19CS652 – OBJECT ORIENTED ANALYSIS AND DESIGN LABORATORY

APPENDIX

E-GOVERNANCE PORTAL FOR PUBLIC SERVICES SYSTEM

ROLLNO: 22BCS100

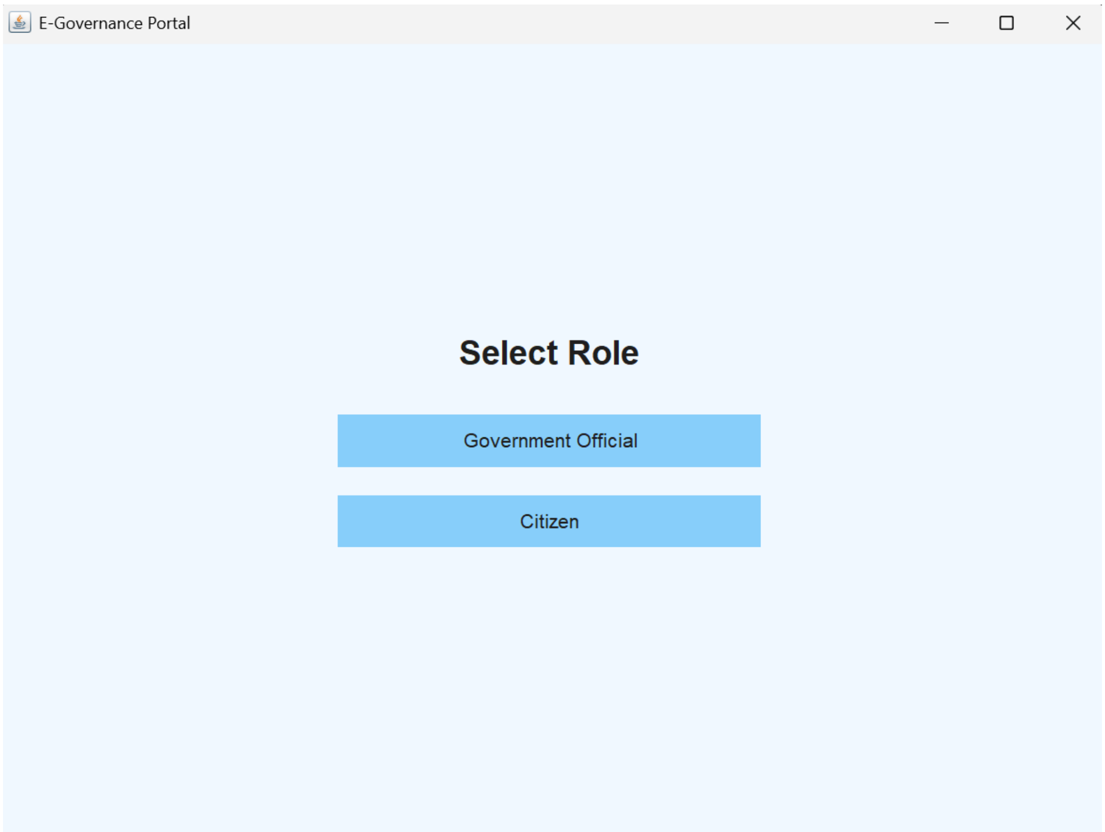
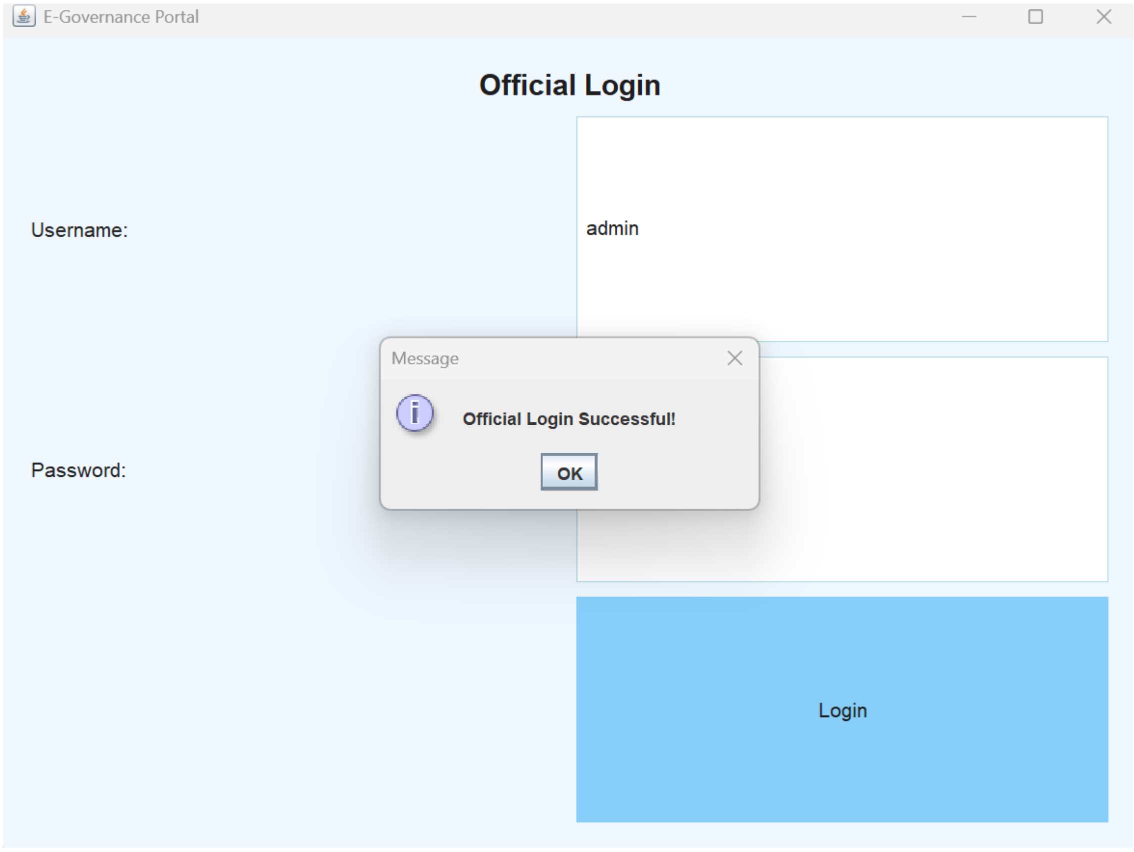


Figure 1: Role Selection Panel



Username: admin

Password:

Official Login Successful!

