**Ex No: GUI**

**Date:**

**Aim:**

To implement the Java programs with graphical user interfaces (GUIs) that perform specific tasks.

**STEP 1: Quiz Application:**

* The QuizApplication class extends JFrame and represents a simple quiz application.
* Questions are added dynamically using the addQuestion method.
* Radio buttons are used for multiple-choice answers, and a ButtonGroup ensures only one option is selected.
* The total score is calculated upon clicking the "Submit" button.
* The correct answers are displayed at the end using a JOptionPane.

**STEP 2: Shopping Cart:**

* The ShoppingCart class extends JFrame and represents a basic shopping cart GUI.
* Users can input item names and prices, add them to the cart, and calculate the total amount.
* The cart's contents are displayed in a JTextArea, and the total amount is updated accordingly.

**STEP 3: Google Account Creation:**

* The GoogleAccount class extends JFrame and represents a GUI for creating a Google account.
* The form includes fields for first name, last name, email, password, confirm password, mobile number, date of birth, and gender.
* Input validation is implemented in the validateInputs method.
* The program uses JOptionPane to display success or error messages.
* Icons and combo boxes enhance the visual appeal of the form.

1. **Create an interactive Quiz form using radio buttons, check box, text box, text area. Write a method to find the total score of the user. Display the right answer to the user at end.**

**CODE**

import javax.swing.\*;

import java.awt.\*;

public class QuizApplication extends JFrame {

private int score = 0;

public QuizApplication() {

setTitle("QuizApplication");

setSize(800, 500);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL\_SCROLLBAR\_ALWAYS);

setContentPane(scrollPane);

JPanel panel = new JPanel();

panel.setLayout(new BoxLayout(panel, BoxLayout.Y\_AXIS));

scrollPane.setViewportView(panel);

// Question 1

addQuestion(panel, "What does CPU stand for?", new String[]{"Central Processing Unit", "Computer Personal Unit", "Central Personal Unit", "Central Processor Unit"}, 0);

// Question 2

addQuestion(panel, "Which programming language is known as the 'language of the web'?", new String[]{"JavaScript", "Python", "Java", "HTML"}, 0);

// Question 3

addQuestion(panel, "What does HTML stand for?", new String[]{"Hypertext Markup Language", "Hyper Transfer Markup Language", "High Tech Markup Language", "Hyperlink and Text Markup Language"}, 0);

JButton submitButton = new JButton("Submit");

panel.add(submitButton);

JTextArea resultArea = new JTextArea();

panel.add(resultArea);

submitButton.addActionListener(e -> {

int totalQuestions = panel.getComponentCount() - 2;

double totalScore = (double) score / totalQuestions \* 100;

resultArea.setText("Your Score: " + totalScore + "%\n\n");

showCorrectAnswers(panel);

});

}

private void addQuestion(JPanel panel, String question, String[] options, int correctIndex) {

JPanel questionPanel = new JPanel();

questionPanel.setLayout(new GridLayout(options.length + 1, 1));

questionPanel.add(new JLabel(question));

ButtonGroup buttonGroup = new ButtonGroup();

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

JRadioButton radioButton = new JRadioButton(options[i]);

final int index = i;

radioButton.addActionListener(e -> {

if (radioButton.isSelected()) {

if (index == correctIndex) {

score++;

}

}

});

buttonGroup.add(radioButton);

questionPanel.add(radioButton);

}

panel.add(questionPanel);

}

private int checkAnswers() {

return score;

}

private void showCorrectAnswers(JPanel panel) {

// Display correct answers and the user's score at the end

String message = "Correct Answers:\n" +

"1. Central Processing Unit\n" +

"2. JavaScript\n" +

"3. Hypertext Markup Language\n\n";

double totalScore = (double) score / 3 \* 100;

message += "Your Score: " + totalScore + "%";

JOptionPane.showMessageDialog(this, message);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

QuizApplication quizApp = new QuizApplication();

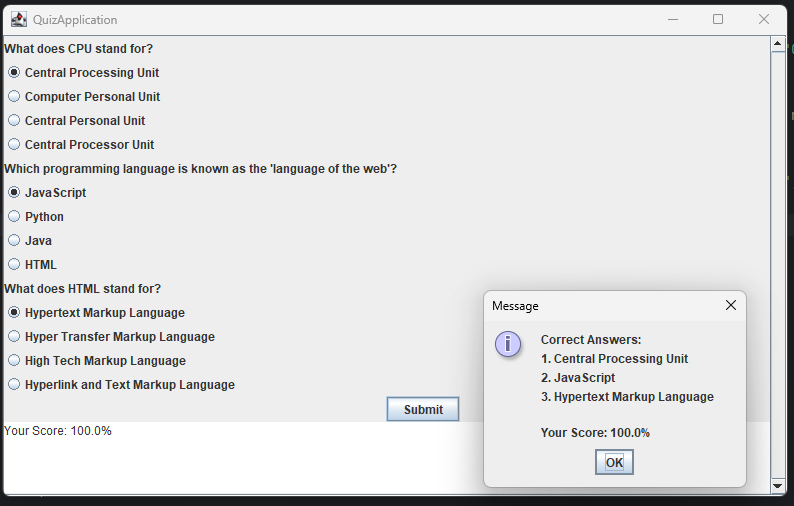
quizApp.setVisible(true);

