Assignment No 4

Name of Lab Assignment

Mind Marathon

Team

Nimra Niamat

SP24-BSE-099

Jannat-ul-mava

Sp24-BSE-049

Amna Zulifqar

Sp24-BSe-013

**Code**

**Welcome Class**

package com.example.mind\_marathon\_project;  
  
import javafx.application.Application;  
  
import javafx.scene.Scene;  
import javafx.animation.ScaleTransition;  
import javafx.geometry.Pos;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.scene.control.Button;  
import javafx.util.Duration;  
import javafx.scene.layout.StackPane;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
import java.io.File;  
import java.io.IOException;  
  
public class Welcome\_page extends Application {  
  
 @Override  
 public void start(Stage stage) throws IOException {  
  
  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage);  
  
  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #f1f5f6;");  
 root.setTop(customTitleBar);  
  
  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #fffcf6; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #3c6ca8; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
  
 ImageView welcomeTag = loadImage("/com/example/mind\_marathon\_project/Welcome\_tag.png");  
 welcomeTag.setFitWidth(300);  
 welcomeTag.setFitHeight(150);  
  
  
 ImageView logo = loadImage("/com/example/mind\_marathon\_project/logo.png");  
 logo.setFitWidth(300);  
 logo.setFitHeight(250);  
  
  
 Button playButton = createPlayButton();  
  
 cardPane.getChildren().addAll(welcomeTag, logo, playButton);  
  
  
 root.setCenter(cardPane);  
 StackPane centerPane = new StackPane(cardPane);  
 centerPane.setAlignment(Pos.*CENTER*);  
 root.setCenter(centerPane);  
  
  
 playButton.setOnAction(e -> {  
 if (isUserSignedUp()) {  
  
 try {  
 new Login\_page2().start(new Stage());  
 stage.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 } else {  
  
 try {  
 new Login\_page().start(new Stage());  
 stage.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
 });  
  
  
 Scene scene = new Scene(root, 800, 600);  
 stage.initStyle(StageStyle.*UNDECORATED*);  
 stage.setMaximized(true);  
 stage.setScene(scene);  
 stage.show();  
 }  
  
 private ImageView loadImage(String path) {  
 try {  
 return new ImageView(new Image(getClass().getResourceAsStream(path)));  
 } catch (NullPointerException e) {  
 throw new RuntimeException("Image not found: " + path, e);  
 }  
 }  
  
 private Button createPlayButton() {  
 Button playButton = new Button("PLAY");  
 playButton.setStyle("-fx-background-color: #3c6ca8;-fx-border-color:#b024cd;-fx-border-width: 2px;-fx-border-radius: 20px;-fx-text-fill: #ffffff;-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 10 20");  
 try {  
 addButtonEffects(playButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 } catch (NullPointerException e) {  
 throw new RuntimeException(e);  
 }  
 return playButton;  
 }  
  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
 }  
  
 private boolean isUserSignedUp() {  
 File userFile = new File("user\_data.txt");  
 return userFile.exists();  
 }  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

**Sign up Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
import java.io.\*;  
  
public class Login\_page extends Application {  
  
 @Override  
 public void start(Stage stage2) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage2);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 VBox vbox = new VBox();  
 vbox.setAlignment(Pos.*CENTER*);  
 vbox.setStyle("-fx-background-color: #fffcf6;" +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 vbox.setMaxWidth(600);  
 vbox.setMaxHeight(500);  
  
 ImageView login;  
 try {  
 login = new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/login.png")));  
 } catch (NullPointerException e) {  
 throw new RuntimeException("Image not found: /com/example/mind\_marathon\_project/login.png", e);  
 }  
 login.setFitWidth(170);  
 login.setFitHeight(90);  
  
 StackPane formContainer = new StackPane();  
 formContainer.setAlignment(Pos.*CENTER*);  
  
 ImageView loginImage;  
 try {  
 loginImage = new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/log\_pic.png")));  
 } catch (NullPointerException e) {  
 throw new RuntimeException("Image not found: /com/example/mind\_marathon\_project/log\_pic.png", e);  
 }  
 loginImage.setFitWidth(350);  
 loginImage.setFitHeight(350);  
  
 VBox formFields = new VBox(40);  
 formFields.setAlignment(Pos.*CENTER*);  
 formFields.setPadding(new Insets(10));  
  
 TextField nameField = new TextField();  
 nameField.setMaxWidth(150);  
 nameField.setMinHeight(40);  
 nameField.setPromptText("\tEnter your name");  
 nameField.setStyle("-fx-background-color: #fffcf6;-fx-background-radius: 20px;-fx-border-width: 2px;" +  
 "-fx-border-radius: 20px;-fx-prompt-text-fill: #675c6e;-fx-border-color: #b78fd6;" +  
 "-fx-text-fill: #3c6ca8;-fx-font-weight: bold;-fx-font-family: Calibri;" +  
 "-fx-padding: 5px;-fx-font-size:14px; ");  
  
 Slider ageSlider = new Slider();  
 ageSlider.setMin(1);  
 ageSlider.setMax(60);  
 ageSlider.setPrefWidth(80);  
 ageSlider.setStyle(  
 "-fx-control-inner-background: #79ccab ;" +  
 "-fx-background-color: #3c6ca8;" +  
 "-fx-border-color: transparent;" +  
 "-fx-base: #b78fd6;"  
 );  
  
 Label ageLabel = new Label("0");  
 ageLabel.setStyle("-fx-text-fill: #ffffff; -fx-font-size: 14px;-fx-font-weight: bold");  
 ageSlider.valueProperty().addListener((obs, oldVal, newVal) -> ageLabel.setText(String.*valueOf*(newVal.intValue())));  
 Label age1 =new Label("Age:");  
 age1.setStyle("-fx-text-fill: #ffffff;-fx-font-size: 14px;-fx-font-weight: bold");  
 HBox ageContainer = new HBox(10,age1 , ageSlider, ageLabel);  
 ageContainer.setAlignment(Pos.*CENTER*);  
  
 PasswordField passwordField = new PasswordField();  
 passwordField.setMaxWidth(150);  
 passwordField.setMinHeight(40);  
 passwordField.setPromptText(" Enter your password");  
 passwordField.setStyle("-fx-background-color: #fffcf6;-fx-prompt-text-fill: #675c6e;" +  
 "-fx-border-color: #b78fd6; " +  
 "-fx-border-radius: 20px;-fx-border-width: 2px;" +  
 "-fx-background-radius: 20px; " +  
 "-fx-padding: 5px; " +  
 "-fx-font-size: 14px;-fx-text-fill: #3c6ca8;-fx-font-weight: bold;-fx-font-family: Calibri;");  
  
 formFields.getChildren().addAll(nameField, ageContainer, passwordField);  
 formContainer.getChildren().addAll(loginImage, formFields);  
  
 Button okButton = new Button("OK");  
 okButton.setStyle("-fx-background-color: #fdf58e;-fx-border-color:#f59eb7;" +  
 "-fx-border-width: 2px;-fx-border-radius: 20px;-fx-text-fill: #3c6ca8;" +  
 "-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;" +  
 "-fx-font-size: 18px;-fx-padding: 10 20");  
 addButtonEffects(okButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
  
// okButton.setOnAction(actionEvent -> {  
// Stage currentStage = (Stage) okButton.getScene().getWindow();  
// String name = nameField.getText();  
// String age = ageLabel.getText(); // or use ageSlider.getValue() if you prefer  
// String password = passwordField.getText();  
//  
// if (name.isEmpty() || password.isEmpty()) {  
// showAlert("Error", "Name and Password cannot be empty!");  
// return;  
// }  
//  
// // Store user data  
// // UserInfo.storeUserData(name, age, password);  
//  
// try {  
// new Login\_page2().start(new Stage());  
// currentStage.close();  
// } catch (Exception e) {  
// throw new RuntimeException(e);  
// }  
//  
// String filepath = "user\_data.txt";  
// File file = new File("User\_data.txt");  
// try (BufferedWriter writer = new BufferedWriter(new FileWriter(filepath, true))) {  
// writer.write("Name: " + name + "\n");  
// writer.write("Age: " + age + "\n");  
// writer.write("Password: " + password + "\n\n");  
// showAlert("Success", "Your login has been stored successfully!");  
// try (BufferedReader reader = new BufferedReader(new FileReader(filepath))) {  
// String storedData = reader.lines().reduce("", (prev, curr) -> prev + curr + "\n");  
// //showAlert("Stored Data", storedData);  
//  
// } catch (IOException e) {  
// e.getMessage();  
// }} catch (IOException e) {  
// throw new RuntimeException(e);  
// }  
//  
// });  
 okButton.setOnAction(e -> {  
 String name = nameField.getText();  
 String password = passwordField.getText();  
 if (name.isEmpty() || password.isEmpty()) {  
 System.*out*.println("Please fill in both fields.");  
 return;  
 }  
  
  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter("user\_data.txt", true))) {  
 writer.write("Name: " + name + "\n");  
 writer.write("Password: " + password + "\n");  
  
  
 writer.flush();  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
  
  
 try {  
 new Login\_page2().start(new Stage());  
 stage2.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 });  
  
  
  
 vbox.getChildren().addAll(login, formContainer, okButton);  
 root.setCenter(vbox);  
  
 Scene scene = new Scene(root, 800, 600);  
 stage2.initStyle(StageStyle.*UNDECORATED*);  
 stage2.setMaximized(true);  
 stage2.setScene(scene);  
 stage2.show();  
  
  
 }  
  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
 }  
  
 private void showAlert(String title, String message) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle(title);  
 alert.setHeaderText(null);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

**Login Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.event.ActionEvent;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.\*;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
import java.io.File;  
import java.io.IOException;  
import java.util.Scanner;  
  
  
public class Login\_page2 extends Application {  
  
 public void start(Stage stage3) throws Exception {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage3);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 VBox vbox = new VBox();  
 vbox.setAlignment(Pos.*CENTER*);  
 vbox.setStyle("-fx-background-color: #fffcf6;" +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 vbox.setMaxWidth(600);  
 vbox.setMaxHeight(500);  
  
 ImageView login;  
 try {  
 login = new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/logi.png")));  
 } catch (NullPointerException e) {  
 throw new RuntimeException("Image not found: /com/example/mind\_marathon\_project/logi.png", e);  
 }  
 login.setFitWidth(170);  
 login.setFitHeight(90);  
  
 StackPane formContainer = new StackPane();  
 formContainer.setAlignment(Pos.*CENTER*);  
  
 ImageView login\_image;  
 try {  
 login\_image = new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/log\_pic.png")));  
 } catch (NullPointerException e) {  
 throw e; }  
 login\_image.setFitWidth(350);  
 login\_image.setFitHeight(350);  
  
 VBox formFields = new VBox(40);  
 formFields.setAlignment(Pos.*CENTER*);  
 formFields.setPadding(new Insets(10));  
  
 TextField nameField = new TextField();  
 nameField.setMaxWidth(150);  
 nameField.setMinHeight(40);  
 nameField.setPromptText("\tEnter your name");  
 nameField.setStyle("-fx-background-color: #fffcf6;-fx-background-radius: 20px;-fx-border-width: 2px;-fx-border-radius: 20px;-fx-prompt-text-fill: #675c6e;-fx-border-color: #b78fd6;-fx-text-fill: #3c6ca8;-fx-font-weight: bold;-fx-font-family: Calibri;-fx-padding: 5px;-fx-font-size:14px; ");  
  
 PasswordField passwordField = new PasswordField();  
 passwordField.setMaxWidth(150);  
 passwordField.setMinHeight(40);  
 passwordField.setPromptText(" Enter your password");  
 passwordField.setStyle("-fx-background-color: #fffcf6;-fx-prompt-text-fill: #675c6e;" +  
 "-fx-border-color: #b78fd6; " +  
 "-fx-border-radius: 20px;-fx-border-width: 2px;" +  
 "-fx-background-radius: 20px; " +  
 "-fx-padding: 5px; " +  
 "-fx-font-size: 14px;-fx-text-fill: #3c6ca8;-fx-font-weight: bold;-fx-font-family: Calibri;");  
  
 // Add form fields to the container  
 formFields.getChildren().addAll(nameField,passwordField);  
 formContainer.getChildren().addAll(login\_image, formFields);  
  
  
 Button okButton = new Button("OK");  
 okButton.setStyle("-fx-background-color: #fdf58e;-fx-border-color:#f59eb7;-fx-border-width: 2px;-fx-border-radius: 20px;-fx-text-fill: #3c6ca8;-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 10 20");  
 try {  
 addButtonEffects(okButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 }  
 catch (NullPointerException e) {  
 throw new RuntimeException( e);  
 }  
  
  
  
// okButton.setOnAction(e -> {  
// Stage currentSatge = (Stage)okButton.getScene().getWindow();  
// String name = nameField.getText();  
// String password = passwordField.getText();  
//  
// if (name.isEmpty() || password.isEmpty()) {  
// System.out.println("Name and password are empty");  
// return;  
// }  
//  
// boolean isAuthenticated = false;  
//  
// try (Scanner scanner = new Scanner(new File("user\_data.txt"))) {  
// while (scanner.hasNextLine()) {  
// String line = scanner.nextLine();  
// if (line.startsWith("Name: ")) {  
// String storedName = line.substring(6).trim();  
// String storedPassword = "";  
//  
// while (scanner.hasNextLine()) {  
// line = scanner.nextLine();  
// if (line.startsWith("Password: ")) {  
// storedPassword = line.substring(10).trim();  
// break;  
// }  
// }  
//  
// if (name.equals(storedName) && password.equals(storedPassword)) {  
// isAuthenticated = true;  
// break;  
// }  
// }  
// }  
// } catch (IOException ex) {  
// System.out.println("Failed to login");  
// return;  
// }  
//  
// if (isAuthenticated) {  
// System.out.println("Successfully logged in");  
// try {  
//  
// new Menu\_page().start(new Stage());  
//  
// currentSatge.close(); // Close the login stage  
// } catch (Exception ex) {  
// ex.printStackTrace();  
// }  
// } else {  
// System.out.println("Failed to login");  
// }  
// });  
 okButton.setOnAction(e -> {  
 String name = nameField.getText();  
 String password = passwordField.getText();  
  
 if (name.isEmpty() || password.isEmpty()) {  
 System.*out*.println("Name and password are empty");  
 return;  
 }  
  
 boolean isAuthenticated = false;  
  
 try (Scanner scanner = new Scanner(new File("user\_data.txt"))) {  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 if (line.startsWith("Name: ")) {  
 String storedName = line.substring(6).trim();  
 String storedPassword = "";  
  
 while (scanner.hasNextLine()) {  
 line = scanner.nextLine();  
 if (line.startsWith("Password: ")) {  
 storedPassword = line.substring(10).trim();  
 break;  
 }  
 }  
  
 if (name.equals(storedName) && password.equals(storedPassword)) {  
 isAuthenticated = true;  
 break;  
 }  
 }  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 return;  
 }  
  
 if (isAuthenticated) {  
 System.*out*.println("Successfully logged in");  
 try {  
 new Menu\_page().start(new Stage());  
 stage3.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Failed to login");  
 }  
 });  
  
