

**Assignment**

**Comsats university Islamabad,Lahore Campus**

***Topic :***

**Console of**

**project**

Submitted by:

Umair Khalid, Hasham Aziz,Huzaifa

Roll no:

SP24-BSE-119,041,047

Submitted to

Sir Shahid Bhatti

Date

12 Dec 2024

This is My App.java

package com.example;

import javafx.animation.Timeline;

import javafx.application.Application;

import javafx.geometry.Insets;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.image.Image;

import javafx.scene.image.ImageView;

import javafx.scene.input.KeyCode;

import javafx.scene.layout.\*;

import javafx.stage.Stage;

import javafx.scene.control.\*;

import javafx.scene.control.Alert.AlertType;

import javafx.scene.media.Media;

import javafx.scene.media.MediaPlayer;

import javafx.scene.media.MediaView;

import java.io.IOException;

import java.util.ArrayList;

import javafx.animation.KeyFrame;

import javafx.util.Duration;

import org.json.JSONObject;

public class App extends Application {

  Sound soundClass = new Sound();

  NewFile dataFile = new NewFile();

  ArrayList<UserData> userInfo = new ArrayList<>();

  private Stage primaryStage;

  @Override

  public void start(Stage primaryStage) throws IOException {

    this.primaryStage = primaryStage;

    // primaryStage.setFullScreen(true);

    showLoginScreen();

  }

  public void showLoginScreen() {

    // GridPane layout = new GridPane();

    VBox layout = new VBox();

    layout.getStyleClass().add("root1");

    layout.setSpacing(8);

    // Apply the CSS class

    // Welcome

    // Label wlcmLabel=new Label("Welcome to fitness journey");

    Image newWlcmImg = new Image(

        "file:C://Users//DELL//Music//table\_tut2//src//main//resources//com//example//homepage.png/");

    ImageView newImageView = new ImageView(newWlcmImg);

    newImageView.setFitHeight(350);

    // HBox wlcmBox=new HBox(newImageView);

    // wlcmBox.setAlignment(Pos.CENTER);

    VBox.setMargin(newImageView, new Insets(10, -45, 0, 90));

    layout.setAlignment(Pos.CENTER);

    // Healthy text Label

    Label healthyLabel1 = new Label("Make");

    Label healthyLabel2 = new Label("Your Body");

    Label healthyLabel3 = new Label("Healthy & Fit");

    healthyLabel1.setId("healthyLabel1");

    healthyLabel2.setId("healthyLabel2");

    healthyLabel3.setId("healthyLabel3");

    // add in Hbox

    HBox healthyBox = new HBox();

    healthyBox.getChildren().addAll(healthyLabel1, healthyLabel2);

    healthyBox.setSpacing(10);

    healthyBox.setAlignment(Pos.CENTER);

    VBox.setMargin(healthyBox, new Insets(0, 0, -20, 0));

    // timeline

    Timeline timeline = new Timeline(

        new KeyFrame(Duration.ZERO, e -> healthyLabel1.setStyle("-fx-text-fill: orange")),

        new KeyFrame(Duration.seconds(2), e -> healthyLabel1.setStyle("-fx-text-fill: black ")),

        new KeyFrame(Duration.seconds(4), e -> healthyLabel1.setStyle("-fx-text-fill: orange")));

    timeline.setCycleCount(Timeline.INDEFINITE); // Repeat the animation

    timeline.play();

    // Timeline for healthyLabel2

    Timeline timeline1 = new Timeline(

        new KeyFrame(Duration.ZERO, e -> healthyLabel2.setStyle("- -fx-text-fill: Black")),

        new KeyFrame(Duration.seconds(2), e -> healthyLabel2.setStyle(" -fx-text-fill:Grey")),

        new KeyFrame(Duration.seconds(4), e -> healthyLabel2.setStyle("-fx-text-fill: Black"))

    );

    timeline1.setCycleCount(Timeline.INDEFINITE); // Repeat the animation

    timeline1.play();

    // Timeline for healthyLabel3

    Timeline timeline2 = new Timeline(

        new KeyFrame(Duration.ZERO, e -> healthyLabel3.setStyle("-fx-text-fill: Blue")),

        new KeyFrame(Duration.seconds(2), e -> healthyLabel3.setStyle("-fx-text-fill:orange")),

        new KeyFrame(Duration.seconds(4), e -> healthyLabel3.setStyle("-fx-text-fill: Blue"))

    );

    timeline2.setCycleCount(Timeline.INDEFINITE); // Repeat the animation

    timeline2.play();

    Button getStartedBtn = new Button("Get Started");

    getStartedBtn.setId("getStartedBtn");

    getStartedBtn.setOnAction(o -> {

      soundClass.soundRun();

      AcessPage();

    });

    layout.getChildren().addAll(newImageView, healthyBox, healthyLabel3, getStartedBtn);

    // VBox.setMargin(btnHBox, new Insets(0,0,0,50));

    Image logoimage = new Image(

        "file:C://Users//DELL//Documents//java ntoepad//table\_tut2//src//main//resources//com//example//logo.jpg");

    // help button on action

    Scene scene1 = new Scene(layout,350,600);

    scene1.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

    primaryStage.getIcons().add(logoimage);

    primaryStage.setScene(scene1);

    primaryStage.setTitle("Background Image Example");

    primaryStage.show();

    // boolean isFullScreen = primaryStage.isFullScreen();

    // primaryStage.setFullScreen(isFullScreen);

  }

  public void AcessPage() {

    VBox accessVBox = new VBox();

    // back Button

    Button backButton = new Button("");

    backButton.getStyleClass().add("backButtonStyle");

    backButton.setOnAction(e -> showLoginScreen());

    // Signin Label

    Label signInLabel1 = new Label("Sign In");

    Label signInLabel2 = new Label("To Your");

    Label signInLabel3 = new Label("Account");

    signInLabel1.setId("signInLabel1");

    signInLabel2.setId("signInLabel2");

    signInLabel3.setId("signInLabel3");

    // VBox.setMargin(signInLabel3, new Insets(-20, 0, 0, 65));

    // add label to hbox

    HBox signInLabelBox = new HBox();

    signInLabelBox.getChildren().addAll(signInLabel1, signInLabel2);

    signInLabelBox.setAlignment(Pos.CENTER);

    signInLabelBox.setSpacing(10);

