|  |  |
| --- | --- |
| Muneeb - ur - Rehman | SP24-BSE-147 |
| 1. Taha Asim | SP24-BSE-088 |
| Husnain Mehmood | SP24-BSE-046 |

**OOP SEMESTER PROJECT**

**SECTION A**

**CODE:**

LOPGIN SIGNUP FORM:

Package com.example.semesterrproject;  
  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Cursor;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.stage.Stage;  
import javafx.util.Duration;  
import javafx.animation.PauseTransition;  
  
import java.io.\*;  
import java.util.HashMap;  
import java.util.Map;  
  
public class LoginSignupForm extends Application {  
  
 private Stage primaryStage;  
  
 private final String FILE\_PATH = "accounts.txt";  
 private Map<String, String> accounts = new HashMap<>();  
  
 @Override  
 public void start(Stage primaryStage) {  
  
 this.primaryStage = primaryStage;  
 loadAccounts();  
  
  
 BorderPane root = new BorderPane();  
  
  
 VBox leftPane = new VBox(10);  
 leftPane.setAlignment(Pos.*CENTER*);  
 leftPane.setStyle("-fx-background-color: #0078D7;");  
 leftPane.setPrefWidth(400);  
  
  
 ImageView logo = new ImageView(new Image("CarLogo.png"));  
 logo.setFitWidth(130);  
 logo.setFitHeight(100);  
  
 Label title = new Label("Car Rental");  
 title.setTextFill(Color.*WHITE*);  
 title.setFont(Font.*font*("Poppins", FontWeight.*BOLD*, 30));  
 title.setAlignment(Pos.*CENTER*);  
  
 StackPane formContainer = new StackPane();  
 formContainer.setPrefWidth(400);  
 formContainer.setStyle("-fx-background-color: #FFFFFF;");  
  
  
 VBox loginForm = createLoginForm(formContainer);  
  
  
 formContainer.getChildren().add(loginForm);  
  
 Button createAccountButton = new Button("Create Account");  
 Button backToLoginButton = new Button("Back to Login");  
  
  
 createAccountButton.setStyle("-fx-background-color: transparent; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 12px 35px 12px 35px; -fx-font-size: 18px; -fx-font-weight: 900; -fx-border-width: 2px; -fx-border-color: #fff; -fx-border-radius: 5px;");  
 createAccountButton.setOnMouseEntered(e -> createAccountButton.setCursor(Cursor.*HAND*));  
 createAccountButton.setOnMouseExited(e -> createAccountButton.setCursor(Cursor.*DEFAULT*));  
 createAccountButton.setOnAction(e -> {  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(createSignupForm(formContainer));  
 leftPane.getChildren().remove(createAccountButton);  
 leftPane.getChildren().add(backToLoginButton);  
 });  
  
  
 backToLoginButton.setStyle("-fx-background-color: transparent; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 12px 35px 12px 35px; -fx-font-size: 18px; -fx-font-weight: 900; -fx-border-width: 2px; -fx-border-color: #fff; -fx-border-radius: 5px;");  
 backToLoginButton.setOnMouseEntered(e -> backToLoginButton.setCursor(Cursor.*HAND*));  
 backToLoginButton.setOnMouseExited(e -> backToLoginButton.setCursor(Cursor.*DEFAULT*));  
 backToLoginButton.setOnAction(e -> {  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(createLoginForm(formContainer));  
 leftPane.getChildren().remove(backToLoginButton);  
 leftPane.getChildren().add(createAccountButton);  
 });  
  
 leftPane.getChildren().addAll(logo, title, createAccountButton);  
 VBox.*setMargin*(createAccountButton, new Insets(30, 0, 0, 0));  
 VBox.*setMargin*(backToLoginButton, new Insets(30, 0, 0, 0));  
  
 root.setLeft(leftPane);  
 root.setCenter(formContainer);  
  
 Scene scene = new Scene(root, 800, 600);  
  
 primaryStage.setTitle("Login, Signup Form");  
 primaryStage.setScene(scene);  
 primaryStage.setMaxHeight(600);  
 primaryStage.setMaxWidth(800);  
 primaryStage.show();  
 }  
  
 private VBox createLoginForm(StackPane formContainer) {  
 VBox loginForm = new VBox(25);  
 loginForm.setPadding(new Insets(30, 70, 30, 70));  
 loginForm.setAlignment(Pos.*CENTER*);  
  
 ImageView usersimage = new ImageView(new Image("userslogin.png"));  
 usersimage.setFitWidth(50);  
 usersimage.setFitHeight(45);  
 Label loginTitle = new Label("User Login");  
 loginTitle.setFont(Font.*font*("Poppins", 25));  
 loginTitle.setStyle("-fx-text-fill: #0078D7; -fx-font-weight: 600;");  
  
 VBox fieldsbox = new VBox(25);  
  
 HBox usernamebox = new HBox(15);  
 ImageView userimage = new ImageView("userimage.png");  
 userimage.setFitWidth(20);  
 userimage.setFitHeight(20);  
 userimage.setPreserveRatio(true);  
 TextField usernameField = new TextField();  
 usernameField.setPromptText("Username");  
 usernameField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 usernamebox.getChildren().addAll(userimage, usernameField);  
 HBox.*setHgrow*(usernameField, Priority.*ALWAYS*);  
 usernamebox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
  
