ASSIGNMENT#01

using System;

using System.Collections.Generic;

namespace Assignment1

{

internal class Program

{

class Product

{

public readonly string Title;

public readonly int Cost;

public int QuantityInCart;

public int AvailableStock;

public Product(string title, int cost, int stock)

{

Title = title;

Cost = cost;

AvailableStock = stock;

QuantityInCart = 0;

}

}

static readonly List<Product> shoppingBag = new List<Product>();

static readonly Dictionary<string, int> salesTracker = new Dictionary<string, int>();

static int totalRevenue = 0;

static void Main()

{

var inventory = new Dictionary<string, Product>

{

{"Men Cotton", new Product("Men Cotton", 4000, 10)},

{"Men Silk", new Product("Men Silk", 4500, 10)},

{"Men Wool", new Product("Men Wool", 4700, 10)},

{"Women Cotton", new Product("Women Cotton", 6000, 10)},

{"Women Silk", new Product("Women Silk", 6500, 10)},

{"Women Wool", new Product("Women Wool", 6700, 10)},

{"Kids Cotton", new Product("Kids Cotton", 3000, 10)},

{"Kids Silk", new Product("Kids Silk", 3500, 10)},

{"Kids Wool", new Product("Kids Wool", 3700, 10)},

{"Hat", new Product("Hat", 500, 15)},

{"Scarf", new Product("Scarf", 700, 15)}

};

bool exit = false;

while (!exit)

{

Console.WriteLine("\n==== STORE MENU ====");

Console.WriteLine("1. Show Inventory");

Console.WriteLine("2. Add Product to Bag");

Console.WriteLine("3. View Shopping Bag");

Console.WriteLine("4. Remove Product from Bag");

Console.WriteLine("5. Complete Purchase");

Console.WriteLine("6. Sales Summary");

Console.WriteLine("7. Exit");

Console.Write("Enter your choice: ");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

DisplayInventory(inventory);

break;

case "2":

AddProductToBag(inventory);

break;

case "3":

DisplayShoppingBag();

break;

case "4":

RemoveProductFromBag(inventory);

break;

case "5":

ProcessCheckout();

break;

case "6":

DisplaySalesSummary();

break;

case "7":

exit = true;

break;

default:

Console.WriteLine("Invalid choice! Try again.");

break;

}

}

}

static void DisplayInventory(Dictionary<string, Product> inventory)

{

Console.WriteLine("\nAvailable Products:");

foreach (var item in inventory)

{

Console.WriteLine($"{item.Value.Title} - Rs.{item.Value.Cost} (Stock: {item.Value.AvailableStock})");

}

}

static void AddProductToBag(Dictionary<string, Product> inventory)

{

Console.Write("\nEnter product name: ");

string productName = Console.ReadLine();

if (!inventory.ContainsKey(productName))

{

Console.WriteLine("Product not found!");

return;

}

Product storeProduct = inventory[productName];

Console.Write("Enter quantity: ");

if (!int.TryParse(Console.ReadLine(), out int amount))

{

Console.WriteLine("Invalid input!");

return;

}

if (amount > storeProduct.AvailableStock)

{

Console.WriteLine("Sorry, insufficient stock!");

return;

}

Product foundInBag = shoppingBag.Find(x => x.Title == productName);

if (foundInBag != null)

{

foundInBag.QuantityInCart += amount;

}

else

{

var newProduct = new Product(storeProduct.Title, storeProduct.Cost, storeProduct.AvailableStock)

{

QuantityInCart = amount

};

shoppingBag.Add(newProduct);

}

storeProduct.AvailableStock -= amount;

Console.WriteLine($"{amount} {productName} added to your bag!");

}

static void DisplayShoppingBag()

{

if (shoppingBag.Count == 0)

{

Console.WriteLine("\nYour shopping bag is empty!");

return;

}

Console.WriteLine("\nShopping Bag Contents:");

int totalCost = 0;

foreach (var item in shoppingBag)

{

int lineTotal = item.Cost \* item.QuantityInCart;

Console.WriteLine($"{item.Title} x {item.QuantityInCart} = Rs.{lineTotal}");

totalCost += lineTotal;

}

Console.WriteLine($"Current Total: Rs.{totalCost}");

}

static void RemoveProductFromBag(Dictionary<string, Product> inventory)

{

Console.Write("\nEnter product name to remove: ");

string productName = Console.ReadLine();

Product foundItem = shoppingBag.Find(x => x.Title == productName);

if (foundItem != null)

{

if (inventory.ContainsKey(productName))

{

inventory[productName].AvailableStock += foundItem.QuantityInCart;

}

shoppingBag.Remove(foundItem);

Console.WriteLine($"{productName} removed from your bag!");

}

else

{

Console.WriteLine("Product not found in your bag!");

}

}

static void ProcessCheckout()

{

if (shoppingBag.Count == 0)

{

Console.WriteLine("Your shopping bag is empty!");

return;

}

int subtotalAmount = 0;

foreach (var item in shoppingBag)

{

int productTotal = item.Cost \* item.QuantityInCart;

if (item.QuantityInCart >= 3)

{

productTotal = (int)(productTotal \* 0.9);

Console.WriteLine($"10% bulk discount applied on {item.Title}");

}

subtotalAmount += productTotal;

if (salesTracker.ContainsKey(item.Title))

salesTracker[item.Title] += item.QuantityInCart;

else

salesTracker[item.Title] = item.QuantityInCart;

}

Console.Write("\nAdd gift wrap for Rs.200 per item? (yes/no): ");

string giftWrapAnswer = Console.ReadLine().ToLower();

if (giftWrapAnswer == "yes")

{

int wrapCharge = shoppingBag.Count \* 200;

subtotalAmount += wrapCharge;

Console.WriteLine($"Gift wrap cost added: Rs.{wrapCharge}");

}

if (subtotalAmount > 20000)

{

subtotalAmount = (int)(subtotalAmount \* 0.95);

Console.WriteLine("5% seasonal discount applied!");

}

Console.Write("\nEnter promo code (or press Enter to skip): ");

string promoInput = Console.ReadLine();

if (promoInput == "SALE10")

{

subtotalAmount = (int)(subtotalAmount \* 0.9);

Console.WriteLine("Promo code SALE10 applied: 10% off!");

}

Console.WriteLine($"\nYour final bill: Rs.{subtotalAmount}");

totalRevenue += subtotalAmount;

shoppingBag.Clear();

Console.WriteLine("Checkout done! Thanks for shopping!");

}

static void DisplaySalesSummary()

{

Console.WriteLine("\n==== SALES SUMMARY ====");

Console.WriteLine($"Total Revenue: Rs.{totalRevenue}");

if (salesTracker.Count == 0)

{

Console.WriteLine("No sales recorded yet.");

return;

}

string bestSeller = "";

int highestSales = 0;

foreach (var entry in salesTracker)

{

if (entry.Value > highestSales)

{

highestSales = entry.Value;

bestSeller = entry.Key;

}

}

Console.WriteLine($"Top-selling product: {bestSeller} ({highestSales} sold)");

}

}

}