    // account Hbox

    HBox accountLabelBox = new HBox(signInLabel3);

    accountLabelBox.setAlignment(Pos.CENTER);

    // login method label

    Label loginMethodLabel = new Label("login with the following methods");

    loginMethodLabel.setId("loginMethodLabel");

    // loginMethodLabel.setTranslateX(50);

    HBox loginLabelHbox = new HBox(loginMethodLabel);

    loginLabelHbox.setAlignment(Pos.CENTER);

    // VBox.setMargin(loginMethodLabel, new Insets(0, 0, 10, 65));

    // login buttons

    Button googleButton = new Button("");

    Button facebookButton = new Button("");

    Button githubButton = new Button("");

    // adding button to hbox

    HBox threeBoxesBtn = new HBox();

    threeBoxesBtn.getChildren().addAll(googleButton, facebookButton, githubButton);

    threeBoxesBtn.setSpacing(20);

    threeBoxesBtn.setAlignment(Pos.CENTER);

    // VBox.setMargin(threeBoxesBtn, new Insets(0, 0, 0, 15));

    // giving class to buttons

    googleButton.getStyleClass().add("loginBtn1");

    facebookButton.getStyleClass().add("loginBtn2");

    githubButton.getStyleClass().add("loginBtn3");

    // Or label

    Label orLabel = new Label("<------OR------>");

    orLabel.setId("orLabel");

    HBox orBox = new HBox();

    orBox.getChildren().add(orLabel);

    orBox.setAlignment(Pos.CENTER);

    VBox.setMargin(orBox, new Insets(15, 0, 0, 0));

    // UserName

    Label nameLabel = new Label("Username");

    TextField nameField = new TextField();

    nameField.setPromptText("Enter your username");

    nameField.setPrefWidth(15);

    nameField.setAlignment(Pos.CENTER);

    HBox nameLabelHBox = new HBox(nameLabel);

    nameLabelHBox.setAlignment(Pos.CENTER);

    // alignment

    nameLabelHBox.setTranslateY(-15);

    nameLabelHBox.setTranslateX(-75);

    nameLabel.setId("nameLabel");

    HBox nameBox = new HBox(nameField);

    nameBox.setAlignment(Pos.TOP\_CENTER);

    nameBox.setTranslateY(-25);

    // Password

    Label passLabel = new Label("Password");

    PasswordField passwordField = new PasswordField();

    passwordField.setPromptText("Enter Your Password");

    passLabel.setId("passLabel");

    HBox passLabelBox = new HBox(passLabel);

    passLabelBox.setAlignment(Pos.CENTER);

    passLabelBox.setTranslateY(-25);

    passLabelBox.setTranslateX(-76);

    HBox passBox = new HBox(passwordField);

    passBox.setAlignment(Pos.CENTER);

    passwordField.setAlignment(Pos.CENTER);

    passBox.setTranslateY(-35);

    nameField.getStyleClass().add("text-field");

    passwordField.getStyleClass().add("password-field");

    // Forget Password

    Label forgetLabel = new Label("Forgot Password?");

    forgetLabel.setId("forgetLabel");

    HBox forgetLabelBox = new HBox(forgetLabel);

    forgetLabelBox.setAlignment(Pos.CENTER);

    forgetLabelBox.setTranslateY(-40);

    forgetLabelBox.setTranslateX(85);

    forgetLabel.setOnMouseClicked(e -> {

      ForgetPasswordMethod();

    });

    // Login

    Button signInBtn = new Button("Login");

    signInBtn.setId("signInBtn");

    HBox signInBtnBox = new HBox(signInBtn);

    signInBtnBox.setAlignment(Pos.CENTER);

    signInBtnBox.setTranslateY(-35);

    // SignUp

    Label newMemLabel = new Label("New Member?");

    newMemLabel.setId("newMemLabel");

    newMemLabel.setTranslateX(20);

    Label registerLabel = new Label("Register Now");

    registerLabel.setId("registerLabel");

    Button signUpBtn = new Button("");

    signUpBtn.setId("signUpBtn");

    Image signUpBtnImg = new Image(getClass().getResource("fast-forward.png").toExternalForm());

    ImageView signUpBtnImView = new ImageView(signUpBtnImg);

    signUpBtnImView.setFitWidth(10);

    signUpBtnImView.setFitHeight(10);

    // give img to button

    signUpBtn.setGraphic(signUpBtnImView);

    signUpBtn.setOnAction(e -> signUpScreen());

    HBox signUpBox = new HBox(registerLabel, signUpBtn);

    signUpBox.setSpacing(120);

    signUpBox.setAlignment(Pos.CENTER);

    VBox signUpVbox = new VBox();

    signUpVbox.getChildren().addAll(newMemLabel, signUpBox);

    signUpVbox.getStyleClass().add("signUpVbox");

    signUpVbox.setPrefHeight(110);

    signUpVbox.setAlignment(Pos.CENTER\_LEFT);

    signUpVbox.setTranslateY(-30);

    signUpVbox.setPrefWidth(350);

    signUpVbox.setMinWidth(350);

    signUpVbox.setMaxWidth(350);

    // add vbox to hbox to bring it to center

    HBox signUpHBox = new HBox(signUpVbox);

    signUpHBox.setAlignment(Pos.CENTER);

    // Navbar

    HBox navHBox = new HBox();

    Button homeBtn = new Button("");

    Button helpButton = new Button("");

    navHBox.getChildren().addAll(homeBtn, helpButton);

    homeBtn.setId("homeBtn");

    helpButton.setId("helpButton");

    navHBox.setId("navHBox");

    navHBox.setPrefWidth(350);

    navHBox.setMinWidth(350);

    navHBox.setMaxWidth(350);

    navHBox.setTranslateY(-10);

    navHBox.setAlignment(Pos.CENTER);

    navHBox.setSpacing(40);

    HBox navParentHbox = new HBox(navHBox);

    navParentHbox.setAlignment(Pos.CENTER);

    homeBtn.setOnAction(i -> {

      showLoginScreen();

    });

    helpButton.setOnAction(j -> {

      HelpSupport helpSupport = new HelpSupport(this);

      try {

        helpSupport.start(primaryStage);

