📦Amazon-Style Full-Stack E-commerce Website for Personal Brand

Target User:

Solo entrepreneur (non-coder) building a professional, secure, mobile-first e-commerce site like Amazon.

🔗Tech Stack

Frontend (client/)

- React (Vite)
- Tailwind CSS
- React Router DOM
- Framer Motion (animations)
- Context API or Redux (cart, auth state)

Backend (server/)

- Node.js + Express.js
- MongoDB (Mongoose)
- JWT (auth)
- bcryptjs (password hashing)
- Multer (image upload)
- dotenv (config)
- CORS, Helmet (security)

Deployment

- Frontend: Vercel
- Backend: Render or Railway
- Database: MongoDB Atlas

🧾Pages List

🕰️Customer Pages

- `/` → Home Page
- `/products` → Product List + Search/Filter
- `/product/:id` → Product Detail Page
- `/cart` → Cart Page
- `/checkout` → Checkout + Address + OTP (COD orders)
- `/login` → Login with Email/Mobile + OTP
- `/register` → Register with OTP

- `/forgot-password` → Forgot password via OTP
- `/account` → My Profile Page
- `/account/orders` → Order List
- `/account/orders/:id` → Order Details
- `/contact`, `/about`, `*` → Info & 404

🔐Authentication Pages

- OTP-based login/signup/reset
- Resend OTP with timer

🔧Admin Pages (Protected)

- `/admin/login` → Admin login
- `/admin/dashboard` → Overview: sales, orders, users
- `/admin/products` → Product List + Manage
- `/admin/products/new` → Add Product
- `/admin/products/:id/edit` → Edit Product
- `/admin/orders` → Manage Orders
- `/admin/orders/:id` → View + Update Status
- `/admin/users` *(optional)* → Customer list

Backend Requirements

🗃️API Routes

Auth

- POST `/api/auth/register`
- POST `/api/auth/login`
- POST `/api/auth/send-otp`
- POST `/api/auth/verify-otp`
- POST `/api/auth/forgot-password`
- PUT `/api/user/update`

Products

- GET `/api/products`
- GET `/api/products/:id`
- POST `/api/products` *(admin)*
- PUT `/api/products/:id` *(admin)*
- DELETE `/api/products/:id` *(admin)*

Orders

- POST `/api/orders`

- GET `/api/orders/user`
- GET `/api/orders` *(admin)*
- GET `/api/orders/:id`
- PUT `/api/orders/:id/status` *(admin)*

Upload

- POST `/api/upload` *(Multer for product images)*

🗄️Models

- User: name, email, phone, password, isAdmin
- Product: name, price, stock, category, image, description
- Order: user ref, items, shipping address, status, total
- OTP: code, target (email/phone), expiresAt, isUsed

🔐OTP Logic

Use Cases

1. Login via OTP
2. Signup via OTP
3. Forgot password via OTP
4. Order confirmation (COD) via OTP

Flow

- User submits phone/email
- Server generates OTP
- OTP saved with expiry (5 mins)
- Sent via email/SMS (Fast2SMS, Twilio, or Nodemailer)
- User enters OTP → verify → action allowed (login/order)

📁Folder Structure

client/

- components/
- pages/
- context/
- routes/
- animations/
- assets/

server/

- models/
- routes/
- controllers/
- middleware/
- uploads/
- .env, server.js

🕖Cursor Prompt

Paste this prompt into Cursor:

```
Create a complete full-stack e-commerce site optimized for a solo seller. Use
React (with Vite) + Tailwind CSS + Framer Motion on the frontend, and Node.js +
Express + MongoDB on the backend.

Frontend:
- Home, Product listing, Product detail, Cart, Checkout
- Login/Register with OTP via mobile/email
- Forgot password via OTP
- My Account, My Orders, Order Detail
- Admin panel with Dashboard, Add/Edit Product, View Orders
- Responsive, mobile-first design with animations

Backend:
- Secure auth with JWT and bcrypt
- Product/order/user routes
- OTP APIs (send, verify)
- Multer for product image uploads
- Admin-protected routes

Structure the project into client/ and server/ folders with proper MVC pattern
and production-ready code. Deployable on Vercel, Render, and MongoDB Atlas.
```

🗜️Next Step

- Paste the above prompt into Cursor
- Customize branding, logo, colors
- Replace demo products with your own
- Ready to launch!