Online Voting System

Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.security.MessageDigest;
import java.util.*;
import java.util.List;


public class VotingApp extends JFrame {
    Connection conn;

    public VotingApp() {
        connectToDatabase();
        createTables();
        showHomePage();
    }

    void connectToDatabase() {
        try {
            Class.forName("org.sqlite.JDBC");
            conn = DriverManager.getConnection("jdbc:sqlite:voting.db");
        } catch (Exception e) {
            e.printStackTrace();
        }
```

```
    }

    void createTables() {
        try (Statement stmt = conn.createStatement()) {
            stmt.executeUpdate("CREATE TABLE IF NOT EXISTS users (id INTEGER PRIMARY KEY AUTOINCREMENT, username TEXT UNIQUE, password TEXT, adhar TEXT, phone TEXT, dob TEXT, voted INTEGER DEFAULT 0)");
            stmt.executeUpdate("CREATE TABLE IF NOT EXISTS votes (party TEXT, count INTEGER)");
            stmt.executeUpdate("CREATE TABLE IF NOT EXISTS status (voting INTEGER, resultDeclared INTEGER)");
            ResultSet rs = stmt.executeQuery("SELECT COUNT(*) as count FROM status");
            if (rs.next() && rs.getInt("count") == 0) {
                stmt.executeUpdate("INSERT INTO status (voting, resultDeclared) VALUES (0, 0)");
                stmt.executeUpdate("INSERT INTO votes (party, count) VALUES ('BJP', 0), ('Congress', 0), ('AAP', 0), ('Others', 0)");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void showHomePage() {
        getContentPane().removeAll();
        setTitle("Voting App - Home");
        setSize(300, 250);
        setLayout(new GridLayout(4, 1));

        JButton userLoginBtn = new JButton("User Login");
        JButton adminLoginBtn = new JButton("Admin Login");
        JButton registerBtn = new JButton("Register");
```

```
    JButton resetPassBtn = new JButton("Reset Password");

    add(userLoginBtn);
    add(adminLoginBtn);
    add(registerBtn);
    add(resetPassBtn);

    userLoginBtn.addActionListener(e -> showUserLogin());
    adminLoginBtn.addActionListener(e -> showAdminLogin());
    registerBtn.addActionListener(e -> showRegisterPage());
    resetPassBtn.addActionListener(e -> showResetPasswordPage());

    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setVisible(true);
    revalidate();
    repaint();
}

void showUserLogin() {
    getContentPane().removeAll();
    setTitle("User Login");
    setSize(400, 200);
    setLayout(new GridLayout(4, 2));

    JTextField username = new JTextField();
    JPasswordField password = new JPasswordField();
    JButton loginBtn = new JButton("Login");
    JButton backBtn = new JButton("Back");
```

```
        add(new JLabel("Username:"));
        add(username);
        add(new JLabel("Password:"));
        add(password);
        add(loginBtn);
        add(backBtn);

        loginBtn.addActionListener(e -> {
            String user = username.getText();
            String pass = new String(password.getPassword());

            try {
                PreparedStatement ps = conn.prepareStatement("SELECT * FROM users WHERE
username=? AND password=?");
                ps.setString(1, user);
                ps.setString(2, hash(pass));
                ResultSet rs = ps.executeQuery();
                if (rs.next()) {
                    showUserPanel(user);
                } else {
                    JOptionPane.showMessageDialog(this, "Invalid credentials");
                }
            } catch (Exception ex) {
                ex.printStackTrace();
            }
        });

        backBtn.addActionListener(e -> showHomePage());
        revalidate();
        repaint();
```

```
}

void showAdminLogin() {
    getContentPane().removeAll();
    setTitle("Admin Login");
    setSize(400, 200);
    setLayout(new GridLayout(3, 2));

    JTextField username = new JTextField();
    JPasswordField password = new JPasswordField();
    JButton loginBtn = new JButton("Login");
    JButton backBtn = new JButton("Back");

    add(new JLabel("Admin Username:"));
    add(username);
    add(new JLabel("Password:"));
    add(password);
    add(loginBtn);
    add(backBtn);

    loginBtn.addActionListener(e -> {
        String user = username.getText();
        String pass = new String(password.getPassword());

        if (user.equals("admin") && pass.equals("admin")) {
            showAdminPanel();
        } else {
            JOptionPane.showMessageDialog(this, "Invalid admin credentials");
        }
```

```
    });

    backBtn.addActionListener(e -> showHomePage());
    revalidate();
    repaint();
}

void showRegisterPage() {
    getContentPane().removeAll();
    setTitle("Register");
    setSize(400, 400);
    setLayout(new GridLayout(7, 2));

