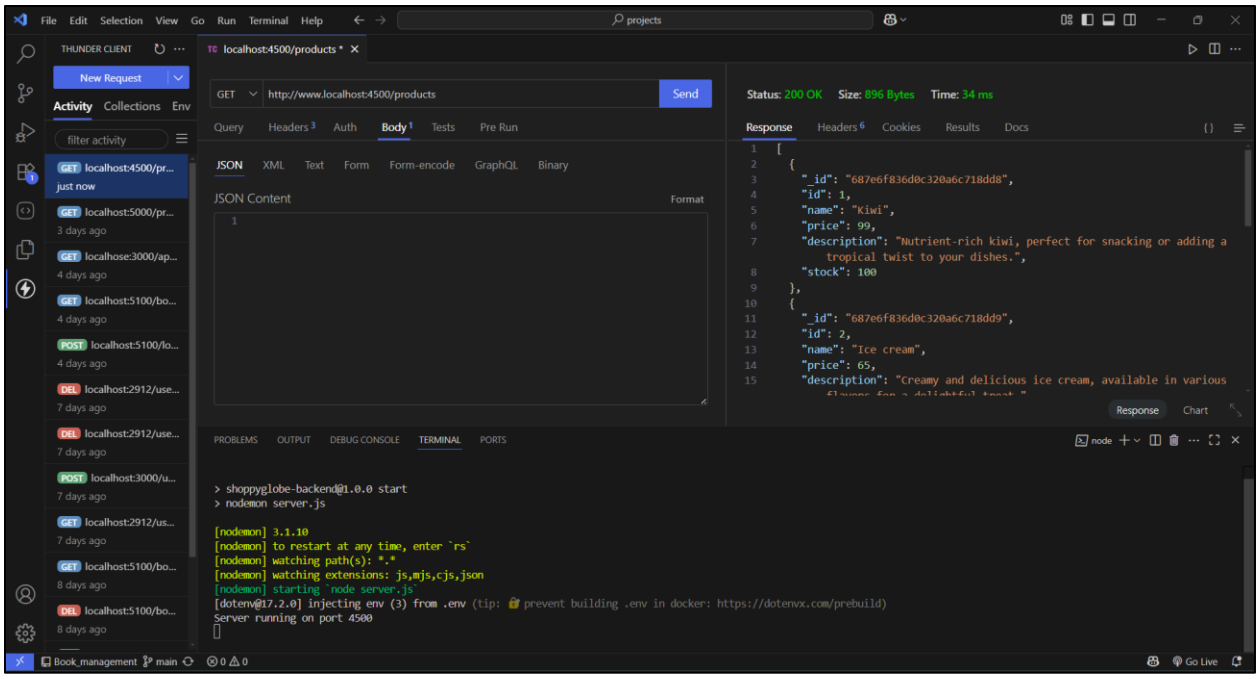
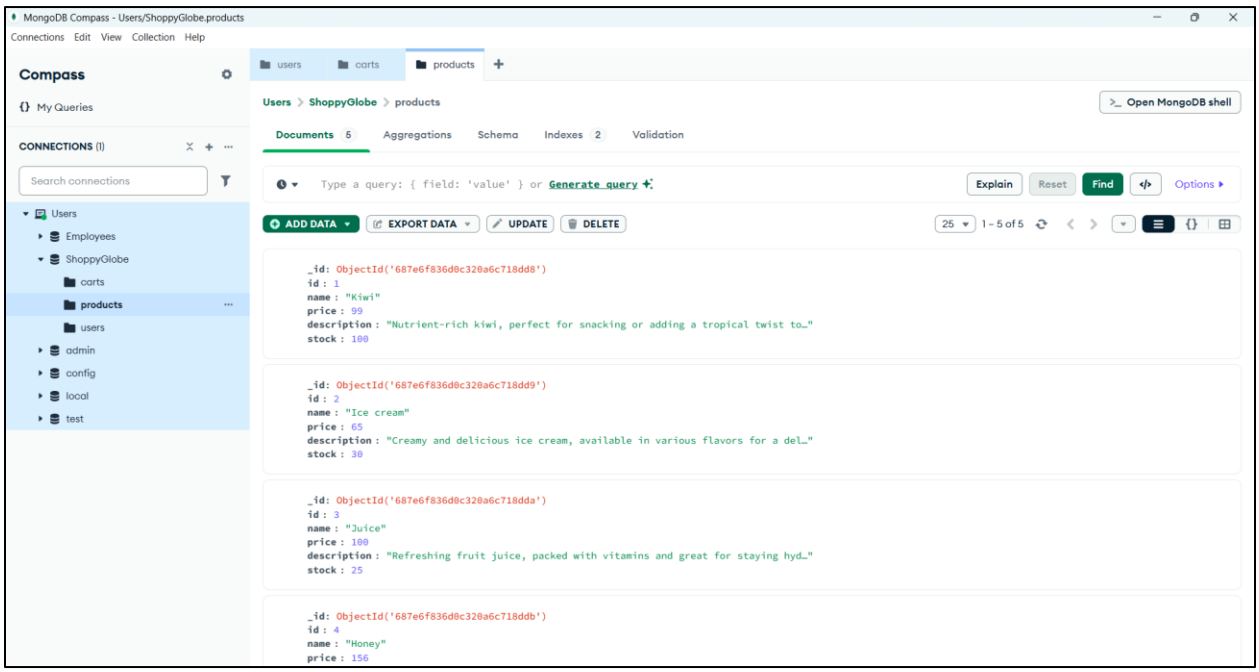
API testing for ShoppyGlobe application using Thunder Client

1. GET /products

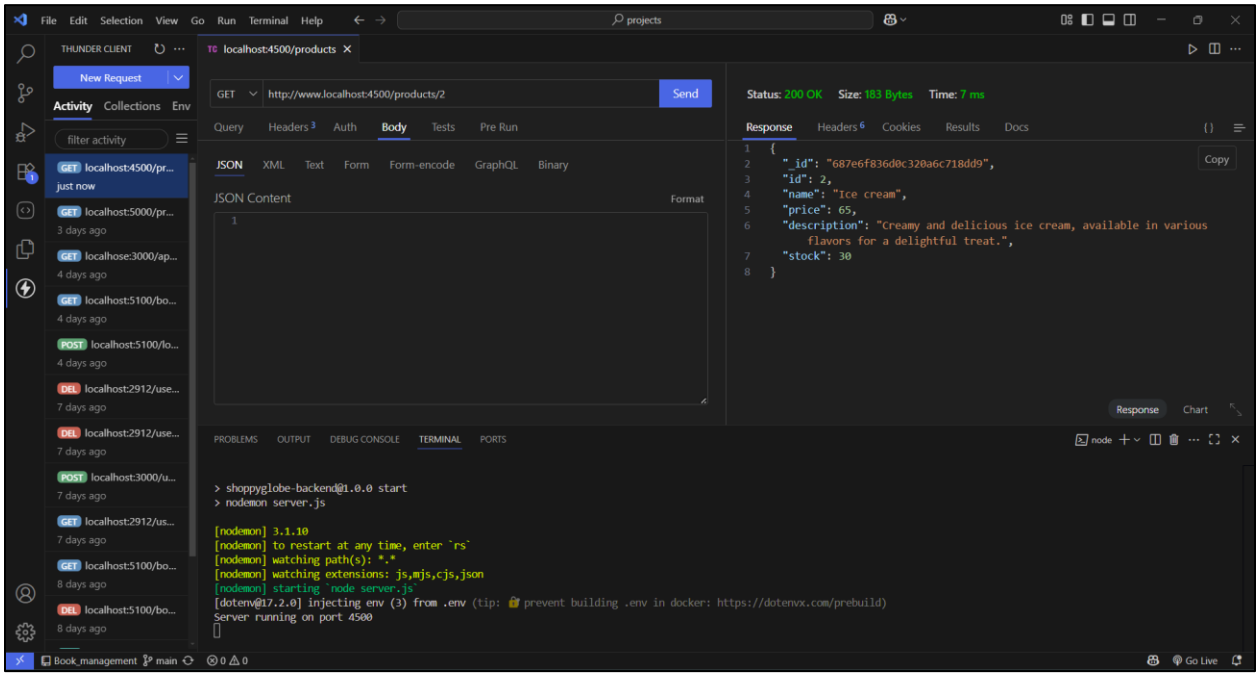
Fetch a list of products from MongoDB.





