# ONLINE VOTING SYSTEM

**A PROJECT REPORT**

***Submitted by***

**AROCKIA JERISH RAJ M**
**(2303811710421011)**

***in partial fulfillment of requirements for the award of the course***

**CGB1201 - JAVA PROGRAMMING**

***In***

**COMPUTER SCIENCE AND ENGINEERING**

**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 **AROCKIA JERISH RAJ M (2303811710421011)** who carriedout the project work during the academic year 2024 - 2025 under my supervision.


**SIGNATURE**

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

**HEAD OF THE DEPARTMENT**

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.


**SIGNATURE**

Mr. M. Saravanan, M.E.,

**SUPERVISOR**

ASSISTANTP ROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 02.12.2024


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 ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201- JAVA PROGRAMMING.**

.

**Signature**

AROCKIA JERISH RAJ M

Place: Samayapuram

Date: 02.12.2024

## ACKNOWLEDGEMENT

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

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for 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 to our project and offering adequate duration in completing our project.

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

I whole heartily thanks to **Dr A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide **Mr. M. SARAVANAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

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

## VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

## MISSION OF DEPARTMENT

**M1: Industry Specific:** To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

**M2: Research:** To prepare students for research-oriented activities.

**M3: Society:** To empower students with the required skills to solve complex technological problems of society.

## PROGRAM EDUCATIONAL OBJECTIVES

### 1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

### 2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

**3. PEO3: Ethics and Values**

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

**PSO 1: Domain Knowledge**

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

**PSO 2: Quality Software**

To apply software engineering principles and practices for developing quality software for scientific and business applications.

**PSO 3: Innovation Ideas**

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

**PROGRAM OUTCOMES (POs)**

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.

# ABSTRACT

An online voting system using Java Swing by combining a clean user interface with logical components to handle voting operations effectively. It manages candidates as a predefined array of strings, with votes tracked using a HashMap to map each candidate to their respective vote count, ensuring efficient data management. To maintain the integrity of the voting process, a HashSet stores the names of voters who have already participated, preventing duplicate votes and upholding fairness. The system's user interface, designed with Swing, uses a JFrame as the main container, divided into a title section, an input panel, and a button panel. The input panel allows users to enter their name in a text field and select a candidate from a JComboBox, while a "Vote" button enables submission. The ActionListener tied to this button validates the input, checks if the voter has already voted, and, if valid, records the vote in the HashMap while adding the voter's name to the HashSet. Feedback mechanisms, such as error messages for duplicate voting or incomplete input and success messages for valid votes, are implemented using JOptionPane, enhancing user experience. A "Show Results" button, also equipped with an ActionListener, displays the current vote counts for all candidates in a formatted dialog box, encapsulating the real-time tallying of results. The Swing-based architecture ensures a clear separation of concerns, with UI components and backend logic interacting seamlessly, making the system modular and easy to expand. This abstraction lays a solid foundation for extending functionalities, such as integrating user authentication, adding encryption for secure voting, or connecting to a database for persistent data storage, while maintaining simplicity and scalability.

**ABSTRACT WITH POs AND PSOs MAPPING**

**CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.**

| ABSTRACT | POs MAPPED | PSOs MAPPED |
|---|---|---|
| An online voting system using Java Swing, providing a user-friendly interface and core functionalities for a simplified voting process. The system manages candidates as a predefined list, tracks vote counts using a HashMap, and ensures voting integrity with a HashSet to prevent duplicate voting. The interface, built with Swing, includes a JFrame divided into a title, input panel, and action buttons, where users can input their name, select a candidate, and cast their vote via a "Vote" button. Validation ensures unique voting, with error or success messages displayed using JOptionPane. A "Show Results" button dynamically presents the total votes for each candidate, encapsulating result tallying within a clean abstraction. The design effectively separates the UI and logic, demonstrating modularity and scalability, while the use of Swing ensures interactivity and ease of future enhancements, such as authentication or persistent data storage. | **PO1 -3**<br>**PO2 -3**<br>**PO3 -3**<br>**PO4 -3**<br>**PO5 -3**<br>**PO6 -3**<br>**PO7 -3**<br>**PO8 -3**<br>**PO9 -3**<br>**PO10 -3**<br>**PO11-3**<br>**PO12 -3** | **PSO1 -3**<br>**PSO2 -3**<br>**PSO3 -3** |