    JTextField username = new JTextField();
    JPasswordField password = new JPasswordField();
    JTextField adhar = new JTextField();
    JTextField phone = new JTextField();
    JTextField dob = new JTextField();

    JButton registerBtn = new JButton("Register");
    JButton backBtn = new JButton("Back");

    add(new JLabel("Username:"));
    add(username);
    add(new JLabel("Password:"));
    add(password);
    add(new JLabel("Aadhar (12 digits):"));
    add(adhar);
    add(new JLabel("Phone:"));
```

```
        add(phone);
        add(new JLabel("DOB (dd-mm-yyyy):"));
        add(dob);
        add(registerBtn);
        add(backBtn);

        registerBtn.addActionListener(e -> {
            String user = username.getText();
            String pass = new String(password.getPassword());
            String ad = adhar.getText();
            String ph = phone.getText();
            String date = dob.getText();

            if (user.isEmpty() || pass.isEmpty() || ad.isEmpty() || ph.isEmpty() || date.isEmpty())
{
                JOptionPane.showMessageDialog(this, "Please fill all fields.");
                return;
            }

            if (ad.length() != 12 || !ad.matches("\\d+")) {
                JOptionPane.showMessageDialog(this, "Invalid Aadhar number.");
                return;
            }

            if (ph.length() < 10 || !ph.matches("\\d+")) {
                JOptionPane.showMessageDialog(this, "Invalid phone number.");
                return;
            }

            try {
```

```
            PreparedStatement ps = conn.prepareStatement("INSERT INTO users (username,
password, adhar, phone, dob) VALUES (?, ?, ?, ?, ?)");
            ps.setString(1, user);
            ps.setString(2, hash(pass));
            ps.setString(3, ad);
            ps.setString(4, ph);
            ps.setString(5, date);
            ps.executeUpdate();
            JOptionPane.showMessageDialog(this, "Registered successfully!");
            showHomePage();
        } catch (Exception ex) {
            JOptionPane.showMessageDialog(this, "User already exists.");
        }
    });

    backBtn.addActionListener(e -> showHomePage());
    revalidate();
    repaint();
  }

  void showResetPasswordPage() {
    getContentPane().removeAll();
    setTitle("Reset Password");
    setSize(400, 250);
    setLayout(new GridLayout(5, 2));

    JTextField username = new JTextField();
    JTextField dob = new JTextField();
    JPasswordField newPass = new JPasswordField();
    JButton resetBtn = new JButton("Reset");
```

```
        JButton backBtn = new JButton("Back");

        add(new JLabel("Username:"));
        add(username);
        add(new JLabel("DOB (dd-mm-yyyy):"));
        add(dob);
        add(new JLabel("New Password:"));
        add(newPass);
        add(resetBtn);
        add(backBtn);

        resetBtn.addActionListener(e -> {
            String user = username.getText();
            String date = dob.getText();
            String newPassword = new String(newPass.getPassword());

            try {
                PreparedStatement ps = conn.prepareStatement("SELECT * FROM users WHERE
username=? AND dob=?");
                ps.setString(1, user);
                ps.setString(2, date);
                ResultSet rs = ps.executeQuery();
                if (rs.next()) {
                    PreparedStatement ps2 = conn.prepareStatement("UPDATE users SET
password=? WHERE username=?");
                    ps2.setString(1, hash(newPassword));
                    ps2.setString(2, user);
                    ps2.executeUpdate();
                    JOptionPane.showMessageDialog(this, "Password reset successful!");
                    showHomePage();
```

```
            } else {
                JOptionPane.showMessageDialog(this, "Invalid details.");
            }
        } catch (Exception ex) {
            ex.printStackTrace();
        }
    });

    backBtn.addActionListener(e -> showHomePage());
    revalidate();
    repaint();
}

void showAdminPanel() {
    getContentPane().removeAll();
    setTitle("Admin Panel");
    setSize(400, 300);
    setLayout(new GridLayout(6, 1));