  
  
 vbox.getChildren().addAll(login,formContainer,okButton);  
 root.setCenter(vbox);  
  
  
 Scene scene = new Scene(root,800,600);  
 stage3.initStyle(StageStyle.*UNDECORATED*);  
 stage3.setMaximized(true);  
 stage3.setScene(scene);  
 stage3.show();  
  
 }  
  
  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
  
  
 }  
 public static void main(String[] args) {  
 *launch*();  
 }  
}

**Menu Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Button;  
import javafx.scene.control.ButtonType;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.scene.shape.Circle;  
import javafx.stage.FileChooser;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.FileNotFoundException;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Scanner;  
  
public class Menu\_page extends Application {  
  
 private ImageView profileImageView;  
 private List<Image> preloadedImages;  
  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage stage5) throws Exception {  
 preloadedImages = loadPreloadedImages();  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage5);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #fffcf6;");  
 root.setTop(customTitleBar);  
  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #f7f2e8; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ffde59; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(650);  
 cardPane.setMaxHeight(500);  
  
 //VBox center=new VBox();  
 //center.setAlignment(Pos.CENTER);  
 Circle borderCircle = new Circle(75); // Radius of the circle  
 borderCircle.setStyle("-fx-fill: rgba(255,228,122,0.71); -fx-stroke: #5eb090; -fx-stroke-width: 7;-fx-background-radius: 70"); // Background color and border  
  
 // Profile image circle  
 profileImageView = new ImageView();  
 // profileImageView.setFitWidth(200);  
 // profileImageView.setFitHeight(200);  
 // profileImageView.setStyle("-fx-background-color: rgba(255,228,122,0.66);-fx-border-color: #5eb090;-fx-border-width: 9;-fx-background-radius: 50;-fx-border-radius: 50;");  
 // profileImageView.setClip(new Circle(70,70,70)); // Circle clipping  
 StackPane profileImageStack = new StackPane();  
 profileImageStack.getChildren().addAll(borderCircle, profileImageView);  
 profileImageStack.setAlignment(Pos.*CENTER*);  
 // "Choose Image" button  
 Button chooseImageButton = new Button("Choose image");  
 chooseImageButton.setStyle("-fx-background-color: #fdf58e; -fx-font-size: 14px; -fx-border-color: #ffc107;-fx-padding: 3px;");  
 //chooseImageButton.setOnAction(e -> chooseImage(stage5));  
 addButtonEffects(chooseImageButton,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 chooseImageButton.setOnAction(e -> openImageSelectionDialog(stage5));  
  
  
 // Name and Age labels  
 Label nameLabel = new Label("JM");  
 nameLabel.setStyle("-fx-font-size: 18px; -fx-font-weight: bold; -fx-text-fill: #000;");  
  
 Label ageLabel = new Label("19");  
 ageLabel.setStyle("-fx-font-size: 18px; -fx-font-weight: bold; -fx-text-fill: #000;");  
  
 HBox labels = new HBox(20,nameLabel, ageLabel);  
 labels.setAlignment(Pos.*CENTER*);  
 VBox profileSection = new VBox(10, profileImageStack, labels);  
 profileSection.setAlignment(Pos.*CENTER*);  
 HBox ImageStack=new HBox();  
 ImageStack.setSpacing(40);  
 ImageStack.getChildren().addAll(profileSection,chooseImageButton);  
 ImageStack.setAlignment(Pos.*CENTER*);  
  
 // Top-left and top-right buttons  
 Button Button1 = new Button();  
 ImageView imageView1 ;  
 try {  
 imageView1 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/award\_png.png").toExternalForm()));  
 }catch(NullPointerException e){  
 throw e;  
 }  
 imageView1.setFitWidth(30);  
 imageView1.setFitHeight(30);  
 Button1.setGraphic(imageView1);  
 Button1.setStyle("-fx-background-color: #3c6ca8;-fx-border-color: #ffe47a;-fx-border-width: 3px;-fx-background-radius: 10px;-fx-border-radius:7px;-fx-padding:4;-fx-background-size: 2");  
  
 Button Button2 = new Button();  
 ImageView imageView ;  
 try {  
 imageView = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/edit\_png.png").toExternalForm()));  
 }catch(NullPointerException e){  
 throw e;  
 }  
 imageView.setFitWidth(30);  
 imageView.setFitHeight(30);  
 Button2.setGraphic(imageView);  
 Button2.setStyle("-fx-background-color: #3c6ca8;-fx-border-color: #ffe47a;-fx-border-width: 3px;-fx-background-radius: 10px;-fx-border-radius:7px;-fx-padding:4;-fx-background-size: 2");  
  
 addButtonEffects(Button2,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 addButtonEffects(Button1,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
  
 // Bottom-left and bottom-right buttons  
 Button Button3 = new Button();  
 ImageView imageView3 ;  
 try {  
 imageView3 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/info\_png.png").toExternalForm()));  
 }catch(NullPointerException e){  
 throw e;  
 }  
 imageView3.setFitWidth(30);  
 imageView3.setFitHeight(30);  
 Button3.setGraphic(imageView3);  
 Button3.setStyle("-fx-background-color:#3c6ca8;-fx-border-color: #ffe47a;-fx-border-width: 3px;-fx-background-radius: 10px;-fx-border-radius:7px;-fx-padding:4;-fx-background-size: 2");  
  
 Button Button4 = new Button();  
 ImageView imageView4 ;  
 try {  
 imageView4 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/shop\_png.png").toExternalForm()));  
 }catch(NullPointerException e){  
 throw e;  
 }  
 imageView4.setFitWidth(30);  
 imageView4.setFitHeight(30);  
 Button4.setGraphic(imageView4);  
 Button4.setStyle("-fx-background-color: #3c6ca8;-fx-border-color: #ffe47a;-fx-border-width: 3px;-fx-background-radius: 10px;-fx-border-radius:7px;-fx-padding:4;-fx-background-size: 2");  
  
 VBox sideButtons = new VBox(10, Button1, Button2, Button3, Button4);  
 sideButtons.setAlignment(Pos.*TOP\_RIGHT*);  
 addButtonEffects(Button3,"/com/example/mind\_marathon\_project/shop\_png.png");  
 addButtonEffects(Button4,"/com/example/mind\_marathon\_project/info\_png.png");  
 // bottomButtons.maxHeight(100);  
 // bottomButtons.maxWidth(200);  
  
 // Play and Exit buttons  
 Button playButton = new Button("\t Play\t\t ");  
 playButton.setStyle("-fx-background-color: #ffe47a; -fx-background-radius: 10px;-fx-font-size: 16px; -fx-font-weight: bold; -fx-text-fill: #000;-fx-padding: 10;");  
 playButton.setOnAction(e -> System.*out*.println("Play clicked"));  
 addButtonEffects(playButton,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 Button exitButton = new Button(" \t Exit \t ");  
 exitButton.setStyle("-fx-background-color: #ffe47a; -fx-background-radius: 10px;-fx-font-size: 16px; -fx-font-weight: bold; -fx-text-fill: #000;-fx-padding: 10;");  
 addButtonEffects(exitButton,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
  
 VBox bottomBox = new VBox(playButton, exitButton);  
 bottomBox.setStyle("-fx-background-color: #3c6ca8;-fx-background-radius: 20px;-fx-border-color: #439576;-fx-border-radius: 20px;-fx-border-width: 2;-fx-padding: 20,20,20,20");  
 bottomBox.setAlignment(Pos.*CENTER*);  
 bottomBox.setMaxWidth(350);  
 bottomBox.setSpacing(10);  
  
 HBox centerBox = new HBox(10, sideButtons);  
 centerBox.setSpacing(250);  
 centerBox.setAlignment(Pos.*CENTER\_LEFT*);  
  
 HBox rightBox = new HBox(10,centerBox,ImageStack);  
 rightBox.setSpacing(100);  
  
 cardPane.getChildren().addAll(rightBox, bottomBox);  
  
  
 root.setCenter(cardPane);  
 playButton.setOnAction(e->{  
 if(playButton.isHover()){  
 try {  
 new Interest\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 }  
 });  
 exitButton.setOnAction(e->{  
 if (exitButton.isHover()) {  
 try {  
 new exit\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
  
  
// Alert exitAlert = new Alert(Alert.AlertType.CONFIRMATION);  
// exitAlert.setTitle("Exit Confirmation");  
//  
// exitAlert.setHeaderText(null);  
// exitAlert.setContentText("Do you really want to exit?");  
//  
//  
//  
// exitAlert.showAndWait().ifPresent(response -> {  
// if (response == ButtonType.OK) {  
//  
// System.exit(0);  
// } else {  
//  
// exitAlert.close();  
// }  
 //});  
 } });  
 Button1.setOnAction(e->{  
 if(Button1.isHover()){}  
 try {  
 new Achievements\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
 Button2.setOnAction(e->{  
 if(Button2.isHover()){}  
 try {  
 new Edit\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
 Button3.setOnAction(e->{  
 if(Button3.isHover()){}  
 try {  
 new info\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
 Button4.setOnAction(e->{  
 if(Button4.isHover()){}  
 try {  
 new ShopStop\_page().start(new Stage());  
 stage5.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
  
  
 Scene scene = new Scene(root, 800, 600);  
 stage5.initStyle(StageStyle.*UNDECORATED*);  
 // stage5.setFullScreen(true);  
 stage5.setMaximized(true);  
 stage5.setScene(scene);  
 stage5.show();  
  
 }  
  
  
 private void chooseImageFromFile(Stage stage) {  
 FileChooser fileChooser = new FileChooser();  
 fileChooser.setTitle("Choose Profile Image");  
 fileChooser.getExtensionFilters().addAll(  
 new FileChooser.ExtensionFilter("Image Files", "\*.png", "\*.jpg", "\*.jpeg")  
 );  
 File file = fileChooser.showOpenDialog(stage);  
 if (file != null) {  
 try {  
 Image profileImage = new Image(new FileInputStream(file));  
 profileImageView.setFitWidth(150); // Diameter of the circle  
 profileImageView.setFitHeight(150);  
 profileImageView.setPreserveRatio(false);// Maintain aspect ratio  
 profileImageView.setClip(new Circle(75, 75, 75));  
 profileImageView.setImage(profileImage);  
 } catch (FileNotFoundException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
  
 private void openImageSelectionDialog(Stage stage) {  
  
 VBox dialogContent = new VBox(10);  
 dialogContent.setPadding(new Insets(20));  
 dialogContent.setStyle("-fx-background-color: #1b548d; -fx-border-color: #5eb090; -fx-border-width: 2;");  
 dialogContent.setAlignment(Pos.*BASELINE\_CENTER*);  
  
 Label instructions = new Label("Select an avatar:");  
 instructions.setStyle("-fx-font-size: 16px;-fx-text-fill: #ffffff; -fx-font-weight: bold;");  
  
 HBox preloadedImagesBox = new HBox(10);  
 preloadedImagesBox.setAlignment(Pos.*CENTER*);  
 for (Image image : preloadedImages) {  
 ImageView imageView = new ImageView(image);  
 imageView.setFitWidth(60);  
 imageView.setFitHeight(60);  
 StackPane imageContainer = new StackPane(imageView);  
 imageContainer.setStyle("-fx-padding: 5; -fx-border-radius: 10; -fx-background-radius: 10;-fx-background-color: #fff4f4;-fx-border-color:#439576 ");  
  
 imageView.setOnMouseClicked(e -> {  
 profileImageView.setImage(image);  
 animateImageSelection(imageContainer);  
 imageContainer.setStyle("-fx-background-color: #fff99f; -fx-border-radius: 10; -fx-background-radius: 10;"); // Yellow background  
 });  
 preloadedImagesBox.getChildren().add(imageContainer);  
 }  
  
 Button chooseFromFileButton = new Button("Choose from computer");  
 chooseFromFileButton.setStyle("-fx-background-color: #ffde59; -fx-font-size: 14px;");  
 chooseFromFileButton.setOnAction(e -> chooseImageFromFile(stage));  
 addButtonEffects(chooseFromFileButton,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 dialogContent.getChildren().addAll(instructions, preloadedImagesBox, chooseFromFileButton);  
  
  
 Stage dialog = new Stage();  
 dialog.initOwner(stage);  
 dialog.initStyle(StageStyle.*UTILITY*);  
 dialog.setScene(new Scene(dialogContent));  
 dialog.show();  
}  
 private void animateImageSelection(StackPane imageContainer) {  
 // Scale transition on image selection  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), imageContainer);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.2);  
 scaleTransition.setToY(1.2);  
 scaleTransition.setAutoReverse(true);  
 scaleTransition.play();  
  
 // Highlight the selected image with a border  
 imageContainer.setStyle("-fx-border-color: #3c6ca8; -fx-border-width: 2; -fx-border-radius: 10; -fx-background-radius: 10;");  
 }  
  
 private List<Image> loadPreloadedImages() {  
 List<Image> images = new ArrayList<>();  
 try {  
 images.add(new Image(getClass().getResource("/com/example/mind\_marathon\_project/avatar1.png").toExternalForm()));  
 images.add(new Image(getClass().getResource("/com/example/mind\_marathon\_project/avatar2.png").toExternalForm()));  
 images.add(new Image(getClass().getResource("/com/example/mind\_marathon\_project/avatar3.png").toExternalForm()));  
 images.add(new Image(getClass().getResource("/com/example/mind\_marathon\_project/avatar4.png").toExternalForm()));  
 images.add(new Image(getClass().getResource("/com/example/mind\_marathon\_project/avatar5.png").toExternalForm()));  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return images;  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Play sound effect  
 // AudioClip clickSound = new AudioClip(getClass().getResource(soundFile).toExternalForm());  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
 // Play sound on click  
 // button.setOnMouseClicked(e -> clickSound.play());  
}  
}

**Achievement Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.ProgressBar;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.scene.text.Font;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
import javafx.animation.KeyFrame;  
import javafx.animation.KeyValue;  
import javafx.animation.Timeline;  
  
public class Achievements\_page extends Application {  
  
 private int score = 0;  
 private ProgressBar progressBar;  
 private Label scoreLabel;  
 private ImageView medal100, medal500, medal1000;  
  
 private boolean isMedal100Locked = true;  
 private boolean isMedal500Locked = true;  
 private boolean isMedal1000Locked = true;  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
  