Figure 2: Government Official Login Form

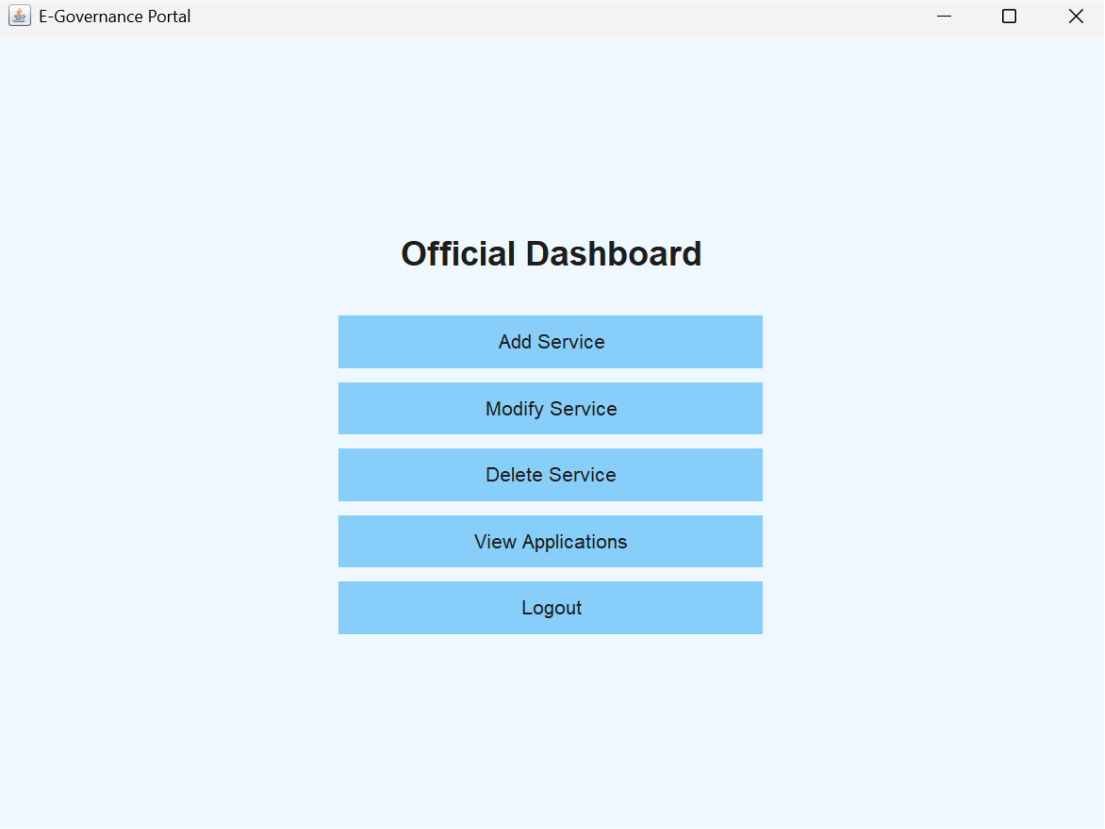
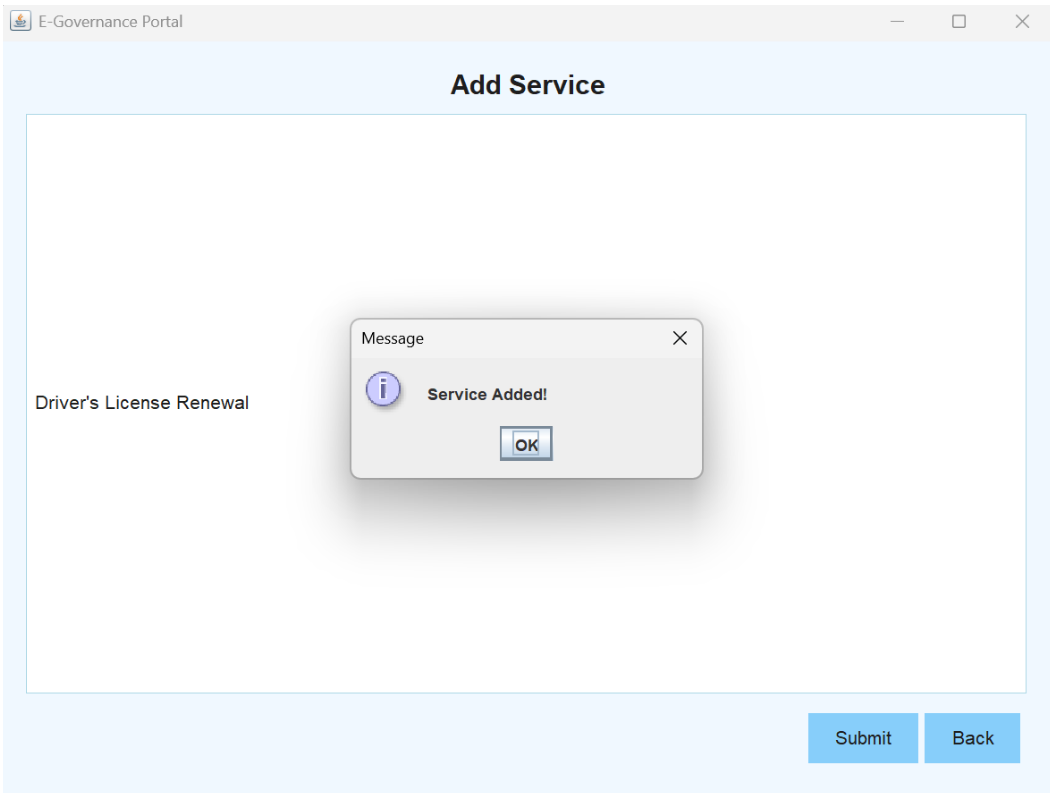


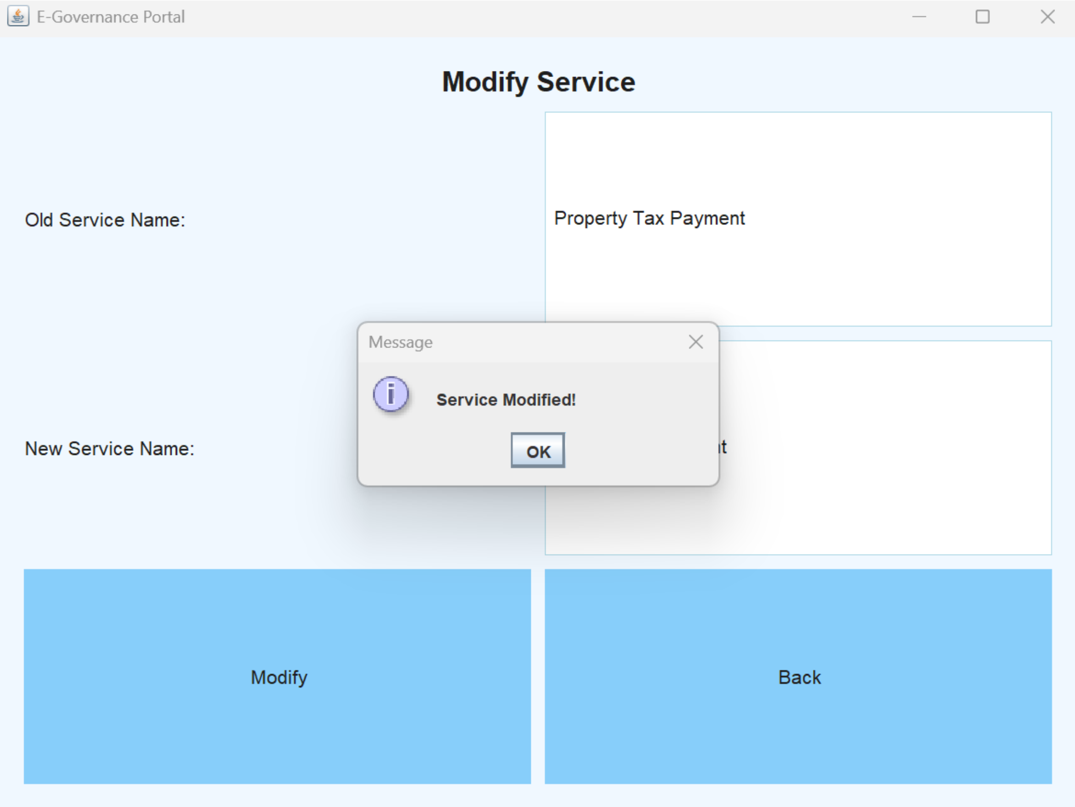
Figure 3: Government Official Dashboard



Driver's License Renewal

Service Added!

