WELCOME

Welcome to our C# shopping cart solution!

Group Members

Our team consists of:

- Sadia Khan (233544)
- Somal Baloch (233530)
- Danish Butt (233606)

Shopping Cart: A Visual Programming Programming C# Solution

This C# solution provides a comprehensive shopping cart experience, allowing users to seamlessly add, remove, and manage products, while also calculating the total cost, applying discounts and sales tax, and offering personalized product recommendations.

Overview of the Shopping Cart Cart System

1. **Product Management**

   Users can easily add or remove products from the cart.

2. **Cart Visualization**

   The cart's contents are displayed in a clear and organized manner.

3. **Cost Calculation**

   The system automatically calculates the total cost, considering discounts and sales tax.

Libraries used in the System

1

```
using System;
```

Purpose: This is the core namespace that provides fundamental classes and base classes for commonly used data types, events, and exceptions.

2

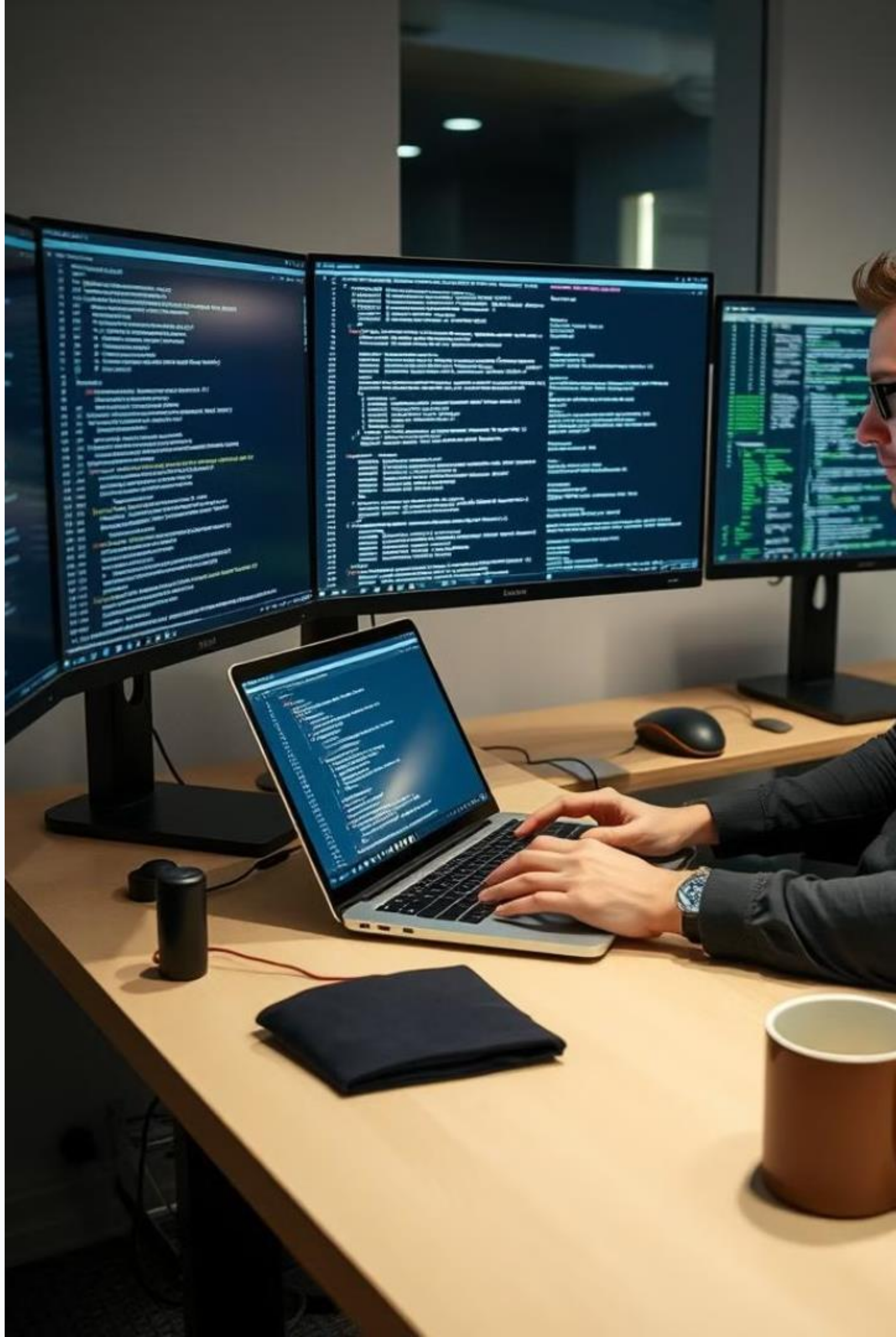
```
using System.Collections.Generic;
```

Purpose: This namespace contains interfaces and classes that define generic collections, which allow you to create strongly typed collections.

