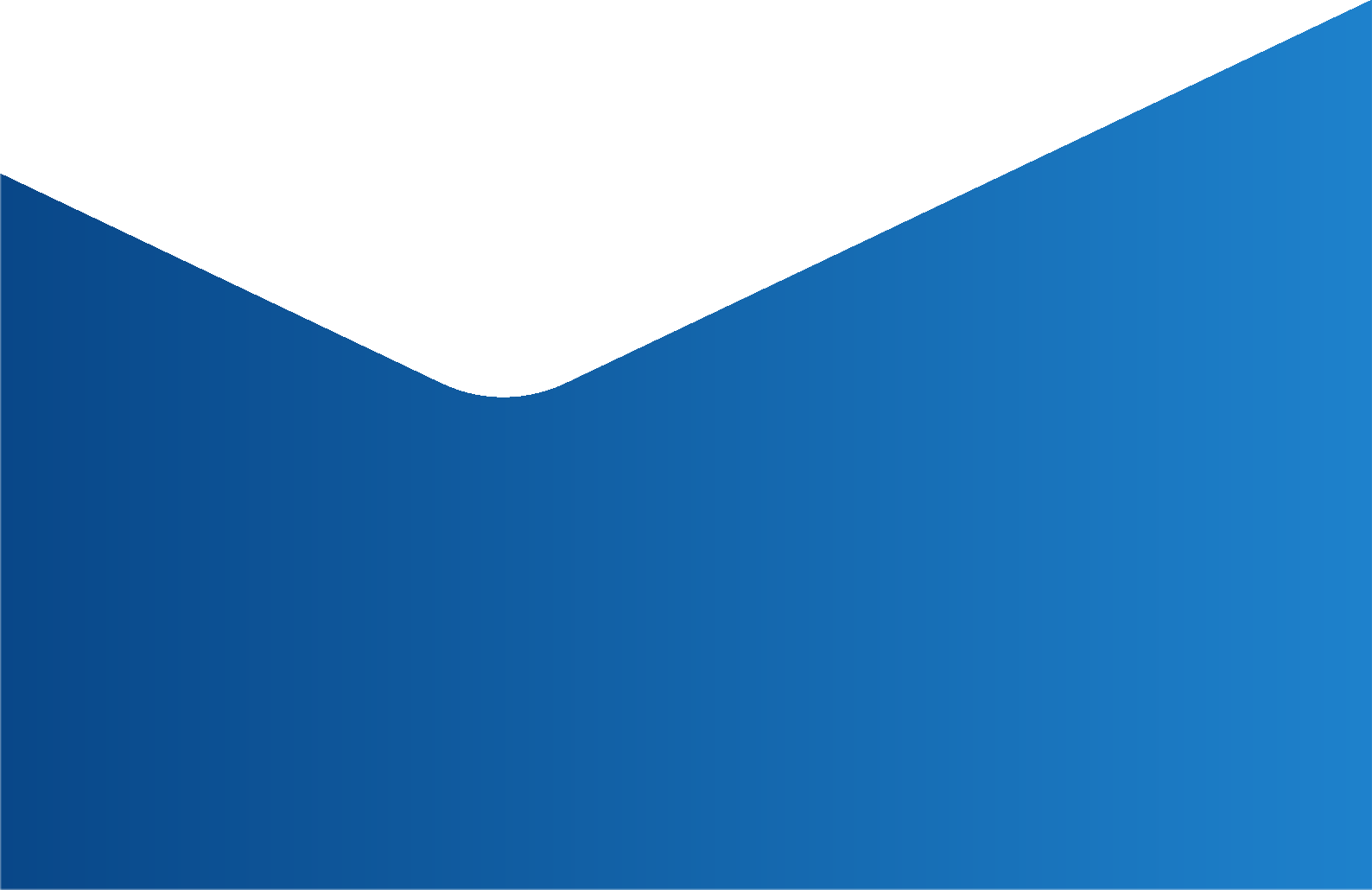
 AIR UNIVERSITY MULTAN CAMPUS



**Project0**

Shopping Cart

SUBMITTED TO:

Mam-Aatka Ali

SUBMITTED BY:

|  |  |
| --- | --- |
| Muhammad Ali | - 233510 |
| Muhammad Hassan | - 233516 |
| Muhammad Haroon | - 233504 |

**Introduction :**

The Shopping Cart System is an advanced console-based application crafted to elevate the online shopping experience for consumers and retailers alike. In the fast-growing world of e-commerce, a reliable shopping cart system is crucial to support smooth and efficient transactions. This application acts as a connection point between users and retailers, simplifying the process of browsing products, managing chosen items, and finalizing purchases.

