****

**Object Oriented Programming**

**ASSIGNMENT # 4**

**Submitted by:**

**Name: Reg No:**

**Hareem Mohal SP24-BSE-133**

**Muhammad Reyan SP24-BSE-080**

**Abis Raza Haider SP24-BSE-145**

**Submitted to:**

**Sir Shahid Bhatti**

**Date:**

**13th Dec,2024**

**Section:**

**BSE-B**

**CLASSES:**

* CafeManagementSystem:

**Source Code:**

package com.example.cafe;  
  
import javafx.application.Application;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.geometry.Rectangle2D;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.effect.GaussianBlur;  
import javafx.scene.layout.\*;  
import javafx.scene.text.Text;  
import javafx.stage.Screen;  
import javafx.stage.Stage;  
  
import java.io.\*;  
import java.util.ArrayList;  
  
public class CafeManagementSystem extends Application {  
 public static int *counter*=0;  
  
 ArrayList<UserDetails> user=new ArrayList<>();  
 public static final String *USER\_FILE* = "User.txt";  
  
  
 public static void main(String[] args) {  
 *launch*();  
 }  
  
 @Override  
 public void start(Stage stage) throws IOException {  
  
 Rectangle2D screenBounds = Screen.*getPrimary*().getVisualBounds();  
 // Background layer  
 StackPane background = new StackPane();  
 background.setId("background");  
  
 background.setEffect(new GaussianBlur(10)); // Apply blur to background only  
  
 // Foreground content  
 Text txt = new Text("Welcome to the Cafe Management System");  
 txt.getStyleClass().add("Text");  
  
 Button loginBtn = new Button("Login");  
 loginBtn.getStyleClass().add("button1");  
  
 Button signupBtn = new Button("Signup");  
 signupBtn.getStyleClass().add("button1");  
  
 VBox vbox2 = new VBox(25,txt,loginBtn,signupBtn);  
 vbox2.setId("vbox2");  
  
  
 VBox root = new VBox(10, vbox2);  
 root.setId("root");  
  
 // Combine background and foreground  
 StackPane mainPane = new StackPane();  
 mainPane.getChildren().addAll(background, root);  
  
 // Scene setup  
 Scene scene = new Scene(mainPane);  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
 signupBtn.setOnAction(e->{  
 signupScreen(stage);  
 });  
 loginBtn.setOnAction(e->{  
 loginScreen(stage);  
  
 });  
  
 // Stage setup  
 stage.setScene(scene);  
 stage.setWidth(screenBounds.getWidth());  
 stage.setHeight(screenBounds.getHeight());  
 stage.setTitle("Cafe Management System");  
 stage.show();  
 }  
 public void loginScreen(Stage stage) {  
//  
  
 // Background Pane  
 StackPane picPane = new StackPane();  
 picPane.setId("loginBackground");  
 picPane.setEffect(new GaussianBlur(10));  
  
 // Login Form  
 Label l1 = new Label("Username");  
 l1.setId("usernameLabel");  
 TextField txtForLoginuser = new TextField();  
 txtForLoginuser.setPromptText("Enter Username"); // Placeholder text  
 txtForLoginuser.setId("username");  
  
 Label l2 = new Label("Password");  
 l2.setId("passwordLabel");  
 PasswordField passwordForLoginUser = new PasswordField();  
 passwordForLoginUser.setPromptText("Enter Password");  
 passwordForLoginUser.setId("password");  
  
 Button loginbtnForMain= new Button("Login");  
 loginbtnForMain.getStyleClass().add("button1");  
 loginbtnForMain.setOnAction(e->{  
 int index1=0;  
 String Username = txtForLoginuser.getText();  
 String Password=passwordForLoginUser.getText();  
 try {  
 loadUsersFromFile();  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 for(int i=0;i<user.size();i++){  
 if (user.get(i).username().equals(Username) && user.get(i).password().equals(Password)){  
 index1=i;  
 }  
 }  
 for (int i=0;i<user.size();i++) {  
 if (user.get(index1).username().equals(Username) && user.get(index1).password().equals(Password)) {  
 Menu m1 = new Menu();  
 m1.menu(stage,user.get(index1).username());  
 break;  
  
 } else {  
 Alert alert = new Alert(Alert.AlertType.ERROR);  
 alert.setTitle("Error");  
 alert.setHeaderText(null);  
 alert.setContentText("Wrong Username or Password");  
 alert.showAndWait();  
 break;  
  
 }  
 }  
 });  
  
 VBox v1 = new VBox(15); // Spacing between elements  
 v1.getChildren().addAll(l1, txtForLoginuser, l2, passwordForLoginUser, loginbtnForMain);  
 v1.setId("vbox3");  
 v1.setAlignment(Pos.CENTER); // Center the login form vertically  
  
 // Forgot Password Link  
 Hyperlink hyperlink = new Hyperlink("Forgot Password?");  
 hyperlink.setId("hyperlink");  
 hyperlink.setOnAction(e->{  
 forgetPassword(stage);  
 });  
  
 // Back Button  
 Button backButtononLogin = new Button();  
 backButtononLogin.setId("backButton");  
 backButtononLogin.setOnAction(e->{  
 try {  
 start(stage);  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 });  
 // Main Container for `#pane3`  
 VBox paneContent = new VBox(10); // Vertical spacing  
 paneContent.getChildren().addAll(v1, hyperlink);  
 paneContent.setAlignment(Pos.CENTER);  
 VBox.setMargin(hyperlink, new Insets(20, 0, 0, 0)); // Add margin above the hyperlink  
  
 Pane p3 = new Pane();  
 p3.setId("pane3");  
 p3.getChildren().add(paneContent);  
  
 // Align `paneContent` to the center of `#pane3`  
 StackPane.setAlignment(paneContent, Pos.CENTER);  
  
 // Align `btn2` to the top left  
 StackPane.setAlignment(backButtononLogin, Pos.TOP\_LEFT);  
  
 // Combine all panes  
 StackPane mainPane = new StackPane();  
 mainPane.getChildren().addAll(picPane, p3, backButtononLogin);  
  
 // Scene setup  
 Scene scene = new Scene(mainPane, 1300, 1000);  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
 stage.setScene(scene);  
 }  
  
public void signupScreen(Stage stage) {  
 StackPane signupImg = new StackPane();  
 signupImg.setId("signImage");  
 signupImg.setEffect(new GaussianBlur(10));  
  
 Label l1 = new Label("Username");  
 l1.getStyleClass().add("textDesign");  
 TextField txtFornewUser = new TextField();  
 txtFornewUser.setPromptText("Enter Username");  
 txtFornewUser.getStyleClass().add("fieldDesign");  
  
 Label l2 = new Label("Email");  
 l2.getStyleClass().add("textDesign");  
 TextField txtFornewEmail = new TextField();  
 txtFornewEmail.setPromptText("Enter Email");  
 txtFornewEmail.getStyleClass().add("fieldDesign");  
  
 Label l3 = new Label("Create Password");  
 l3.getStyleClass().add("textDesign");  
 PasswordField passwordForNewSign = new PasswordField();  
 passwordForNewSign.setPromptText("Enter Password");  
 passwordForNewSign.getStyleClass().add("fieldDesign");  
  
 Label l4 = new Label("Confirm Password");  
 l4.getStyleClass().add("textDesign");  
 PasswordField passwordCForNewSign = new PasswordField();  
 passwordCForNewSign .setPromptText("Enter Confirm Password");  
 passwordCForNewSign.getStyleClass().add("fieldDesign");  
  
