# **THEORY ASSIGNMENT 04**

This is my main class that leads us to a interface consisting of admin login and user login.

package com.example.projectwithgui;  
  
import javafx.animation.ScaleTransition;  
import javafx.application.Application;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.stage.Stage;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.util.Duration;  
  
public class Main extends Application {  
 //This is my main class that further leads us to respective stages  
 @Override  
 public void start(Stage primaryStage) {  
  
 Image icon = new Image("Icon.png"); // Path to the icon  
 primaryStage.getIcons().add(icon);  
 //background image set  
 Image backgroundImage = new Image("orangebg.png");  
 ImageView backgroundView = new ImageView(backgroundImage);  
 backgroundView.setFitWidth(1550);  
 backgroundView.setFitHeight(720);  
 backgroundView.setPreserveRatio(false);  
 primaryStage.setFullScreen(true);  
  
 //welcome label  
 Label welcomeLabel = new Label("Welcome to ATDOOR");  
 welcomeLabel.setFont(Font.*font*("Halvetica", FontWeight.*BOLD*, 52));  
 welcomeLabel.setTextFill(Color.*web*("#d83724"));  
 welcomeLabel.setStyle("-fx-effect: dropshadow(gaussian, #ffffff, 5, 0.5, 2, 2);");  
 welcomeLabel.setAlignment(Pos.*CENTER*);  
 welcomeLabel.setStyle("-fx-font-weight: bold;");  
  
  
  
 Button adminLoginButton = new Button("Admin Login");  
 Button userLoginButton = new Button("User Login");  
 Button exitbutton = new Button("Exit");  
 adminLoginButton.setEffect(new DropShadow());  
 userLoginButton.setEffect(new DropShadow());  
  
 //beautifying the buttons  
 styleButton(adminLoginButton);  
 styleButton(userLoginButton);  
 styleexitbutton(exitbutton);  
 addButtonAnimation(adminLoginButton);  
 addButtonAnimation(userLoginButton);  
  
 //defining functionalities  
 adminLoginButton.setOnAction(e -> {  
 AdminLogin adminLogin = new AdminLogin(primaryStage);  
 adminLogin.show();  
 });  
 userLoginButton.setOnAction(e -> {  
 UserLoginOptions userLoginOptions = new UserLoginOptions(primaryStage);  
 userLoginOptions.show();  
 });  
 exitbutton.setOnAction(e -> {  
 primaryStage.close();  
 });  
  
 VBox buttonLayout = new VBox(20);  
 buttonLayout.getChildren().addAll(adminLoginButton, userLoginButton,exitbutton);  
 buttonLayout.setAlignment(Pos.*CENTER*);  
  
 VBox mainLayout = new VBox(30);  
 mainLayout.getChildren().addAll(welcomeLabel, buttonLayout);  
 mainLayout.setAlignment(Pos.*TOP\_CENTER*);  
 mainLayout.setPadding(new Insets(330, 0, 0, 0));  
 //using staack pane bcz buttons are stacked over backgorund  
 StackPane root = new StackPane();  
 root.getChildren().addAll(backgroundView, mainLayout);  
 //adding everything to scene  
 Scene scene = new Scene(root, 1550, 670);  
 primaryStage.setTitle("E-Commerce System - Login");  
 primaryStage.setScene(scene);  
 primaryStage.setFullScreen(true);  
 primaryStage.setResizable(true);  
 primaryStage.setFullScreenExitHint("");  
 primaryStage.setFullScreenExitKeyCombination(null);  
 primaryStage.show();  
 }  
 //styling buttons method  
// private void styleButton(Button button) {  
// button.setFont(new Font("Didot", 18));  
// button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; -fx-padding: 10 20; "  
// + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73;-fx-border-width: 2;");  
// button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;"));  
// button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;"));  
// }  
  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 16));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
  
 private void styleexitbutton(Button button){  
 button.setFont(new Font("Didot",20));  
 button.setStyle("-fx-background-color: #a85700; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #813303;-fx-border-width: 2;");  
 }  
 //buttons animations  
 private void addButtonAnimation(Button button) {  
 ScaleTransition scaleTransition = new ScaleTransition(Duration.*millis*(200), button);  
 scaleTransition.setFromX(1.0);  
 scaleTransition.setFromY(1.0);  
 scaleTransition.setToX(1.1);  
 scaleTransition.setToY(1.1);  
 button.setOnMouseEntered(e -> scaleTransition.playFromStart());  
 button.setOnMouseExited(e -> scaleTransition.stop());  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

If we click admin login admin panel is opened that is given below  
package com.example.projectwithgui;  
  
import javafx.application.Platform;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.control.Label;  
import javafx.scene.control.PasswordField;  
import javafx.scene.control.TextField;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.HBox;  
import javafx.scene.layout.StackPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.stage.Stage;  
  
public class AdminLogin {  
 private final String ADMIN\_USERNAME = "Wajahat";  
 private final String ADMIN\_PASSWORD = "1234";  
 private final Stage stage;  
 private volatile boolean showError = true;  
  
 public AdminLogin(Stage stage) {  
 this.stage = stage;  
 }  
  
 public void show() {  
 //setting the complete interface  
 Image backgroundImage = new Image("orangebg.png");  
 ImageView backgroundView = new ImageView(backgroundImage);  
 backgroundView.setFitWidth(1550);  
 backgroundView.setFitHeight(720);  
 backgroundView.setPreserveRatio(false);  
  
 Label usernameLabel = new Label("Username:");  
 usernameLabel.setFont(new Font("Didot", 18));  
 usernameLabel.setTextFill(Color.*BLACK*);  
  
 TextField usernameField = new TextField();  
 usernameField.setPrefWidth(250);  
 usernameField.setStyle("-fx-background-radius: 5; -fx-padding: 5 10;");  
  
 HBox usernameLayout = new HBox(10, usernameLabel, usernameField);  
 usernameLayout.setAlignment(Pos.*CENTER*);  
  
 Label passwordLabel = new Label("Password:");  
 passwordLabel.setFont(new Font("Didot", 18));  
 passwordLabel.setTextFill(Color.*BLACK*);  
  
 PasswordField passwordField = new PasswordField();  
 passwordField.setPrefWidth(250);  
 passwordField.setStyle("-fx-background-radius: 5; -fx-padding: 5 10;");  
  
 HBox passwordLayout = new HBox(10, passwordLabel, passwordField);  
 passwordLayout.setAlignment(Pos.*CENTER*);  
  
 Label statusLabel = new Label();  
 statusLabel.setFont(new Font("Didot", 14));  
 statusLabel.setTextFill(Color.*BLACK*);  
 statusLabel.setVisible(false);  
  
 Button loginButton = new Button("Login");  
 Button cancelButton = new Button("Cancel");  
 styleButton(loginButton);  
 styleButton(cancelButton);  
  
 loginButton.setOnAction(e -> {  
 String username = usernameField.getText().trim();  
 String password = passwordField.getText().trim();  
  
 if (username.isEmpty() || password.isEmpty()) {  
 showBlinkingNotification(statusLabel, "\*\*Fields can't be left empty!\*\*");  
 } else if (username.equals(ADMIN\_USERNAME) && password.equals(ADMIN\_PASSWORD)) {  
 showError = false;  
 ShowAdminMenu adminMenu = new ShowAdminMenu();  
 adminMenu.show(stage);  
 } else {  
 showBlinkingNotification(statusLabel, "\*\*Invalid Credentials!\*\*");  
 }  
 });  
  
 cancelButton.setOnAction(e -> {  
 showError = false;  
 Main main = new Main();  
 main.start(stage);  
 });  
  
 HBox buttonLayout = new HBox(25, loginButton, cancelButton);  
 buttonLayout.setPadding(new Insets(0, 0,0 , 130));  
 buttonLayout.setAlignment(Pos.*CENTER*);  
  
