

APPENDIX

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
```

```
public class ECommercePlatform {
```

```
    // Product Class
```

```
    static class Product {
```

```
        private int id;
```

```
        private String name;
```

```
        private double price;
```

```
        private int quantity;
```

```
        public Product(int id, String name, double price, int quantity) {
```

```
            this.id = id;
```

```
            this.name = name;
```

```
            this.price = price;
```

```
            this.quantity = quantity;
```

```
        }
```

```
        public int getId() {
```

```
            return id;
```

```
        }
```

```
        public String getName() {
```

```
            return name;
```

```
        }
```

```

public double getPrice() {
    return price;
}

```

```

public int getQuantity() {
    return quantity;
}

```

```

public void reduceQuantity(int amount) {
    if (quantity >= amount) {
        quantity -= amount;
    }
}

```

```

@Override
public String toString() {
    return "Product{" +
        "id=" + id +
        ", name=" + name + "\"" +
        ", price=" + price +
        ", quantity=" + quantity +
        "'";
}
}

```

```

// Cart Class
static class Cart {
    private List<CartItem> cartItems = new ArrayList<>();
}

```

```

        // CartItem class to store quantity of each product in the cart
static class CartItem {
    Product product;
    int quantity;

    CartItem(Product product, int quantity) {
        this.product = product;
        this.quantity = quantity;
    }

    public double getTotalPrice() {
        return product.getPrice() * quantity;
    }
}

    public void addProduct(Product product, int quantity) {
        // Check if product already in cart, then update the quantity
        for (CartItem item : cartItems) {
            if (item.product.getId() == product.getId()) {
                // Check if there is enough stock to add to the cart
                if (item.product.getQuantity() >= (item.quantity + quantity)) {
                    item.quantity += quantity;
                    product.reduceQuantity(quantity);
                    System.out.println(quantity + " " + product.getName() + " added to cart.");
                    return;
                } else {
                    System.out.println("Not enough quantity of " + product.getName());
                    return;
                }
            }
        }
    }

```

```

        }
    }

    // If product is not in cart, add it
    if (product.getQuantity() >= quantity) {
        cartItems.add(new CartItem(product, quantity));
        product.reduceQuantity(quantity);
        System.out.println(quantity + " " + product.getName() + " added to cart.");
    } else {
        System.out.println("Not enough quantity of " + product.getName());
    }
}

public double calculateTotal() {
    double total = 0;
    for (CartItem item : cartItems) {
        total += item.getTotalPrice();
    }
    return total;
}

public void displayCart() {
    if (cartItems.isEmpty()) {
        System.out.println("Your cart is empty.");
    } else {
        for (CartItem item : cartItems) {
            System.out.println(item.product.getName() + " - Quantity: " + item.quantity + " - Total: $" + item.getTotalPrice());
        }
    }
}

```

```

    }
}

// User Class
static class User {
private String username;
private String password;

public User(String username, String password) {
this.username = username;
this.password = password;
}

public boolean login(String inputUsername, String inputPassword) {
return inputUsername.equals(username) && inputPassword.equals(password);
}

public String getUsername() {
return username;
}
}

// Order Class
static class Order {
private Cart cart;
private double totalAmount;

public Order(Cart cart) {

```

```

this.cart = cart;
this.totalAmount = cart.calculateTotal();
    }

public void processOrder() {
    System.out.println("Processing Order...");
    System.out.println("Total amount: $" + totalAmount);
    System.out.println("Order successfully placed!");
    }
}

// Main Function
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    // Create sample products
    Product product1 = new Product(1, "Laptop", 999.99, 5);
    Product product2 = new Product(2, "Smartphone", 599.99, 10);

    // Create user
    User user = new User("vishnu", "1234");

    System.out.println("Welcome to the E-Commerce Platform!");

    // User Login
    System.out.println("Enter username:");
    String username = scanner.nextLine();
    System.out.println("Enter password:");
    String password = scanner.nextLine();

```

```

if (user.login(username, password)) {
    System.out.println("Login successful!");

    // Create a shopping cart
    Cart cart = new Cart();
    while (true) {
        System.out.println("\nSelect an option:");
        System.out.println("1. Add Laptop to cart");
        System.out.println("2. Add Smartphone to cart");
        System.out.println("3. View Cart");
        System.out.println("4. Checkout");
        System.out.println("5. Exit");

        // Input for menu choice
        int choice = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character left by nextInt()

        switch (choice) {
            case 1:
                cart.addProduct(product1, 1);
                break;
            case 2:
                cart.addProduct(product2, 1);
                break;
            case 3:
                cart.displayCart();
                break;
            case 4:

```

```

        Order order = new Order(cart);
    order.processOrder();
    return; // Exit after processing order
case 5:
    System.out.println("Exiting...");
    return; // Exit program
default:
    System.out.println("Invalid option. Try again.");
        }
    }
    } else {
    System.out.println("Invalid credentials. Please try again.");
        }
    }
}

```


RESULTS AND DISCUSSION

```
Welcome to the E-Commerce Platform!  
Enter username:vishnu  
Enter password:1234  
Login successful!
```

```
Select an option:  
1. Add Laptop to cart  
2. Add Smartphone to cart  
3. View Cart  
4. Checkout  
5. Exit  
1  
1 Laptop added to cart.
```

```
Select an option:  
1. Add Laptop to cart  
2. Add Smartphone to cart  
3. View Cart  
4. Checkout  
5. Exit
```

```
2  
1 Smartphone added to cart.  
  
Select an option:  
1. Add Laptop to cart  
2. Add Smartphone to cart  
3. View Cart  
4. Checkout  
5. Exit  
3  
Laptop - Quantity: 1 - Total: $999.99  
Smartphone - Quantity: 1 - Total: $599.99  
  
Select an option:  
1. Add Laptop to cart  
2. Add Smartphone to cart  
3. View Cart  
4. Checkout  
5. Exit  
4  
Processing Order...  
Total amount: $1599.98  
Order successfully placed!
```