E-Commerce Project Development Phases

■ Phase 1: MVP (2 Weeks)

Goal: Working basic e-commerce shop to show junior-level skills.
Frontend (React + Tailwind):
- Home page with product listing.
- Product detail page (clickable cards).
- Shopping cart page (add/remove items).
- Checkout page (dummy checkout flow).
Backend (Node.js + Express):
- REST API: GET /products, GET /products/:id, POST /cart, POST /checkout.
- Basic error handling.
Database (PostgreSQL):
- users (id, name, email, password).
- products (id, name, price, stock, description).
- orders (id, user_id, total, status).
- order_items (id, order_id, product_id, qty, price).
Deliverable: Simple, working shop (add-to-cart → checkout).

■ Phase 2: Prototype 2 (Next Iteration, +2–4 weeks)

Goal: Add more realism, advanced learning.
Frontend:
- User authentication (login/register).
- Responsive UI with Tailwind refinements.
- Protected routes (only logged-in users can checkout).
- Add product management UI for shop owner.
Backend:
- JWT-based authentication.
- CRUD API for shop owner (add, edit, delete products).
- Transaction handling for stock deduction.
- File/image upload for product photos.
Database:
- Add constraints & indexes.
- Implement transactions for order + stock updates.
Extra Features:
- Testing (React Testing Library, Jest, Supertest).
- Basic SEO improvements (meta tags, titles).
- Error logging & validation.
Deliverable: Realistic shop where users can sign up, log in, buy products, and shop owner can manage products.

■ Phase 3: Complete Project (Senior-Level, Full System)

Goal: Full-fledged e-commerce platform with microservices and advanced features.
Frontend (React → Next.js):
- Server-Side Rendering (SSR) for SEO.
- Advanced search and filtering.
- Product recommendations.
- Payment gateway integration (Stripe, PayPal).
Backend (Node.js + Express + Java Microservices):

- Java microservice: invoicing (PDF generator).
- Java microservice: inventory forecasting.
- Delivery tracking integration.
- Admin dashboard for analytics (sales, profit, users).

Database:
- PostgreSQL with JSONB, triggers, views.
- Redis for caching.
- ElasticSearch for advanced search.

Infrastructure:
- GitHub CI/CD pipelines.
- Docker for containerization.
- Deployment on AWS/GCP.
- HTTPS enabled (Let’s Encrypt).

Deliverable: Senior-level portfolio project with full stack, microservices, and production-grade features.