INTERNSHALA TRAININGS

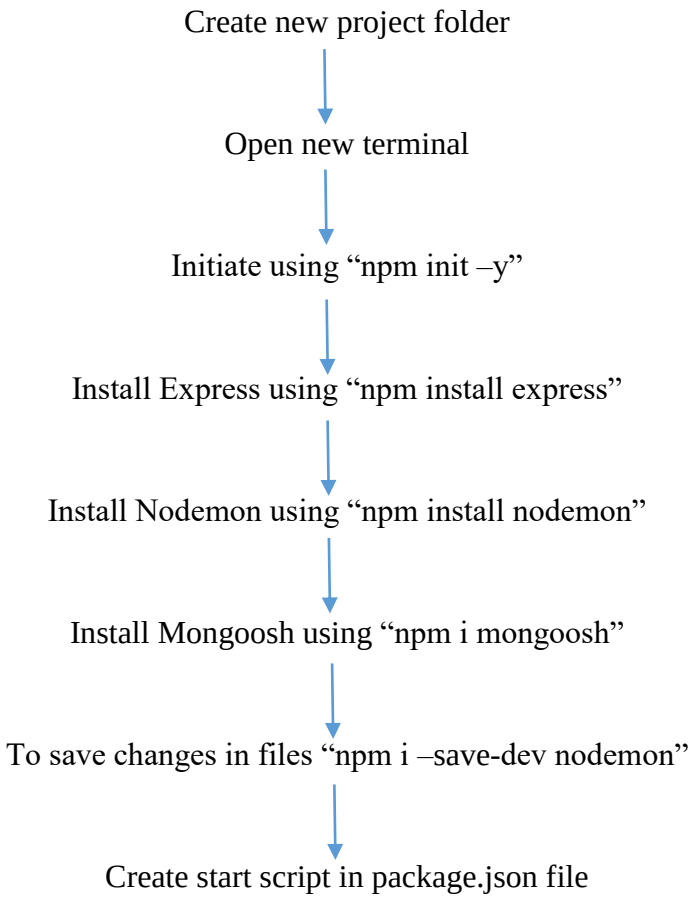
Project

Build APIs with Node.js and Express.js for Shoppyglobe E-commerce

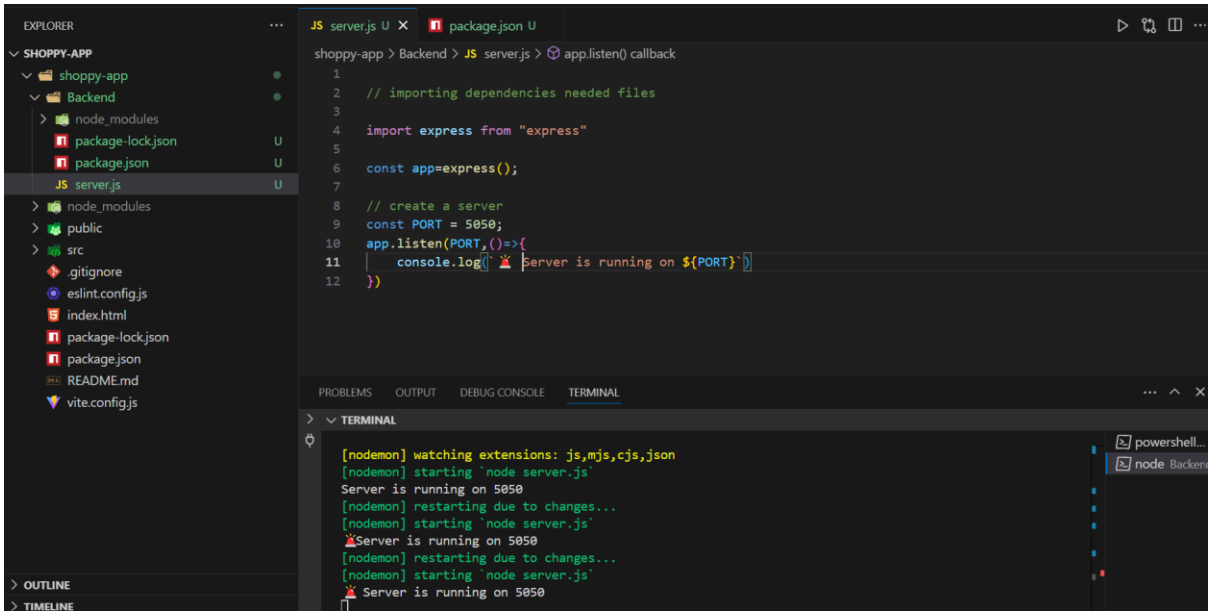
Objective:

Create the backend for the ShoppyGlobe application using Node.js, Express, and MongoDB.

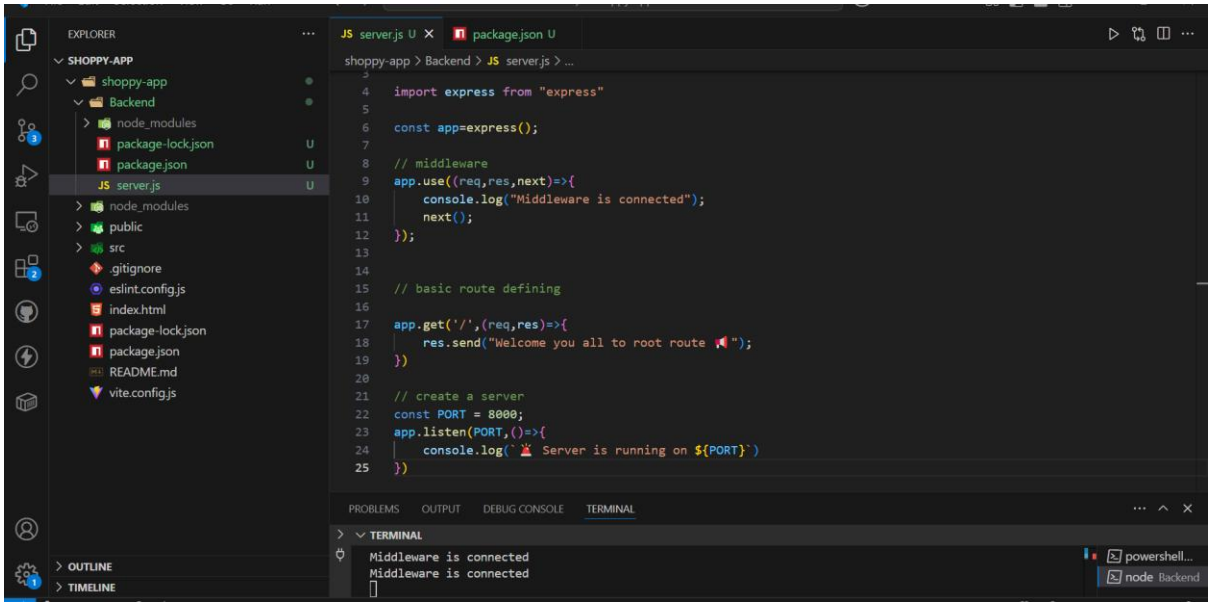
Project setup:



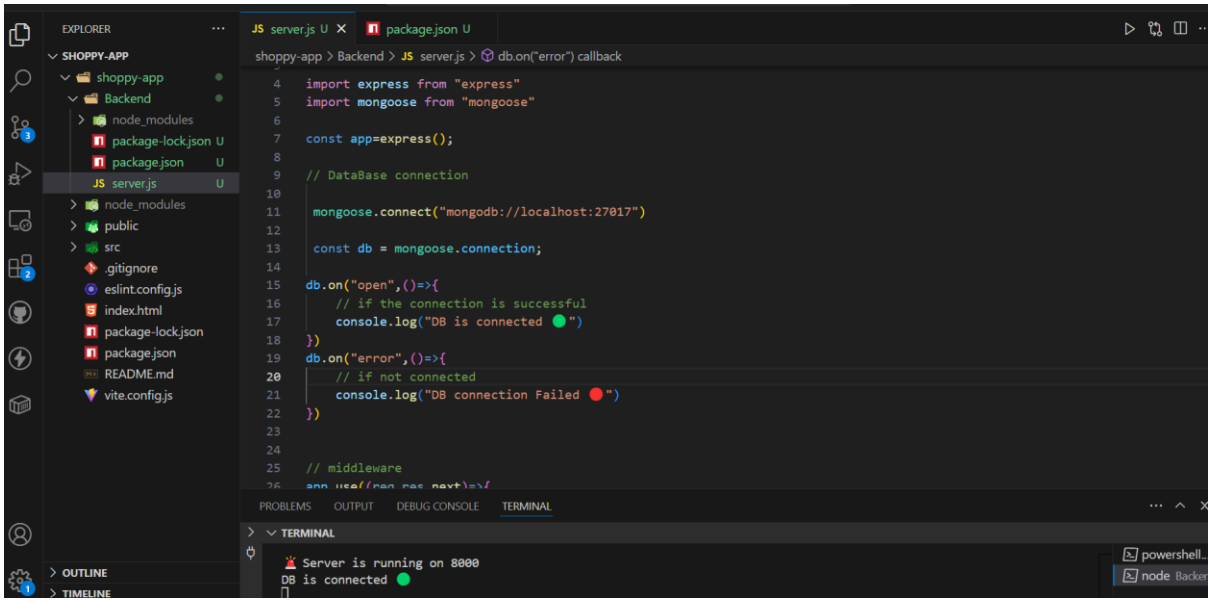
Create a server:



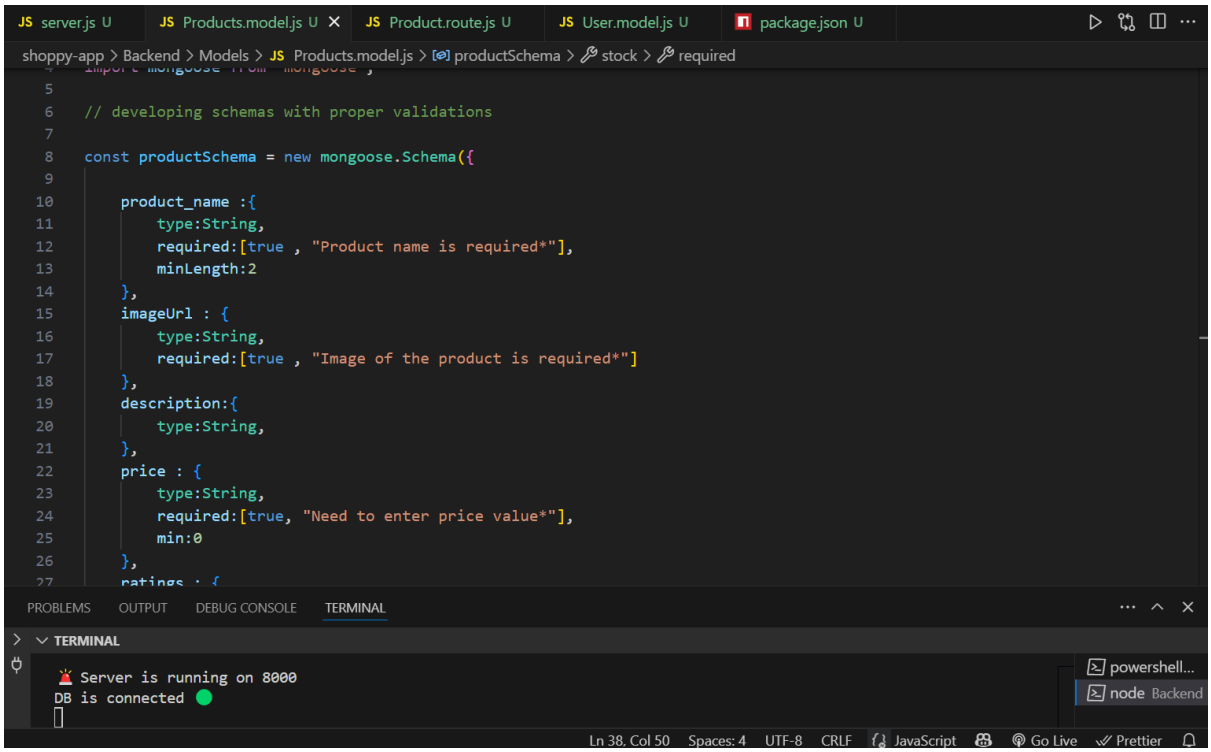
To connect middleware and initial route:



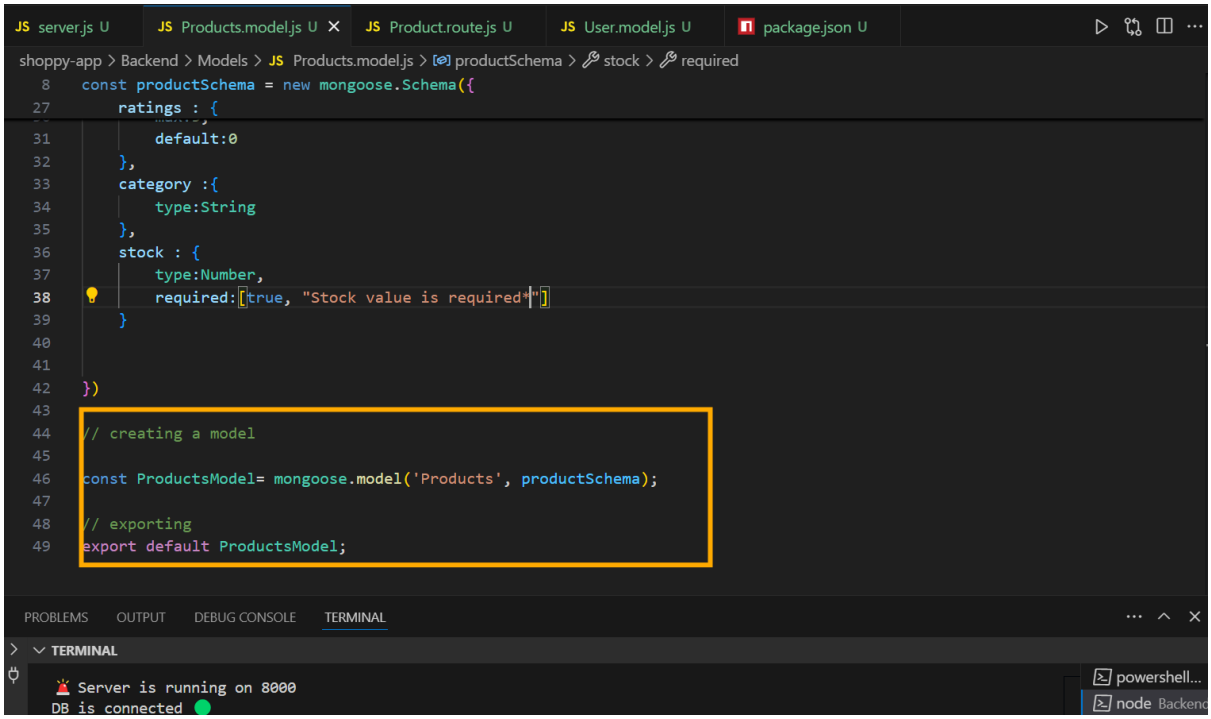
Connect with MongoDB:



Created a model for products and defining schemas with proper validations



Exported products model



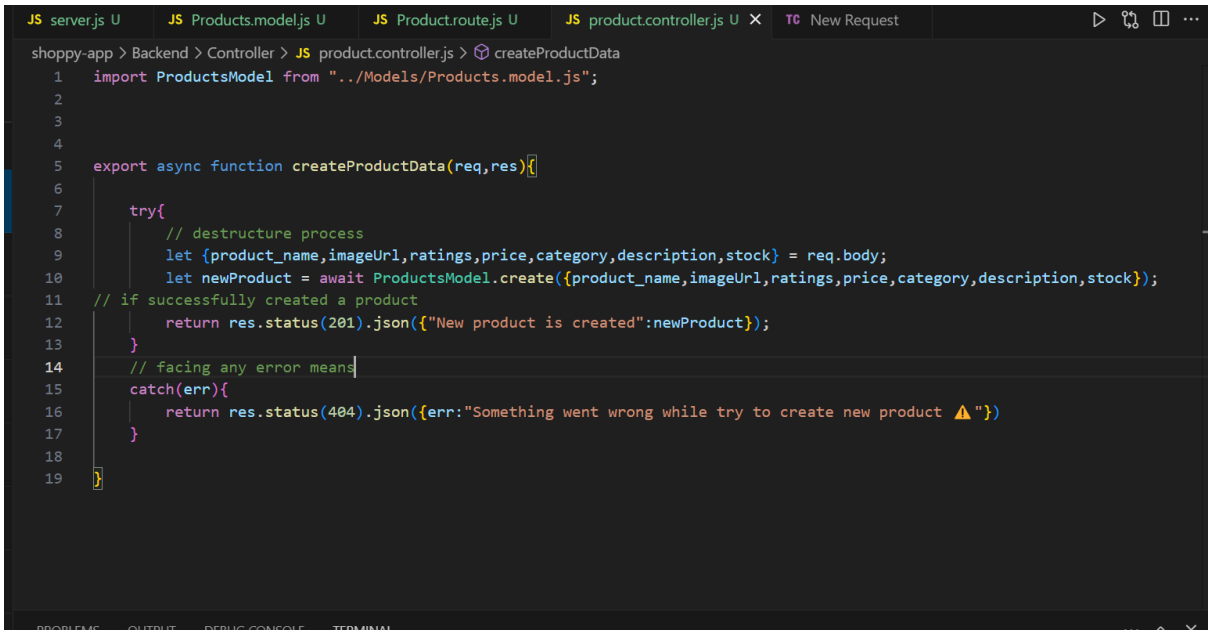
CRUD operations for products

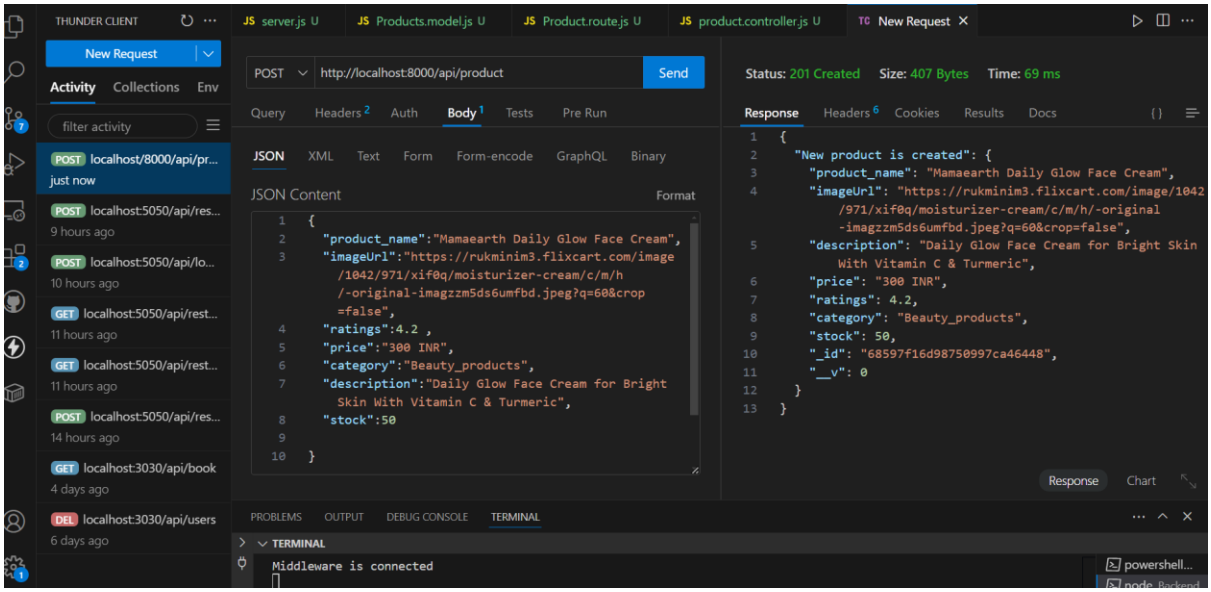
Create new product

1)I have defined the route to create new product – app.post("/api/product",createProductData)

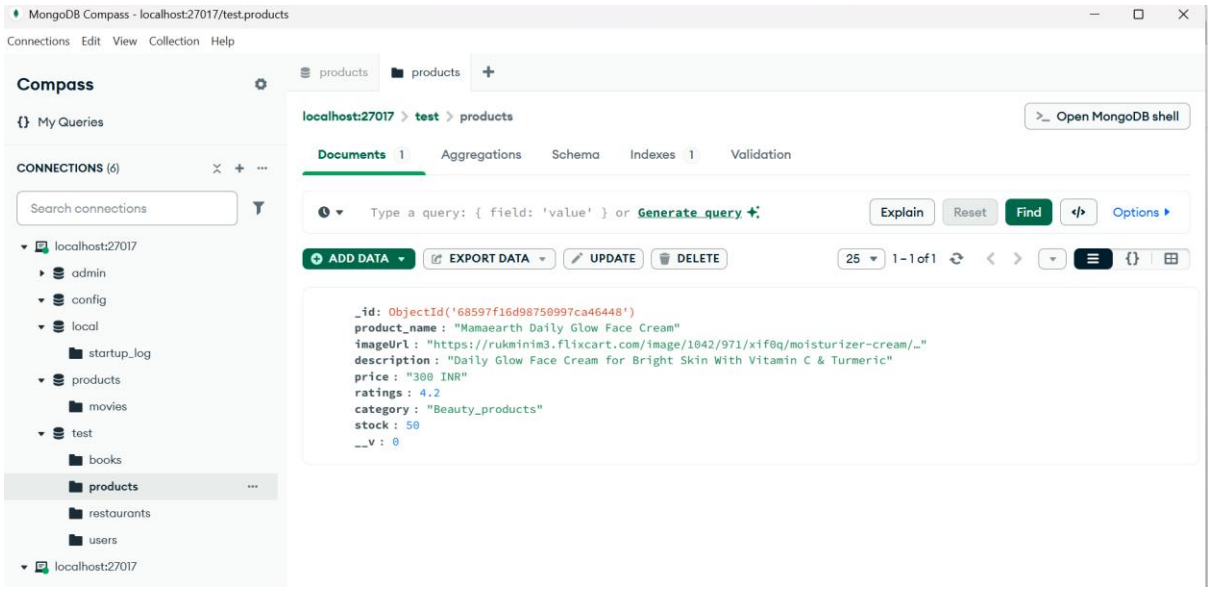
2) I have written logic in controller part

3) Tested with Thunder-Client extension



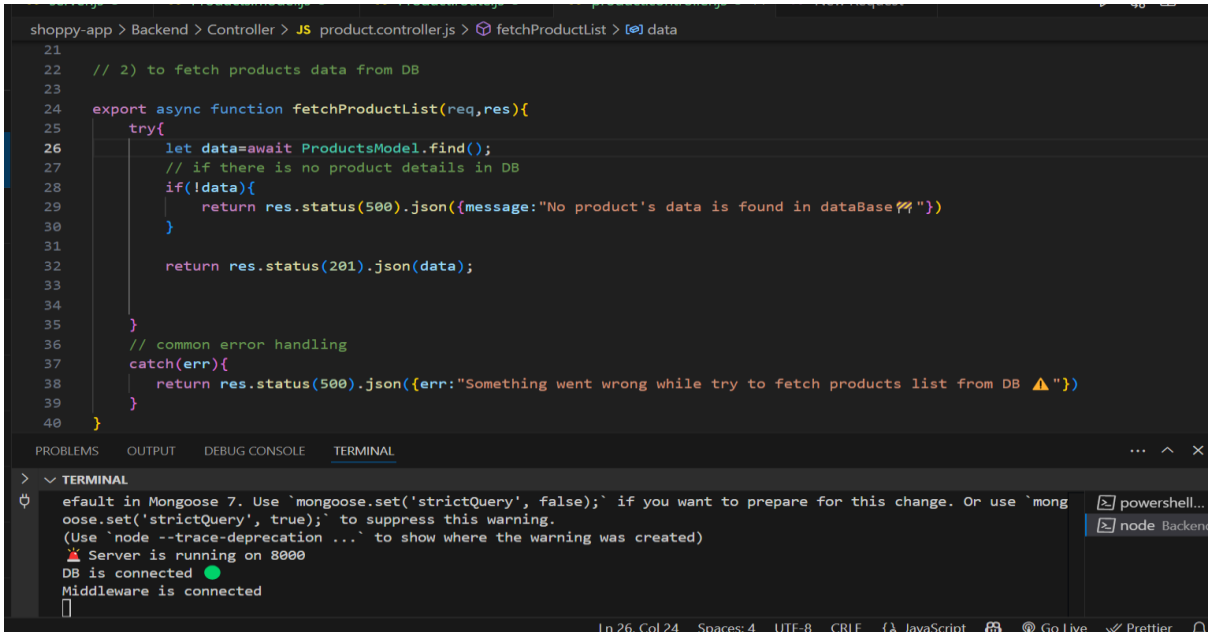


Response in MongoDB compass

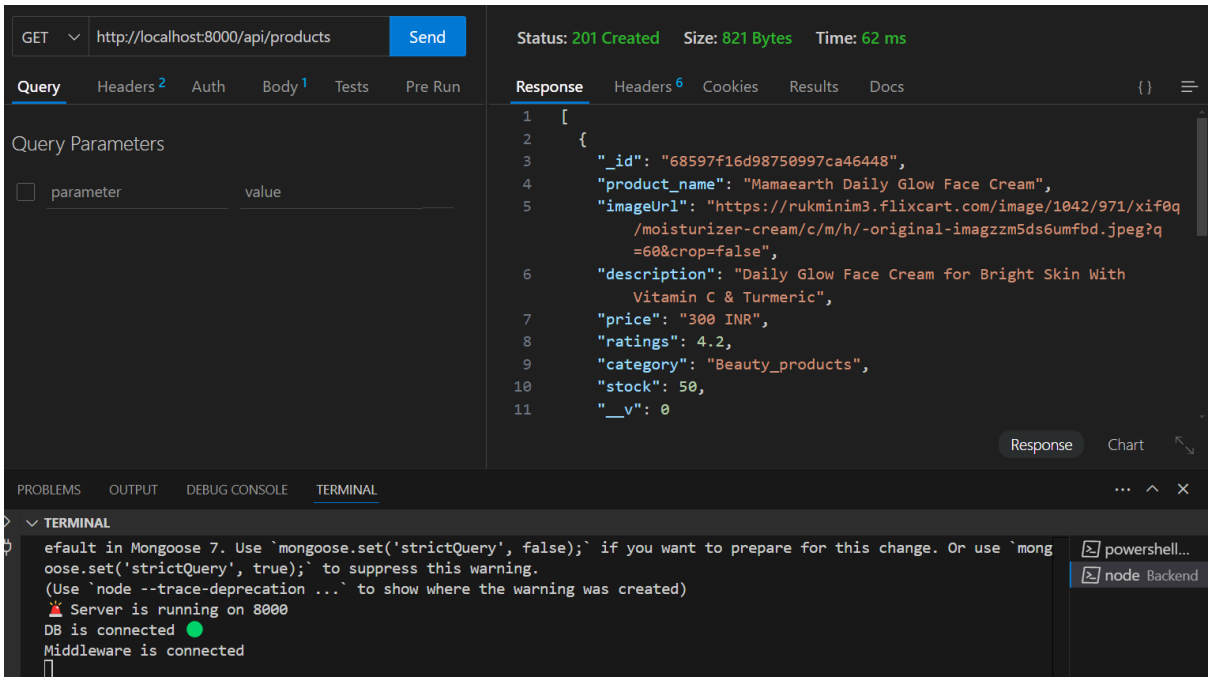


Fetch the list of products

- Created route for fetching data – “app.get(“/api/products”,fetchProductList)
- Wrote logic in controller part
- Check with Thunder-client extension

