E-COMMERCE CART APPLICATION

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Structure for Product
typedef struct {
    int id;
    char name[50];
    float price;
    int stock;
} Product;

// Structure for Cart Item
typedef struct {
    int productId;
    int quantity;
} CartItem;
// Structure for Order
typedef struct {
    int orderId;
    CartItem items[10];
    int itemCount;
    char status[20];
} Order;
```

```
// Structure for User
typedef struct {
    char username[50];
    CartItem cart[10];
    int cartCount;
    Order orders[10];
    int orderCount;
} User;
void browseProducts(Product products[], int productCount);
void addToCart(User *user, Product products[], int productCount);
void updateQuantity(User *user);
void viewCart(User *user, Product products[], int productCount);
void checkout(User *user, Product products[], int productCount);
void viewOrderHistory(User *user);
void trackShipment(User *user);
void browseProducts(Product products[], int productCount) {
    printf("Available Products:\n");
    for (int i = 0; i < productCount; i++) {
        printf("%d: %s - $%.2f (Stock: %d)\n", products[i].id, products[i].name,
products[i].price, products[i].stock);
    }
}
void addToCart(User *user, Product products[], int productCount) {
    int productId, quantity;
    printf("Enter Product ID to add to cart: ");
    scanf("%d", &productId);
```

```
    printf("Enter Quantity: ");
    scanf("%d", &quantity);

    for (int i = 0; i < productCount; i++) {
        if (products[i].id == productId && products[i].stock >= quantity) {
            user->cart[user->cartCount].productId = productId;
            user->cart[user->cartCount].quantity = quantity;
            user->cartCount++;
            products[i].stock -= quantity;
            printf("Added to cart successfully!\n");
            return;
        }
    }
    printf("Product not found or insufficient stock.\n");
}
void updateQuantity(User *user) {
    int productId, quantity;
    printf("Enter Product ID to update quantity: ");
    scanf("%d", &productId);
    printf("Enter new Quantity: ");
    scanf("%d", &quantity);

    for (int i = 0; i < user->cartCount; i++) {
        if (user->cart[i].productId == productId) {
            user->cart[i].quantity = quantity;
            printf("Quantity updated successfully!\n");
            return;
```

```
        }
    }
    printf("Product not found in cart.\n");
}
void viewCart(User *user, Product products[], int productCount) {
    printf("Your Cart:\n");
    for (int i = 0; i < user->cartCount; i++) {
        for (int j = 0; j < productCount; j++) {
            if (products[j].id == user->cart[i].productId) {
                printf("%s - Quantity: %d\n", products[j].name, user->cart[i].quantity);
            }
        }
    }
}
void checkout(User *user, Product products[], int productCount) {
    printf("Checking out...\n");
    Order order;
    order.orderId = user->orderCount + 1;
    order.itemCount = user->cartCount;
    strcpy(order.status, "Processing");
    for (int i = 0; i < user->cartCount; i++) {
        order.items[i] = user->cart[i];
    }
    user->orders[user->orderCount] = order;
    user->orderCount++;
    user->cartCount = 0;
    printf("Order placed successfully!\n");
```

```
}
void viewOrderHistory(User *user) {
    printf("Order History:\n");
    for (int i = 0; i < user->orderCount; i++) {
        printf("Order ID: %d, Status: %s\n", user->orders[i].orderId, user-
>orders[i].status);
    }
}
void trackShipment(User *user) {
    int orderId;
    printf("Enter Order ID to track: ");
    scanf("%d", &orderId);

    for (int i = 0; i < user->orderCount; i++) {
        if (user->orders[i].orderId == orderId) {
            printf("Order ID: %d, Status: %s\n", user->orders[i].orderId, user-
>orders[i].status);
            return;
        }
    }
    printf("Order not found.\n");
}
int main() {
    Product products[3] = {
        {1, "Product A", 10.0, 100},
        {2, "Product B", 20.0, 50},
        {3, "Product C", 30.0, 30}
    };
```

```
User user = {"JohnDoe", {}, 0, {}, 0};
int choice;

do {
    printf("\nE-commerce Cart Application\n");
    printf("1. Browse Products\n");
    printf("2. Add to Cart\n");
    printf("3. Update Quantity\n");
    printf("4. View Cart\n");
    printf("5. Checkout\n");
    printf("6. View Order History\n");
    printf("7. Track Shipment\n");
    printf("0. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

    switch (choice) {
        case 1:
            browseProducts(products, 3);
            break;
        case 2:
            addToCart(&user, products, 3);
            break;
        case 3:
            updateQuantity(&user);
            break;
        case 4:
```

```
            viewCart(&user, products, 3);
            break;
        case 5:
            checkout(&user, products, 3);
            break;
        case 6:
            viewOrderHistory(&user);
            break;
        case 7:
            trackShipment(&user);
            break;
        case 0:
            printf("Exiting...\n");
            break;
        default:
            printf("Invalid choice. Please try again.\n");
    }
  } while (choice != 0);

  return 0;
}
```