Figure 4: Service Addition Form



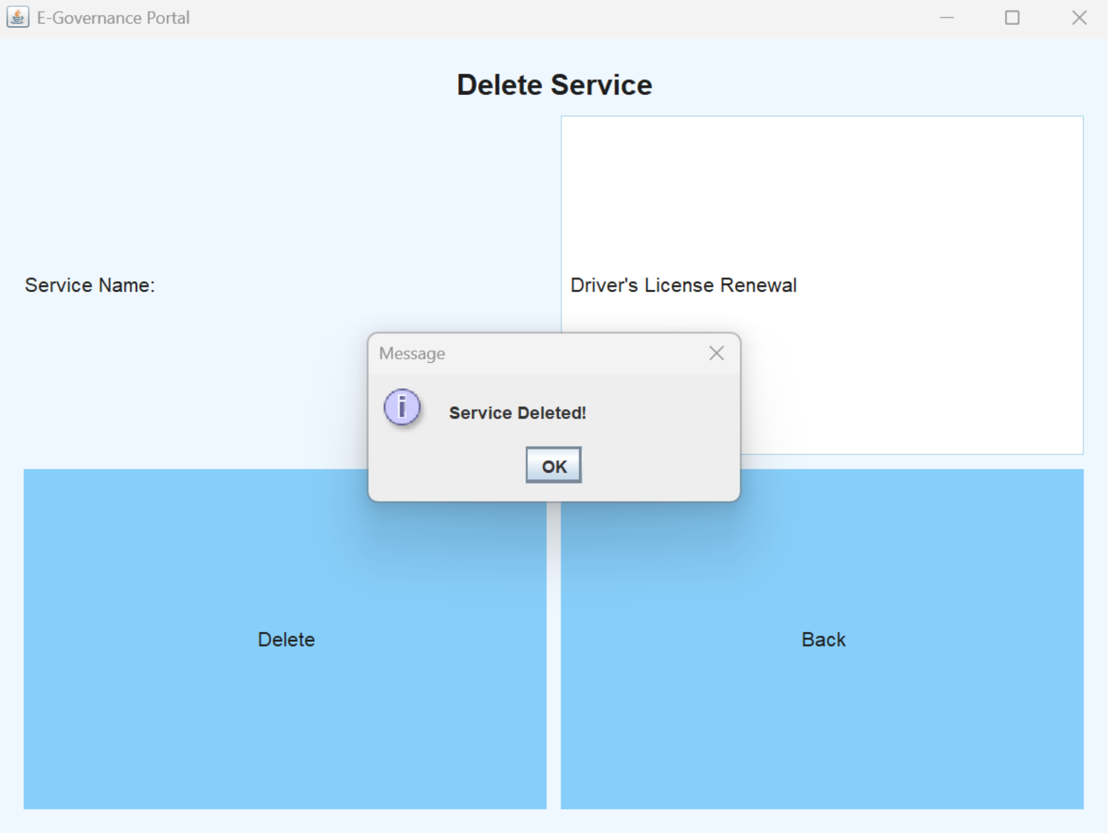
Modify Service

Old Service Name: Property Tax Payment

New Service Name: t

Service Modified!

Figure 5: Service Modification Form

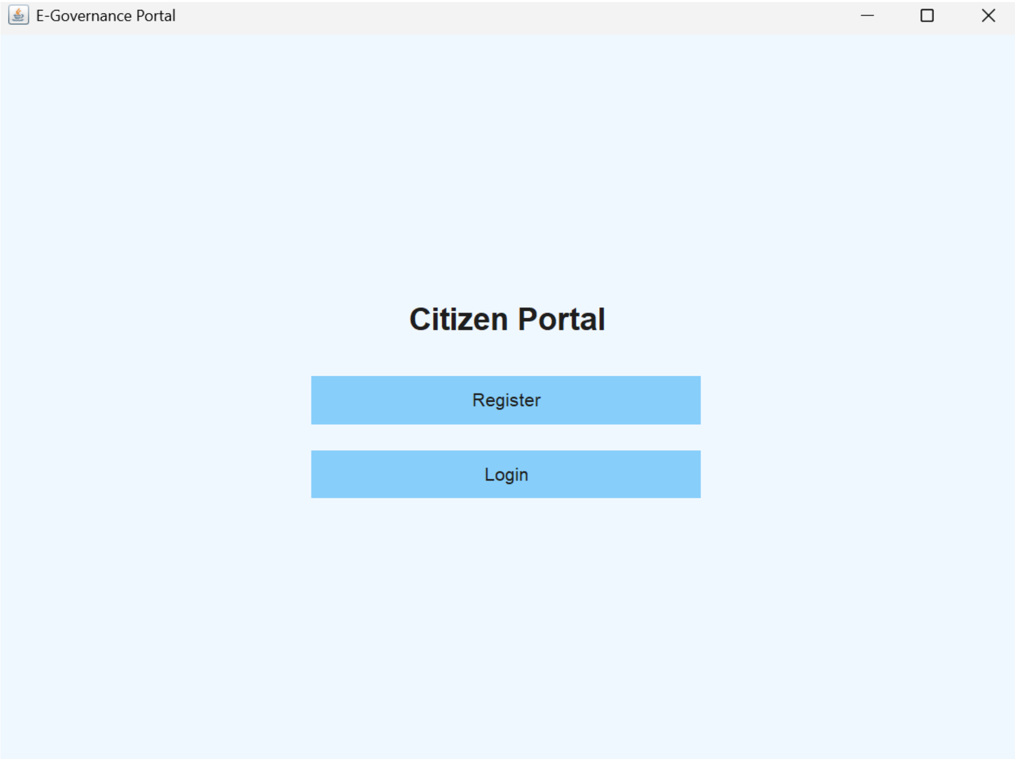


Delete Service

Service Name: Driver's License Renewal

Service Deleted!

