Source Code:

import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

import java.util.HashMap;

import java.awt.geom.Point2D;

public class oppa extends JFrame implements ActionListener {

// Declare components for input

private JLabel voterIdLabel, titleLabel;

private JTextField voterIdField;

private JButton submitButton, showButton;

// Declare components for displaying the PC (Photo, Candidate Name, Party)

private JPanel candidatePanel; // Panel to hold candidate radio buttons and images

private JRadioButton[] candidateButtons;

private ButtonGroup candidateGroup;

// Candidate data

private String[] candidateNames = {"Candidate 1", "Candidate 2", "Candidate 3", "Candidate 4", "Candidate 5","Candidate 6"};

private String[] candidateParties = {"Party A", "Party B", "Party C", "Party D", "Party E","Party F"};

private String[] candidateImagePaths = {

"C:\\Users\\nagas\\Downloads\\Candidate1.jpeg",

"C:\\Users\\nagas\\Downloads\\Candidate2.jpeg",

"C:\\Users\\nagas\\Downloads\\Candidate3.jpeg",

"C:\\Users\\nagas\\Downloads\\Candidate4.jpeg",

"C:\\Users\\nagas\\Downloads\\Candidate5.jpeg",

"C:\\Users\\nagas\\Downloads\\Candidate6.jpg"

};

class GradientPanel extends JPanel {

private Color color1 = new Color(255, 153, 51); // Saffron (Orange)

private Color color2 = Color.WHITE; // White

private Color color3 = new Color(76, 175, 80); // Green

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

Graphics2D g2d = (Graphics2D) g;

// Define the points for the gradient

Point2D start = new Point2D.Float(0, 0);

Point2D end = new Point2D.Float(getWidth(), getHeight());

// Define color stops: starting at 0% (color1), 50% (color2), and 100% (color3)

float[] fractions = {0.0f, 0.5f, 1.0f};

Color[] colors = {color1, color2, color3};

// Create a LinearGradientPaint with three color stops

LinearGradientPaint gradient = new LinearGradientPaint(start, end, fractions, colors);

// Set the gradient as the paint

g2d.setPaint(gradient);

g2d.fillRect(0, 0, getWidth(), getHeight()); // Fill the panel with the gradient

}

}

public oppa() {

// Set up the frame

setTitle("Voting System");

setSize(800, 600);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null); // Center the window

setLayout(new FlowLayout());

// Set Background Color

//getContentPane().setBackground(new Color(220, 220, 255)); // Light blue background

GradientPanel gradientPanel = new GradientPanel(); // Create the gradient panel

setContentPane(gradientPanel); // Set it as the content pane

// Title Label

titleLabel = new JLabel("Voting System");

titleLabel.setFont(new Font("Arial", Font.BOLD, 24));

titleLabel.setForeground(new Color(0, 102, 204)); // Dark blue color

add(titleLabel);

// Voter ID Input

voterIdLabel = new JLabel("Voter ID:");

voterIdField = new JTextField(20);

add(voterIdLabel);

add(voterIdField);

// Buttons

submitButton = new JButton("Cast Vote");

showButton = new JButton("Show All Votes");

// Customize button colors

submitButton.setBackground(new Color(0, 153, 51)); // Green color

submitButton.setForeground(Color.WHITE);

showButton.setBackground(new Color(0, 102, 204)); // Blue color

showButton.setForeground(Color.WHITE);

// Add action listeners

submitButton.addActionListener(this);

showButton.addActionListener(e -> authenticateAndLoadData());

add(submitButton);

add(showButton);

// Create panel for candidates

candidatePanel = new JPanel();

candidatePanel.setLayout(new GridLayout(candidateNames.length, 1)); // Adjust layout as needed

candidateButtons = new JRadioButton[candidateNames.length];

candidateGroup = new ButtonGroup();

// Load candidate data and create radio buttons

loadCandidateData();

// Add candidate panel to the frame

JScrollPane scrollPane = new JScrollPane(candidatePanel);

scrollPane.setPreferredSize(new Dimension(600, 300)); // Set size of scroll pane

add(scrollPane);

setVisible(true); // Make the frame visible

}

private void loadCandidateData() {

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

// Create radio button for each candidate

candidateButtons[i] = new JRadioButton(candidateNames[i] + " (" + candidateParties[i] + ")");

candidateGroup.add(candidateButtons[i]);

