ShoppyGlobe Backend – Project Documentation

# Github Link :-

https://github.com/noubahar123/ShoppyGlobe-backend.git

# 1) Project Overview

Goal: Build Node.js + Express + MongoDB backend for ShoppyGlobe with authentication and cart features.

# 2) Tech Stack

- Node.js, Express 5  
- MongoDB (Atlas)  
- Mongoose  
- JWT (jsonwebtoken), bcryptjs  
- express-validator, morgan, cors

# 3) How to Run (No .env)

1. Install dependencies: npm install  
2. Start server: npm run dev  
3. Seed demo products: npm run seed  
4. Server URL: http://localhost:5000  
  
Hardcoded config used:  
PORT = 5000  
MONGODB\_URI = mongodb+srv://shoppyglobe:shoppyglobe123@cluster0.d6xr0h5.mongodb.net/shoppyglobe?retryWrites=true&w=majority&appName=Cluster0  
JWT\_SECRET = "supersecretkey"  
JWT\_EXPIRES\_IN = "7d"

# 4) API Routes Summary

Products (Read-Only):  
- GET /products — list products  
- GET /products/:id — single product by id  
  
Auth:  
- POST /auth/register — { name, email, password }  
- POST /auth/login — { email, password } → returns { token, user }  
  
Cart (Protected with Bearer token):  
- GET /cart — list current user's cart items (populated product)  
- POST /cart — { productId, quantity }  
- PUT /cart/:id — { quantity }  
- DELETE /cart/:id — remove cart item