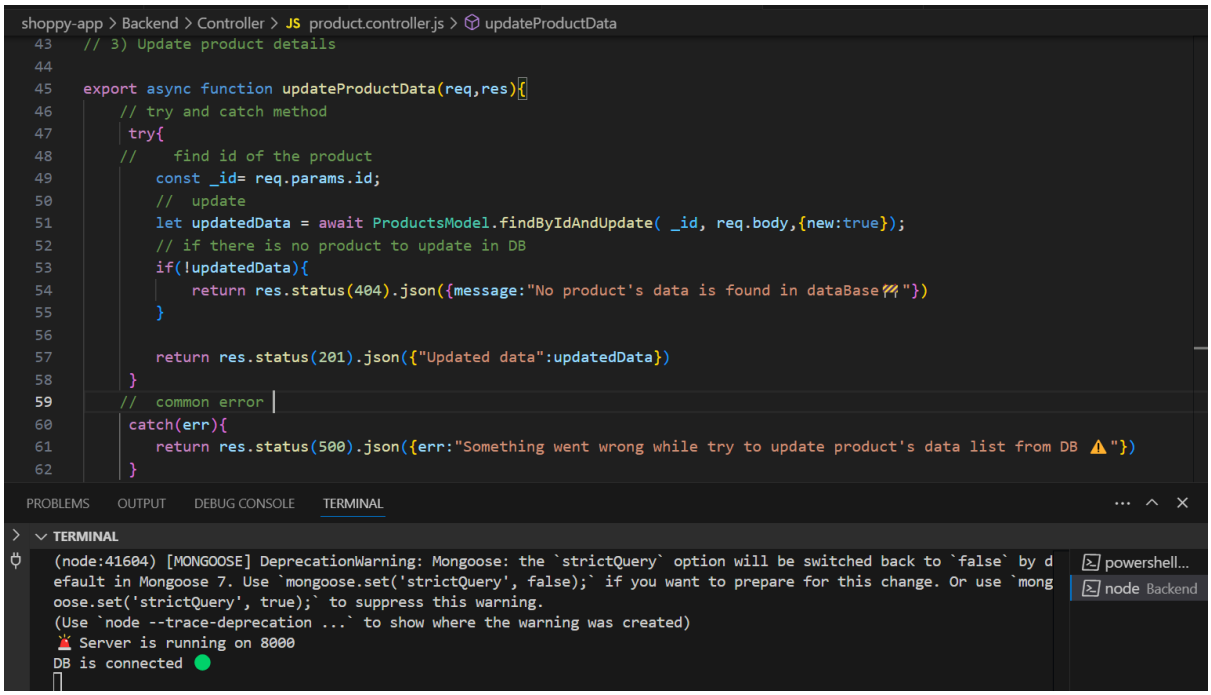


Thunder client response

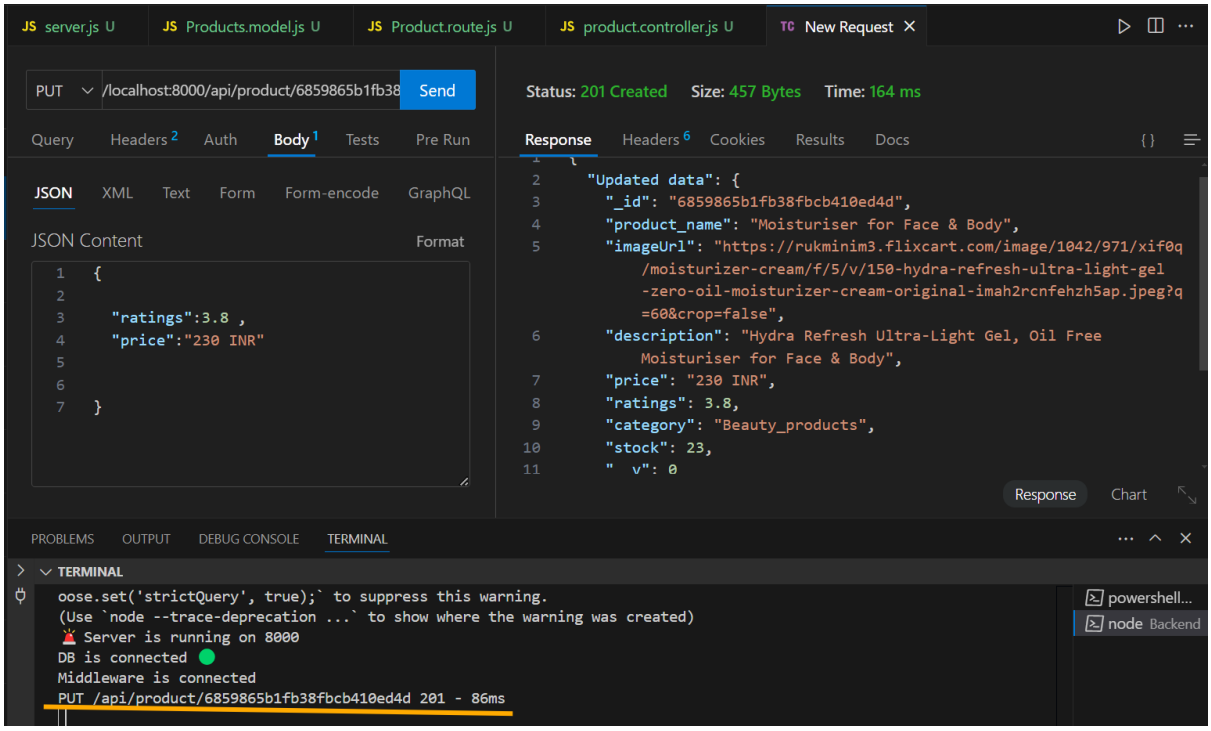


Update Data

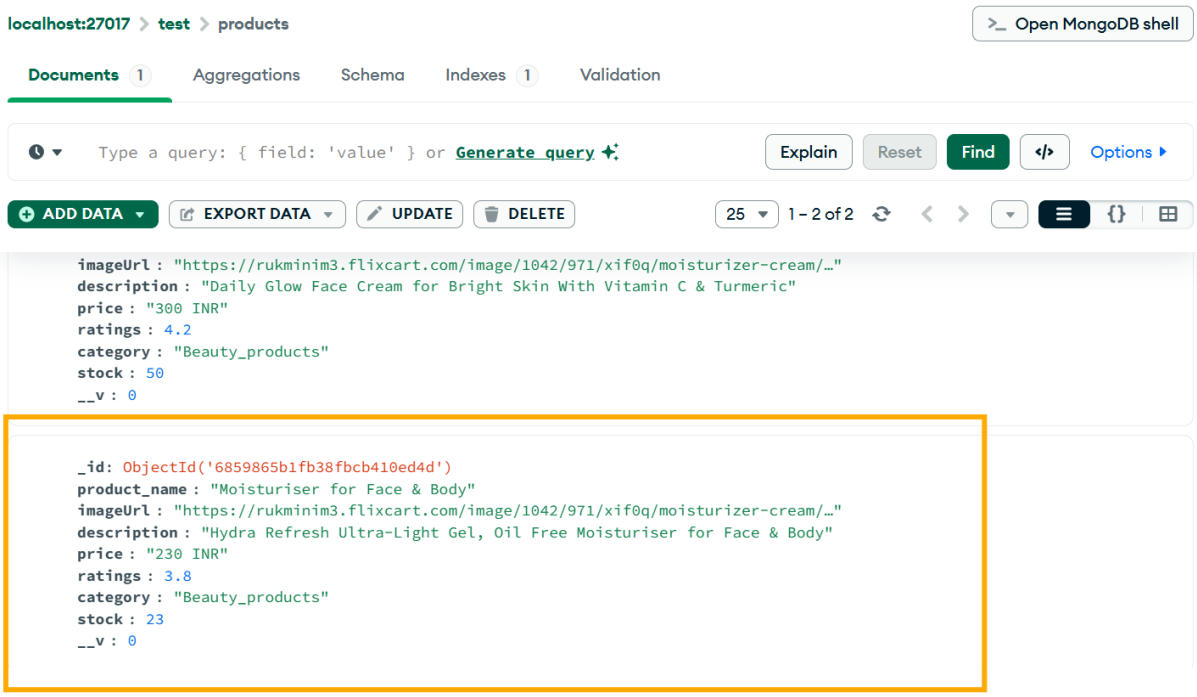
- Defining route to update from registered data – "/api/product/:id",updateProductData
- I have written a logic in controller part
- Check using thunder-client extension



