Koneru Lakshmaiah Education Foundation

(Deemed to be University)

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

Election Voting System

SUBMITTED BY:

NAME	ID NUMBER
Keerti Krishna Sreenivas S	2300031039

UNDER THE GUIDANCE OF

Mr. Mohammad Ashique Azad

Assistant Professor, CSE.

KL UNIVERSITY

Green fields, Vaddeswaram – 522 502

Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES-1

CERTIFICATE

This is to certify that the project based laboratory report entitled "**Election Voting System**" submitted by **Mr. Keerti Krishna Sreenivas S,** bearing Regd. No.2300031039 respectively to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in "**Computational Thinking for Object Oriented Design** "course in IB Tech II Semester, is a Bonafede record of the work carried out by him under my supervision during the academic year 2023 – 2024.

PROJECT SUPERVISOR   **HEAD OF THE DEPARTMENT**

 Mr. Mohammad Ashique Azad   Dr. D. Haritha

ACKNOWLEDGEMENTS

ABSTRACT

The emergence of digital technologies has revolutionized various sectors, including the electoral process. In line with this advancement, this project presents the design and implementation of an Election Voting System (EVS) using Java programming language. The system aims to streamline the voting process, enhance efficiency, and ensure the integrity and security of electoral operations. The EVS employs a client-server architecture to facilitate secure communication between the voter and the central server. The system provides a user-friendly interface for voters to cast their votes electronically, eliminating the need for manual ballot papers and reducing the likelihood of errors. Additionally, the system ensures anonymity and confidentiality, preserving the principles of democratic elections. In conclusion, the Election Voting System developed in Java provides a robust platform for conducting elections in a digital environment. By leveraging advanced technologies, the system offers a secure, efficient, and transparent voting experience, thereby contributing to the integrity and credibility of democratic processes.

INDEX

INTRODUCTION

This project implements an Election Voting System applet using Java Swing, showcasing candidate lists and vote status. It offers a user-friendly interface for tracking candidates and their votes across different regions. Through graphical presentation, it aids in understanding regional voting patterns.

Concepts:

Class:

A class is a blueprint for creating objects in Java. It defines the data and methods common to all objects of a certain kind..

Syntax:

```
public class ClassName {
    // Fields (variables)
    // Constructors
    // Methods
}
```

Method:

A method is a block of code that performs a specific task. It is defined within a class and can be called to execute its functionality.

Syntax:

```
returnType methodName(parameter1Type parameter1, parameter2Type parameter2, ...) {
 // Method body
}
```

Constructor:

A constructor is a special type of method that is called when an object is created. It initializes the object's state.

Syntax:

```
public ClassName(parameter1Type parameter1, parameter2Type parameter2, ...) {

 // Constructor body

}
```

Inheritance:

Inheritance is a mechanism in Java where a class inherits properties and behaviors from another class. It allows code reusability and establishes a parent-child relationship between classes.

Syntax:

```
public class Subclass extends Superclass {

// Subclass members

}
```

Static Method:

A static method belongs to the class rather than to instances of the class. It can be called without creating an instance of the class.

Syntax:

```
public static returnType methodName(parameter1Type parameter1, parameter2Type parameter2, ...) {

// Method body

}
```

AIM:

The aim of the project is to develop a digitalized election voting system applet to track candidates and their votes according to regions.

Advantages:

The project offers several advantages, including enhanced efficiency by streamlining candidate and vote tracking, improved transparency through digitalization, increased accessibility to voting information, real-time updates for informed decision-making, and reduced errors through automation, ultimately leading to a more reliable election process

Disadvantages:

While the project presents several advantages, it's also important to acknowledge potential drawbacks. One concern is the project's reliance on technology, making it vulnerable to technical issues like software glitches or hardware failures, which could disrupt the election process. Additionally, the digitalized system introduces security risks, such as cyber attacks or unauthorized access, posing threats to the confidentiality and integrity of voter data. There's also a risk of exacerbating the digital divide, as some voters may lack access to the necessary technology or internet connectivity, potentially excluding certain demographics from participation. Resistance to change may further complicate implementation efforts, particularly among those less familiar with technology. Moreover, the project may incur substantial costs for development, infrastructure, and maintenance, raising financial considerations for electoral authorities. Lastly, legal and ethical questions regarding data privacy, voter authentication, and the validity of electronic voting methods may arise, necessitating robust regulatory frameworks and ethical guidelines. Balancing the benefits of digitalization with these potential challenges is essential to ensure the fairness, inclusivity, and security of the electoral process.

