import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.Color;  
  
  
public class CariñoMidtermExam extends JFrame implements ActionListener {  
  
 // Components for the calculator  
 private JTextField display;  
 private JButton[] numberButtons = new JButton[10];  
 private JButton[] functionButtons = new JButton[8];  
 private JButton addButton, subButton, mulButton, divButton;  
 private JButton decButton, equButton, delButton, clrButton;  
  
 // Variables for calculations  
 private double num1 = 0, num2 = 0, result = 0;  
 private char operator;  
  
 // Components for login form  
 private JTextField emailField = new JTextField(20);  
 private JPasswordField passwordField = new JPasswordField(20);  
 private JButton loginButton = new JButton("Login");  
  
 // Card layout to switch between login and calculator  
 private CardLayout cardLayout = new CardLayout();  
 private JPanel mainPanel = new JPanel(cardLayout);  
  
 public CariñoMidtermExam() {  
 // Frame settings  
 setTitle("Calculator App");  
 setSize(420, 550);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 setLocationRelativeTo(null);  
  
 // Setup login panel  
 JPanel loginPanel = setupLoginPanel();  
  
 // Setup calculator panel  
 JPanel calculatorPanel = setupCalculatorPanel();  
  
 // Add panels to main panel with CardLayout  
 mainPanel.add(loginPanel, "Login");  
 mainPanel.add(calculatorPanel, "Calculator");  
  
 add(mainPanel);  
 cardLayout.show(mainPanel, "Login");  
  
 setVisible(true);  
 }  
  
 private JPanel setupLoginPanel() {  
 JPanel loginPanel = new JPanel(new GridBagLayout());  
 GridBagConstraints gbc = new GridBagConstraints();  
 gbc.insets = new Insets(10, 10, 10, 10);  
 gbc.fill = GridBagConstraints.HORIZONTAL;  
  
 // Load and resize logo image  
 ImageIcon originalLogoIcon = new ImageIcon("SiteLogo.jpg");  
 Image originalLogoImage = originalLogoIcon.getImage();  
 Image resizedLogoImage = originalLogoImage.getScaledInstance(250, -1, Image.SCALE\_SMOOTH);  
 ImageIcon resizedLogoIcon = new ImageIcon(resizedLogoImage);  
 JLabel logoLabel = new JLabel(resizedLogoIcon);  
  
 // Logo  
 gbc.gridx = 0;  
 gbc.gridy = 0;  
 gbc.gridwidth = 3;  
 gbc.anchor = GridBagConstraints.CENTER;  
 loginPanel.add(logoLabel, gbc);  
  
 // Welcome label  
 JLabel welcomeLabel = new JLabel("Welcome to the Calculator App");  
 welcomeLabel.setFont(new Font("Arial", Font.BOLD, 20));  
 gbc.gridx = 0;  
 gbc.gridy = 1;  
 gbc.gridwidth = 3;  
 gbc.anchor = GridBagConstraints.CENTER;  
 gbc.fill = GridBagConstraints.NONE; // Ensure the label is not stretched  
 loginPanel.add(welcomeLabel, gbc);  
  
 // Email label  
 gbc.gridx = 0;  
 gbc.gridy = 2; // Adjusted to move below the welcome label  
 gbc.gridwidth = 1;  
 gbc.anchor = GridBagConstraints.CENTER;  
 gbc.fill = GridBagConstraints.HORIZONTAL; // Reset fill to horizontal for fields  
 loginPanel.add(new JLabel("Email:"), gbc);  
  
 // Email field  
 gbc.gridx = 1;  
 gbc.gridy = 2;  
 gbc.gridwidth = 2;  
 gbc.anchor = GridBagConstraints.CENTER;  
 loginPanel.add(emailField, gbc);  
  
 // Password label  
 gbc.gridx = 0;  
 gbc.gridy = 3; // Adjusted to move below the email field  
 gbc.gridwidth = 1;  
 gbc.anchor = GridBagConstraints.CENTER;  
 loginPanel.add(new JLabel("Password:"), gbc);  
  
 // Password field  
 gbc.gridx = 1;  
 gbc.gridy = 3;  
 gbc.gridwidth = 2;  
 gbc.anchor = GridBagConstraints.CENTER;  
 loginPanel.add(passwordField, gbc);  
  
 // Login button  
 gbc.gridx = 0;  
 gbc.gridy = 4; // Adjusted to move below the password field  
 gbc.gridwidth = 3;  
 gbc.anchor = GridBagConstraints.CENTER;  
 gbc.fill = GridBagConstraints.NONE;  
 loginButton.addActionListener(e -> login());  
 loginPanel.add(loginButton, gbc);  
  
 return loginPanel;  
}  
  
 private void login() {  
 String username = emailField.getText();  
 String password = new String(passwordField.getPassword());  
 if (username.equals("ivan@gmail.com") && password.equals("admin")) {  
 JOptionPane.showMessageDialog(this, "Login Successful");  
 cardLayout.show(mainPanel, "Calculator");  
 } else {  
 JOptionPane.showMessageDialog(this, "Invalid username or password");  
 }  
 }  
  
 private JPanel setupCalculatorPanel() {   
 // Display field  
 display = new JTextField();  
 display.setEditable(false);  
 display.setFont(new Font("Arial", Font.PLAIN, 24));  
 display.setHorizontalAlignment(JTextField.RIGHT);  
 //Bowrain  
 Color[] ColorRainbow = {  
 Color.blue, Color.red, Color.green, Color.yellow, Color.orange, new Color(143, 0, 255)/\*Violet \*/, new Color(75, 0, 130)/\*Indigo \*/  
 };  
  