Response in Thunder-client

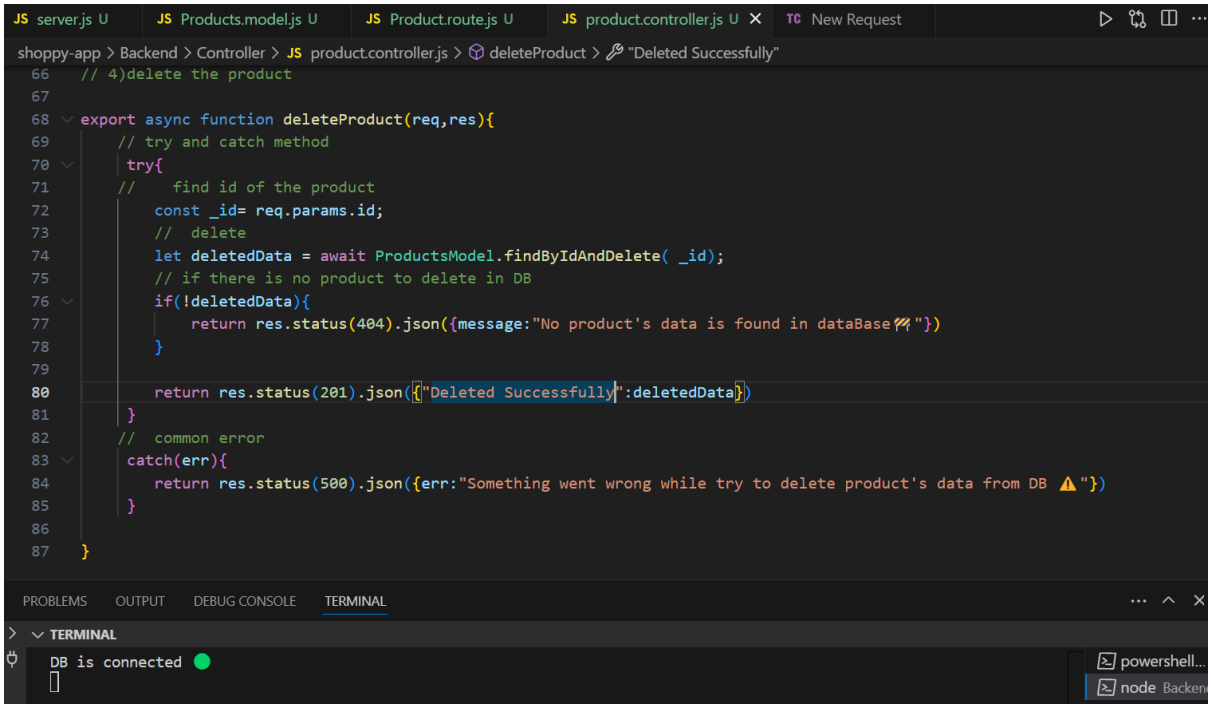


Response in MongoDB compass

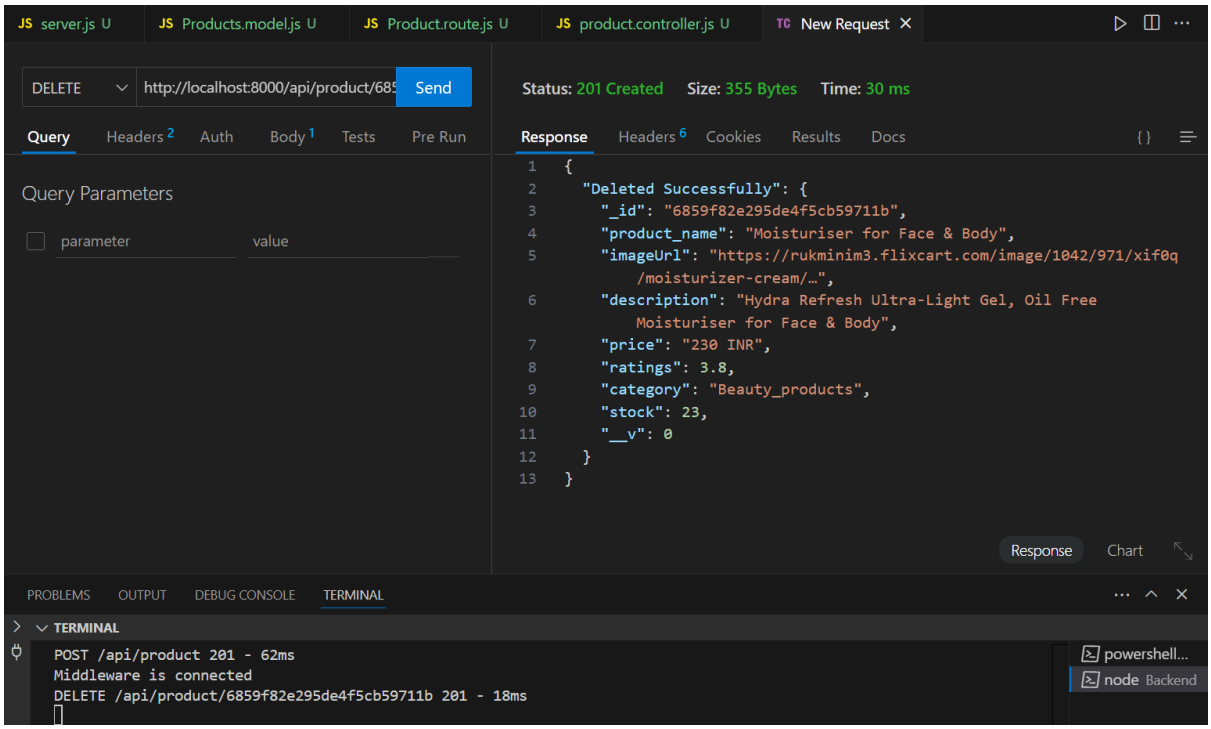


Delete the Product

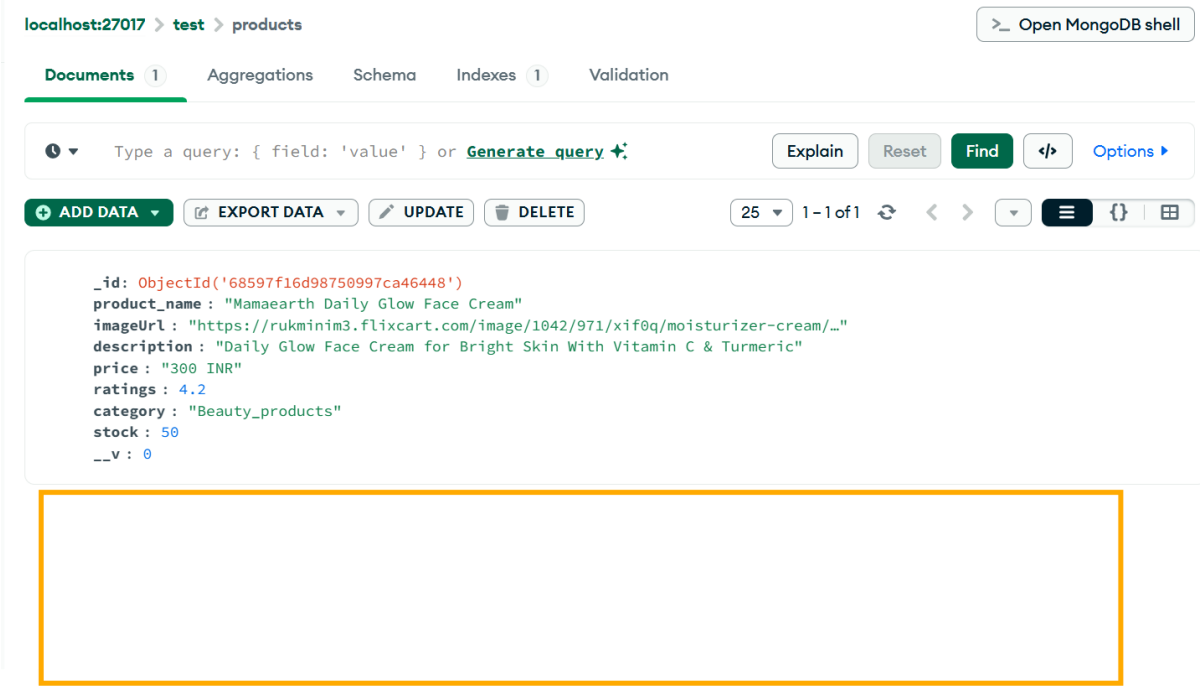
- Created the route to perform delete the product – "/api/product/:id:,deleteProduct
- Defining the logic in controller part
- Check with thunder-client