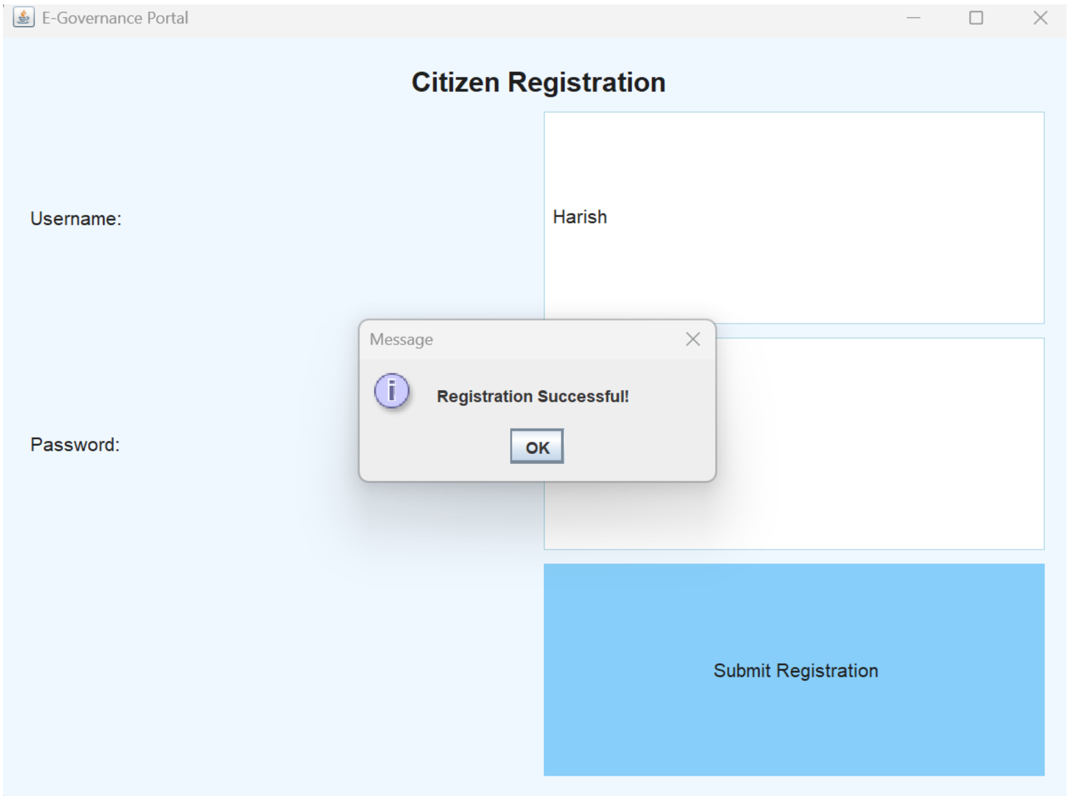
Figure 6: Service Deletion Form



E-Governance Portal

Citizen Portal

Figure 7: Citizen Register/Login Panel



E-Governance Portal

Citizen Registration

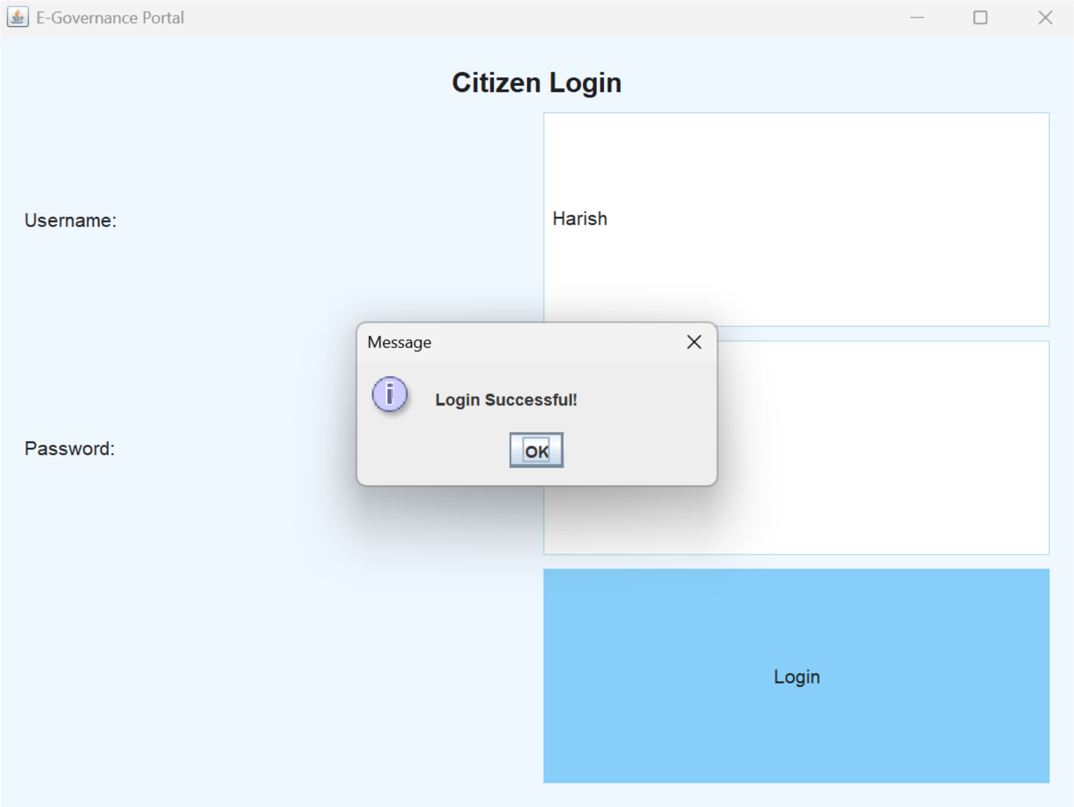
Username: Harish

Password:

Message

Registration Successful!

Figure 8: Citizen Registration Form



E-Governance Portal

Citizen Login

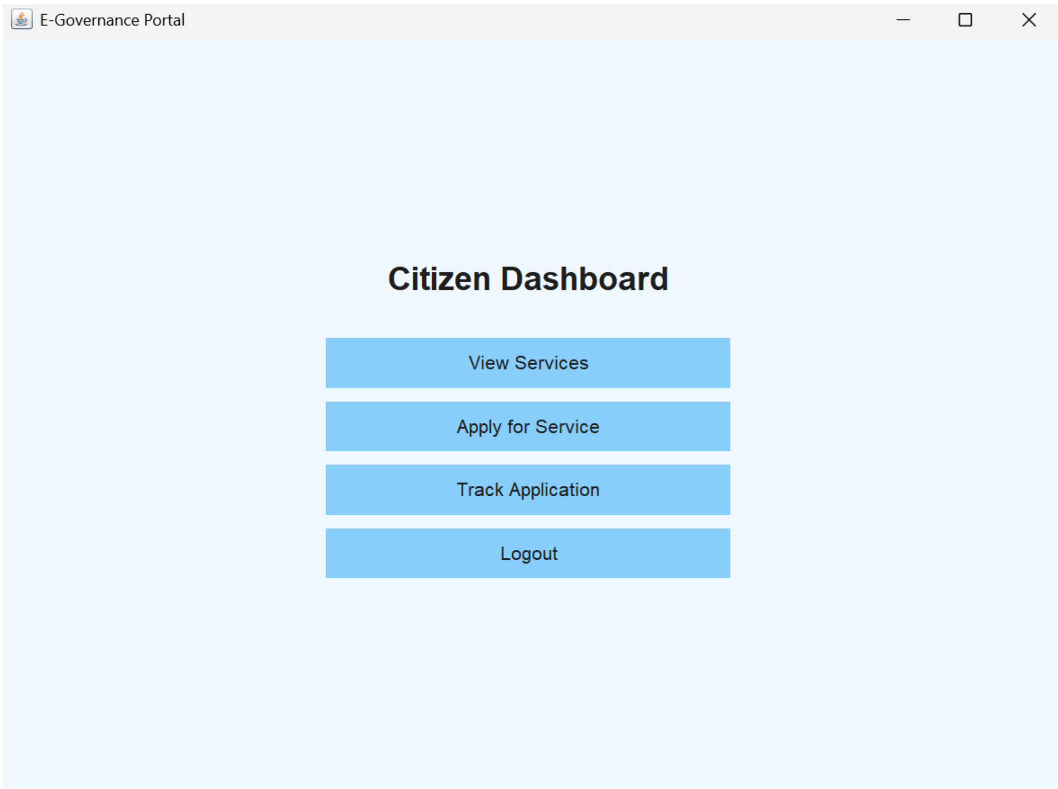
Username: Harish

Password:

Message

Login Successful!