 Button btn = new Button("Signup");  
 btn.getStyleClass().add("button1");  
 btn.setOnAction(e -> {  
 String username = txtFornewUser.getText();  
 String email = txtFornewEmail.getText();  
 String password1 = passwordForNewSign.getText();  
 String confirmPassword = passwordCForNewSign.getText();  
  
 if (password1.equals(confirmPassword)) {  
 user.add(new UserDetails(email, password1, username));  
 try {  
 inputuser();  
 Alert alert = new Alert(Alert.AlertType.INFORMATION);  
 alert.setTitle("Success");  
 alert.setHeaderText(null);  
 alert.setContentText("User successfully created");  
 alert.showAndWait();  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 } else {  
 Alert alert = new Alert(Alert.AlertType.ERROR);  
 alert.setTitle("Error");  
 alert.setHeaderText(null);  
 alert.setContentText("Passwords do not match");  
 alert.showAndWait();  
  
 }  
 });  
  
 Button btn2 = new Button("Back to Login");  
 btn2.getStyleClass().add("button1");  
 btn2.setOnAction(e -> {  
 loginScreen(stage);  
 });  
  
 VBox v2 = new VBox(10);  
 v2.getChildren().addAll(l1, txtFornewUser, l2, txtFornewEmail, l3, passwordForNewSign, l4, passwordCForNewSign, btn, btn2);  
 v2.setId("vbox4");  
 v2.setAlignment(Pos.CENTER);  
  
 VBox.setMargin(btn, new Insets(10, 0, 0, 0));  
 VBox.setMargin(btn2, new Insets(5, 0, 0, 0));  
  
 StackPane mainPane1 = new StackPane();  
 mainPane1.getChildren().addAll(signupImg, v2);  
 Scene scene = new Scene(mainPane1);  
  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
 stage.setScene(scene);  
 stage.show();  
}  
public void forgetPassword(Stage stage) {  
 Stage p1=new Stage();  
 StackPane forgetImg=new StackPane();  
 forgetImg.setId("forgetImg");  
 forgetImg.setEffect(new GaussianBlur(10));  
 Label l3=new Label("Email");  
 l3.getStyleClass().add("textDesign");  
 TextField txtToMatchEmail = new TextField();  
 txtToMatchEmail.setPromptText("Enter your Email");  
 txtToMatchEmail.getStyleClass().add("fieldDesign");  
 Label l1 = new Label("Enter New Password");  
 l1.getStyleClass().add("textDesign");  
 PasswordField newPassword = new PasswordField();  
 newPassword.setPromptText("Enter New Password");  
 newPassword.getStyleClass().add("fieldDesign");  
 Label l2 = new Label("Confirm Password");  
 l2.getStyleClass().add("textDesign");  
 PasswordField confirmPassword = new PasswordField();  
 confirmPassword.setPromptText("Enter Confirm Password");  
 confirmPassword.getStyleClass().add("fieldDesign");  
 Button savePasswordBtn = new Button("Save Password");  
 savePasswordBtn.getStyleClass().add("button1");  
 savePasswordBtn.setOnAction(e->{  
 try {  
 loadUsersFromFile();  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 String Email = txtToMatchEmail.getText();  
 String Password = newPassword.getText();  
 String ConfirmPassword = confirmPassword.getText();  
 for(UserDetails u1:user){  
 if(u1.Email().equals(Email)){  
 try {  
 updatepassword(Email,Password);  
 break;  
 } catch (IOException ex) {  
 throw new RuntimeException(ex);  
 }  
 }else{  
 Alert errorAlert = new Alert(Alert.AlertType.ERROR);  
 errorAlert.setTitle("Invalid Email");  
 errorAlert.setHeaderText("Error Alert");  
 errorAlert.setContentText("Email entered is invalid");  
 errorAlert.showAndWait();  
 break;  
 }  
 }  
  
 p1.close();  
 });  
 VBox v3 = new VBox(10);  
 v3.getChildren().addAll(l3,txtToMatchEmail,l1,newPassword,l2,confirmPassword,savePasswordBtn);  
 v3.setId("vbox4");  
 v3.setAlignment(Pos.CENTER);  
 StackPane mainPane = new StackPane();  
 mainPane.getChildren().addAll(forgetImg, v3);  
 Scene scene2 = new Scene(mainPane,800,600);  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene2.getStylesheets().add(stylesheet.toExternalForm());  
 }  
  
 p1.setScene(scene2);  
 p1.setTitle("Forget Password Form");  
 p1.show();  
  
}  
public void inputuser() throws IOException {  
  
 try {  
 BufferedWriter r1=new BufferedWriter(new FileWriter(USER\_FILE,true));  
  
  
 r1.write(user.get(counter).Email() + ";" + user.get(counter).password() + ";" + user.get(counter).username()+"\n");  
 counter++;  
r1.close();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
 }  
 public void login(Stage stage) throws IOException {  
  
  
  
 }  
 public void updatepassword( String Email,String password1) throws IOException {  
  
 for (int i = 0; i < user.size(); i++) {  
 if (user.get(i).Email().equals(Email)) {  
 user.get(i).setPassword(password1);  
 break;  
 }  
 }  
 BufferedWriter w1=new BufferedWriter(new FileWriter(USER\_FILE,false));  
 for(UserDetails u1:user){  
 w1.write(u1.Email() + ";" + u1.password() + ";" + u1.username()+"\n");  
 }  
  
w1.close();  
 Alert errorAlert = new Alert(Alert.AlertType.INFORMATION);  
 errorAlert.setTitle("Password Updated");  
 errorAlert.setHeaderText("password updated");  
 errorAlert.setContentText("password updated Successfully");  
 errorAlert.showAndWait();  
 }  
 public void loadUsersFromFile() throws IOException {  
 user.clear(); // Clear the list before loading  
 BufferedReader reader = new BufferedReader(new FileReader(*USER\_FILE*));  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(";");  
 if (parts.length == 3) {  
 user.add(new UserDetails(parts[0], parts[1], parts[2])); // Email, Password, Username  
 }  
 }  
 reader.close();  
 }  
  
  
  
  
}

**INVENTORY SYSTEM:**

**Source Code:**

package com.example.cafe;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableArray;  
import javafx.collections.ObservableList;  
import javafx.scene.control.Alert;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.cell.PropertyValueFactory;  
  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.List;  
  
public class InventorySystem {  
  
 public static final String *USER\_FILE*="inventory.txt";  
 List<Product> products=new ArrayList<>();  
  
 ObservableList<Product> observableProducts= FXCollections.*observableArrayList*();  
  
  
 public void addProduct( int id,String name, double price,String imagePath){  
  
 products.add(new Product(id,name,price,imagePath));  
  
 saveInfoToFile();  
 }  
  
  
public void saveInfoToFile() {  
 try {  
 // Create a new ObjectOutputStream to overwrite the file  
 ObjectOutputStream fileObject = new ObjectOutputStream(new FileOutputStream(*USER\_FILE*));  
  
 // Write the updated list of products to the file  
 fileObject.writeObject(products);  
  
 fileObject.close();  
 } catch (IOException e) {  
 throw new RuntimeException("Error saving to file", e);  
 }  
}  
  
  
public void removeProduct(String name) {  
 try {  
 loadFromFile(); // Load products from the file  
  