 HBox passwordbox = new HBox(15);  
 ImageView passimage = new ImageView("lockimage.png");  
 passimage.setFitWidth(18);  
 passimage.setFitHeight(20);  
 passimage.setPreserveRatio(true);  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPromptText("Password");  
 passwordField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 passwordbox.getChildren().addAll(passimage, passwordField);  
 HBox.*setHgrow*(passwordField, Priority.*ALWAYS*);  
 passwordbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
  
 HBox showpassfieldbox = new HBox(15);  
 ImageView showpassimage = new ImageView("lockimage.png");  
 showpassimage.setFitWidth(18);  
 showpassimage.setFitHeight(20);  
 showpassimage.setPreserveRatio(true);  
  
 HBox showpassbox = new HBox();  
 CheckBox showpass = new CheckBox("Show Password");  
 showpass.setStyle("-fx-font-size: 16px");  
 TextField showpassField = new TextField();  
 showpassField.setPromptText("Show Password");  
 showpassField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 showpassfieldbox.setVisible(false);  
 showpassfieldbox.setManaged(false);  
  
  
 showpassfieldbox.getChildren().addAll(showpassimage, showpassField);  
 HBox.*setHgrow*(showpassField, Priority.*ALWAYS*);  
 showpassfieldbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
 showpass.setOnAction(e -> {  
 if (showpass.isSelected()) {  
 showpassField.setText(passwordField.getText());  
 showpassfieldbox.setVisible(true);  
 showpassfieldbox.setManaged(true);  
 passwordbox.setVisible(false);  
 passwordbox.setManaged(false);  
 }  
 else {  
 passwordField.setText(showpassField.getText());  
 showpassfieldbox.setVisible(false);  
 showpassfieldbox.setManaged(false);  
 passwordbox.setVisible(true);  
 passwordbox.setManaged(true);  
 }  
 });  
 showpassbox.getChildren().addAll(showpass);  
  
 fieldsbox.getChildren().addAll(usernamebox, passwordbox, showpassfieldbox);  
  
  
  
 HBox forgotPasswordHBox = new HBox();  
 Button forgotPasswordButton = new Button("Forgot your Password?");  
 forgotPasswordButton.setStyle("-fx-background-color: transparent; -fx-text-fill: #0078D7; -fx-font-size: 16px; -fx-padding: 0px;");  
 forgotPasswordButton.setOnMouseEntered(e -> forgotPasswordButton.setCursor(Cursor.*HAND*));  
 forgotPasswordHBox.setOnMouseExited(e -> forgotPasswordButton.setCursor(Cursor.*DEFAULT*));  
 forgotPasswordButton.setOnAction(e -> {  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(createForgotPasswordForm(formContainer));  
 });  
 forgotPasswordHBox.getChildren().addAll(forgotPasswordButton);  
  
 Button loginButton = new Button("Login");  
 loginButton.setStyle("-fx-background-color: #0078D7; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 13px 0 13px 0; -fx-font-size: 18px; -fx-font-weight: 900;");  
 loginButton.prefWidthProperty().bind(loginForm.widthProperty());  
 loginButton.setOnMouseEntered(e -> loginButton.setCursor(Cursor.*HAND*));  
 loginButton.setOnMouseExited(e -> loginButton.setCursor(Cursor.*DEFAULT*));  
 loginButton.setOnAction(e -> {  
 String username = usernameField.getText();  
 String password = passwordField.isVisible() ? passwordField.getText() : showpassField.getText();  
  
 if (username.isEmpty() || password.isEmpty()) {  
 showMessage(formContainer, "Username and Password cannot be empty", Color.*RED*);  
 } else if (accounts.containsKey(username) && accounts.get(username).equals(password)) {  
 showMessage(formContainer, "Login Successful!", Color.*GREEN*);  
  
 // Transition to the CarRentalUI stage  
 PauseTransition pause = new PauseTransition(Duration.*seconds*(1.5)); // Show success message briefly  
 pause.setOnFinished(ev -> {  
 this.primaryStage.close();  
 new Dashboard().start(new Stage());  
 });  
 pause.play();  
 } else {  
 showMessage(formContainer, "Invalid Username or Password!", Color.*RED*);  
 }  
 });  
  
  
 loginForm.getChildren().addAll(usersimage, loginTitle, fieldsbox, showpassbox, loginButton,forgotPasswordHBox);  
 return loginForm;  
 }  
  
 private VBox createSignupForm(StackPane formContainer) {  
 VBox signupForm = new VBox(25);  
 signupForm.setPadding(new Insets(30, 70, 30, 70));  
 signupForm.setAlignment(Pos.*CENTER*);  
  
 ImageView accountimage = new ImageView(new Image("accountimage.png"));  
 accountimage.setFitWidth(50);  
 accountimage.setFitHeight(53);  
 Label signupTitle = new Label("Create an Account");  
 signupTitle.setFont(Font.*font*("Poppins", 25));  
 signupTitle.setStyle("-fx-text-fill: #0078D7; -fx-font-weight: 600;");  
  