Note: 1- Low, 2-Medium, 3- High

# TABLE OF CONTENTS


| CHAPTER NO. | TITLE | PAGE NO. |
|---|---|---|
| | **ABSTRACT** | **viii** |
| **1** | **INTRODUCTION** | **1** |
| | 1.1 Objective | 1 |
| | 1.2 Overview | 1 |
| | 1.3 Java Programming Concepts | 2 |
| **2** | **PROJECT METHODOLOGY** | **3** |
| | 2.1 Proposed Work | 3 |
| | 2.2 Block Diagram | 3 |
| **3** | **MODULE DESCRIPTION** | **4** |
| | 3.1 User Interface (UI) Module | 4 |
| | 3.2 Logic Module | 4 |
| | 3.3 Data Management Module | 4 |
| | 3.4 Results Display Module | 4 |
| **4** | **CONCLUSION & FUTURE SCOPE** | **5** |
| | 4.1 Conclusion | 5 |
| | 4.2 Future Scope | 5 |
| | **REFERENCES** | **7** |
| | **APPENDIX A(SOURCE CODE)** | **8** |
| | **APPENDIX B(SCREENSHOTS)** | **11** |

# CHAPTER 1

# INTRODUCTION

## 1.1 Objective

The objective of the online voting system is to provide a secure, user-friendly, and efficient platform for conducting elections digitally. It aims to ensure fairness by validating voters and preventing duplicate voting, facilitate accessibility for all users through an intuitive graphical interface, and streamline the vote counting process with real-time results. Additionally, the system seeks to uphold transparency and reliability, making the voting experience seamless while offering scalability for future enhancements like authentication mechanisms, data encryption, and integration with external databases for robust data management.

## 1.2 Overview

The online voting system is a digital application designed to facilitate secure and efficient voting processes. Built using Java Swing, the system provides an interactive graphical user interface (GUI) where voters can enter their name, select a candidate from a predefined list, and cast their votes. It uses data structures like HashMap to manage vote counts and HashSet to track voter participation, ensuring that each voter can cast a vote only once. The system validates inputs, prevents duplicate voting, and provides feedback through error and success messages using JOptionPane. It also includes a feature to display real-time election results, showing the vote counts for all candidates. The system encapsulates key functionalities like input handling, validation, vote recording, and result presentation within a modular design. This foundation can be extended to include advanced features like authentication, encryption, and persistent data storage, making it a scalable and reliable solution for modern digital elections.

## 1.3 Java Programming Concepts

**The basic concepts of Object-Oriented Programming (OOP) are:**

- ✓ **Encapsulation**:The data related to candidates (votes) and voters (voters) is encapsulated within HashMap and HashSet, ensuring that access is controlled and only specific parts of the program can modify them through well-defined actions.

- ✓ **Abstraction**:The program hides implementation details, such as how votes are stored and validated, from the user. The Swing GUI abstracts the underlying logic and presents only the essential functionalities (voting and viewing results) to the user.
- ✓ **Polymorphism**:Polymorphism is observed in the use of ActionListeners for different buttons (e.g., "Vote" and "Show Results"). Both buttons execute different behaviors (actionPerformed logic) depending on the specific action triggered, but they implement the same ActionListener interface.
- ✓ **Inheritance**:While the code doesn't explicitly use inheritance directly in user-defined classes, the use of Swing components (e.g., JFrame, JLabel, JButton) demonstrates inheritance, as these classes inherit properties and methods from their respective parent classes in the Swing library.