Future Enhancements:

- 1. Security
- 2. Accessibility
- 3. Scalability
- 4. Usability

- 5. Integration
- 6. Efficiency
- 7. Interactivity
- 8. Customization
- 9. Data Analytics
- 10. Mobile Compatibility

Security: Future enhancements could involve the widespread adoption of decentralized identity systems, blockchain-based authentication, and quantum-resistant encryption to ensure robust protection against emerging threats.

Accessibility: Advancements might include the development of neurodiversity-friendly interfaces, such as customizable sensory experiences for individuals with autism, and the integration of assistive technologies like exoskeletons for physical accessibility.

Scalability: Future enhancements could see the implementation of edge computing solutions, enabling processing and storage closer to the point of data generation to reduce latency and support massive scalability for IoT applications.

SYSTEM REQUIREMENTS

➢ SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows:

Language   : Java

Operating System**:** Windows Xp or later.
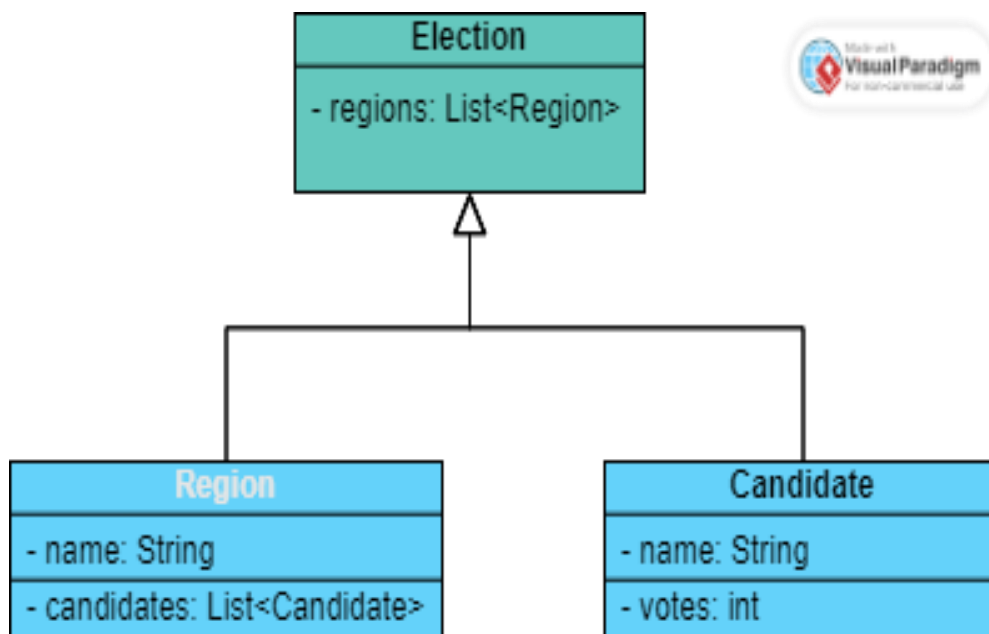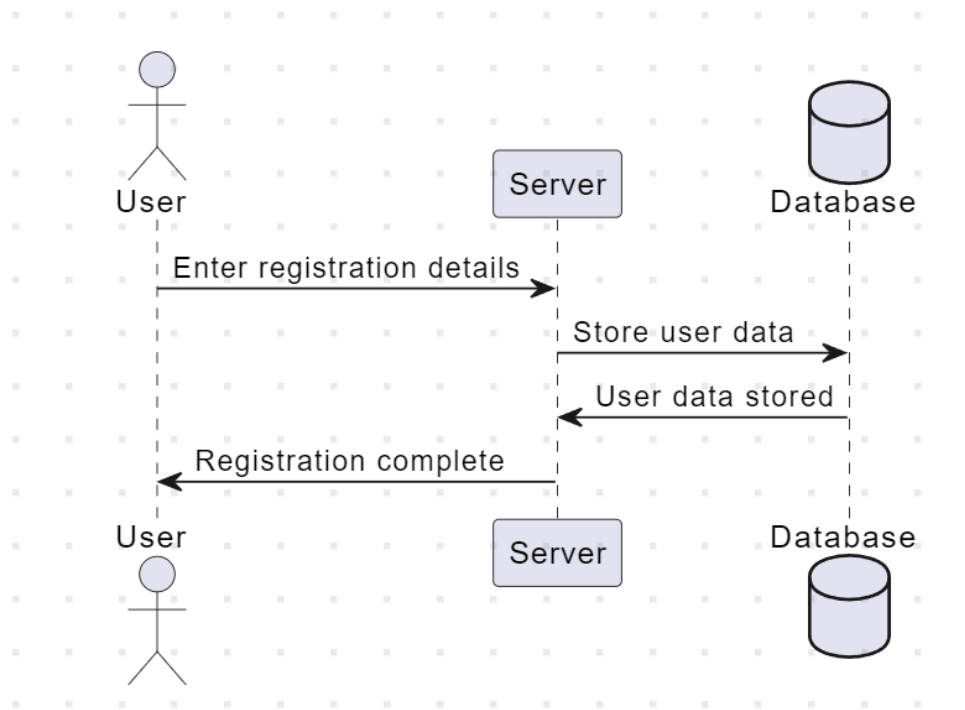
                         Mac OS