      } catch (Exception e1) {

        e1.printStackTrace();

      }

    });

    // add to main Vbox

    accessVBox.getChildren().addAll(signInLabelBox, accountLabelBox, loginLabelHbox, threeBoxesBtn, orBox,

        nameLabelHBox,

        nameBox, passLabelBox, passBox, forgetLabelBox, signInBtnBox, signUpHBox, navParentHbox);

    accessVBox.setSpacing(10);

    // creating a borderpane

    BorderPane borderLayout = new BorderPane();

    borderLayout.setTop(backButton);

    borderLayout.setCenter(accessVBox);

    borderLayout.requestLayout();

    signInBtn.setOnAction(e -> {

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

      String pass = passwordField.getText().trim();

      if (name.contains("~") || pass.contains("~")) {

        Alert alert = new Alert(Alert.AlertType.ERROR);

        alert.setTitle("Input Violation");

        alert.setHeaderText("Invalid Entry");

        alert.setContentText("The symbol '~' is not allowed in any field.");

        alert.showAndWait();

        return;

      }

      try {

        if (dataFile.checkInFile(name, pass)) {

          properApp();

        } else {

          Alert alert = new Alert(Alert.AlertType.INFORMATION);

          alert.setTitle("wrong entry");

          alert.setHeaderText("Incorrect username or password");

          alert.setContentText("Please try again");

          alert.showAndWait();

        }

      } catch (Exception j) {

        System.out.println(j.getMessage());

      }

    });

    // Set on Action on Sign Up button

    signUpBtn.setOnAction(e -> {

      signUpScreen();

    });

    borderLayout.setStyle("-fx-background-color:#272727");

    Scene accessScene = new Scene(borderLayout, 350, 630);

    accessScene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

    primaryStage.setTitle("Access Page");

    primaryStage.show();

    primaryStage.setScene(accessScene);

    // primaryStage.setFullScreen(true);

  }

  public void ForgetPasswordMethod() {

    VBox forgetVbox = new VBox(15);

    forgetVbox.setAlignment(Pos.CENTER);

    // button

    Button backButton = new Button("<-");

    backButton.getStyleClass().add("backButtonStyle");

    backButton.setOnAction(e -> AcessPage());

    Label emailforgetLabel = new Label("Email");

    TextField emailForget = new TextField();

    emailForget.setPromptText("Enter your email");

    Label movieLabel = new Label("Which was your Favourite childhood movie?");

    TextField movieField = new TextField();

    movieField.setPromptText("Enter the movie name here");

    Label schoolLabel = new Label("Plz Tell Us Name of Your First School?");

    TextField schoolField = new TextField();

    schoolField.setPromptText("Enter the school name here");

    Button resetBtn = new Button("Proceed");

    resetBtn.setOnKeyPressed(event -> {

      if (event.getCode() == KeyCode.ENTER) {

        String email = emailForget.getText().trim();

        String school = schoolField.getText().trim();

        String movie = movieField.getText().trim();

        if (dataFile.checkUser(email, school, movie)) {

          ResetPassword(email);

        } else {

          Alert alert = new Alert(AlertType.ERROR);

          alert.setTitle("NO User registered");

          alert.setHeaderText("NO User found");

          alert.setContentText("You are not registered in our database");

          alert.showAndWait();

        }

      }

    });

    resetBtn.setOnAction(e -> {

      String email = emailForget.getText().trim();

      String school = schoolField.getText().trim();

      String movie = movieField.getText().trim();

      if (dataFile.checkUser(email, school, movie)) {

        ResetPassword(email);

      } else {

        Alert alert = new Alert(AlertType.ERROR);

        alert.setTitle("NO User registered");

        alert.setHeaderText("NO User found");

        alert.setContentText("You are not registered in our database");

        alert.showAndWait();

      }

    });

    forgetVbox.getChildren().addAll(backButton, emailforgetLabel, emailForget, schoolLabel, schoolField, movieLabel,

        movieField, resetBtn);

    Scene scene = new Scene(forgetVbox, 350, 600);

    primaryStage.setTitle("forget Password");

    primaryStage.setScene(scene);

    primaryStage.show();

    // primaryStage.setFullScreen(true);

  }

  public void ResetPassword(String email) {

    VBox resVBox = new VBox();

    resVBox.setSpacing(10);

    resVBox.setAlignment(Pos.CENTER);

    String newEmail = email;

    // button

    Button backButton = new Button("<-");

    backButton.getStyleClass().add("backButtonStyle");

    backButton.setOnAction(e -> AcessPage());

    Label newPasswordLabel = new Label("New Password");

    TextField newPasswordField = new TextField();

    newPasswordField.setPromptText("Enter a new password");

    Label confirmPasswordLabel = new Label("Confirm Password");

    TextField confirmPasswordField = new TextField();

    Button proceedButton = new Button("Proceed");

    resVBox.getChildren().addAll(backButton, newPasswordLabel, newPasswordField, confirmPasswordLabel,

        confirmPasswordField, proceedButton);

    proceedButton.setOnKeyPressed(event -> {

      if (event.getCode() == KeyCode.ENTER) {

        String newPass = newPasswordField.getText().trim();

        String confirmPass = confirmPasswordField.getText().trim();

        if (newPass.equals(confirmPass)) {

          dataFile.updataPassword(newEmail, newPass);

          // alert

          Alert alert = new Alert(AlertType.INFORMATION);

          alert.setTitle("Password Changed");

          alert.setHeaderText("Password Changed Successfully");

          alert.setContentText("Your password has been changed successfully");

          alert.showAndWait();

        } else {

          Alert alert = new Alert(AlertType.ERROR);

          alert.setTitle("Password Mismatch");

          alert.setHeaderText("Password Mismatch");

          alert.setContentText("Passwords do not match");

          alert.showAndWait();

        }

      }

    });

    proceedButton.setOnAction(m -> {

      String newPass = newPasswordField.getText().trim();

      String confirmPass = confirmPasswordField.getText().trim();

      if (newPass.equals(confirmPass)) {

        dataFile.updataPassword(newEmail, newPass);