**Project related concepts**

- ✓ **GUI Development and User Interaction:**The system uses Java Swing to create an interactive graphical user interface (GUI) with components like JFrame, JLabel, JTextField, JButton, and JComboBox. This ensures a user-friendly experience, allowing voters to easily input their details, select candidates, and view results.
- ✓ **Data Validation and Integrity**:The program validates user input (e.g., ensuring a voter hasn't already voted and that all required fields are filled). It maintains the integrity of the voting process by using a HashSet to track unique voters and a HashMap to store and manage vote counts efficiently.
- ✓ **Event Handling**:The application employs ActionListeners to handle button clicks dynamically. For example, the "Vote" button records votes after validation, while the "Show Results" button displays real-time voting results. This event-driven architecture makes the application responsive and interactive.

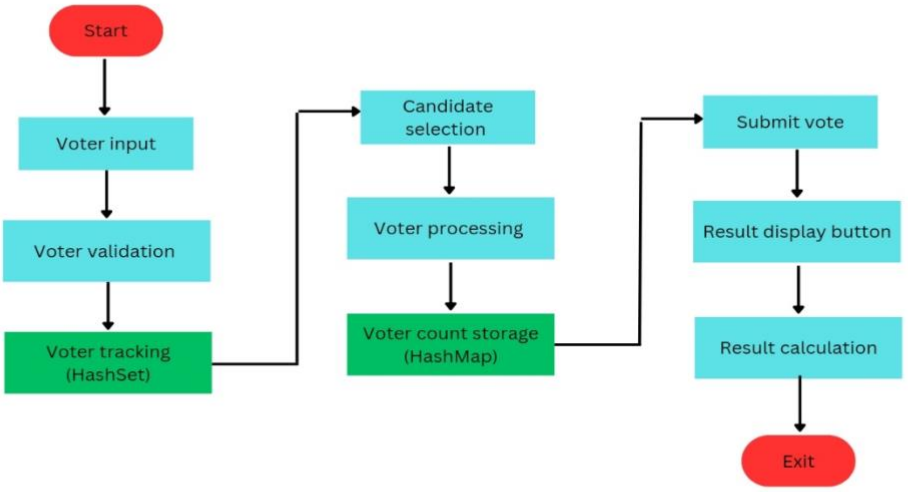
# CHAPTER 2

# PROJECT METHODOLOGY

## 2.1 Proposed Work

The proposed work for this online voting system aims to develop a secure, efficient, and user-friendly digital platform for conducting elections, utilizing Java Swing for a graphical user interface (GUI) that enables voters to cast votes and view results seamlessly. The system will incorporate key functionalities such as user authentication to validate voter identity, prevention of duplicate voting through unique voter tracking using data structures like HashMap and HashSet, and real-time vote tallying with dynamic result display. It will provide clear feedback to users through dialogs and ensure voting integrity. Future enhancements may include encryption for secure vote transmission, database integration for persistent data storage, and role-based access control for voters and administrators, making the system scalable and reliable for modern digital voting needs.

## 2.2 Block Diagram



# CHAPTER 3
# MODULE DESCRIPTION

## 3.1 User Interface(UI) Module

The User Interface (UI) module in the online voting system allows voters to interact with the system. It consists of components such as a text field for entering the voter's name, a combo box for selecting the candidate, and buttons for submitting the vote and viewing the results. The UI ensures smooth interaction by guiding the voter through the voting process, providing clear prompts and messages to inform users if they need to correct any input, like entering a name or choosing a candidate.

## 3.2 Logic Module

The Logic module is the heart of the voting system, responsible for processing the voter's input and ensuring the integrity of the voting process. This module validates the voter's information, ensuring that they have entered a unique name and have not voted previously. It processes the vote by recording the selected candidate and updating their vote count. The Logic module also handles the calculation of the results, ensuring that the final vote count is accurate and up to date for display when requested.