 // Function buttons  
 addButton = new JButton("+");  
 subButton = new JButton("-");  
 mulButton = new JButton("\*");  
 divButton = new JButton("/");  
 decButton = new JButton(".");  
 equButton = new JButton("=");  
 delButton = new JButton("Delete");  
 clrButton = new JButton("Clear");  
  
 functionButtons[0] = addButton;  
 functionButtons[1] = subButton;  
 functionButtons[2] = mulButton;  
 functionButtons[3] = divButton;  
 functionButtons[4] = decButton;  
 functionButtons[5] = equButton;  
 functionButtons[6] = delButton;  
 functionButtons[7] = clrButton;  
  
 for (JButton button : functionButtons) {  
 button.addActionListener(this);  
 //Bowrain  
 button.setFont(new Font("Arial", Font.PLAIN, 35));  
 button.setForeground(Color.white);  
 button.setBackground(Color.RED);  
   
   
 }  
  
 // Number buttons  
 for (int i = 0; i < 10; i++) {  
 numberButtons[i] = new JButton(String.valueOf(i));  
 numberButtons[i].addActionListener(this);  
 // Bowrain  
 numberButtons[i].setFont(new Font("Arial", Font.PLAIN, 35));  
 numberButtons[i].setBackground(ColorRainbow [i % ColorRainbow.length]);  
 numberButtons[i].setForeground(Color.WHITE);  
 }  
  
 // Panel for buttons  
 JPanel panel = new JPanel(new GridLayout(4, 4, 10, 10));  
 for (int i = 1; i <= 3; i++) panel.add(numberButtons[i]);  
 panel.add(addButton);  
 for (int i = 4; i <= 6; i++) panel.add(numberButtons[i]);  
 panel.add(subButton);  
 for (int i = 7; i <= 9; i++) panel.add(numberButtons[i]);  
 panel.add(mulButton);  
 panel.add(decButton);  
 panel.add(numberButtons[0]);  
 panel.add(equButton);  
 panel.add(divButton);  
  
 JPanel calculatorPanel = new JPanel(new BorderLayout());  
 calculatorPanel.add(display, BorderLayout.NORTH);  
 calculatorPanel.add(panel, BorderLayout.CENTER);  
  
 JPanel bottomPanel = new JPanel(new GridLayout(1, 2));  
 bottomPanel.add(delButton);  
 bottomPanel.add(clrButton);  
 calculatorPanel.add(bottomPanel, BorderLayout.SOUTH);  
  
 return calculatorPanel;  
 }  
  
 public static void main(String[] args) {  
 SwingUtilities.invokeLater(CariñoMidtermExam::new);  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 for (int i = 0; i < 10; i++) {  
 if (e.getSource() == numberButtons[i]) {  
 display.setText(display.getText().concat(String.valueOf(i)));  
 }  
 }  
 if (e.getSource() == decButton) {  
 if (!display.getText().contains(".")) {  
 display.setText(display.getText().concat("."));  
 }  
 }  
 if (e.getSource() == addButton) {  
 num1 = Double.parseDouble(display.getText());  
 operator = '+';  
 display.setText("");  
   
 }  
 if (e.getSource() == subButton) {  
 num1 = Double.parseDouble(display.getText());  
 operator = '-';  
 display.setText("");  
 }  
 if (e.getSource() == mulButton) {  
 num1 = Double.parseDouble(display.getText());  
 operator = '\*';  
 display.setText("");  
 }  
 if (e.getSource() == divButton) {  
 num1 = Double.parseDouble(display.getText());  
 operator = '/';  
 display.setText("");  
 }  
 if (e.getSource() == equButton) {  
 num2 = Double.parseDouble(display.getText());  
 switch (operator) {  
 case '+':  
 result = num1 + num2;  
 break;  
 case '-':  
 result = num1 - num2;  
 break;  
 case '\*':  
 result = num1 \* num2;  
 break;  
 case '/':  
 result = num1 / num2;  
 break;  
 }  
 display.setText(String.valueOf(result));  
 num1 = result;  
 }  
 if (e.getSource() == clrButton) {  
 display.setText("");  
 }  
 if (e.getSource() == delButton) {  
 String string = display.getText();  
 if (string.length() > 0) {  
 display.setText(string.substring(0, string.length() - 1));  
 }  
 }  
 }  
}