Thunder - client response

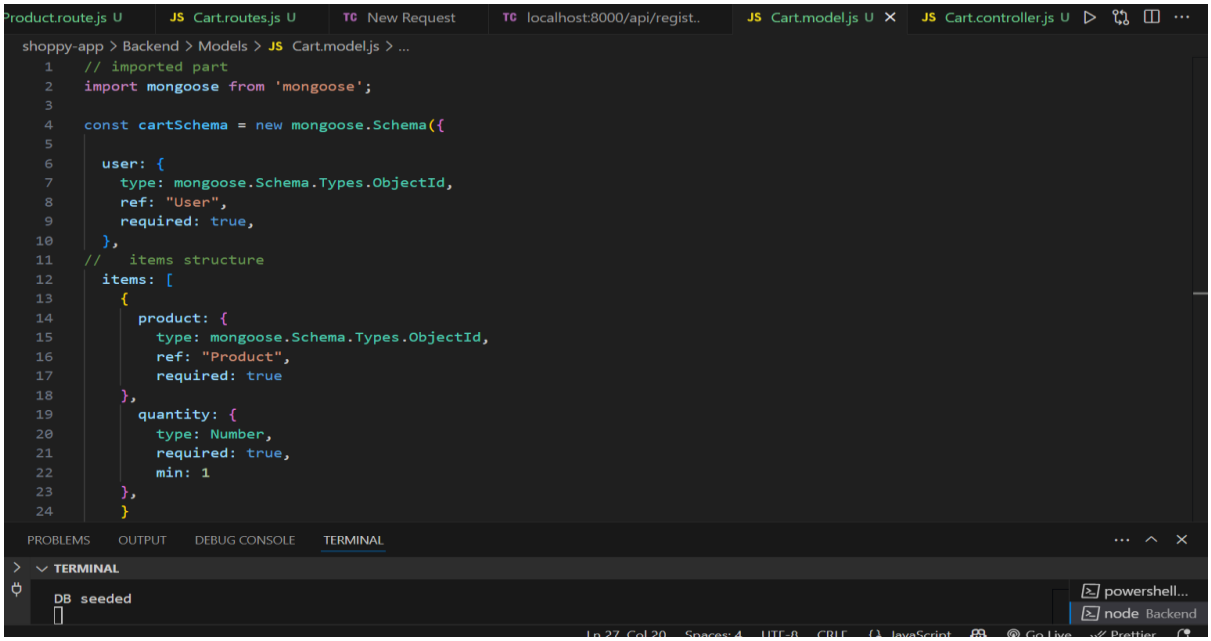


MongoDB compass



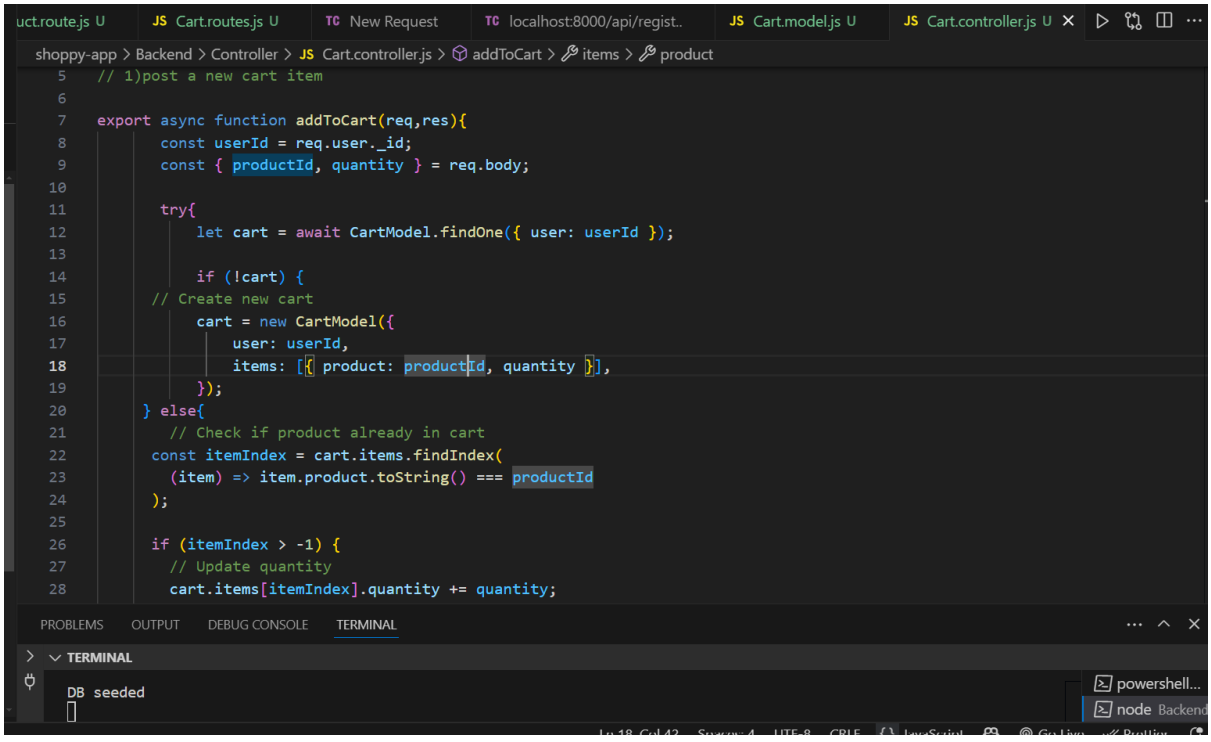
CRUD operations for Cart

Defined schemas and model for cart

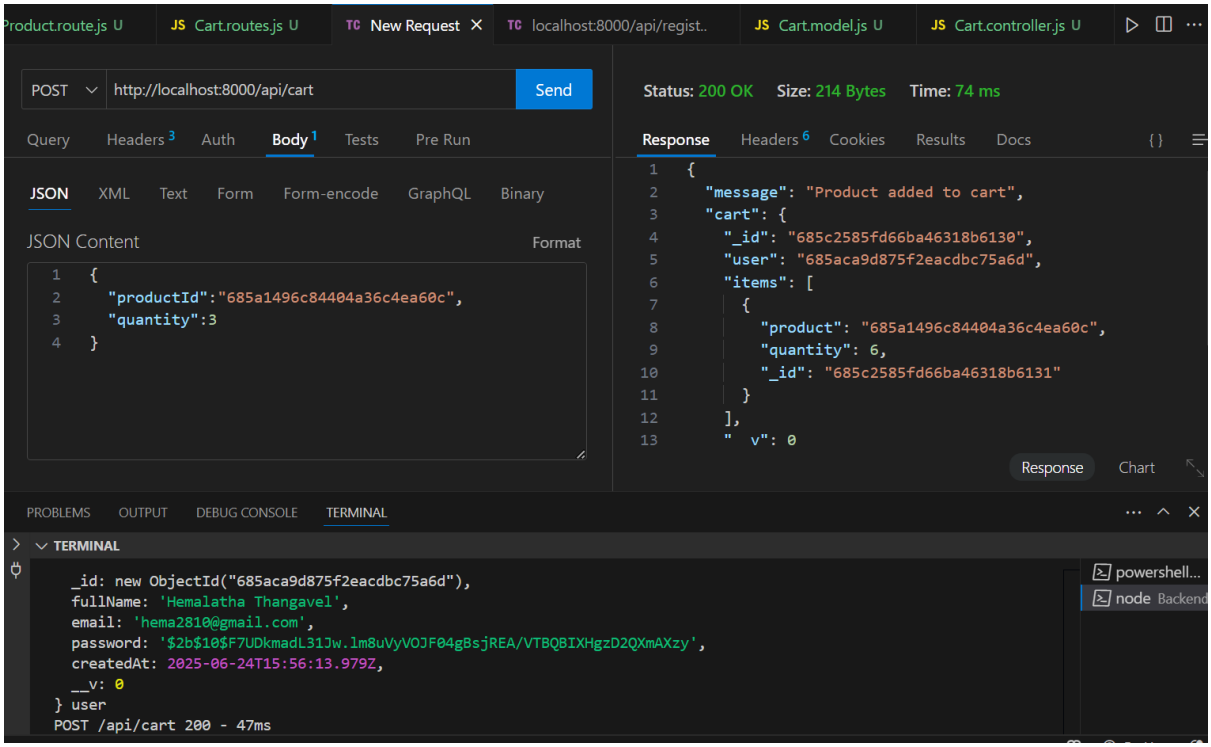


Create new cart item

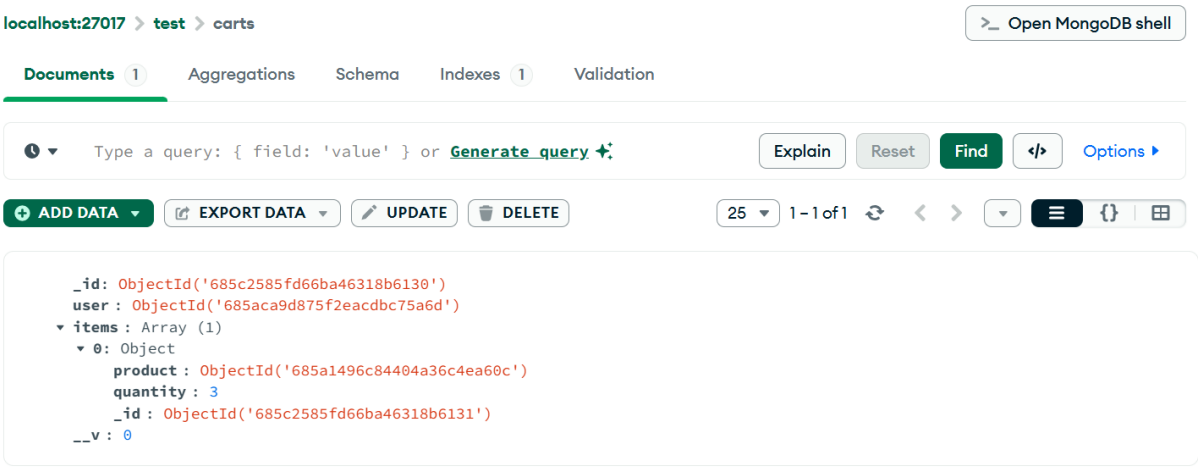
- I have defined the route to create new cart item –"/api/cart",createCartItem
- Created schema and model for cart items
- Defined the logic in controller part
- Check with thunder-client



Response in Thunder-client

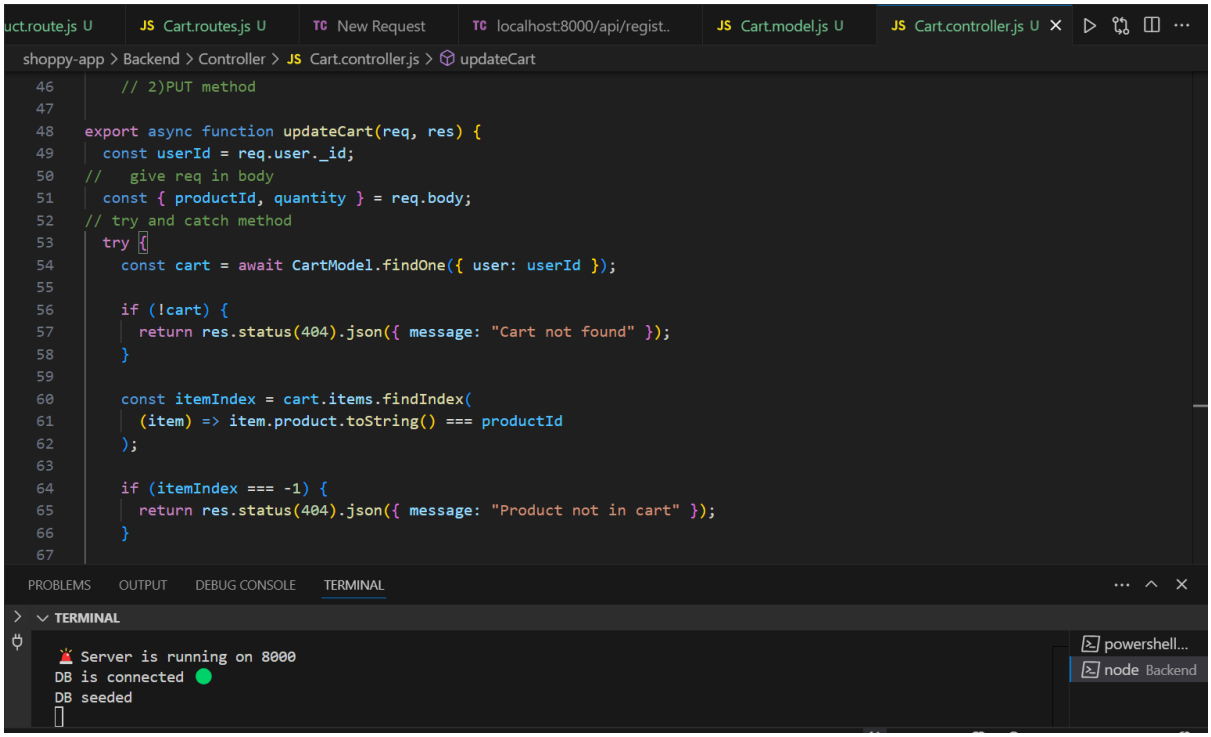


Response in MongoDB compass

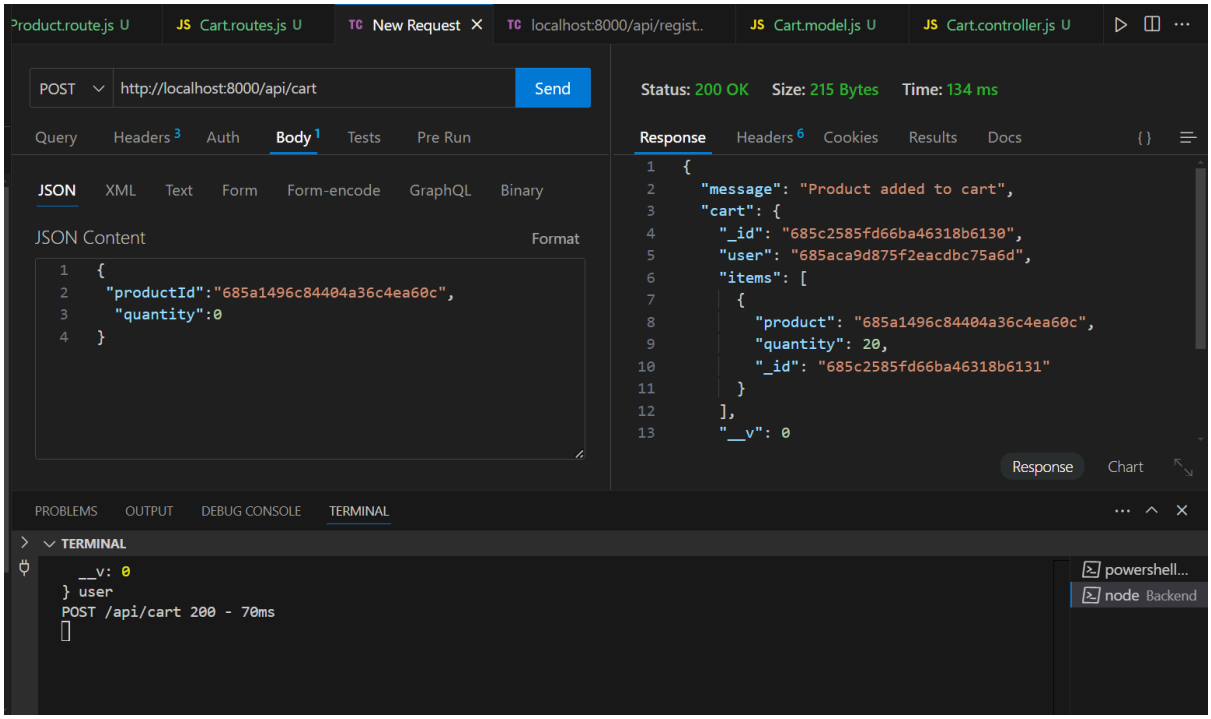


PUT/cart/

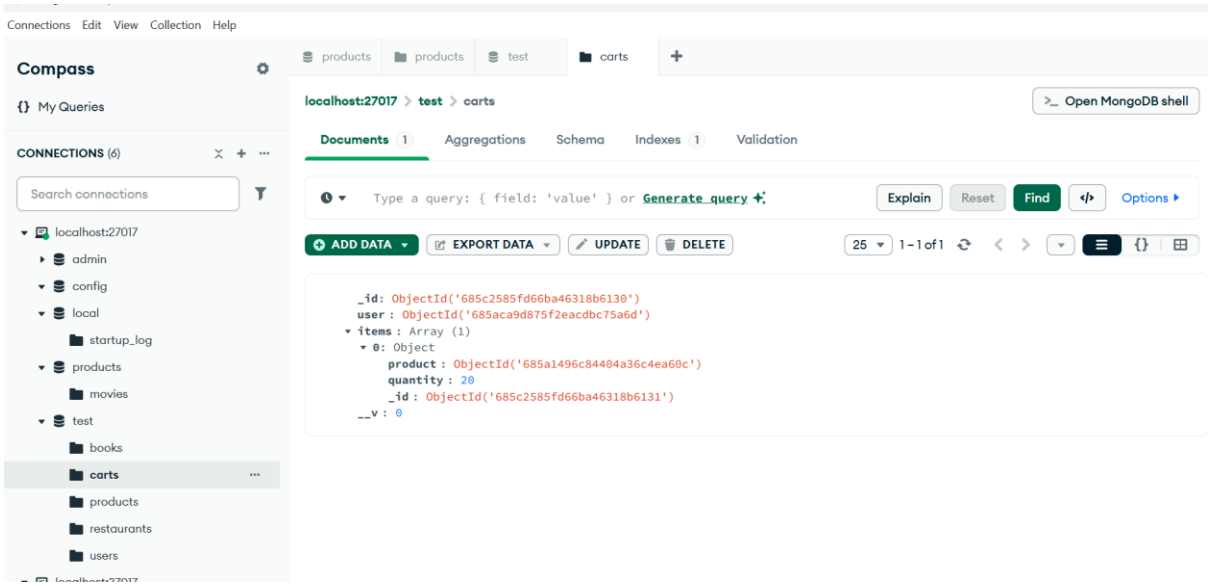
Update the quantity of a product in the cart