        // alert

        Alert alert = new Alert(AlertType.INFORMATION);

        alert.setTitle("Password Changed");

        alert.setHeaderText("Password Changed Successfully");

        alert.setContentText("Your password has been changed successfully");

        alert.showAndWait();

        AcessPage();

      } else {

        Alert alert = new Alert(AlertType.ERROR);

        alert.setTitle("Password Mismatch");

        alert.setHeaderText("Password Mismatch");

        alert.setContentText("Passwords do not match");

        alert.showAndWait();

      }

    });

    Scene scene = new Scene(resVBox, 350, 600);

    primaryStage.setTitle("Reset Password");

    primaryStage.setScene(scene);

    primaryStage.show();

  }

  public void signUpScreen() {

    VBox newLayout = new VBox();

    newLayout.setSpacing(20);

    newLayout.getStyleClass().add("root2");

    Button backButton = new Button("<-");

    backButton.getStyleClass().add("backButtonStyle");

    backButton.setOnAction(e -> AcessPage());

    // Create a Title Bar with the Back Button

    HBox topBar = new HBox();

    // topBar.setAlignment(Pos.CENTER\_LEFT);

    topBar.setSpacing(10); // Add some spacing if needed

    topBar.setPadding(new Insets(10, 10, 10, 10)); // Padding for the top bar

    topBar.getChildren().add(backButton);

    // newLayout.add(topBar, 0, 0, 2, 1); // Span the title bar across columns

    VBox.setMargin(topBar, new Insets(0, 0, 0, 0));

    // Name

    Label nameClient = new Label("Name");

    nameClient.getStyleClass().add("LabelStyling");

    TextField nameClientField = new TextField();

    nameClientField.setPromptText("Enter your username");

    // hbox for name

    HBox nameBox = new HBox(nameClient, nameClientField);

    nameBox.setSpacing(10);

    // password

    Label passwordLabel1 = new Label("Password");

    passwordLabel1.getStyleClass().add("LabelStyling");

    passwordLabel1.setId("passwordLabel1");

    PasswordField passwordField1 = new PasswordField();

    passwordField1.setPromptText("Enter your new passsword");

    HBox passwordBox = new HBox(passwordLabel1, passwordField1);

    // Email

    Label emailClient = new Label("email");

    emailClient.getStyleClass().add("LabelStyling");

    TextField emailClientField = new TextField();

    emailClientField.setPromptText("Enter your email");

    HBox emailBox = new HBox(emailClient, emailClientField);

    emailBox.setSpacing(10);

    // Label for identification

    Label idenLabel = new Label("These are the Questions For Password Recovery Next Time");

    Label schoolLabel = new Label("what was the name of your first high School?");

    TextField schoolField = new TextField();

    Label movieLabel = new Label("What is your favorite childhood movie?");

    TextField movieField = new TextField();

    // button register

    Button registerButton = new Button("Register");

    newLayout.getChildren().addAll(topBar, nameBox, passwordBox, emailBox, idenLabel, schoolLabel, schoolField,

        movieLabel, movieField, registerButton);

    registerButton.setOnAction(a -> {

      String regName = nameClientField.getText().trim();

      String regPassword = passwordField1.getText().trim();

      String regEmail = emailClientField.getText().trim();

      String school = schoolField.getText().trim();

      String movie = movieField.getText().trim();

      if (regName.contains("~") || regPassword.contains("~") || regEmail.contains("~") || school.contains("~")

          || movie.contains("~")) {

        Alert alert = new Alert(Alert.AlertType.ERROR);

        alert.setTitle("Input Violation");

        alert.setHeaderText("Invalid Entry");

        alert.setContentText("The symbol '~' is not allowed in any field.");

        alert.showAndWait();

        return;

      }

      if (!(dataFile.authenticate(regEmail))) {

        userInfo.add(new UserData(regName, regPassword, regEmail, school, movie));

        dataFile.newFileWriter(userInfo);

        // Alert

        Alert alertRegister = new Alert(Alert.AlertType.CONFIRMATION);

        alertRegister.setTitle("REGISTRATION");

        alertRegister.setHeaderText("MISSION PASSED");

        alertRegister.setContentText("Register successfully");

        alertRegister.showAndWait();

      } else {

        Alert alertRegister = new Alert(Alert.AlertType.ERROR);

        alertRegister.setTitle("REGISTRATION Error");

        alertRegister.setContentText("User already registered");

        alertRegister.showAndWait();

      }

    });

    Scene signUpScene = new Scene(newLayout, 400, 550);

    signUpScene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

    System.out.println(getClass().getResource("style.css").toExternalForm());

    // Stage stage = new Stage();

    primaryStage.setScene(signUpScene);

    primaryStage.setTitle("Enter required Data");

    primaryStage.show();

  }

  public void properApp() {

    // Create a VBox for scrollable content

    VBox vBox = new VBox();

    vBox.setAlignment(Pos.TOP\_CENTER);

    vBox.setStyle(" -fx-background-color:rgb(17, 17, 28)");

    vBox.setSpacing(10);

    // Back Button

    Button backButton = new Button("<-");

    backButton.getStyleClass().add("backButtonStyle");

    backButton.setOnAction(e -> AcessPage());

    // Main Label

    Label mainLabel = new Label("What's Your \n Main Goal");

    mainLabel.setId("mainLabel");

    // Add Images (Diet Images)

    Image dietImg1 = new Image(getClass().getResource("calCalculator.jpg").toExternalForm());

    ImageView dietImageView1 = new ImageView(dietImg1);

    dietImageView1.setFitHeight(160);

    dietImageView1.setFitWidth(350);

    Label dieLabel1 = new Label("Calorie Calculator");

    StackPane dieStackPane1 = new StackPane();

    dieStackPane1.getChildren().addAll(dietImageView1, dieLabel1);

    dieStackPane1.setOnMouseClicked(e -> {

      calorieCal();

    });

    Image dietImg2 = new Image(getClass().getResource("workout.jpeg").toExternalForm());

    ImageView dietImageView2 = new ImageView(dietImg2);

    dietImageView2.setFitHeight(160);

    dietImageView2.setFitWidth(350);

    Label dieLabel2 = new Label("Workout videos");