 VBox formLayout = new VBox(20, usernameLayout, passwordLayout, buttonLayout,statusLabel);  
 formLayout.setAlignment(Pos.*CENTER*);  
 formLayout.setPadding(new Insets(300, 0, 0, 0));  
  
 StackPane root = new StackPane();  
 root.getChildren().addAll(backgroundView, formLayout);  
  
 Scene scene = new Scene(root);  
 stage.setFullScreen(true);  
 stage.setScene(scene);  
 stage.setFullScreenExitHint("");  
 stage.setFullScreenExitKeyCombination(null);  
 stage.setTitle("Admin Login");  
 }  
  
// private void styleButton(Button button) {  
// button.setFont(new Font("Didot", 18));  
// button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; -fx-padding: 10 20; "  
// + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;");  
// button.setEffect(new DropShadow());  
// button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #031b73; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #263988; -fx-border-width: 2;"));  
// button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;"));  
// }  
  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 16));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
 //method that calls a thread that blinks the notifcation  
 private void showBlinkingNotification(Label statusLabel, String message) {  
 showError = true;  
 statusLabel.setText(message);  
 statusLabel.setVisible(true);  
  
 new Thread(() -> {  
 while (showError) {  
 Platform.*runLater*(() -> statusLabel.setVisible(!statusLabel.isVisible()));  
 try {  
 Thread.*sleep*(500);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 Platform.*runLater*(() -> statusLabel.setVisible(false));  
 }).start();  
 }  
}

After successfully logging there opens a admin menu where he has 6 main options in which almost 4 are functional  
package com.example.projectwithgui;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.GridPane;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
  
public class ShowAdminMenu {  
 public void show(Stage stage) {  
  
 VBox root = new VBox(20);  
 root.setPadding(new Insets(20, 0, 0, 0));  
  
 // Create a custom MenuBar with only a Cancel button  
 root.getChildren().add(createMenuBar(stage));  
  
 GridPane gridPane = new GridPane();  
 gridPane.setHgap(90); // Horizontal gap between buttons  
 gridPane.setVgap(90); // Vertical gap between buttons  
 gridPane.setAlignment(Pos.*CENTER*);  
  
 // Create buttons with images  
 Button categoriesButton = createImageButton("categories.png");  
 Button viewSalesButton = createImageButton("view\_sales.png");  
 Button addCategoryButton = createImageButton("add\_category.png");  
 Button productsManagementButton = createImageButton("products\_management.png");  
 Button viewRevenueButton = createImageButton("view\_revenue.png");  
 Button customersDatabaseButton = createImageButton("customers\_database.png");  
  
 categoriesButton.setOnAction(e -> {  
 Categories categories = new Categories();  
 categories.show(stage);  
 });  
 viewSalesButton.setOnAction(e -> {  
 Sales sales = new Sales();  
 sales.show(stage);  
 });  
 addCategoryButton.setOnAction(e -> {  
 AddNewCategory addNewCategory = new AddNewCategory();  
 addNewCategory.show(stage);  
 });  
 productsManagementButton.setOnAction(e -> {  
 ProductsManagement productsManagement = new ProductsManagement();  
 productsManagement.show(stage);  
 });  
 customersDatabaseButton.setOnAction(e -> {  
 CustomersDatabase cdb = new CustomersDatabase();  
 cdb.show(stage);  
 });  
 viewRevenueButton.setOnAction(e -> {  
 Revenue revenue = new Revenue();  
 revenue.show();  
 });  
  
 // Add buttons to the grid  
 gridPane.add(categoriesButton, 0, 0);  
 gridPane.add(viewSalesButton, 1, 0);  
 gridPane.add(addCategoryButton, 2, 0);  
 gridPane.add(productsManagementButton, 0, 1);  
 gridPane.add(viewRevenueButton, 1, 1);  
 gridPane.add(customersDatabaseButton, 2, 1);  
  
 // Add the GridPane to the VBox  
 root.getChildren().add(gridPane);  
  
 // Create a scene and set it on the stage  
 Scene scene = new Scene(root, 1500, 670);  
 stage.setScene(scene);  
 stage.setTitle("Admin Menu");  
 stage.setFullScreen(true);  
 stage.setFullScreenExitHint("");  
 stage.setFullScreenExitKeyCombination(null);  
 root.setStyle("-fx-background-color: #ffc784;");  
 stage.show();  
 }  
  
 private VBox createMenuBar(Stage stage) {  
 // Create a VBox for the "Menu Bar" with a Cancel button  
 VBox menuBar = new VBox();  
 Button cancelButton = new Button("Cancel");  
 cancelButton.setStyle("-fx-font-size: 13px; -fx-background-color: #d88f1e; -fx-text-fill: #000000;");  
 cancelButton.setOnAction(e -> {  
 Main main = new Main();  
 main.start(stage); // Navigate back to the Main class  
 });  
  
 // Add the cancel button to the menu bar  
 menuBar.getChildren().add(cancelButton);  
 menuBar.setPadding(new Insets(10, 0, 0, 10)); // Set padding for the button  
  
 return menuBar;  
 }  
  
 private Button createImageButton(String imagePath) {  
 Image image = new Image(imagePath); // Load the image  
 ImageView imageView = new ImageView(image);  
 imageView.setFitWidth(200);  
 imageView.setFitHeight(200); // To make the pictures of a specific size  
 imageView.setPreserveRatio(true);  
  
 Button button = new Button();  
 button.setGraphic(imageView); // Add picture as graphics of button  
 button.setStyle("-fx-background-radius: 20; -fx-border-radius: 20; -fx-background-color: #ffdc97;");  
 button.setEffect(new DropShadow(10, Color.*BLACK*));  
 button.setOnMouseEntered(e -> button.setEffect(new DropShadow(20, Color.*DARKGRAY*)));  
 button.setOnMouseExited(e -> button.setEffect(new DropShadow(10, Color.*GRAY*)));  
  
 button.setPrefSize(250, 250); // Set button size  
 return button;  
 }  
}

In admin menu there is a button to add new category another one to show all categories. A button on right bottom to show customers database( all the usernames with their passwords). Another button to display total revenue till now generated.  
A button to manage products that has further options to add and remove products…leading to different stages…with functionality

package com.example.projectwithgui;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.GridPane;  
import javafx.stage.FileChooser;  
import javafx.stage.Stage;  
  
import java.io.\*;  
import java.util.List;  
  
public class ProductsManagementFunctions {  
  
 public void show(Stage stage) {  
 GridPane gridPane = new GridPane();  
 gridPane.setHgap(20);  
 gridPane.setVgap(20);  
 gridPane.setPadding(new Insets(20));  
 gridPane.setAlignment(Pos.*CENTER*);  
 //to select category you want the new product to be under stored  
 ComboBox<String> categoryComboBox = new ComboBox<>();  
 List<String> categories = CategoryManager.*getCategories*();  
 if (categories.isEmpty()) {  
 categoryComboBox.setPromptText("No categories available");  
 } else {  
 categoryComboBox.getItems().addAll(categories);  
 categoryComboBox.setPromptText("Select a Category");  
 }  
  
 Label productNameLabel = new Label("Product Name:");  
 TextField productNameField = new TextField();  
 productNameField.setPromptText("Enter Product Name");  
  
 Label productPriceLabel = new Label("Product Price:");  
 TextField productPriceField = new TextField();  
 productPriceField.setPromptText("Enter Product Price");  
  
 Label productQuantityLabel = new Label("Product Quantity:");  
 TextField productQuantityField = new TextField();  
 productQuantityField.setPromptText("Enter Product Quantity");  
  
 Label discountLabel = new Label("Discount (if any):");  
 TextField discountField = new TextField();  
 discountField.setPromptText("Enter Discount (if any)");  
  
 Label descriptionLabel = new Label("Description:");  
 TextArea descriptionField = new TextArea();  
 descriptionField.setPromptText("Enter Product Description");  
 descriptionField.setWrapText(true);  
  
