import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.event.\*;

import java.sql.\*;

import static javax.swing.WindowConstants.EXIT\_ON\_CLOSE;

public class Registration extends JFrame {

private JTextField idField, nameField, addressField, contactField;

private JRadioButton maleButton, femaleButton;

private JComboBox<String> yearBox, monthBox, dayBox;

private JCheckBox termsBox;

private JButton registerButton, exitButton;

private ButtonGroup genderGroup;

private JTable userTable;

private DefaultTableModel tableModel;

public Registration() {

setTitle("Registration Form");

setLayout(null);

setSize(950, 550);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

// ID

JLabel idLabel = new JLabel("ID:");

idLabel.setBounds(30, 20, 100, 25);

add(idLabel);

idField = new JTextField();

idField.setBounds(130, 20, 150, 25);

add(idField);

// Name

JLabel nameLabel = new JLabel("Name:");

nameLabel.setBounds(30, 60, 100, 25);

add(nameLabel);

nameField = new JTextField();

nameField.setBounds(130, 60, 150, 25);

add(nameField);

// Gender

JLabel genderLabel = new JLabel("Gender:");

genderLabel.setBounds(30, 100, 100, 25);

add(genderLabel);

maleButton = new JRadioButton("Male");

femaleButton = new JRadioButton("Female");

maleButton.setBounds(130, 100, 60, 25);

femaleButton.setBounds(200, 100, 80, 25);

genderGroup = new ButtonGroup();

genderGroup.add(maleButton);

genderGroup.add(femaleButton);

add(maleButton);

add(femaleButton);

// DOB

JLabel dobLabel = new JLabel("DOB:");

dobLabel.setBounds(30, 140, 100, 25);

add(dobLabel);

yearBox = new JComboBox<>();

monthBox = new JComboBox<>();

dayBox = new JComboBox<>();

yearBox.setBounds(130, 140, 70, 25);

monthBox.setBounds(210, 140, 70, 25);

dayBox.setBounds(290, 140, 60, 25);

add(yearBox);

add(monthBox);

add(dayBox);

for (int y = 1990; y <= 2030; y++) yearBox.addItem(String.valueOf(y));

for (int m = 1; m <= 12; m++) monthBox.addItem(String.format("%02d", m));

for (int d = 1; d <= 31; d++) dayBox.addItem(String.format("%02d", d));

// Address

JLabel addressLabel = new JLabel("Address:");

addressLabel.setBounds(30, 180, 100, 25);

add(addressLabel);

addressField = new JTextField();

addressField.setBounds(130, 180, 150, 25);

add(addressField);

// Contact

JLabel contactLabel = new JLabel("Contact:");

contactLabel.setBounds(30, 220, 100, 25);

add(contactLabel);

contactField = new JTextField();

contactField.setBounds(130, 220, 150, 25);

add(contactField);

// Terms

termsBox = new JCheckBox("I agree to terms");

termsBox.setBounds(30, 260, 200, 25);

add(termsBox);

// Buttons

registerButton = new JButton("Register");

registerButton.setBounds(30, 300, 100, 30);

add(registerButton);

exitButton = new JButton("Exit");

exitButton.setBounds(150, 300, 100, 30);

add(exitButton);

// Table

tableModel = new DefaultTableModel();

userTable = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(userTable);

scrollPane.setBounds(350, 20, 560, 450);

add(scrollPane);

tableModel.addColumn("ID");

tableModel.addColumn("Name");

tableModel.addColumn("Gender");

tableModel.addColumn("DOB");

tableModel.addColumn("Address");

tableModel.addColumn("Contact");

// Load users into table

loadUsers();

// Actions

registerButton.addActionListener(evt -> registerUser());

exitButton.addActionListener(e -> System.exit(0));

}

private void registerUser() {

String id = idField.getText().trim();

String name = nameField.getText().trim();

String address = addressField.getText().trim();

String contact = contactField.getText().trim();

String gender = maleButton.isSelected() ? "Male" : (femaleButton.isSelected() ? "Female" : "");

String dob = yearBox.getSelectedItem() + "-" + monthBox.getSelectedItem() + "-" + dayBox.getSelectedItem();

if (id.isEmpty() || name.isEmpty() || address.isEmpty() || contact.isEmpty() || gender.isEmpty()) {

JOptionPane.showMessageDialog(this, "Please fill in all fields including ID.");

return;

}

if (!termsBox.isSelected()) {

JOptionPane.showMessageDialog(this, "Please agree to the terms.");

return;

}

try {

Connection conn = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/registration\_database",

"root",

""

);

String query = "INSERT INTO users (id, name, gender, dob, address, contact) VALUES (?, ?, ?, ?, ?, ?)";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, Integer.parseInt(id));

pstmt.setString(2, name);

pstmt.setString(3, gender);

pstmt.setString(4, dob);

pstmt.setString(5, address);

pstmt.setString(6, contact);

int rows = pstmt.executeUpdate();

if (rows > 0) {

tableModel.addRow(new Object[]{id, name, gender, dob, address, contact});

JOptionPane.showMessageDialog(this, "Registration successful!");

clearForm();

} else {

JOptionPane.showMessageDialog(this, "Registration failed.");

}

pstmt.close();

conn.close();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Database Error: " + ex.getMessage());

ex.printStackTrace();

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(this, "ID must be a number.");

}

}

private void loadUsers() {

try {

Connection conn = DriverManager.getConnection(

"jdbc:mysql://localhost:3306/registration\_database",

"root",

""

);

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM users");

while (rs.next()) {

tableModel.addRow(new Object[]{

rs.getInt("id"),

rs.getString("name"),

rs.getString("gender"),

rs.getString("dob"),

rs.getString("address"),

rs.getString("contact")

});

}

rs.close();

stmt.close();

conn.close();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(this, "Failed to load users: " + ex.getMessage());

ex.printStackTrace();

}

}

private void clearForm() {

idField.setText("");

nameField.setText("");

addressField.setText("");

contactField.setText("");

genderGroup.clearSelection();

yearBox.setSelectedIndex(0);

monthBox.setSelectedIndex(0);

dayBox.setSelectedIndex(0);

termsBox.setSelected(false);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> new Registration().setVisible(true));

}

}