 // Find the product to remove  
 Product productToRemove = null;  
 for (Product p : products) {  
 if (p.getName().equals(name)) {  
 productToRemove = p;  
 break; // Exit the loop once the product is found  
 }  
 }  
  
 // If the product is found, remove it  
 if (productToRemove != null) {  
 products.remove(productToRemove); // Remove the product from the list  
  
 // Display confirmation alert  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setTitle("Confirmation");  
 alert.setHeaderText(null);  
 alert.setContentText("Product Deleted Successfully");  
 alert.showAndWait();  
  
 // Save the updated list to the file  
 saveInfoToFile();  
 } else {  
 // If the product was not found, show an alert  
 Alert alert = new Alert(Alert.AlertType.*WARNING*);  
 alert.setTitle("Warning");  
 alert.setHeaderText(null);  
 alert.setContentText("Product not found");  
 alert.showAndWait();  
 }  
  
 } catch (FileNotFoundException e) {  
 throw new RuntimeException("File not found", e);  
 }  
}  
 public void loadFromFile() throws FileNotFoundException {  
 products.clear();  
  
 ObjectInputStream fileObject= null;  
 try {  
 fileObject = new ObjectInputStream(new FileInputStream(*USER\_FILE*));  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
 try {  
 try {  
 try {  
 products = (List<Product>) fileObject.readObject();  
 } catch (ClassNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 fileObject.close();  
 } catch (IOException e) {  
 throw new RuntimeException(e);  
 }  
  
  
 }catch(Exception e){  
 throw new RuntimeException(e);  
 }  
 }  
  
 public void updatePrice(String name,double price) {  
 try {  
 loadFromFile();  
 } catch (FileNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
 Product productToUpdate=null;  
 for(Product p2 : products) {  
 if(p2.getName().equals(name)) {  
 productToUpdate=p2;  
 }  
 }  
 if(productToUpdate != null) {  
 productToUpdate.setPrice(price);  
 saveInfoToFile();  
 Alert alert = new Alert(Alert.AlertType.*CONFIRMATION*);  
 alert.setTitle("Confirmation");  
 alert.setHeaderText(null);  
 alert.setContentText("Product Updated Successfully");  
 alert.showAndWait();  
 }  
 else{  
 Alert alert = new Alert(Alert.AlertType.*WARNING*);  
 alert.setTitle("Warning");  
 alert.setHeaderText(null);  
 alert.setContentText("Product not found");  
 alert.showAndWait();  
 }  
  
  
 }  
 public void addToTable(int id, String name, double price, TableColumn<Product, String> idCol, TableColumn<Product,String> nameCol, TableColumn<Product, String> priceCol, TableView<Product> table,Product product) throws FileNotFoundException {  
 observableProducts.add(product);  
 idCol.setCellValueFactory(new PropertyValueFactory<>("id"));  
 nameCol.setCellValueFactory(new PropertyValueFactory<>("name"));  
 priceCol.setCellValueFactory(new PropertyValueFactory<>("price"));  
 table.setItems(observableProducts);  
 }  
 public void removeFromTable(String name) {  
 for(Product o:observableProducts){  
 if(o.getName().equals(name)) {  
 observableProducts.remove(o);  
 break;  
 }  
 }  
 }  
 public void updateFromTable(int id,String name,double price, TableView<Product> table) {  
 for(Product o:observableProducts){  
 if(o.getName().equals(name) && o.getId()==id) {  
 o.setPrice(price);  
 break;  
 }  
  
 }  
 table.refresh();  
 }  
  
  
}

**INVENTORY TABLE APP:**

**Source Code:**

package com.example.cafe;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Pos;  
import javafx.scene.control.\*;  
import javafx.scene.Scene;  
import javafx.scene.effect.GaussianBlur;  
import javafx.scene.image.Image;  
import javafx.scene.layout.BorderPane;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.image.ImageView;  
import javafx.stage.FileChooser;  
import javafx.stage.Stage;  
import java.io.File;  
import java.io.FileNotFoundException;  
  
  
public class InventoryTableApp {  
  
InventorySystem i1 = new InventorySystem();  
  
 public void inventory (Stage stage,String user) {  
 final String[] imagePath={null};  
  
 StackPane root = new StackPane();  
 root.setId("main-stackpane");  
  
// Background Layer  
 StackPane backgroundPane = new StackPane();  
 backgroundPane.setId("background-layer");  
 backgroundPane.setEffect(new GaussianBlur(10)); // Apply blur only to the background  
  
// Inner StackPane for content  
 StackPane contentPane = new StackPane();  
 contentPane.setId("content-stackpane");  
  
// Title  
 Label titleLabel = new Label("List of Items");  
 titleLabel.setId("title-label");  
 StackPane.*setAlignment*(titleLabel, Pos.*TOP\_LEFT*);  
  
// Table  
 TableView<Product> table = new TableView<>();  
 table.setId("inventory-table");  
 TableColumn<Product, String> idCol = new TableColumn<>("ID");  
 TableColumn<Product, String> nameCol = new TableColumn<>("Name");  
 TableColumn<Product, String> priceCol = new TableColumn<>("Price");  
 table.getColumns().addAll(idCol, nameCol, priceCol);  
 table.setItems(i1.getObservableProducts());  
 idCol.setCellValueFactory(new PropertyValueFactory<>("id"));  
 nameCol.setCellValueFactory(new PropertyValueFactory<>("name"));  
 priceCol.setCellValueFactory(new PropertyValueFactory<>("price"));  
  
  
  
// Form Fields  
 TextField idField = new TextField();  
 idField.setPromptText("Add ID");  
 idField.getStyleClass().add("input-field");  
  
 TextField nameField = new TextField();  
 nameField.setPromptText("Add Name");  
 nameField.getStyleClass().add("input-field");  
  
 TextField priceField = new TextField();  
 priceField.setPromptText("Add Price");  
 priceField.getStyleClass().add("input-field");  
 VBox imageBox = new VBox(10);  
 ImageView imageView = new ImageView();  
 imageView.setFitWidth(100);  
 imageView.setFitHeight(100);  
 imageView.getStyleClass().add("image-preview");  
  
  
  
 Button uploadButton = new Button("Upload Picture");  
 uploadButton.getStyleClass().add("action-button");  
 uploadButton.setOnAction(e -> {  
 FileChooser fileChooser = new FileChooser();  
 File file = fileChooser.showOpenDialog(stage);  
 if (file != null) {  
 imageView.setImage(new Image(file.toURI().toString()));  
 imagePath[0] = file.getAbsolutePath(); // Full path of the file  
  
 }  
 });  
  
 Button addButton = new Button("Add");  
 addButton.getStyleClass().add("action-button");  
 addButton.setOnAction(event -> {  
 int id = Integer.*parseInt*(idField.getText());  
 String name=nameField.getText();  
 double price=Double.*parseDouble*(priceField.getText());  
 i1.addProduct(id,name,price,imagePath[0]);  
 Product product = new Product(id,name,price,imagePath[0]);  
 try {  
 i1.addToTable(id,name,price,idCol,nameCol,priceCol,table,product);  
 } catch (FileNotFoundException e) {  
 throw new RuntimeException(e);  
 }  
  
 Alert alert =new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Add Product");  
 alert.setHeaderText(null);  
 alert.setContentText("Product Added Succesfully");  
 alert.showAndWait();  
 idField.clear();  
 nameField.clear();  
 priceField.clear();  
 });  
 addButton.setDefaultButton(true);  
  
