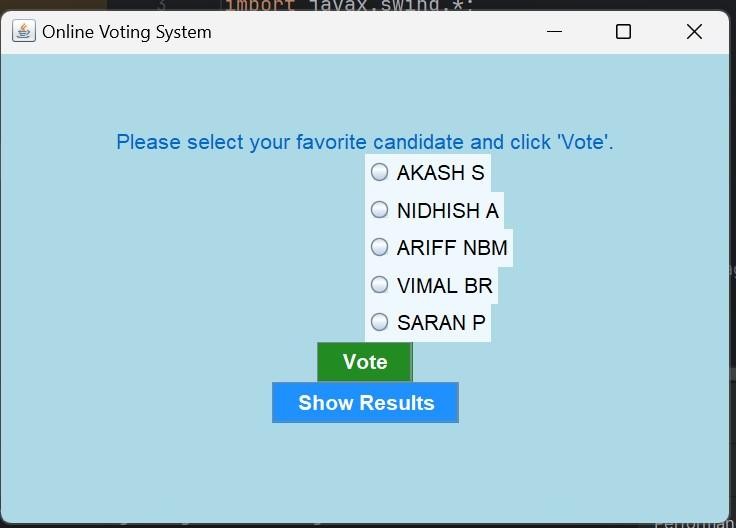
16

# APPENDIX B - SCREENSHOTS

## LOGIN PAGE :



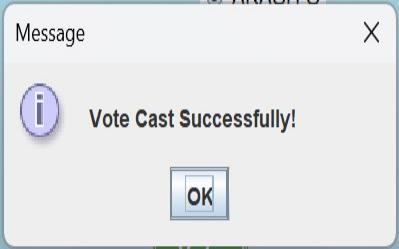
**CANDIDATE LIST :**



## CANDIDATE SELECTION :



**VOTE STATUS :**



## RESULT PAGE :