    StackPane dieStackPane2 = new StackPane();

    dieStackPane2.getChildren().addAll(dietImageView2, dieLabel2);

    dieStackPane2.setOnMouseClicked(o -> {

      VideoHandler videoHandler=new VideoHandler(this);

      try{

        videoHandler.start(primaryStage);

      }catch (Exception e){

        System.out.println(e.getMessage());

      }

    });

    Image dietImg3 = new Image(getClass().getResource("workoutPlan.png").toExternalForm());

    ImageView dietImageView3 = new ImageView(dietImg3);

    dietImageView3.setFitHeight(160);

    dietImageView3.setFitWidth(350);

    Label dieLabel3 = new Label("30 day workout plan");

    StackPane dieStackPane3 = new StackPane();

    dieStackPane3.getChildren().addAll(dietImageView3, dieLabel3);

    vBox.getChildren().addAll(mainLabel, dieStackPane1, dieStackPane2, dieStackPane3);

    //styling of label

    dieLabel1.getStyleClass().add("properAppLabel");

    dieLabel2.getStyleClass().add("properAppLabel");

    dieLabel3.getStyleClass().add("properAppLabel");

    //styling of images

    dietImageView1.setId("dietImageView1");

    dietImageView2.setId("dietImageView2");

    dietImageView3.setId("dietImageVie3");

    // Add VBox to ScrollPane

    ScrollPane scrollPane = new ScrollPane(vBox);

    scrollPane.setId("PropScrollPane");

    scrollPane.setFitToWidth(true);

    scrollPane.setVbarPolicy(ScrollPane.ScrollBarPolicy.ALWAYS);

    BorderPane borderPane = new BorderPane();

    borderPane.setStyle(" -fx-background-color:rgb(17, 17, 28)");

    borderPane.setTop(backButton);

    borderPane.setCenter(scrollPane); // Scrollable content

    // Create a Scene with the BorderPane

    Scene scene = new Scene(borderPane, 400, 550);

    scene.getStylesheets().add(getClass().getResource("style.css").toExternalForm());

    primaryStage.setScene(scene);

    primaryStage.setTitle("Proper App Layout");

    primaryStage.show();

    // primaryStage.setFullScreen(true);

  }

  public void calorieCal() {

     // Back Button

     Button backButton = new Button("Back");

     backButton.getStyleClass().add("backButtonStyle");

     backButton.setOnAction(e -> properApp());

    // String sex, int age, int feet, int inches, int lbs, String activityLevel

    Label genderLabel = new Label("Gender");

    ToggleGroup toggleGroup = new ToggleGroup();

    RadioButton maleButton = new RadioButton("male");

    maleButton.setToggleGroup(toggleGroup);

    RadioButton femaleButton = new RadioButton("female");

    femaleButton.setToggleGroup(toggleGroup);

    Label age = new Label("Age");

    TextField ageField = new TextField();

    Label feetLabel = new Label("Feet");

    TextField feetField = new TextField();

    Label inchesLabel = new Label("Inches");

    TextField inchesField = new TextField();

    Label lbsLabel = new Label("Lbs");

    TextField lbsField = new TextField();

    Label activityLevelLabel = new Label("Activity Level");

    ComboBox<String> ActivityLevel = new ComboBox<>();

    ActivityLevel.getItems().addAll( "Lightly Active", "Moderately Active", "Active", "Super Active");

    Button submitDataBtn = new Button("Check");

    // Cast the selected toggle to RadioButton and get its text

    submitDataBtn.setOnAction(e -> {

      try {

        // Collect user input

        Toggle selectedToggle = toggleGroup.getSelectedToggle();

        if (selectedToggle == null) {

          System.out.println("Please select a gender.");

          return;

        }

        RadioButton selectedRadioButton = (RadioButton) selectedToggle;

        String gender = selectedRadioButton.getText();

        int userAge = Integer.parseInt(ageField.getText());

        int userFeet = Integer.parseInt(feetField.getText());

        int userInches = Integer.parseInt(inchesField.getText());

        int userlbs = Integer.parseInt(lbsField.getText());

        String activityString = ActivityLevel.getValue();

        // Call the API

        String result = NutritionAPI.getNutritionInfo(gender, userAge, userFeet, userInches, userlbs, activityString);

        // Parse the result

        if (result.startsWith("Error")) {

          System.out.println(result);

          return;

        }

        // Extract specific data using NutritionDataHandler

        String calories = NutritionDataHandler.extractCalorieNeeds(new JSONObject(result));

        String bmi = NutritionDataHandler.extractBMI(new JSONObject(result));

        String waterIntake = NutritionDataHandler.extractWaterIntake(new JSONObject(result));

      //back button

        Button backButton1 = new Button("Back");

        backButton.getStyleClass().add("backButtonStyle");

        backButton.setOnAction(t -> properApp());

        // Update UI with extracted data

        Label calorieLabel = new Label("Calories: " + calories);

        Label bmiLabel = new Label("BMI: " + bmi);

        Label waterLabel = new Label("Water Intake: " + waterIntake);

        VBox resultVBox = new VBox(15,backButton1,calorieLabel, bmiLabel, waterLabel);

        resultVBox.setAlignment(Pos.CENTER);

        Scene resultScene = new Scene(resultVBox, 300, 200);

        primaryStage.setScene(resultScene);

        primaryStage.show();

      } catch (Exception ex) {

        ex.printStackTrace();

        System.out.println("Please fill out all fields correctly.");

      }

    });

    VBox calVBox = new VBox(15);

    calVBox.setAlignment(Pos.CENTER);

    calVBox.getChildren().addAll(backButton,genderLabel, maleButton, femaleButton, age, ageField, feetLabel, feetField,

        inchesLabel, inchesField, lbsLabel, lbsField, activityLevelLabel, ActivityLevel, submitDataBtn);

    Scene newsScene = new Scene(calVBox,350,600);

    primaryStage.setScene(newsScene);

    primaryStage.show();

    // primaryStage.setFullScreen(true);

  }

  public static void main(String[] args) {

    launch();

  }

}

**In this class we are handling all types of sounds**

package com.example;

// import javafx.application.Platform;

import javafx.scene.media.AudioClip;

public class Sound {

    public void soundRun() {

        try {

            String soundFile = getClass().getResource("/sound/btnSound.wav").toString();

