📘 Documentation – Simple Online Shopping Project

1. Introduction

This project is a **mini e-commerce system** where users can browse products, add them to cart, and place an order.

- **Backend**: ASP.NET Core Web API (with EF Core + JWT authentication).
- **Frontend**: Static HTML + jQuery AJAX for API calls.

---

2. Features

Authentication

- Register new users.
- Login to get a JWT token.

Products

- List all products.
- View product details. (Optional)
- Admin can add/edit/delete products.

Cart & Orders

- Add products to cart (stored in DB per user).
- View cart.
- Checkout → create an order.
- View order history.

3. Architecture

```
Frontend (HTML + jQuery)
   |
   |---- AJAX (JSON)
   |
Backend (ASP.NET Core Web API)
   |---- Controllers (Auth, Products, Cart, Orders)
   |---- EF Core (SQLite/SQL Server)
   |---- JWT Authentication
```

---

4. Entities (Database Schema)

1. **User**
   - Id, Email, Password
2. **Product**
   - Id, Name, Price, Description.
3. **CartItem**
   - Id, UserId (FK), ProductId (FK), Quantity.
4. **Order**
   - Id, UserId (FK), OrderDate, Total.
5. **OrderItem**
   - Id, OrderId (FK), ProductId (FK), Quantity, Price.

---

5. API Endpoints

Auth

- POST /api/auth/register → Register user.
- POST /api/auth/login → Login + get JWT.

Products

- GET /api/products → List all products.
- POST /api/products → Add product (Admin only).

Cart

- GET /api/cart → Get user's cart.
- POST /api/cart/add → Add product to cart.
- DELETE /api/cart/remove/{productId} → Remove product.

Orders

- POST /api/orders/checkout → Checkout current cart.
- GET /api/orders → View user's orders.

---

6. Frontend Pages

1. **Home (products.html)** → List products, "Add to Cart" button.
2. **Cart (cart.html)** → Show cart items, "Checkout" button.
3. **Login (login.html)** → Login form.
4. **Register (register.html)** → Registration form.
5. **Orders (orders.html)** → Show order history.

---

7. Learning Outcomes

- Build **REST API** with ASP.NET Core Web API.
- Manage **database relationships** with EF Core.
- Secure APIs with **JWT authentication**.
- Use **jQuery AJAX** for dynamic frontend.
- Understand **cart and order processing** logic.