 // File chooser for product image it chooses file from the system  
 Button chooseImageButton = new Button("Choose Product Image");  
 styleButton(chooseImageButton);  
 Label selectedImageLabel = new Label("No file selected");  
 ImageView productImageView = new ImageView();  
 //it is used to store absolute path of image  
 String[] absolutePath = new String[1];  
  
 chooseImageButton.setOnAction(e -> {  
 FileChooser fileChooser = new FileChooser();  
 fileChooser.setTitle("Select Product Image");  
 fileChooser.getExtensionFilters().add(  
 new FileChooser.ExtensionFilter("Image Files", "\*.png", "\*.jpg", "\*.jpeg")  
 );  
  
  
 File selectedFile = fileChooser.showOpenDialog(stage);  
  
 if (selectedFile != null) {  
 try {  
 // Save the absolute path of the selected image  
 absolutePath[0] = selectedFile.getAbsolutePath();  
  
 // Load and display the image using the selected file path  
 Image productImage = new Image(new FileInputStream(selectedFile), 250, 250, true, true);  
 productImageView.setImage(productImage);  
  
 selectedImageLabel.setText("Image Selected: " + absolutePath[0]);  
 } catch (FileNotFoundException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Failed to load image.");  
 }  
 } else {  
 selectedImageLabel.setText("No file selected");  
 }  
 });  
  
 Button addProductButton = new Button("Add Product");  
 styleButton(addProductButton);  
 Button goBackButton = new Button("Go Back");  
 styleButton(goBackButton);  
 //metod to save all prodcut details in file  
 addProductButton.setOnAction(e -> {  
 saveProductDetails(  
 categoryComboBox,  
 productNameField,  
 productPriceField,  
 productQuantityField,  
 discountField,  
 descriptionField,  
 selectedImageLabel,  
 productImageView  
 );  
 });  
  
 // Action to go back  
 goBackButton.setOnAction(e -> {  
 ProductsManagement productsManagement = new ProductsManagement();  
 productsManagement.show(stage);  
 });  
  
 gridPane.add(new Label("Category:"), 0, 0);  
 gridPane.add(categoryComboBox, 1, 0);  
  
 gridPane.add(productNameLabel, 0, 1);  
 gridPane.add(productNameField, 1, 1);  
  
 gridPane.add(productPriceLabel, 0, 2);  
 gridPane.add(productPriceField, 1, 2);  
  
 gridPane.add(productQuantityLabel, 0, 3);  
 gridPane.add(productQuantityField, 1, 3);  
  
 gridPane.add(discountLabel, 0, 4);  
 gridPane.add(discountField, 1, 4);  
  
 gridPane.add(descriptionLabel, 0, 5);  
 gridPane.add(descriptionField, 1, 5);  
  
 gridPane.add(chooseImageButton, 0, 6);  
 gridPane.add(selectedImageLabel, 1, 6);  
 gridPane.add(productImageView, 1, 7);  
  
 gridPane.add(addProductButton, 0, 8);  
 gridPane.add(goBackButton, 4, 8);  
  
  
 Scene scene = new Scene(gridPane, 1550, 670);  
 stage.setScene(scene);  
 stage.setTitle("Manage Products");  
 stage.show();  
 }  
  
 private void saveProductDetails(  
 ComboBox<String> categoryComboBox,  
 TextField productNameField,  
 TextField productPriceField,  
 TextField productQuantityField,  
 TextField discountField,  
 TextArea descriptionField,  
 Label selectedImageLabel,  
 ImageView productImageView  
 ) {  
 String category = categoryComboBox.getValue();  
 String productName = productNameField.getText().trim();  
 String productPrice = productPriceField.getText().trim();  
 String productQuantity = productQuantityField.getText().trim();  
 String discount = discountField.getText().trim();  
 String description = descriptionField.getText().trim();  
 String image = selectedImageLabel.getText();  
  
 if (category == null || category.isEmpty() || productName.isEmpty() || productPrice.isEmpty() || productQuantity.isEmpty()) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "All fields except Discount are required.");  
 return;  
 }  
  
 try {  
 double price = Double.*parseDouble*(productPrice);  
 int quantity = Integer.*parseInt*(productQuantity);  
 double discountValue = discount.isEmpty() ? 0 : Double.*parseDouble*(discount);  
  
 // Save product details to category file  
 File categoryFile = new File(category.replaceAll("\\s+", "\_") + ".txt");  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(categoryFile, true))) {  
 writer.write("Product Name: " + productName + "|");  
 writer.write("Price: " + price + "|");  
 writer.write("Quantity: " + quantity + "|");  
 writer.write("Discount: " + discountValue + "%|");  
 writer.write("Description: " + description + "|");  
 writer.write("Image: " + (image.equals("No file selected") ? "None" : image));  
 writer.write("\n"); //seperator  
 }  
  
 showAlert(Alert.AlertType.*INFORMATION*, "Success", "Product added successfully!");  
  
 } catch (NumberFormatException | IOException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Invalid input or error while saving.");  
 }  
 }  
  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 16));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
 private void showAlert(Alert.AlertType alertType, String title, String message) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

package com.example.projectwithgui;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.GridPane;  
import javafx.stage.FileChooser;  
import javafx.stage.Stage;  
  
import java.io.\*;  
import java.util.List;  
  
public class ProductsManagementFunctions {  
  
 public void show(Stage stage) {  
 GridPane gridPane = new GridPane();  
 gridPane.setHgap(20);  
 gridPane.setVgap(20);  
 gridPane.setPadding(new Insets(20));  
 gridPane.setAlignment(Pos.*CENTER*);  
 //to select category you want the new product to be under stored  
 ComboBox<String> categoryComboBox = new ComboBox<>();  
 List<String> categories = CategoryManager.*getCategories*();  
 if (categories.isEmpty()) {  
 categoryComboBox.setPromptText("No categories available");  
 } else {  
 categoryComboBox.getItems().addAll(categories);  
 categoryComboBox.setPromptText("Select a Category");  
 }  
  
 Label productNameLabel = new Label("Product Name:");  
 TextField productNameField = new TextField();  
 productNameField.setPromptText("Enter Product Name");  
  
 Label productPriceLabel = new Label("Product Price:");  
 TextField productPriceField = new TextField();  
 productPriceField.setPromptText("Enter Product Price");  
  
 Label productQuantityLabel = new Label("Product Quantity:");  
 TextField productQuantityField = new TextField();  
 productQuantityField.setPromptText("Enter Product Quantity");  
  
 Label discountLabel = new Label("Discount (if any):");  
 TextField discountField = new TextField();  
 discountField.setPromptText("Enter Discount (if any)");  
  
 Label descriptionLabel = new Label("Description:");  
 TextArea descriptionField = new TextArea();  
 descriptionField.setPromptText("Enter Product Description");  
 descriptionField.setWrapText(true);  
  
 // File chooser for product image it chooses file from the system  
 Button chooseImageButton = new Button("Choose Product Image");  
 styleButton(chooseImageButton);  
 Label selectedImageLabel = new Label("No file selected");  
 ImageView productImageView = new ImageView();  
 //it is used to store absolute path of image  
 String[] absolutePath = new String[1];  
  
 chooseImageButton.setOnAction(e -> {  
 FileChooser fileChooser = new FileChooser();  
 fileChooser.setTitle("Select Product Image");  
 fileChooser.getExtensionFilters().add(  
 new FileChooser.ExtensionFilter("Image Files", "\*.png", "\*.jpg", "\*.jpeg")  
 );  
  
  
 File selectedFile = fileChooser.showOpenDialog(stage);  
  
 if (selectedFile != null) {  
 try {  
 // Save the absolute path of the selected image  
 absolutePath[0] = selectedFile.getAbsolutePath();  
  