 HBox accountusernamebox = new HBox(15);  
 ImageView userimage = new ImageView("userimage.png");  
 userimage.setFitWidth(20);  
 userimage.setFitHeight(20);  
 userimage.setPreserveRatio(true);  
 TextField usernameField = new TextField();  
 usernameField.setPromptText("Username");  
 usernameField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 accountusernamebox.getChildren().addAll(userimage, usernameField);  
 HBox.*setHgrow*(usernameField, Priority.*ALWAYS*);  
 accountusernamebox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
 HBox accountemailbox = new HBox(15);  
 ImageView mailimage = new ImageView("mailimage.png");  
 mailimage.setFitWidth(22);  
 mailimage.setFitHeight(17);  
 mailimage.setPreserveRatio(true);  
 TextField emailField = new TextField();  
 emailField.setPromptText("Email");  
 emailField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 accountemailbox.getChildren().addAll(mailimage, emailField);  
 HBox.*setHgrow*(emailField, Priority.*ALWAYS*);  
 accountemailbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
  
 HBox accountpasswordbox = new HBox(15);  
 ImageView passimage = new ImageView("lockimage.png");  
 passimage.setFitWidth(18);  
 passimage.setFitHeight(20);  
 passimage.setPreserveRatio(true);  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPromptText("Password");  
 passwordField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 accountpasswordbox.getChildren().addAll(passimage, passwordField);  
 HBox.*setHgrow*(passwordField, Priority.*ALWAYS*);  
 accountpasswordbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
  
 HBox showpassfieldbox = new HBox(15);  
 ImageView showpassimage = new ImageView("lockimage.png");  
 showpassimage.setFitWidth(18);  
 showpassimage.setFitHeight(20);  
 showpassimage.setPreserveRatio(true);  
  
 HBox showpassbox = new HBox();  
 CheckBox showpass = new CheckBox("Show Password");  
 showpass.setStyle("-fx-font-size: 16px");  
 TextField showpassField = new TextField();  
 showpassField.setPromptText("Show Password");  
 showpassField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 showpassfieldbox.setVisible(false);  
 showpassfieldbox.setManaged(false);  
  
  
 showpassfieldbox.getChildren().addAll(showpassimage, showpassField);  
 HBox.*setHgrow*(showpassField, Priority.*ALWAYS*);  
 showpassfieldbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
 showpass.setOnAction(e -> {  
 if (showpass.isSelected()) {  
 showpassField.setText(passwordField.getText());  
 showpassfieldbox.setVisible(true);  
 showpassfieldbox.setManaged(true);  
 accountpasswordbox.setVisible(false);  
 accountpasswordbox.setManaged(false);  
 }  
 else {  
 passwordField.setText(showpassField.getText());  
 showpassfieldbox.setVisible(false);  
 showpassfieldbox.setManaged(false);  
 accountpasswordbox.setVisible(true);  
 accountpasswordbox.setManaged(true);  
 }  
 });  
 showpassbox.getChildren().addAll(showpass);  
  
  
 Button signupButton = new Button("Sign Up");  
 signupButton.setStyle("-fx-background-color: #0078D7; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 13px 0 13px 0; -fx-font-size: 18px; -fx-font-weight: 900;");  
 signupButton.prefWidthProperty().bind(signupForm.widthProperty());  
 signupButton.setOnAction(e -> {  
 String username = usernameField.getText();  
 String email = emailField.getText();  
 String password = passwordField.isVisible() ? passwordField.getText() : showpassField.getText();  
  
 if (username.isEmpty() || email.isEmpty() ||password.isEmpty() ) {  
 showMessage(formContainer, "All fields must be filled!", Color.*RED*);  
 }  
 else if (password.length() < 8) {  
 showMessage(formContainer, "Password must be at least 8 characters long!", Color.*RED*);  
 }  
 else if(!email.matches("^[\\w.%+-]+@[\\w.-]+\\.[a-zA-Z]{2,6}$")){  
 showMessage(formContainer, "Email address format is invalid!", Color.*RED*);  
 }  
 else if (accounts.containsKey(username)) {  
 showMessage(formContainer, "Username already exists!", Color.*RED*);  
 } else {  
 accounts.put(username, password);  
 saveAccounts();  
 showMessage(formContainer, "Account Created Successfully!", Color.*GREEN*);  
 }  
 });  
  
 signupButton.setOnMouseEntered(e -> signupButton.setCursor(Cursor.*HAND*));  
 signupButton.setOnMouseExited(e -> signupButton.setCursor(Cursor.*DEFAULT*));  
  
  
  
 signupForm.getChildren().addAll(accountimage, signupTitle, accountusernamebox, accountemailbox, accountpasswordbox, showpassfieldbox, showpassbox, signupButton);  
 return signupForm;  
 }  
  
 private VBox createForgotPasswordForm(StackPane formContainer) {  
 VBox forgotPasswordForm = new VBox(25);  
 forgotPasswordForm.setPadding(new Insets(50));  
 forgotPasswordForm.setAlignment(Pos.*CENTER*);  
  