    JButton startBtn = new JButton("Start Voting");
    JButton stopBtn = new JButton("Stop Voting");
    JButton resultBtn = new JButton("Declare Result");
    JButton viewResult = new JButton("View Result");
    JButton resetBtn = new JButton("Reset All Data");
    JButton backBtn = new JButton("Back");

    add(startBtn);
    add(stopBtn);
    add(resultBtn);
```

```
        add(viewResult);
        add(resetBtn);
        add(backBtn);

        startBtn.addActionListener(e -> updateStatus(1, 0));
        stopBtn.addActionListener(e -> updateStatus(0, 0));
        resultBtn.addActionListener(e -> updateStatus(0, 1));
        viewResult.addActionListener(e -> showResult());
        resetBtn.addActionListener(e -> resetAllData());
        backBtn.addActionListener(e -> showHomePage());

        revalidate();
        repaint();
    }

    void updateStatus(int voting, int resultDeclared) {
        try {
            PreparedStatement ps = conn.prepareStatement("UPDATE status SET voting=?,
resultDeclared=?");
            ps.setInt(1, voting);
            ps.setInt(2, resultDeclared);
            ps.executeUpdate();
            JOptionPane.showMessageDialog(this, "Status updated");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void resetAllData() {
        try (Statement stmt = conn.createStatement()) {
```

```
            stmt.executeUpdate("DELETE FROM users");
            stmt.executeUpdate("UPDATE votes SET count=0");
            stmt.executeUpdate("UPDATE status SET voting=0, resultDeclared=0");
            JOptionPane.showMessageDialog(this, "All data reset.");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    void showUserPanel(String username) {
        getContentPane().removeAll();
        setTitle("User Panel - " + username);
        setSize(300, 300);
        setLayout(new GridLayout(6, 1));

        JButton voteBJP = new JButton("Vote for BJP");
        JButton voteCongress = new JButton("Vote for Congress");
        JButton voteAAP = new JButton("Vote for AAP");
        JButton voteOthers = new JButton("Vote for Others");
        JButton viewResult = new JButton("View Result");
        JButton logoutBtn = new JButton("Logout");

        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM status");
            if (rs.next()) {
                boolean voting = rs.getInt("voting") == 1;
                boolean resultDeclared = rs.getInt("resultDeclared") == 1;
```

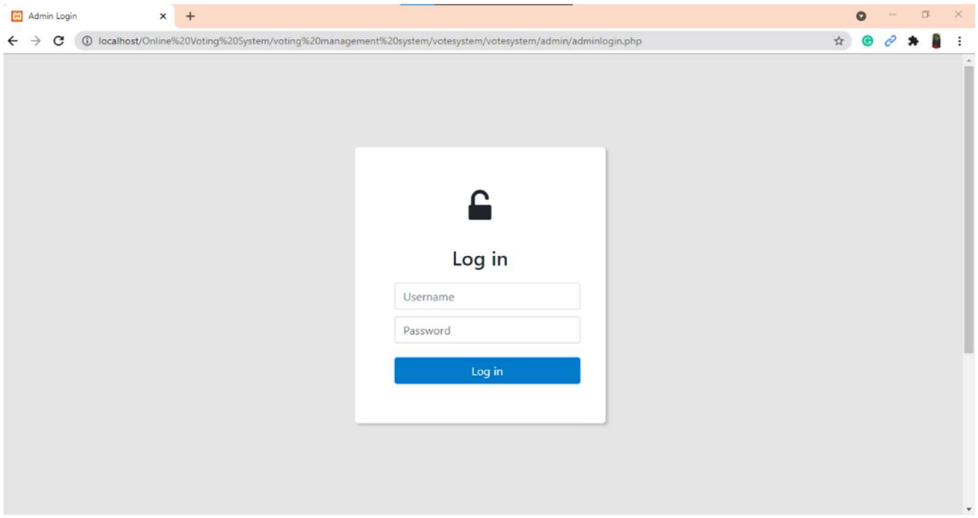
```
        PreparedStatement ps = conn.prepareStatement("SELECT voted FROM users
WHERE username=?");
        ps.setString(1, username);
        ResultSet userRs = ps.executeQuery();
        boolean voted = userRs.next() && userRs.getInt("voted") == 1;

        if (!voting || voted || resultDeclared) {
           voteBJP.setEnabled(false);
           voteCongress.setEnabled(false);
           voteAAP.setEnabled(false);
           voteOthers.setEnabled(false);
        }

        voteBJP.addActionListener(e -> castVote("BJP", username));
        voteCongress.addActionListener(e -> castVote("Congress", username));
        voteAAP.addActionListener(e -> castVote("AAP", username));
        voteOthers.addActionListener(e -> castVote("Others", username));
        viewResult.addActionListener(e -> {
           if (resultDeclared)
              showResult();
           else
              JOptionPane.showMessageDialog(this, "Results not declared yet.");
        });