➢ HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

RAM   : 8GB

Processor   : RYZEN 5

CLASS DIAGRAM:

ALGORITHM

❖ Initialize a **hashmap candidatesMap** to store candidates and their corresponding parties.

❖ Initialize a **hashmap regionVotesMap** to store votes for each region and party.

❖ Create a GUI with two panels: one for candidate list and one for vote status.

❖ Display candidate list in the left panel with party and candidate names.

❖ When the "Vote" button is clicked, prompt user to select a party to vote for.

❖ If a party is selected, prompt user to select a region to vote in.

❖ Increment the vote count for the selected party in the selected region.

❖ Update the vote status display in the right panel with the latest vote counts.

❖ Repeat steps 5-8 until the voting process is complete.

❖ Display the final vote status with the winner(s) of each region.

IMPLEMENTATION

```java
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.HashMap;


public class ElectionVotingSystemApplet extends JFrame implements ActionListener {


    private HashMap<String, String> candidatesMap = new HashMap<>();

    private HashMap<String, HashMap<String, Integer>> regionVotesMap = new
HashMap<>();

    private JTextArea candidateListArea;

    private JTextArea voteStatusArea;


    private final String[] parties = {"BJP", "Congress", "JD(S)"};

    private final String[] candidates = {"Narendra Modi", "Rahul Gandhi", "Kumaraswamy"};

    private final String[] regions = {"Region A", "Region B", "Region C"};


    public ElectionVotingSystemApplet() {

        setTitle("Election Voting System");

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
   setSize(1000, 600);

// Set background colors

Color backgroundColor = new Color(240, 240, 240);

Color buttonColor = new Color(0, 153, 204);


JPanel leftPanel = new JPanel();

leftPanel.setBackground(backgroundColor);

JPanel rightPanel = new JPanel();

rightPanel.setBackground(backgroundColor);


setLayout(new GridLayout(1, 2));


leftPanel.setLayout(new BorderLayout());

