Public Deployment Plan (Render + Vercel)

A simple, low-stress public deployment path for your e-commerce app: Backend on Render + Frontend on Vercel. This mirrors a common real-world setup and is recruiter-friendly.

1) Deploy Backend on Render

1. Go to render.com → New → Web Service
2. Connect your GitHub repo
3. Select the backend folder
4. Settings: Build command = npm install; Start command = npm start; Environment = Node
5. Add environment variables:
6. - MONGO_URI=
7. - JWT_SECRET=
8. - FRONTEND_ORIGIN=https://your-vercel-domain.vercel.app
9. - PORT=4000
10. Deploy and wait for “Live”
11. Test: https://your-backend.onrender.com/health (expect status OK / db connected)

2) Deploy Frontend on Vercel

1. Go to vercel.com → New Project
2. Import the same repo
3. Choose the frontend folder
4. Add environment variable:
5. - NEXT_PUBLIC_API_URL=https://your-backend.onrender.com
6. Deploy
7. Result: a URL like https://ecommerce-app.vercel.app

3) Update CORS / Origin (Important)

- In Render backend env, set FRONTEND_ORIGIN to your Vercel domain (e.g., https://ecommerce-app.vercel.app).
- This is important so cookies + auth work correctly.

4) Final Recruiter Checklist

- Local: runs with Docker Compose

- Public: deployed with Render (API) + Vercel (UI)
- You can say: “I handled environment differences between Docker, local dev, and production.”

What Not To Do Right Now

- Don’t deploy Docker to production yet
- Don’t over-engineer CI
- Don’t worry about HTTPS (Vercel + Render handle it)