             AudioClip notificationSound = new AudioClip(soundFile);

            notificationSound.play();

        } catch (Exception e) {

            System.err.println("Error: " + e.getMessage());

        }

    }

}

**In This class We are handling all types of videos**

package com.example;

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.HBox;

import javafx.scene.layout.VBox;

import javafx.scene.media.Media;

import javafx.scene.media.MediaPlayer;

import javafx.scene.media.MediaView;

import javafx.stage.Stage;

import javafx.util.Duration;

public class VideoHandler extends Application {

Stage primaryStage;

App app;

public  VideoHandler(App app1){

    this.app = app1;

}

    @Override

    public void start(Stage primaryStage)  {

        this.primaryStage = primaryStage;

        videoMenu();

    }

    public void videoMenu(){

    Button backBtn=new Button("Back");

    backBtn.setOnAction(e -> {

        app.properApp();

    });

        Label abs=new Label("Abs");

        HBox absBox=new HBox(abs);

       absBox.setAlignment(Pos.CENTER);

       abs.setOnMouseClicked(y->{

           absVideos();

       });

        Label back=new Label("Back");

        HBox backBox=new HBox(back);

        backBox.setAlignment(Pos.CENTER);

        Label bicep=new Label("Biceps");

        HBox bicepBox=new HBox(bicep);

        bicepBox.setAlignment(Pos.CENTER);

        Label calf=new Label("Calf");

        HBox calfBox=new HBox(calf);

        calfBox.setAlignment(Pos.CENTER);

        Label chest=new Label("Chest");

        HBox chestBox=new HBox(chest);

        chestBox.setAlignment(Pos.CENTER);

        Label forearm=new Label("Forearm");

        HBox foreBox=new HBox(forearm);

        foreBox.setAlignment(Pos.CENTER);

        VBox videoMenuVbox=new VBox();

        videoMenuVbox.setSpacing(15);

        videoMenuVbox.setAlignment(Pos.CENTER);

        videoMenuVbox.getChildren().addAll(backBtn,absBox,backBox,bicepBox,calfBox,chestBox,foreBox);

        Scene scene=new Scene(videoMenuVbox,350,600);

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public  void absVideos(){

        Button backButton = new Button("<-");

        backButton.getStyleClass().add("backButtonStyle");

        backButton.setOnAction(e -> videoMenu());

        Label videoLabel = new Label("videos");

        String videoUrl = getClass().getResource("/video/abs1.mp4").toExternalForm();

        System.out.println(videoUrl);

        Media media = new Media(videoUrl);

        MediaPlayer mediaPlayer = new MediaPlayer(media);

        MediaView mediaView = new MediaView(mediaPlayer);

        mediaPlayer.play();

        mediaPlayer.setOnEndOfMedia(() -> mediaPlayer.seek(Duration.ZERO));

        // control height and width

        mediaView.setFitHeight(200);

        mediaView.setFitWidth(200);

        mediaView.setOnMouseClicked(j -> {

            if (mediaPlayer.getStatus() == MediaPlayer.Status.PLAYING) {

                mediaPlayer.pause();

            } else {

                mediaPlayer.play();

            }

        });

        VBox videoVBox = new VBox();

        videoVBox.setStyle("-fx-background-color: black");

        videoVBox.setId("videoVBox");

        videoVBox.getChildren().addAll(backButton, mediaView, videoLabel);

        videoVBox.setAlignment(Pos.CENTER);

        Scene scene = new Scene(videoVBox, 350, 600);

        primaryStage.setScene(scene);

    }

}

**In this Class we are handling all types of File reading,writing functions**

package com.example;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileWriter;

import java.io.IOException;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.Scanner;

import javafx.scene.Scene;

public class NewFile {

    File file;

    public NewFile() {

        file = new File("C:\\Users\\DELL\\Music\\table\_tut2\\src\\main\\resources\\com\\example", "Data.txt");

        try {

            file.createNewFile();

        } catch (Exception j) {

            System.out.println(j.getMessage());

        }

    }

    public ArrayList<UserData> readFile() {

        ArrayList<UserData> newUser = new ArrayList<>();

        try {

            Scanner scanner = new Scanner(file);

            while (scanner.hasNextLine()) {

                String line = scanner.nextLine();

                String[] parts = line.split("~");

                if (parts.length == 5) {

                    newUser.add(new UserData(parts[0], parts[1], parts[2], parts[3], parts[4]));

                }

            }

        } catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        return newUser;

    }

    public boolean checkUser(String email, String school, String movie) {

        try (Scanner scanner = new Scanner(file)) {

            while (scanner.hasNextLine()) {

                String line = scanner.nextLine();

                String[] parts = line.split("~");

                if (parts.length == 5) {

                    String fileUserEmail = parts[2];

                    String fileSchool = parts[3];

                    String fileMovie = parts[4];

                    if (fileUserEmail.equals(email) && fileSchool.equalsIgnoreCase(school)

                            && fileMovie.equalsIgnoreCase(movie)) {

                        return true;

                    }

                }

            }

        } catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        return false;

    }

    public void updataPassword(String email, String newPassword) {

        ArrayList<UserData> userArrayList = readFile();

        boolean check = false;

        for (UserData newUser : userArrayList) {

            if (newUser.getUserEmail().equals(email)) {

                newUser.setUserPassword(newPassword);

                check = true;

            }

        }

        if (check) {

            try (FileWriter fileWriter = new FileWriter(file, false)) {

                for (UserData userData : userArrayList) {

                    fileWriter.write(

                            userData.getUserName() + "~" + userData.getUserPassword() + "~" + userData.getUserEmail()

                                    + "~" + userData.getSchool() + "~" + userData.getMovie() + "\n");

                }

                fileWriter.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

    public void newFileWriter(ArrayList<UserData> user) {

        try (FileWriter fileWriter = new FileWriter(file, true)) {

            for (UserData userData : user) {

                try {

                    fileWriter.write(

                            userData.getUserName() + "~" + userData.getUserPassword() + "~" + userData.getUserEmail()

                                    + "~" + userData.getSchool() + "~" + userData.getMovie() + "\n");

                } catch (IOException e) {

                    System.out.println(e.getMessage());

                    e.printStackTrace();