        add(voteBJP);
        add(voteCongress);
        add(voteAAP);
        add(voteOthers);
        add(viewResult);
        add(logoutBtn);
```

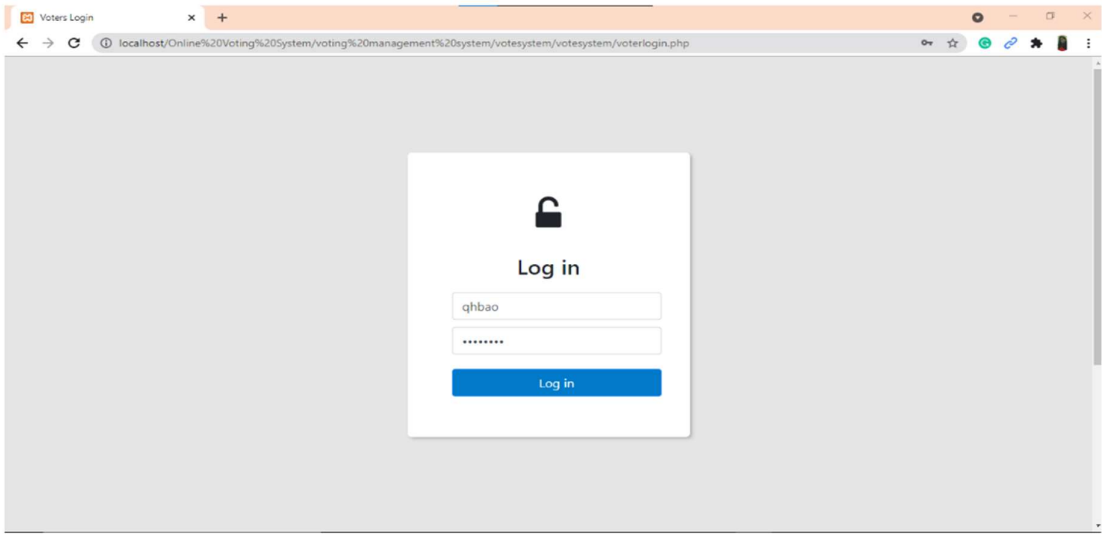
```
            logoutBtn.addActionListener(e -> showHomePage());
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    revalidate();
    repaint();
  }

  void castVote(String party, String username) {
    try {
        PreparedStatement ps1 = conn.prepareStatement("UPDATE votes SET count = count +
1 WHERE party = ?");
        ps1.setString(1, party);
        ps1.executeUpdate();

        PreparedStatement ps2 = conn.prepareStatement("UPDATE users SET voted = 1
WHERE username = ?");
        ps2.setString(1, username);
        ps2.executeUpdate();

        JOptionPane.showMessageDialog(this, "Vote casted successfully for " + party);
        showUserPanel(username);
    } catch (Exception e) {
        e.printStackTrace();
    }
  }

  void showResult() {
```

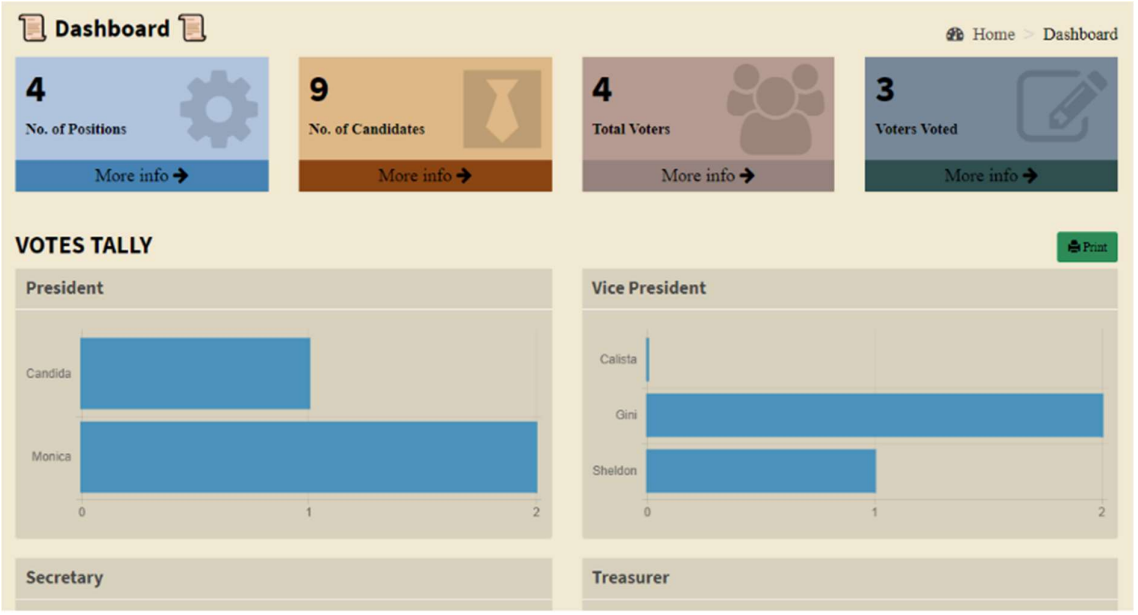
```
        try {
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM votes");