2. GET /products/:id

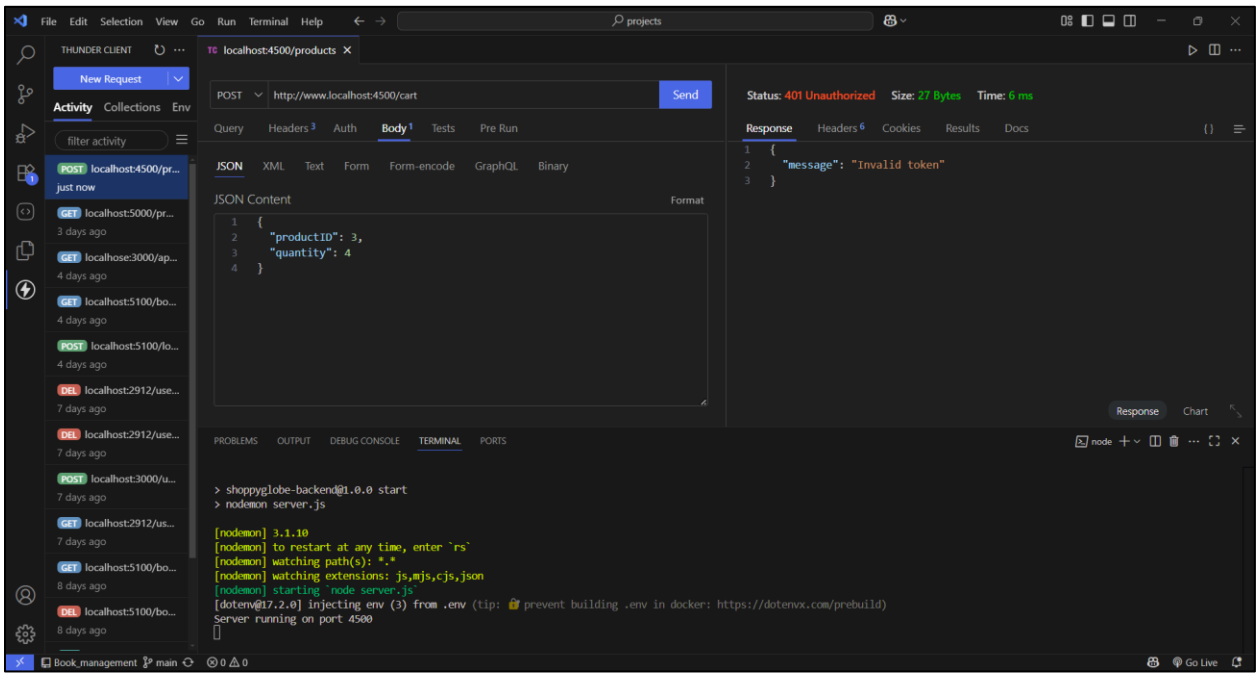
Fetch details of a single product by its ID.



3. POST /cart

Add a product to the shopping cart.

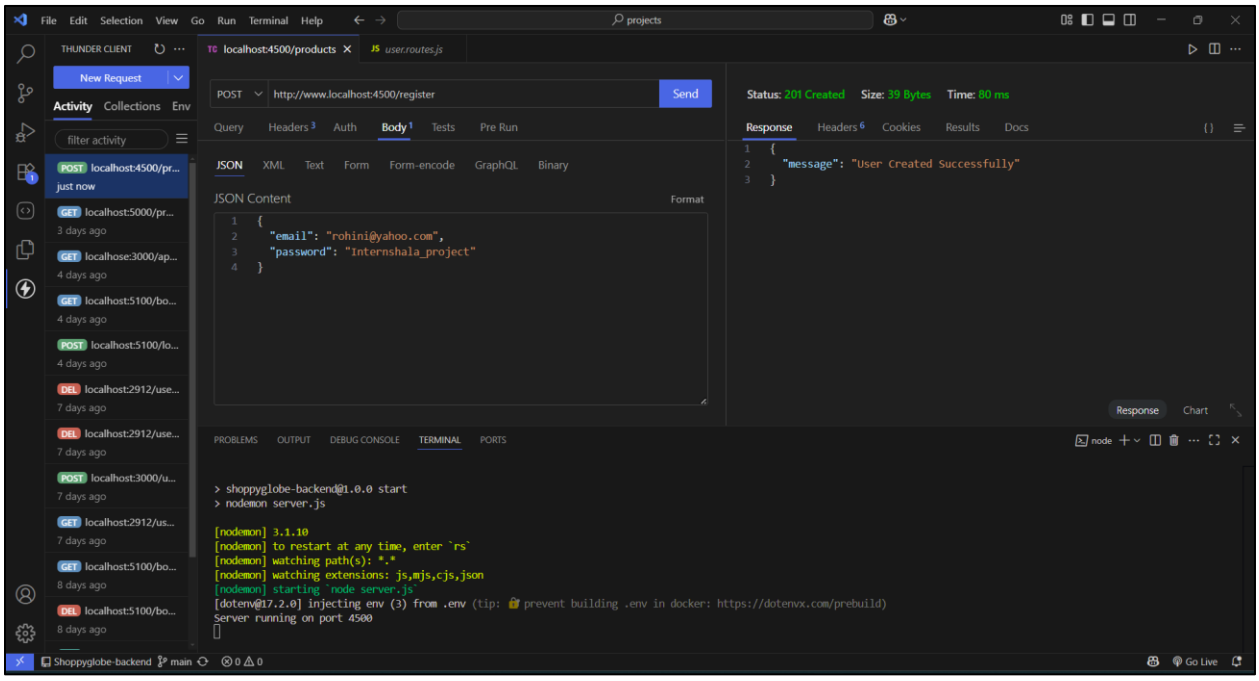
Before authentication:

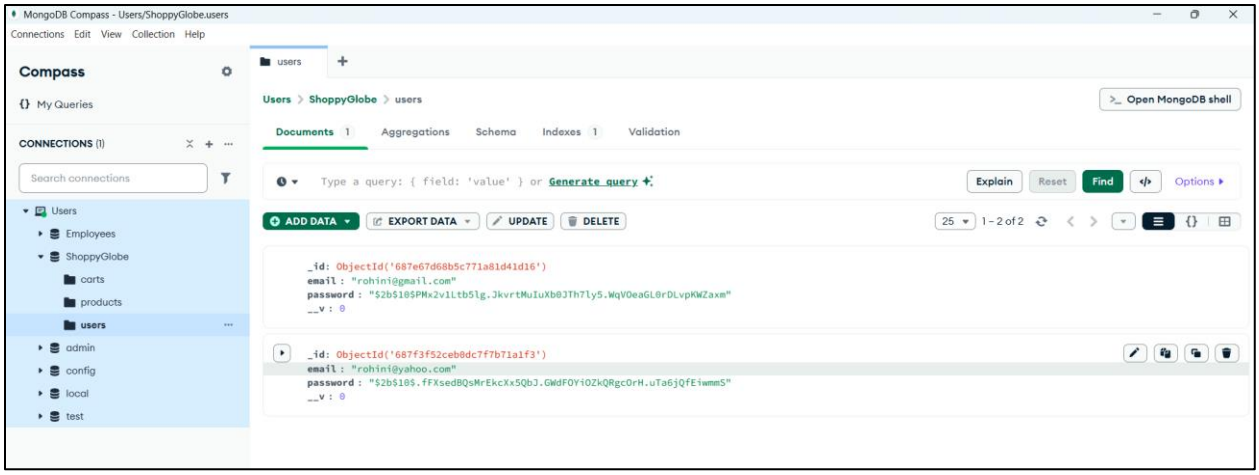


Users have to create their account and authenticate before accessing the cart.

- **User registration**

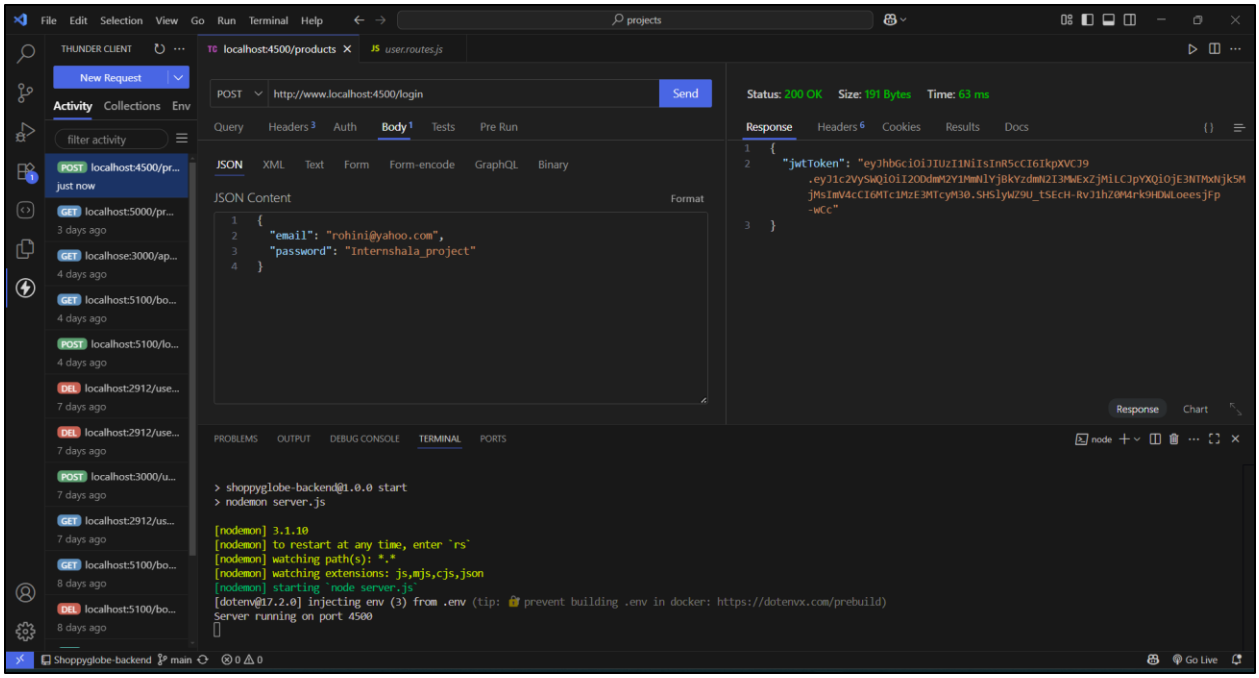
POST /register
Creates the user in the Mongo DB