 // Load and display the image using the selected file path  
 Image productImage = new Image(new FileInputStream(selectedFile), 250, 250, true, true);  
 productImageView.setImage(productImage);  
  
 selectedImageLabel.setText("Image Selected: " + absolutePath[0]);  
 } catch (FileNotFoundException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Failed to load image.");  
 }  
 } else {  
 selectedImageLabel.setText("No file selected");  
 }  
 });  
  
 Button addProductButton = new Button("Add Product");  
 styleButton(addProductButton);  
 Button goBackButton = new Button("Go Back");  
 styleButton(goBackButton);  
 //metod to save all prodcut details in file  
 addProductButton.setOnAction(e -> {  
 saveProductDetails(  
 categoryComboBox,  
 productNameField,  
 productPriceField,  
 productQuantityField,  
 discountField,  
 descriptionField,  
 selectedImageLabel,  
 productImageView  
 );  
 });  
  
 // Action to go back  
 goBackButton.setOnAction(e -> {  
 ProductsManagement productsManagement = new ProductsManagement();  
 productsManagement.show(stage);  
 });  
  
 gridPane.add(new Label("Category:"), 0, 0);  
 gridPane.add(categoryComboBox, 1, 0);  
  
 gridPane.add(productNameLabel, 0, 1);  
 gridPane.add(productNameField, 1, 1);  
  
 gridPane.add(productPriceLabel, 0, 2);  
 gridPane.add(productPriceField, 1, 2);  
  
 gridPane.add(productQuantityLabel, 0, 3);  
 gridPane.add(productQuantityField, 1, 3);  
  
 gridPane.add(discountLabel, 0, 4);  
 gridPane.add(discountField, 1, 4);  
  
 gridPane.add(descriptionLabel, 0, 5);  
 gridPane.add(descriptionField, 1, 5);  
  
 gridPane.add(chooseImageButton, 0, 6);  
 gridPane.add(selectedImageLabel, 1, 6);  
 gridPane.add(productImageView, 1, 7);  
  
 gridPane.add(addProductButton, 0, 8);  
 gridPane.add(goBackButton, 4, 8);  
  
  
 Scene scene = new Scene(gridPane, 1550, 670);  
 stage.setScene(scene);  
 stage.setTitle("Manage Products");  
 stage.show();  
 }  
  
 private void saveProductDetails(  
 ComboBox<String> categoryComboBox,  
 TextField productNameField,  
 TextField productPriceField,  
 TextField productQuantityField,  
 TextField discountField,  
 TextArea descriptionField,  
 Label selectedImageLabel,  
 ImageView productImageView  
 ) {  
 String category = categoryComboBox.getValue();  
 String productName = productNameField.getText().trim();  
 String productPrice = productPriceField.getText().trim();  
 String productQuantity = productQuantityField.getText().trim();  
 String discount = discountField.getText().trim();  
 String description = descriptionField.getText().trim();  
 String image = selectedImageLabel.getText();  
  
 if (category == null || category.isEmpty() || productName.isEmpty() || productPrice.isEmpty() || productQuantity.isEmpty()) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "All fields except Discount are required.");  
 return;  
 }  
  
 try {  
 double price = Double.*parseDouble*(productPrice);  
 int quantity = Integer.*parseInt*(productQuantity);  
 double discountValue = discount.isEmpty() ? 0 : Double.*parseDouble*(discount);  
  
 // Save product details to category file  
 File categoryFile = new File(category.replaceAll("\\s+", "\_") + ".txt");  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(categoryFile, true))) {  
 writer.write("Product Name: " + productName + "|");  
 writer.write("Price: " + price + "|");  
 writer.write("Quantity: " + quantity + "|");  
 writer.write("Discount: " + discountValue + "%|");  
 writer.write("Description: " + description + "|");  
 writer.write("Image: " + (image.equals("No file selected") ? "None" : image));  
 writer.write("\n"); //seperator  
 }  
  
 showAlert(Alert.AlertType.*INFORMATION*, "Success", "Product added successfully!");  
  
 } catch (NumberFormatException | IOException ex) {  
 showAlert(Alert.AlertType.*ERROR*, "Error", "Invalid input or error while saving.");  
 }  
 }  
  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 16));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
 private void showAlert(Alert.AlertType alertType, String title, String message) {  
 Alert alert = new Alert(alertType);  
 alert.setTitle(title);  
 alert.setContentText(message);  
 alert.showAndWait();  
 }  
}

The stage where we can add new product  
  
package com.example.projectwithgui;  
  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.effect.DropShadow;  
import javafx.scene.layout.VBox;  
import javafx.scene.text.Font;  
import javafx.stage.Stage;  
  
public class ProductsManagement {  
 public void show(Stage stage) {  
  
 VBox vBox = new VBox(30); //to add spacing between the button  
 vBox.setAlignment(Pos.*CENTER*);  
  
 Button addProductButton = new Button("Add Product");  
 Button removeProductButton = new Button("Remove Product");  
 Button modifyProductButton = new Button("Modify Product");  
 Button goback = new Button("Go Back");  
 styleButton(addProductButton);  
 styleButton(removeProductButton);  
 styleButton(modifyProductButton);  
 styleButton(goback);  
 //functionalitites of buttons  
 addProductButton.setOnAction(e -> {  
 ProductsManagementFunctions addProduct = new ProductsManagementFunctions();  
 addProduct.show(stage);  
 });  
  
 goback.setOnAction(e -> {  
 ShowAdminMenu showAdminMenu = new ShowAdminMenu();  
 showAdminMenu.show(stage);  
 });  
  
 removeProductButton.setOnAction(e->{  
 RemoveProducts rp = new RemoveProducts();  
 rp.show(stage);  
 });  
  
 vBox.getChildren().addAll(addProductButton, removeProductButton, modifyProductButton, goback);  
 Scene scene = new Scene(vBox, 1550, 670);  
 stage.setScene(scene);  
 vBox.setStyle("-fx-background-color: #fdd880");  
 stage.setTitle("Products Management");  
 stage.setFullScreen(true);  
 stage.setFullScreenExitHint("");  
 stage.setFullScreenExitKeyCombination(null);  
 stage.show();  
 }  
// private void styleButton(Button button) {  
// button.setFont(new Font("Didot", 35)); // Set font size  
// button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; -fx-padding: 10 20; "  
// + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;");  
// button.setEffect(new DropShadow());  
//  
// button.setMinHeight(100);  
// button.setMaxHeight(100);  
//  
// button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #031b73; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #263988; -fx-border-width: 2;"));  
// button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #263988; -fx-text-fill: white; "  
// + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #031b73; -fx-border-width: 2;"));  
// }  
  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 35));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
  
 button.setMinHeight(100);  
 button.setMaxHeight(100);  
  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
}

package com.example.projectwithgui;  
  
import javafx.collections.FXCollections;  
import javafx.collections.ObservableList;  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.layout.\*;  
import javafx.scene.paint.Color;  
import javafx.stage.Stage;  
  
import java.io.BufferedReader;  
import java.io.File;  
import java.io.FileReader;  
import java.io.IOException;  
  
public class CustomersDatabase {  
 private TableView<Customers> table;  
 private ObservableList<Customers> customerList;  
  
 public void show(Stage stage) {  
 // Create a VBox layout to hold the components  
 VBox root = new VBox(20);  
 root.setPadding(new Insets(20));  
  
 HBox topMenuBar = new HBox(10);  
 topMenuBar.setStyle("-fx-background-color: #ffc784; -fx-border-color: #b5b2b2; -fx-border-width: 1;");  
  
 Button cancelButton = new Button("Cancel");  
 cancelButton.setOnAction(e -> {  
 ShowAdminMenu adminMenu = new ShowAdminMenu();  
 adminMenu.show(stage); // Navigate back to the Admin Menu  
 });  
 cancelButton.setStyle("-fx-background-color: #d88f1e;");  
 topMenuBar.getChildren().add(cancelButton);  
  