3

```
using System.Linq;
```

Purpose: LINQ (Language Integrated Query) provides a way to query collections in a more readable and expressive manner using syntax similar to SQL.



Classes in c# code

1 class Product

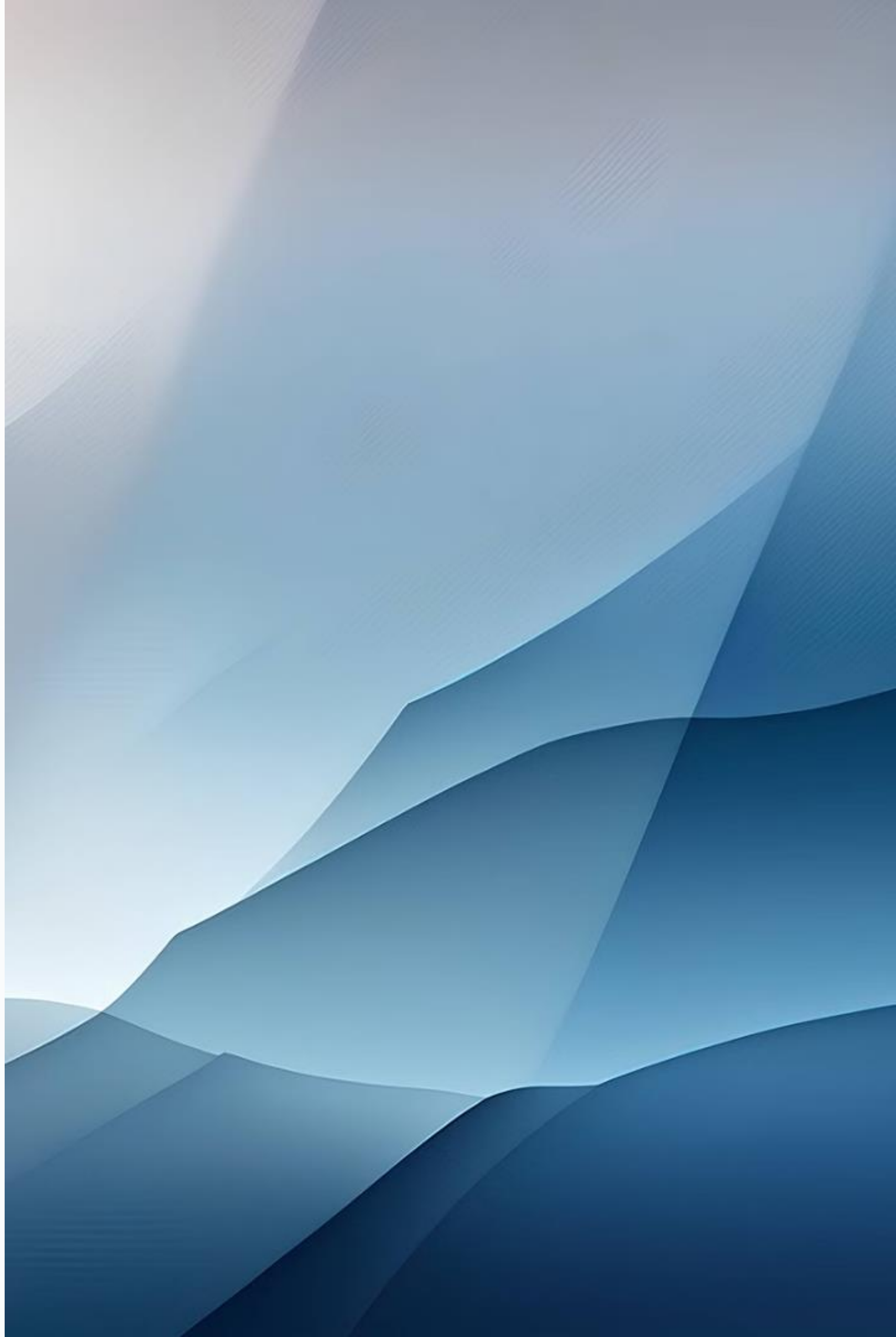
This class contains details about the products.