// Create a panel to hold the image and radio button

JPanel candidateRow = new JPanel();

candidateRow.setLayout(new FlowLayout(FlowLayout.LEFT)); // Align left

// Load and scale image

ImageIcon icon = new ImageIcon(new ImageIcon(candidateImagePaths[i]).getImage().getScaledInstance(100, 100, Image.SCALE\_SMOOTH));

JLabel candidateImage = new JLabel(icon);

candidateRow.add(candidateImage);

candidateRow.add(candidateButtons[i]);

candidatePanel.add(candidateRow); // Add to the main panel

}

}

@Override

public void actionPerformed(ActionEvent e) {

String voterId = voterIdField.getText();

if (!authenticateVoter(voterId)) {

JOptionPane.showMessageDialog(this, "Invalid Voter ID!", "Authentication Error", JOptionPane.ERROR\_MESSAGE);

return;

}

int selectedCandidateIndex = getSelectedCandidateIndex();

if (selectedCandidateIndex == -1) {

JOptionPane.showMessageDialog(this, "Please select a candidate!", "Selection Error", JOptionPane.ERROR\_MESSAGE);

return;

}

insertVote(voterId, selectedCandidateIndex + 1); // Candidate IDs are assumed to be 1-based.

}

private boolean authenticateVoter(String voterId) {

return voterId.matches("V\\d{3}"); // Example: V001 or V123 format

}

private int getSelectedCandidateIndex() {

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

if (candidateButtons[i].isSelected()) {

return i;

}

}

return -1;

}

private void authenticateAndLoadData() {

String password = JOptionPane.showInputDialog(this, "Enter password to view vote details:");

if ("123456".equals(password)) {

loadDataFromDatabase(); // Call to load voty from database

} else {

JOptionPane.showMessageDialog(this, "Incorrect password!", "Authentication Failed", JOptionPane.ERROR\_MESSAGE);

}

}

private void insertVote(String voterId, int candidateId) {

String url = "jdbc:mysql://localhost:3306/Voting";

String user = "root";

String password = "Mysql@2005";

try (Connection connection = DriverManager.getConnection(url, user, password)) {

String sql = "INSERT INTO voty (voter\_id, candidate\_id) VALUES (?, ?)";

try (PreparedStatement preparedStatement = connection.prepareStatement(sql)) {

preparedStatement.setString(1, voterId);

preparedStatement.setInt(2, candidateId);

int rowsAffected = preparedStatement.executeUpdate();

JOptionPane.showMessageDialog(this, "Vote cast successfully! " + rowsAffected + " row(s) inserted.");

}

clearFields();

} catch (SQLException ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(this, "Error casting vote.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void loadDataFromDatabase() {

String url = "jdbc:mysql://localhost:3306/Voting";

String user = "root";

String password = "Mysql@2005";

try (Connection connection = DriverManager.getConnection(url, user, password)) {

String sql = "SELECT voter\_id, candidate\_id FROM voty";

Statement statement = connection.createStatement();

ResultSet resultSet = statement.executeQuery(sql);

// Create a table model for displaying vote data

DefaultTableModel voteTableModel = new DefaultTableModel(new String[]{"Voter ID", "Candidate ID"}, 0);

JTable voteTable = new JTable(voteTableModel);

JScrollPane voteScrollPane = new JScrollPane(voteTable);

voteScrollPane.setPreferredSize(new Dimension(600, 300)); // Adjust size as needed

while (resultSet.next()) {

String voterId = resultSet.getString("voter\_id");

int candidateId = resultSet.getInt("candidate\_id");

voteTableModel.addRow(new Object[]{voterId, candidateId});

}

resultSet.close();

statement.close();

if (voteTableModel.getRowCount() == 0) {

JOptionPane.showMessageDialog(this, "No votes found in the database.", "Info", JOptionPane.INFORMATION\_MESSAGE);

} else {

JOptionPane.showMessageDialog(this, voteScrollPane, "Vote Details", JOptionPane.INFORMATION\_MESSAGE);

}

} catch (SQLException e) {

e.printStackTrace();

JOptionPane.showMessageDialog(this, "Error loading votes from database.", "Error", JOptionPane.ERROR\_MESSAGE);

}

}

private void clearFields() {

voterIdField.setText("");

candidateGroup.clearSelection(); // Clear selected candidate

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new oppa());

}

}