});

}

}

**OUTPUT**



1. **Create a shopping cart program using Java Swing. The program should allow users to add items to the cart, display the cart's contents (including the items and their prices), and calculate the total amount to be paid. Create a Java Swing application that fulfills these requirements.**

**CODE**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.List;

class ShoppingItem {

private String itemName;

private double price;

public ShoppingItem(String itemName, double price) {

this.itemName = itemName;

this.price = price;

}

public String getItemName() {

return itemName;

}

public double getPrice() {

return price;

}

}

public class ShoppingCart extends JFrame {

private List<ShoppingItem> cartItems;

private JTextArea cartContents;

private JLabel totalLabel;

private int totalItems;

private double totalAmount;

public ShoppingCart() {

setTitle("Shopping Cart");

setSize(400, 300);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLayout(new BoxLayout(getContentPane(), BoxLayout.Y\_AXIS));

cartItems = new ArrayList<>();

totalAmount = 0.0;

totalItems = 0;

JLabel itemNameLabel = new JLabel("Item Name:");

JTextField itemNameField = new JTextField(20);

JLabel priceLabel = new JLabel("Price:");

JTextField priceField = new JTextField(10);

JButton addButton = new JButton("Add to Cart");

cartContents = new JTextArea(10, 30);

cartContents.setEditable(false);

JScrollPane scrollPane = new JScrollPane(cartContents);

totalLabel = new JLabel("Total Items: 0");

JButton calculateButton = new JButton("Calculate Total");

JTextField totalField = new JTextField(15);

totalField.setEditable(false);

addButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String itemName = itemNameField.getText();

double price = Double.parseDouble(priceField.getText());

ShoppingItem newItem = new ShoppingItem(itemName, price);

cartItems.add(newItem);

totalAmount += price;

totalItems++;

totalLabel.setText("Total Items: " + totalItems);

displayCart();

}

});

calculateButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

totalField.setText("Total Amount: $" + totalAmount);

}

});

add(itemNameLabel);

add(itemNameField);

add(priceLabel);

add(priceField);

add(addButton);

add(scrollPane);

add(totalLabel);

add(calculateButton);

add(totalField);

}

private void displayCart() {

cartContents.setText("Cart Contents:\n\n");

for (ShoppingItem item : cartItems) {

cartContents.append(item.getItemName() + " - $" + item.getPrice() + "\n");

}

cartContents.append("\nTotal Amount: $" + totalAmount);

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

ShoppingCart cart = new ShoppingCart();

cart.setVisible(true);

});

}

}

**OUTPUT**



1. **Write a program to create google account with details as shown in figure.**

**CODE**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.image.BufferedImage;

import java.io.File;

import javax.imageio.ImageIO;

import javax.swing.border.LineBorder;

public class GoogleAccount extends JFrame implements ActionListener {

private JLabel firstNameLabel, lastNameLabel, emailLabel, passwordLabel, confirmPasswordLabel, mobileLabel, dobLabel, genderLabel;

private JTextField firstNameField, lastNameField, emailField, mobileField, dobField;

private JPasswordField passwordField, confirmPasswordField;

private JButton createButton, resetButton, cancelButton;

private ImageIcon imageIcon;

private JComboBox<String> countryCodes, genderComboBox;

public GoogleAccount() {

super("Create your Google Account");

setSize(500, 600);

setLayout(new BorderLayout());

firstNameLabel = new JLabel("First Name");

lastNameLabel = new JLabel("Last Name");

emailLabel = new JLabel("Email");

passwordLabel = new JLabel("Password");

confirmPasswordLabel = new JLabel("Confirm Password");

mobileLabel = new JLabel("Mobile Number");

dobLabel = new JLabel("Date of Birth (MM/DD/YYYY)");

genderLabel = new JLabel("Gender");

firstNameLabel.setOpaque(true);

firstNameLabel.setBackground(new Color(230, 230, 250));

lastNameLabel.setOpaque (true);

lastNameLabel.setBackground(new Color(230, 230, 250));

emailLabel.setOpaque(true);

emailLabel.setBackground(new Color(230, 230, 250));

passwordLabel.setOpaque(true);

passwordLabel.setBackground(new Color(230, 230, 250));

confirmPasswordLabel.setOpaque(true);

confirmPasswordLabel.setBackground(new Color(230, 230, 250));

mobileLabel.setOpaque(true);

mobileLabel.setBackground(new Color(230, 230, 250));

dobLabel.setOpaque(true);

dobLabel.setBackground(new Color(230, 230, 250));

genderLabel.setOpaque(true);

genderLabel.setBackground(new Color(230, 230, 250));