            Map<String, Integer> results = new HashMap<>();
            int maxVotes = -1;
            while (rs.next()) {
                String party = rs.getString("party");
                int count = rs.getInt("count");
                results.put(party, count);
                if (count > maxVotes) maxVotes = count;
            }

            List<String> winners = new ArrayList<>();
            for (Map.Entry<String, Integer> entry : results.entrySet()) {
                if (entry.getValue() == maxVotes) {
                    winners.add(entry.getKey());
                }
            }

            StringBuilder res = new StringBuilder("Results:\n");
            for (Map.Entry<String, Integer> entry : results.entrySet()) {
                res.append(entry.getKey()).append(": ").append(entry.getValue()).append("
votes\n");
            }

            res.append("\nWinner: ").append(winners.size() > 1 ? "Draw between " +
String.join(", ", winners) : winners.get(0));
            JOptionPane.showMessageDialog(this, res.toString());
        } catch (Exception e) {
```

```
            e.printStackTrace();
        }
    }

    String hash(String input) throws Exception {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] hashed = md.digest(input.getBytes("UTF-8"));
        StringBuilder sb = new StringBuilder();
        for (byte b : hashed) sb.append(String.format("%02x", b));
        return sb.toString();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(VotingApp::new);
    }
}
```

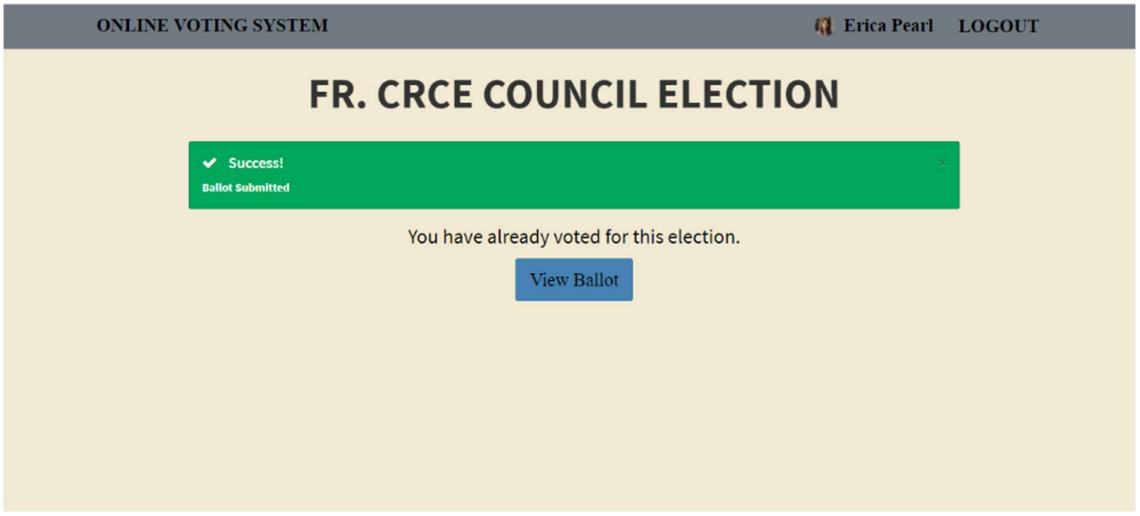
Output:




Voters Login
localhost/Online%20Voting%20System/voting%20management%20system/votesystem/votesystem/voterlogin.php
Log in
qhbao
Log in


Dashboard
Home > Dashboard
4
No. of Positions
More info
9
No. of Candidates
More info
4
Total Voters
More info
3
Voters Voted
More info
VOTES TALLY
Print
President
Candida
Monica
0
1
2
Vice President
Calista
Gini
Sheldon
0
1
2
Secretary
Treasurer


Positions
Home > Dashboard
+New
Show 10 entries
Search:
Description
Maximum Vote
Tools
President 1 Edit Delete
Vice President 1 Edit Delete
Secretary 1 Edit Delete
Treasurer 1 Edit Delete
Showing 1 to 4 of 4 entries
Previous 1 Next


ONLINE VOTING SYSTEM
Erica Pearl
LOGOUT
FR. CRCE COUNCIL ELECTION
Success!
Ballot Submitted
You have already voted for this election.
View Ballot