Project Documentation: ShoppyGlobe – Backend API using Node.js & Express

1. Project Overview

This project is a complete **E-commerce Backend API** built using **Node.js, Express, and MongoDB Atlas**, with **JWT authentication and Cart functionality**.

With this API, we can:

- Register & Login Users securely
- Fetch products from database
- Add products to cart
- Update quantity in cart
- Remove product from cart
- Get user's entire cart

All cart routes are **protected and require JWT authentication**.

Smart features used:

- Password encryption (bcrypt)
- Token-based authentication (JWT)
- Error-handling middleware
- MongoDB Atlas cloud database

2. Middleware Overview

a) Authentication Middleware

This middleware protects cart routes. It verifies the token sent in the request header.

✔ If the token is valid, it allows the request to proceed
✕ If invalid/missing, it returns **401 Unauthorized**

Why useful?

- Ensures only logged-in users can add/view cart items.
- Prevents unauthorized access to private data.

b) Error Handling Middleware

There are two error handlers:

notFoundHandler

Used when a route does not exist.

"Route /unknown not found"

globalErrorHandler

Catches internal errors and prevents application crash.

Why is it important?

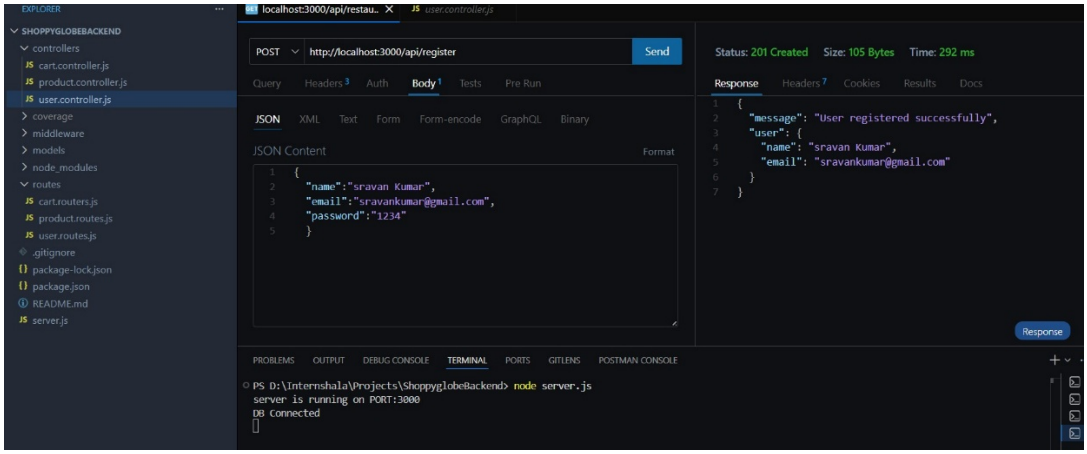
- Returns clean error messages
- Sends proper status codes like 500, 404, 400

