Point-of-sales-for-supermarket JAVA Codes

AccSall.java

```
public interface AccSall {
   abstract String getId();
   abstract String getName();
   abstract double getPrice();
   abstract double getedPrice();


}
```

CartItem.java

```
public class CartItem {

   private AccSall item;
   private int quantity;

   // Constructor
   public CartItem(AccSall item, int quantity) {
      this.item = item;
      this.quantity = quantity;
   }

   // Getters and Setters
   public AccSall getItem() {
      return item;
   }

   public int getQuantity() {
      return quantity;
   }

   public void setQuantity(int quantity) {
      this.quantity = quantity;
   }

   // Calculate subtotal
   public double getSubtotal() {
      return item.getedPrice() * quantity;
   }

   @Override
   public String toString() {
      return String.format(item.getName() + " x " + getQuantity() + " = " + getSubtotal());
   }
}
```

Cart.java

```
// Cart class
public class Cart {

  private CartItem[] items;
  private int count;


  // Constructor - creates an empty cart
  public Cart() {
    this.items = new CartItem[10];
    this.count = 0;
  }

  // Add a product to cart
  public void addProduct(AccSall product, int quantity) {
    // Check if product already in cart
    for (int i = 0; i < count; i++) {

      if (items[i].getItem().getId() == product.getId()) {

        // Update quantity
        items[i].setQuantity(items[i].getQuantity() + quantity);

        return;
      }
    }

    // Check cart is full
    if (count >= 10) {

      System.out.println("Cart is full!");
      System.out.println("************************************************************");

      return;
    }

    // Add new item to cart
    items[count] = new CartItem(product, quantity);

    count++;
  }

  // Remove a product from cart
  public void removeProduct(String productId) {

    for (int i = 0; i < count; i++) {

      if (items[i].getItem().getId().equals(productId)) {

        items[i] = items[count - 1];

        items[count - 1] = null;
        count--;

        return;
      }
    }
  }

  // Calculate total price
  public double getTotal() {

    double total = 0;

    for (int i = 0; i < count; i++) {
```

```
            total += items[i].getSubtotal();
        }
        return total;

    }

    // Clear all items from cart
    public void clearCart() {

        items = new CartItem[10];
        count = 0;
    }

    // Get all items in cart
    public CartItem[] getItems() {

        CartItem[] result = new CartItem[count];

        for (int i = 0; i < count; i++) {

            result[i] = items[i];

        }
        return result;
    }

    public int getCount() {
        return count;
    }
}
```

Product.java

```
// Product class
public class Product implements AccSall {
    private String id;
    private String name;
    private double price;
    private int stock;

    // Constructor
    public Product(String id, String name, double price, int stock) {
        this.id = id;
        this.name = name;
        this.price = price;
        this.stock = stock;
    }

    //getters override
    @Override
    public String getId() {
        return id;
    }

    @Override
    public String getName() {
        return name;
    }

    @Override
    public double getPrice() {
        return price;
```

```
    }

    @Override
    public double getedPrice() {
        return price;  // No discount for regular products
    }

    public int getStock() {
        return stock;
    }

    public void updateStock(int quantity) {
        this.stock += quantity;
    }

    @Override
    public String toString() {
        return "ID: " + id + ", Name: " + name + ", Price: (RS)" + price + ", Stock: " + stock;
    }

}
```

SupermarketDriver.java (Main)

```
import java.util.Scanner;

//main drive ea
public class SupermarketDriver {
    private Product[] inventory;
    private int productCount;
    private Cart cart;
    private Scanner scanner; // i wnat to globale scanner

    // Constructor
    public SupermarketDriver() {

        this.inventory = new Product[100];
        this.productCount = 0;
        this.cart = new Cart();
        this.scanner = new Scanner(System.in);
        addingInventory();
    }

    // add sample inventory
     void addingInventory() {
        Product p1 = new Product("P001", "Rice 1kg", 140.00, 100);
        Product p2 = new Product("P002", "Book CR Page 120", 250.00, 100);
        Product p3 = new Product("P003", "Milk 1L", 120.00, 150);
        Product p4 = new Product("P004", "Eggs", 180.00, 100);

        addProduct(p1);
        addProduct(p2);
        addProduct(p3);
        addProduct(p4);
    }

    // Add product to inventory
     void addProduct(Product product) {

        if (productCount < 100) {

            inventory[productCount++] = product;
        }
    }
```

```
// Find product by ID
 Product findProduct(String id) {

   for (int i = 0; i < productCount; i++) {

      if (inventory[i].getId().equals(id)) {

         return inventory[i];
      }
   }
   return null;
}

// Main menu
public void start() {
   boolean running = true;
   while (running) {
      displayMenu();
      int choice = scanner.nextInt();
      scanner.nextLine(); //new error ekkak

      switch (choice) {
         case 1: addToCart();
         break;
         case 2: removeFromCart();
         break;
         case 3: viewCart();
         break;
         case 4: checkout();
         break;
         case 5: viewProducts();
         break;
         case 6: running = false;
         break;
         default: System.out.println("Invalid choice!");
      }
      System.out.println("************************************************************");
   }
}

// Display main menu
 void displayMenu() {

   System.out.println("            WELCOME           ");
   System.out.println("******* Supermarket Management System ********");

   System.out.println("1. Add to Cart");

   System.out.println("2. Remove from Cart");

   System.out.println("3. View Cart");

   System.out.println("4. Checkout");

   System.out.println("5. View Products");

   System.out.println("6. Exit");

   System.out.print("Choose option: ");


}

// Add product to cart
 void addToCart() {
```

```
    //ui ID
    System.out.print("Enter product ID: ");
    String id = scanner.nextLine();
    //ui qyt
    System.out.print("Enter quantity: ");
    int quantity = scanner.nextInt();

    Product product = findProduct(id);

    if (product != null && product.getStock() >= quantity) {

      cart.addProduct(product, quantity);

      product.updateStock(-quantity);

      System.out.println("Product added to cart!");
      System.out.println("************************************************************");

    } else {

      System.out.println("Product not found or it not in stock!");
      System.out.println("************************************************************");

    }
  }

  // Remove product  cart
  void removeFromCart() {

    System.out.print("Enter product ID to remove: ");
    String id = scanner.nextLine();

    cart.removeProduct(id);
    System.out.println("Product removed from cart!");
  }

  // View cart
  void viewCart() {
    System.out.println("****** Cart Contents *******");

    CartItem[] items = cart.getItems();

    for (CartItem item : items) {

      System.out.println(item);

    }
    System.out.printf("Total: "+ cart.getTotal());
    System.out.println(" ");
    System.out.println("************************************************************");

  }

  // Process checkout
  void checkout() {

    if (cart.getCount() == 0) {

      System.out.println("Cart is empty!");
      System.out.println("************************************************************");

      return;
    }

    viewCart();
```

```
      System.out.print("Enter payment amount: (RS) ");
      double payment = scanner.nextDouble();

      if (payment >= cart.getTotal()) {

        System.out.println("Change: " + (payment - cart.getTotal()));
        cart.clearCart();

        System.out.println("Thank you for your purchase!");
        System.out.println("************************************************************");

      } else {

        System.out.println("You can't make a payment.!");
        System.out.println("************************************************************");

      }
    }

    // View available products
     void viewProducts() {
      System.out.println("******** Available Products ********");

      for (int i = 0; i < productCount; i++) {

        System.out.println(inventory[i]);

      }
    }

    // Main method
    public static void main(String[] args) {

      SupermarketDriver sp1 = new SupermarketDriver();

      sp1.start();
    }
}
```