 // Create the root layout (VBox or whatever layout you are using)  
 VBox root1 = new VBox(20);  
  
 // Add the topMenuBar at the top of the root layout  
 root.getChildren().add(topMenuBar);  
  
 Label titleLabel = new Label("Customers Database");  
 titleLabel.setStyle("-fx-font-size: 24px; -fx-font-weight: bold; -fx-text-fill: #88570b; -fx-padding: 10px 0;");  
  
 // Create the table and its columns  
 table = new TableView<>();  
 customerList = FXCollections.*observableArrayList*();  
  
 // Add columns for username and password  
 TableColumn<Customers, String> usernameColumn = new TableColumn<>("Username");  
 usernameColumn.setCellValueFactory(cellData -> cellData.getValue().getUsername());  
  
 TableColumn<Customers, String> passwordColumn = new TableColumn<>("Password");  
 passwordColumn.setCellValueFactory(cellData -> cellData.getValue().getPassword());  
  
 // Add columns to table  
 table.getColumns().add(usernameColumn);  
 table.getColumns().add(passwordColumn);  
  
 // Set table styling  
 table.setStyle("-fx-font-size: 14px; -fx-border-color: #e0e0e0; -fx-background-color: #fafafa;");  
 table.setColumnResizePolicy(TableView.*CONSTRAINED\_RESIZE\_POLICY*);  
 table.setItems(customerList); // Set the data to be displayed  
  
 // Set column styling  
 usernameColumn.setStyle("-fx-font-weight: bold; -fx-background-color: #f0f0f0;");  
 passwordColumn.setStyle("-fx-font-weight: bold; -fx-background-color: #f0f0f0;");  
  
 table.getColumns().forEach(column -> {  
 column.setStyle("-fx-font-size: 16px; -fx-font-weight: bold; -fx-background-color: #c4c4c4; -fx-text-fill: #333;");  
 });  
  
 // Set row styling  
 table.setRowFactory(tv -> {  
 TableRow<Customers> row = new TableRow<>();  
 row.setStyle("-fx-background-color: #d88f1e;");  
 row.hoverProperty().addListener((obs, wasHovered, isNowHovered) -> {  
 if (isNowHovered) {  
 row.setStyle("-fx-background-color:#d88f1e;");  
 } else {  
 row.setStyle("-fx-background-color: #ffffff;");  
 }  
 });  
 return row;  
 });  
  
 // Load customer data from the file  
 loadCustomerData();  
  
 // Add components to the layout  
 root.getChildren().addAll(titleLabel, table);  
  
 // Create a Scene and set it on the stage  
 Scene scene = new Scene(root, 1550, 670);  
 scene.setFill(Color.*WHITE*); // Set scene background color  
 stage.setTitle("Customers Database");  
 stage.setScene(scene);  
 stage.setFullScreenExitHint("");  
 stage.setFullScreenExitKeyCombination(null);  
 stage.setFullScreen(true);  
 stage.show();  
 }  
  
 private MenuBar createMenuBar(Stage stage) {  
 // Create a MenuBar  
 MenuBar menuBar = new MenuBar();  
  
 // Create the "Cancel" MenuItem and set its action  
 MenuItem cancelItem = new MenuItem("Cancel");  
 cancelItem.setOnAction(e -> {  
 // Action for returning to Admin Menu  
 ShowAdminMenu adminMenu = new ShowAdminMenu();  
 adminMenu.show(stage); // Show Admin Menu  
 });  
  
 // Create a Menu and add the "Cancel" item to it  
 Menu cancelMenu = new Menu();  
 cancelMenu.getItems().add(cancelItem);  
  
 // Add the cancelMenu to the MenuBar  
 menuBar.getMenus().add(cancelMenu);  
  
 // Position the Cancel menu to the left  
 cancelMenu.setStyle("-fx-padding: 0 20px 0 20px;");  
  
 return menuBar;  
 }  
  
 private void loadCustomerData() {  
 File file = new File("users.txt");  
 if (file.exists()) {  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 // Assuming each line contains "username,password" format  
 String[] userData = line.split(",");  
 if (userData.length == 2) {  
 customerList.add(new Customers(userData[0].trim(), userData[1].trim()));  
 }  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 } else {  
 System.*err*.println("users.txt file not found!");  
 }  
 }  
}

package com.example.projectwithgui;  
  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.Button;  
import javafx.scene.layout.VBox;  
import javafx.scene.text.Font;  
import javafx.scene.effect.DropShadow;  
import javafx.stage.Stage;  
  
import java.util.List;  
  
public class Categories {  
  
 public void show(Stage stage) {  
 // Retrieve categories from CategoryManager  
 List<String> categoriesList = CategoryManager.*getCategories*();  
  
 VBox vBox = new VBox(20);  
 vBox.setAlignment(Pos.*CENTER*);  
  
 // Title label for categories  
 Button title = new Button("Available Categories");  
 title.setFont(new Font("Arial", 24));  
 title.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 title.setEffect(new DropShadow());  
 vBox.getChildren().add(title);  
  
 // Add buttons for each category  
 for (String category : categoriesList) {  
 Button categoryButton = new Button(category);  
 styleButton(categoryButton);  
 vBox.getChildren().add(categoryButton);  
 }  
  
 // Back button  
 Button backButton = new Button("Back to Admin Menu");  
 styleButton(backButton);  
 backButton.setOnAction(e -> {  
 ShowAdminMenu adminMenu = new ShowAdminMenu();  
 adminMenu.show(stage);  
 });  
 vBox.getChildren().add(backButton);  
  
 Scene scene = new Scene(vBox, 1550, 670);  
 stage.setScene(scene);  
 vBox.setStyle("-fx-background-color: #fdd880");  
 stage.setTitle("Categories");  
 stage.setFullScreen(true);  
 stage.setFullScreenExitHint("");  
 stage.setFullScreenExitKeyCombination(null);  
 stage.show();  
 }  
  
 // Method to style buttons  
 private void styleButton(Button button) {  
 button.setFont(new javafx.scene.text.Font("Didot", 16));  
 button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; -fx-padding: 10 20; "  
 + "-fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;");  
 button.setEffect(new DropShadow());  
 button.setOnMouseEntered(e -> button.setStyle("-fx-background-color: #8c5d15; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #d88f1e; -fx-border-width: 2;"));  
 button.setOnMouseExited(e -> button.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: white; "  
 + "-fx-padding: 10 20; -fx-background-radius: 5; -fx-border-radius: 5; -fx-border-color: #8c5d15; -fx-border-width: 2;"));  
 }  
  
}

Now the user interface after user login is :

package com.example.projectwithgui;  
  
import javafx.geometry.Insets;  
import javafx.geometry.Pos;  
import javafx.scene.Node;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.image.Image;  
import javafx.scene.image.ImageView;  
import javafx.scene.layout.\*;  
import javafx.stage.Stage;  
  
import java.io.\*;  
import java.util.List;  
  
public class UserInterface {  
 //this class has three functianltites 1 user interface 2 cart 3 checkout  
 private VBox productDisplayArea;  
 private ComboBox<String> categoryFilter;  
 private Cart cart; // Cart instance  
  
 public UserInterface() {  
 cart = new Cart(); // Initialize cart  
 }  
  
 public void show(Stage primaryStage) {  
 // Top Menu Bar  
 HBox topMenuBar = new HBox(10);  
 topMenuBar.setPadding(new Insets(10));  
 topMenuBar.setStyle("-fx-background-color: #ffc784; -fx-border-color: #fff784; -fx-border-width: 1;");  
 topMenuBar.setAlignment(Pos.*CENTER\_RIGHT*);  
  