## 3.3 Data Management Module

The Data Management module is responsible for securely managing and storing all the data involved in the voting process. This includes tracking voters using a HashSet to ensure that each voter can only cast one vote and using a HashMap to store and update the vote count for each candidate. The module ensures that all relevant data, including voter details and vote counts, is properly maintained and can be easily accessed or updated throughout the voting session.

## 3.4 Results Display Module

The Results Display module is responsible for showing the voting results to the user. After the voting process is complete, it calculates the total number of votes for each candidate and displays the results in an easy-to-read format. This module ensures that voters can view the current vote standings in real-time, providing transparency and ensuring the integrity of the voting process by presenting accurate and up-to-date results to users.

# CHAPTER 4

# CONCLUSION & FUTURE SCOPE

## 4.1 CONCLUSION

In conclusion, the online voting system designed in this project demonstrates a comprehensive approach to conducting secure, reliable, and transparent elections in an online environment. By incorporating key modules such as the User Interface, Logic, Data Management, and Results Display, the system ensures smooth interaction for voters while maintaining the integrity of the voting process. The User Interface module is intuitive, allowing voters to easily enter their information, select candidates, and submit their votes, while also providing real-time feedback and guidance. The Logic module plays a crucial role in validating voter inputs, ensuring that each voter can only cast one vote, and processing the votes efficiently. It also manages the calculation of results, ensuring that the system is fair and the vote tallying is accurate. The Data Management module securely tracks voter information and stores the votes, preventing fraud and ensuring that the data is easily accessible for future use or verification. Finally, the Results Display module provides a transparent and real-time view of the voting outcomes, ensuring that all stakeholders can observe the election results as they unfold. This holistic design ensures that the system can handle elections in a secure, accurate, and transparent manner, making it a powerful tool for modern-day online voting applications.

## 4.2 FUTURE SCOPE

The future scope of this online voting system is vast, with numerous possibilities for improvement and expansion. One key area for development is the integration of advanced security features, such as multi-factor authentication (MFA) or biometric verification, to further prevent unauthorized access and ensure the identity of voters. Additionally, expanding the system to handle larger-scale elections, including the ability to manage more candidates, multiple voting stations, or even international elections, would increase its versatility. Enhancements in user experience, such as mobile app support and multilingual options, could make the system more accessible to a global audience. Furthermore, incorporating blockchain technology could provide an immutable and transparent record of votes, ensuring higher levels of trust and security. The system could also evolve to allow for real-time analytics, including live voter participation tracking and vote projections, which would enhance the transparency and credibility of the election process. As technology continues to evolve, the

online voting system has the potential to incorporate artificial intelligence for fraud detection and voter behavior analysis, adding another layer of integrity to the system. Overall, the future of online voting systems lies in their ability to adapt to emerging technologies and provide a secure, efficient, and transparent voting experience.

# REFERENCES

## Java Books:


### 1."Head First Java" by Kathy Sierra and Bert Bates

This book is a great resource for beginners learning Java, with a focus on object-oriented programming concepts and real-world application development.

### 2. "Effective Java" by Joshua Bloch

A deeper dive into best practices for writing clean, maintainable Java code. It covers advanced topics like Java collections, concurrency, and design patterns that could be applied to more complex payroll systems.


## Websites:


### 1. GeeksforGeeks - Java Tutorials

- URL: https://www.geeksforgeeks.org/java/
- A comprehensive collection of tutorials on Java, covering topics like classes, objects, inheritance, encapsulation, and more. Great for learning the core concepts of Java and applying them in projects like Online Voting System.

### 2. W3Schools - Java Tutorial

- URL: https://www.w3schools.com/java/
- A beginner-friendly resource that offers tutorials on Java programming, including object-oriented principles and core Java concepts.


## YouTube Links:


### 1. Java for Beginners - Java Brains

- URL: https://www.youtube.com/user/koushks

- Offers Java tutorials from the basics to advanced concepts. The channel provides detailed guides on Java programming, including working with objects and classes, which are crucial for building an Online Voting System.