 ImageView resetPasswordimage = new ImageView(new Image("resetpassword.png"));  
 resetPasswordimage.setFitWidth(50);  
 resetPasswordimage.setFitHeight(50);  
 Label forgotPasswordTitle = new Label("Reset Password");  
 forgotPasswordTitle.setFont(Font.*font*("Poppins", 25));  
 forgotPasswordTitle.setStyle("-fx-text-fill: #0078D7; -fx-font-weight: 600;");  
  
  
 HBox usernamebox = new HBox(15);  
 ImageView userimage = new ImageView("userimage.png");  
 userimage.setFitWidth(20);  
 userimage.setFitHeight(20);  
 userimage.setPreserveRatio(true);  
 TextField usernameField = new TextField();  
 usernameField.setPromptText("Username");  
 usernameField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 usernamebox.getChildren().addAll(userimage, usernameField);  
 HBox.*setHgrow*(usernameField, Priority.*ALWAYS*);  
 usernamebox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
 HBox passwordbox = new HBox(15);  
 ImageView passimage = new ImageView("lockimage.png");  
 passimage.setFitWidth(18);  
 passimage.setFitHeight(20);  
 passimage.setPreserveRatio(true);  
 PasswordField newPasswordField = new PasswordField();  
 newPasswordField.setPromptText("New Password");  
 newPasswordField.setStyle("-fx-border-color: #0078D7; -fx-border-radius: 2; -fx-padding: 10; -fx-font-size: 15px; -fx-border-width: 0 0 3 0; -fx-background-color: transparent;");  
 passwordbox.getChildren().addAll(passimage, newPasswordField);  
 HBox.*setHgrow*(newPasswordField, Priority.*ALWAYS*);  
 passwordbox.setAlignment(Pos.*BOTTOM\_LEFT*);  
  
 Button resetPasswordButton = new Button("Reset Password");  
 resetPasswordButton.setStyle("-fx-background-color: #0078D7; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 13px 0 13px 0; -fx-font-size: 18px; -fx-font-weight: 900;");  
 resetPasswordButton.prefWidthProperty().bind(forgotPasswordForm.widthProperty());  
 resetPasswordButton.setOnAction(e -> {  
 String username = usernameField.getText();  
 String newPassword = newPasswordField.getText();  
  
 if (accounts.containsKey(username)) {  
 accounts.put(username, newPassword);  
 saveAccounts();  
 showMessage(formContainer, "Password Reset Successfully!", Color.*GREEN*);  
 } else {  
 showMessage(formContainer, "Username not found!", Color.*RED*);  
 }  
 });  
 resetPasswordButton.setOnMouseEntered(e -> resetPasswordButton.setCursor(Cursor.*HAND*));  
 resetPasswordButton.setOnMouseExited(e -> resetPasswordButton.setCursor(Cursor.*DEFAULT*));  
  
 Button backToLoginButton = new Button("Back to Login");  
 backToLoginButton.setStyle("-fx-background-color: transparent; -fx-text-fill: #0078D7; -fx-font-size: 15px;");  
 backToLoginButton.setOnAction(e -> {  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(createLoginForm(formContainer));  
 });  
 backToLoginButton.setOnMouseEntered(e -> backToLoginButton.setCursor(Cursor.*HAND*));  
 backToLoginButton.setOnMouseExited(e -> backToLoginButton.setCursor(Cursor.*DEFAULT*));  
  
 VBox.*setMargin*(resetPasswordButton, new Insets(20, 0,0 ,0 ));  
  
 forgotPasswordForm.getChildren().addAll(resetPasswordimage, forgotPasswordTitle, usernamebox, passwordbox, resetPasswordButton, backToLoginButton);  
 return forgotPasswordForm;  
  
 }  
  
  
 private void loadAccounts() {  
 try (BufferedReader reader = new BufferedReader(new FileReader(FILE\_PATH))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(":");  
 if (parts.length == 2) {  
 accounts.put(parts[0], parts[1]);  
 }  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error loading accounts: " + e.getMessage());  
 }  
 }  
  
 private void saveAccounts() {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE\_PATH))) {  
 for (Map.Entry<String, String> entry : accounts.entrySet()) {  
 writer.write(entry.getKey() + ":" + entry.getValue());  
 writer.newLine();  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error saving accounts: " + e.getMessage());  
 }  
 }  
  
 private void showMessage(StackPane formContainer, String message, Color color) {  
 Label confirmationMessage = new Label(message);  
 confirmationMessage.setFont(Font.*font*("Arial", FontWeight.*BOLD*, 16));  
 confirmationMessage.setTextFill(color);  
  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(confirmationMessage);  
  
 PauseTransition pause = new PauseTransition(Duration.*seconds*(2));  
 pause.setOnFinished(e -> {  
 formContainer.getChildren().clear();  
 formContainer.getChildren().add(createLoginForm(formContainer));  
 });  
 pause.play();  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

DASHBOARD

package com.example.semesterrproject;  
  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Cursor;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.ScrollBar;  
import javafx.scene.control.ScrollPane;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.scene.control.Label;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.stage.Screen;  
import javafx.stage.Stage;  
import java.util.List;  
  
public class Dashboard extends Application {  
  
 private CarsList carsList;  
 private Stage primaryStage;  
  