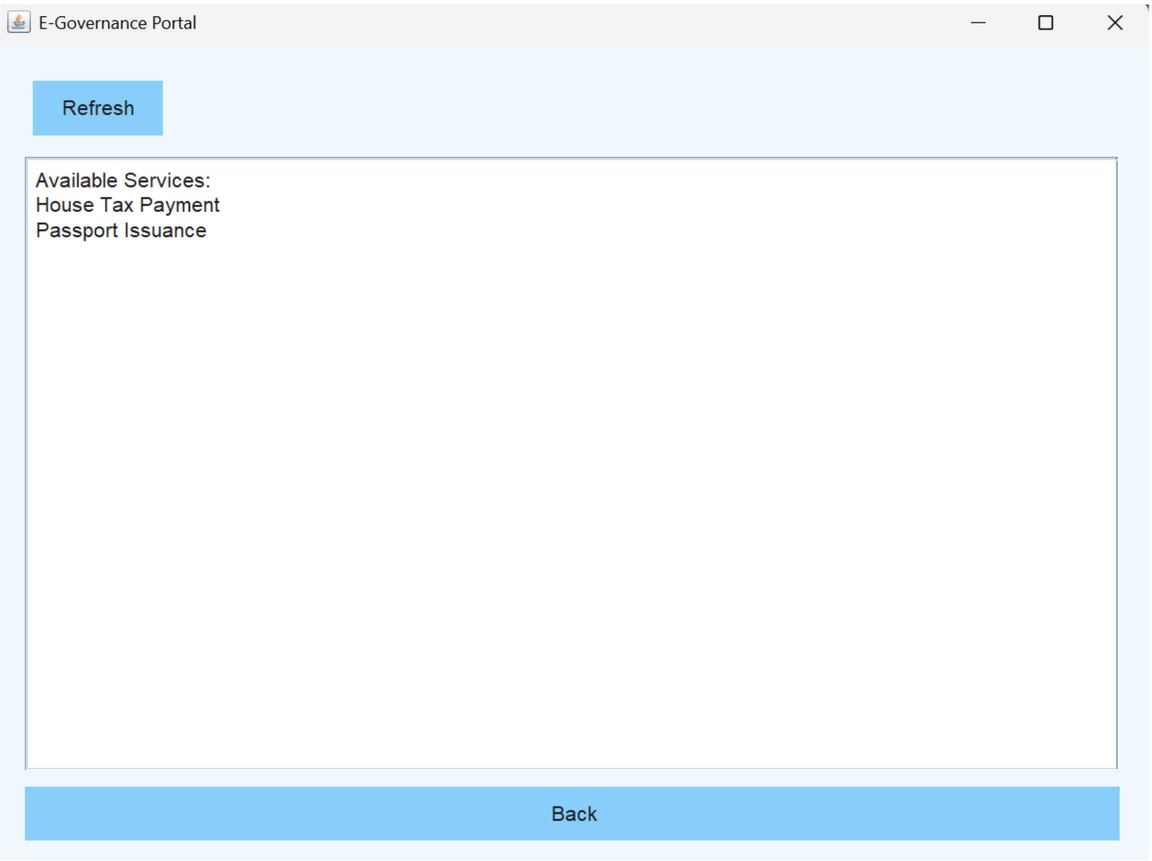
Figure 9: Citizen Login Form



E-Governance Portal

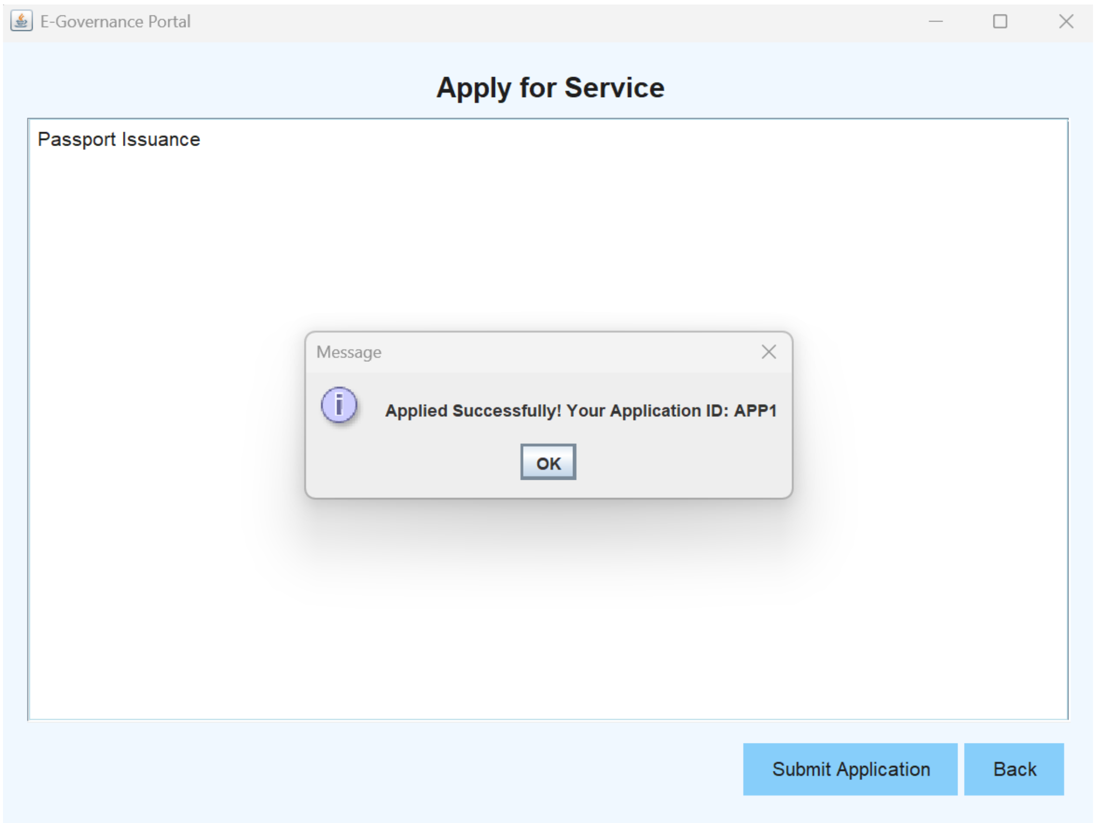
Citizen Dashboard

Figure 10: Citizen Dashboard



Available Services:
House Tax Payment