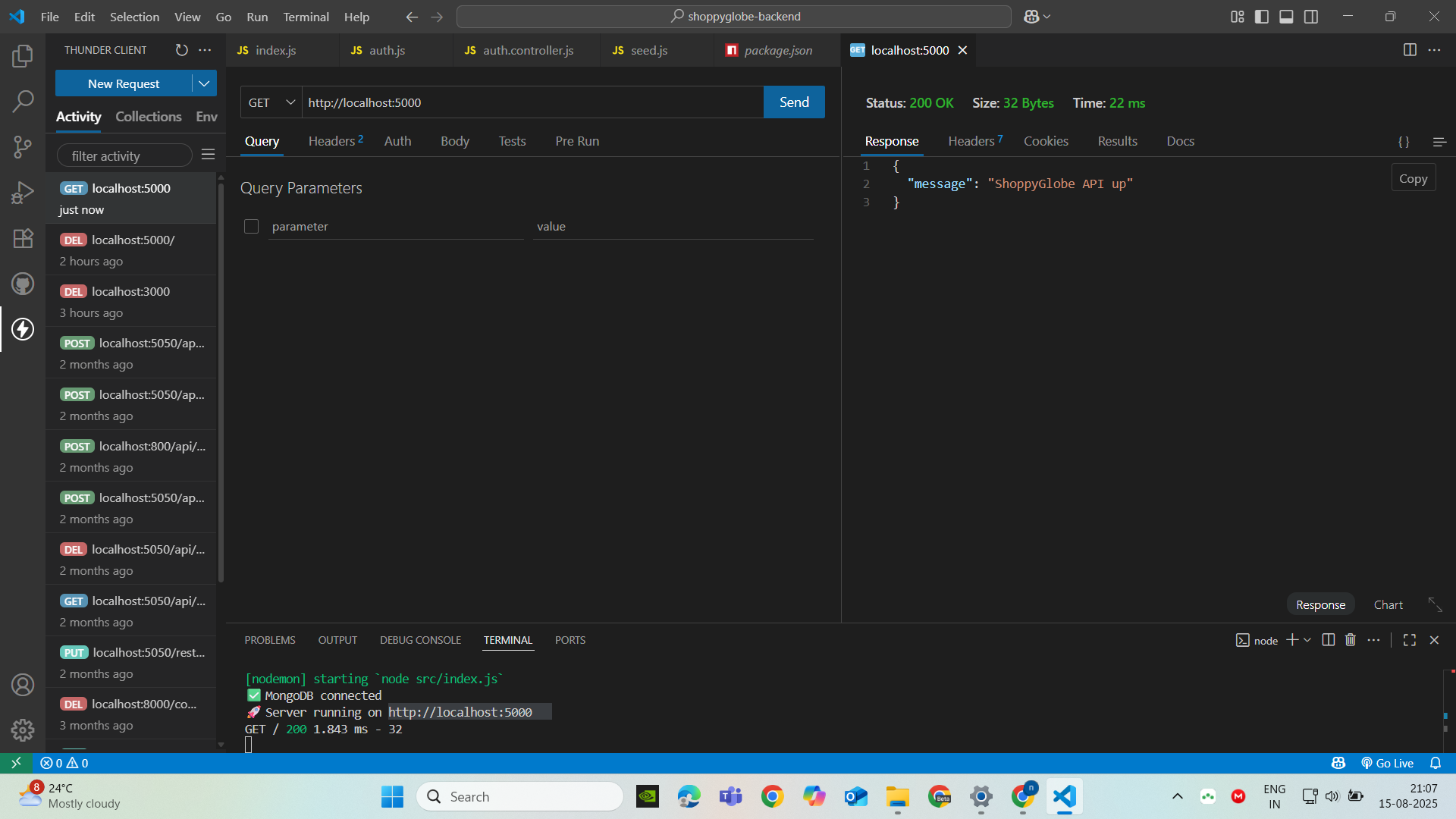
# 5) Global Behaviors

- Centralized error handling (404 + error middleware)  
- Validation: ObjectId checks; product existence; stock constraints; quantity >= 1  
- Auth middleware: expects Authorization: Bearer <token> (no quotes)

# 6) Thunder Client Test Flow & Screenshots

Follow this order. After each step, take a screenshot and paste it below.

## 6.1 Server Health — GET /

[PASTE SCREENSHOT HERE]

## 6.2 Register — POST /auth/register

A screenshot of a computer

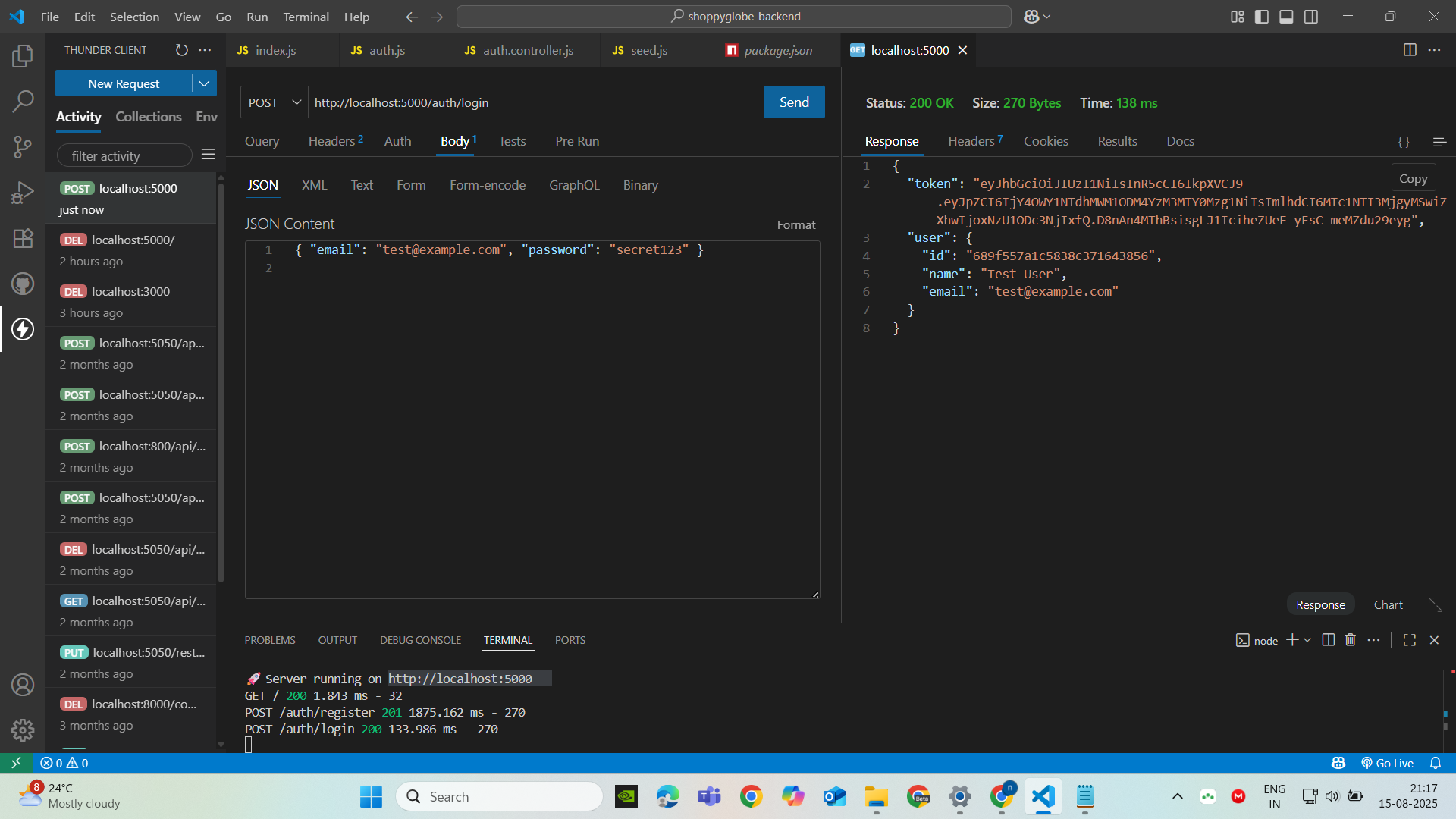
AI-generated content may be incorrect.