Response in thunder-client

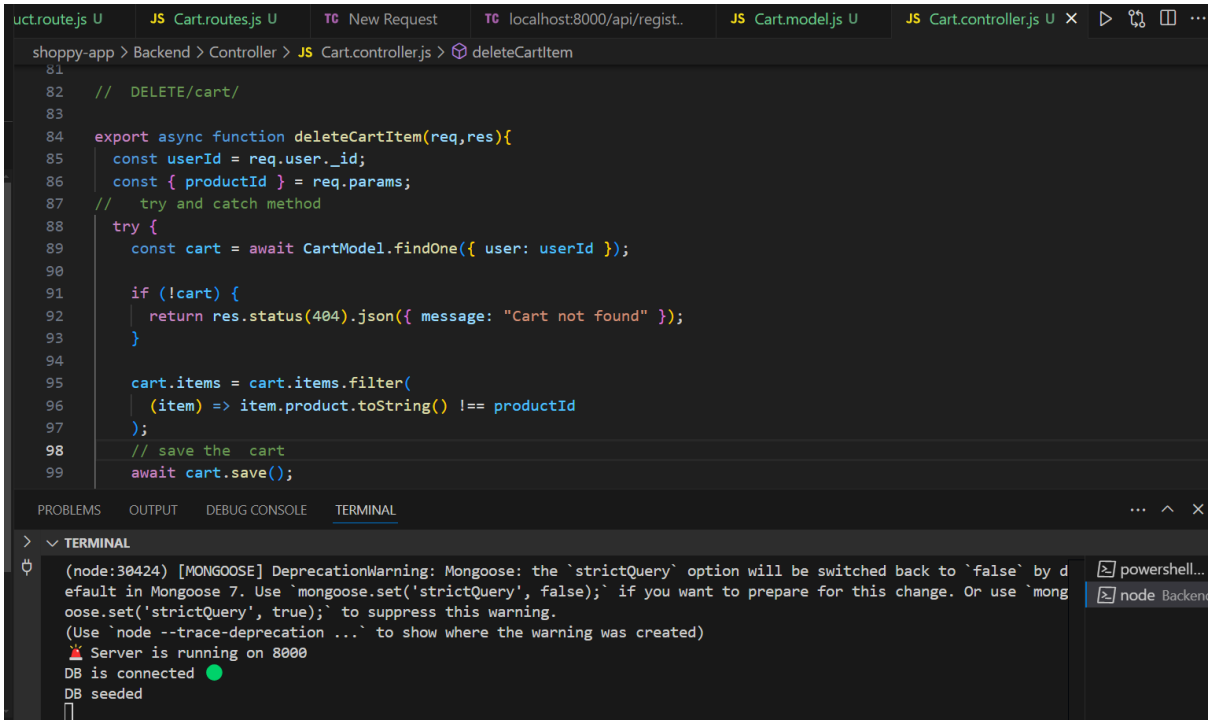


Response in MongoDB compass

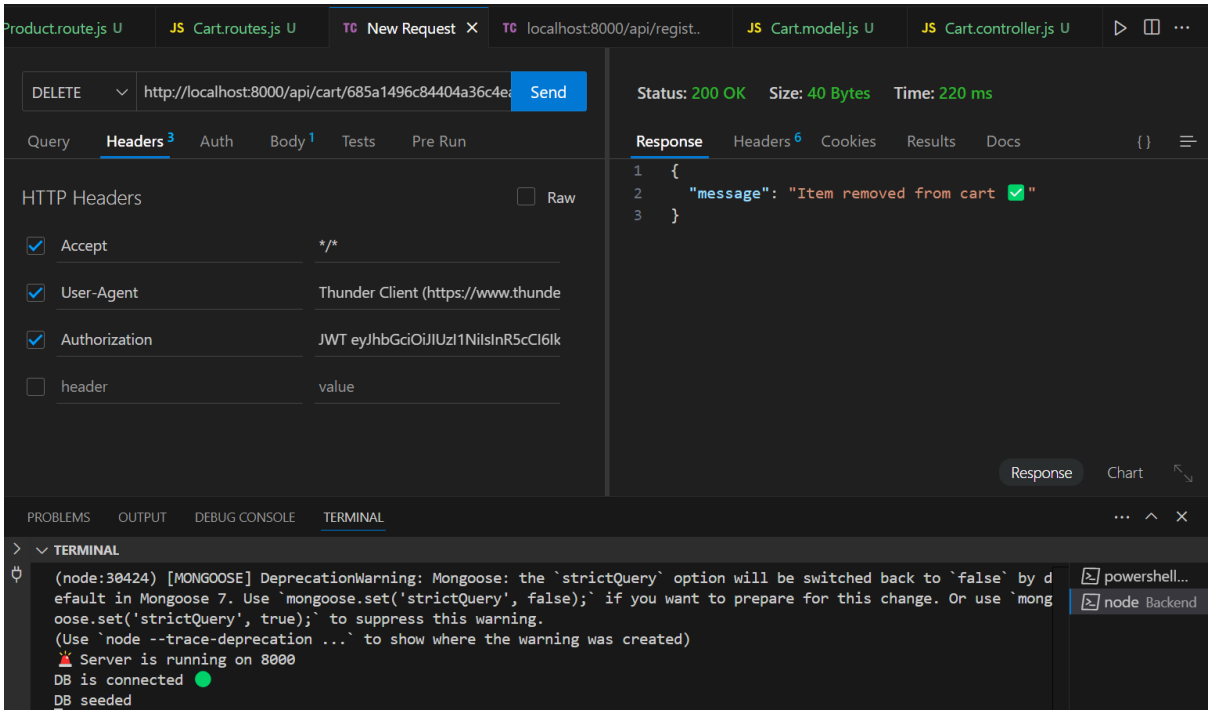


DELETE/cart/ :

Remove a product from the cart.



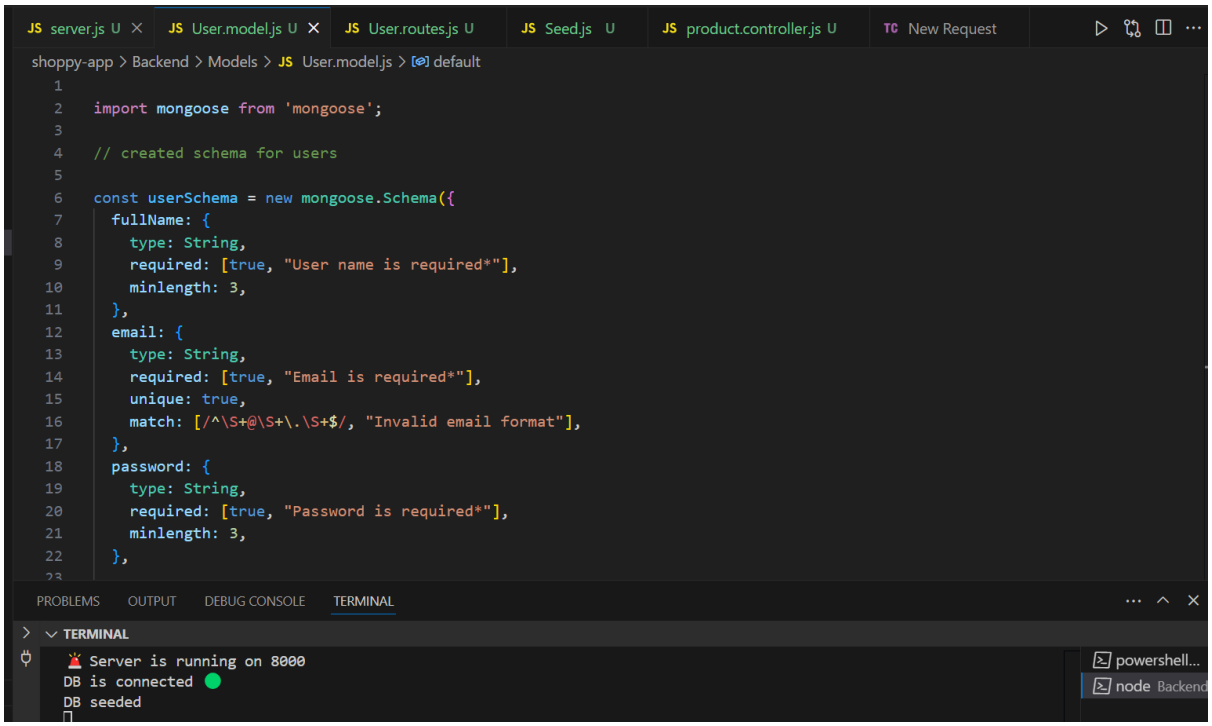
Response in thunder-client

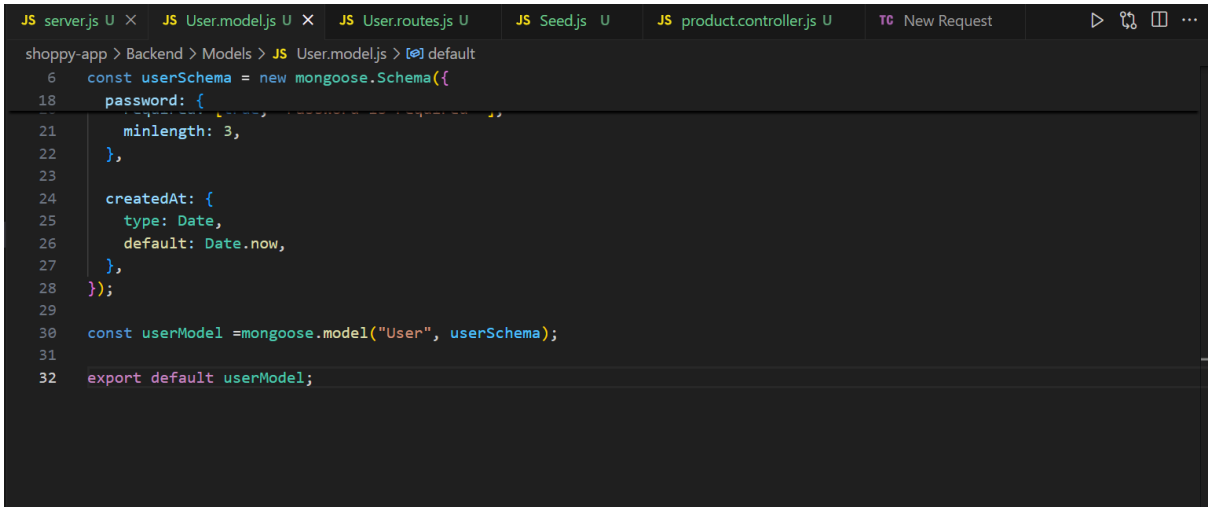


Authentication of User

- Created schema for users in Models part with proper validations
- Developed a model for schema and performed default export

User Schema

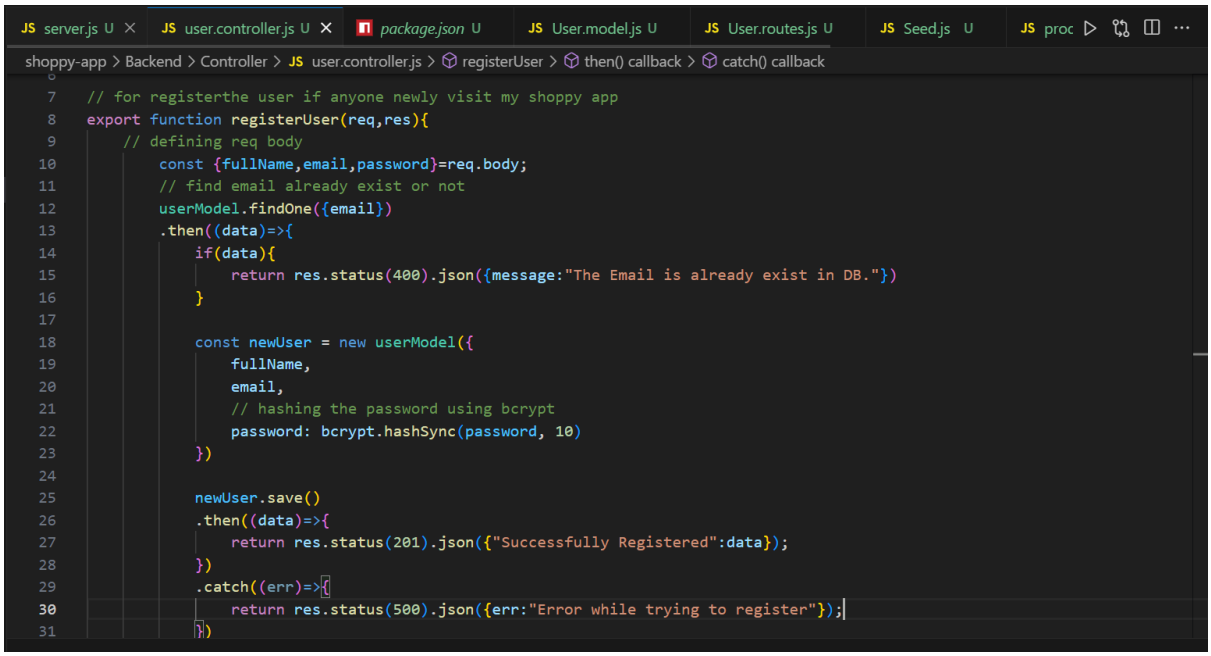




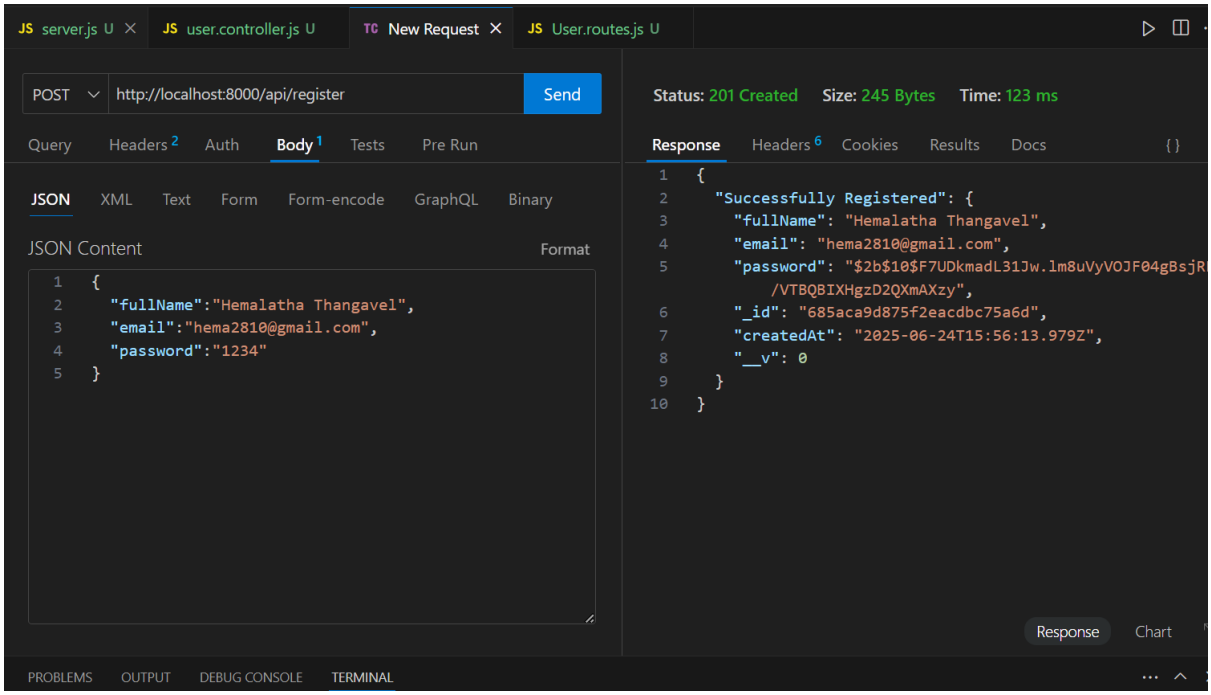
Controller part for users

Registration of users

- I have created route to post user details for registration – "/api/register",registerUser.
- Installed bcrypt using "npm i bcrypt" for hashing the password.