3. API Routes Explanation

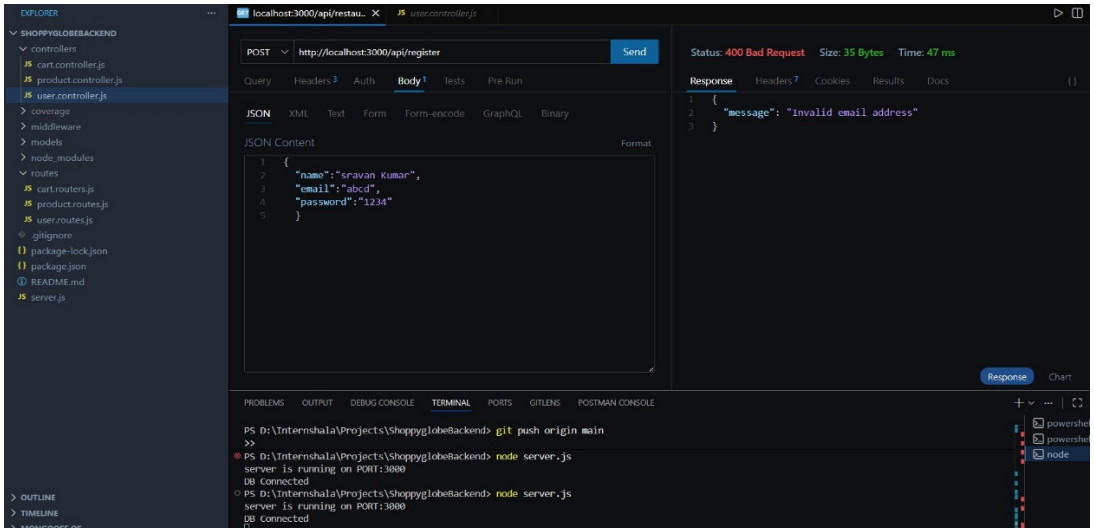
User Authentication Routes

POST /api/register — Register a New User

✔ Validates name, email, password

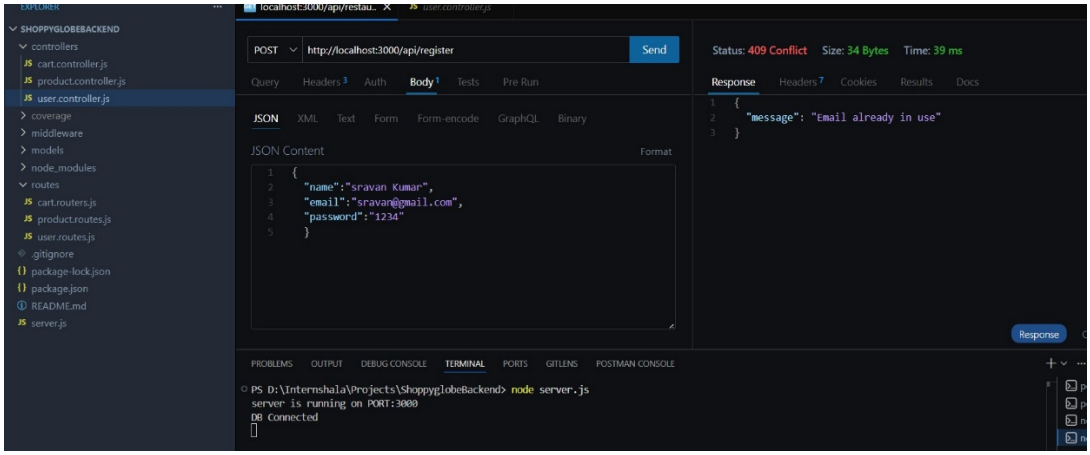


✔ Validates proper email format



