# 🛒 Project: **Nexora – E-Commerce Backend System**

### 🔑 Tech Stack

* **Backend Framework:** Django + Django REST Framework (DRF)
* **Database:** MySQL
* **Auth:** JWT Authentication (using djangorestframework-simplejwt)
* **API Docs:** Swagger / DRF YASG
* **Deployment (optional):** AWS EC2 or PythonAnywhere

## 📌 Project Features (Backend Only)

### 1. **Authentication & Authorization**

* User registration & login (JWT token based).
* Roles: **Admin, Seller, Customer**.
* Role-based permissions (e.g., only admin can add categories, sellers can add products).

### 2. **Product Management**

* CRUD APIs for products (/api/products/).
* Category & brand management (/api/categories/, /api/brands/).
* Product search & filter (by price, category, rating, keyword).

### 3. **Cart & Checkout APIs**

* Add/remove items in cart.
* Update quantity in cart.
* Checkout API → creates **order** from cart items.

### 4. **Order Management**

* Create Order API.
* Order History API (customer can see past orders).
* Order Status update API (admin/seller side → e.g., "Pending", "Shipped", "Delivered").

### 5. **Payment Gateway Integration**

* Payment API endpoint (simulate Razorpay/Stripe).
* Store transaction details in database.
* Order gets confirmed only if payment is successful.

### 6. **Customer APIs**

* User profile API (update email, address, phone).
* Wishlist API (add/remove products to wishlist).

### 7. **Admin APIs**

* Manage products, categories, users, and sellers.
* Apply discount coupons.
* Manage inventory (stock in/out).

### 8. **Extra Features**

* API Documentation (Swagger or DRF YASG).
* Error handling + custom exception middleware.
* Logging (user actions, errors).
* API rate limiting (prevent brute force).

📂 **Folder Structure Example**

nexora/

│── nexora/ # Main project

│── accounts/ # Auth, User, Roles

│── products/ # Products, Categories, Wishlist

│── cart/ # Cart & Checkout

│── orders/ # Orders, Payments

│── common/ # Utils, Middlewares, Custom Exceptions

│── requirements.txt

│── manage.py

👉 This project will look very strong in your resume because it is:  
✔ End-to-End backend only (no frontend).  
✔ Real-world **e-commerce use case** (companies love this).  
✔ Uses **Python, Django, DRF, MySQL, REST APIs** (your exact skills).  
✔ Covers advanced concepts like **JWT auth, role-based access, payments, API docs**.