 @Override  
 public void start(Stage primaryStage) {  
 this.primaryStage = primaryStage;  
  
 carsList = new CarsList();  
 List<Car> availableCars = carsList.getAvailableCars();  
  
 BorderPane mainLayout = new BorderPane();  
 mainLayout.setPadding(new Insets(20));  
 BorderPane.*setMargin*(mainLayout, new Insets(0,0,20,0));  
  
 HBox headerbox = new HBox();  
 Label header = new Label("Car Rental Management System");  
 header.setFont(new Font("Arial", 25));  
 header.setStyle("-fx-font-weight: bold; -fx-text-fill: #000; -fx-padding: 10;");  
 headerbox.getChildren().add(header);  
 headerbox.setAlignment(Pos.*CENTER*);  
  
 mainLayout.setTop(headerbox);  
  
 VBox sidebar = new VBox(20);  
 sidebar.setPadding(new Insets(20));  
 sidebar.setStyle("-fx-background-color: #202020; -fx-pref-width: 125px; -fx-border-radius: 20px; -fx-background-radius: 20px;");  
  
 ImageView logo = new ImageView("CarLogo.png");  
 logo.setFitHeight(80);  
 logo.setFitWidth(100);  
  
 VBox sidebarIcons = new VBox(50);  
  
 ImageView carlistlogo = new ImageView("carlist.png");  
 carlistlogo.setFitHeight(40);  
 carlistlogo.setFitWidth(40);  
  
 ImageView bookedcars = new ImageView("booked.png");  
 bookedcars.setFitHeight(40);  
 bookedcars.setFitWidth(40);  
  
 ImageView userData = new ImageView("user.png");  
 userData.setFitHeight(40);  
 userData.setFitWidth(40);  
  
 sidebarIcons.getChildren().addAll(carlistlogo, bookedcars, userData);  
 sidebarIcons.setAlignment(Pos.*CENTER*);  
  
 VBox logoutBox = new VBox();  
 ImageView logout = new ImageView("logout.png");  
 logout.setFitHeight(35);  
 logout.setFitWidth(35);  
 logoutBox.getChildren().addAll(logout);  
 logoutBox.setAlignment(Pos.*CENTER*);  
  
 VBox.*setMargin*(logo, new Insets(0, 0, 110, 0));  
 VBox.*setMargin*(logoutBox, new Insets(130, 0, 0, 0));  
  
 sidebar.getChildren().addAll(logo, sidebarIcons, logoutBox);  
  
 // Create the central grid for cars  
 GridPane carGrid = new GridPane();  
 carGrid.setHgap(25);  
 carGrid.setVgap(25);  
 carGrid.setPadding(new Insets(30));  
  
  
 for (int i = 0; i < availableCars.size(); i++) {  
 Car car = availableCars.get(i);  
  
 VBox carCardbox = new VBox(10);  
 carCardbox.setAlignment(Pos.*CENTER*);  
 carCardbox.setStyle("-fx-background-color: #fff; -fx-border-radius: 10px; -fx-padding: 20; -fx-background-radius: 5; -fx-border-width: 1;");  
  
  
 Label carNameLabel = new Label(car.getName());  
 carNameLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
  
 Label carLocationLabel = new Label(car.getLocation());  
 carLocationLabel.setStyle("-fx-font-size: 14px;");  
  
 ImageView carImageView = new ImageView(car.getImageUrl());  
 carImageView.setFitWidth(230);  
 carImageView.setFitHeight(195);  
 carImageView.setPreserveRatio(true);  
  
 Button viewDetailsButton = new Button("View Details");  
 viewDetailsButton.setStyle("-fx-background-color: #0078D7; -fx-text-fill: white; -fx-background-radius: 5; -fx-padding: 12px 40px 12px 40px; -fx-font-size: 18px; -fx-font-weight: 900;");  
 viewDetailsButton.setOnAction(e -> {  
 primaryStage.close();  
 new CarDetailView(car).start(new Stage()); // Pass the selected car to the details view  
 });  
 viewDetailsButton.setOnMouseEntered(e -> viewDetailsButton.setCursor(Cursor.*HAND*));  
 viewDetailsButton.setOnMouseExited(e -> viewDetailsButton.setCursor(Cursor.*DEFAULT*));  
  
 DropShadow dropShadow = new DropShadow();  
 dropShadow.setRadius(20);  
 dropShadow.setOffsetY(5);  
 dropShadow.setOffsetX(5);  
 Color shadowColor = Color.*rgb*(0,0,0,0.2);  
 dropShadow.setColor(shadowColor);  
  
 carCardbox.setEffect(dropShadow);  
  
 carCardbox.getChildren().addAll(carNameLabel, carLocationLabel, carImageView, viewDetailsButton);  
  
 int column = i % 3;  
 int row = i / 3;  
 carGrid.add(carCardbox, column, row);  
 }  
  