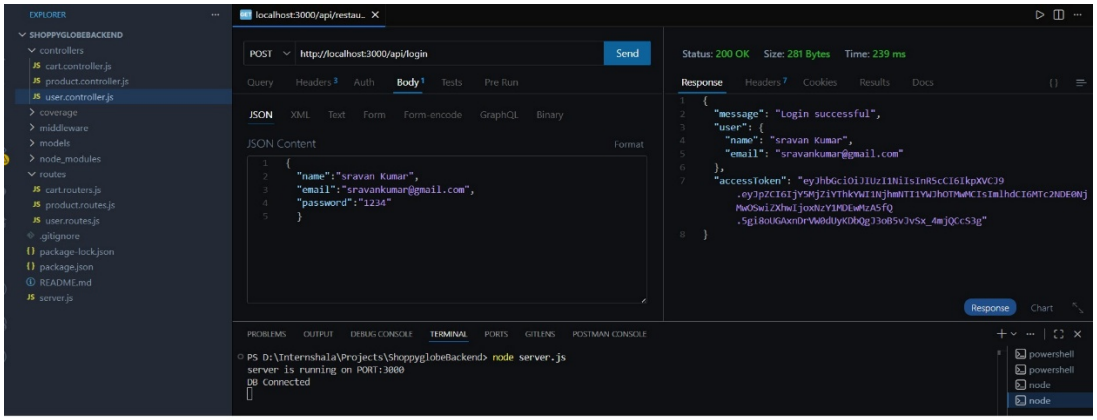
✔ Hashes password before saving

✕ If email exists → 409 Conflict

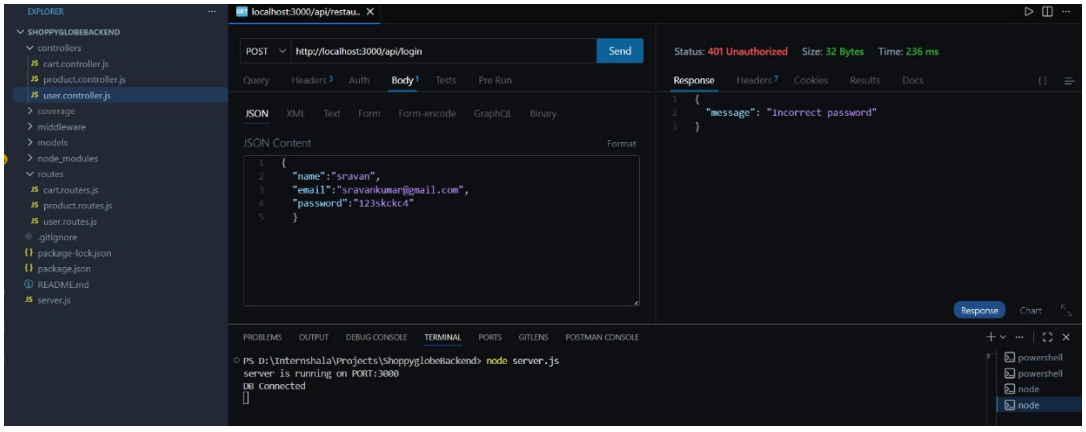


POST /api/login — Authenticate User

✔ Validates email & password
✔ Returns a JWT token on success



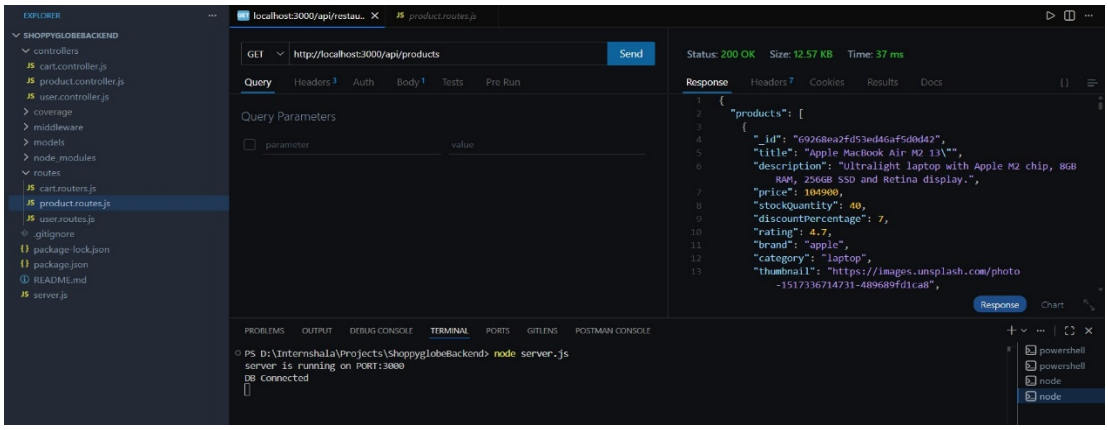