ImageIcon firstNameIcon = resizeImage("F:\\New folder\\name.png", 15, 15);

ImageIcon lastNameIcon = resizeImage("F:\\New folder\\name.png", 15, 15);

ImageIcon emailIcon = resizeImage("F:\\New folder\\email.png", 15, 15);

ImageIcon passwordIcon = resizeImage("F:\\New folder\\password.png", 15, 15);

ImageIcon confirmPassIcon = resizeImage("F:\\New folder\\confirm\_password.png", 15, 15);

firstNameField = new JTextField(15);

lastNameField = new JTextField(15);

emailField = new JTextField(15);

passwordField = new JPasswordField(15);

confirmPasswordField = new JPasswordField(15);

mobileField = new JTextField(15);

dobField = new JTextField(15);

createButton = new JButton("Create Account");

resetButton = new JButton("Reset");

cancelButton = new JButton("Cancel");

String[] genders = {"Male", "Female", "Other"};

genderComboBox = new JComboBox<>(genders);

String[] countryArray = {"+1", "+44", "+91", "+61", "+81"};

countryCodes = new JComboBox<>(countryArray);

imageIcon = new ImageIcon("F:\\New folder\\Google.png");

Image img = imageIcon.getImage();

Image newImg = img.getScaledInstance(500, 150, Image.SCALE\_SMOOTH);

imageIcon = new ImageIcon(newImg);

JLabel imageLabel = new JLabel(imageIcon);

firstNameField.setBorder(new LineBorder(Color.BLUE, 1));

lastNameField.setBorder(new LineBorder(Color.BLUE, 1));

emailField.setBorder(new LineBorder(Color.BLUE, 1));

passwordField.setBorder(new LineBorder(Color.BLUE, 1));

confirmPasswordField.setBorder(new LineBorder(Color.BLUE, 1));

mobileField.setBorder(new LineBorder(Color.BLUE, 1));

dobField.setBorder(new LineBorder(Color.BLUE, 1));

JPanel formPanel = new JPanel(new GridBagLayout());

GridBagConstraints gbc = new GridBagConstraints();

gbc.anchor = GridBagConstraints.WEST;

gbc.insets = new Insets(5, 5, 5, 5);

gbc.gridx = 0;

gbc.gridy = 0;

formPanel.add(firstNameLabel, gbc);

gbc.gridx = 1;

formPanel.add(firstNameField, gbc);

gbc.gridx = 0;

gbc.gridy = 1;

formPanel.add(lastNameLabel, gbc);

gbc.gridx = 1;

formPanel.add(lastNameField, gbc);

gbc.gridx = 0;

gbc.gridy = 2;

formPanel.add(emailLabel, gbc);

gbc.gridx = 1;

formPanel.add(emailField, gbc);

gbc.gridx = 0;

gbc.gridy = 3;

formPanel.add(passwordLabel, gbc);

gbc.gridx = 1;

formPanel.add(passwordField, gbc);

gbc.gridx = 0;

gbc.gridy = 4;

formPanel.add(confirmPasswordLabel, gbc);

gbc.gridx = 1;

formPanel.add(confirmPasswordField, gbc);

gbc.gridx = 0;

gbc.gridy = 5;

formPanel.add(mobileLabel, gbc);

gbc.gridx = 2; // This is the adjustment for the "Mobile Number" field

formPanel.add(countryCodes, gbc);

gbc.gridx = 1;

formPanel.add(mobileField, gbc);

gbc.gridx = 0;

gbc.gridy = 6;

formPanel.add(dobLabel, gbc);

gbc.gridx = 1;

formPanel.add(dobField, gbc);

gbc.gridx = 0;

gbc.gridy = 7;

formPanel.add(genderLabel, gbc);

gbc.gridx = 1;

formPanel.add(genderComboBox, gbc);

gbc.gridx = 0;

gbc.gridy = 8;

formPanel.add(createButton, gbc);

gbc.gridx = 1;

formPanel.add(resetButton, gbc);

gbc.gridx = 2;

formPanel.add(cancelButton, gbc);

createButton.setBackground(Color.GREEN);

createButton.setForeground(Color.WHITE);

resetButton.setBackground(Color.ORANGE);

cancelButton.setBackground(Color.RED);

resetButton.setForeground(Color.WHITE);

cancelButton.setForeground(Color.WHITE);

createButton.addActionListener(this);

resetButton.addActionListener(this);

cancelButton.addActionListener(this);

add(imageLabel, BorderLayout.NORTH);

add(formPanel, BorderLayout.CENTER);

setVisible(true);