 @Override  
 public void start(Stage stage7) throws Exception {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage7);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #fffcf6;");  
 root.setTop(customTitleBar);  
  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #1b548d; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 HBox titleBar = new HBox();  
 titleBar.setAlignment(Pos.*CENTER*);  
 titleBar.setStyle("-fx-background-color: #ffffff;-fx-background-radius: 20px;-fx-border-radius: 20px;-fx-border-color:#ff7bac;-fx-border-width: 2px;-fx-padding: 5,0,5,0");  
 titleBar.setSpacing(10);  
 titleBar.setMaxWidth(300);  
 titleBar.setMaxHeight(90);  
 Label headerLabel = new Label("Achievements");  
 headerLabel.setStyle("-fx-text-fill: #1b548d;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS';-fx-font-size: 24px");  
 headerLabel.setAlignment(Pos.*CENTER*);  
  
 ImageView star;  
 try{  
 star=new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/star.png").toExternalForm()));  
 }  
 catch(NullPointerException e){  
 throw e;  
 }  
 star.setFitWidth(50);  
 star.setFitHeight(50);  
 titleBar.getChildren().addAll(headerLabel, star);  
  
 Region spacer = new Region();  
 VBox.*setVgrow*(spacer, Priority.*ALWAYS*);  
 spacer.setPrefHeight(50);  
 Region spacer2 = new Region();  
 VBox.*setVgrow*(spacer2, Priority.*ALWAYS*);  
 spacer2.setPrefHeight(30);  
 Region spacer3 = new Region();  
 VBox.*setVgrow*(spacer3, Priority.*ALWAYS*);  
 spacer3.setPrefHeight(10);  
  
 Button backButton = new Button();  
  
  
 ImageView arrowImageView;  
 try {  
 arrowImageView = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/arrow.png").toExternalForm()));  
 } catch (NullPointerException e) {  
 throw e;  
 }  
 arrowImageView.setFitHeight(40);  
 arrowImageView.setFitWidth(40);  
 backButton.setGraphic(arrowImageView);  
 addButtonEffects(backButton, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 backButton.setAlignment(Pos.*TOP\_LEFT*);  
 backButton.setStyle("-fx-background-color: #1b548d; -fx-border-color: #1b548d;");  
 backButton.setOnAction(e -> goBackToMenu());  
  
 VBox headerBox = new VBox(backButton);  
 headerBox.setAlignment(Pos.*CENTER\_LEFT*);// Add some spacing after the label for better layout  
 headerBox.setPadding(new Insets(0, 0, 10, 0));  
  
 ImageView medal;  
 try{  
 medal=new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/medal\_pic.png").toExternalForm()));  
 }  
 catch(NullPointerException e){  
 throw e;  
 }  
 medal.setFitHeight(40);  
 medal.setFitWidth(40);  
 progressBar = new ProgressBar(0);  
 progressBar.setPrefWidth(300);  
 progressBar.setStyle( "-fx-accent: #ffdd51; " + // Yellow fill color  
 "-fx-control-inner-background: #ffffff; " + // Transparent background  
 "-fx-background-radius: 20; " + // Rounded corners  
 "-fx-padding: 5; " + // Add padding  
 "-fx-border-radius: 20; " + // Rounded border  
 "-fx-border-color: #ffde59;");  
 scoreLabel = new Label("Score: 0");  
 scoreLabel.setFont(new Font("Comic Sans MS", 20));  
 scoreLabel.setStyle("-fx-text-fill: white;-fx-font-weight: bold");  
  
 try {  
 medal100 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/lock.png").toExternalForm()));  
 medal500 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/lock.png").toExternalForm()));  
 medal1000 = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/lock.png").toExternalForm()));  
 } catch (NullPointerException e) {  
 throw e;  
 }  
 medal100.setFitWidth(80);  
 medal100.setFitHeight(80);  
 medal500.setFitWidth(80);  
 medal500.setFitHeight(82);  
 medal1000.setFitWidth(82);  
 medal1000.setFitHeight(84);  
  
 Label medalLabel100 = new Label("Problem-Solver");  
 Label medalLabel500 = new Label("Skilled Analyst");  
 Label medalLabel1000 = new Label("Sage of Wisdom");  
 medalLabel100.setStyle("-fx-text-fill: #f1f5f6;-fx-font-weight: bold;-fx-font-size: 9px;");  
 medalLabel500.setStyle("-fx-text-fill: #f1f5f6;-fx-font-weight: bold;-fx-font-size: 9px");  
 medalLabel1000.setStyle("-fx-text-fill: #f1f5f6;-fx-font-weight: bold;-fx-font-size: 9px");  
  
 VBox medalBox100 = new VBox(medal100, medalLabel100);  
 VBox medalBox500 = new VBox(medal500, medalLabel500);  
 VBox medalBox1000 = new VBox(medal1000, medalLabel1000);  
  
 medalBox100.setAlignment(Pos.*CENTER*);  
 medalBox500.setAlignment(Pos.*CENTER*);  
 medalBox1000.setAlignment(Pos.*CENTER*);  
  
// Set spacing between the medal image and the label  
 medalBox100.setSpacing(10);  
 medalBox500.setSpacing(10);  
 medalBox1000.setSpacing(10);  
  
 HBox medalRow = new HBox(20, medalBox100, medalBox500, medalBox1000);  
 medalRow.setAlignment(Pos.*CENTER*);  
 medalRow.setSpacing(50);  
  
 cardPane.getChildren().addAll( headerBox,titleBar,spacer,medal,progressBar, scoreLabel,spacer2, medalRow,spacer3);  
 cardPane.setAlignment(Pos.*CENTER*);  
  
 root.setCenter(cardPane);  
 backButton.setOnAction(e->{  
  
 try {  
 new Menu\_page().start(new Stage());  
 stage7.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
  
 });  
  
 Scene scene = new Scene(root, 800, 600);  
 stage7.initStyle(StageStyle.*UNDECORATED*);  
 stage7.setScene(scene);  
 stage7.setMaximized(true);  
 stage7.show();  
 updateScore(500);  
  
  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Play sound effect  
 // AudioClip clickSound = new AudioClip(getClass().getResource(soundFile).toExternalForm());  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
 // Play sound on click  
 // button.setOnMouseClicked(e -> clickSound.play());  
 }  
  
 public void updateScore(int gameResult) {  
 // Increment score based on the game result  
 score += gameResult;  
  
 // Ensure score does not exceed 1000  
 if (score > 1000) {  
 score = 1000;  
 }  
  
 // Update progress bar and medals  
 progressBar.setProgress((double) score / 1000);  
 animateProgressBar((double) score / 1000);  
  
 scoreLabel.setText("Score: " + score);  
  
 // Unlock medals based on the score  
 if (score >= 100 && isMedal100Locked) {  
 medal100.setImage(new Image(getClass().getResource("/com/example/mind\_marathon\_project/100\_score.png").toExternalForm()));  
 isMedal100Locked = false;  
 animateUnlock(medal100);  
 }  
 if (score >= 500 && isMedal500Locked) {  
 medal500.setImage(new Image(getClass().getResource("/com/example/mind\_marathon\_project/500\_score.png").toExternalForm()));  
 isMedal500Locked = false;  
 animateUnlock(medal500);  
 }  
 if (score >= 1000 && isMedal1000Locked) {  
 medal1000.setImage(new Image(getClass().getResource("/com/example/mind\_marathon\_project/1000\_score.png").toExternalForm()));  
 isMedal1000Locked = false;  
 animateUnlock(medal1000);  
 }  
 }  
  
 private void animateUnlock(ImageView medal) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(500), medal);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.2);  
 scaleTransition.setToY(1.2);  
 scaleTransition.setAutoReverse(true);  
 scaleTransition.setCycleCount(2);  
 scaleTransition.play();  
 }  
  
 private void animateProgressBar(double targetProgress) {  
 Timeline timeline = new Timeline(  
 new KeyFrame(Duration.*millis*(500), new KeyValue(progressBar.progressProperty(), targetProgress))  
 );  
 timeline.play();  
 }  
 private void goBackToMenu() {  
 System.*out*.println("Back to Menu");  
 }  
}

**Info Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
  
public class info\_page extends Application {  
 @Override  
 public void start(Stage stage8) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage8);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #1b548d; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 Button backButton = new Button();  
  
 ImageView arrowImageView;  
 try {  
 arrowImageView = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/arrow.png").toExternalForm()));  
 } catch (NullPointerException e) {  
 throw e;  
 }  
 arrowImageView.setFitHeight(40);  
 arrowImageView.setFitWidth(40);  
 backButton.setGraphic(arrowImageView);  
 addButtonEffects(backButton, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 backButton.setAlignment(Pos.*TOP\_LEFT*);  
 backButton.setStyle("-fx-background-color: transparent; -fx-border-color: transparent;");  
 backButton.setOnAction(e -> goBackToMenu());  
  
 VBox headerBox = new VBox(backButton);  
 headerBox.setAlignment(Pos.*CENTER\_LEFT*);// Add some spacing after the label for better layout  
 headerBox.setPadding(new Insets(0, 0, 0, 0));  
  
 Label title= new Label("Info Bank");  
 title.setAlignment(Pos.*CENTER*);  
 title.setStyle("-fx-text-fill: white;-fx-font-size: 30px;-fx-font-family: 'Comic Sans MS';-fx-font-weight: bold;-fx-padding: 20");  
  
  
 ImageView letterBox;  
 try{  
 letterBox = new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/background\_pic.png")));  
 }  
 catch(NullPointerException e){  
 throw e;  
 }  
 letterBox.setFitHeight(350);  
 letterBox.setFitWidth(300);  
  
 cardPane.getChildren().addAll(headerBox,title,letterBox);  
 root.setCenter(cardPane);  
 backButton.setOnAction(e ->{  
 try {  
 new Menu\_page().start(new Stage());  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 stage8.close();  
 });  
 Scene scene = new Scene(root, 800, 600);  
 stage8.initStyle(StageStyle.*UNDECORATED*);  
 stage8.centerOnScreen();  
 stage8.setMaximized(true);  
 stage8.setScene(scene);  
 stage8.show();  
  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Play sound effect  
 // AudioClip clickSound = new AudioClip(getClass().getResource(soundFile).toExternalForm());  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
 // Play sound on click  
 // button.setOnMouseClicked(e -> clickSound.play());  
 }  
 private void goBackToMenu() {  
 System.*out*.println("Back to Menu");  
 }  
}

**Shop stop Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
public class ShopStop\_page extends Application {  
  
 @Override  
 public void start(Stage stage6) throws Exception {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage6);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 Button backButton = new Button();  
  
 ImageView arrowImageView = loadImage("/com/example/mind\_marathon\_project/arrow.png");  
 arrowImageView.setFitHeight(40);  
 arrowImageView.setFitWidth(35);  
 backButton.setGraphic(arrowImageView);  
 addButtonEffects(backButton, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 backButton.setAlignment(Pos.*TOP\_LEFT*);  
 backButton.setStyle("-fx-background-color: #1b548d; -fx-border-color: #1b548d;");  
  
 VBox headerBox = new VBox(backButton);  
 headerBox.setAlignment(Pos.*CENTER\_LEFT*);  
 headerBox.setPadding(new Insets(0, 0, 10, 0));  
  
 HBox titleBar = new HBox();  
 titleBar.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-radius: 20px;-fx-border-color:#ff7bac;-fx-border-width: 2px;-fx-padding: 5,0,5,0");  
 titleBar.setSpacing(10);  
 titleBar.setAlignment(Pos.*CENTER*);  
 titleBar.setMaxWidth(300);  
 titleBar.setMaxHeight(90);  
  
 Label headerLabel = new Label(" Shop Stop");  
 headerLabel.setStyle("-fx-text-fill: #b78fd6;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS';-fx-font-size: 24px");  
 headerLabel.setAlignment(Pos.*CENTER*);  
  
 Region spacer2 = new Region();  
 HBox.*setHgrow*(spacer2, Priority.*ALWAYS*);  
  
 ImageView money = loadImage("/com/example/mind\_marathon\_project/money.png");  
 money.setFitWidth(50);  
 money.setFitHeight(50);  
 titleBar.getChildren().addAll(headerLabel, money);  
  
 Region spacer1 = new Region();  
 HBox.*setHgrow*(spacer1, Priority.*ALWAYS*);  
  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #1b548d; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 0px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 // Create a Label for displaying purchase success message  
 Label purchaseMessage = new Label();  
 purchaseMessage.setStyle("-fx-text-fill: green; -fx-font-weight: bold;");  
 purchaseMessage.setAlignment(Pos.*CENTER*);  
  
 HBox hbox1 = new HBox();  
 hbox1.setAlignment(Pos.*CENTER*);  
 hbox1.setSpacing(20);  
 hbox1.setStyle("-fx-background-color: #1b548d;");  
  
 ImageView hintlogo = loadImage("/com/example/mind\_marathon\_project/hint\_image.png");  
 hintlogo.setFitHeight(110);  
 hintlogo.setFitWidth(150);  
  
 Button buyButtonhint = new Button("BUY");  
 buyButtonhint.setStyle("-fx-background-color: #fffcf6;-fx-border-color:#b78fd6;-fx-border-width: 3px;-fx-border-radius: 18px;-fx-text-fill: #3d3939;-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 10 10");  
 addButtonEffects(buyButtonhint, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 buyButtonhint.setOnAction(e -> {  
 // Show a success message for Hint purchase  
 purchaseMessage.setText("You have successfully purchased a Hint!");  
 });  
  
 hbox1.getChildren().addAll(hintlogo, buyButtonhint);  
  
 HBox hbox2 = new HBox();  
 hbox2.setAlignment(Pos.*CENTER*);  
 hbox2.setSpacing(20);  
 hbox2.setStyle("-fx-background-color: #1b548d;");  
  
 ImageView lifelogo = loadImage("/com/example/mind\_marathon\_project/life\_image.png");  
 lifelogo.setFitHeight(90);  
 lifelogo.setFitWidth(150);  
  
 Button buyButtonlife = new Button("BUY");  
 buyButtonlife.setStyle("-fx-background-color: #fffcf6;-fx-border-color:#b78fd6;-fx-border-width: 3px;-fx-border-radius: 18px;-fx-text-fill: #3d3939;-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 10 10");  
 addButtonEffects(buyButtonlife, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 buyButtonlife.setOnAction(e -> {  
 // Show a success message for Life purchase  
 purchaseMessage.setText("You have successfully purchased a Life!");  
 purchaseMessage.setStyle("-fx-text-fill: White; -fx-font-weight: bold; -fx-font-size: 20px;");  
 });  
  
 hbox2.getChildren().addAll(lifelogo, buyButtonlife);  
  
 VBox vbox = new VBox(1, headerBox, titleBar, hbox1, hbox2, purchaseMessage);  
 vbox.setStyle("-fx-alignment: center; -fx-padding: 1; -fx-background-color: #1b548d;");  
 vbox.setAlignment(Pos.*CENTER*);  
  
 cardPane.getChildren().addAll(vbox);  
  
 root.setCenter(cardPane);  
  
 backButton.setOnAction(e -> {  
 try {  
 new Menu\_page().start(new Stage());  
 stage6.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 });  
  