 ScrollPane scrollPane = new ScrollPane(carGrid);  
 scrollPane.setFitToWidth(true);  
 scrollPane.setHbarPolicy(ScrollPane.ScrollBarPolicy.*NEVER*);  
 scrollPane.setVbarPolicy(ScrollPane.ScrollBarPolicy.*AS\_NEEDED*);  
 scrollPane.setStyle("-fx-border-width: 0; -fx-background-color: transparent;");  
  
 mainLayout.setLeft(sidebar);  
 mainLayout.setCenter(scrollPane);  
  
 double width = Screen.*getPrimary*().getVisualBounds().getWidth();  
 double height = Screen.*getPrimary*().getVisualBounds().getHeight();  
  
 Scene scene = new Scene(mainLayout, width, height);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle("Dashboard");  
 primaryStage.show();  
 }  
  
 public static void main(String[] args)  
 {  
 *launch*(args);  
 }  
}

CAR DETAIL VIEW

package com.example.semesterrproject;  
  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.geometry.Rectangle2D;  
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.Screen;  
import javafx.stage.Stage;  
  
import java.io.\*;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
  
public class CarDetailView extends Application {  
  
 private Stage primaryStage;  
 private Car car;  
 Label carstatusLabel;  
  
 public CarDetailView(Car car) {  
 this.car = car;  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
 this.primaryStage = primaryStage;  
 BorderPane mainLayout = new BorderPane();  
  
  
 VBox sidebar = new VBox(20);  
 sidebar.setPadding(new Insets(10));  
 sidebar.setStyle("-fx-background-color: #202020; -fx-pref-width: 80px;");  
  
  
 for (int i = 0; i < 5; i++) {  
 Label icon = new Label("Icon " + (i + 1));  
 icon.setStyle("-fx-text-fill: white;");  
 sidebar.getChildren().add(icon);  
 }  
  
  
 VBox content = new VBox(20);  
 content.setPadding(new Insets(20));  
 content.setAlignment(Pos.*CENTER*);  
  
  
 ImageView carImage = new ImageView(new Image(car.getImageUrl()));  
 carImage.setFitWidth(300);  
 carImage.setPreserveRatio(true);  
  
 Label carName = new Label(car.getName());  
 carName.setStyle("-fx-font-size: 18px; -fx-font-weight: bold;");  
 Label carPrice = new Label("Rs. " + car.getPricePerDay() + " per Day");  
 carPrice.setStyle("-fx-font-size: 16px; -fx-text-fill: gray;");  
  
 HBox specs = new HBox(20);  
 specs.setAlignment(Pos.*CENTER*);  
  
 VBox seatsInfo = createSpecBox("Seats", car.getSeats() + " Persons");  
 VBox fuelInfo = createSpecBox("Fuel", car.getIsWithFuel() ? "With Fuel" : "Without Fuel");  
 VBox locationInfo = createSpecBox("Location", car.getLocation());  
 specs.getChildren().addAll(seatsInfo, fuelInfo, locationInfo);  
  
 DatePicker pickupDatePicker = new DatePicker();  
 pickupDatePicker.setPromptText("Select Pickup Date");  
  
 DatePicker dropOffDatePicker = new DatePicker();  
 dropOffDatePicker.setPromptText("Select Drop-Off Date");  
  
 Label statusLabel = new Label(car.isAvailable() ? "Status: Available" : "Status: Not Available");  
 statusLabel.setStyle("-fx-font-size: 14px; -fx-font-weight: bold;");  
  
 HBox buttons = new HBox(20);  
 buttons.setAlignment(Pos.*CENTER*);  
 Button bookNow = new Button("BOOK NOW");  
  
 bookNow.setDisable(!car.isAvailable());  
  
 bookNow.setStyle("-fx-background-color: #000; -fx-text-fill: white; -fx-font-size: 14px;");  
 Button backButton = new Button("Back");  
 backButton.setStyle("-fx-font-size: 14px;");  
  
 buttons.getChildren().addAll(bookNow, backButton);  
  
  
 bookNow.setOnAction(e -> {  
 LocalDate pickupDate = pickupDatePicker.getValue();  
 LocalDate dropoffDate = dropOffDatePicker.getValue();  
  
 if (pickupDate == null || dropoffDate == null || pickupDate.isAfter(dropoffDate)) {  
 new Alert(Alert.AlertType.*ERROR*, "Please select valid Pick-up and Drop-Off dates.").showAndWait();  
 return;  
 }  
  
 if (!car.isAvailableDuring(pickupDate, dropoffDate)) {  
 new Alert(Alert.AlertType.*ERROR*, "Car is not available for the selected dates.").showAndWait();  
 return;  
 }  
  
 Reservation newReservation = new Reservation(car.getId(), car.getName(), pickupDate, dropoffDate);  
 car.addReservation(newReservation);  
  
 new Alert(Alert.AlertType.*INFORMATION*, "Car successfully booked!").showAndWait();  
 new BookingConfirmation(car, pickupDate, dropoffDate).start(new Stage());  
 primaryStage.close();  
 });  
  
  
 backButton.setOnAction(e -> {  
 primaryStage.close();  
 new Dashboard().start(new Stage());  
 });  
  
 content.getChildren().addAll(carImage, carName, carPrice, specs, pickupDatePicker, dropOffDatePicker,statusLabel, buttons);  
  
 mainLayout.setLeft(sidebar);  
 mainLayout.setCenter(content);  
  