 // Category Filter ComboBox  
 categoryFilter = new ComboBox<>();  
 categoryFilter.getItems().addAll("View All Categories");  
 categoryFilter.setStyle("-fx-background-color: #d88f1e");  
 categoryFilter.getItems().addAll(CategoryManager.*getCategories*());  
 categoryFilter.setValue("View All Categories");  
 categoryFilter.setOnAction(e -> {  
 String selectedCategory = categoryFilter.getValue();  
 if ("View All Categories".equals(selectedCategory)) {  
 loadAllProducts();  
 } else {  
 filterProductsByCategory(selectedCategory);  
 }  
 });  
  
 // Cart and Logout Buttons  
 Button cartButton = new Button("Cart");  
 cartButton.setStyle("-fx-background-color: #d88f1e;");  
 cartButton.setOnAction(e -> showCart());  
 Button logout = new Button("Logout");  
 logout.setStyle("-fx-background-color: #d88f1e;");  
 logout.setOnAction(e->{  
 UserLoginOptions ulo = new UserLoginOptions(primaryStage);  
 ulo.show();  
 });  
 Button conatactAdmin = new Button("Contact Admin");  
 conatactAdmin.setStyle("-fx-background-color: #d88f1e;");  
 conatactAdmin.setOnAction(e->{  
 //code here under development  
 ContactAdmin ca = new ContactAdmin();  
 ca.show(primaryStage);  
 });  
 Button tools = new Button("Tools");  
 tools.setStyle("-fx-background-color: #d88f1e;");  
 tools.setOnAction(e->{  
 Tools t = new Tools();  
 t.show(primaryStage);  
 });  
// Image image = new Image("tools.png"); // Ensure the image is placed in src/resources  
// ImageView imageView = new ImageView(image);  
  
 // Set the image as the button's graphic  
 //tools.setGraphic(imageView);  
 //to create space betweem buttons  
 Region spacer = new Region();  
 HBox.*setHgrow*(spacer, Priority.*ALWAYS*);  
 topMenuBar.getChildren().addAll(new Label("Filter by Category:"), categoryFilter, spacer, tools,conatactAdmin,cartButton, logout);  
  
  
 productDisplayArea = new VBox(10);  
 productDisplayArea.setPadding(new Insets(10));  
 ScrollPane scrollPane = new ScrollPane(productDisplayArea);  
 scrollPane.setFitToWidth(true);  
  
 // Load all products initially to display without using any filter  
 loadAllProducts();  
  
 // Main Layout  
 VBox mainLayout = new VBox(10);  
 mainLayout.getChildren().addAll(topMenuBar, scrollPane);  
 // mainLayout.setStyle("-fx-background-color: #5c420d");  
  
 Scene scene = new Scene(mainLayout, 1550, 670);  
 primaryStage.setTitle("Welcome to Our Shopping Cart");  
 primaryStage.setScene(scene);  
 primaryStage.setFullScreenExitHint("");  
 primaryStage.setFullScreenExitKeyCombination(null);  
 primaryStage.setFullScreen(true);  
 primaryStage.show();  
 }  
  
 private void logout(Stage primaryStage) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Logout");  
 alert.setHeaderText(null);  
 alert.setContentText("You have successfully logged out.");  
 alert.showAndWait();  
  
 UserLoginOptions userLogin = new UserLoginOptions(primaryStage);  
 try {  
 userLogin.show(); // Pass the current stage  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 }  
  
 private void loadAllProducts() {  
 productDisplayArea.getChildren().clear();  
 List<Product> allProducts = CategoryManager.*getAllProducts*();  
 //loop all products  
 for (Product product : allProducts) {  
 productDisplayArea.getChildren().add(createProductBox(product));  
 }  
 }  
 //method to filter products by category  
 private void filterProductsByCategory(String category) {  
 productDisplayArea.getChildren().clear();  
 List<Product> products = CategoryManager.*getProductsForCategory*(category);  
 if (!products.isEmpty()) {  
 for (Product product : products) {  
 productDisplayArea.getChildren().add(createProductBox(product));  
 }  
 } else {  
 productDisplayArea.getChildren().add(new Label("No products found in this category."));  
 }  
 }  
   
 //creating product box to display each single product in hbox and then displaying it in scrollpane  
 private HBox createProductBox(Product product) {  
 HBox productBox = new HBox(10);  
 productBox.setPadding(new Insets(10));  
 productBox.setStyle("-fx-border-color: #ffffff; -fx-border-width: 1; -fx-background-color: #ffffff;");  
 productBox.setAlignment(Pos.*CENTER\_LEFT*);  
  
 ImageView productImage = new ImageView();  
 productImage.setFitWidth(130);  
 productImage.setFitHeight(130);  
  
 try {  
 String imagePath = product.getImage();  
 if (imagePath != null && !imagePath.isEmpty()) {  
 File imageFile = new File(imagePath);  
 if (imageFile.exists()) {  
 productImage.setImage(new Image("file:" + imageFile.getAbsolutePath()));//using of absolute method make it loaad from anywhere in system  
 } else {  
 productImage.setImage(new Image("placeholder.png")); //it is the image displayed when no pic is givne by amdin  
 }  
 } else {  
 productImage.setImage(new Image("placeholder.png"));  
 }  
 } catch (Exception e) {  
 productImage.setImage(new Image("placeholder.png"));  
 }  
  
 VBox productDetails = new VBox(5);  
 productDetails.setStyle("-fx-text-fill: #ffffff;");  
 Label nameLabel = new Label("Product Name: " + product.getName());  
 Label priceLabel = new Label("Price: RS\_" + product.getPrice());  
 Label quantityLabel = new Label("Quantity: " + product.getQuantity());  
 Label discountLabel = new Label("Discount: " + product.getDiscount() + "%");  
 Label descriptionLabel = new Label("Description: " + product.getDescription());  
 productDetails.getChildren().addAll(nameLabel, priceLabel, quantityLabel, discountLabel, descriptionLabel);  
  
 //for applying consistent styling to all labels  
 for (Node node : productDetails.getChildren()) {  
 if (node instanceof Label) {  
 ((Label) node).setStyle("-fx-text-fill: #000000; -fx-font-size: 14px;");  
 }  
 }  
  
 Button addToCartButton = new Button("Add to Cart");  
 addToCartButton.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: #000000; -fx-font-size: 14px;");  
 addToCartButton.setOnAction(e -> addToCart(product));  
 VBox imageAndButton = new VBox(10, productImage, addToCartButton);  
 imageAndButton.setAlignment(Pos.*CENTER*);  
  
 productBox.getChildren().addAll(imageAndButton, productDetails);  
 return productBox;  
 }  
  
  
  
 private void addToCart(Product product) {  
 cart.addProduct(product);  
  
 // little bit styling of cart class  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("🛒 Cart Update");  
 alert.setHeaderText("Success! Item Added to Cart 🎉");  
 alert.setContentText("The product \"" + product.getName() + "\" has been successfully added to your cart.\n\n" +  
 "Cart Total: " + cart.getProducts().size() + " items.");  
  
  
 ImageView cartIcon = new ImageView(new Image("cart\_icon.png"));  
 cartIcon.setFitWidth(50);  
 cartIcon.setFitHeight(50);  
 alert.setGraphic(cartIcon);  
  
  
 DialogPane dialogPane = alert.getDialogPane();  
 dialogPane.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-background-color: #f8f8f8;");  
 dialogPane.lookup(".header-panel").setStyle("-fx-background-color: #dd8e13; -fx-text-fill: white;");  
  
 alert.showAndWait();  
 }  
  
  
  
 //Here we use both borderPane and GridPan bcz first we make borderPane then inside it make Vbox in which we make Grid pane tHIS SAVE US from making a table  
 private void showCart() {  
 Stage cartStage = new Stage();  
 cartStage.setTitle("Shopping Cart");  
  
  
 BorderPane mainLayout = new BorderPane();  
 mainLayout.setPadding(new Insets(10));  
 mainLayout.setStyle("-fx-background-color: #f7f7f7;");  
  