 Button removeButton = new Button("Remove");  
 removeButton.getStyleClass().add("action-button");  
 removeButton.setOnAction(event -> {  
 String name=nameField.getText();  
 int id=Integer.*parseInt*(idField.getText());  
 double price=Double.*parseDouble*(priceField.getText());  
 i1.removeProduct(name);  
 i1.removeFromTable(name);  
 idField.clear();  
 nameField.clear();  
 priceField.clear();  
 });  
  
  
 Button updateButton = new Button("Update Price");  
 updateButton.getStyleClass().add("action-button");  
 updateButton.setOnAction(e->{  
 String name=nameField.getText();  
 int id=Integer.*parseInt*(idField.getText());  
 Double price=Double.*parseDouble*(priceField.getText());  
 i1.updatePrice(name,price);  
 i1.updateFromTable(id,name,price,table);  
 idField.clear();  
 nameField.clear();  
 priceField.clear();  
  
 });  
 Menu menu = new Menu();  
 Button backButtononLogin = new Button();  
 backButtononLogin.setAlignment(Pos.*TOP\_LEFT*);  
 backButtononLogin.getStyleClass().add("backBtn");  
 backButtononLogin.setOnAction(e->{  
 menu.menu(stage,user);  
 });  
  
// Image Upload Section  
 imageBox.getChildren().addAll(imageView, uploadButton);  
  
// Layout Organization  
 VBox formBox = new VBox(10, idField, nameField, priceField, addButton, removeButton, updateButton);  
 formBox.setId("form-container");  
  
 HBox contentBox = new HBox(20, table, imageBox, formBox);  
 contentBox.setId("content-container");  
  
// Add content to inner StackPane  
 contentPane.getChildren().addAll(contentBox);  
  
// Add layers to root StackPane  
 root.getChildren().addAll(backgroundPane, titleLabel, contentPane,backButtononLogin);  
  
// CSS Styling  
 Scene scene = new Scene(root, 800, 600);  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
  
 stage.setScene(scene);  
 stage.setTitle("Inventory Table");  
 stage.show();  
  
 }  
  
  
  
  
  
 }

**Menu:**

**Source Code:**

package com.example.cafe;  
  
import javafx.collections.ObservableList;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.effect.GaussianBlur;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
import java.io.IOException;  
  
public class Menu {  
  
 public void menu(Stage stage, String user) {  
 // StackPane for background and effects  
 StackPane menupic = new StackPane();  
 menupic.setId("menupic");  
 menupic.setEffect(new GaussianBlur(10));  
  
 // Text displaying the welcome message  
 Text menuUsername = new Text("Welcome " + user);  
 menuUsername.getStyleClass().add("Text"); // Use this ID for styling via CSS  
  
 // Buttons  
 Button dashboard = new Button("Dashboard");  
 dashboard.getStyleClass().add("button1");  
 dashboard.setStyle("dropshadow(three-pass-box, rgba(0,0,0,0.3), 5, 0, 2, 2);");  
 dashboard.setOnAction(e->{  
 Dashboardy d2 = new Dashboardy();  
 d2.dashboardScreen(stage,user);  
 });  
 Button inventory = new Button("Inventory");  
 inventory.getStyleClass().add("button1");  
 inventory.setOnAction(event -> {  
 InventoryTableApp i1= new InventoryTableApp();  
 i1.inventory(stage,user);  
 });  
 Button Menubtn = new Button("Menu");  
 Menubtn.getStyleClass().add("button1");  
 Menubtn.setOnAction(event -> {  
 MenuPage m1= new MenuPage();  
 m1.menuDisplay(stage,user);  
 });  
  
  
 Button customers = new Button("Customers");  
 customers.getStyleClass().add("button1");  
 CafeManagementSystem cms = new CafeManagementSystem();  
 Button backButtononLogin = new Button();  
 backButtononLogin.getStyleClass().add("backBtn");  
 backButtononLogin.setAlignment(Pos.*TOP\_LEFT*);  
 backButtononLogin.setOnAction(e->{  
 cms.loginScreen(stage);  
 });  
  
 // VBox for buttons and text  
 VBox menuVbox = new VBox();  
 menuVbox.setAlignment(Pos.*CENTER*);  
 menuVbox.setId("menuVbox");  
 menuVbox.getChildren().addAll(menuUsername, dashboard, inventory,Menubtn, customers);  
  
 // StackPane for the main menu content  
 StackPane mainMenuPane = new StackPane();  
 mainMenuPane.getChildren().addAll(menupic, menuVbox,backButtononLogin);  
  
 // Scene setup  
 Scene mainMenuScene = new Scene(mainMenuPane, 300, 300);  
  
 // Load the stylesheet  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 mainMenuScene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
  
 // Set and show the stage  
 stage.setScene(mainMenuScene);  
 stage.show();  
 }  
}

**Product:**

**Source Code:**

package com.example.cafe;  
  
import java.io.Serializable;  
  
public class Product implements Serializable {  
 private int id;  
 private String name;  
 private double price;  
 private String imagePath;  
  
 public Product(int id, String name, double price, String imagePath) {  
 this.id = id;  
 this.name = name;  
 this.price = price;  
  
 this.imagePath = imagePath;  
 }  
  
 public void setId(int id) {  
 this.id = id;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setPrice(double price) {  
 this.price = price;  
 }  
  
 public void setImagePath(String filepath) {  
 this.imagePath = filepath;  
 }  
  
 public int getId() {  
 return id;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public String getImagePath() {  
 return imagePath;  
 }  
}

**User Details:**

**Source Code:**

package com.example.cafe;  
  
public class UserDetails {  
 private String username;  
 private String password;  
 private String Email;  
  
 public UserDetails(String email, String password, String username) {  
 this.Email = email;  
 this.password = password;  
 this.username = username;  
  
  
 }  
  
 public String username() {  
 return username;  
 }  
  
 public String password() {  
 return password;  
 }  
  
 public void setUsername(String username) {  
 this.username = username;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public void setEmail(String email) {  
 Email = email;  
 }  
  
 public String Email() {  
 return Email;  
 }  
  
 @Override  
 public String toString() {  
 return "UserDetails{" +  
 "username='" + username + '\'' +  
 ", password='" + password + '\'' +  
 ", Email='" + Email + '\'' +  
 '}';  
 }  
}

**MENU PAGE:**

**Source Code:**

package com.example.cafe;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.effect.GaussianBlur;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
  
public class MenuPage {  
 public void menuDisplay(Stage stage,String user) {  
  
//  
 BorderPane borderPane = new BorderPane();  
borderPane.setId("mainback");  
 // Menu Section  
 VBox menuSection = new VBox(10);  
 menuSection.setId("menu-section");  
 menuSection.setPadding(new Insets(10));  
 menuSection.setAlignment(Pos.*TOP\_CENTER*);  
  
 Label menuLabel = new Label("Menu");  
 menuLabel.setId("menu-label");  
  
 // Menu items with scroll  
 ScrollPane menuScrollPane = new ScrollPane();  
 VBox menuItems = new VBox(10);  
  
 menuItems.setPadding(new Insets(10));  
  
 // Example menu items  
 for (int i = 1; i <= 5; i++) {  
 VBox menuItem = new VBox(10);  
 menuItem.setId("menu-item");  
 menuItem.setAlignment(Pos.*CENTER*);  
 menuItem.setStyle("-fx-border-color: black; -fx-border-width: 1px; -fx-padding: 10;");  
  