rightPanel.setLayout(new BorderLayout());


JLabel leftHeaderLabel = new JLabel("<html><b>Candidate List</b></html>");

leftHeaderLabel.setFont(new Font("Arial", Font.BOLD, 20));

leftHeaderLabel.setHorizontalAlignment(JLabel.CENTER);

leftPanel.add(leftHeaderLabel, BorderLayout.NORTH);


candidateListArea = new JTextArea();

candidateListArea.setEditable(false);

candidateListArea.setFont(new Font("Arial", Font.PLAIN, 16));
```

```java
candidateListArea.setBackground(Color.WHITE);

JScrollPane leftScrollPane = new JScrollPane(candidateListArea);

leftPanel.add(leftScrollPane, BorderLayout.CENTER);


JButton voteButton = new JButton("Vote");

voteButton.addActionListener(this);

voteButton.setBackground(buttonColor);

leftPanel.add(voteButton, BorderLayout.SOUTH);


JLabel rightHeaderLabel = new JLabel("Vote Status");

rightHeaderLabel.setFont(new Font("Arial", Font.BOLD, 20));

rightHeaderLabel.setHorizontalAlignment(JLabel.CENTER);

rightPanel.add(rightHeaderLabel, BorderLayout.NORTH);

voteStatusArea = new JTextArea();

voteStatusArea.setEditable(false);

voteStatusArea.setFont(new Font("Arial", Font.PLAIN, 16));

voteStatusArea.setBackground(Color.WHITE);

JScrollPane rightScrollPane = new JScrollPane(voteStatusArea);

rightPanel.add(rightScrollPane, BorderLayout.CENTER);


add(leftPanel);

add(rightPanel);
```

```java
  // Initialize candidatesMap with candidates and parties

  for (int i = 0; i < parties.length; i++) {

    candidatesMap.put(parties[i], candidates[i]);

  }

  // Initialize regionVotesMap

  for (String region : regions) {

    HashMap<String, Integer> regionVotes = new HashMap<>();

    for (String party : parties) {

      regionVotes.put(party, 0);

    }

    regionVotesMap.put(region, regionVotes);

  }


  updateCandidateList();

  updateVoteStatus();

  setVisible(true);

}


public void actionPerformed(ActionEvent e) {

  String selectedParty = (String) JOptionPane.showInputDialog(

      this,

      "Select a party to vote for:",

      "Vote",
```

```java
