**✅ Development Document Text (for this step)**

**📄 Backend Setup (Node + Express + Sequelize)**

* Initialized project with npm init
* Installed packages: express, mysql2, sequelize, cors, dotenv, bcrypt, jsonwebtoken
* Setup folder structure: controllers, models, routes, config, etc.
* Created MySQL connection using Sequelize in config/db.js
* Setup .env for DB and secret configs
* Created server.js as backend entry point
* Verified connection to MySQL DB

**📘 User Guide Text (for this step)**

This is backend setup for the e-commerce website.  
No user-facing UI yet — backend server is prepared and connected to database successfully.

**📄 Development Doc (for this step)**

**🛠️ Authentication System (Register/Login)**

* Created User model (with role field for RBAC)
* Setup routes: /api/auth/register and /api/auth/login
* Passwords hashed using bcrypt
* JWT issued on successful login
* Created authController.js for logic
* Tested using Postman with sample data

**📘 User Guide (for this step)**

**👤 User Login/Registration**

* Users can register using name, email, password
* On login, a secure token is returned
* This token is used to keep the user logged in
* Keep your token safe – it's your login key

**📄 Development Document (JWT Middleware)**

* Created authMiddleware.js to validate JWT from request headers
* Middleware extracts user data and attaches it to req.user
* Applied it to a sample protected route /api/protected/dashboard
* Verified with token from login endpoint using Postman

**📘 User Guide (Protected Route)**

* After logging in, you receive a token
* This token must be sent in the **Authorization** header
* Use it to access protected pages like Dashboard, Cart, etc.
* If the token is missing or invalid, access is denied

**📄 Development Document (Frontend Setup)**

* Created React project using Vite
* Installed react-router-dom and axios
* Set up folder structure (components, pages, context, etc.)
* Configured routing for /login and /register
* Tested that frontend runs successfully on localhost:5173

**📘 User Guide (Frontend Setup)**

* The e-commerce website opens on port 5173
* /login and /register routes are available
* Login/Register pages will be connected to backend soon
* More pages (home, products, cart) will be added step-by-step

**📄 Development Document (Frontend Auth Integration)**

* Connected Register & Login forms to backend API
* Used axios to POST data to /auth/register and /auth/login
* Handled success & error messages
* Stored JWT token in localStorage for session tracking
* Used useNavigate() for redirecting after login/register

**📘 User Guide (Login/Register)**

* Visit /register to create an account
* Visit /login to sign in using your email & password
* After login, token is saved and user is redirected
* Token will be used for secure access to other pages

**📄 Development Doc (Auth Context & Protected Routes)**

* Implemented AuthContext to manage login state globally
* Used useEffect to persist token from localStorage
* Added Navbar with dynamic login/logout buttons
* Protected routes using <PrivateRoute> wrapper

**📘 User Guide (Login Session & Navbar)**

* After login, the user remains logged in even after refresh
* Navbar shows different buttons based on login status
* Secure pages like Cart and Orders are only accessible when logged in
* Logout clears the session and redirects to login

 **OTP + Email on Registration**

 **Admin Order List & Status Update**

 **User Dashboard/Profile**

 **Search + Filter + Sort (Frontend)**

 **Payment Integration**

 **Start Deployment**