Passport Issuance

Figure 11: Services View Panel

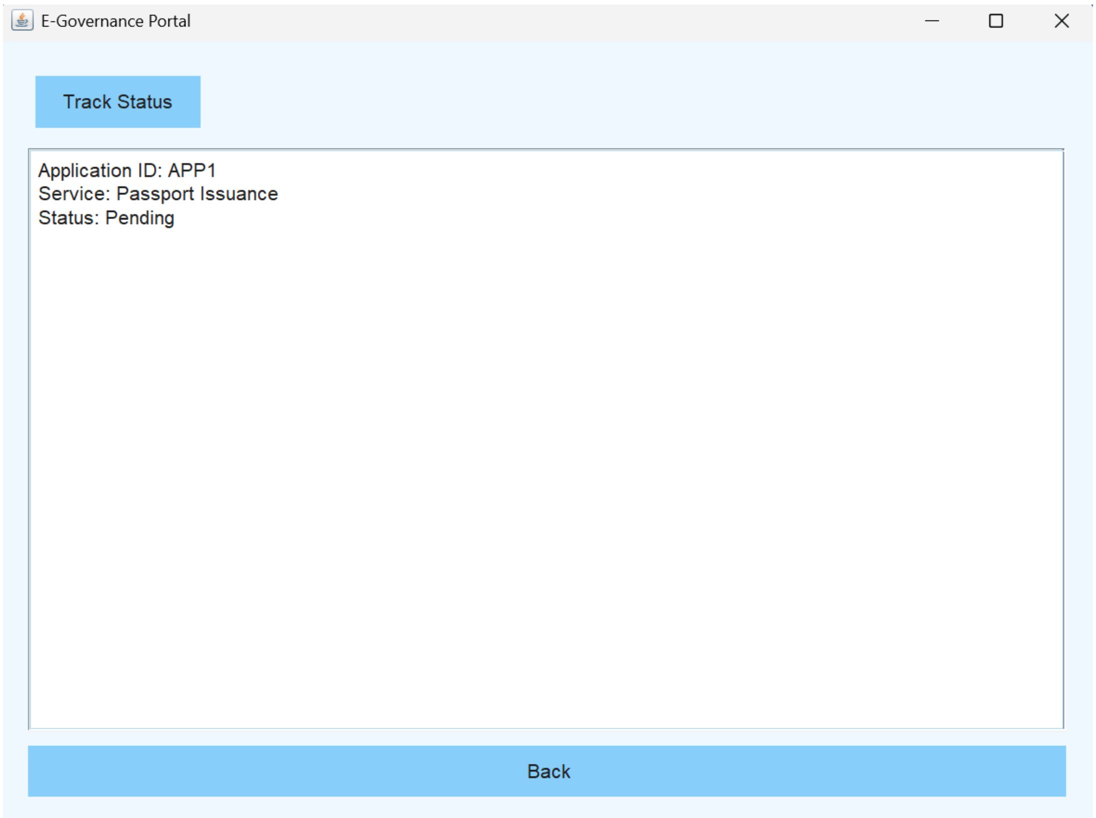


Apply for Service

Passport Issuance

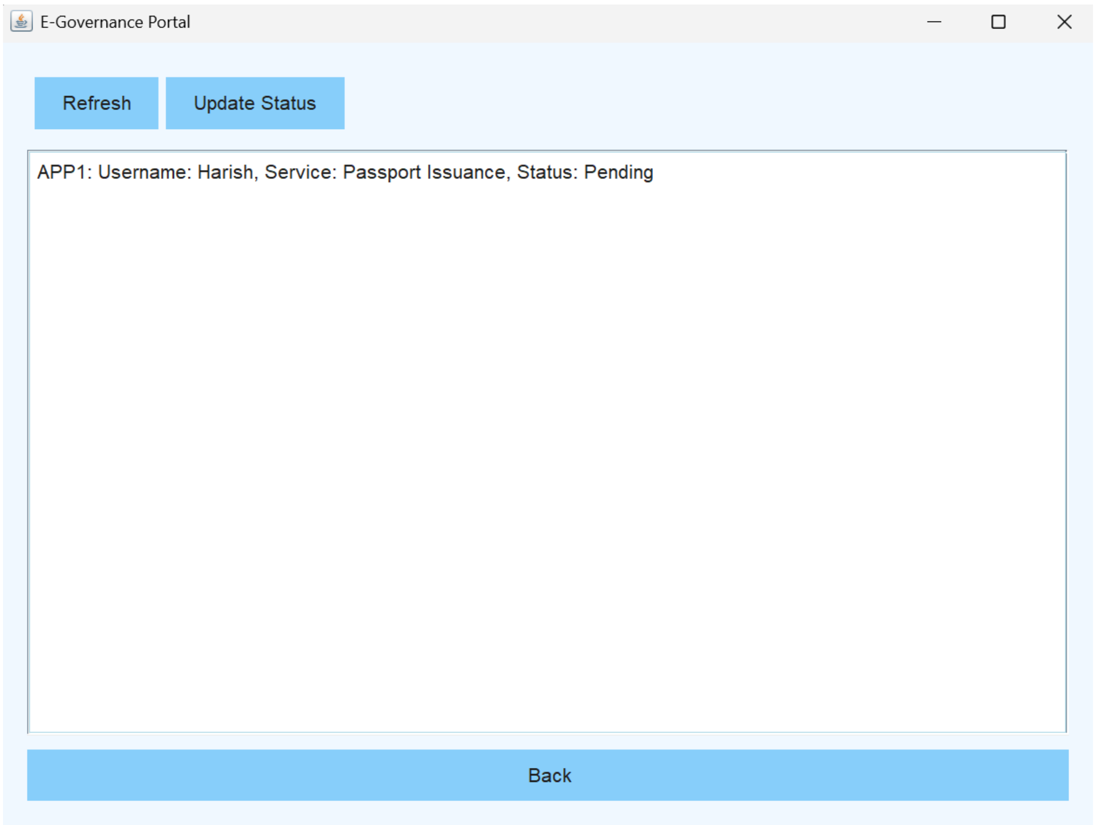
Applied Successfully! Your Application ID: APP1