        JOptionPane.QUESTION_MESSAGE,

        null,

        parties,

        parties[0]);

    if (selectedParty != null && !selectedParty.isEmpty()) {

        voteForParty(selectedParty);

    }

}


private void voteForParty(String party) {

    String selectedRegion = (String) JOptionPane.showInputDialog(

        this,

        "Select a region to vote in:",

        "Vote",

        JOptionPane.QUESTION_MESSAGE,

        null,

        regions,

        regions[0]);


    if (selectedRegion != null && !selectedRegion.isEmpty()) {

        HashMap<String, Integer> regionVotes = regionVotesMap.get(selectedRegion);

        regionVotes.put(party, regionVotes.get(party) + 1);

        updateVoteStatus();
```

```java
        }

    }

    private void updateCandidateList() {

        StringBuilder candidateListBuilder = new StringBuilder();

        candidateListBuilder.append("Candidate List:\n");

        for (String party : parties) {

            candidateListBuilder.append(party).append(" -
").append(candidatesMap.get(party)).append("\n");

        }

        candidateListArea.setText(candidateListBuilder.toString());

    }


    private void updateVoteStatus() {

        StringBuilder statusBuilder = new StringBuilder();

        statusBuilder.append("Vote Status:\n");

        for (String region : regions) {

            statusBuilder.append(region).append(":\n");

            HashMap<String, Integer> regionVotes = regionVotesMap.get(region);

            for (String party : parties) {

                statusBuilder.append(candidatesMap.get(party)).append("
(").append(party).append(")").append("        -        ").append(regionVotes.get(party)).append("
votes\n");
```

```java
        }

        statusBuilder.append("\n");

    }

    voteStatusArea.setText(statusBuilder.toString()); }

public static void main(String[] args) {

    SwingUtilities.invokeLater(ElectionVotingSystemApplet::new);

}

}
```
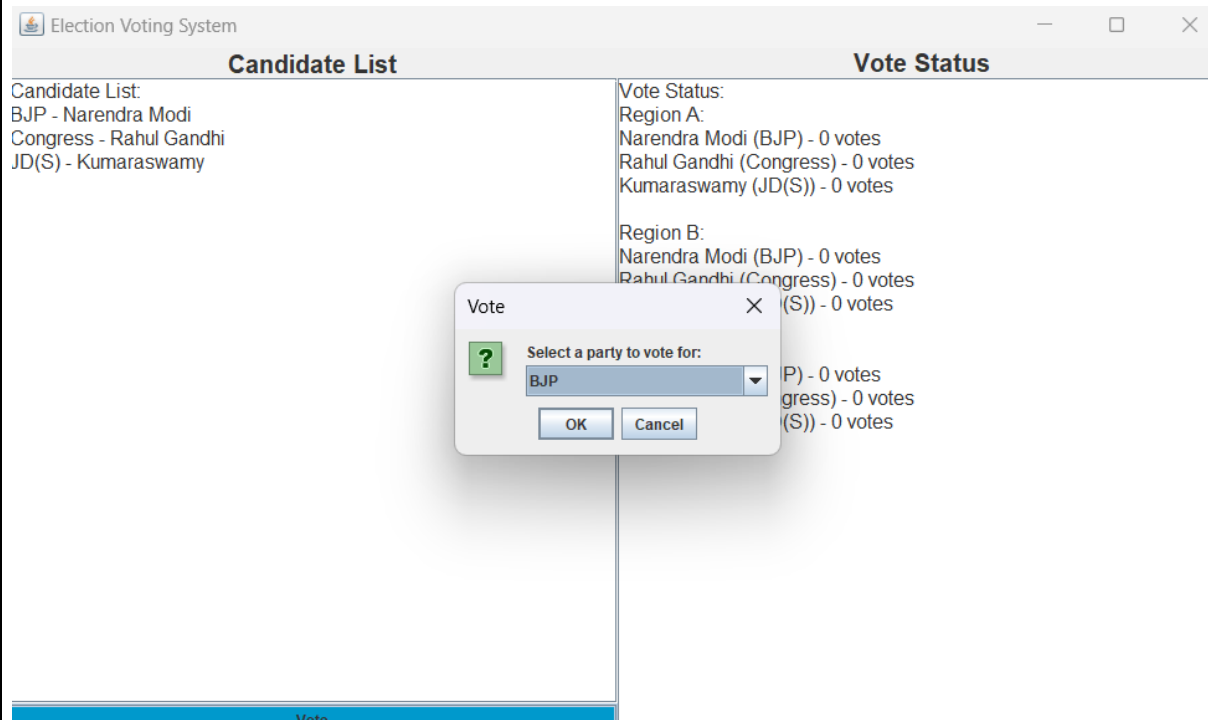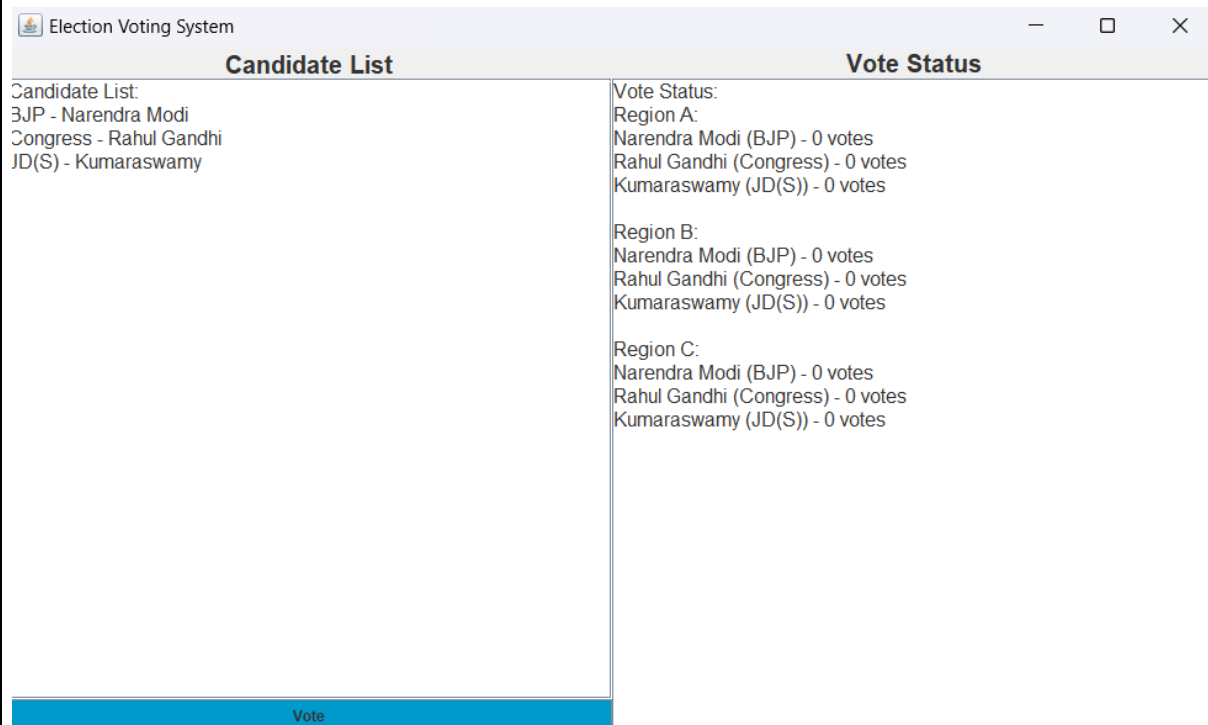
INTEGRATION AND SYSTEM TESTING