 VBox cartItemsLayout = new VBox(10);  
 cartItemsLayout.setPadding(new Insets(10));  
 cartItemsLayout.setStyle("-fx-border-color: #e0e0e0; -fx-border-width: 2; -fx-background-color: #ffffff; -fx-background-radius: 10;");  
 Label cartHeader = new Label("Shopping Cart");  
 cartHeader.setStyle("-fx-font-size: 22px; -fx-font-weight: bold; -fx-text-fill: #333;");  
  
 GridPane cartGrid = new GridPane();  
 cartGrid.setHgap(10);  
 cartGrid.setVgap(10);  
 cartGrid.setPadding(new Insets(10));  
 cartGrid.setStyle("-fx-background-color: #ffdc97; -fx-background-radius: 5; -fx-border-color:A68A51FF; -fx-border-width: 2;");  
  
 Label productHeader = new Label("Product Details");  
 Label priceHeader = new Label("Price");  
 Label totalHeader = new Label("Total");  
  
  
 String headerStyle = "-fx-font-weight: bold; -fx-font-size: 16px; -fx-text-fill: #000000;";  
 productHeader.setStyle(headerStyle);  
 priceHeader.setStyle(headerStyle);  
 totalHeader.setStyle(headerStyle);  
  
  
 cartGrid.add(productHeader, 0, 0);  
 cartGrid.add(priceHeader, 2, 0);  
 cartGrid.add(totalHeader, 3, 0);  
  
  
 int row = 1;  
 for (Product product : cart.getCartItems()) {  
 Label productLabel = new Label(product.getName());  
 productLabel.setStyle("-fx-text-fill: #000000; -fx-font-size: 14px;");  
  
  
 Label priceLabel = new Label("Rs\_" + product.getPrice());  
 priceLabel.setStyle("-fx-text-fill: #000000;");  
  
 Label totalLabel = new Label("Rs\_" + (product.getPrice()));  
 totalLabel.setStyle("-fx-text-fill: #000000;");  
  
 Button removeButton = new Button("Remove");  
 removeButton.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: #000000; -fx-font-size: 14px;");  
 removeButton.setOnAction(e -> {  
 cart.getCartItems().remove(product);  
 showCart(); //to refresh the cart after removal or addition of new products  
 cartStage.close();  
 });  
  
 cartGrid.add(productLabel, 0, row);  
 cartGrid.add(priceLabel, 2, row);  
 cartGrid.add(totalLabel, 3, row);  
 cartGrid.add(removeButton, 4, row);  
  
 row++;  
 }  
  
 cartItemsLayout.getChildren().addAll(cartHeader, cartGrid);  
  
 VBox orderSummaryLayout = new VBox(10);  
 orderSummaryLayout.setPadding(new Insets(10));  
 orderSummaryLayout.setStyle("-fx-border-color: #a68a51; -fx-border-width: 2; -fx-background-color: #ffdc97; -fx-background-radius: 10;");  
  
 Label summaryHeader = new Label("Order Summary");  
 summaryHeader.setStyle("-fx-font-size: 20px; -fx-font-weight: bold; -fx-text-fill: #333;");  
  
  
 double totalCost = 0;  
 for (Product product : cart.getCartItems()) {  
 // totalCost += product.getPrice() \* product.getQuantity(); as still the logic for quntity is under development  
 totalCost += product.getPrice();  
 }  
  
 Label itemsLabel = new Label("Items: " + cart.getCartItems().size());  
 itemsLabel.setStyle("-fx-font-size: 14px; -fx-text-fill: #555;");  
  
 Label totalCostLabel = new Label("Total Cost: Rs\_" + totalCost);  
 totalCostLabel.setStyle("-fx-font-size: 18px; -fx-font-weight: bold; -fx-text-fill: #333;");  
  
 Button checkoutButton = new Button("Checkout");  
 checkoutButton.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: #000000; -fx-font-size: 14px; -fx-font-weight: bold;");  
 checkoutButton.setOnAction(e -> {  
 showCheckout();  
 cartStage.close();  
 });  
  
 orderSummaryLayout.getChildren().addAll(summaryHeader, itemsLabel, totalCostLabel, checkoutButton);  
  
 //layout setting  
 mainLayout.setTop(cartItemsLayout);  
 mainLayout.setBottom(orderSummaryLayout);  
  
 Scene cartScene = new Scene(mainLayout, 800, 600);  
 cartStage.setScene(cartScene);  
 cartStage.show();  
 }  
  
  
  
  
//  
// private void updateOrderSummary(VBox cartItemsLayout) {  
// // Logic to dynamically update the summary when cart changes  
// }  
  
  
 private void showCheckout() {  
 Stage checkoutStage = new Stage();  
 checkoutStage.setTitle("Checkout");  
  
 GridPane mainLayout = new GridPane();  
 mainLayout.setPadding(new Insets(10));  
 mainLayout.setHgap(20);  
 mainLayout.setVgap(20);  
 mainLayout.setStyle("-fx-background-color: #f5f5f5;");  
 //hbox for billing address  
 VBox billingSection = new VBox(10);  
 billingSection.setPadding(new Insets(15));  
 billingSection.setStyle("-fx-background-color: #ffdc97; -fx-border-color: #a68a51; -fx-border-width: 1; -fx-background-radius: 8;");  
 billingSection.setPrefWidth(400);  
 Label billingHeader = new Label("1. Billing Address");  
 billingHeader.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
 TextField firstNameField = new TextField();  
 firstNameField.setPromptText("First Name");  
 TextField lastNameField = new TextField();  
 lastNameField.setPromptText("Last Name");  
 TextField emailField = new TextField();  
 emailField.setPromptText("Email Address");  
 TextField phoneField = new TextField();  
 phoneField.setPromptText("Phone Number");  
 TextField addressField = new TextField();  
 addressField.setPromptText("Billing Address");  
 TextField cityField = new TextField();  
 cityField.setPromptText("City");  
 ComboBox<String> countryBox = new ComboBox<>();  
 countryBox.getItems().addAll("Pakistan", "USA", "UK", "India");  
 countryBox.setPromptText("Country");  
 billingSection.getChildren().addAll(billingHeader, firstNameField, lastNameField, emailField, phoneField, addressField, cityField, countryBox);  
 //section for selecting payment method  
 VBox paymentSection = new VBox(10);  
 paymentSection.setPadding(new Insets(15));  
 paymentSection.setStyle("-fx-background-color: #ffdc97; -fx-border-color: #a68a51; -fx-border-width: 1; -fx-background-radius: 8;");  
 paymentSection.setPrefWidth(400);  
 Label paymentHeader = new Label("2. Payment Method");  
 paymentHeader.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
 RadioButton payOnlineOption = new RadioButton("Pay Online");  
 RadioButton cashOnDeliveryOption = new RadioButton("Cash on Delivery");  
 ToggleGroup paymentGroup = new ToggleGroup();  
 payOnlineOption.setToggleGroup(paymentGroup);  
 cashOnDeliveryOption.setToggleGroup(paymentGroup);  
 cashOnDeliveryOption.setSelected(true);  
 //make the functionality of online payment radiobutton  
 payOnlineOption.setOnAction(e -> {  
 if (payOnlineOption.isSelected()) {  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Online Payment");  
 alert.setHeaderText(null);  
 alert.setContentText("This part is under development. Please choose another payment method.");  
 alert.showAndWait();  
 cashOnDeliveryOption.setSelected(true); // Automatically select Cash on Delivery as Pay Online is under development  
 }  
 });  
  
 paymentSection.getChildren().addAll(paymentHeader, payOnlineOption, cashOnDeliveryOption);  
  