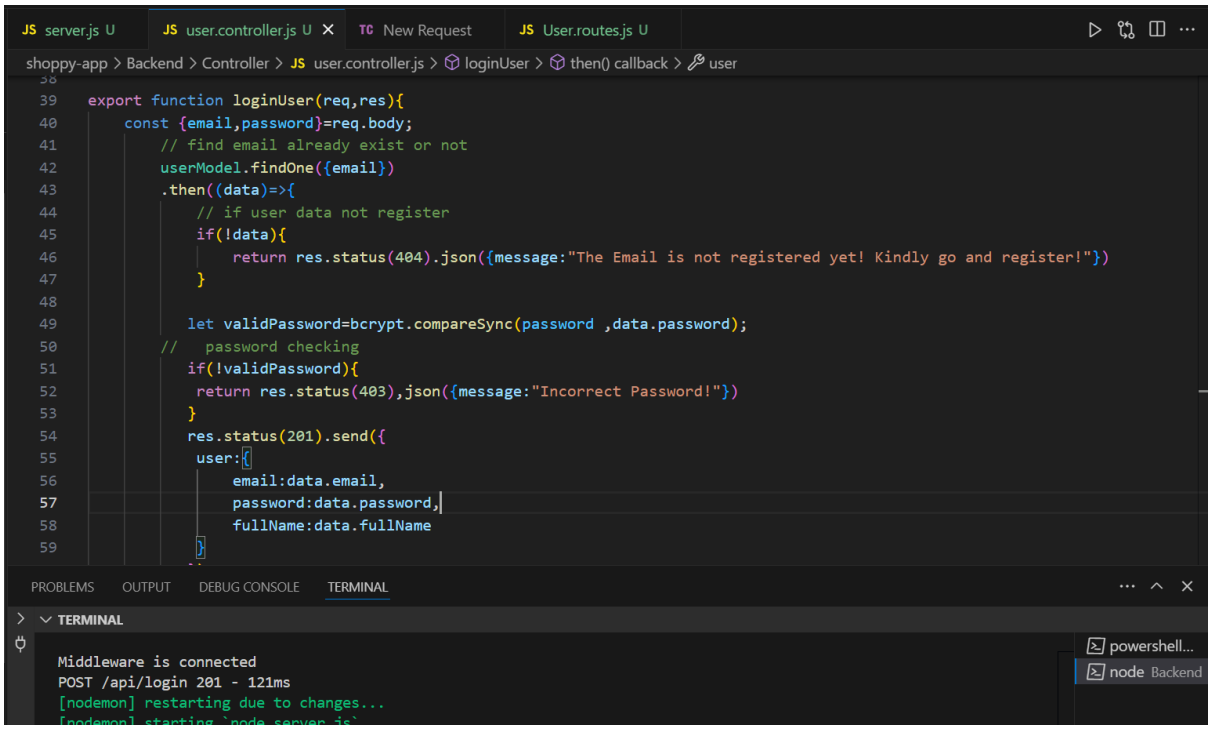


Response in Thunder-Client

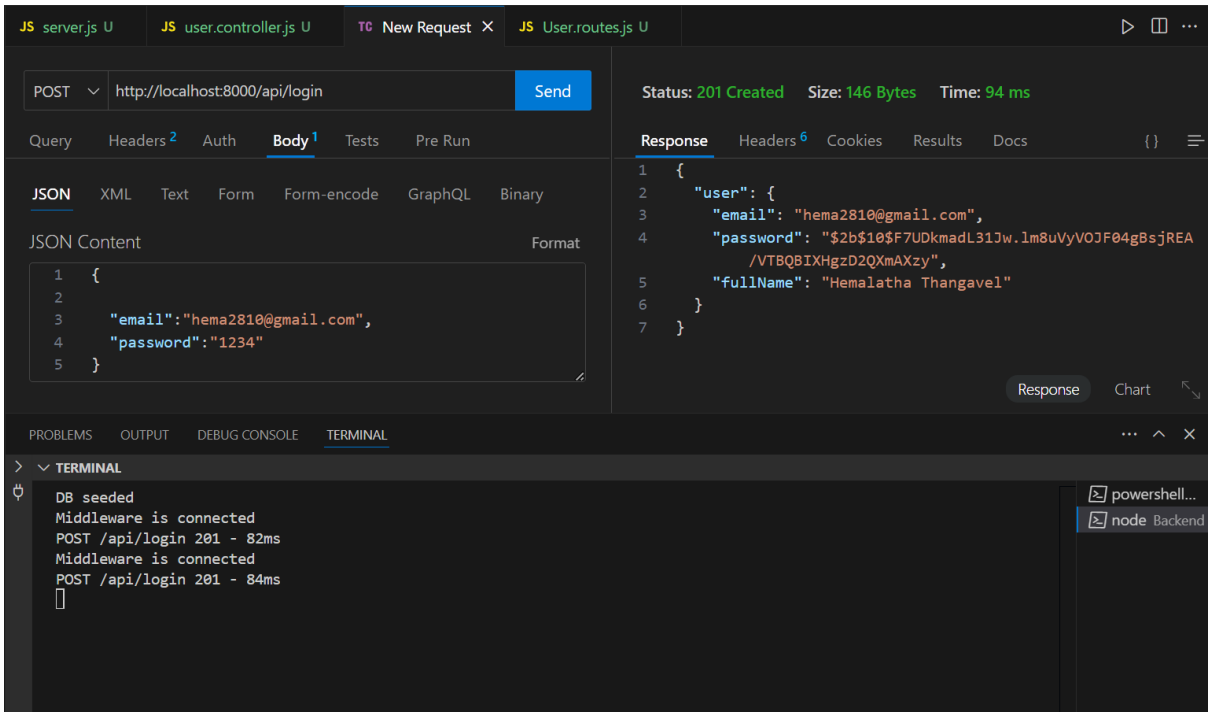


Login by user

- For login the user,I have created the route in model part – "/api/login",loginUser.
- I have written the logic to login the page in controller part.
- Validate the email and password with proper error handling.
- Validate the password using bcrypt.
- Check with thunder-client.