OUTPUTS :

Screen Shots:

Election Voting System

Candidate List
Vote Status

Candidate List:
BJP - Narendra Modi
Congress - Rahul Gandhi
JD(S) - Kumaraswamy

Vote Status:
Region A:
Narendra Modi (BJP) - 0 votes
Rahul Gandhi (Congress) - 0 votes
Kumaraswamy (JD(S)) - 0 votes

Region B:
Narendra Modi (BJP) - 0 votes
Rahul Gandhi (Congress) - 0 votes
(S)) - 0 votes

P) - 0 votes
gress) - 0 votes
(S)) - 0 votes

Vote

Vote

Select a region to vote in:
Region A

OK   Cancel

Election Voting System

Candidate List
Vote Status

Candidate List:
BJP - Narendra Modi
Congress - Rahul Gandhi
JD(S) - Kumaraswamy

Vote Status:
Region A:
Narendra Modi (BJP) - 1 votes
Rahul Gandhi (Congress) - 0 votes
Kumaraswamy (JD(S)) - 0 votes

Region B:
Narendra Modi (BJP) - 0 votes
Rahul Gandhi (Congress) - 0 votes
Kumaraswamy (JD(S)) - 0 votes

Region C:
Narendra Modi (BJP) - 0 votes
Rahul Gandhi (Congress) - 0 votes
Kumaraswamy (JD(S)) - 0 votes

Vote

CONCLUSION

The Java project for an election voting system concluded with the successful creation of an Android application that allows students to vote online using a mobile device. The application adheres to the software development life cycle and incorporates a 3-step security process to prevent phishing attacks. This provides a convenient and secure way for students to vote, addressing the issue of low voter turnout due to the inconvenience of traditional voting methods. The project highlights the potential of smart technologies in enhancing democratic processes and the importance of robust security measures in online voting systems.