Figure 12: Service Application Form



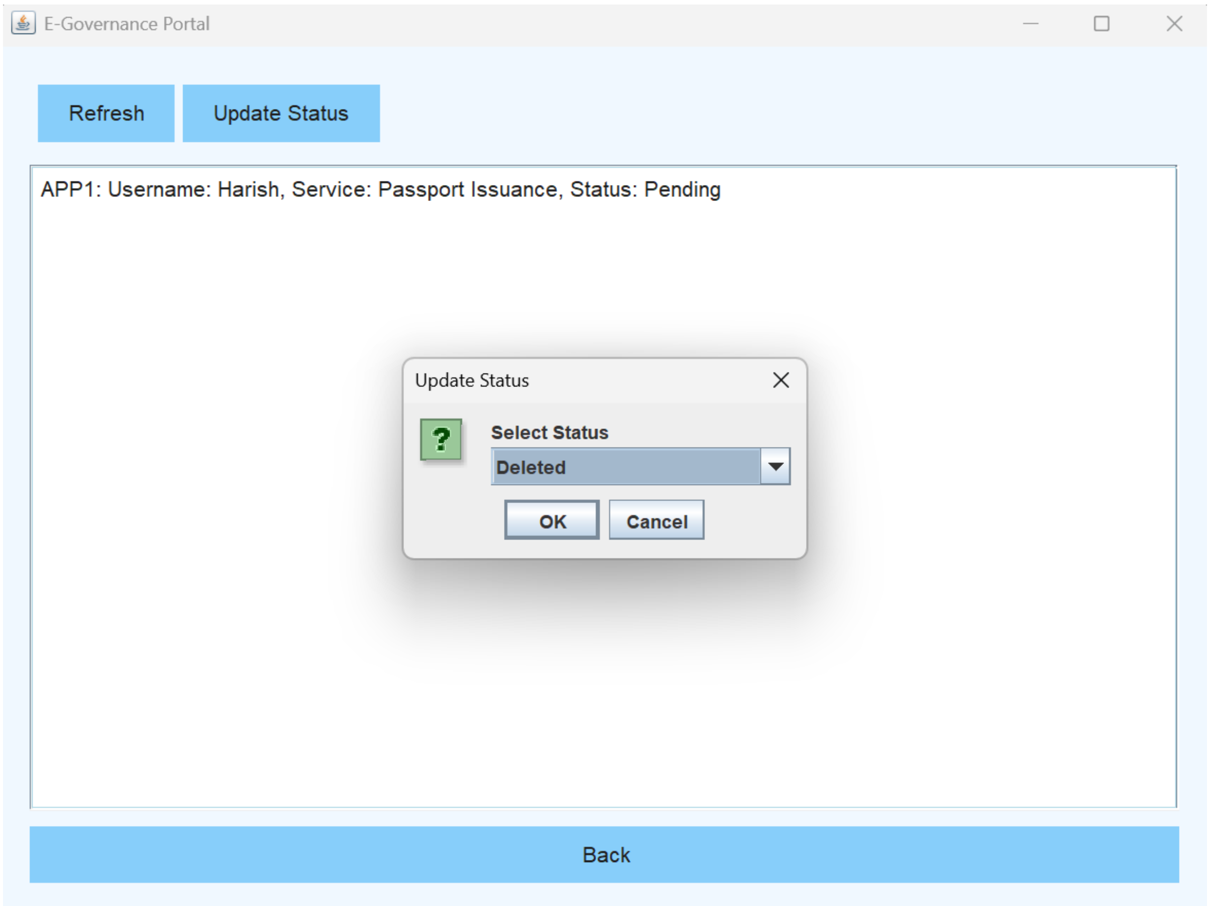
Application ID: APP1
Service: Passport Issuance
Status: Pending

Figure 13: Track Service Status Panel



APP1: Username: Harish, Service: Passport Issuance, Status: Pending

Figure 14: Application Details Panel



APP1: Username: Harish, Service: Passport Issuance, Status: Pending

Update Status

Select Status: Deleted

Figure 15: Application Status Updation Panel

CODE:

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
public class EGovernancePortal extends JFrame implements ActionListener {
   CardLayout card;
   JPanel rolePanel, officialLoginPanel, citizenRolePanel, registerPanel, loginPanel, officialDashboardPanel, citizenDashboardPanel;
   JPanel addServicePanel, modifyServicePanel, deleteServicePanel, viewApplicationsPanel, citizenServicesPanel, citizenApplyPanel, citizenTrackPanel;
   JButton govOfficialBtn, citizenBtn, officialLoginBtn, citizenRegisterBtn, citizenLoginBtn;
   JButton addServiceBtn, modifyServiceBtn, deleteServiceBtn, viewApplicationsBtn, logoutOfficialBtn;
   JButton viewServicesBtn, applyServiceBtn, trackServiceBtn, logoutCitizenBtn;
```

```
    JTextField officialUsernameField, citizenUsernameField, citizenRegUsernameField,
serviceNameField, modifyOldServiceField, modifyNewServiceField, deleteServiceField;
    JPasswordField officialPasswordField, citizenPasswordField, citizenRegPasswordField;
    JTextArea applicationListArea, availableServicesArea, applyDocField, trackStatusField;
    private String currentCitizen;
    private static final String DB_URL =
"jdbc:mysql://localhost:3306/egovernance?useSSL=false";
    private static final String DB_USER = "root"; // Replace with your MySQL username
    private static final String DB_PASSWORD = "harish0511"; // Replace with your MySQL
password
    private Connection conn;
    public EGovernancePortal() {
        setTitle("E-Governance Portal");
        setSize(800, 600);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        getContentPane().setBackground(PRIMARY_COLOR);
        card = new CardLayout();
        setLayout(card);
        try {
            conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);
        } catch (SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this, "Database connection failed!");
            System.exit(1);
        }
        rolePanel = createRolePanel();
        officialLoginPanel = createOfficialLoginPanel();
        citizenRolePanel = createCitizenRolePanel();
        registerPanel = createCitizenRegisterPanel();
        loginPanel = createCitizenLoginPanel();
        officialDashboardPanel = createOfficialDashboardPanel();
        citizenDashboardPanel = createCitizenDashboardPanel();
        addServicePanel = createAddServicePanel();
        modifyServicePanel = createModifyServicePanel();
        deleteServicePanel = createDeleteServicePanel();
        viewApplicationsPanel = createViewApplicationsPanel();
        citizenServicesPanel = createCitizenServicesPanel();
        citizenApplyPanel = createCitizenApplyPanel();
        citizenTrackPanel = createCitizenTrackPanel();
        add(rolePanel, "role");
        add(officialLoginPanel, "officialLogin");
        add(citizenRolePanel, "citizenRole");
        add(registerPanel, "register");
        add(loginPanel, "login");
        add(officialDashboardPanel, "officialDashboard");
        add(citizenDashboardPanel, "citizenDashboard");
        add(addServicePanel, "addService");
        add(modifyServicePanel, "modifyService");
        add(deleteServicePanel, "deleteService");
        add(viewApplicationsPanel, "viewApplications");
        add(citizenServicesPanel, "citizenServices");
        add(citizenApplyPanel, "citizenApply");
        add(citizenTrackPanel, "citizenTrack");
        card.show(getContentPane(), "role");
       setVisible(true);
    }
    private JPanel createOfficialLoginPanel() {
        JPanel panel = createFormPanel("Official Login");
        JPanel form = new JPanel(new GridLayout(3, 2, 10, 10));
        form.setBackground(PRIMARY_COLOR);
        officialUsernameField = createStyledTextField();
        officialPasswordField = createStyledPasswordField();
        officialLoginBtn = createStyledButton("Login");
        officialLoginBtn.addActionListener(this);
        form.add(createStyledLabel("Username:"));
        form.add(officialUsernameField);
        form.add(createStyledLabel("Password:"));
```