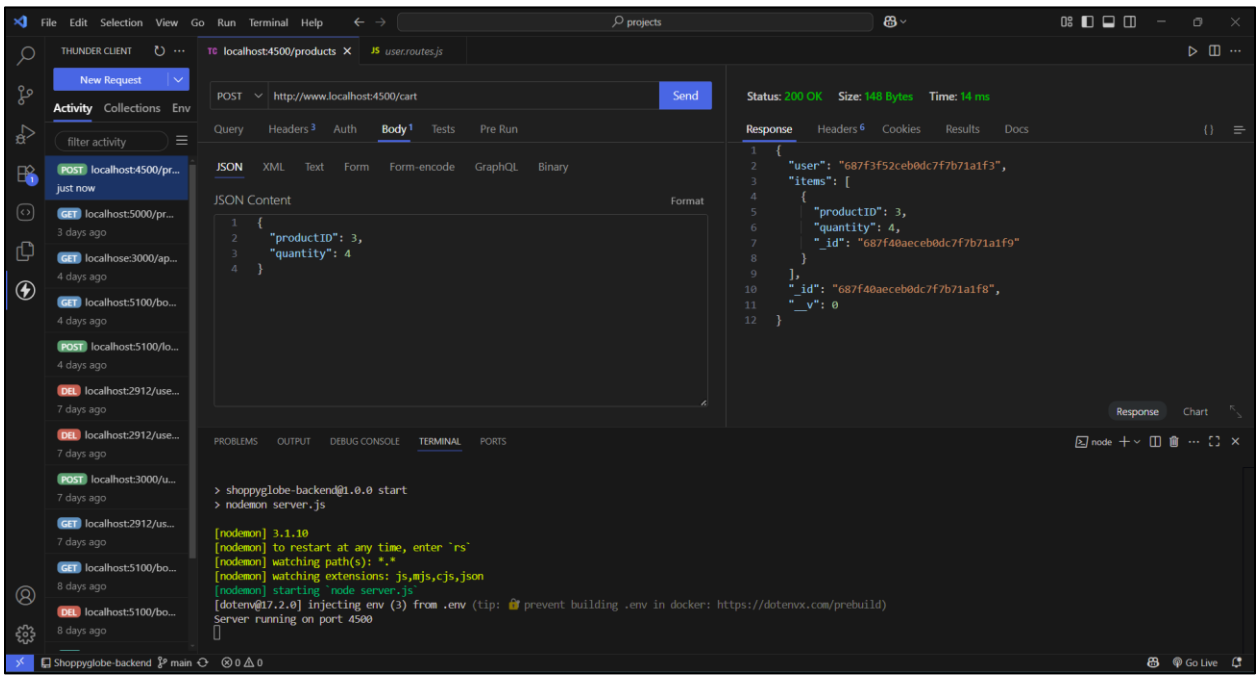
- **User login**

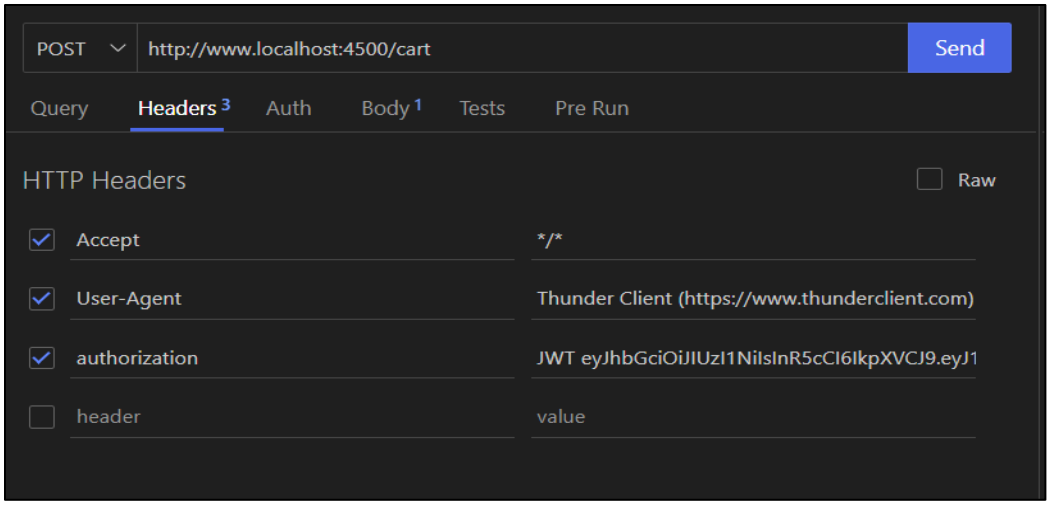
POST /login
Generates the JWT token which can be used for login and access the cart

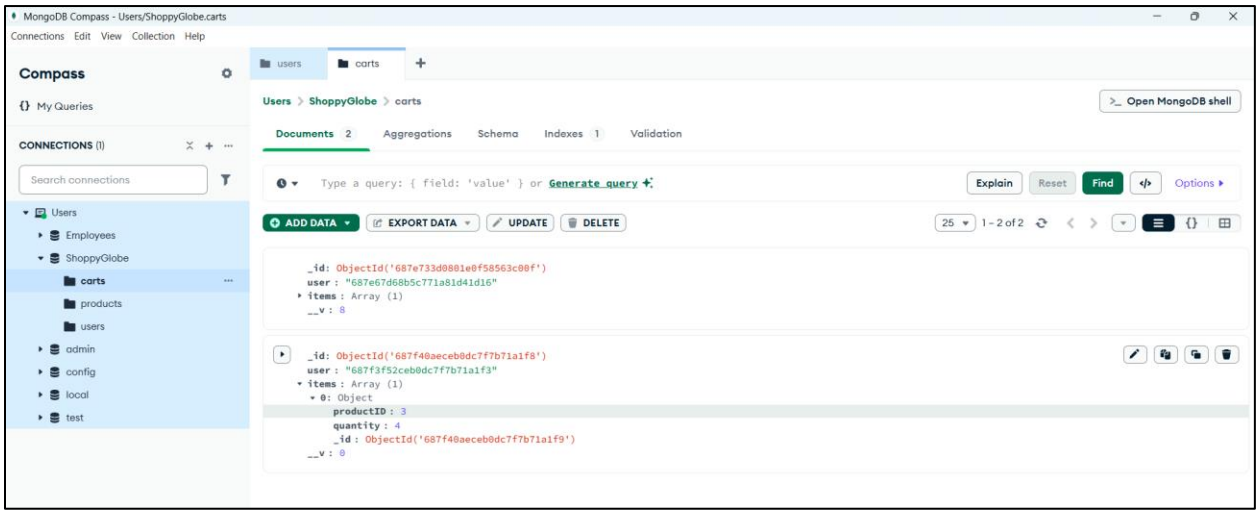


After Authentication:

Added the JWT token in Headers. Now, the users will be able to add the products to cart.