 Label productName = new Label("Product " + i);  
 productName.setId("product-name-" + i);  
  
 Label productPrice = new Label("Price: $" + (10 \* i));  
 productPrice.setId("product-price-" + i);  
  
 ImageView productImage = new ImageView(new Image("file:home9.jpg")); // Replace with valid path  
 productImage.setFitHeight(80);  
 productImage.setFitWidth(80);  
 productImage.setId("product-image-" + i);  
  
 Button addToOrder = new Button("Add to Order");  
 addToOrder.setId("add-to-order");  
  
 menuItem.getChildren().addAll(productImage, productName, productPrice, addToOrder);  
 menuItems.getChildren().add(menuItem);  
 }  
 menuScrollPane.setContent(menuItems);  
 menuScrollPane.setFitToWidth(true);  
  
 menuSection.getChildren().addAll(menuLabel, menuScrollPane);  
  
 // Order Table Section  
 TableView<String> orderTable = new TableView<>();  
 orderTable.setId("order-table");  
 orderTable.setStyle("-fx-border-color: black; -fx-border-width: 2px;");  
 orderTable.setPlaceholder(new Label("No items in the order"));  
  
 TableColumn<String, String> productNameCol = new TableColumn<>("Product Name");  
 TableColumn<String, String> quantityCol = new TableColumn<>("Quantity");  
 TableColumn<String, String> priceCol = new TableColumn<>("Price");  
 orderTable.getColumns().addAll(productNameCol, quantityCol, priceCol);  
  
 // Add a scroll pane to the table  
 ScrollPane tableScrollPane = new ScrollPane(orderTable);  
 tableScrollPane.setFitToWidth(true);  
 tableScrollPane.setFitToHeight(true);  
  
 // Payment Section  
 VBox paymentSection = new VBox(10);  
 paymentSection.setId("payment-section");  
 paymentSection.setPadding(new Insets(10));  
 paymentSection.setAlignment(Pos.*CENTER\_RIGHT*);  
  
 Label totalLabel = new Label("Total:");  
 totalLabel.setId("total-label");  
  
 TextField totalField = new TextField();  
 totalField.setId("total-field");  
 totalField.setEditable(false);  
  
 Label amountPaidLabel = new Label("Amount paid by customer:");  
 amountPaidLabel.setId("amount-paid-label");  
  
 TextField amountPaidField = new TextField();  
 amountPaidField.setId("amount-paid-field");  
  
 Label changeLabel = new Label("Change:");  
 changeLabel.setId("change-label");  
  
 TextField changeField = new TextField();  
 changeField.setId("change-field");  
 changeField.setEditable(false);  
  
 HBox actionButtons = new HBox(10);  
 actionButtons.setAlignment(Pos.*CENTER*);  
 Button removeButton = new Button("Remove");  
 removeButton.getStyleClass().add("button1");  
 Button payButton = new Button("Pay");  
 payButton.getStyleClass().add("button1");  
 Button printReceiptButton = new Button("Print Bill Receipt");  
 printReceiptButton.getStyleClass().add("button1");  
Button backbutton = new Button("Back");  
backbutton.getStyleClass().add("button1");  
backbutton.setOnAction(actionEvent -> {  
 Menu m1 = new Menu();  
 m1.menu(stage,user);  
});  
 actionButtons.getChildren().addAll(removeButton, payButton, printReceiptButton,backbutton);  
  
 paymentSection.getChildren().addAll(totalLabel, totalField, amountPaidLabel, amountPaidField, changeLabel, changeField, actionButtons);  
  
 // Layout the sections  
 borderPane.setLeft(menuSection);  
 borderPane.setCenter(tableScrollPane);  
 borderPane.setBottom(paymentSection);  
  
  
 // Set widths dynamically  
 menuSection.prefWidthProperty().bind(borderPane.widthProperty().multiply(0.4)); // 40% of the width  
 tableScrollPane.prefWidthProperty().bind(borderPane.widthProperty().multiply(0.6)); // 60% of the width  
  
 // Create scene  
 Scene scene4 = new Scene(borderPane, 1000, 600);  
 stage.setScene(scene4);  
 // Adding Stylesheet  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene4.getStylesheets().add(stylesheet.toExternalForm());  
 System.*out*.println("CSS loaded successfully");  
 } else {  
 System.*out*.println("CSS not found");  
 }  
  
 stage.setTitle("Cafe Management System");  
 stage.show();  
  
  
 }  
}

**DASHBOARD:**

**Source Code:**

package com.example.cafe;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.layout.VBox;  
import javafx.scene.text.Text;  
import javafx.stage.Stage;  
  
import java.io.\*;  
  
public class Dashboardy {  
 private static final String *DASHBOARD\_FILE* = "dashboard\_data.txt";  
  
 // Fields to hold metrics  
 private double totalIncome = 0.0;  
 private int totalCustomers = 0;  
 private int totalSoldProducts = 0;  
 private double finalIncome;  
 private int finalCustomers;  
 private int finalProducts;  
  
 public void dashboardScreen(Stage stage, String user) {  
 // Load data from file  
 loadDashboardData();  
  
 // VBox for layout  
 VBox dashboardBox = new VBox(20);  
 dashboardBox.setAlignment(Pos.CENTER);  
 dashboardBox.setPadding(new Insets(20));  
 dashboardBox.setId("dashboardBox");  
  
 // Text for metrics  
 Text incomeText = new Text("Total Income: $" + String.*format*("%.2f", totalIncome));  
 incomeText.setId("income-text");  
  
 Text customersText = new Text("Total Customers: " + totalCustomers);  
 customersText.setId("customers-text");  
  
 Text soldProductsText = new Text("Total Sold Products: " + totalSoldProducts);  
 soldProductsText.setId("sold-products-text");  
  
 // Back to Menu button  
 Button backButton = new Button("Back to Menu");  
 backButton.getStyleClass().add("button1");  
 backButton.setOnAction(e -> {  
 Menu menu = new Menu();  
 menu.menu(stage, user);  
 });  
  
 // Add components to the VBox  
 dashboardBox.getChildren().addAll(incomeText, customersText, soldProductsText, backButton);  
  
 // Scene setup  
 Scene dashboardScene = new Scene(dashboardBox, 400, 300);  
  
 // Add CSS  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 dashboardScene.getStylesheets().add(stylesheet.toExternalForm());  
 }  
  
 // Set and show the stage  
 stage.setScene(dashboardScene);  
 stage.setTitle("Dashboard");  
 stage.show();  
 }  
  
 // Method to load data from the file  
 private void loadDashboardData() {  
 File file = new File(*DASHBOARD\_FILE*);  
 if (!file.exists()) {  
 saveDashboardData(); // Create the file with initial values if not exists  
 }  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 String[] parts = line.split(":");  
 if (parts.length == 2) {  
  
 if ("TotalIncome".equals(parts[0])) {  
 totalIncome = finalIncome(Double.*parseDouble*(parts[1]));  
 } else if ("TotalCustomers".equals(parts[0])) {  
 totalCustomers = finalCustomers(Integer.*parseInt*(parts[1]));  
 } else if ("TotalSoldProducts".equals(parts[0])) {  
 totalSoldProducts = finalProducts(Integer.*parseInt*(parts[1]));  
 }  
 }  
  