✕ If wrong password → 401 Unauthorized



Product Routes

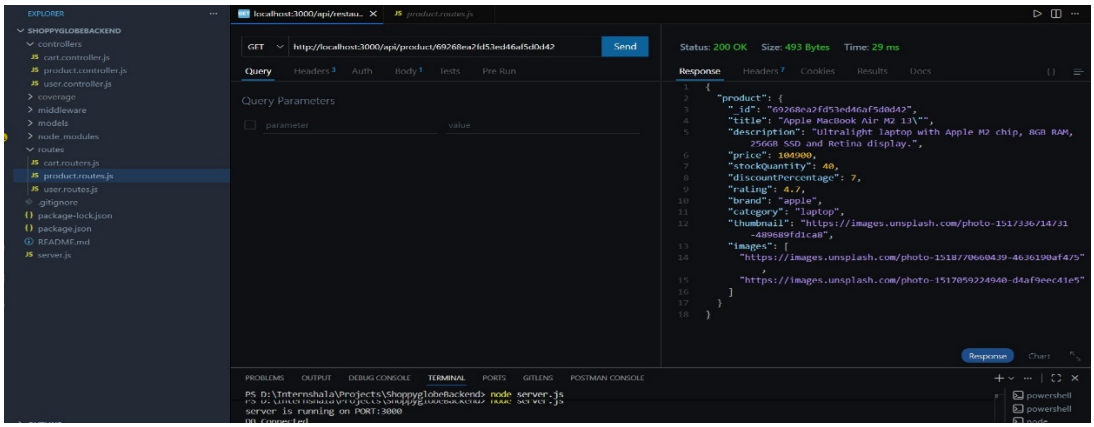
GET /api/products — Fetch All Products

✔ Returns list of products stored in MongoDB Atlas
✔ Data was inserted directly in Atlas (not through API)