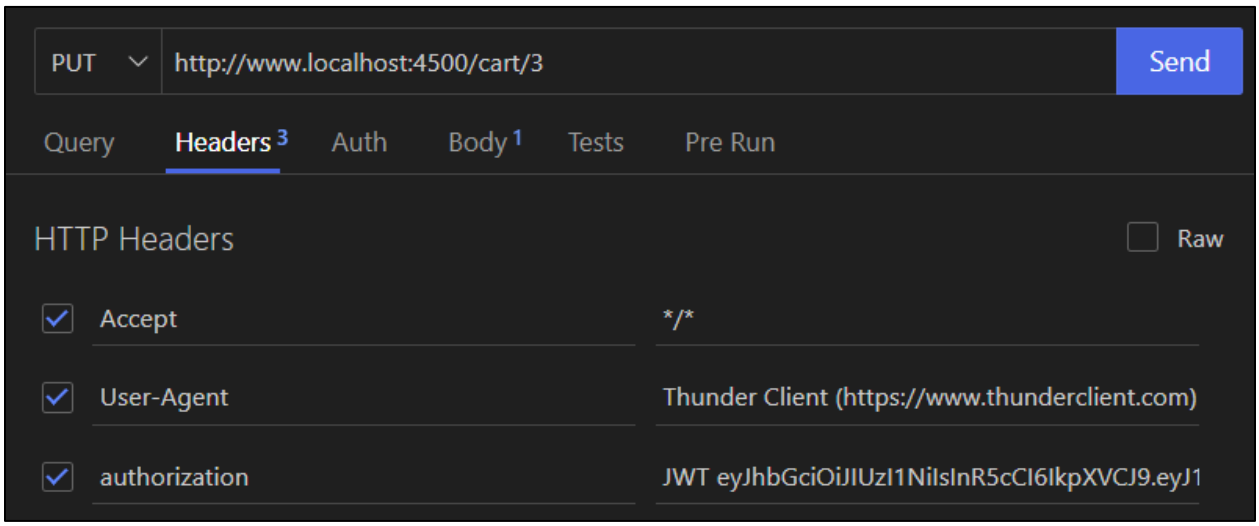


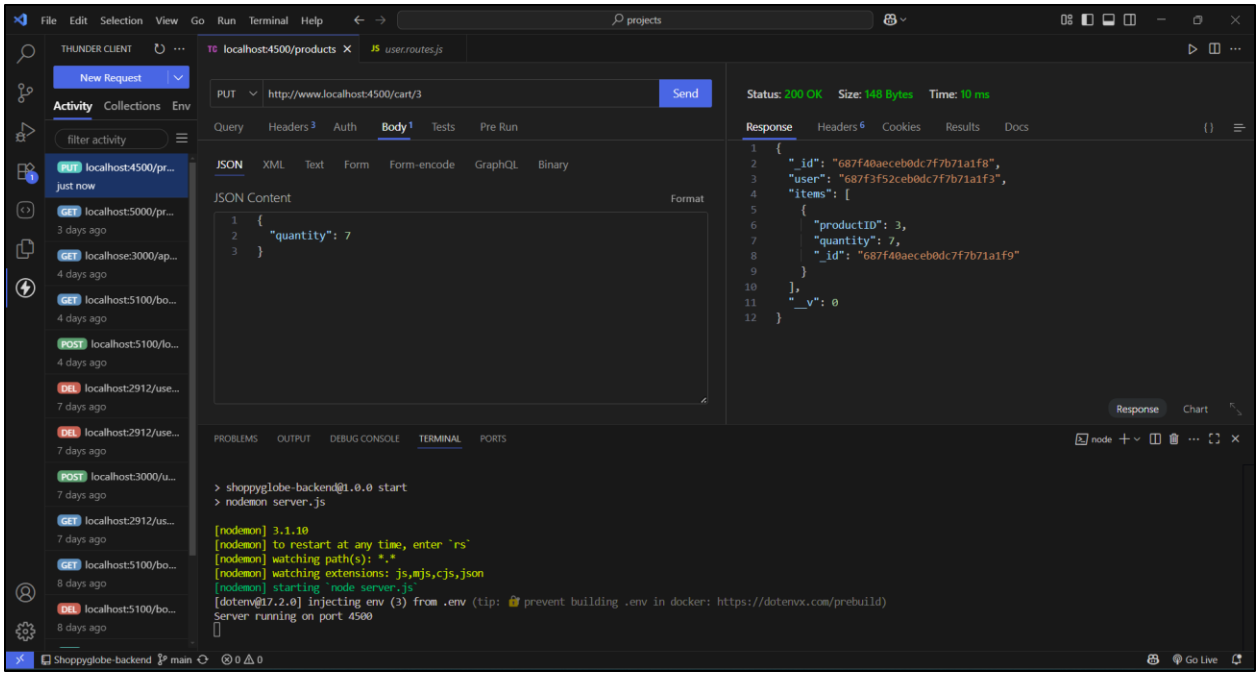


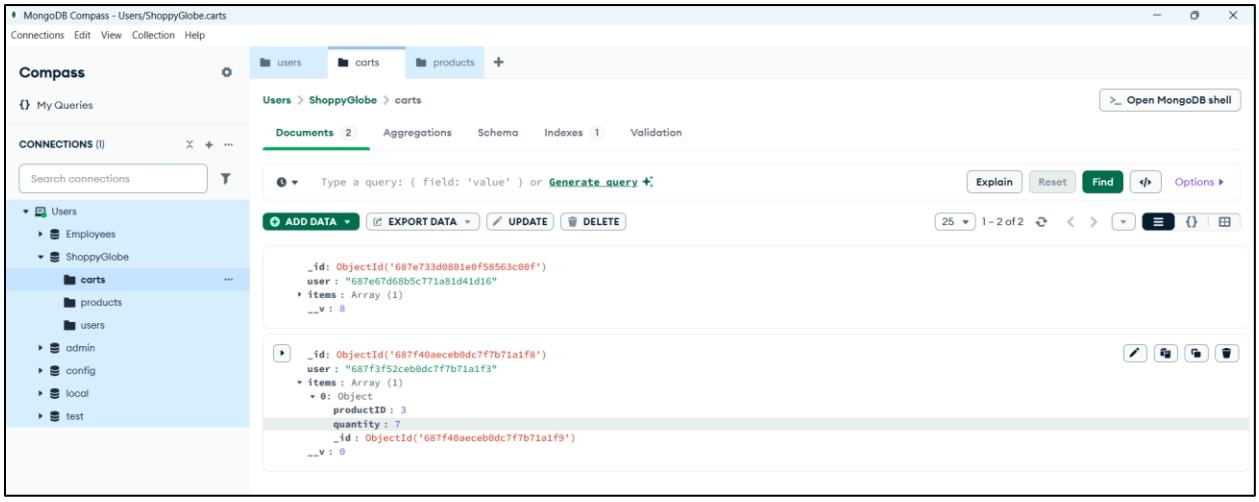


4. PUT /cart/:id

Update the quantity of a product in the cart.