  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error reading dashboard data: " + e.getMessage());  
 }  
 }  
  
 // Method to save data to the file  
 public void saveDashboardData() {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(*DASHBOARD\_FILE*,true))) {  
 writer.write("TotalIncome:" + totalIncome + "\n");  
 writer.write("TotalCustomers:" + totalCustomers + "\n");  
 writer.write("TotalSoldProducts:" + totalSoldProducts + "\n");  
 } catch (IOException e) {  
 System.*err*.println("Error saving dashboard data: " + e.getMessage());  
 }  
 }  
  
 // Methods to update metrics  
 public void updateTotalIncome(double income) {  
 totalIncome += income;  
// saveDashboardData();  
 }  
  
 public void updateTotalCustomers(int count) {  
 totalCustomers += count;  
// saveDashboardData();  
 }  
  
 public void updateTotalSoldProducts(int count) {  
 totalSoldProducts += count;  
 saveDashboardData();  
 }  
 public double finalIncome(double income){  
 finalIncome+=income;  
 return finalIncome;  
 }  
 public Integer finalCustomers(int customers){  
finalCustomers += customers;  
return finalCustomers;  
}  
public Integer finalProducts(int products){  
 finalProducts += products;  
 return finalProducts;  
}  
 }

**COSTUMER PAGE:**

**Source Code:**

package com.example.cafe;  
  
import javafx.beans.property.SimpleStringProperty;  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.TableColumn;  
import javafx.scene.control.TableView;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.effect.GaussianBlur;  
import javafx.scene.layout.StackPane;  
import javafx.stage.Stage;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class CustomerPage {  
  
 public void customerPage(Stage stage,String user,ObservableList<Customer> customers) {  
  
  
 // Create the main StackPane  
 StackPane mainPane = new StackPane();  
  
 // StackPane 1: Background Image  
 StackPane backgroundStack = new StackPane();  
 backgroundStack.setPrefSize(1000, 600); // Set preferred size of the background pane  
 // Set the background image (CSS will handle this)  
 backgroundStack.setId("backgroundStack");  
 backgroundStack.setEffect(new GaussianBlur(10));  
  
 // StackPane 2: Content (TableView and other UI components)  
 StackPane contentStack = new StackPane();  
// contentStack.setStyle("-fx-background-color: rgba(255, 255, 255, 0.7);"); // Transparent white background  
 contentStack.setPrefSize(1000, 600);  
  
 // TableView for Customer Information  
 TableView<Customer> customerTable = new TableView<>();  
 customerTable.setId("customerTable");  
  
 // Define columns for the table  
 TableColumn<Customer, String> customerIdColumn = new TableColumn<>("Customer ID");  
 customerIdColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getId()));  
 TableColumn<Customer, String> customerNumberColumn = new TableColumn<>("Customer Number");  
 customerNumberColumn.setCellValueFactory(cellData -> new SimpleStringProperty(cellData.getValue().getNumber()));  
  
 // Add columns to the table  
 customerTable.getColumns().addAll(customerIdColumn, customerNumberColumn);  
  
 // Add customer list to the table  
 customerTable.setItems(customers);  
 customerTable.refresh();  
 // Add table to the content stack  
 contentStack.getChildren().add(customerTable);  
 Button backButtononLogin = new Button();  
 backButtononLogin.setAlignment(Pos.*TOP\_LEFT*);  
 backButtononLogin.getStyleClass().add("backBtn");  
 backButtononLogin.setOnAction(e->{  
 Menu menu = new Menu();  
 menu.menu(stage,user);  
 });  
 // Add the background and content stacks to the main pane  
 mainPane.getChildren().addAll(backgroundStack, contentStack,backButtononLogin);  
  
 // Create scene and set it to the stage  
 Scene scene = new Scene(mainPane, 1000, 600);  
  
 // Apply CSS styles to the scene  
 var stylesheet = getClass().getResource("/com/example/cafe/style.css");  
 if (stylesheet != null) {  
 scene.getStylesheets().add(stylesheet.toExternalForm());  
 System.*out*.println("CSS loaded successfully");  
 } else {  
 System.*out*.println("CSS not found");  
 }  
  
 stage.setTitle("Customer Page");  
 stage.setScene(scene);  
 stage.show();  
 }  
}

**STYLE SHEET:**

#background {  
 -fx-background-image: url(home8.jpg);  
 -fx-background-size: 100% 100%; /\* Ensure the image fits within the scene without stretching or cropping \*/  
 -fx-background-repeat: no-repeat; /\* Prevent image repetition \*/  
 -fx-background-attachment: fixed;  
 -fx-alignment: center;  
 -fx-backround-image-height: 300;  
 -fx-bacground-image-width: 300;  
}  
  
#root {  
 -fx-alignment: center; /\* Center content inside the VBox \*/  
}  
  
.Text {  
 -fx-font-weight: bold;  
 -fx-fill: white;  
 -fx-font-size: 3em;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.3), 5, 0.2, 0, 2);/\* Text shadow effect \*/  
}  
  
#vbox2 {  
 /\*-fx-background-color: rgba(86, 110, 61, 0.8); !\* Background with opacity \*!\*/  
 -fx-padding: 3em; /\* Padding inside the VBox \*/  
 -fx-border: solid 2px black; /\* Border style \*/  
 -fx-border-radius: 20px; /\* Rounded corners \*/  
 -fx-background-radius: 20px; /\* Rounded corners for background \*/  
 -fx-alignment: center; /\* Center content inside each VBox \*/  
 -fx-max-height: 100em;  
 -fx-max-width: 20em;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 10, 0.5, 0, 4);  
  
}  
  
#btn1, #btn2 {  
 -fx-background-color: #0775ec; /\* Initial background color \*/  
 -fx-text-fill: white; /\* Text color \*/  
 -fx-font-size: 20px; /\* Font size \*/  
 -fx-padding: 10px 30px; /\* Padding for size \*/  
 -fx-border-radius: 20px; /\* Rounded border corners \*/  
 -fx-background-radius: 20px; /\* Rounded background corners \*/  
 -fx-cursor: hand; /\* Pointer cursor on hover \*/  
 -fx-effect: dropshadow(three-pass-box, rgba(0,0,0,0.3), 5, 0, 2, 2); /\* Optional shadow \*/  
 -fx-transition: all 0.5s ease-in-out; /\* Smooth transition for all properties \*/  
}  
#btn1:pressed,#btn2:pressed {  
 -fx-background-color: #0775ec; /\* Even darker green \*/  
 -fx-scale-x: 1; /\* Reset scaling \*/  
 -fx-scale-y: 1;  
}  
  
/\* Button Hover Effect \*/  
#btn1:hover, #btn2:hover {  
 -fx-background-color: #0775ec;  
 -fx-opacity: 0.5;/\* Darker background on hover \*/  
 -fx-padding: 10px 50px; /\* Increase padding for size change \*/  
 -fx-border-radius: 20px; /\* Additional rounding on hover \*/  
 -fx-background-radius: 20px;  
 -fx-scale-x: 1.05; /\* Slight scale effect \*/  
 -fx-scale-y: 1.05;  
}  
#loginBackground{  
 -fx-background-image: url(home17.jpg);  
 -fx-background-size: 100% 100%;  
 -fx-background-repeat: no-repeat;  
 /\*-fx-background-position: center center;\*/  
 -fx-alignment: center;  
 -fx-backgorund-attachment: fixed;  
}  
#backButton{  
 -fx-background-image: url(arrow\_back\_24dp\_E8EAED\_FILL0\_wght400\_GRAD0\_opsz24.png);  
 -fx-background-size: cover;  
 -fx-max-height: 10px;  
 -fx-max-width: 30px;  
 -fx-background-color: transparent;  
 -fx-cursor: hand;  
  