 Scene scene = new Scene(root, 800, 600);  
 stage6.initStyle(StageStyle.*UNDECORATED*);  
 stage6.setMaximized(true);  
 stage6.setScene(scene);  
 stage6.show();  
 }  
  
 // Helper method to load images  
 private ImageView loadImage(String path) {  
 try {  
 return new ImageView(new Image(getClass().getResourceAsStream(path)));  
 } catch (NullPointerException e) {  
 System.*out*.println("Image not found: " + path);  
 return new ImageView();  
 }  
 }  
  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

**Edit Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.geometry.Insets;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
import javax.swing.\*;  
import java.io.File;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.util.Scanner;  
  
public class Edit\_page extends Application {  
 @Override  
 public void start(Stage stage9) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage9);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #fffcf6;");  
 root.setTop(customTitleBar);  
  
 Button backButton = new Button();  
 ImageView arrowImageView;  
 try {  
 arrowImageView = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/arrow.png").toExternalForm()));  
 } catch (NullPointerException e) {  
 throw e;  
 }  
 arrowImageView.setFitHeight(30);  
 arrowImageView.setFitWidth(30);  
 backButton.setGraphic(arrowImageView);  
 addButtonEffects(backButton, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 backButton.setAlignment(Pos.*TOP\_RIGHT*);  
 backButton.setStyle("-fx-background-color: transparent; -fx-border-color: transparent;");  
 backButton.setOnAction(e -> goBackToMenu());  
  
 Label headerLabel = new Label("Edit Information");  
 headerLabel.setStyle("-fx-background-radius:20px;-fx-background-size: 20px;-fx-background-color: #fff4f4;-fx-border-color: #ff7bac;-fx-border-radius: 20px;-fx-border-width: 2;-fx-font-size: 24px; -fx-text-fill: #1b548d;-fx-font-family:'Comic Sans MS' ;-fx-font-weight: bold;-fx-padding: 10,10,10,10;");  
 headerLabel.setAlignment(Pos.*TOP\_CENTER*);  
  
 HBox headerBox1 = new HBox(headerLabel);  
 headerBox1.setAlignment(Pos.*TOP\_CENTER*);  
 headerBox1.setSpacing(10);  
  
 VBox headerBox = new VBox(backButton);  
 headerBox.setAlignment(Pos.*CENTER\_LEFT*);// Add some spacing after the label for better layout  
 headerBox.setPadding(new Insets(0, 0, 10, 0)); // Space between arrow and label  
 // TextFields for input  
 VBox fields=new VBox();  
 fields.setSpacing(30);  
 fields.setAlignment(Pos.*CENTER*);  
 TextField nameField = new TextField("XYz"); // Default value as seen in the image  
 nameField.setMaxWidth(200);  
 nameField.setStyle("-fx-background-color: #fcfcfc;-fx-background-radius: 10px;-fx-border-width: 2;-fx-border-color: #ff7bac ;-fx-border-radius: 10px; -fx-text-fill: #4a4a4a;-fx-font-family:'Comic Sans MS';-fx-padding: 10,5,10,5 ;");  
  
 TextField ageField = new TextField("Age"); // Default value as seen in the image  
 ageField.setMaxWidth(200);  
 ageField.setStyle("-fx-background-color: #ffffff;-fx-background-radius: 10px;-fx-border-width: 2;-fx-border-color: #ff7bac ; -fx-border-radius: 10px; -fx-text-fill: #4a4a4a;-fx-font-family:'Comic Sans MS' ;-fx-padding: 10,5,10,5 ;");  
  
 TextField passwordField= new TextField("123"); // Default value as seen in the image  
 passwordField.setMaxWidth(200);  
 passwordField.setStyle("-fx-background-color: #fcfcfc;-fx-background-radius: 10px;-fx-border-width: 2;-fx-border-color: #ff7bac ; -fx-border-radius: 10px; -fx-text-fill: #4a4a4a;-fx-font-family:'Comic Sans MS' ;-fx-padding: 10,5,10,5 ;");  
 fields.getChildren().addAll(nameField, ageField, passwordField);  
 // OK Button  
 Button okButton = new Button(" OK ");  
 okButton.setStyle("-fx-background-radius:20px;-fx-background-size: 100px;-fx-background-color: #ffde59;-fx-border-color: #ff7bac;-fx-border-radius: 20px;-fx-border-width: 2;-fx-font-size: 14px; -fx-text-fill: #3d3939;-fx-font-family:'Comic Sans MS' ;-fx-font-weight: bold;-fx-padding: 10,10,10,10;");  
 okButton.setOnAction(e -> handleOkButton(nameField.getText(), ageField.getText(),passwordField.getText()));  
 try {  
 addButtonEffects(okButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 }  
 catch (NullPointerException e) {  
 throw new RuntimeException( e);  
 }  
  
 VBox cardPane = new VBox(40,headerBox,headerBox1,fields, okButton);  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #1b548d; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 root.setCenter(cardPane);  
 backButton.setOnAction(e->{  
 try {  
 new Menu\_page().start(new Stage());  
 stage9.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
  
  
  
 okButton.setOnAction(e -> {  
 String name = nameField.getText();  
 String password= passwordField.getText();  
 String age = ageField.getText();  
  
 if (name.isEmpty() || password.isEmpty() || age.isEmpty()) {  
 System.*out*.println("Name, password, or age is empty");  
 return;  
 }  
  
 boolean isAuthenticated = false;  
 StringBuilder updatedFileContent = new StringBuilder();  
  
 try (Scanner scanner = new Scanner(new File("user\_data.txt"))) {  
 while (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 boolean isUserUpdated = false;  
  
  
 if (line.startsWith("Name: ")) {  
 String storedName = line.substring(6).trim();  
 String storedPassword = "";  
 String storedAge = "";  
  
  
 while (scanner.hasNextLine()) {  
 line = scanner.nextLine();  
 if (line.startsWith("Password: ")) {  
 storedPassword = line.substring(10).trim();  
 } else if (line.startsWith("Age: ")) {  
 storedAge = line.substring(5).trim();  
 }  
  
  
 if (!storedPassword.isEmpty() && !storedAge.isEmpty()) {  
 break;  
 }  
 }  
  
  
 if (name.equals(storedName) && password.equals(storedPassword)) {  
 isAuthenticated = true;  
 updatedFileContent.append("Name: ").append(name).append("\n"); // Keep the name unchanged  
 updatedFileContent.append("Password: ").append(password).append("\n"); // Keep the password unchanged  
 updatedFileContent.append("Age: ").append(age).append("\n"); // Update the age  
 isUserUpdated = true;  
 }  
 }  
  
  
 if (!isUserUpdated) {  
 updatedFileContent.append(line).append("\n");  
 }  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 return;  
 }  
  
  
 if (isAuthenticated) {  
 try (FileWriter writer = new FileWriter("user\_data.txt")) {  
 writer.write(updatedFileContent.toString());  
 System.*out*.println("Successfully logged in and updated details");  
  
 // Show an alert that the login details have been updated successfully  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Update Successful");  
 alert.setHeaderText("Login Details Updated");  
 alert.setContentText("Your login details, have been successfully updated.");  
 alert.showAndWait();  
  
  
 try {  
 new Menu\_page().start(new Stage());  
 stage9.close();  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 } else {  
 System.*out*.println("Failed to login");  
 }  
 });  
  
  
  
 Scene scene = new Scene(root, 800, 600);  
 stage9.initStyle(StageStyle.*UNDECORATED*);  
 stage9.setMaximized(true);  
 stage9.setScene(scene);  
 stage9.show();  
 }  
 private void handleOkButton(String name, String age, String otherInfo) {  
 System.*out*.println("Name: " + name);  
 System.*out*.println("Age: " + age);  
 System.*out*.println("Other Info: " + otherInfo);  
 // Here you can add additional logic to process the input  
 }  
  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
 }  
 private void goBackToMenu() {  
 System.*out*.println("Back to Menu");  
 }  
}

**Art Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.AnimationTimer;  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
public class Art\_page extends Application {  
  
 private Label questionLabel;  
 private List<Button> answerButtons;  
 private Label scoreLabel;  
 private Label timerLabel;  
 private Label lifeLabel;  
 private ImageView hintButton;  
 private Label hintLabel;  
 private Label questionNumberLabel;  
 private int currentScore = 0;  
 private int currentLife = 5;  
 private int questionIndex = 0;  
 private long remainingTime = 30000; // 30 seconds in milliseconds  
  
 private List<Question> questions = new ArrayList<>();  
 private AnimationTimer timer;  
  
 @Override  
 public void start(Stage primaryStage) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(primaryStage);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
 // Initialize questions  
 questions.add(new Question("Which country has kangaroo as national animal?", "Australia", "UK", "USA", "UAE", "Kangaroos are native to Australia."));  
 questions.add(new Question("What is the largest mammal on Earth?", "Blue Whale", "Elephant", "Giraffe", "Lion", "The blue whale is the largest animal ever known to have lived on Earth."));  
 questions.add(new Question("What is the chemical symbol for gold?", "Au", "Ag", "Fe", "Cu", "The chemical symbol for gold is Au."));  
 questions.add(new Question("Which planet is known as the Red Planet?", "Mars", "Venus", "Jupiter", "Saturn", "Mars is often called the 'Red Planet' because of its reddish appearance."));  
 questions.add(new Question("Who painted the Mona Lisa?", "Leonardo da Vinci", "Michelangelo", "Raphael", "Donatello", "Leonardo da Vinci painted the Mona Lisa."));  
  
 Collections.*shuffle*(questions);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #439576; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #1b548d; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(700);  
 cardPane.setMaxHeight(500);  
  
 // Top bar  
 Region spacer1 = new Region();  
 HBox.*setHgrow*(spacer1, Priority.*ALWAYS*);  
 spacer1.setMaxSize(150,150);  
 Region spacer2 = new Region();  
 HBox.*setHgrow*(spacer2, Priority.*ALWAYS*);  
 spacer2.setMaxSize(150,150);  
  
 HBox topBar = new HBox();  
 topBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 20,30,20,30");  
 topBar.setAlignment(Pos.*CENTER*);  
  
 ImageView coinImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/coins\_button.png").toString()));  
 coinImage.setFitWidth(30);  
 coinImage.setFitHeight(30);  
 scoreLabel = new Label(" 0");  
 scoreLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView timerImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/clock\_button.png").toString()));  
 timerImage.setFitWidth(30);  
 timerImage.setFitHeight(30);  
 timerLabel = new Label(" 30");  
 timerLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView lifeImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/life\_button.png").toString()));  
 lifeImage.setFitWidth(30);  
 lifeImage.setFitHeight(30);  
 lifeLabel = new Label(" 5");  
 lifeLabel.setStyle("-fx-font-size: 18px;");  
  
 topBar.getChildren().addAll(coinImage, scoreLabel, spacer1,timerImage, timerLabel,spacer2, lifeImage, lifeLabel);  
  
  
 // Center content  
 VBox center = new VBox(20);  
 center.setAlignment(Pos.*CENTER*);  
  
 questionLabel = new Label(questions.get(questionIndex).getQuestion());  
 questionLabel.setMaxWidth(600);  
 questionLabel.setMaxHeight(400);  
 questionLabel.setStyle("-fx-font-size: 18px;-fx-background-color: #ecf1ef;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 20px;-fx-background-radius: 20px;-fx-padding: 20;-fx-text-alignment: center");  
 center.getChildren().add(questionLabel);  
  
 answerButtons = new ArrayList<>();  
 HBox row1 = new HBox();  
 row1.setAlignment(Pos.*CENTER*);  
 row1.setSpacing(20);  
 HBox row2 = new HBox();  
 row2.setAlignment(Pos.*CENTER*);  
 row2.setSpacing(20);  
  
 Button button1 = new Button();  
 button1.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button1.setOnAction(event -> checkAnswer(button1));  
 button1.setPrefSize(150,50);  
 answerButtons.add(button1);  
  
 Button button2 = new Button();  
 button2.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d;-fx-background-radius: 20px; -fx-padding: 10px 20px;");  
 button2.setOnAction(event -> checkAnswer(button2));  
 button2.setPrefSize(150,50);  
 answerButtons.add(button2);  
 row1.getChildren().addAll(button1,button2);  
  
 Button button3 = new Button();  
 button3.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button3.setOnAction(event -> checkAnswer(button3));  
 button3.setPrefSize(150,50);  
 answerButtons.add(button3);  
  
 Button button4 = new Button();  
 button4.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button4.setOnAction(event -> checkAnswer(button4));  
 button4.setPrefSize(150,50);  
 answerButtons.add(button4);  
 row2.getChildren().addAll(button3,button4);  
 for(int i=0;i<4;i++){  
 Button button=answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 }  
 VBox layout = new VBox();  
 layout.setAlignment(Pos.*CENTER*);  
 layout.setPrefHeight(200);  
 // layout.setPrefWidth(200);  
 layout.setSpacing(15);  
 layout.getChildren().addAll(row1,row2);  
  
 // Bottom bar  
 HBox bottomBar = new HBox(20);  
 bottomBar.setAlignment(Pos.*CENTER*);  
 bottomBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 10,30,10,30");  
 // topBar.setSpacing(80);  
 bottomBar.setAlignment(Pos.*CENTER*);  
  
 hintButton = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/hint\_button.png").toExternalForm()));  
 Button hint=new Button();  
 hintButton.setFitWidth(30);  
 hint.setStyle("-fx-background-color: #1b548d;-fx-border-color: #ffe47a;-fx-border-radius: 20px;-fx-background-radius: 20px;");  
 hintButton.setFitHeight(30);  
 hint.setGraphic(hintButton);  
 hint.setOnMouseClicked(event -> showHint());  
  
 questionNumberLabel = new Label("< 1 /" + questions.size() + " >");  
 questionNumberLabel.setStyle("-fx-font-size: 18px;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 bottomBar.getChildren().add( questionNumberLabel);  
 root.setBottom(bottomBar);  
  
 VBox all\_labels=new VBox(10);  
 all\_labels.setAlignment(Pos.*CENTER*);  
 all\_labels.getChildren().addAll(topBar,cardPane,bottomBar);  
  
 // Hint label  
 hintLabel = new Label();  
 hintLabel.setStyle("-fx-background-color: #ffffa5; -fx-text-fill: black; -fx-padding: 5px; -fx-border-color: black;");  
 hintLabel.setVisible(false);  
 center.getChildren().add(hintLabel);  
 cardPane.getChildren().addAll(center,layout,hint);  
 root.setCenter(all\_labels);  
 // Timer  
 startTimer();  
  
 Scene scene = new Scene(root, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.initStyle(StageStyle.*UNDECORATED*);  
 primaryStage.setMaximized(true);  
 primaryStage.show();  
  