 HBox sectionsLayout = new HBox(20);  
 sectionsLayout.getChildren().addAll(billingSection, paymentSection);  
 //third section for order review before payment  
 HBox reviewSection = new HBox(20);  
 reviewSection.setPadding(new Insets(15));  
 reviewSection.setStyle("-fx-background-color: #ffdc97; -fx-border-color: #a68a51; -fx-border-width: 1; -fx-background-radius: 8;");  
 reviewSection.setPrefWidth(800);  
 Label reviewHeader = new Label("3. Review Your Order");  
 reviewHeader.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");  
 //Label orderSummaryLabel = new Label("Order Summary:");  
 double totalBill = cart.calculateTotal();  
 Label subtotalLabel = new Label("Subtotal: Rs\_" + totalBill);  
 Label shippingCostLabel = new Label("Shipping: Rs\_500");  
 Label totalCostLabel = new Label("Grand Total: Rs\_" + (totalBill + 500));  
 Button placeOrderButton = new Button("Place Order");  
 placeOrderButton.setStyle("-fx-background-color: #d88f1e; -fx-text-fill: #000000; -fx-font-size: 14px; -fx-font-weight: bold;");  
 placeOrderButton.setOnAction(e -> {  
 //collecting checkout data to store it in file  
 String firstName = firstNameField.getText();  
 String lastName = lastNameField.getText();  
 String email = emailField.getText();  
 String phone = phoneField.getText();  
 String address = addressField.getText();  
 String city = cityField.getText();  
 String country = countryBox.getValue();  
 String paymentMethod = cashOnDeliveryOption.isSelected() ? "Cash on Delivery" : "Pay Online";  
 double grandTotal = totalBill + 500; //default shipping fee  
  
 // storing the data in text file  
 try (PrintWriter writer = new PrintWriter(new FileWriter("orderDetails.txt", true))) {  
 writer.println("----- Order Details -----");  
 writer.println("Customer Name: " + firstName + " " + lastName);  
 writer.println("Email: " + email);  
 writer.println("Phone: " + phone);  
 writer.println("Address: " + address);  
 writer.println("City: " + city);  
 writer.println("Country: " + country);  
 writer.println("Payment Method: " + paymentMethod);  
 writer.println("Subtotal: Rs\_" + totalBill);  
 writer.println("Shipping: Rs\_500");  
 writer.println("Grand Total: Rs\_" + grandTotal);  
 writer.println("\n");  
 //storing revenue in seperate file to show admin when needed  
 try (PrintWriter revenueWriter = new PrintWriter(new FileWriter("revenue.txt", true))) {  
 revenueWriter.println("Revenue from Order: Rs\_" + grandTotal);  
 revenueWriter.println("Total Revenue So Far: Rs\_" + getTotalRevenue());  
 revenueWriter.println("\n");  
 }  
  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
  
 //alert for successful order  
 Alert alert = new Alert(Alert.AlertType.*INFORMATION*);  
 alert.setTitle("Order Placed");  
 alert.setHeaderText(null);  
 alert.setContentText("Thank you! Your order has been placed successfully.");  
 alert.showAndWait();  
  
 cart.clearCart();  
 checkoutStage.close();  
 });  
  
 reviewSection.getChildren().addAll(reviewHeader, subtotalLabel, shippingCostLabel, totalCostLabel, placeOrderButton);  
 mainLayout.add(sectionsLayout, 0, 0);  
 mainLayout.add(reviewSection, 0, 1);  
  
 Scene checkoutScene = new Scene(mainLayout, 900, 600);  
 checkoutStage.setScene(checkoutScene);  
 checkoutStage.show();  
 }  
  
 // A method to read the total revenue from the file (for calculating total revenue so far)  
 private double getTotalRevenue() {  
 double totalRevenue = 0.0;  
  
 try (BufferedReader reader = new BufferedReader(new FileReader("revenue.txt"))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 if (line.startsWith("Revenue from Order")) {  
 String revenueStr = line.split(":")[1].trim();  
 revenueStr = revenueStr.replace("Rs\_", "").trim();  
 try {  
 totalRevenue += Double.*parseDouble*(revenueStr);  
 } catch (NumberFormatException e) {  
 System.*out*.println("Error parsing revenue: " + revenueStr);  
 }  
 }  
 }  
 } catch (IOException ex) {  
 ex.printStackTrace();  
 }  
 return totalRevenue;  
 }  
  
}

The most important class here is category manager that makes files and store data and also make it able to read or append

package com.example.projectwithgui;  
  
import java.io.\*;  
import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
//most important class consists of all method to take products from files etc  
public class CategoryManager {  
 private static final List<String> *categories* = new ArrayList<>();  
 private static final String *CATEGORY\_FILE* = "categories.txt";  
 //load all existing categories  
 static {  
 *loadCategories*();  
 }  
  
 public static void addCategory(String categoryName) {  
 if (!*categories*.contains(categoryName)) {  
 *categories*.add(categoryName);  
 *saveCategories*();  
 *createCategoryFile*(categoryName);//for creating file for every category  
 } else {  
 System.*out*.println("Category already exists!");  
 }  
 }  
  
 public static List<String> getCategories() {  
 return *categories*;  
 }  
  
 public static List<Product> getProductsForCategory(String category) {  
 List<Product> products = new ArrayList<>();  
 String fileName = category.replaceAll("\\s+", "\_") + ".txt";  
  
 try (BufferedReader reader = new BufferedReader(new FileReader(fileName))) {  
 String line;  
  
 while ((line = reader.readLine()) != null) {  
 String[] fields = line.split("\\|");  
 String name = "", description = "", image = "";  
 double price = 0.0;  
 int quantity = 0;  
 double discount = 0.0;  
  
 for (String field : fields) {  
 String[] keyValue = field.split(": ", 2);  
 if (keyValue.length < 2) continue;  
  
 String key = keyValue[0].trim();  
 String value = keyValue[1].trim();  
  
 switch (key) {  
 case "Product Name":  
 name = value;  
 break;  
 case "Price":  
 price = Double.*parseDouble*(value);  
 break;  
 case "Quantity":  
 quantity = Integer.*parseInt*(value);  
 break;  
 case "Discount":  
 discount = Double.*parseDouble*(value.replace("%", ""));  
 break;  
 case "Description":  
 description = value;  
 break;  
 case "Image":  
 image = value.replace("Image Selected: ", "").trim();  
 break;  
 }  
 }  
  
 if (!name.isEmpty()) {  
 products.add(new Product(name, price, quantity, discount, description, image, category));  
 }  
 }  
 } catch (IOException e) {  
 System.*out*.println("Error reading file for category " + category + ": " + e.getMessage());  
 }  
  
 return products;  
 }  
  
  
 // Save categories to a file  
 private static void saveCategories() {  
 try (BufferedWriter writer = new BufferedWriter(new FileWriter(*CATEGORY\_FILE*))) {  
 for (String category : *categories*) {  
 writer.write(category);  
 writer.newLine();  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error saving categories: " + e.getMessage());  
 }  
 }  
  
 // Load categories from a file  
 private static void loadCategories() {  
 File file = new File(*CATEGORY\_FILE*);  
 if (file.exists()) {  
 try (BufferedReader reader = new BufferedReader(new FileReader(file))) {  
 String line;  
 while ((line = reader.readLine()) != null) {  
 *categories*.add(line);  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error loading categories: " + e.getMessage());  
 }  
 }  
 }  
 public static List<Product> getAllProducts() {  
 // Combine all products from all categories into a single list to display as userinterface starts  
 List<Product> allProducts = new ArrayList<>();  
 for (String category : *getCategories*()) {  
 allProducts.addAll(*getProductsForCategory*(category));  
 }  
 return allProducts;  
 }  
  
  
 // Create a file for each category to display their products  
 private static void createCategoryFile(String categoryName) {  
 String fileName = categoryName.replaceAll("\\s+", "\_") + ".txt";  
 File file = new File(fileName);  
 if (!file.exists()) {  
 try {  
 if (file.createNewFile()) {  
 System.*out*.println("File created for category: " + file.getName());  
 }  
 } catch (IOException e) {  
 System.*err*.println("Error creating category file: " + e.getMessage());  
 }  
 }  
 }  
  
  
}