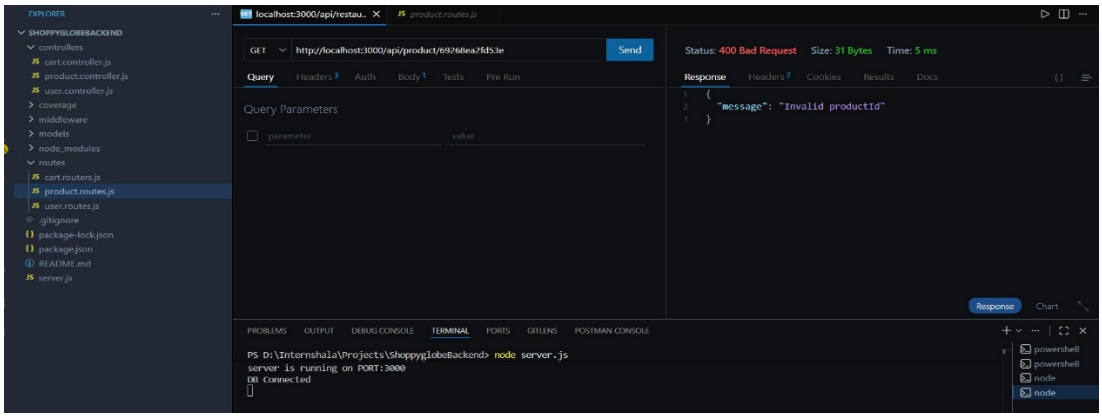


GET /api/product/:id — Fetch Product by ID

✔ Fetches single product details using MongoDB _id



✕ If ID not valid → 404 Not Found

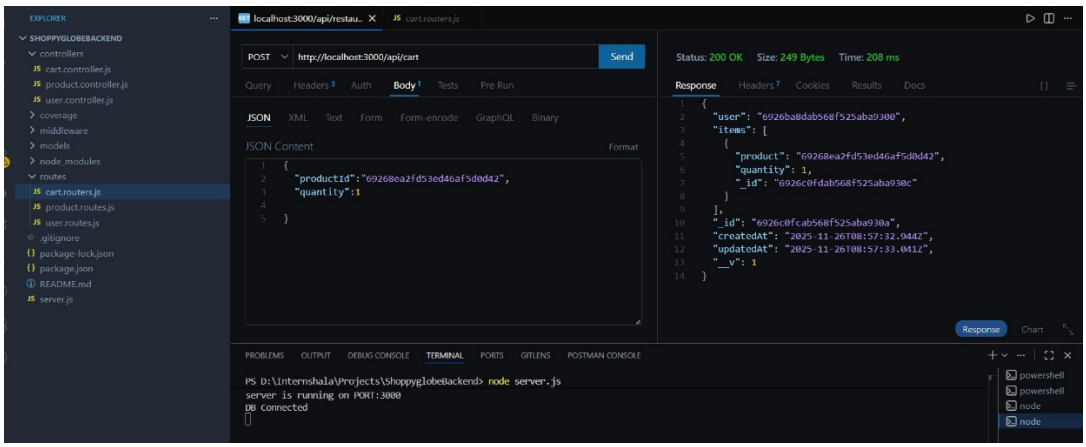


Cart Routes (Protected with Token)

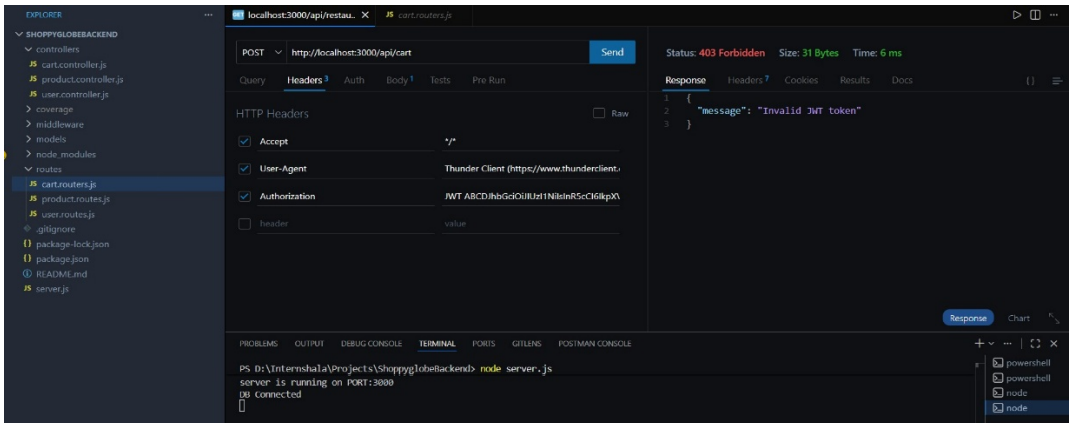
Note:- Cart API cannot be tested without Backend Token (Login Needed)

POST /api/cart — Add Item to Cart

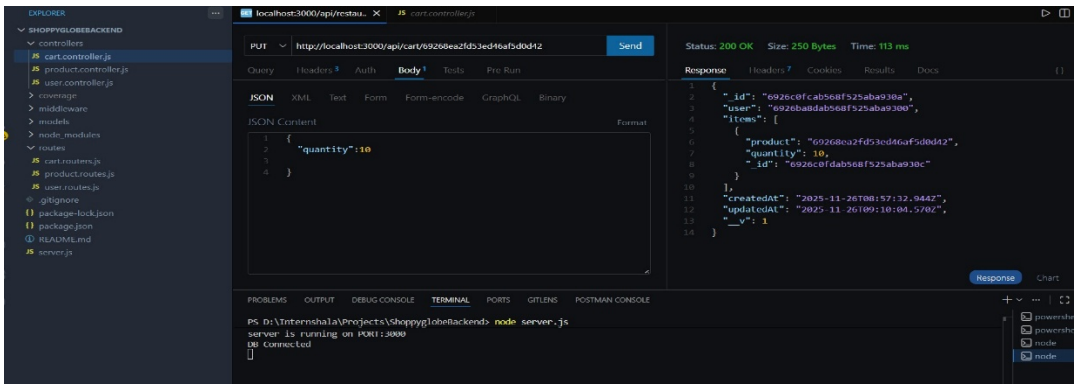
✔ Creates new cart if user has no cart
✔ Adds product + quantity