                }

            }

            fileWriter.close();

            user.clear();

        } catch (IOException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

    }

    public boolean checkInFile(String name, String pass) throws Exception {

        Scanner scanner = new Scanner(file);

        while (scanner.hasNextLine()) {

            String line = scanner.nextLine();

            String[] parts = line.split("~");

            if (parts.length == 5) {

                String fileUsername = parts[0];

                String fileId = parts[1];

                if (fileUsername.equalsIgnoreCase(name) && fileId.equals(pass)) {

                    return true;

                }

            }

        }

        return false;

    }

    public boolean authenticate(String email) {

        try (Scanner scanner = new Scanner(file)) {

            while (scanner.hasNextLine()) {

                String line = scanner.nextLine();

                String parts[] = line.split("~");

                if (parts.length == 5) {

                    String newEmail = parts[2];

                    if (newEmail.equals(email)) {

                        return true;

                    }

                }

            }

        } catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        return false;

    }

}

**In this class we are handling all types userDetails**

package com.example;

public class UserData {

    // dataFile.writeInFile(regName, regPassword, regEmail, regAge);

    String userName;

    String userPassword;

    String userEmail;

    String school;

    String movie;

    public UserData(String userName, String userPassword, String userEmail, String school,

            String movie) {

        this.userName = userName;

        this.userPassword = userPassword;

        this.userEmail = userEmail;

        this.school = school;

        this.movie = movie;

    }

    public String getUserName() {

        return userName;

    }

    public String getUserPassword() {

        return userPassword;

    }

    public String getUserEmail() {

        return userEmail;

    }

    public void setUserName(String userName) {

        this.userName = userName;

    }

    public void setUserPassword(String userPassword) {

        this.userPassword = userPassword;

    }

    public void setUserEmail(String userEmail) {

        this.userEmail = userEmail;

    }

    public void setSchool(String school) {

        this.school = school;

    }

    public void setMovie(String movie) {

        this.movie = movie;

    }

    public String getSchool() {

        return school;

    }

    public String getMovie() {

        return movie;

    }

}

This is the class for help and support

package com.example;

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.control.TextArea;

import javafx.scene.layout.HBox;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.awt.\*;

import java.io.IOException;

import java.net.URI;

import java.net.URISyntaxException;

import java.net.URLEncoder;

import java.nio.charset.StandardCharsets;

public class HelpSupport extends Application{

    private App app;

    private Stage primaryStage;

    public HelpSupport(App app1) {

        this.app=app1;

    }

    @Override

    public void start(Stage stage)  {

        this.primaryStage=stage;

        helpMenu();

    }

    public void helpMenu(){

        Button faqs=new Button("Frequently Asked Questions");

        Button contactAt = new Button("Contact Us ");

        Button emailBtn = new Button("Send Email");

        VBox vBox=new VBox();

        vBox.getChildren().addAll(faqs,contactAt,emailBtn);

        vBox.setSpacing(20);

        vBox.setAlignment(Pos.CENTER);

        contactAt.setOnAction(o->{

            contactUs();

        });

        emailBtn.setOnAction(y->{

            openHelpSupport();

        });

        Scene scene=new Scene(vBox,350,600);

        primaryStage.setScene(scene);

        primaryStage.show();

    }

    public void openHelpSupport() {

        Button emailBtn = new Button("Send Email");

        Button backBtn = new Button("Back");

        // Text Area for Subject and Body

        Label subjectLabel = new Label("Subject:");

        TextArea subjectField = new TextArea();

        subjectField.setPromptText("Enter subject");

        subjectField.setWrapText(true);

        subjectField.setMaxHeight(50);

        Label bodyLabel = new Label("Body:");

        TextArea bodyField = new TextArea();

        bodyField.setPromptText("Enter body text");

        bodyField.setWrapText(true);

        VBox helpsupport = new VBox( subjectLabel, subjectField, bodyLabel, bodyField, emailBtn, backBtn);

        helpsupport.setSpacing(10);

        helpsupport.setAlignment(Pos.CENTER);

        Scene helpSupportScene = new Scene(helpsupport, 350, 600);

        // stage.setScene(helpSupportScene);

        // Back button functionality

        backBtn.setOnAction(e -> {

            app.AcessPage();

        });

        // Email button functionality

        emailBtn.setOnAction(e -> {

            String subject = subjectField.getText();

            String body = bodyField.getText();

            openGmail(subject, body);

        });

        primaryStage.setScene(helpSupportScene);

        primaryStage.show();

    }

    public void openGmail(String subject, String body) {

        String recipientEmail = "huzaifaliaquat62@gmail.com";

        // URL encode the subject and body to handle special characters

        String encodedSubject = URLEncoder.encode(subject, StandardCharsets.UTF\_8);

        String encodedBody = URLEncoder.encode(body, StandardCharsets.UTF\_8);

        // Construct the Gmail URL

        String gmailURL = "https://mail.google.com/mail/?view=cm&fs=1&to=" + recipientEmail + "&su=" + encodedSubject + "&body=" + encodedBody;

        if (Desktop.isDesktopSupported()) {

            Desktop desktop = Desktop.getDesktop();

             if (desktop.isSupported(Desktop.Action.BROWSE)) {

                try {

                    desktop.browse(new URI(gmailURL));

                } catch (IOException | URISyntaxException e) {

                    e.printStackTrace(); // Log the error for debugging

                }

            }

            else {

                System.out.println("BROWSE action is not supported!");

            }

        } else {

            System.out.println("Desktop is not supported on this platform.");

        }

    }

   public void contactUs(){

    Button backBtn = new Button("Back");

    backBtn.setOnAction(i->{

        helpMenu();

    });

    Label phoneLabel1 = new Label("03328307060");

    Label phoneLabel2 = new Label("03328307061");

    Label phoneLabel3 = new Label("03328307062");

    VBox contactVbox=new VBox();

    contactVbox.setAlignment(Pos.CENTER);

    contactVbox.getChildren().addAll(backBtn,phoneLabel1,phoneLabel2,phoneLabel3);

    Scene scene=new Scene(contactVbox,350,600);

    primaryStage.setScene(scene);

    primaryStage.show();