 }  
  
 private void startTimer() {  
 timer = new AnimationTimer() {  
 private long lastUpdate = System.*nanoTime*();  
  
 @Override  
 public void handle(long now) {  
 long elapsedTime = now - lastUpdate;  
 lastUpdate = now;  
 remainingTime -= elapsedTime / 1\_000\_000; // Convert to milliseconds  
  
 if (remainingTime <= 0) {  
 this.stop();  
 handleTimeUp();  
 }  
  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 }  
 };  
 timer.start();  
 }  
  
 private void checkAnswer(Button button) {  
 if (button.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 button.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 currentScore += 1;  
 scoreLabel.setText(String.*valueOf*(currentScore));  
 } else {  
 button.setStyle("-fx-background-color: red; -fx-text-fill: white;");  
 currentLife--;  
 for (Button cbutton : answerButtons) {  
 if (cbutton.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 cbutton.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 break;  
 }  
 } lifeLabel.setText(String.*valueOf*(currentLife));  
 if (currentLife == 0) {  
 //add game over or result page  
 return;  
 }  
 }  
  
 answerButtons.forEach(btn -> btn.setDisable(true));  
  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> nextQuestion());  
 }  
 },  
 1000  
 );  
 }  
  
 private void nextQuestion() {  
 if (questionIndex < questions.size() - 1) {  
 questionIndex++;  
 questionLabel.setText(questions.get(questionIndex).getQuestion());  
 for (int i = 0; i < 4; i++) {  
 Button button = answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 button.setDisable(false);  
 button.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 10px; -fx-padding: 10px 20px;-fx-text-fill: black" +  
 "");  
 }  
 questionNumberLabel.setText("< " + (questionIndex + 1) + "/" + questions.size() + " >");  
 remainingTime = 30000;  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 } else {  
 System.*out*.println("No more questions");  
 //add game over or result page  
 }  
 }  
  
 private void handleTimeUp() {  
 System.*out*.println("Time's up!");  
 //add game over or result page  
 }  
  
  
 private void showHint() {  
 if (currentLife > 0) {  
 currentLife--;  
 lifeLabel.setText(String.*valueOf*(currentLife));  
 hintLabel.setText(questions.get(questionIndex).getHint());  
 hintLabel.setVisible(true);  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> hintLabel.setVisible(false));  
 }  
 },  
 3000  
 );  
 }  
 }  
  
 private static class Question {  
 private final String question;  
 private final String correctAnswer;  
 private final List<String> options;  
 private final String hint;  
  
 public Question(String question, String correctAnswer, String option1, String option2, String option3, String hint) {  
 this.question = question;  
 this.correctAnswer = correctAnswer;  
 this.options = new ArrayList<>();  
 this.options.add(option1);  
 this.options.add(option2);  
 this.options.add(option3);  
 this.options.add(correctAnswer);  
 Collections.*shuffle*(this.options);  
 this.hint = hint;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getCorrectAnswer() {  
 return correctAnswer;  
 }  
  
 public List<String> getOptions() {  
 return options;  
 }  
  
 public String getHint() {  
 return hint;  
 }  
 }  
}

**Science Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.AnimationTimer;  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Alert;  
import javafx.scene.control.Button;  
import javafx.scene.control.ButtonType;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
import java.util.Optional;  
  
import static java.math.RoundingMode.*UP*;  
import static javax.management.Query.*TIMES*;  
  
public class Science\_page extends Application {  
  
 private Label questionLabel;  
 private List<Button> answerButtons;  
 private Label scoreLabel;  
 private Label timerLabel;  
 private Label lifeLabel;  
 private ImageView hintButton;  
 private Label hintLabel;  
 private Label questionNumberLabel;  
 private int currentScore = 0;  
 private int currentLife = 5;  
 private int questionIndex = 0;  
 private long remainingTime = 30000; // 30 seconds in milliseconds  
  
 private List<Question> questions = new ArrayList<>();  
 private AnimationTimer timer;  
  
 @Override  
 public void start(Stage primaryStage) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(primaryStage);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 questions.add(new Question("What is the capital of France?", "Paris", "Berlin", "Madrid", "Rome", "The capital of France is Paris."));  
 questions.add(new Question("What is the process by which plants make their food?", "Photosynthesis", "Respiration", "Digestion", "Fermentation", "Plants make their food through the process of photosynthesis."));  
 questions.add(new Question("Who wrote 'Romeo and Juliet'?", "William Shakespeare", "Charles Dickens", "Mark Twain", "Jane Austen", "William Shakespeare wrote 'Romeo and Juliet'."));  
 questions.add(new Question("What is the boiling point of water at sea level?", "100°C", "90°C", "80°C", "70°C", "The boiling point of water at sea level is 100°C."));  
 questions.add(new Question("Which organ is responsible for pumping blood in the human body?", "Heart", "Lungs", "Liver", "Kidneys", "The heart is responsible for pumping blood in the human body."));  
 questions.add(new Question("What is the smallest prime number?", "2", "1", "3", "5", "The smallest prime number is 2."));  
 questions.add(new Question("Which ocean is the largest on Earth?", "Pacific Ocean", "Atlantic Ocean", "Indian Ocean", "Arctic Ocean", "The Pacific Ocean is the largest ocean on Earth."));  
 questions.add(new Question("What gas do humans exhale?", "Carbon dioxide", "Oxygen", "Nitrogen", "Helium", "Humans exhale carbon dioxide."));  
 questions.add(new Question("Which element is represented by the symbol 'O'?", "Oxygen", "Osmium", "Gold", "Silver", "The symbol 'O' represents oxygen."));  
 questions.add(new Question("Who discovered gravity?", "Isaac Newton", "Albert Einstein", "Galileo Galilei", "Nikola Tesla", "Isaac Newton discovered gravity."));  
 questions.add(new Question("Which continent is the Sahara Desert located on?", "Africa", "Asia", "South America", "Australia", "The Sahara Desert is located in Africa."));  
 questions.add(new Question("How many continents are there on Earth?", "7", "5", "6", "8", "There are 7 continents on Earth."));  
 questions.add(new Question("Which bird is known for its colorful plumage and mimicry?", "Parrot", "Eagle", "Penguin", "Sparrow", "Parrots are known for their colorful plumage and mimicry."));  
 questions.add(new Question("What is the hardest natural substance on Earth?", "Diamond", "Gold", "Iron", "Platinum", "Diamond is the hardest natural substance on Earth."));  
 questions.add(new Question("Which animal is known as the 'Ship of the Desert'?", "Camel", "Horse", "Elephant", "Donkey", "The camel is known as the 'Ship of the Desert'."));  
  
  
 Collections.*shuffle*(questions);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #439576; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #1b548d; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(700);  
 cardPane.setMaxHeight(500);  
  
 // Top bar  
 Region spacer1 = new Region();  
 HBox.*setHgrow*(spacer1, Priority.*ALWAYS*);  
 spacer1.setMaxSize(150,150);  
 Region spacer2 = new Region();  
 HBox.*setHgrow*(spacer2, Priority.*ALWAYS*);  
 spacer2.setMaxSize(150,150);  
  
 HBox topBar = new HBox();  
 topBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 20,30,20,30");  
 topBar.setAlignment(Pos.*CENTER*);  
  
 ImageView coinImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/coins\_button.png").toString()));  
 coinImage.setFitWidth(30);  
 coinImage.setFitHeight(30);  
 scoreLabel = new Label(" 0");  
 scoreLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView timerImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/clock\_button.png").toString()));  
 timerImage.setFitWidth(30);  
 timerImage.setFitHeight(30);  
 timerLabel = new Label(" 30");  
 timerLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView lifeImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/life\_button.png").toString()));  
 lifeImage.setFitWidth(30);  
 lifeImage.setFitHeight(30);  
 lifeLabel = new Label(" 5");  
 lifeLabel.setStyle("-fx-font-size: 18px;");  
  
 topBar.getChildren().addAll(coinImage, scoreLabel, spacer1,timerImage, timerLabel,spacer2, lifeImage, lifeLabel);  
  
  
 // Center content  
 VBox center = new VBox(20);  
 center.setAlignment(Pos.*CENTER*);  
  
 questionLabel = new Label(questions.get(questionIndex).getQuestion());  
 questionLabel.setMaxWidth(600);  
 questionLabel.setMaxHeight(400);  
 questionLabel.setStyle("-fx-font-size: 18px;-fx-background-color: #ecf1ef;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 20px;-fx-background-radius: 20px;-fx-padding: 20;-fx-text-alignment: center");  
 center.getChildren().add(questionLabel);  
  
 answerButtons = new ArrayList<>();  
 HBox row1 = new HBox();  
 row1.setAlignment(Pos.*CENTER*);  
 row1.setSpacing(20);  
 HBox row2 = new HBox();  
 row2.setAlignment(Pos.*CENTER*);  
 row2.setSpacing(20);  
  
 Button button1 = new Button();  
 button1.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button1.setOnAction(event -> checkAnswer(button1));  
 addButtonEffects(button1,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 button1.setPrefSize(150,50);  
 answerButtons.add(button1);  
  
 Button button2 = new Button();  
 button2.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d;-fx-background-radius: 20px; -fx-padding: 10px 20px;");  
 button2.setOnAction(event -> checkAnswer(button2));  
 button2.setPrefSize(150,50);  
 addButtonEffects(button2,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 answerButtons.add(button2);  
 row1.getChildren().addAll(button1,button2);  
  
 Button button3 = new Button();  
 button3.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button3.setOnAction(event -> checkAnswer(button3));  
 addButtonEffects(button3,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 button3.setPrefSize(150,50);  
 answerButtons.add(button3);  
  
 Button button4 = new Button();  
 button4.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button4.setOnAction(event -> checkAnswer(button4));  
 addButtonEffects(button4,"/com/example/mind\_marathon\_project/click\_sound.mp3");  
 button4.setPrefSize(150,50);  
  
 answerButtons.add(button4);  
 for(int i=0;i<4;i++){  
 Button button=answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 }  
 row2.getChildren().addAll(button3,button4);  
 VBox layout = new VBox();  
 layout.setAlignment(Pos.*CENTER*);  
 layout.setPrefHeight(200);  
 // layout.setPrefWidth(200);  
 layout.setSpacing(15);  
 layout.getChildren().addAll(row1,row2);  
  
 // Bottom bar  
 HBox bottomBar = new HBox(20);  
 bottomBar.setAlignment(Pos.*CENTER*);  
 bottomBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 10,30,10,30");  
 // topBar.setSpacing(80);  
 Region spacer = new Region();  
 HBox.*setHgrow*(spacer, Priority.*ALWAYS*);  
 Region spacer4 = new Region();  
 HBox.*setHgrow*(spacer4, Priority.*ALWAYS*);  
 Button backButton = new Button();  
 ImageView arrowImageView;  
 try {  
 arrowImageView = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/arrow.png").toExternalForm()));  
 } catch (NullPointerException e) {  
 throw e;  
 }  
 arrowImageView.setFitHeight(40);  
 arrowImageView.setFitWidth(40);  
 backButton.setGraphic(arrowImageView);  
 backButton.setStyle("-fx-background-color: #1b548d;");  
 addButtonEffects(backButton, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 backButton.setAlignment(Pos.*TOP\_LEFT*);  
 // backButton.setOnAction(e -> goBackToMenu());  
  
 VBox headerBox = new VBox(backButton);  
 headerBox.setAlignment(Pos.*BOTTOM\_RIGHT*);// Add some spacing after the label for better layout  
 headerBox.setPadding(new Insets(0, 0, 10, 0));  
  
  
 hintButton = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/hint\_button.png").toExternalForm()));  
 Button hint=new Button();  
 hintButton.setFitWidth(30);  
 hint.setStyle("-fx-background-color: #1b548d;-fx-border-color: #ffe47a;-fx-border-radius: 20px;-fx-background-radius: 20px;");  
 hintButton.setFitHeight(30);  
 hint.setGraphic(hintButton);  
 addButtonEffects(hint, "/com/example/mind\_marathon\_project/click\_sound.mp3");  
 hint.setOnMouseClicked(event -> showHint());  
  
 questionNumberLabel = new Label("< 1 /" + questions.size() + " >");  
 questionNumberLabel.setStyle("-fx-font-size: 18px;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 bottomBar.getChildren().addAll(headerBox,spacer, questionNumberLabel,spacer4);  
 root.setBottom(bottomBar);  
  
 VBox all\_labels=new VBox(10);  
 all\_labels.setAlignment(Pos.*CENTER*);  
 all\_labels.getChildren().addAll(topBar,cardPane,bottomBar);  
  
 // Hint label  
 hintLabel = new Label();  
 hintLabel.setStyle("-fx-background-color: #ffffa5; -fx-text-fill: black; -fx-padding: 5px; -fx-border-color: black;");  
 hintLabel.setVisible(false);  
 center.getChildren().add(hintLabel);  
 cardPane.getChildren().addAll(center,layout,hint);  
 root.setCenter(all\_labels);  
 // Timer  
 startTimer();  
 backButton.setOnAction(e->{  
 try {  
 new Menu\_page().start(new Stage());  
 primaryStage.close();  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
  
  
 Scene scene = new Scene(root, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.initStyle(StageStyle.*UNDECORATED*);  
 primaryStage.setMaximized(true);  
 primaryStage.show();  
 }  
  
 private void startTimer() {  
 timer = new AnimationTimer() {  
 private long lastUpdate = System.*nanoTime*();  
  
 @Override  
 public void handle(long now) {  
 long elapsedTime = now - lastUpdate;  
 lastUpdate = now;  
 remainingTime -= elapsedTime / 1\_000\_000; // Convert to milliseconds  
  
 if (remainingTime <= 0) {  
 this.stop();  
 try {  
 handleTimeUp();  
 } catch (Exception e) {  
 throw new RuntimeException(e);  
 }  
 }  
  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 }  
 };  
 timer.start();  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Play sound effect  
 // AudioClip clickSound = new AudioClip(getClass().getResource(soundFile).toExternalForm());  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
 // Play sound on click  
 // button.setOnMouseClicked(e -> clickSound.play());  
 }  
 private void checkAnswer(Button button) {  
 if (button.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 button.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 currentScore += 1;  
 scoreLabel.setText(String.*valueOf*(currentScore));  
 } else {  
 button.setStyle("-fx-background-color: red; -fx-text-fill: white;");  
 currentLife--;  
 for (Button cbutton : answerButtons) {  
 if (cbutton.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 cbutton.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 break;  
 }  
 } lifeLabel.setText(String.*valueOf*(currentLife));  
 if (currentLife == 0) {  
 //add game over or result page  
 return;  
 }  
 }  
  
 answerButtons.forEach(btn -> btn.setDisable(true));  
  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> nextQuestion());  
 }  
 },  
 1000  
 );  
 }  
  
 private void nextQuestion() {  
 if (questionIndex < questions.size() - 1) {  
 questionIndex++;  
 questionLabel.setText(questions.get(questionIndex).getQuestion());  
 for (int i = 0; i < 4; i++) {  
 Button button = answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 button.setDisable(false);  
 button.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 10px; -fx-padding: 10px 20px;-fx-text-fill: black" +  
 "");  
 }  
 questionNumberLabel.setText("< " + (questionIndex + 1) + "/" + questions.size() + " >");  
 remainingTime = 30000;  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 } else {  
 System.*out*.println("No more questions");  
 //add game over or result page  
 }  
 }  
  
 private void handleTimeUp() throws Exception {  
 Alert showAlert = new Alert(Alert.AlertType.*INFORMATION*);  
  