2 class Cart

This class consists of methods to add,remove products in cart.

3 class Program

It initializes a console-based shopping cart application, displaying a welcome message and setting up a list of available products for users to interact with.



Functions :

Two types of functions are used in the Shopping Cart System :

- **User-defined function :** Custom functions created by the developer to handle specific tasks.
- **Buillt-in function :** Predefined functions provided by the programming language that facilitate common operations.

Built-in Functions :

- `Console.Clear()` : Clears the console window.
- `Console.Title` : Sets the title of the console window.
- `Console.ForegroundColor` : Changes the color of the text output in the console.
- `Environment.Exit(0)` : Terminates the current process and returns an exit code to the operating system.
- `Console.WriteLine()`: Outputs a line of text to the console.
- `Console.ReadLine()`: Reads a line of input from the console.

Built-in Functions :

- `IsExpired()` : Checks if the cart has expired based on the current date and time.
- `C$"{product.Name}"` : Allows for easier and more readable string formatting.
- `int.TryParse()` : Attempts to convert a string representation of a number to its integer equivalent.
- `DateTime.Now` : Retrieves the current date and time.
- `TimeSpan` : Represents a time interval.
- `Console.ResetColor()`: Resets the console text color to its default, ensuring that subsequent text appears in the standard color.

User-defined functions:

- `static void DisplayWelcomeTitle()`: Displays a welcome message and prompts the user for their name.
- `static void ChooseMobilePhone()`:Displays mobile phone brands for the user to select from.
- `static void PauseScreen()`: Presumably pauses the screen until the user presses a key.
- `static int GetValidIndex(int maxOption, string prompt)`: Presumably prompts the user for an index and validates it against the maximum option.
- `static string GetValidInput(string prompt, string[] validOptions)`: Presumably prompts the user for input and validates it.

User-defined functions:

- `static void Header()`: Presumably displays a header for the application
- `DisplayWelcomeTitle()`: This function displays a welcome message and prompts the user for their `ProductRecommendation()`: This function likely provides product recommendations to the user.
- `AddProductToCart()`: Displays product categories and allows the user to add product to cart.
- `public void ViewCartWithIndices()`: Displays the items in the cart along with their indices, allowing users to see what they have added.
- `public int ItemsCount()`: Returns the number of items currently in the cart.

User-defined functions:

- `RemoveProductByIndex()`: Removes a product from the cart based on the specified index
- `RemoveProductByIndex()`: Removes a product from the cart based on the specified.
- index`ProductRecommendation()`: This function likely provides product recommendations to the user.
- `AddProductToCart()`: Displays product categories and allows the user to add product to cart.
- `ItemsCount()`: Returns the number of items currently in the cart.

Adding and Removing Products

Adding Products Products

Users can search for and select products to add to their cart with a single click.

Removing Products

If a user changes their mind, they can easily remove items from the cart.

Quantity Management

Users can adjust the quantity of a product in the cart to meet their needs.