**Key Features :**

The Shopping Cart System offers a range of essential features designed to improve the shopping experience for users and supply retailers with vital tools. The main features of the system include:

## Inventory Management :

* Real-time inventory tracking allows retailers to monitor stock levels instantly as sales occur.
* Provides users with up-to-date product availability information.
* Automatically updates item status when sold out, reducing customer frustration.
* Enhances customer satisfaction by preventing issues with unavailable items.
* Helps retailers effectively manage inventory, reducing risks of overstocking or stockouts.

## Product Categories :

* + The system supports efficient product categorization, allowing users to browse categories seamlessly.
  + This organized structure helps users locate items of interest quickly, without navigating unrelated products.
  + Categorizing products enhances navigation and contributes to a more user-friendly shopping experience.
  + Retailers can easily manage and update listings, ensuring customers always see the latest offerings.

## User Interaction :

* + Users interact with the Shopping Cart System through a console application, providing a straightforward and engaging experience.
  + The interface is designed to be intuitive, making navigation easy for users.
  + It guides users through key functionalities, including viewing products, managing their shopping cart, and completing the checkout process.

## Viewing Products :

* + Upon launch, users receive a welcome message and a main menu.
  + They can choose to view available products, prompting a catalog display.
  + Each item is listed with its name, price, and category for easy browsing.
  + The structured layout helps users explore the catalog and identify items of interest.
  + This organized presentation simplifies navigation and selection of products.

## Managing Cart :

* + Users add products to their cart by entering the product ID and specifying the quantity.
  + The system prompts for confirmation, giving users control over their selections.
  + Users can view their cart anytime, seeing a summary with item subtotals, discounts, sales tax, and the total amount.
  + The cart allows flexibility: users can remove items or adjust quantities as needed.
  + This feature simplifies cart management, making it easy for users to customize their shopping experience.

## Discount Mechanisms :

* + A built-in discount mechanism automatically applies discounts when predefined thresholds are met.
  + This feature encourages larger purchases while enhancing customer satisfaction through savings.
  + Retailers can set custom discount rules for promotions or seasonal sales, enabling flexible marketing strategies.
  + By managing discounts effectively, the system helps boost sales and adds value for customers.
  + Together, these features contribute to a smooth and enjoyable shopping experience, making e-commerce straightforward and beneficial for both users and retailers.

## Payment Methods :

* + The Shopping Cart System integrates with multiple payment gateways to ensure secure transactions.
  + Customers can choose from various payment methods, including credit cards, digital wallets, and other online options.
  + Offering multiple payment options caters to diverse customer preferences, encouraging more completed purchases.
  + Secure transaction processing builds user trust, allowing customers to shop confidently with their payment information protected.

## Checking Out :

* + Users can proceed to checkout via the menu when ready to complete their purchase.
  + The system displays a detailed order summary, allowing users to review items before finalizing.
  + Total cost is calculated, with automatic application of discounts and sales tax.
  + Users confirm their purchase, and payment is processed through the selected gateway.
  + Upon successful payment, users receive an order confirmation, completing the transaction.
  + This intuitive process ensures a smooth, user-friendly experience, boosting satisfaction and fostering repeat visits.

**Project Features :**

The Shopping Cart System is equipped with a range of essential features designed to enhance user experience and streamline e-commerce operations. Below are the key functionalities that define the system:

* Add and Remove Products :
  + Users can easily add products to their shopping cart by specifying the product ID and quantity.
  + This straightforward process enables quick selection, allowing users to build their cart efficiently.
  + Users can remove items from the cart at any time, with instant updates reflecting the current selections for an organized shopping experience.
* Item Quantity Management :
  + The system supports item quantity management, letting users specify the number of units for each product.
  + Particularly useful for purchasing multiple items of the same type, this feature recalculates the total cost in real-time as users adjust quantities.
  + Users receive accurate information about their potential expenditure, facilitating informed purchasing decisions.
* Apply Discounts :
  + To enhance customer satisfaction, the Shopping Cart System includes an automated discount mechanism.
  + Discounts are applied based on predefined purchase thresholds or promotional events, encouraging larger purchases.
  + This functionality benefits customers through savings while aiding retailers in driving sales during promotional periods.
* Product Recommendations :
  + The system features a recommendation engine that suggests additional products based on items already in the user’s cart.
  + By analyzing selected categories and product types, the system proposes complementary items of interest.
  + This personalized suggestion feature enhances the shopping experience and increases sales opportunities for retailers by encouraging exploration of more products.
* Cart Expiration Management :
  + To enhance security and manage user sessions, the system includes a cart expiration feature.
  + If a cart remains inactive for 30 minutes, it expires, prompting users to restart their shopping session.
  + This function protects user data and ensures inventory accuracy by clearing abandoned carts that could lead to confusion over product availability.