 System.*out*.println("Time's up!");  
 // new Result\_page().start(new Stage());  
  
 }  
  
  
 private void showHint() {  
 if (currentLife > 0) {  
 currentLife--;  
 lifeLabel.setText(String.*valueOf*(currentLife));  
 hintLabel.setText(questions.get(questionIndex).getHint());  
 hintLabel.setVisible(true);  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> hintLabel.setVisible(false));  
 }  
 },  
 3000  
 );  
 }  
 }  
  
 private static class Question {  
 private final String question;  
 private final String correctAnswer;  
 private final List<String> options;  
 private final String hint;  
  
 public Question(String question, String correctAnswer, String option1, String option2, String option3, String hint) {  
 this.question = question;  
 this.correctAnswer = correctAnswer;  
 this.options = new ArrayList<>();  
 this.options.add(option1);  
 this.options.add(option2);  
 this.options.add(option3);  
 this.options.add(correctAnswer);  
 Collections.*shuffle*(this.options);  
 this.hint = hint;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getCorrectAnswer() {  
 return correctAnswer;  
 }  
  
 public List<String> getOptions() {  
 return options;  
 }  
  
 public String getHint() {  
 return hint;  
 }  
 }  
  
}

**History Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.AnimationTimer;  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
public class History\_page extends Application {  
  
 private Label questionLabel;  
 private List<Button> answerButtons;  
 private Label scoreLabel;  
 private Label timerLabel;  
 private Label lifeLabel;  
 private ImageView hintButton;  
 private Label hintLabel;  
 private Label questionNumberLabel;  
 private int currentScore = 0;  
 private int currentLife = 5;  
 private int questionIndex = 0;  
 private long remainingTime = 30000; // 30 seconds in milliseconds  
  
 private List<Question> questions = new ArrayList<>();  
 private AnimationTimer timer;  
  
 @Override  
 public void start(Stage primaryStage) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(primaryStage);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 questions.add(new Question("What is the capital of France?", "Paris", "Berlin", "Madrid", "Rome", "The capital of France is Paris."));  
 questions.add(new Question("What is the process by which plants make their food?", "Photosynthesis", "Respiration", "Digestion", "Fermentation", "Plants make their food through the process of photosynthesis."));  
 questions.add(new Question("Who wrote 'Romeo and Juliet'?", "William Shakespeare", "Charles Dickens", "Mark Twain", "Jane Austen", "William Shakespeare wrote 'Romeo and Juliet'."));  
 questions.add(new Question("What is the boiling point of water at sea level?", "100°C", "90°C", "80°C", "70°C", "The boiling point of water at sea level is 100°C."));  
 questions.add(new Question("Which organ is responsible for pumping blood in the human body?", "Heart", "Lungs", "Liver", "Kidneys", "The heart is responsible for pumping blood in the human body."));  
 questions.add(new Question("What is the smallest prime number?", "2", "1", "3", "5", "The smallest prime number is 2."));  
 questions.add(new Question("Which ocean is the largest on Earth?", "Pacific Ocean", "Atlantic Ocean", "Indian Ocean", "Arctic Ocean", "The Pacific Ocean is the largest ocean on Earth."));  
 questions.add(new Question("What gas do humans exhale?", "Carbon dioxide", "Oxygen", "Nitrogen", "Helium", "Humans exhale carbon dioxide."));  
 questions.add(new Question("Which element is represented by the symbol 'O'?", "Oxygen", "Osmium", "Gold", "Silver", "The symbol 'O' represents oxygen."));  
 questions.add(new Question("Who discovered gravity?", "Isaac Newton", "Albert Einstein", "Galileo Galilei", "Nikola Tesla", "Isaac Newton discovered gravity."));  
 questions.add(new Question("Which continent is the Sahara Desert located on?", "Africa", "Asia", "South America", "Australia", "The Sahara Desert is located in Africa."));  
 questions.add(new Question("How many continents are there on Earth?", "7", "5", "6", "8", "There are 7 continents on Earth."));  
 questions.add(new Question("Which bird is known for its colorful plumage and mimicry?", "Parrot", "Eagle", "Penguin", "Sparrow", "Parrots are known for their colorful plumage and mimicry."));  
 questions.add(new Question("What is the hardest natural substance on Earth?", "Diamond", "Gold", "Iron", "Platinum", "Diamond is the hardest natural substance on Earth."));  
 questions.add(new Question("Which animal is known as the 'Ship of the Desert'?", "Camel", "Horse", "Elephant", "Donkey", "The camel is known as the 'Ship of the Desert'."));  
  
  
 Collections.*shuffle*(questions);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #439576; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #1b548d; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(700);  
 cardPane.setMaxHeight(500);  
  
  
 Region spacer1 = new Region();  
 HBox.*setHgrow*(spacer1, Priority.*ALWAYS*);  
 spacer1.setMaxSize(150,150);  
 Region spacer2 = new Region();  
 HBox.*setHgrow*(spacer2, Priority.*ALWAYS*);  
 spacer2.setMaxSize(150,150);  
  
 HBox topBar = new HBox();  
 topBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 20,30,20,30");  
 topBar.setAlignment(Pos.*CENTER*);  
  
 ImageView coinImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/coins\_button.png").toString()));  
 coinImage.setFitWidth(30);  
 coinImage.setFitHeight(30);  
 scoreLabel = new Label(" 0");  
 scoreLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView timerImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/clock\_button.png").toString()));  
 timerImage.setFitWidth(30);  
 timerImage.setFitHeight(30);  
 timerLabel = new Label(" 30");  
 timerLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView lifeImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/life\_button.png").toString()));  
 lifeImage.setFitWidth(30);  
 lifeImage.setFitHeight(30);  
 lifeLabel = new Label(" 5");  
 lifeLabel.setStyle("-fx-font-size: 18px;");  
  
 topBar.getChildren().addAll(coinImage, scoreLabel, spacer1,timerImage, timerLabel,spacer2, lifeImage, lifeLabel);  
  
  
 // Center content  
 VBox center = new VBox(20);  
 center.setAlignment(Pos.*CENTER*);  
  
 questionLabel = new Label(questions.get(questionIndex).getQuestion());  
 questionLabel.setMaxWidth(600);  
 questionLabel.setMaxHeight(400);  
 questionLabel.setStyle("-fx-font-size: 18px;-fx-background-color: #ecf1ef;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 20px;-fx-background-radius: 20px;-fx-padding: 20;-fx-text-alignment: center");  
 center.getChildren().add(questionLabel);  
  
 answerButtons = new ArrayList<>();  
 HBox row1 = new HBox();  
 row1.setAlignment(Pos.*CENTER*);  
 row1.setSpacing(20);  
 HBox row2 = new HBox();  
 row2.setAlignment(Pos.*CENTER*);  
 row2.setSpacing(20);  
  
 Button button1 = new Button();  
 button1.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button1.setOnAction(event -> checkAnswer(button1));  
 button1.setPrefSize(150,50);  
 answerButtons.add(button1);  
  
 Button button2 = new Button();  
 button2.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d;-fx-background-radius: 20px; -fx-padding: 10px 20px;");  
 button2.setOnAction(event -> checkAnswer(button2));  
 button2.setPrefSize(150,50);  
 answerButtons.add(button2);  
 row1.getChildren().addAll(button1,button2);  
  
 Button button3 = new Button();  
 button3.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button3.setOnAction(event -> checkAnswer(button3));  
 button3.setPrefSize(150,50);  
 answerButtons.add(button3);  
  
 Button button4 = new Button();  
 button4.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button4.setOnAction(event -> checkAnswer(button4));  
 button4.setPrefSize(150,50);  
 answerButtons.add(button4);  
 for(int i=0;i<4;i++){  
 Button button=answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 }  
 row2.getChildren().addAll(button3,button4);  
 VBox layout = new VBox();  
 layout.setAlignment(Pos.*CENTER*);  
 layout.setPrefHeight(200);  
 // layout.setPrefWidth(200);  
 layout.setSpacing(15);  
 layout.getChildren().addAll(row1,row2);  
  
 // Bottom bar  
 HBox bottomBar = new HBox(20);  
 bottomBar.setAlignment(Pos.*CENTER*);  
 bottomBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 10,30,10,30");  
 // topBar.setSpacing(80);  
 bottomBar.setAlignment(Pos.*CENTER*);  
  
 hintButton = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/hint\_button.png").toExternalForm()));  
 Button hint=new Button();  
 hintButton.setFitWidth(30);  
 hint.setStyle("-fx-background-color: #1b548d;-fx-border-color: #ffe47a;-fx-border-radius: 20px;-fx-background-radius: 20px;");  
 hintButton.setFitHeight(30);  
 hint.setGraphic(hintButton);  
 hint.setOnMouseClicked(event -> showHint());  
  
 questionNumberLabel = new Label("< 1 /" + questions.size() + " >");  
 questionNumberLabel.setStyle("-fx-font-size: 18px;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 bottomBar.getChildren().add( questionNumberLabel);  
 root.setBottom(bottomBar);  
  
 VBox all\_labels=new VBox(10);  
 all\_labels.setAlignment(Pos.*CENTER*);  
 all\_labels.getChildren().addAll(topBar,cardPane,bottomBar);  
  
 // Hint label  
 hintLabel = new Label();  
 hintLabel.setStyle("-fx-background-color: #ffffa5; -fx-text-fill: black; -fx-padding: 5px; -fx-border-color: black;");  
 hintLabel.setVisible(false);  
 center.getChildren().add(hintLabel);  
 cardPane.getChildren().addAll(center,layout,hint);  
 root.setCenter(all\_labels);  
 // Timer  
 startTimer();  
  
 Scene scene = new Scene(root, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.initStyle(StageStyle.*UNDECORATED*);  
 primaryStage.setMaximized(true);  
 primaryStage.show();  
  
 }  
  
 private void startTimer() {  
 timer = new AnimationTimer() {  
 private long lastUpdate = System.*nanoTime*();  
  
 @Override  
 public void handle(long now) {  
 long elapsedTime = now - lastUpdate;  
 lastUpdate = now;  
 remainingTime -= elapsedTime / 1\_000\_000; // Convert to milliseconds  
  
 if (remainingTime <= 0) {  
 this.stop();  
 handleTimeUp();  
 }  
  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 }  
 };  
 timer.start();  
 }  
  
 private void checkAnswer(Button button) {  
 if (button.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 button.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 currentScore += 1;  
 scoreLabel.setText(String.*valueOf*(currentScore));  
 } else {  
 button.setStyle("-fx-background-color: red; -fx-text-fill: white;");  
 currentLife--;  
 for (Button cbutton : answerButtons) {  
 if (cbutton.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 cbutton.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 break;  
 }  
 } lifeLabel.setText(String.*valueOf*(currentLife));  
 if (currentLife == 0) {  
 //add game over or result page  
 return;  
 }  
 }  
  
 answerButtons.forEach(btn -> btn.setDisable(true));  
  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> nextQuestion());  
 }  
 },  
 1000  
 );  
 }  
  
 private void nextQuestion() {  
 if (questionIndex < questions.size() - 1) {  
 questionIndex++;  
 questionLabel.setText(questions.get(questionIndex).getQuestion());  
 for (int i = 0; i < 4; i++) {  
 Button button = answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 button.setDisable(false);  
 button.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 10px; -fx-padding: 10px 20px;-fx-text-fill: black" +  
 "");  
 }  
 questionNumberLabel.setText("< " + (questionIndex + 1) + "/" + questions.size() + " >");  
 remainingTime = 30000;  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 } else {  
 System.*out*.println("No more questions");  
 //add game over or result page  
 }  
 }  
  
 private void handleTimeUp() {  
 System.*out*.println("Time's up!");  
 //add game over or result page  
 }  
  
  
 private void showHint() {  
 if (currentLife > 0) {  
 currentLife--;  
 lifeLabel.setText(String.*valueOf*(currentLife));  
 hintLabel.setText(questions.get(questionIndex).getHint());  
 hintLabel.setVisible(true);  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> hintLabel.setVisible(false));  
 }  
 },  
 3000  
 );  
 }  
 }  
  
 private static class Question {  
 private final String question;  
 private final String correctAnswer;  
 private final List<String> options;  
 private final String hint;  
  
 public Question(String question, String correctAnswer, String option1, String option2, String option3, String hint) {  
 this.question = question;  
 this.correctAnswer = correctAnswer;  
 this.options = new ArrayList<>();  
 this.options.add(option1);  
 this.options.add(option2);  
 this.options.add(option3);  
 this.options.add(correctAnswer);  
 Collections.*shuffle*(this.options);  
 this.hint = hint;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getCorrectAnswer() {  
 return correctAnswer;  
 }  
  
 public List<String> getOptions() {  
 return options;  
 }  
  
 public String getHint() {  
 return hint;  
 }  
 }  
}

**Geography Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.AnimationTimer;  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
public class Geography\_page extends Application {  
  
 private Label questionLabel;  
 private List<Button> answerButtons;  
 private Label scoreLabel;  
 private Label timerLabel;  
 private Label lifeLabel;  
 private ImageView hintButton;  
 private Label hintLabel;  
 private Label questionNumberLabel;  
 private int currentScore = 0;  
 private int currentLife = 5;  
 private int questionIndex = 0;  
 private long remainingTime = 30000; // 30 seconds in milliseconds  
  
 private List<Question> questions = new ArrayList<>();  
 private AnimationTimer timer;  
  
 @Override  
 public void start(Stage primaryStage) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(primaryStage);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
 // Initialize questions  
 questions.add(new Question("Which country has kangaroo as national animal?", "Australia", "UK", "USA", "UAE", "Kangaroos are native to Australia."));  
 questions.add(new Question("What is the largest mammal on Earth?", "Blue Whale", "Elephant", "Giraffe", "Lion", "The blue whale is the largest animal ever known to have lived on Earth."));  
 questions.add(new Question("What is the chemical symbol for gold?", "Au", "Ag", "Fe", "Cu", "The chemical symbol for gold is Au."));  
 questions.add(new Question("Which planet is known as the Red Planet?", "Mars", "Venus", "Jupiter", "Saturn", "Mars is often called the 'Red Planet' because of its reddish appearance."));  
 questions.add(new Question("Who painted the Mona Lisa?", "Leonardo da Vinci", "Michelangelo", "Raphael", "Donatello", "Leonardo da Vinci painted the Mona Lisa."));  
  