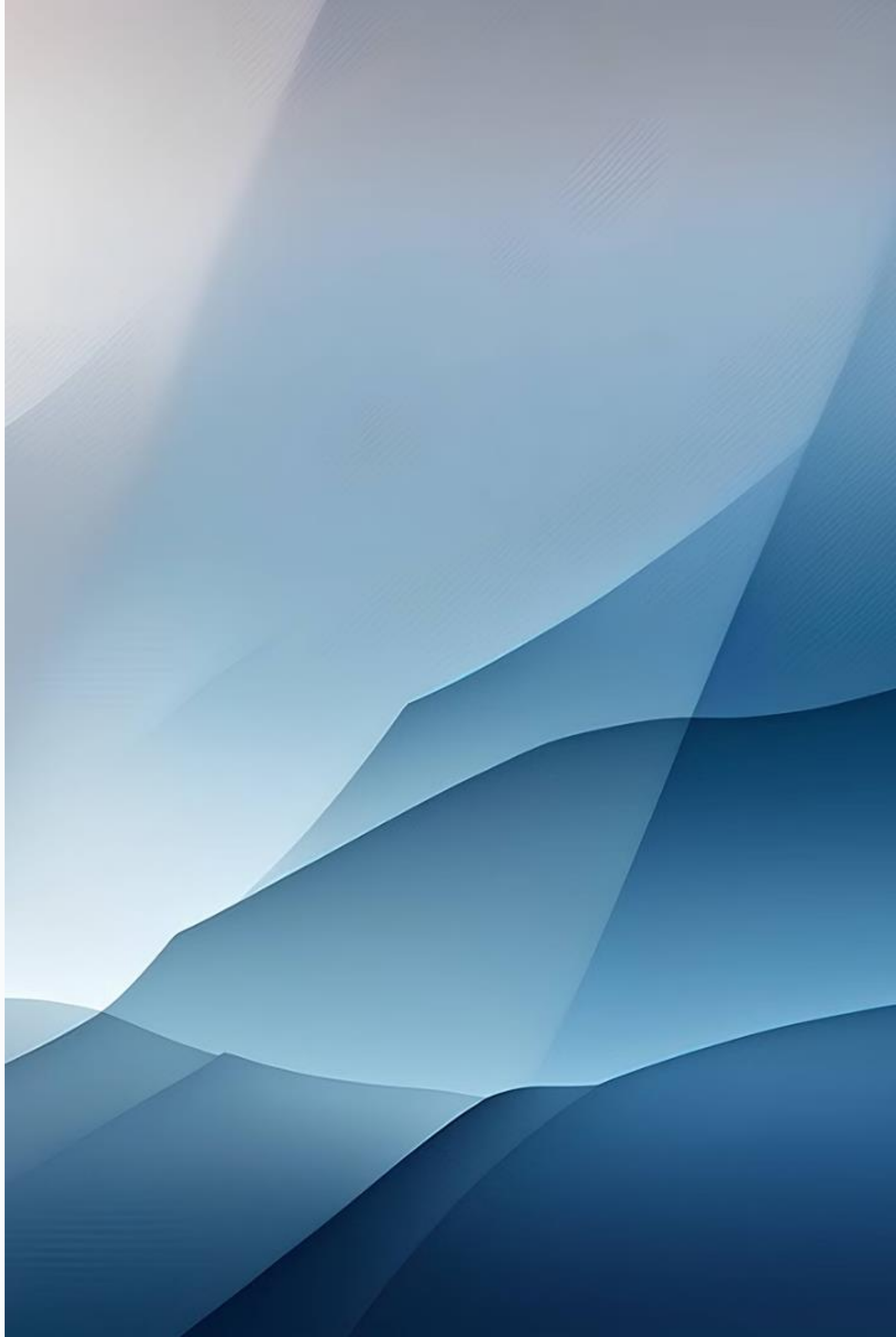
Viewing and Managing the the Cart

1 **Cart Overview**

Users can view a list of all items in their cart.

2 **Sorting products**

Users can also see the quantities and prices.



Calculating the Total Cost

Subtotal	$150.00
Discounts	-$20.00
Sales Tax	$10.50
Total	$140.50

Applying Discounts and Sales Sales Tax

Discounts

The system can apply various discounts, such as promotional codes, loyalty program rewards, or bulk pricing.

Sales Tax

The shopping cart automatically calculates and applies the appropriate sales tax based on the user's location.

Product Recommendations

Similar Products

The system suggests items that are similar to the ones already in the user's cart.

Featured Products

Users are presented with a diverse array of popular items.

Complementary Products

The system recommends items that may complement the user's current cart selections.

Cart Expiration and Checkout Checkout Process

1

Cart Expiration

Carts are set to expire after a certain period of inactivity to ensure data is up-to-date.

2

Checkout

Users can seamlessly proceed to checkout, where they can review their order and complete the purchase.

Conclusion and Key Takeaways Takeaways

The C# shopping cart solution provides a comprehensive and user-friendly experience, empowering customers to manage their purchases with ease. Key features include seamless product management, transparent cost calculations, personalized recommendations, and a streamlined checkout process.

Thank You!