```
        form.add(officialPasswordField);
        form.add(new JLabel(""));
        form.add(officialLoginBtn);
        panel.add(form, BorderLayout.CENTER);
        return panel;
    }
    private JPanel createCitizenRegisterPanel() {
        JPanel panel = createFormPanel("Citizen Registration");
        JPanel form = new JPanel(new GridLayout(3, 2, 10, 10));
        form.setBackground(PRIMARY_COLOR);
        citizenRegUsernameField = createStyledTextField();
        citizenRegPasswordField = createStyledPasswordField();
        JButton registerSubmit = createStyledButton("Submit Registration");
        registerSubmit.addActionListener(e -> {
            String user = citizenRegUsernameField.getText().trim();
            String pass = new String(citizenRegPasswordField.getPassword()).trim();
            if (!user.isEmpty() && !pass.isEmpty()) {
                try {
                    PreparedStatement ps = conn.prepareStatement("INSERT INTO citizens
(username, password) VALUES (?, ?)");
                    ps.setString(1, user);
                    ps.setString(2, pass);
                    ps.executeUpdate();
                    JOptionPane.showMessageDialog(this, "Registration Successful!");
                    card.show(getContentPane(), "citizenRole");
                } catch (SQLException ex) {
                    JOptionPane.showMessageDialog(this, "Registration failed: " +
ex.getMessage());
                }
            } else {
                JOptionPane.showMessageDialog(this, "Fill all fields!");
            }
        });
        form.add(createStyledLabel("Username:"));
        form.add(citizenRegUsernameField);
        form.add(createStyledLabel("Password:"));
        form.add(citizenRegPasswordField);
        form.add(new JLabel(""));
        form.add(registerSubmit);
        panel.add(form, BorderLayout.CENTER);
        return panel;
    }
    private JPanel createCitizenLoginPanel() {
        JPanel panel = createFormPanel("Citizen Login");
        JPanel form = new JPanel(new GridLayout(3, 2, 10, 10));
        form.setBackground(PRIMARY_COLOR);

        citizenUsernameField = createStyledTextField();
        citizenPasswordField = createStyledPasswordField();
        JButton loginSubmit = createStyledButton("Login");
           loginSubmit.addActionListener(e -> {
            String user = citizenUsernameField.getText().trim();
            String pass = new String(citizenPasswordField.getPassword()).trim();
            if (user.isEmpty() || pass.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Please enter both username and
password!");
                return;
            }
                try {
                if (conn == null || conn.isClosed()) {
                    conn = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD);
                    System.out.println("Reconnected to database during login");
                }
                    System.out.println("Attempting login for user: " + user);
                PreparedStatement ps = conn.prepareStatement("SELECT password FROM
citizens WHERE username = ?");
                ps.setString(1, user);
                ResultSet rs = ps.executeQuery();
                if (rs.next() && rs.getString("password").equals(pass)) {
                    currentCitizen = user; // Set the current user
                    System.out.println("Login successful for user: " + user);
                    JOptionPane.showMessageDialog(this, "Login Successful!");
                    citizenUsernameField.setText("");
                    citizenPasswordField.setText("");
                    card.show(getContentPane(), "citizenDashboard");
                } else {
                    JOptionPane.showMessageDialog(this, "Invalid credentials!");
                    System.out.println("Invalid credentials for user: " + user);
                }
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(this, "Login failed: " + ex.getMessage());
                System.out.println("SQLException during login: " + ex.getMessage());
            }
        });
        form.add(createStyledLabel("Username:"));
        form.add(citizenUsernameField);
        form.add(createStyledLabel("Password:"));
        form.add(citizenPasswordField);
        form.add(new JLabel(""));
        form.add(loginSubmit);
        panel.add(form, BorderLayout.CENTER);
        return panel;
    }
    private JPanel createOfficialDashboardPanel() {
        JPanel panel = new JPanel();
        panel.setBackground(PRIMARY_COLOR);
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.setBorder(BorderFactory.createEmptyBorder(50, 50, 50, 50));
        JLabel title = new JLabel("Official Dashboard");
        title.setFont(new Font("Arial", Font.BOLD, 24));
        title.setForeground(TEXT_COLOR);
```

```
        title.setAlignmentX(Component.CENTER_ALIGNMENT);
        addServiceBtn = createStyledButton("Add Service");
        modifyServiceBtn = createStyledButton("Modify Service");
        deleteServiceBtn = createStyledButton("Delete Service");
        viewApplicationsBtn = createStyledButton("View Applications");
        logoutOfficialBtn = createStyledButton("Logout");
        addServiceBtn.addActionListener(this);
        modifyServiceBtn.addActionListener(this);
        deleteServiceBtn.addActionListener(this);
        viewApplicationsBtn.addActionListener(this);
        logoutOfficialBtn.addActionListener(this);
        panel.add(Box.createVerticalGlue());
        panel.add(title);
        panel.add(Box.createRigidArea(new Dimension(0, 30)));
        panel.add(addServiceBtn);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(modifyServiceBtn);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(deleteServiceBtn);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(viewApplicationsBtn);
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(logoutOfficialBtn);
        panel.add(Box.createVerticalGlue());
        return panel;
    }

    private JPanel createModifyServicePanel() {
        JPanel panel = createFormPanel("Modify Service");
        JPanel form = new JPanel(new GridLayout(3, 2, 10, 10));
        form.setBackground(PRIMARY_COLOR);
        modifyOldServiceField = createStyledTextField();
        modifyNewServiceField = createStyledTextField();
        JButton modify = createStyledButton("Modify");
        JButton back = createStyledButton("Back");
        modify.addActionListener(e -> {
            String oldService = modifyOldServiceField.getText().trim();
            String newService = modifyNewServiceField.getText().trim();
            if (oldService.isEmpty() || newService.isEmpty()) {
                JOptionPane.showMessageDialog(this, "Both fields are required!");
                return;
            }
            try {
                if (conn == null || conn.isClosed()) {
                    conn = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD);
                    System.out.println("Reconnected to database for modification");
                }
                System.out.println("Checking service: " + oldService);
                PreparedStatement checkPs = conn.prepareStatement(
                    "SELECT service_name FROM services WHERE LOWER(service_name) =
LOWER(?)"
                );
                checkPs.setString(1, oldService);
                ResultSet rs = checkPs.executeQuery();
                if (rs.next()) {
                    String exactOldService = rs.getString("service_name");
                    System.out.println("Service found: " + exactOldService);
                        // Update applications table first to maintain foreign key integrity
                    PreparedStatement updateAppsPs = conn.prepareStatement(
                        "UPDATE applications SET service_name = ? WHERE service_name = ?"
                    );
                    updateAppsPs.setString(1, newService);
                    updateAppsPs.setString(2, exactOldService);
                    int appsUpdated = updateAppsPs.executeUpdate();
                    System.out.println("Updated " + appsUpdated + " application records");
                        // Update services table
                    PreparedStatement updatePs = conn.prepareStatement(
                        "UPDATE services SET service_name = ? WHERE service_name = ?"
                    );
                    updatePs.setString(1, newService);
                    updatePs.setString(2, exactOldService);
                    int rowsUpdated = updatePs.executeUpdate();
                    if (rowsUpdated > 0) {
                        JOptionPane.showMessageDialog(this, "Service Modified!");
                        modifyOldServiceField.setText("");
                        modifyNewServiceField.setText("");
                    } else {
                        JOptionPane.showMessageDialog(this, "Update failed unexpectedly!");
                    }
                } else {
                    JOptionPane.showMessageDialog(this, "Service not found!");
                }
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(this, "Error modifying service: " +
ex.getMessage());
                System.out.println("SQLException: " + ex.getMessage());
            }
        });
        back.addActionListener(e -> card.show(getContentPane(), "officialDashboard"));
        form.add(createStyledLabel("Old Service Name:"));
        form.add(modifyOldServiceField);
        form.add(createStyledLabel("New Service Name:"));
        form.add(modifyNewServiceField);
        form.add(modify);
        form.add(back);
        panel.add(form, BorderLayout.CENTER);
        return panel;
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new EGovernancePortal());
    }
```

Result:

Thus, the project demo of E-Governance portal for Public Services System using java has been implemented successfully.