 Rectangle2D screenBounds = Screen.*getPrimary*().getVisualBounds();  
 double screenWidth = screenBounds.getWidth();  
 double screenHeight = screenBounds.getHeight();  
  
 Scene scene = new Scene(mainLayout, screenWidth, screenHeight);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle("Car Detail View");  
 primaryStage.show();  
 }  
  
 private VBox createSpecBox(String title, String detail) {  
 VBox box = new VBox(5);  
 box.setAlignment(Pos.*CENTER*);  
  
 Label titleLabel = new Label(title);  
 titleLabel.setStyle("-fx-font-size: 14px; -fx-text-fill: gray;");  
 Label detailLabel = new Label(detail);  
 detailLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
  
 box.getChildren().addAll(titleLabel, detailLabel);  
 return box;  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

CAR:

package com.example.semesterrproject;  
  
  
import java.io.\*;  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.Date;  
import java.util.List;  
  
public class Car {  
 private int id;  
 private String name;  
 private int seats;  
 private String location;  
 private double pricePerDay;  
 private String imageUrl;  
 private boolean isWithFuel;  
 private boolean isAvailable;  
 private LocalDate pickupDate;  
 private LocalDate dropOffDate;  
 private List<Reservation> reservationList;  
  
 public Car(int id, String name, int seats, String location, double pricePerDay, String imageUrl, boolean isWithFuel) {  
 this.name = name;  
 this.seats = seats;  
 this.location = location;  
 this.isWithFuel = isWithFuel;  
 this.pricePerDay = pricePerDay;  
 this.imageUrl = imageUrl;  
 this.isAvailable = true;  
 this.pickupDate = null;  
 this.dropOffDate = null;  
 this.id = id;  
 this.reservationList = new ArrayList<>();  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public int getSeats() {  
 return seats;  
 }  
  
 public double getPricePerDay() {  
 return pricePerDay;  
 }  
  
 public String getImageUrl() {  
 return imageUrl;  
 }  
  
 public String getLocation() {  
 return location;  
 }  
  
 public boolean getIsWithFuel(){  
 return isWithFuel;  
 }  
  
 public LocalDate getPickupDate(){  
 return pickupDate;  
 }  
  
 public LocalDate getDropOffDate(){  
 return dropOffDate;  
 }  
  
 public boolean isAvailable(){  
 return isAvailable;  
 }  
  
 public boolean isAvailableDuring(LocalDate pickupDate, LocalDate dropoffDate) {  
 List<Reservation> reservations = getReservationsFromFile();  
 for (Reservation reservation : reservations) {  
 if (reservation.getCarId() == this.id &&  
 !(dropoffDate.isBefore(reservation.getPickupDate()) || pickupDate.isAfter(reservation.getDropoffDate()))) {  
 return false;  
 }  
 }  
 return true;  
 }  
  
  
 public void addReservation(Reservation reservation) {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter("booking.txt", true))) {  
 writer.write(reservation.getCarId() + "," +  
 reservation.getCarName() + "," +  
 reservation.getPickupDate() + "," +  
 reservation.getDropoffDate());  
 writer.newLine();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
  
  
 private List<Reservation> getReservationsFromFile() {  
 List<Reservation> reservations = new ArrayList<>();  
 try (BufferedReader reader = new BufferedReader(new FileReader("booking.txt"))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(",");  
 if (parts.length == 4) {  
 int carId = Integer.*parseInt*(parts[0]);  
 String carName = parts[1];  
 LocalDate pickupDate = LocalDate.*parse*(parts[2]);  
 LocalDate dropoffDate = LocalDate.*parse*(parts[3]);  
 reservations.add(new Reservation(carId, carName, pickupDate, dropoffDate));  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return reservations;  
 }  
  
  
 public int getId(){  
 return id;  
 }  
}

CAR LIST:

package com.example.semesterrproject;  
  
import java.time.LocalDate;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Locale;  
  
public class CarsList {  
 private List<Car> carslist;  
  
 public CarsList() {  
 carslist = new ArrayList<>();  
 loadSampleCars();  
 }  
  
 private void loadSampleCars() {  
 carslist.add(new Car(1, "Honda Civic RS 2024", 5, "Lahore", 20000, "civic.jpg", false));  
 carslist.add(new Car(2, "Toyota Corolla Altis", 5, "Islamabad", 17000, "corolla.png", false));  
 carslist.add(new Car(3, "Suzuki Swift 2023", 4, "Multan", 25000, "swift.png", true));  
 carslist.add(new Car(4, "Changan Alsvin", 5, "Sheikhupura", 15000, "alsvin.png", false));  
 carslist.add(new Car(5, "Toyota Revo", 5, "Islamabad", 25000, "revo.png", false));  
 carslist.add(new Car(6, "Suzuki Wagon-R", 4, "Faisalabad", 20000, "wagon-r.jpg", true));  
 }  
  
 public List<Car> getAvailableCars() {  
 List<Car> availableCars = new ArrayList<>();  
 for (Car car : carslist) {  
 if (car.isAvailable()) {  
 availableCars.add(car);  
 }  
 }  
 return availableCars;  
 }  
}