 Collections.shuffle(questions);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.CENTER);  
 cardPane.setStyle("-fx-background-color: #439576; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #1b548d; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(700);  
 cardPane.setMaxHeight(500);  
  
 // Top bar  
 Region spacer1 = new Region();  
 HBox.setHgrow(spacer1, Priority.ALWAYS);  
 spacer1.setMaxSize(150,150);  
 Region spacer2 = new Region();  
 HBox.setHgrow(spacer2, Priority.ALWAYS);  
 spacer2.setMaxSize(150,150);  
  
 HBox topBar = new HBox();  
 topBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 20,30,20,30");  
 topBar.setAlignment(Pos.CENTER);  
  
 ImageView coinImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/coins\_button.png").toString()));  
 coinImage.setFitWidth(30);  
 coinImage.setFitHeight(30);  
 scoreLabel = new Label(" 0");  
 scoreLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView timerImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/clock\_button.png").toString()));  
 timerImage.setFitWidth(30);  
 timerImage.setFitHeight(30);  
 timerLabel = new Label(" 30");  
 timerLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView lifeImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/life\_button.png").toString()));  
 lifeImage.setFitWidth(30);  
 lifeImage.setFitHeight(30);  
 lifeLabel = new Label(" 5");  
 lifeLabel.setStyle("-fx-font-size: 18px;");  
  
 topBar.getChildren().addAll(coinImage, scoreLabel, spacer1,timerImage, timerLabel,spacer2, lifeImage, lifeLabel);  
  
  
 // Center content  
 VBox center = new VBox(20);  
 center.setAlignment(Pos.CENTER);  
  
 questionLabel = new Label(questions.get(questionIndex).getQuestion());  
 questionLabel.setMaxWidth(600);  
 questionLabel.setMaxHeight(400);  
 questionLabel.setStyle("-fx-font-size: 18px;-fx-background-color: #ecf1ef;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 20px;-fx-background-radius: 20px;-fx-padding: 20;-fx-text-alignment: center");  
 center.getChildren().add(questionLabel);  
  
 answerButtons = new ArrayList<>();  
 HBox row1 = new HBox();  
 row1.setAlignment(Pos.CENTER);  
 row1.setSpacing(20);  
 HBox row2 = new HBox();  
 row2.setAlignment(Pos.CENTER);  
 row2.setSpacing(20);  
  
 Button button1 = new Button();  
 button1.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button1.setOnAction(event -> checkAnswer(button1));  
 button1.setPrefSize(150,50);  
 answerButtons.add(button1);  
  
 Button button2 = new Button();  
 button2.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d;-fx-background-radius: 20px; -fx-padding: 10px 20px;");  
 button2.setOnAction(event -> checkAnswer(button2));  
 button2.setPrefSize(150,50);  
 answerButtons.add(button2);  
 row1.getChildren().addAll(button1,button2);  
  
 Button button3 = new Button();  
 button3.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button3.setOnAction(event -> checkAnswer(button3));  
 button3.setPrefSize(150,50);  
 answerButtons.add(button3);  
  
 Button button4 = new Button();  
 button4.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button4.setOnAction(event -> checkAnswer(button4));  
 button4.setPrefSize(150,50);  
 answerButtons.add(button4);  
 row2.getChildren().addAll(button3,button4);  
 VBox layout = new VBox();  
 layout.setAlignment(Pos.CENTER);  
 layout.setPrefHeight(200);  
 // layout.setPrefWidth(200);  
 layout.setSpacing(15);  
 layout.getChildren().addAll(row1,row2);  
  
 // Bottom bar  
 HBox bottomBar = new HBox(20);  
 bottomBar.setAlignment(Pos.CENTER);  
 bottomBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 10,30,10,30");  
 // topBar.setSpacing(80);  
 bottomBar.setAlignment(Pos.CENTER);  
  
 hintButton = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/hint\_button.png").toExternalForm()));  
 Button hint=new Button();  
 hintButton.setFitWidth(30);  
 hint.setStyle("-fx-background-color: #1b548d;-fx-border-color: #ffe47a;-fx-border-radius: 20px;-fx-background-radius: 20px;");  
 hintButton.setFitHeight(30);  
 hint.setGraphic(hintButton);  
 hint.setOnMouseClicked(event -> showHint());  
  
 questionNumberLabel = new Label("< 1 /" + questions.size() + " >");  
 questionNumberLabel.setStyle("-fx-font-size: 18px;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 bottomBar.getChildren().add( questionNumberLabel);  
 root.setBottom(bottomBar);  
  
 VBox all\_labels=new VBox(10);  
 all\_labels.setAlignment(Pos.CENTER);  
 all\_labels.getChildren().addAll(topBar,cardPane,bottomBar);  
  
 // Hint label  
 hintLabel = new Label();  
 hintLabel.setStyle("-fx-background-color: #ffffa5; -fx-text-fill: black; -fx-padding: 5px; -fx-border-color: black;");  
 hintLabel.setVisible(false);  
 center.getChildren().add(hintLabel);  
 cardPane.getChildren().addAll(center,layout,hint);  
 root.setCenter(all\_labels);  
 // Timer  
 startTimer();  
  
 Scene scene = new Scene(root, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.initStyle(StageStyle.UNDECORATED);  
 primaryStage.setMaximized(true);  
 primaryStage.show();  
  
 }  
  
 private void startTimer() {  
 timer = new AnimationTimer() {  
 private long lastUpdate = System.nanoTime();  
  
 @Override  
 public void handle(long now) {  
 long elapsedTime = now - lastUpdate;  
 lastUpdate = now;  
 remainingTime -= elapsedTime / 1\_000\_000; // Convert to milliseconds  
  
 if (remainingTime <= 0) {  
 this.stop();  
 handleTimeUp();  
 }  
  
 timerLabel.setText(String.valueOf(remainingTime / 1000));  
 }  
 };  
 timer.start();  
 }  
  
 private void checkAnswer(Button button) {  
 if (button.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 button.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 currentScore += 1;  
 scoreLabel.setText(String.valueOf(currentScore));  
 } else {  
 button.setStyle("-fx-background-color: red; -fx-text-fill: white;");  
 currentLife--;  
 for (Button cbutton : answerButtons) {  
 if (cbutton.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 cbutton.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 break;  
 }  
 } lifeLabel.setText(String.valueOf(currentLife));  
 if (currentLife == 0) {  
 //add game over or result page  
 return;  
 }  
 }  
  
 answerButtons.forEach(btn -> btn.setDisable(true));  
  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.runLater(() -> nextQuestion());  
 }  
 },  
 1000  
 );  
 }  
  
 private void nextQuestion() {  
 if (questionIndex < questions.size() - 1) {  
 questionIndex++;  
 questionLabel.setText(questions.get(questionIndex).getQuestion());  
 for (int i = 0; i < 4; i++) {  
 Button button = answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 button.setDisable(false);  
 button.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 10px; -fx-padding: 10px 20px;-fx-text-fill: black" +  
 "");  
 }  
 questionNumberLabel.setText("< " + (questionIndex + 1) + "/" + questions.size() + " >");  
 remainingTime = 30000;  
 timerLabel.setText(String.valueOf(remainingTime / 1000));  
 } else {  
 System.out.println("No more questions");  
 //add game over or result page  
 }  
 }  
  
 private void handleTimeUp() {  
 System.out.println("Time's up!");  
 //add game over or result page  
 }  
  
  
 private void showHint() {  
 if (currentLife > 0) {  
 currentLife--;  
 lifeLabel.setText(String.valueOf(currentLife));  
 hintLabel.setText(questions.get(questionIndex).getHint());  
 hintLabel.setVisible(true);  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.runLater(() -> hintLabel.setVisible(false));  
 }  
 },  
 3000  
 );  
 }  
 }  
  
 private static class Question {  
 private final String question;  
 private final String correctAnswer;  
 private final List<String> options;  
 private final String hint;  
  
 public Question(String question, String correctAnswer, String option1, String option2, String option3, String hint) {  
 this.question = question;  
 this.correctAnswer = correctAnswer;  
 this.options = new ArrayList<>();  
 this.options.add(option1);  
 this.options.add(option2);  
 this.options.add(option3);  
 this.options.add(correctAnswer);  
 Collections.shuffle(this.options);  
 this.hint = hint;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getCorrectAnswer() {  
 return correctAnswer;  
 }  
  
 public List<String> getOptions() {  
 return options;  
 }  
  
 public String getHint() {  
 return hint;  
 }  
 }  
}

**Puzzle Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.AnimationTimer;  
import javafx.application.Application;  
import javafx.application.Platform;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
  
import java.util.ArrayList;  
import java.util.Collections;  
import java.util.List;  
  
public class Puzzle\_page extends Application {  
  
 private Label questionLabel;  
 private List<Button> answerButtons;  
 private Label scoreLabel;  
 private Label timerLabel;  
 private Label lifeLabel;  
 private ImageView hintButton;  
 private Label hintLabel;  
 private Label questionNumberLabel;  
 private int currentScore = 0;  
 private int currentLife = 5;  
 private int questionIndex = 0;  
 private long remainingTime = 30000; // 30 seconds in milliseconds  
  
 private List<Question> questions = new ArrayList<>();  
 private AnimationTimer timer;  
  
 @Override  
 public void start(Stage primaryStage) {  
 CustomTitleBar customTitleBar = new CustomTitleBar(primaryStage);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
 // Initialize questions  
 questions.add(new Question("Which country has kangaroo as national animal?", "Australia", "UK", "USA", "UAE", "Kangaroos are native to Australia."));  
 questions.add(new Question("What is the largest mammal on Earth?", "Blue Whale", "Elephant", "Giraffe", "Lion", "The blue whale is the largest animal ever known to have lived on Earth."));  
 questions.add(new Question("What is the chemical symbol for gold?", "Au", "Ag", "Fe", "Cu", "The chemical symbol for gold is Au."));  
 questions.add(new Question("Which planet is known as the Red Planet?", "Mars", "Venus", "Jupiter", "Saturn", "Mars is often called the 'Red Planet' because of its reddish appearance."));  
 questions.add(new Question("Who painted the Mona Lisa?", "Leonardo da Vinci", "Michelangelo", "Raphael", "Donatello", "Leonardo da Vinci painted the Mona Lisa."));  
  
 Collections.*shuffle*(questions);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #439576; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #1b548d; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(700);  
 cardPane.setMaxHeight(500);  
  
 // Top bar  
 Region spacer1 = new Region();  
 HBox.*setHgrow*(spacer1, Priority.*ALWAYS*);  
 spacer1.setMaxSize(150,150);  
 Region spacer2 = new Region();  
 HBox.*setHgrow*(spacer2, Priority.*ALWAYS*);  
 spacer2.setMaxSize(150,150);  
  
 HBox topBar = new HBox();  
 topBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 20,30,20,30");  
 topBar.setAlignment(Pos.*CENTER*);  
  
 ImageView coinImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/coins\_button.png").toString()));  
 coinImage.setFitWidth(30);  
 coinImage.setFitHeight(30);  
 scoreLabel = new Label(" 0");  
 scoreLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView timerImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/clock\_button.png").toString()));  
 timerImage.setFitWidth(30);  
 timerImage.setFitHeight(30);  
 timerLabel = new Label(" 30");  
 timerLabel.setStyle("-fx-font-size: 18px;");  
  
 ImageView lifeImage = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/life\_button.png").toString()));  
 lifeImage.setFitWidth(30);  
 lifeImage.setFitHeight(30);  
 lifeLabel = new Label(" 5");  
 lifeLabel.setStyle("-fx-font-size: 18px;");  
  
 topBar.getChildren().addAll(coinImage, scoreLabel, spacer1,timerImage, timerLabel,spacer2, lifeImage, lifeLabel);  
  
  
 // Center content  
 VBox center = new VBox(20);  
 center.setAlignment(Pos.*CENTER*);  
  
 questionLabel = new Label(questions.get(questionIndex).getQuestion());  
 questionLabel.setMaxWidth(600);  
 questionLabel.setMaxHeight(400);  
 questionLabel.setStyle("-fx-font-size: 18px;-fx-background-color: #ecf1ef;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 20px;-fx-background-radius: 20px;-fx-padding: 20;-fx-text-alignment: center");  
 center.getChildren().add(questionLabel);  
  
 answerButtons = new ArrayList<>();  
 HBox row1 = new HBox();  
 row1.setAlignment(Pos.*CENTER*);  
 row1.setSpacing(20);  
 HBox row2 = new HBox();  
 row2.setAlignment(Pos.*CENTER*);  
 row2.setSpacing(20);  
  
 Button button1 = new Button();  
 button1.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button1.setOnAction(event -> checkAnswer(button1));  
 button1.setPrefSize(150,50);  
 answerButtons.add(button1);  
  
 Button button2 = new Button();  
 button2.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d;-fx-background-radius: 20px; -fx-padding: 10px 20px;");  
 button2.setOnAction(event -> checkAnswer(button2));  
 button2.setPrefSize(150,50);  
 answerButtons.add(button2);  
 row1.getChildren().addAll(button1,button2);  
  
 Button button3 = new Button();  
 button3.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button3.setOnAction(event -> checkAnswer(button3));  
 button3.setPrefSize(150,50);  
 answerButtons.add(button3);  
  
 Button button4 = new Button();  
 button4.setStyle("-fx-background-color: #ffffff; -fx-text-fill: #1b548d; -fx-background-radius: 20px;-fx-padding: 10px 20px;");  
 button4.setOnAction(event -> checkAnswer(button4));  
 button4.setPrefSize(150,50);  
 answerButtons.add(button4);  
 for(int i=0;i<4;i++){  
 Button button=answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 }  
 row2.getChildren().addAll(button3,button4);  
 VBox layout = new VBox();  
 layout.setAlignment(Pos.*CENTER*);  
 layout.setPrefHeight(200);  
 // layout.setPrefWidth(200);  
 layout.setSpacing(15);  
 layout.getChildren().addAll(row1,row2);  
  
 // Bottom bar  
 HBox bottomBar = new HBox(20);  
 bottomBar.setAlignment(Pos.*CENTER*);  
 bottomBar.setStyle("-fx-background-color: #ffe47a;-fx-padding: 10,30,10,30");  
 // topBar.setSpacing(80);  
 bottomBar.setAlignment(Pos.*CENTER*);  
  
 hintButton = new ImageView(new Image(getClass().getResource("/com/example/mind\_marathon\_project/hint\_button.png").toExternalForm()));  
 Button hint=new Button();  
 hintButton.setFitWidth(30);  
 hint.setStyle("-fx-background-color: #1b548d;-fx-border-color: #ffe47a;-fx-border-radius: 20px;-fx-background-radius: 20px;");  
 hintButton.setFitHeight(30);  
 hint.setGraphic(hintButton);  
 hint.setOnMouseClicked(event -> showHint());  
  
 questionNumberLabel = new Label("< 1 /" + questions.size() + " >");  
 questionNumberLabel.setStyle("-fx-font-size: 18px;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 bottomBar.getChildren().add( questionNumberLabel);  
 root.setBottom(bottomBar);  
  