}  
  
.backBtn {  
 -fx-background-image: url('arrow\_back\_24dp\_E8EAED\_FILL0\_wght400\_GRAD0\_opsz24.png');  
 -fx-background-size: cover;  
 -fx-max-height: 24px; /\* Match the image size \*/  
 -fx-max-width: 24px;  
 -fx-background-color: transparent;  
 -fx-cursor: hand;  
  
 /\* Ensure proper alignment \*/  
 -fx-translate-x: -610px; /\* Adjust as needed based on layout \*/  
 -fx-translate-y: -295px;  
 -fx-padding: 12px 22px;  
  
  
}  
  
  
#pane3 {  
  
 -fx-padding: 5em; /\* Padding around content \*/  
 -fx-padding-left:5em;  
 -fx-border: solid 2px black; /\* Border style \*/  
 -fx-border-radius: 20px; /\* Rounded corners \*/  
 -fx-background-radius: 20px; /\* Rounded corners for background \*/  
 -fx-alignment: center; /\* Center content inside the pane \*/  
 -fx-max-height: 30em;  
 -fx-max-width: 25em;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 10, 0.5, 0, 4);  
}  
  
#hyperlink {  
 -fx-font-size: 18px; /\* Increased font size for the hyperlink \*/  
 -fx-text-fill: lightblue;  
 -fx-alignment: center; /\* Center-align text in the hyperlink \*/  
 -fx-cursor: hand;  
 -fx-underline: false; /\* Add underline for better visibility \*/  
 -fx-faint-focus-color: transparent;  
 -fx-focus-color: transparent;  
}  
  
#vbox3 {  
 -fx-padding: 2em; /\* Padding inside the VBox \*/  
 -fx-border-radius: 10px; /\* Rounded corners \*/  
 -fx-background-radius: 10px;  
 -fx-alignment: center; /\* Center-align VBox content \*/  
 -fx-max-height: 40em;  
}  
  
#backButton {  
 -fx-background-image: url(arrow\_back\_24dp\_E8EAED\_FILL0\_wght400\_GRAD0\_opsz24.png);  
 -fx-background-size: cover;  
 -fx-background-color: transparent;  
 -fx-min-height: 40px; /\* Increased button size \*/  
 -fx-min-width: 40px; /\* Increased button size \*/  
 -fx-cursor: hand;  
}  
#usernameLabel{  
 -fx-text-fill: white;  
 -fx-font-size: 1.5em;  
 -fx-font-weight: bold;  
  
  
  
}  
#loginButton {  
 -fx-background-color: #0775ec; /\* Button background color \*/  
 -fx-text-fill: white; /\* Text color \*/  
 -fx-font-size: 22px; /\* Increased font size \*/  
 -fx-padding: 10px 30px; /\* Increased padding for larger buttons \*/  
 -fx-border-radius: 20px; /\* Rounded border corners \*/  
 -fx-background-radius: 20px; /\* Rounded background corners \*/  
 -fx-cursor: hand; /\* Pointer cursor on hover \*/  
 -fx-effect: dropshadow(three-pass-box, rgba(0,0,0,0.3), 5, 0, 2, 2); /\* Optional shadow \*/  
 -fx-transition: all 0.5s ease-in-out; /\* Smooth transition for all properties \*/  
}  
  
#username, #password {  
 -fx-font-size: 18px; /\* Font size for input text \*/  
 -fx-pref-height: 40px; /\* Height of the text fields \*/  
 -fx-pref-width: 300px; /\* Width of the text fields \*/  
 -fx-padding: 5px; /\* Padding inside text fields \*/  
 -fx-background-color: #ffffff; /\* Background color \*/  
 -fx-border-color: #61a2b1; /\* Border color \*/  
 -fx-border-width: 2px; /\* Border thickness \*/  
 -fx-background-radius: 15px; /\* Rounded corners for the background \*/  
 -fx-border-radius: 15px; /\* Rounded corners for the border \*/  
}  
  
#passwordLabel {  
 -fx-text-fill: white;  
 -fx-font-size: 1.5em;  
 -fx-font-weight: bold;  
  
}  
#loginButton:hover{  
 -fx-background-color: #0775ec; /\* Darker background on hover \*/  
 -fx-opacity: 0.5;  
 -fx-padding: 10px 40px; /\* Increase padding for size change \*/  
 -fx-border-radius: 20px; /\* Additional rounding on hover \*/  
 -fx-background-radius: 20px;  
}  
#signImage{  
 -fx-background-image: url(home9.jpg);  
 -fx-background-size: 100% 100%;  
 -fx-background-repeat: no-repeat;  
 /\*-fx-background-position: center center;\*/  
 -fx-alignment: center;  
 -fx-backgorund-attachment: fixed;  
}  
.textDesign{  
 -fx-text-fill: white;  
 -fx-font-size: 1.5em;  
 -fx-font-weight: bold;  
}  
  
.fieldDesign {  
 -fx-font-size: 18px;  
 -fx-pref-height: 40px;  
 -fx-pref-width: 250px;  
 -fx-padding: 5px;  
 -fx-background-color: #ffffff;  
 -fx-border-color: #61a2b1;  
 -fx-border-width: 2px;  
 -fx-background-radius: 10px;  
 -fx-border-radius: 10px;  
}  
  
#vbox4 {  
 -fx-padding: 2em; /\* Padding inside the VBox \*/  
 -fx-border-radius: 10px; /\* Rounded corners \*/  
 -fx-background-radius: 10px;  
 -fx-alignment: center; /\* Center-align VBox content \*/  
 -fx-max-height: 35em; /\* Reduce max height \*/  
 -fx-max-width: 25em; /\* Reduce max width \*/  
 -fx-spacing: 10px; /\* Spacing between elements inside VBox \*/  
}  
#forgetImg{  
 -fx-background-image: url(home7.jpg);  
 -fx-background-size: 100% 100%;  
 -fx-background-repeat: no-repeat;  
 -fx-alignment: center;  
 -fx-background-attachment: fixed;  
  
}  
.eyeButton {  
 -fx-background-color: transparent;  
 -fx-border: none;  
 -fx-cursor: hand;  
 -fx-font-size: 16px;  
 -fx-text-fill: gray;  
}  
  
.eyeButton:hover {  
 -fx-text-fill: #61a2b1; /\* Change color on hover \*/  
}  
  
  
.button1 {  
 -fx-background-color: #0775ec; /\* Button color \*/  
 -fx-text-fill: white; /\* White text \*/  
 -fx-font-size: 18px; /\* Font size \*/  
 -fx-font-weight: bold; /\* Bold font \*/  
 -fx-border-radius: 15; /\* Rounded corners \*/  
 -fx-background-radius: 15; /\* Match background corners with border \*/  
 -fx-padding: 10 30; /\* Padding for height and width \*/  
 -fx-border-width: 2; /\* Border width \*/  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.3), 5, 0.2, 0, 2); /\* Reduced shadow \*/  
}  
  
