import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Random;

public class RegisterForEvent {

public static void main(String[] args) {

JFrame frame = new JFrame("Register For Event");

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setSize(500, 600);

frame.setLayout(null);

JPanel panel = new JPanel();

panel.setBackground(Color.WHITE);

panel.setBounds(20, 20, 450, 530);

panel.setLayout(null);

panel.setBorder(BorderFactory.createLineBorder(Color.BLACK, 2)); // Cool Border

frame.add(panel);

JLabel titleLabel = new JLabel("SEARCH EVENTS");

titleLabel.setFont(new Font("Courier New", Font.BOLD, 18));

titleLabel.setBounds(150, 10, 200, 30);

panel.add(titleLabel);

// Event Name Field

JLabel eventNameLabel = new JLabel("Event Name:");

eventNameLabel.setBounds(30, 50, 100, 25);

panel.add(eventNameLabel);

JTextField eventNameField = new JTextField();

eventNameField.setBounds(140, 50, 200, 25);

panel.add(eventNameField);

// Organization Search Field

JLabel orgLabel = new JLabel("Event Organization:");

orgLabel.setBounds(30, 90, 150, 25);

panel.add(orgLabel);

JTextField orgField = new JTextField();

orgField.setBounds(180, 90, 160, 25);

panel.add(orgField);

// Event Time Range Selection

JLabel timeLabel = new JLabel("Event Time:");

timeLabel.setBounds(30, 130, 100, 25);

panel.add(timeLabel);

String[] timeOptions = {"Morning", "Afternoon", "Evening", "Night"};

JComboBox<String> timeDropdown = new JComboBox<>(timeOptions);

timeDropdown.setBounds(140, 130, 200, 25);

panel.add(timeDropdown);

// Calendar Selection Button

JLabel dateLabel = new JLabel("Select Date:");

dateLabel.setBounds(30, 170, 120, 25);

panel.add(dateLabel);

JButton calendarButton = new JButton("Open Calendar");

calendarButton.setBounds(160, 170, 150, 30);

panel.add(calendarButton);

JTextField dateField = new JTextField();

dateField.setBounds(160, 210, 150, 25);

panel.add(dateField);

// Placeholder for Calendar Function (Manual Entry for Now)

calendarButton.addActionListener(e -> JOptionPane.showMessageDialog(frame, "Calendar Feature Coming Soon!"));

// Search by Map Button

JButton mapsButton = new JButton("Search by Map");

mapsButton.setBounds(140, 250, 180, 30);

panel.add(mapsButton);

mapsButton.addActionListener(e -> JOptionPane.showMessageDialog(frame, "Opening Maps..."));

// Search Button with Glow Effect

JButton searchButton = new JButton("SEARCH");

searchButton.setBounds(180, 330, 100, 30);

panel.add(searchButton);

searchButton.setForeground(Color.WHITE);

searchButton.setBackground(Color.BLACK);

searchButton.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseEntered(java.awt.event.MouseEvent evt) {

searchButton.setBackground(Color.RED);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

searchButton.setBackground(Color.BLACK);

}

});

// Progress Bar for Loading Effect

JProgressBar progressBar = new JProgressBar(0, 100);

progressBar.setBounds(140, 370, 200, 25);

progressBar.setStringPainted(true);

progressBar.setVisible(false);

panel.add(progressBar);

// Random Fun Quotes for Search

String[] funQuotes = {

"Finding the best events for you! 🎉",

"Hold on! Searching for epic moments! 🚀",

"Almost there... good things take time! ⏳",

"Cracking the event database... 🔍",

"Getting the party list ready! 🎶"

};

// Action Listener for Search Button

searchButton.addActionListener(e -> {

progressBar.setVisible(true);

Timer timer = new Timer(50, new ActionListener() {

int progress = 0;

Random rand = new Random();

@Override

public void actionPerformed(ActionEvent evt) {

progress += 10;

progressBar.setValue(progress);

if (progress >= 100) {

((Timer) evt.getSource()).stop();

JOptionPane.showMessageDialog(frame,

funQuotes[rand.nextInt(funQuotes.length)] + "\n\n" + // Fun Quote

"Event: " + eventNameField.getText() +

"\nOrganization: " + orgField.getText() +

"\nTime: " + timeDropdown.getSelectedItem() +

"\nDate: " + dateField.getText()

);

progressBar.setVisible(false);

}

}

});

timer.start();

});

frame.setVisible(true);

}

}