✕ JWT not valid → 403 Forbidden

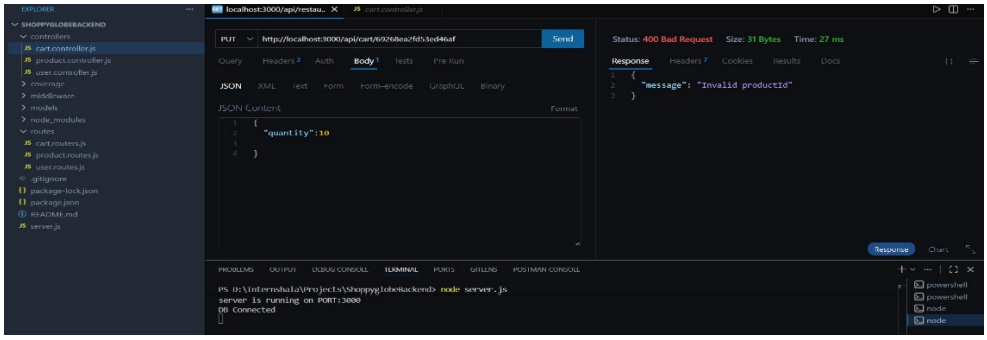


✔ If product exists → updates quantity instead of duplicating

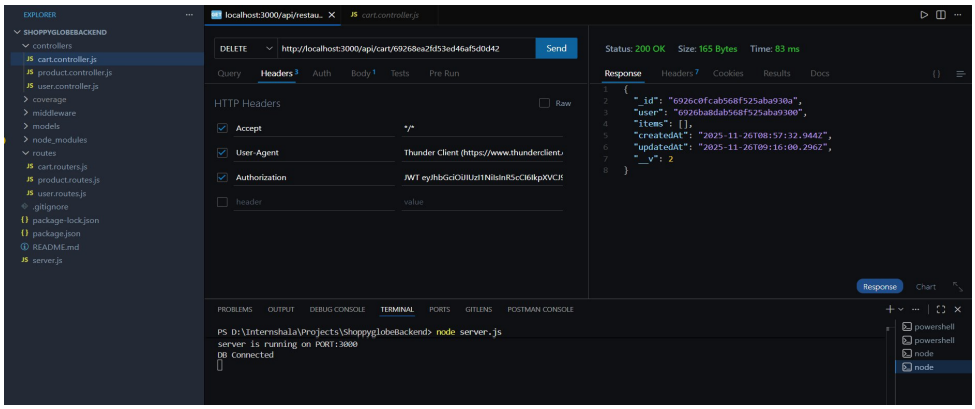


PUT /api/cart — Update Product Quantity

✔ Modifies quantity for a specific product
✕ If product not found → 404



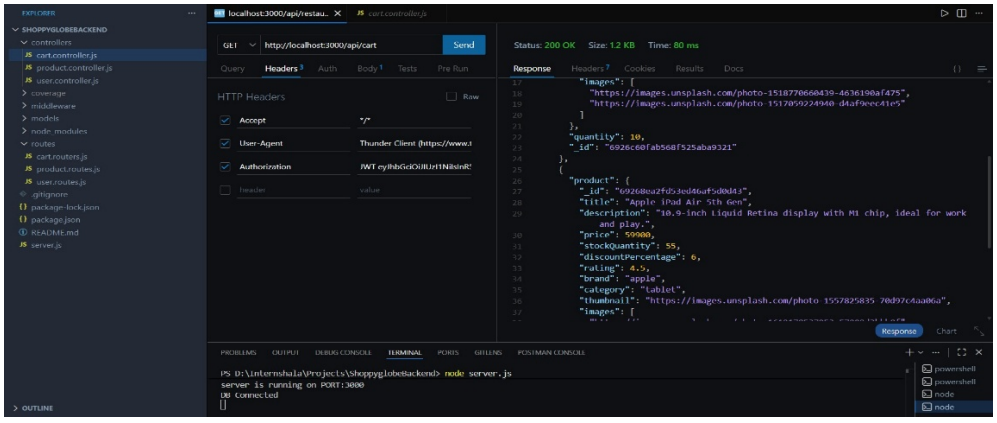
DELETE /api/cart — Remove Product from Cart