   }

}

**This is the class in which we are sending request to Api**

package com.example;

import java.net.HttpURLConnection;

import java.net.URL;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import org.json.JSONObject;

public class NutritionAPI {

    private static final String API\_URL = "https://nutrition-calculator.p.rapidapi.com/api/nutrition-info";

    private static final String API\_KEY = "1ffac417e4mshe27a3240bec784bp12b9edjsnfaa20d391036";

    public static String getNutritionInfo(String sex, int age, int feet, int inches, int lbs, String activityLevel) {

        try {

            // Construct the URL

            String urlString = String.format(

                    "%s?measurement\_units=std&sex=%s&age\_value=%d&age\_type=yrs&feet=%d&inches=%d&lbs=%d&activity\_level=%s",

                    API\_URL, sex, age, feet, inches, lbs, activityLevel);

            URL url = new URL(urlString);

            // Open connection

            HttpURLConnection connection = (HttpURLConnection) url.openConnection();

            connection.setRequestMethod("GET");

            connection.setRequestProperty("x-rapidapi-key", API\_KEY);

            connection.setRequestProperty("x-rapidapi-host", "nutrition-calculator.p.rapidapi.com");

            // Check the response

            int responseCode = connection.getResponseCode();

            if (responseCode == HttpURLConnection.HTTP\_OK) {

                BufferedReader reader = new BufferedReader(new InputStreamReader(connection.getInputStream()));

                StringBuilder response = new StringBuilder();

                String line;

                while ((line = reader.readLine()) != null) {

                    response.append(line);

                }

                reader.close();

                // Parse and return the response

                JSONObject jsonResponse = new JSONObject(response.toString());

                return jsonResponse.toString(2); // Pretty print JSON

            } else {

                return "Error: " + responseCode;

            }

        } catch (Exception e) {

            e.printStackTrace();

            return "Error: " + e.getMessage();

        }

    }

}

**In this class we are handling the data given from Api**

package com.example;

import org.json.JSONArray;

import org.json.JSONObject;

import java.util.ArrayList;

import java.util.List;

public class NutritionDataHandler {

    public static void getHandledData(String result) {

        String jsonResponse = result; // API JSON response

        try {

            JSONObject jsonObject = new JSONObject(jsonResponse);

            // Extract Calorie Needs and BMI

            String calorieNeeds = extractCalorieNeeds(jsonObject);

            System.out.println("Estimated Daily Caloric Needs: " + calorieNeeds);

            String bmi = extractBMI(jsonObject);

            System.out.println("BMI: " + bmi);

            // Extract Macronutrients

            List<String> macronutrients = extractMacronutrients(jsonObject);

            System.out.println("Macronutrients: " + macronutrients);

            // Extract Water Intake

            String water = extractWaterIntake(jsonObject);

            System.out.println("Water Intake: " + water);

            // Extract Key Vitamins (e.g., Vitamin A, C, D, and B12)

            List<String> keyVitamins = extractKeyVitamins(jsonObject);

            System.out.println("Key Vitamins: " + keyVitamins);

            // Extract Key Minerals (e.g., Calcium, Iron, Potassium, and Zinc)

            List<String> keyMinerals = extractKeyMinerals(jsonObject);

            System.out.println("Key Minerals: " + keyMinerals);

        } catch (Exception e) {

            e.printStackTrace();

        }

    }

    public static String extractCalorieNeeds(JSONObject jsonObject) {

        return jsonObject.getJSONObject("BMI\_EER").getString("Estimated Daily Caloric Needs");

    }

   public static String extractBMI(JSONObject jsonObject) {

        return jsonObject.getJSONObject("BMI\_EER").getString("BMI");

    }

   public static List<String> extractMacronutrients(JSONObject jsonObject) {

        List<String> macronutrientList = new ArrayList<>();

        JSONArray macronutrientsTable = jsonObject

                .getJSONObject("macronutrients\_table")

                .getJSONArray("macronutrients-table");

        for (int i = 1; i < macronutrientsTable.length(); i++) {

            JSONArray row = macronutrientsTable.getJSONArray(i);

            String nutrient = row.getString(0);

            if (nutrient.equalsIgnoreCase("Carbohydrate") ||

                nutrient.equalsIgnoreCase("Protein") ||

                nutrient.equalsIgnoreCase("Fat")) {

                macronutrientList.add(nutrient + ": " + row.getString(1));

            }

        }

        return macronutrientList;

    }

    public static String extractWaterIntake(JSONObject jsonObject) {

        JSONArray macronutrientsTable = jsonObject

                .getJSONObject("macronutrients\_table")

                .getJSONArray("macronutrients-table");

        for (int i = 1; i < macronutrientsTable.length(); i++) {

            JSONArray row = macronutrientsTable.getJSONArray(i);

            if (row.getString(0).equalsIgnoreCase("Total Water")) {

                return row.getString(1); // Return water intake

            }

        }

        return "Water intake not found";

    }

    public static List<String> extractKeyVitamins(JSONObject jsonObject) {

        List<String> vitaminsList = new ArrayList<>();

        JSONArray vitaminsTable = jsonObject

                .getJSONObject("vitamins\_table")

                .getJSONArray("vitamins-table");

        List<String> keyVitamins = List.of("Vitamin A", "Vitamin C", "Vitamin D", "Vitamin B12");

        for (int i = 1; i < vitaminsTable.length(); i++) {

            JSONArray row = vitaminsTable.getJSONArray(i);

            if (keyVitamins.contains(row.getString(0))) {

                vitaminsList.add(row.getString(0) + ": " + row.getString(1));

            }

        }

        return vitaminsList;

    }

   public static List<String> extractKeyMinerals(JSONObject jsonObject) {

        List<String> mineralsList = new ArrayList<>();

        JSONArray mineralsTable = jsonObject

                .getJSONObject("minerals\_table")

                .getJSONArray("essential-minerals-table");

        List<String> keyMinerals = List.of("Calcium", "Iron", "Potassium", "Zinc");

        for (int i = 1; i < mineralsTable.length(); i++) {

            JSONArray row = mineralsTable.getJSONArray(i);

            if (keyMinerals.contains(row.getString(0))) {

                mineralsList.add(row.getString(0) + ": " + row.getString(1));

            }

        }

        return mineralsList;

    }

}