/\* Hover effect for buttons with slightly larger shadow \*/  
.button1:hover {  
 -fx-background-color: #0775ec; /\* Same color \*/  
 -fx-opacity: 0.6;  
 -fx-text-fill: white;  
 -fx-cursor: hand; /\* Pointer cursor \*/  
 -fx-scale-x: 1.03;  
 -fx-scale-y: 1.03;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.4), 6, 0.2, 0, 3); /\* Slightly larger shadow on hover \*/  
}  
  
/\* Button press effect with minimal shadow \*/  
.button1:pressed {  
 -fx-background-color: #0775ec;  
 -fx-scale-x: 1;  
 -fx-scale-y: 1;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.2), 4, 0.1, 0, 1); /\* Very subtle shadow on press \*/  
}  
  
  
#menupic{  
 -fx-background-image: url(pexels-scottwebb-2530586.jpg);  
 -fx-background-size: 100% 100%;  
 -fx-background-repeat: no-repeat;  
 -fx-alignment: center;  
  
  
}  
  
  
  
#menuVbox{  
 -fx-padding: 4em; /\* Padding inside the VBox \*/  
 -fx-border-radius: 10px; /\* Rounded corners \*/  
 -fx-background-radius: 10px;  
 -fx-alignment: center; /\* Center-align VBox content \*/  
 -fx-max-height: 70em; /\* Reduce max height \*/  
 -fx-max-width: 40em; /\* Reduce max width \*/  
 -fx-spacing: 25px; /\* Spacing between elements inside VBox \*/  
  
  
}  
  
  
/\* Content StackPane styling \*/  
#background-layer {  
 -fx-padding: 10;  
 -fx-background-image: url(home11.jpg);  
 -fx-background-size: 100% 100%;  
 -fx-background-repeat: no-repeat;  
 -fx-alignment: center;  
}  
  
/\* Title styling \*/  
#title-label {  
 -fx-font-size: 24px;  
 -fx-font-weight: bold;  
 -fx-text-fill: white;  
 -fx-padding: 15px 0 0 115px;  
}  
  
/\* Table styling \*/  
#inventory-table {  
 -fx-border-color: #d4d4d4;  
 -fx-border-width: 2px;  
 -fx-background-radius: 8px;  
 -fx-border-radius: 8px;  
 -fx-font-size: 14px;  
 -fx-padding: 10;  
}  
  
/\* Form container \*/  
#form-container {  
 -fx-padding: 20;  
 -fx-border-color: #d4d4d4;  
 -fx-border-width: 2px;  
 -fx-background-color: #ffffff;  
 -fx-border-radius: 8px;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 10, 0.5, 0, 4);  
 -fx-background-radius: 8px;  
}  
  
/\* Content container styling \*/  
#content-container {  
 -fx-spacing: 20px;  
 -fx-padding: 10px;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 10, 0.5, 0, 4);  
 -fx-alignment: center;  
}  
  
/\* Input field styling \*/  
.input-field {  
 -fx-pref-width: 200px;  
 -fx-padding: 8px;  
 -fx-border-color: #7b8b9a;  
 -fx-border-width: 2px;  
 -fx-background-radius: 5px;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 10, 0.5, 0, 4);  
 -fx-border-radius: 5px;  
 -fx-font-size: 14px;  
}  
  
/\* Button styling \*/  
.action-button {  
 -fx-background-color: #0078d7;  
 -fx-text-fill: white;  
 -fx-font-size: 14px;  
 -fx-font-weight: bold;  
 -fx-border-radius: 5px;  
 -fx-background-radius: 5px;  
 -fx-padding: 10 20;  
 -fx-cursor: hand;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.2), 5, 0.2, 0, 2);  
}  
  
.action-button:hover {  
 -fx-background-color: #005bb5;  
 -fx-opacity: 0.6;  
}  
  
/\* Image preview \*/  
.image-preview {  
 -fx-border-color: #ccc;  
 -fx-border-width: 2px;  
 -fx-border-radius: 8px;  
 -fx-background-radius: 8px;  
}  
/\* Main pane styling \*/  
.main-pane {  
 -fx-background-color: #f5f5f5;  
}  
  
/\* Background pane styling \*/  
.background-pane {  
 -fx-background-image: url("background.jpg");  
 -fx-background-size: cover;  
 -fx-background-position: center;  
}  
  
/\* Content box styling \*/  
.content-box {  
 -fx-background-color: rgba(255, 255, 255, 0.9);  
 -fx-background-radius: 10;  
 -fx-padding: 20;  
}  
  
/\* Menu label styling \*/  
.menu-label {  
 -fx-font-size: 24px;  
 -fx-text-fill: #333;  
}  
  
/\* Menu container styling \*/  
.menu-container {  
 -fx-spacing: 10;  
}  
  
/\* Menu item styling \*/  
.menu-item {  
 -fx-border-color: #ccc;  
 -fx-border-width: 1;  
 -fx-border-radius: 5;  
 -fx-padding: 10;  
 -fx-background-color: #ffffff;  
 -fx-background-radius: 5;  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.2), 3, 0, 1, 1);  
}  
  
.menu-item:hover {  
 -fx-background-color: #f0f8ff;  
}  
  
.menu-item-name {  
 -fx-font-size: 16px;  
 -fx-text-fill: #444;  
}  
  
.menu-item-price {  
 -fx-font-size: 14px;  
 -fx-text-fill: #777;  
}  
  
.menu-item-button {  
 -fx-background-color: #28a745;  
 -fx-text-fill: #fff;  
 -fx-font-size: 14px;  
 -fx-border-radius: 5;  
 -fx-padding: 5 10;  
}  
  
.menu-item-button:hover {  
 -fx-background-color: #218838;  
}  
  
/\* Menu item image styling \*/  
.menu-item-image {  
 -fx-effect: dropshadow(gaussian, rgba(0, 0, 0, 0.5), 5, 0, 2, 2);  
}  
  
/\* Scroll pane styling \*/  
.menu-scroll-pane {  
 -fx-background-color: transparent;  
 -fx-border-color: transparent;  
}  
  
/\* Order table styling \*/  
.order-table {  
 -fx-border-color: #ccc;  
 -fx-border-width: 1;  
}  
  
/\* Payment section styling \*/  
.payment-box {  
 -fx-spacing: 10;  
}  
  
.total-label {  
 -fx-font-size: 18px;  
 -fx-text-fill: #000;  
}  
  
.amount-paid-field {  
 -fx-background-color: #fff;  
 -fx-border-color: #ccc;  
 -fx-border-radius: 5;  
 -fx-padding: 5;  
}  
  
.pay-button {  
 -fx-background-color: #007bff;  
 -fx-text-fill: #fff;  
 -fx-border-radius: 5;  
 -fx-padding: 5 10;  
}  
  
.pay-button:hover {  
 -fx-background-color: #0056b3;  
}  
  
.remove-button {  
 -fx-background-color: #dc3545;  
 -fx-text-fill: #fff;  
 -fx-border-radius: 5;  
 -fx-padding: 5 10;  
}  
  
.remove-button:hover {  
 -fx-background-color: #c82333;  
}  
  
.print-button {  
 -fx-background-color: #ffc107;  
 -fx-text-fill: #000;  
 -fx-border-radius: 5;  
 -fx-padding: 5 10;  
}  
  
.print-button:hover {  
 -fx-background-color: #e0a800;  
}  
  
.change-label {  
 -fx-font-size: 18px;  
 -fx-text-fill: #000;  
}