MongoDB registered User

A screenshot of a computer

AI-generated content may be incorrect.

## 6.3 Login — POST /auth/login



## 6.4 List Products — GET /products

A screenshot of a computer

AI-generated content may be incorrect.

## A screenshot of a computer AI-generated content may be incorrect.MongoDb- All Products

## 6.5 Get Product by Id — GET /products/:id

A screenshot of a computer

AI-generated content may be incorrect.

MongoDB - ProductById

## A screenshot of a computer AI-generated content may be incorrect.

## 6.6 Add To Cart — POST /cart

## 

ADDED CART – IN MONGODB

A screenshot of a computer

AI-generated content may be incorrect.

## 6.7 View My Cart — GET /cart

A screenshot of a computer

AI-generated content may be incorrect.

## Check Cart – MongoDB

A screenshot of a computer

AI-generated content may be incorrect.

## 6.8 Update Cart Item — PUT /cart/:id

## A screenshot of a computer AI-generated content may be incorrect.

## Updated Cart – In MongoDB

A screenshot of a computer

AI-generated content may be incorrect.

## 6.9 Delete Cart Item — DELETE /cart/:id

A screenshot of a computer

AI-generated content may be incorrect.

# Deleted Cart – In MongoDB

A screenshot of a computer

AI-generated content may be incorrect.

# 7) Negative Tests (Error Handling)

Take screenshots for these cases:

- GET /products/123 → 400 Invalid product id

A screenshot of a computer

AI-generated content may be incorrect.

- POST /cart without token → 401 No token provided

A screenshot of a computer

AI-generated content may be incorrect.

- POST /cart with non-existent product → 404 Product not found

A screenshot of a computer

AI-generated content may be incorrect.

- POST /cart with quantity exceeding stock → 400 Quantity exceeds available stock

A screenshot of a computer

AI-generated content may be incorrect.

# 8) MongoDB Atlas Screenshots

Screenshots of Atlas collections after seeding and cart/auth tests.

shoppyglobe.products

A screenshot of a computer

AI-generated content may be incorrect.

shoppyglobe.cartitems

A screenshot of a computer

AI-generated content may be incorrect.

Shoppyglobe.users

A screenshot of a computer

AI-generated content may be incorrect.

# 9) Code References

- src/index.js — server setup, routes, DB connect (/shoppyglobe)  
- src/models/ — Product, User, CartItem schemas  
- src/routes/ — products (GET only), auth, cart  
- src/controllers/ — logic for each route  
- src/middleware/auth.js — JWT verification  
- src/middleware/errorHandler.js — error handling  
- src/seed/seed.js — seeds products

# 10) Submission Checklist

[ Yes] GitHub repo with code + README  
[ Yes] Thunder Client screenshots (section 6 & 7)  
[ Yes] MongoDB Atlas screenshots (section 8)  
[ Yes] App runs: npm run dev shows MongoDB connected + server URL