Got it 👍 — here's a **step-by-step documentation (no code, just clear steps)** on how to build the E-Commerce API project.

🛒 E-Commerce API — Step by Step Guide

1. Project Setup

1. Create a new project folder.
2. Run `npm init -y` to set up `package.json`.
3. Install dependencies: `express`, `mongoose`, `dotenv`, `bcrypt`, `jsonwebtoken`, `morgan`, `nodemon`.
4. Create folder structure:
5. `src/config` → DB connection
6. `src/models` → Mongoose schemas
7. `src/controllers` → Business logic
8. `src/routes` → Route definitions
9. `src/middleware` → Auth & validation
10. `src/server.js` → Entry point

---

2. Environment Setup

1. Create a `.env` file with:
2. `PORT`
3. `MONGO_URI`
4. `JWT_SECRET`
5. Connect to MongoDB using Mongoose.

---

3. User Authentication

1. Create a `User` model with fields: `username`, `email`, `password`, `role` (default: customer).
2. Add password hashing before saving.
3. Create routes for:
4. `POST /auth/signup`
5. `POST /auth/login`
6. Return a JWT token on login.

---

4. Product Management

1. Create a `Product` model with fields: `name`, `description`, `price`, `stock`, `category`.

2. Routes:
3. `POST /products` → Add product (admin only).
4. `GET /products` → Get all products.
5. `GET /products/:id` → Get product by ID.
6. `PUT /products/:id` → Update product (admin only).
7. `DELETE /products/:id` → Delete product (admin only).

---

5. Cart System

1. Decide: store cart as a field inside `User` OR a separate `Cart` collection.
2. Easier: `Cart` collection with `userId` + list of items.
3. Routes:
4. `POST /cart/add` → Add product to cart.
5. `POST /cart/remove` → Remove product.
6. `GET /cart` → View user's cart.

---

6. Orders

1. Create an `Order` model: `userId`, `items`, `totalPrice`, `status` (default: Pending).
2. Routes:
3. `POST /orders` → Place an order (moves items from cart to orders).
4. `GET /orders` → Get all orders of logged-in user.
5. (Optional) Admin: update order status.

---

7. Middleware

1. **Auth middleware**: Verify JWT token and attach `req.user`.
2. **Role middleware**: Allow only admins to manage products.

---

8. Extra Features (Optional)

- Search/filter products (by category, price).
- Pagination on product list.
- Wishlist system.
- Reviews for products.
- Payment simulation (just change order status).

---

9. Testing

1. Use Postman or curl to test all endpoints.
2. Check authentication flow: signup → login → token usage.
3. Verify admin vs. customer permissions.

---

10. Deployment (Optional)

1. Push code to GitHub.
2. Deploy with Render, Vercel, or Railway.
3. Use MongoDB Atlas for database hosting.

👉Follow these steps one by one, and you'll end up with a fully functional E-Commerce API project.

Would you like me to turn this into a **checklist format** (so you can tick off each step as you progress), or keep it as a linear guide?