These project features work together to create a comprehensive shopping experience that meets the needs of both consumers and retailers, fostering an environment conducive to successful online transactions.

**Overview :**

The Shopping Cart System is designed to provide users with a seamless and intuitive workflow from the moment they enter the application until they complete their purchase. Below is a detailed description of the typical steps a user would follow during their shopping journey.

* Step 1: Browsing Products :
  + Upon launching the application, users are greeted with a welcome message and presented with a menu of options.
  + Users can select the option to view available products, prompting the system to display a categorized list of items, including their names, prices, and categories.
  + This organized layout allows users to navigate easily through various product offerings and identify items of interest.
* Step 2: Adding Products to the Cart :
  + Once users find products they wish to purchase, they can add them to their shopping cart by entering the product ID and specifying the desired quantity.
  + The system confirms their selections, ensuring users have control over their choices.
  + Users can add multiple items at once, allowing for efficient cart management as they shop.
* Step 3: Managing the Shopping Cart :
  + After adding products, users can view their shopping cart at any time.
  + This feature displays a summary of all items currently in the cart, including subtotals, applied discounts, sales tax, and the total amount due.
  + Users can adjust quantities or remove items as needed, providing flexibility to refine their selections before checkout.
* Step 4: Proceeding to Checkout :
  + When users are satisfied with their cart contents, they can proceed to the checkout process.
  + Selecting the checkout option presents a detailed summary of their order for final review.
  + The system automatically calculates the total amount due, applying any applicable discounts and sales tax.
  + Once satisfied with the summary, users confirm their purchase.
* Step 5: Completing the Purchase :
  + Upon confirmation, the system processes the payment through the selected payment gateway.
  + After successful payment, users receive an order confirmation, completing the transaction.
  + This structured workflow enhances the shopping experience and encourages users to return for future purchases, as it is designed to be both efficient and user-friendly.

**Code Structure :**

The Shopping Cart System is built on several key classes that work together to create a cohesive and functional application. The primary classes involved are \*\***Product**\*\*, \*\***CartItem**\*\*, \*\***ShoppingCart**\*\*, and \*\***Program**\*\*. Each class plays a specific role in the system, contributing to its overall architecture and functionality.

* **Product Class :**

The **Product** class serves as a blueprint for individual items available for purchase in the shopping cart system.

Properties:

* **Id**: A unique identifier for each product.
* **Name**: The human-readable title of the product.
* **Price**: The cost of the product, crucial for transaction calculations.
* **Category**: Enables effective organization of products.
* **Constructor**: Initializes the properties.
* **ToString()** **Method**: Returns a string representation of the product for easy display within the application.
* **CartItem Class :**

- The **CartItem** class represents a product that has been added to the shopping cart.

Properties:

* **Product**: An instance of the \*\*Product\*\* class.
* **Quantity**: Specifies how many units of the product are in the cart.
* **Constructor**: Initializes the properties.
* **ToString() Method**: Provides a string representation of the cart item.
* **GetTotalPrice()** **Method**: Calculates the total cost of the cart item based on its quantity, ensuring accurate pricing in the shopping cart.
* **ShoppingCart Class :**

The **ShoppingCart** class is central to the application, managing a collection of **CartItem** objects.

Properties:

* **Items**: A list of **CartItem** objects.
* **SalesTax**: A variable to handle sales tax calculations.
* **CartExpiration**: Tracks the validity of the cart.

Key Methods**:**

* **AddProduct()**: Adds a product to the cart or updates its quantity if it already exists.
* **RemoveProduct():** Removes a specified quantity of a product from the cart.
* **GetSubtotal()**: Calculates the subtotal of items in the cart.
* **GetDiscountAmount():** Computes any applicable discounts.
* **GetTotal():** Calculates the overall total, including tax and discounts.
* **Program Class :**
* The **Program** class serves as the entry point for the application.
* **Main** **Method**: Initializes instances of both **ShoppingCart** and **Product**.
* **User** **Interaction**: Orchestrates interactions with users, displaying a menu that allows them to view products, manage their cart, and proceed to checkout.
* **Helper** **Methods**: Includes methods such as **ViewProducts()** and **AddToCart(** to facilitate user actions, ensuring the application runs smoothly and effectively responds to user inputs.

**ScreenShots :**