BOOKING CONFIRMATION:

package com.example.semesterrproject;  
  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.geometry.Rectangle2D;  
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.Screen;  
import javafx.stage.Stage;  
  
import java.time.LocalDate;  
  
public class BookingConfirmation extends Application {  
  
 private Stage primaryStage;  
  
 private Car car;  
 private LocalDate pickupDate;  
 private LocalDate dropoffDate;  
  
 public BookingConfirmation(Car car, LocalDate pickupDate, LocalDate dropoffDate) {  
 this.car = car;  
 this.pickupDate = pickupDate;  
 this.dropoffDate = dropoffDate;  
 }  
  
 @Override  
 public void start(Stage primaryStage) {  
  
 this.primaryStage = primaryStage;  
  
 VBox mainLayout = new VBox(20);  
 mainLayout.setPadding(new Insets(20));  
 mainLayout.setAlignment(Pos.*TOP\_CENTER*);  
  
 Label title = new Label("Hey! User,");  
 title.setStyle("-fx-font-size: 24px; -fx-font-weight: bold;");  
 Label subtitle = new Label("Your order is waiting for approval.");  
 subtitle.setStyle("-fx-font-size: 16px; -fx-text-fill: gray;");  
 Label instruction = new Label("You will receive a confirmation email soon with your booking details.");  
 instruction.setStyle("-fx-font-size: 14px; -fx-text-fill: gray;");  
  
  
 ImageView pendingIcon = new ImageView(new Image("pendingapproval.png")); // Replace with actual image path  
 pendingIcon.setFitWidth(100);  
 pendingIcon.setPreserveRatio(true);  
  
  
 GridPane detailsGrid = new GridPane();  
 detailsGrid.setHgap(20);  
 detailsGrid.setVgap(30);  
 detailsGrid.setPadding(new Insets(10));  
 detailsGrid.setAlignment(Pos.*CENTER\_LEFT*);  
  
 detailsGrid.addRow(0, createDetailRow("Car Information", car.getName(), "car.png"));  
 detailsGrid.addRow(1, createDetailRow("Location", car.getLocation(), "location.png"));  
 detailsGrid.addRow(2, createDetailRow("Price", "Rs. " + car.getPricePerDay() + " per-Day", "price.png"));  
 Label dateLabel = new Label("Pickup Date: " + pickupDate + "\nDrop-Off Date: " + dropoffDate);  
 dateLabel.setStyle("-fx-font-size: 14px;");  
  
  
 Button goToDashboard = new Button("Go to Dashboard");  
 goToDashboard.setStyle("-fx-background-color: #000; -fx-text-fill: white; -fx-padding: 10 20; -fx-font-size: 14px;");  
  
 goToDashboard.setOnAction(e -> {  
 primaryStage.close();  
 new Dashboard().start(new Stage());  
 });  
  
 mainLayout.getChildren().addAll(title, subtitle, instruction, pendingIcon, detailsGrid, dateLabel, goToDashboard);  
  
 Rectangle2D screenBounds = Screen.*getPrimary*().getVisualBounds();  
 double screenWidth = screenBounds.getWidth();  
 double screenHeight = screenBounds.getHeight();  
  
 Scene scene = new Scene(mainLayout, screenWidth, screenHeight);  
 primaryStage.setScene(scene);  
 primaryStage.setTitle("Booking Confirmation");  
 primaryStage.show();  
 }  
  
 private HBox createDetailRow(String label, String value, String iconPath) {  
 HBox row = new HBox(15);  
 row.setAlignment(Pos.*CENTER\_LEFT*);  
  
  
 ImageView icon = new ImageView(new Image(iconPath)); // Replace with actual icon path  
 icon.setFitWidth(50);  
 icon.setPreserveRatio(true);  
  
  
 VBox textContainer = new VBox(5);  
 Label labelText = new Label(label);  
 labelText.setStyle("-fx-font-size: 14px; -fx-text-fill: gray;");  
 Label valueText = new Label(value);  
 valueText.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
  
 textContainer.getChildren().addAll(labelText, valueText);  
 row.getChildren().addAll(icon, textContainer);  
  
 return row;  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

RESERVATION:

package com.example.semesterrproject;  
  
import java.time.LocalDate;  
  
public class Reservation {  
 private int carId;  
 private String carName;  
 private LocalDate pickupDate;  
 private LocalDate dropoffDate;  
  
 public Reservation(int carId, String carName, LocalDate pickupDate, LocalDate dropoffDate) {  
 this.carId = carId;  
 this.carName = carName;  
 this.pickupDate = pickupDate;  
 this.dropoffDate = dropoffDate;  
 }  
  
 public int getCarId() {  
 return carId;  
 }  
  
 public String getCarName() {  
 return carName;  
 }  
  
 public LocalDate getPickupDate() {  
 return pickupDate;  
 }  
  
 public LocalDate getDropoffDate() {  
 return dropoffDate;  
 }  
}