## APPENDIX A (SOURCE CODE)

```
Import javax.swing.*;
Import java.awt.*;
Import java.awt.event.ActionEvent;
Import java.awt.event.ActionListener;
Import java.util.HashMap;
Import java.util.HashSet;

Public class Main {
   // Candidate names
   Private static final String[] candidates = {“Candidate A”, “Candidate B”, “Candidate C”};
   Private static final HashMap<String, Integer> votes = new HashMap<>(); // Stores votes for
candidates
   Private static final HashSet<String> voters = new HashSet<>(); // Keeps track of voters to ensure
unique voting

   Public static void main(String[] args) {
      // Initialize vote counts for each candidate
      For (String candidate : candidates) {
         Votes.put(candidate, 0);
      }

      // Create the main frame
      JFrame frame = new JFrame(“Online Voting System”);
      Frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      Frame.setSize(500, 400);
      Frame.setLayout(new BorderLayout());

      // Title Label
      JLabel title = new JLabel(“Vote for Your Candidate”, JLabel.CENTER);
      Title.setFont(new Font(“Arial”, Font.BOLD, 18));
      Frame.add(title, BorderLayout.NORTH);

      // Panel for user input
      JPanel inputPanel = new JPanel();
      inputPanel.setLayout(new GridLayout(3, 2, 10, 10));

      // Voter name input
      JLabel voterLabel = new JLabel(“Enter Your Name:”);
      JTextField voterNameField = new JTextField();
      inputPanel.add(voterLabel);
```

```
    // Candidate selection
      JLabel candidateLabel = new JLabel("Choose Your Candidate:");
      JComboBox<String> candidateComboBox = new JComboBox<>(candidates);
      inputPanel.add(candidateLabel);
      inputPanel.add(candidateComboBox);

      // Vote button
      JButton voteButton = new JButton("Vote");
      inputPanel.add(new JLabel()); // Empty cell for spacing
      inputPanel.add(voteButton);
      frame.add(inputPanel, BorderLayout.CENTER);

      // Results Button
      JButton resultsButton = new JButton("Show Results");
      resultsButton.setFont(new Font("Arial", Font.BOLD, 14));
      frame.add(resultsButton, BorderLayout.SOUTH);

      // ActionListener for the vote button
      voteButton.addActionListener(new ActionListener() {
         @Override
         Public void actionPerformed(ActionEvent e) {
            String voterName = voterNameField.getText().trim();
            String chosenCandidate = (String) candidateComboBox.getSelectedItem();

            If (voterName.isEmpty()) {
               JOptionPane.showMessageDialog(frame, "Please enter your name!", "Error",
JOptionPane.ERROR_MESSAGE);
               Return;
            }

            If (voters.contains(voterName)) {
               JOptionPane.showMessageDialog(frame, "You have already voted!", "Error",
JOptionPane.ERROR_MESSAGE);
               Return;
            }

            // Record the vote
            Voters.add(voterName);
            Votes.put(chosenCandidate, votes.get(chosenCandidate) + 1);

            JOptionPane.showMessageDialog(frame, "Thank you for voting!", "Success",
JOptionPane.INFORMATION_MESSAGE);
            voterNameField.setText(""); // Clear the voter name field
         }
```

```
// ActionListener for the results button
        resultsButton.addActionListener(new ActionListener() {
            @Override
            Public void actionPerformed(ActionEvent e) {
                // Display voting results
                StringBuilder resultMessage = new StringBuilder("Voting Results:\n");
                For (String candidate : candidates) {
                    resultMessage.append(candidate).append(": ").append(votes.get(candidate)).append(" votes\n");
                }

                JOptionPane.showMessageDialog(frame, resultMessage.toString(), "Results",
JOptionPane.INFORMATION_MESSAGE);
            }
        });

        // Show the frame
        Frame.setVisible(true);
    }
}
```

# APPENDIX B(SCREENSHOTS)