}

public void actionPerformed(ActionEvent e) {

if (e.getSource() == createButton) {

if (validateInputs()) {

JOptionPane.showMessageDialog(this, "Account created successfully!");

} else {

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

}

} else if (e.getSource() == resetButton) {

resetFields();

} else if (e.getSource() == cancelButton) {

System.exit(0);

}

}

private boolean validateInputs() {

return !firstNameField.getText().isEmpty() &&

!lastNameField.getText().isEmpty() &&

isEmailValid(emailField.getText()) &&

!passwordField.getText().isEmpty() &&

!confirmPasswordField.getText().isEmpty() &&

!mobileField.getText().isEmpty() &&

!dobField.getText().isEmpty();

}

private boolean isEmailValid(String email) {

return email.contains(".") && email.substring(email.indexOf(".") + 1).length() >= 2;

}

private void resetFields() {

firstNameField.setText("");

lastNameField.setText("");

emailField.setText("");

passwordField.setText("");

confirmPasswordField.setText("");

mobileField.setText("");

dobField.setText("");

countryCodes.setSelectedIndex(0);

genderComboBox.setSelectedIndex(0);

}

private ImageIcon resizeImage(String imagePath, int width, int height) {

ImageIcon resizedIcon = null;

try {

BufferedImage image = ImageIO.read(new File(imagePath));

Image img = image.getScaledInstance(width, height, Image.SCALE\_SMOOTH);

resizedIcon = new ImageIcon(img);

} catch (Exception ex) {

ex.printStackTrace();

}

return resizedIcon;

}

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

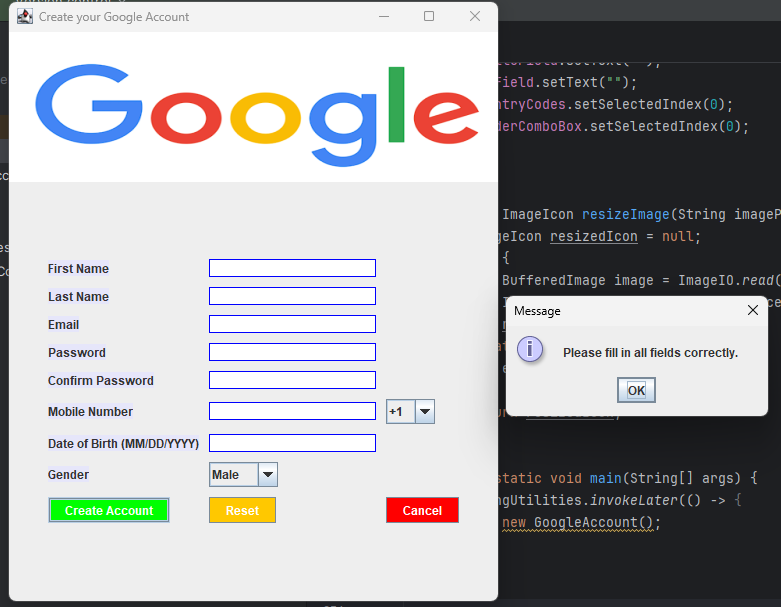
new GoogleAccount();

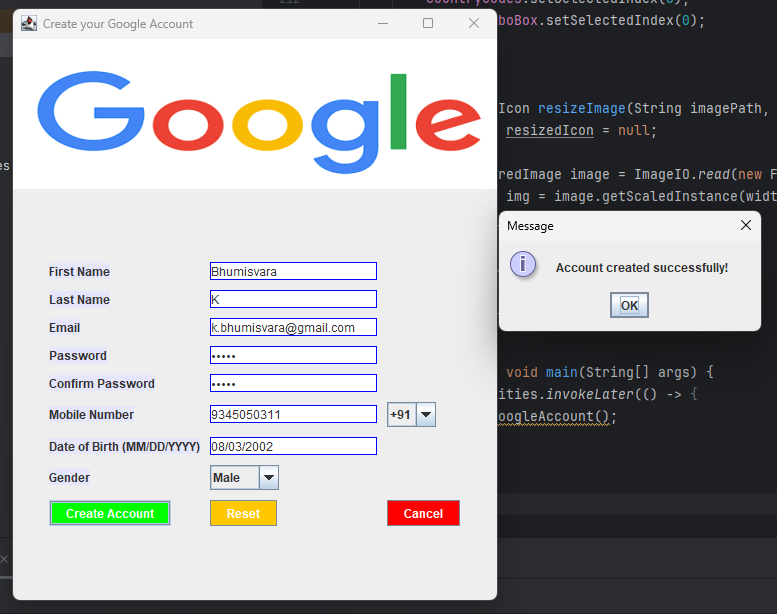
});

}

}

**OUTPUT**





**RESULT**

Thus, The GUI program in Java Swing has been successfully implemented.