✔ Removes product from user's cart

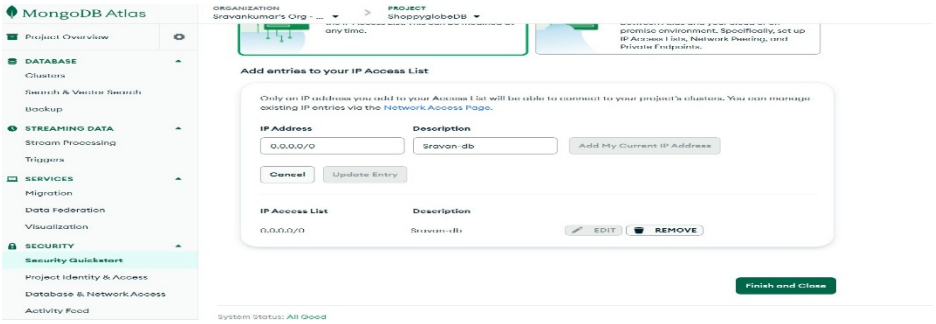
GET /api/cart — Get User Cart with Product Details

✔ Returns products with populated details (name, price, img…)
✔ Requires Authorization Token



4. MongoDB Atlas Setup Summary

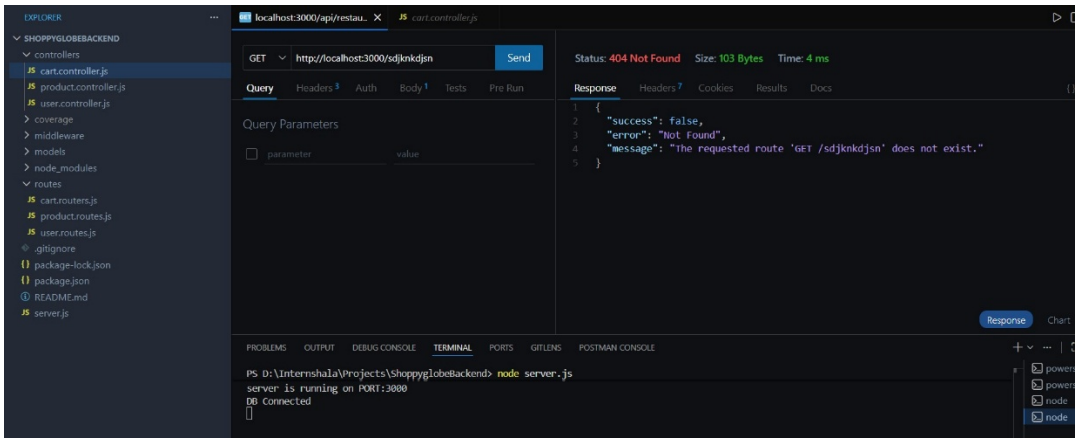
✔ Used **cloud database** instead of local MongoDB
✔ Allowed public access using IP 0.0.0.0/0



✔ Created **non-admin DB user** for safety

5. Error Handling Summary

✕ If path not found → 404



- No raw system errors are sent to client
- Proper status codes returned:
    - 400 (Bad Request)
    - 401 (Unauthorized)
    - 404 (Not Found)
    - 500 (Internal Server Error)

✔ Server never crashes even if an error occurs

Final Result

successfully developed:

Feature	**Status**
User Auth + JWT	✔
Password Security (bcrypt)	✔

Feature	Status
Product API + MongoDB	✔
Protected Cart System	✔
Error handlers	✔
Cloud DB + Testing	✔

ShoppyGlobe Backend is fully secure, scalable, and production-ready!

Top of Form

Bottom of Form

Top of Form

Bottom of Form