🔥 This is the **Final Part (Part 6)** 🚀

Now I’ll give you:

1. 📑 **API Endpoint Documentation (table format)**
2. 📝 **Resume-Ready Project Description**
3. 💡 **Interview Explanation (how to present)**
4. 🌐 **Push to GitHub (best practices)**

# 1️⃣ API Endpoint List (Nexora Backend)

| Module | Method | Endpoint | Description | Role |
| --- | --- | --- | --- | --- |
| **Auth** | POST | /api/auth/register/ | Register user | All |
|  | POST | /api/auth/login/ | Login (JWT) | All |
|  | POST | /api/auth/refresh/ | Refresh token | All |
|  | GET/PUT | /api/auth/me/ | View / update profile | Auth only |
| **Catalog** | GET | /api/catalog/categories/ | List categories | All |
|  | POST | /api/catalog/categories/ | Create category | Admin/Seller |
|  | GET | /api/catalog/products/ | List/search products | All |
|  | POST | /api/catalog/products/ | Create product | Admin/Seller |
| **Wishlist** | GET | /api/catalog/wishlist/ | My wishlist | Customer |
|  | POST | /api/catalog/wishlist/ | Add product to wishlist | Customer |
|  | DELETE | /api/catalog/wishlist/{id}/ | Remove product | Customer |
| **Cart** | GET | /api/cart/ | View my cart | Customer |
|  | POST | /api/cart/ | Add product | Customer |
|  | PUT | /api/cart/{id}/ | Update quantity | Customer |
|  | DELETE | /api/cart/{id}/ | Remove item | Customer |
| **Checkout** | POST | /api/checkout/ | Create order from cart | Customer |
| **Orders** | GET | /api/orders/ | View my orders | Customer |
|  | PATCH | /api/orders/{id}/ | Update status | Admin |
| **Payments** | POST | /api/payments/ | Pay for order | Customer |
| **Coupons** | POST | /api/coupons/ | Create coupon | Admin |
|  | GET | /api/coupons/ | List coupons | Admin |

# 2️⃣ Resume-Ready Project Description

**Project: Nexora – E-commerce Backend (Python, Django, DRF, MySQL)**

* Designed and developed a **scalable backend system** for an e-commerce platform with role-based access (**Admin, Seller, Customer**).
* Implemented **JWT authentication** with custom user roles and secure profile management.
* Built REST APIs for **catalog management** (products, categories, brands) with search and filtering.
* Developed **cart and checkout system** with automatic **stock deduction**.
* Designed **order management workflow** (Pending → Paid → Shipped → Delivered) with simulated **payment gateway integration**.
* Added **wishlist and coupon system** for better customer engagement.
* Integrated **Swagger API documentation**, **rate-limiting**, and **error logging** for production readiness.
* Deployed using **AWS EC2 + MySQL (RDS)** with Gunicorn and Nginx for scalability.

**Tech Stack:** Python, Django, DRF, MySQL, JWT, Swagger, AWS EC2

# 3️⃣ How to Explain in Interview (Sample Answer)

🧑‍💼 **Q: Can you explain your project?**  
👉 “Yes, I built an **E-commerce backend system called Nexora** using Python, Django, and DRF.  
It’s a complete backend project with role-based access – **Admin, Seller, and Customer**.

* Admin can manage users, coupons, products, and update order statuses.
* Sellers can add products and manage inventory.
* Customers can browse products, manage a cart, place orders, make payments, and maintain a wishlist.

I implemented **JWT authentication**, **RESTful APIs**, and **MySQL database** for data persistence. I also added **Swagger docs**, **rate limiting**, and **logging** to make it production-ready.  
This project gave me strong experience in **backend development, API design, database modeling, and deployment** on AWS.”

# 4️⃣ Push to GitHub (Best Practices)

# 1. Create repo on GitHub: "nexora-backend"

git init

git remote add origin https://github.com/yourusername/nexora-backend.git

# 2. Add .gitignore

echo ".venv/" >> .gitignore

echo "\_\_pycache\_\_/" >> .gitignore

echo "\*.sqlite3" >> .gitignore

echo "\*.log" >> .gitignore

echo ".env" >> .gitignore

# 3. Commit code

git add .

git commit -m "Initial commit - Nexora Backend Project"

# 4. Push to GitHub

git branch -M main

git push -u origin main

📌 **Tips**

* Put **README.md** with: project intro, setup steps, API docs link.
* Add **requirements.txt** → pip freeze > requirements.txt
* Add **sample .env file** (without secrets).

✅ With this, your project is **end-to-end complete** and **interview-ready**.  
It shows **backend-only expertise**, **real-world features**, and **deployability**.