 VBox all\_labels=new VBox(10);  
 all\_labels.setAlignment(Pos.*CENTER*);  
 all\_labels.getChildren().addAll(topBar,cardPane,bottomBar);  
  
 // Hint label  
 hintLabel = new Label();  
 hintLabel.setStyle("-fx-background-color: #ffffa5; -fx-text-fill: black; -fx-padding: 5px; -fx-border-color: black;");  
 hintLabel.setVisible(false);  
 center.getChildren().add(hintLabel);  
 cardPane.getChildren().addAll(center,layout,hint);  
 root.setCenter(all\_labels);  
 // Timer  
 startTimer();  
  
 Scene scene = new Scene(root, 800, 600);  
 primaryStage.setScene(scene);  
 primaryStage.initStyle(StageStyle.*UNDECORATED*);  
 primaryStage.setMaximized(true);  
 primaryStage.show();  
  
 }  
  
 private void startTimer() {  
 timer = new AnimationTimer() {  
 private long lastUpdate = System.*nanoTime*();  
  
 @Override  
 public void handle(long now) {  
 long elapsedTime = now - lastUpdate;  
 lastUpdate = now;  
 remainingTime -= elapsedTime / 1\_000\_000; // Convert to milliseconds  
  
 if (remainingTime <= 0) {  
 this.stop();  
 handleTimeUp();  
 }  
  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 }  
 };  
 timer.start();  
 }  
  
 private void checkAnswer(Button button) {  
 if (button.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 button.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 currentScore += 1;  
 scoreLabel.setText(String.*valueOf*(currentScore));  
 } else {  
 button.setStyle("-fx-background-color: red; -fx-text-fill: white;");  
 currentLife--;  
 for (Button cbutton : answerButtons) {  
 if (cbutton.getText().equals(questions.get(questionIndex).getCorrectAnswer())) {  
 cbutton.setStyle("-fx-background-color: green; -fx-text-fill: white;");  
 break;  
 }  
 } lifeLabel.setText(String.*valueOf*(currentLife));  
 if (currentLife == 0) {  
 //add game over or result page  
 return;  
 }  
 }  
  
 answerButtons.forEach(btn -> btn.setDisable(true));  
  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> nextQuestion());  
 }  
 },  
 1000  
 );  
 }  
  
 private void nextQuestion() {  
 if (questionIndex < questions.size() - 1) {  
 questionIndex++;  
 questionLabel.setText(questions.get(questionIndex).getQuestion());  
 for (int i = 0; i < 4; i++) {  
 Button button = answerButtons.get(i);  
 button.setText(questions.get(questionIndex).getOptions().get(i));  
 button.setDisable(false);  
 button.setStyle("-fx-background-color: #f1f5f6;-fx-background-radius: 20px;-fx-border-color: #1b548d;-fx-border-width:3;-fx-border-radius: 10px; -fx-padding: 10px 20px;-fx-text-fill: black" +  
 "");  
 }  
 questionNumberLabel.setText("< " + (questionIndex + 1) + "/" + questions.size() + " >");  
 remainingTime = 30000;  
 timerLabel.setText(String.*valueOf*(remainingTime / 1000));  
 } else {  
 System.*out*.println("No more questions");  
 //add game over or result page  
 }  
 }  
  
 private void handleTimeUp() {  
 System.*out*.println("Time's up!");  
 //add game over or result page  
 }  
  
  
 private void showHint() {  
 if (currentLife > 0) {  
 currentLife--;  
 lifeLabel.setText(String.*valueOf*(currentLife));  
 hintLabel.setText(questions.get(questionIndex).getHint());  
 hintLabel.setVisible(true);  
 new java.util.Timer().schedule(  
 new java.util.TimerTask() {  
 @Override  
 public void run() {  
 Platform.*runLater*(() -> hintLabel.setVisible(false));  
 }  
 },  
 3000  
 );  
 }  
 }  
  
 private static class Question {  
 private final String question;  
 private final String correctAnswer;  
 private final List<String> options;  
 private final String hint;  
  
 public Question(String question, String correctAnswer, String option1, String option2, String option3, String hint) {  
 this.question = question;  
 this.correctAnswer = correctAnswer;  
 this.options = new ArrayList<>();  
 this.options.add(option1);  
 this.options.add(option2);  
 this.options.add(option3);  
 this.options.add(correctAnswer);  
 Collections.*shuffle*(this.options);  
 this.hint = hint;  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String getCorrectAnswer() {  
 return correctAnswer;  
 }  
  
 public List<String> getOptions() {  
 return options;  
 }  
  
 public String getHint() {  
 return hint;  
 }  
 }  
}

**Choose Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.ToggleButton;  
import javafx.scene.image.Image;  
import javafx.scene.control.ToggleGroup;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.VBox;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
public class Interest\_page extends Application {  
 @Override  
 public void start(Stage stage4) throws Exception {  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage4);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #ffffff;");  
 root.setTop(customTitleBar);  
  
 VBox cardPane = new VBox(20);  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #1b548d; " +  
 "-fx-background-radius: 20px;-fx-border-width: 3;" +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 20px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 ImageView imagelogo;  
 try{  
 imagelogo=new ImageView(new Image(getClass().getResourceAsStream("/com/example/mind\_marathon\_project/iterest\_tag.png")));  
 }  
 catch(NullPointerException e){  
 throw new RuntimeException(e);  
 }  
 imagelogo.setFitHeight(150);  
 imagelogo.setFitWidth(400);  
  
 ToggleGroup toggleGroup = new ToggleGroup();  
  
 ToggleButton scienceBtn = createToggleButton("Science",toggleGroup);  
 ToggleButton artBtn = createToggleButton("Art",toggleGroup);  
 ToggleButton geographyBtn = createToggleButton("Geography",toggleGroup);  
 ToggleButton historyBtn = createToggleButton("History",toggleGroup);  
 ToggleButton puzzleBtn = createToggleButton("Puzzle",toggleGroup);  
 ToggleButton sportsBtn = createToggleButton("Sports",toggleGroup);  
  
 scienceBtn.setToggleGroup(toggleGroup);  
 artBtn.setToggleGroup(toggleGroup);  
 geographyBtn.setToggleGroup(toggleGroup);  
 historyBtn.setToggleGroup(toggleGroup);  
 puzzleBtn.setToggleGroup(toggleGroup);  
 sportsBtn.setToggleGroup(toggleGroup);  
 // HBox to contain two rows of buttons  
 HBox row1 = new HBox(10, scienceBtn, artBtn);  
 row1.setAlignment(Pos.*CENTER*);  
 HBox row2 = new HBox(10, geographyBtn, historyBtn);  
 row2.setAlignment(Pos.*CENTER*);  
 HBox row3 = new HBox(10, puzzleBtn, sportsBtn);  
 row3.setAlignment(Pos.*CENTER*);  
  
 Button okButton = new Button("OK");  
 okButton.setStyle("-fx-background-color: #fdf58e;-fx-border-color:#f59eb7;-fx-border-width: 2px;-fx-border-radius: 20px;-fx-text-fill: #1b548d;-fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 10 20");  
 try {  
 addButtonEffects(okButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 }  
 catch (NullPointerException e) {  
 throw new RuntimeException( e);  
 }  
 okButton.setOnAction(e -> System.*out*.println("Name: " ));  
 okButton.setOnAction(e -> System.*out*.println("Selected: " +  
 (toggleGroup.getSelectedToggle() != null ? ((ToggleButton) toggleGroup.getSelectedToggle()).getText() : "None")));  
  
  
 VBox vbox = new VBox(20, imagelogo, row1, row2, row3, okButton);  
 vbox.setStyle("-fx-alignment: center; -fx-padding: 20; -fx-background-color: #1b548d;");  
 vbox.setAlignment(Pos.*CENTER*);  
  
 cardPane.getChildren().addAll(vbox);  
  
 root.setCenter(cardPane);  
 okButton.setOnAction(e->{  
 if(scienceBtn.isSelected()){  
 try {  
 new Science\_page().start(new Stage());  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
 stage4.close();  
 } else if (artBtn.isSelected()) {  
 new Art\_page().start(new Stage());  
 stage4.close();  
  
 } else if (geographyBtn.isSelected()) {  
 new Geography\_page().start(new Stage());  
 stage4.close();  
 } else if (historyBtn.isSelected()) {  
 new History\_page().start(new Stage());  
 stage4.close();  
  
  
 } else if (puzzleBtn.isSelected()) {  
 new Puzzle\_page().start(new Stage());  
 stage4.close();  
  
 } else if (sportsBtn.isSelected()) {  
 new Sports\_page().start(new Stage());  
 stage4.close();  
  
 } else{  
 try {  
 new Menu\_page().start(new Stage());  
 } catch (Exception ex) {  
 throw new RuntimeException(ex);  
 }  
  
 }  
 });  
  
 Scene scene = new Scene(root,800,600);  
 stage4.initStyle(StageStyle.*UNDECORATED*);  
 stage4.setMaximized(true);  
 stage4.setScene(scene);  
 stage4.show();  
  
 }  
 private ToggleButton createToggleButton(String text, ToggleGroup toggleGroup) {  
 ToggleButton toggleButton = new ToggleButton(text);  
 toggleButton.setAlignment(Pos.*CENTER*);  
 toggleButton.setPrefWidth(150);  
 toggleButton.setPrefHeight(40);  
 toggleButton.setStyle("-fx-background-color: #fffcf6; -fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-text-fill: #1b548d; -fx-padding:15 25 15 25;");  
  
 toggleButton.setToggleGroup(toggleGroup);  
 // Toggle behavior to show a yellow glow when selected  
 toggleGroup.selectedToggleProperty().addListener((observable, oldToggle, newToggle) -> {  
 if (oldToggle != null) {  
 ((ToggleButton) oldToggle).setStyle("-fx-background-color: #fffcf6; -fx-background-radius: 20px;-fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-text-fill: #1b548d; -fx-padding:15 25 15 25;");  
 }  
 if (newToggle != null) {  
 ((ToggleButton) newToggle).setStyle("-fx-background-color: #ffffa5;-fx-background-radius: 20px; -fx-text-fill: #3d3939; -fx-font-family: Calibri;-fx-font-weight:bold;-fx-font-size: 18px;-fx-padding: 15 25 15 25;");  
 }  
 });  
  
 return toggleButton;  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 // Play sound effect  
 // AudioClip clickSound = new AudioClip(getClass().getResource(soundFile).toExternalForm());  
  
 // Add hover effects  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
  
 // Play sound on click  
 // button.setOnMouseClicked(e -> clickSound.play());  
 }  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

**Exit Class**

package com.example.mind\_marathon\_project;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
import javafx.stage.StageStyle;  
import javafx.util.Duration;  
  
public class exit\_page extends Application {  
 @Override  
 public void start(Stage stage10) throws Exception {  
  
 CustomTitleBar customTitleBar = new CustomTitleBar(stage10);  
 BorderPane root = new BorderPane();  
 root.setStyle("-fx-background-color: #fffcf6;");  
 root.setTop(customTitleBar);  
 VBox cardPane = new VBox();  
 cardPane.setAlignment(Pos.*CENTER*);  
 cardPane.setStyle("-fx-background-color: #f7f2e8; " +  
 "-fx-background-radius: 20px; " +  
 "-fx-border-radius: 20px; " +  
 "-fx-border-color: #ff7bac; " +  
 "-fx-padding: 40px;");  
 cardPane.setMaxWidth(600);  
 cardPane.setMaxHeight(500);  
  
 HBox hBox = new HBox();  
 Label text=new Label("Are you sure you \nwant to exit?");  
 text.setStyle("-fx-text-fill: white;");  
 hBox.setStyle("-fx-background-color: #3c6ca8;-fx-background-radius: 20px;-fx-border-radius: 20px;-fx-border-width:4;-fx-border-color: #ff7bac;-fx-padding: 20px;-fx-text-fill: white ;-fx-font-weight: bold;-fx-font-family: 'Comic Sans MS'");  
 hBox.setMaxWidth(300);  
 hBox.setMaxHeight(1000);  
 hBox.setAlignment(Pos.*CENTER*);  
 hBox.getChildren().add(text);  
  
  
 Button yesButton = new Button(" Yes ");  
 yesButton.setStyle("-fx-background-radius:10px;-fx-background-size: 100px;-fx-background-color: #86cc38;-fx-border-color: #ff7bac;-fx-border-radius: 10px;-fx-border-width: 2;-fx-font-size: 14px; -fx-text-fill: #ffffff;-fx-font-family:'Comic Sans MS' ;-fx-font-weight: bold;-fx-padding:11");  
 try {  
 addButtonEffects(yesButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 }  
 catch (NullPointerException e) {  
 throw new RuntimeException( e);  
 }  
 yesButton.setOnAction(e -> System.*exit*(0));  
  
 Button noButton = new Button(" No ");  
 noButton.setStyle("-fx-background-radius:10px;-fx-background-size: 100px;-fx-background-color: #fa4848;-fx-border-color: #ff7bac;-fx-border-radius: 10px;-fx-border-width: 2;-fx-font-size: 14px; -fx-text-fill: #ffffff;-fx-font-family:'Comic Sans MS' ;-fx-font-weight: bold;-fx-padding: 11;");  
 try {  
 addButtonEffects(noButton, "/com/example/mind\_marathon\_project/main\_button.mp3");  
 }  
 catch (NullPointerException e) {  
 throw new RuntimeException( e);  
 }  
  
  
 HBox hBox1 = new HBox();  
 hBox1.setSpacing(30);  
 hBox1.setAlignment(Pos.*CENTER*);  
  
 hBox1.getChildren().addAll(yesButton,noButton);  
 VBox vBox = new VBox();  
 vBox.setAlignment(Pos.*CENTER*);  
 vBox.setSpacing(30);  
 vBox.getChildren().addAll(hBox,hBox1);  
 cardPane.getChildren().addAll(vBox);  
 root.setCenter(cardPane);  
  
 Scene scene = new Scene(root, 800, 600);  
 stage10.initStyle(StageStyle.*UNDECORATED*);  
 stage10.setMaximized(true);  
 stage10.setScene(scene);  
 stage10.show();  
  
  
 }  
 private void addButtonEffects(Button button, String soundFile) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> {  
 scaleTransition.stop();  
 button.setScaleX(1.0);  
 button.setScaleY(1.0);  
 });  
 }  
}

**GitHUb link:**

**https://github.com/Jannat-ul-Mava/Mind-Marathon-Game**