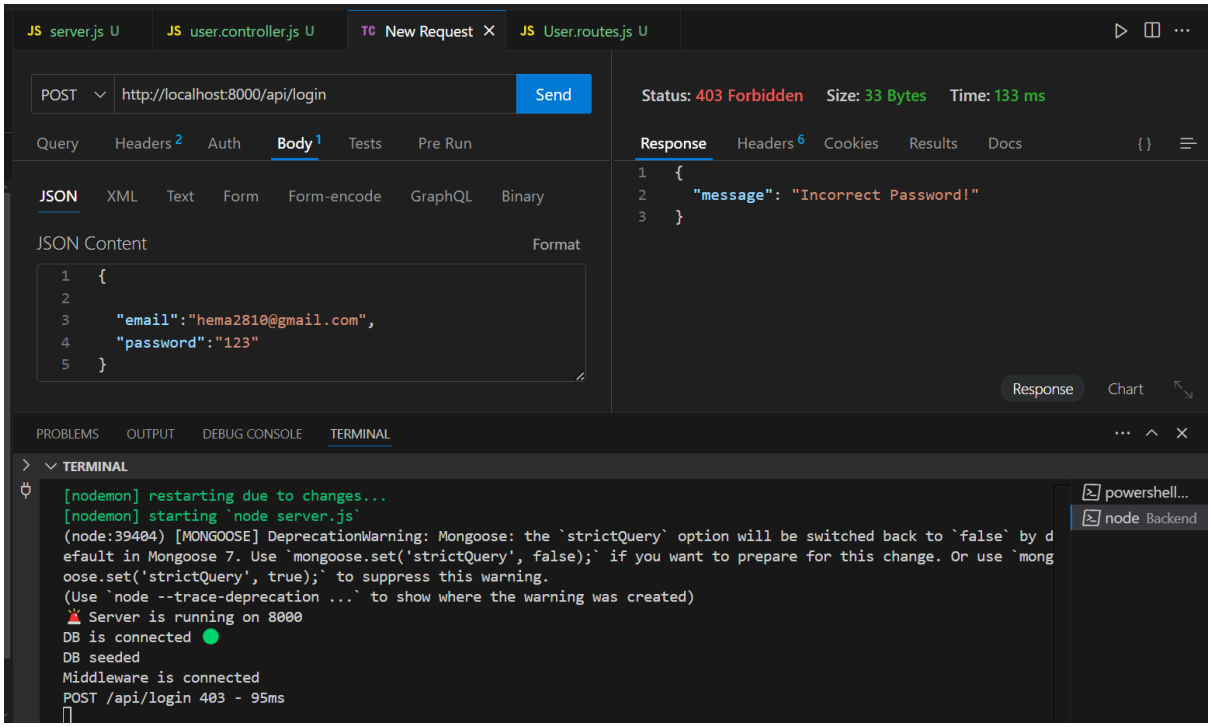


Response in Thunder-Client

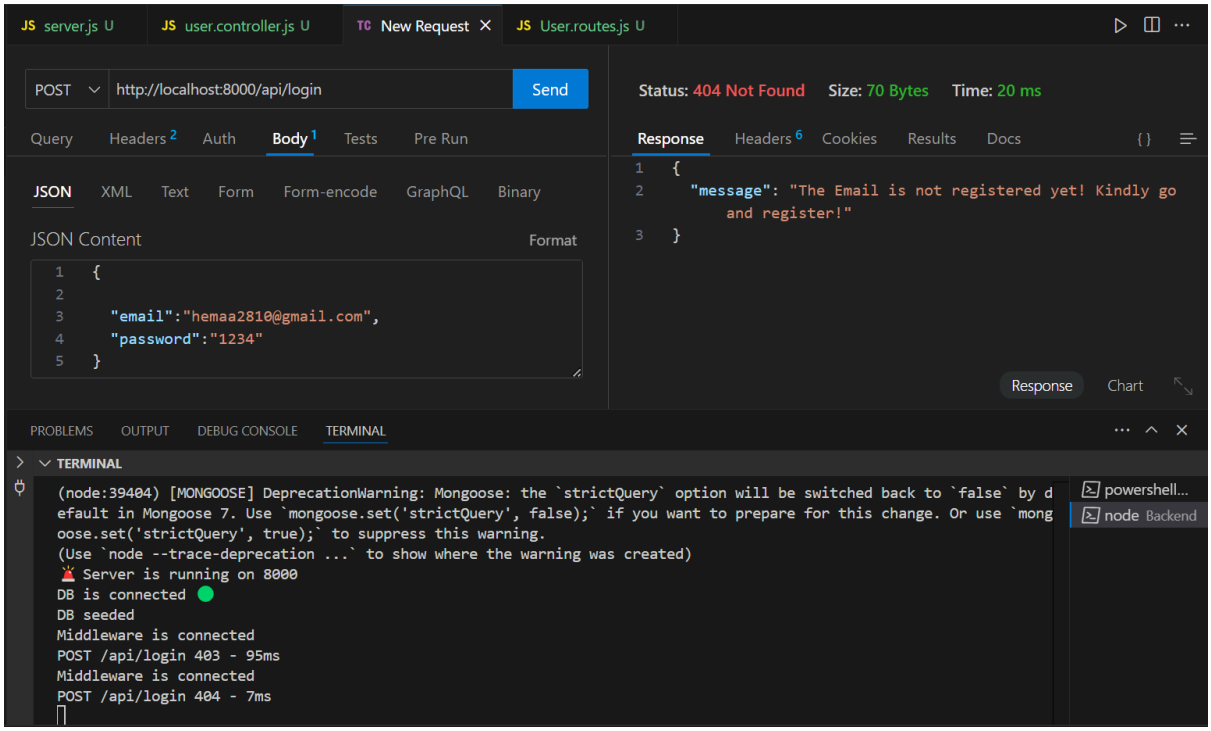


Error handling

If password is wrong-Email is correct

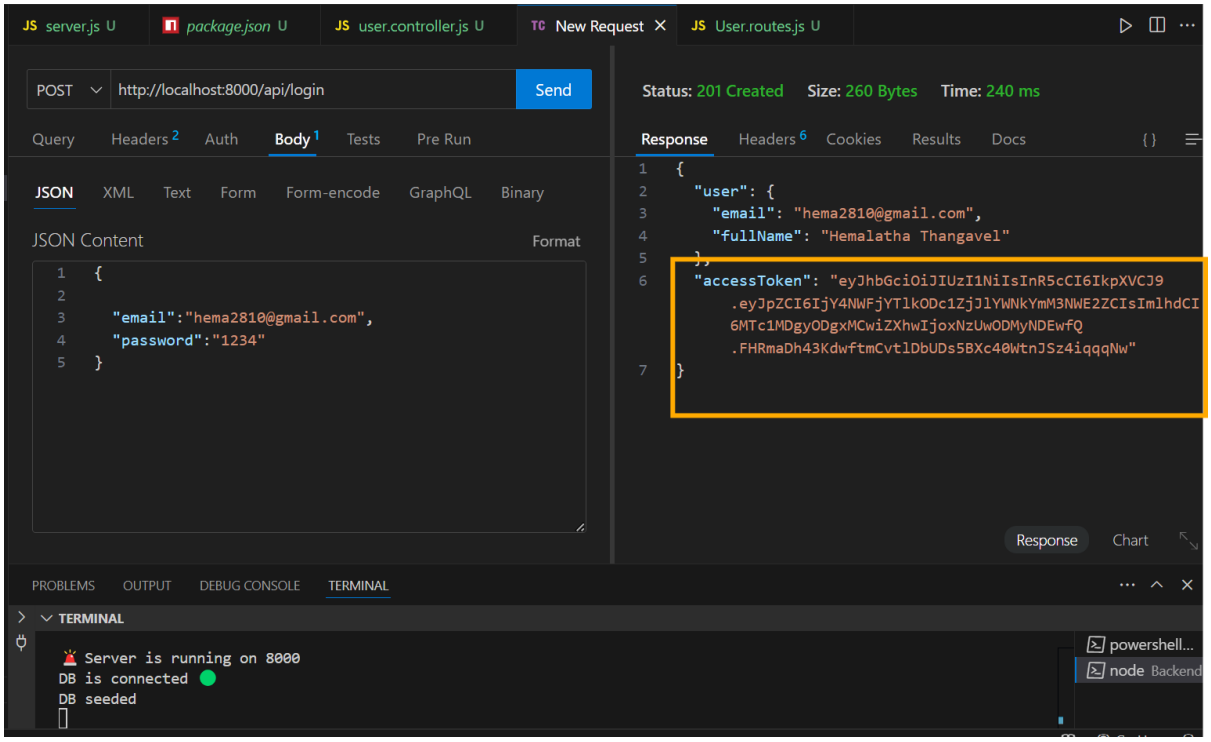


If email is incorrect - password is correct

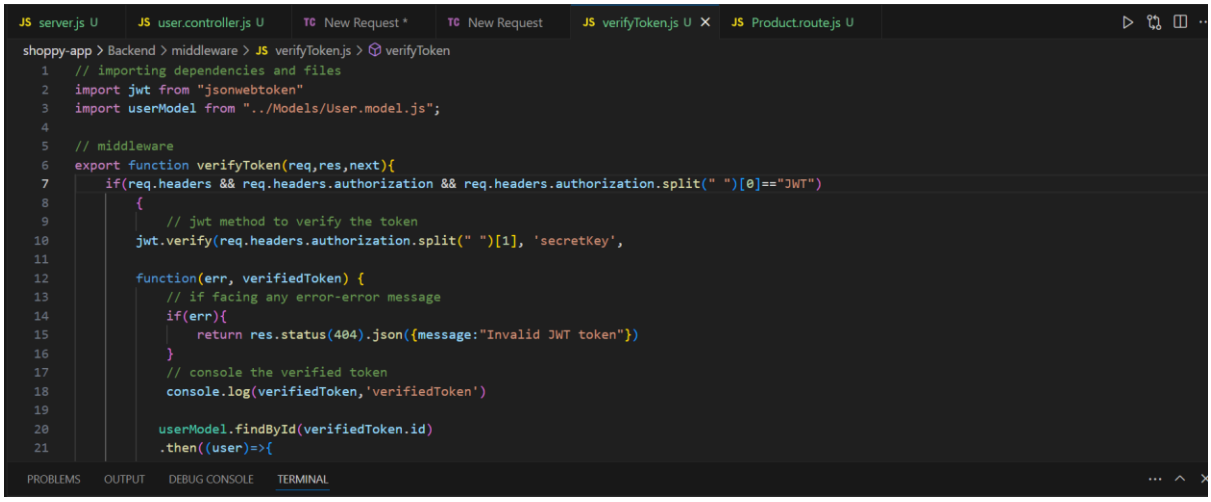


Authorization

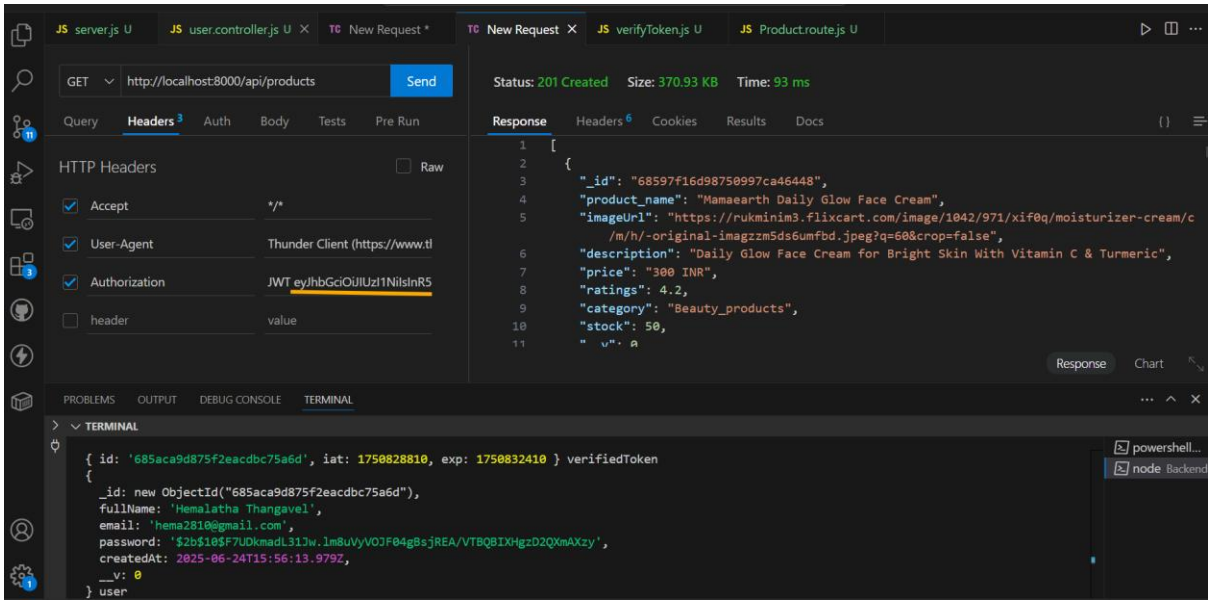
- JSON web token is installed using “npm i jsonwebtoken”
- I have created a access_token

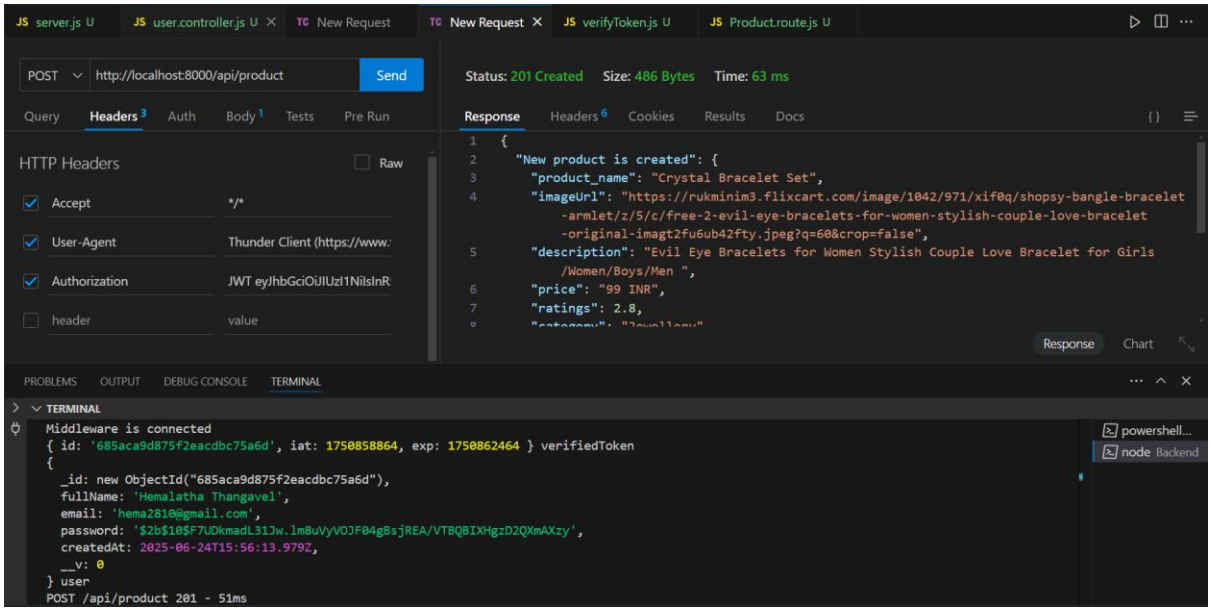


Verify token



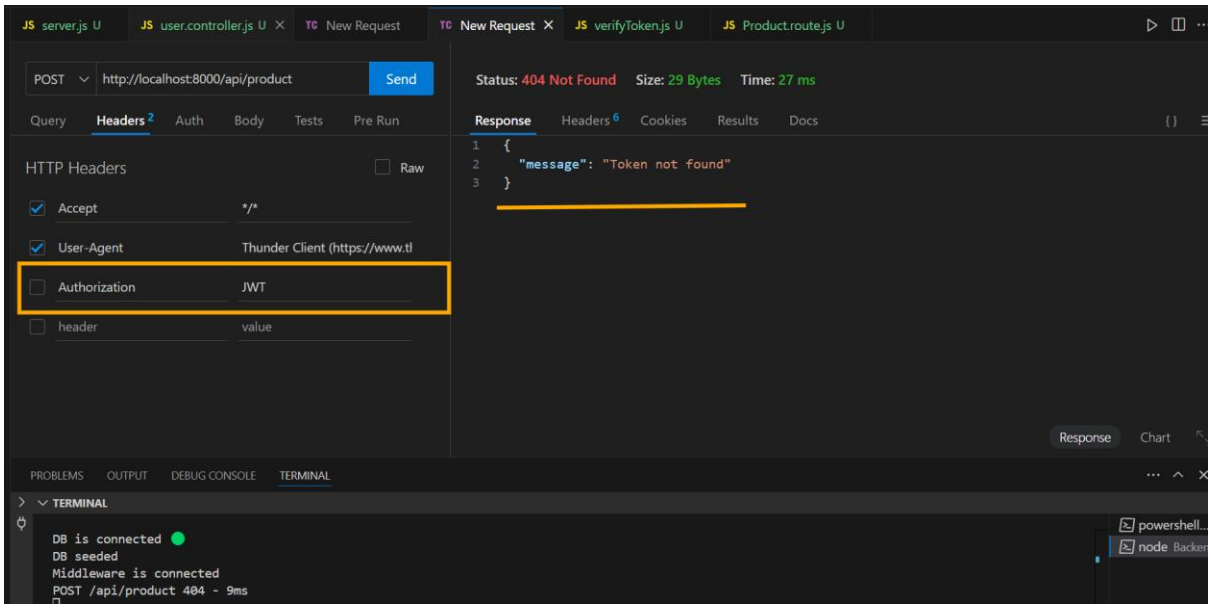
Response in Thunder-Client



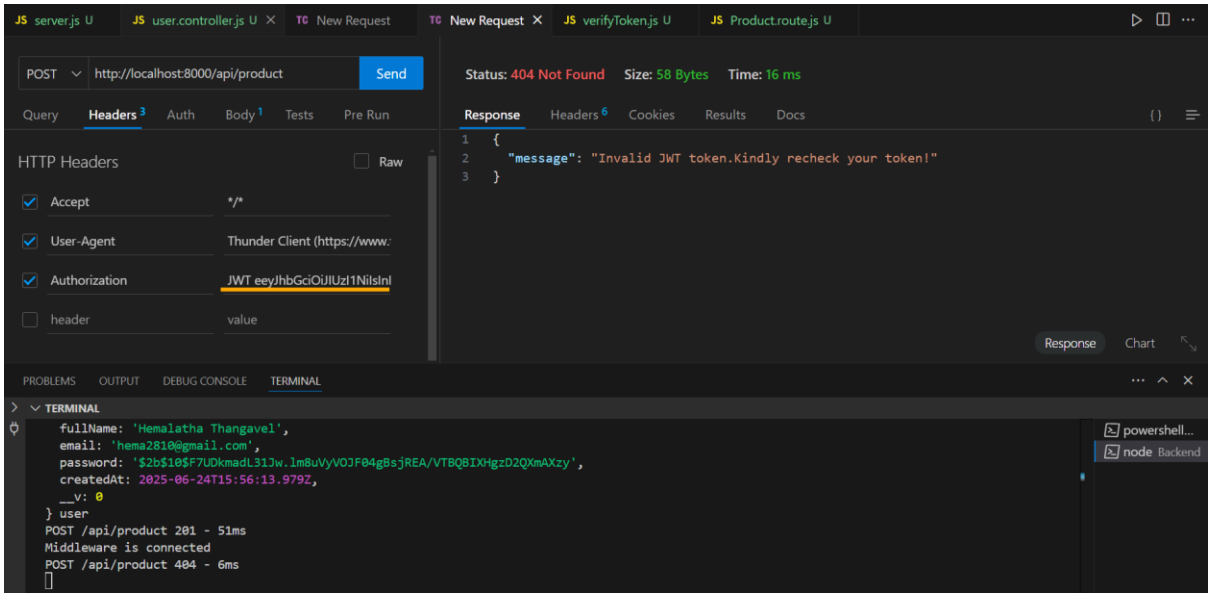


ERROR handling

If I not click authorization check box in headers



If I give invalid or wrong access token for authorization



Conclusion

- Node.js and Express API Setup
- connected with MongoDB compass
- defined routes for products and cart
- performed CRUD operations with proper error handling
- Used MongoDB to store product data and cart items.
- Added screenshots from MongoDB Database
- Performed Authentication & Authorization
- Tested with Thunder-Client
- Ensured all api is working
- included comments and documentations

GitHub link

https://github.com/Hema2802/Backend-ecommerce-shoppy--app