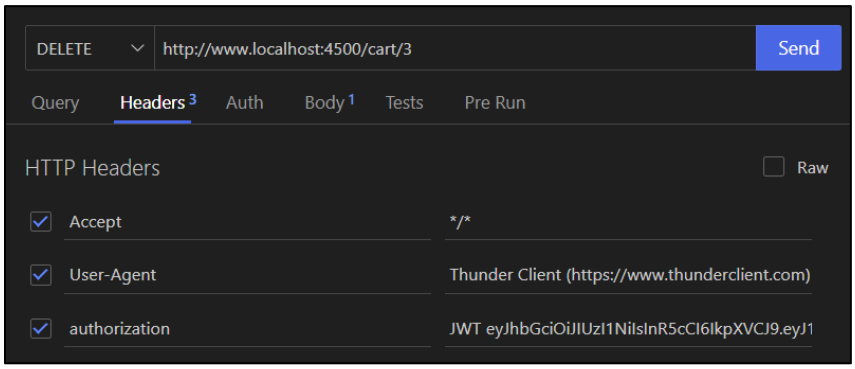


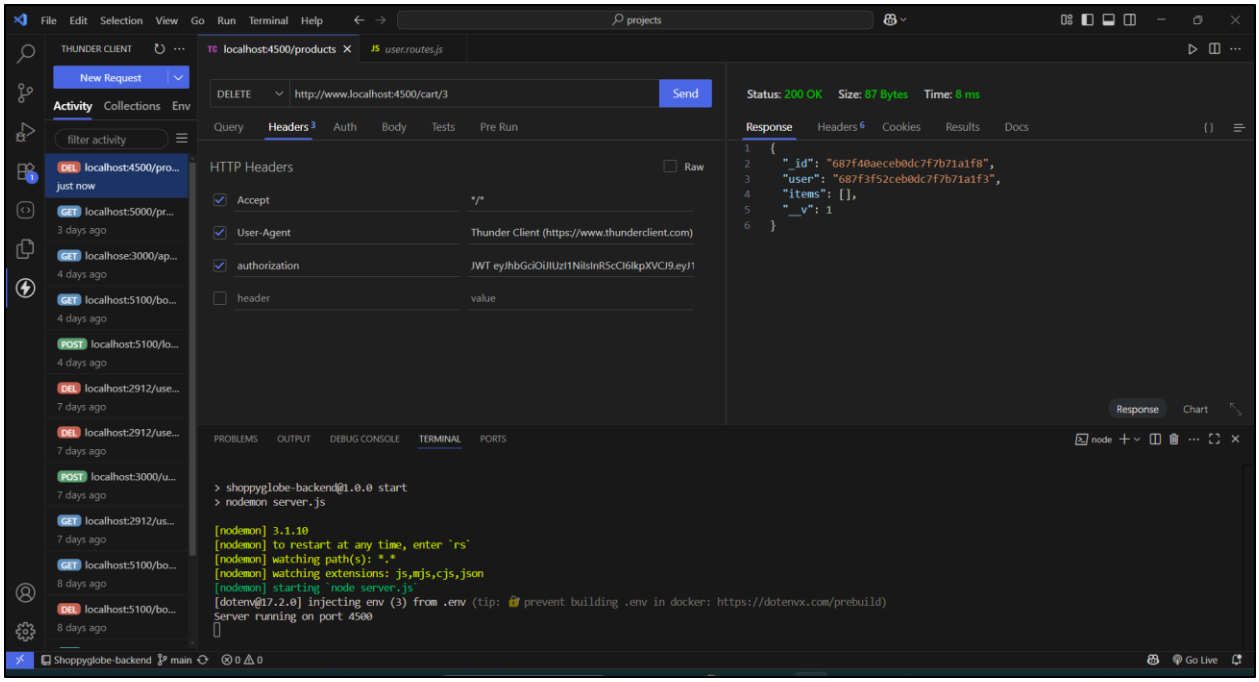




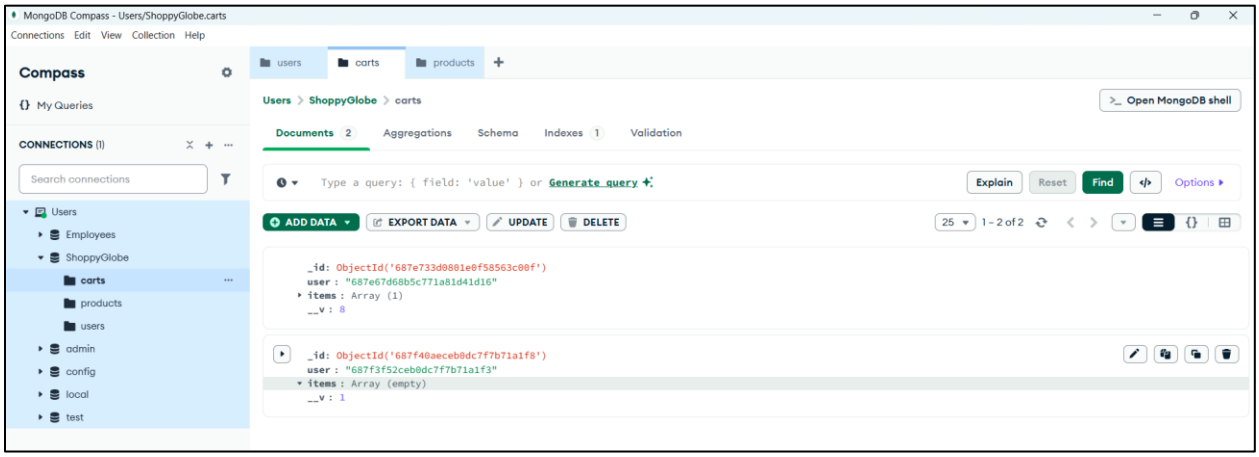
5. DELETE /cart/:id

Remove a product from the cart.





Deleted the item. So, now the cart is empty.

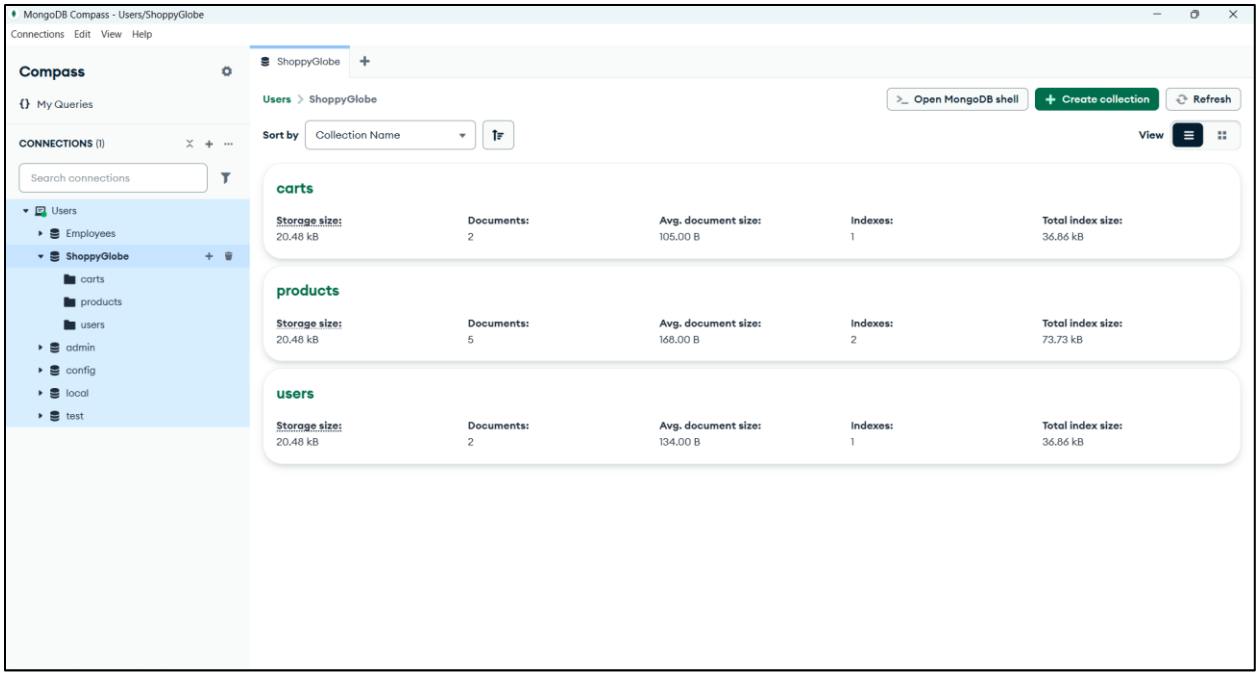


Screenshots from MongoDB:

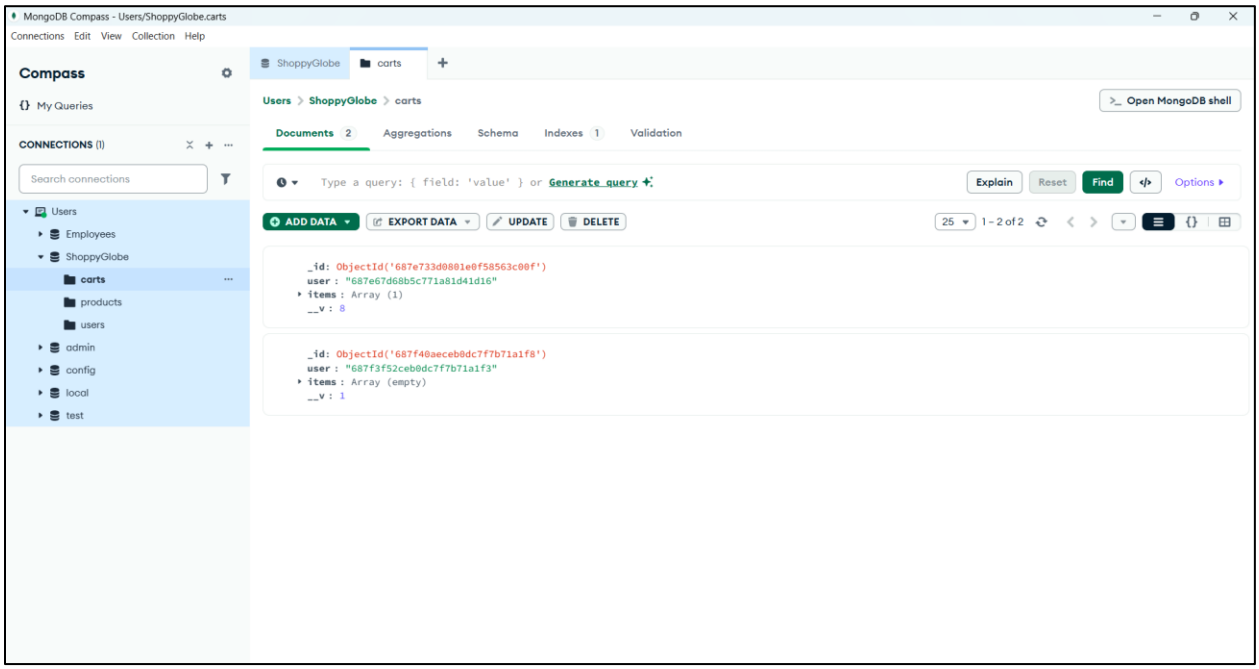
Database Name: ShoppyGlobe

It has 3 collections:

- Carts
- Products
- Users



Carts Collection:

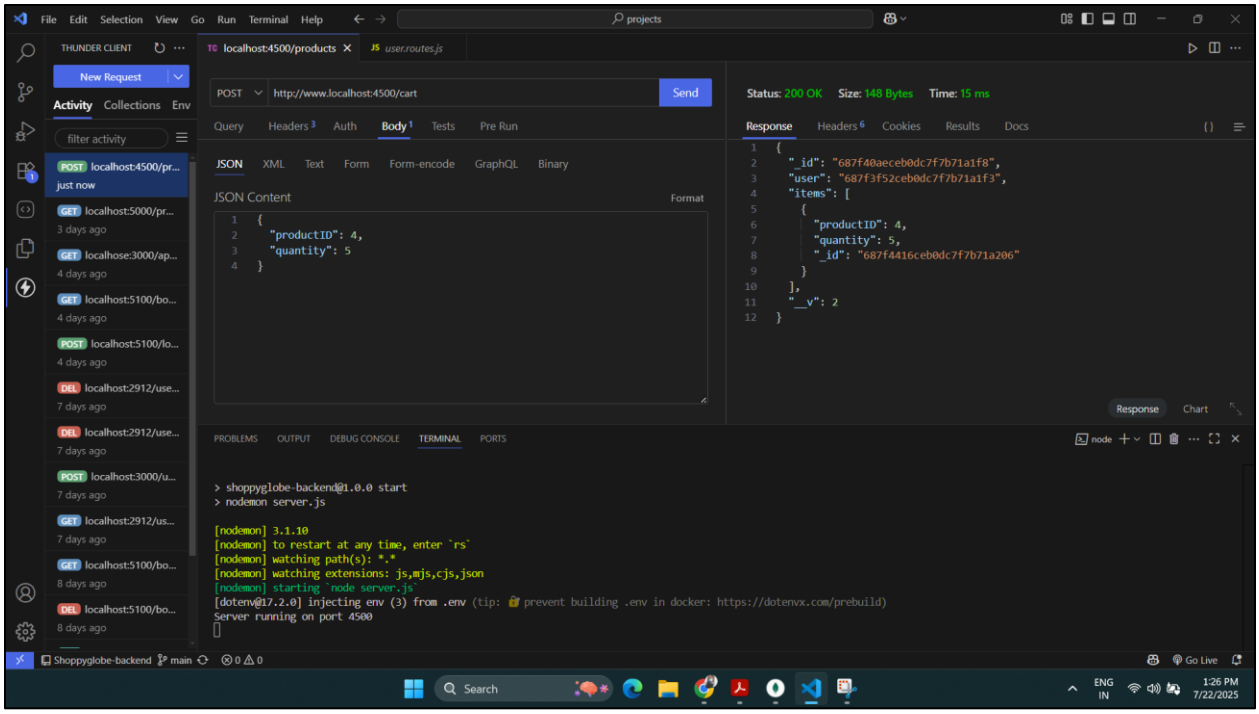


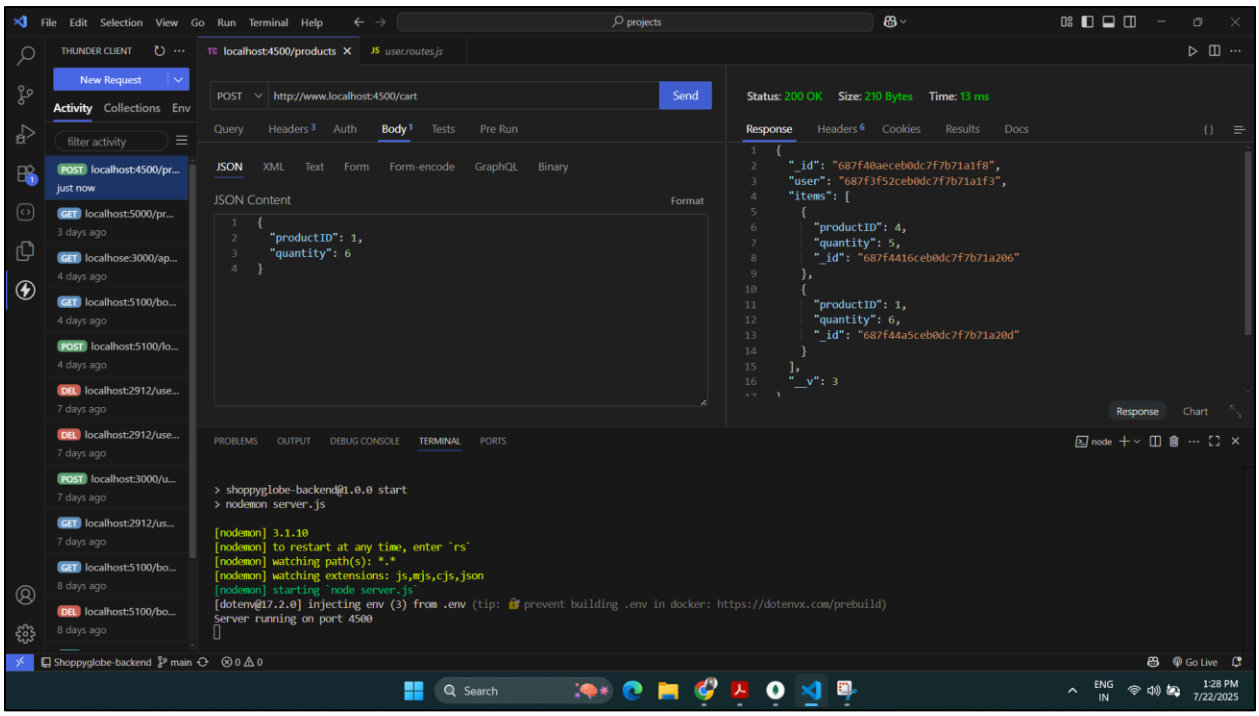
CRUD operation on Cart:

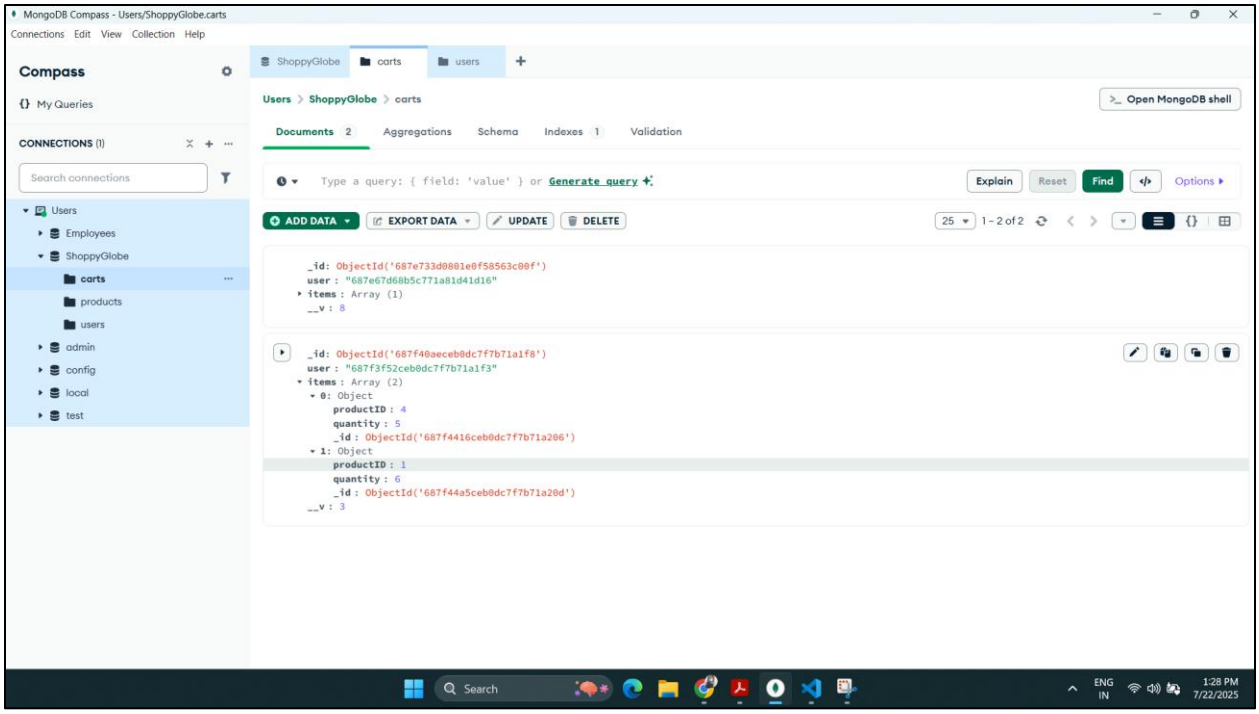
1. Create document

POST /cart

Add the items to the cart


THUNDER CLIENT
New Request
Activity Collections Env
filter activity
localhost:4500/products
user.routes.js
POST http://www.localhost:4500/cart
Send
Status: 200 OK Size: 148 Bytes Time: 15 ms
Query Headers 3 Auth Body 1 Tests Pre Run
Response Headers 6 Cookies Results Docs
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"productID": 4,
"quantity": 5
}
{
"_id": "687f40aeceb0dc7f7b71a1f8",
"user": "687f3f52ceb0dc7f7b71a1f3",
"items": [
{
"productID": 4,
"quantity": 5,
"_id": "687f4416ceb0dc7f7b71a206"
}
],
"__v": 2
}
Response
Chart
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
> shoppyglobe-backend@1.0.0 start
> nodemon server.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
[dotenv@17.2.0] injecting env (3) from .env (tip: prevent building .env in docker: https://dotenvx.com/prebuild)
Server running on port 4500
Shoppyglobe-backend main
Go Live
Search
ENG IN
1:26 PM 7/22/2025

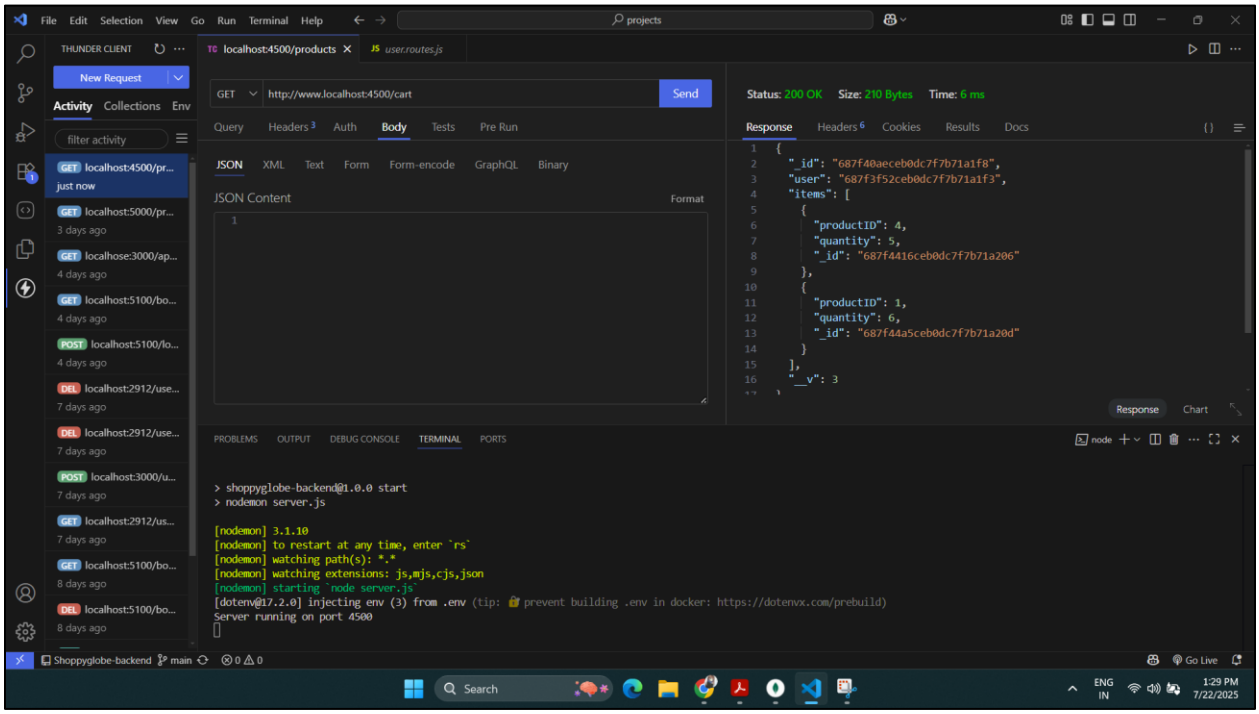

THUNDER CLIENT
New Request
Activity Collections Env
filter activity
localhost:4500/products
user.routes.js
POST http://www.localhost:4500/cart
Send
Status: 200 OK Size: 210 Bytes Time: 13 ms
Query Headers 3 Auth Body 1 Tests Pre Run
Response Headers 6 Cookies Results Docs
JSON XML Text Form Form-encode GraphQL Binary
JSON Content
Format
{
"productID": 1,
"quantity": 6
}
{
"_id": "687f40aeceb0dc7f7b71a1f8",
"user": "687f3f52ceb0dc7f7b71a1f3",
"items": [
{
"productID": 4,
"quantity": 5,
"_id": "687f4416ceb0dc7f7b71a206"
},
{
"productID": 1,
"quantity": 6,
"_id": "687f44a5ceb0dc7f7b71a20d"
}
],
"__v": 3
Response
Chart
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node
> shoppyglobe-backend@1.0.0 start
> nodemon server.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
[dotenv@17.2.0] injecting env (3) from .env (tip: prevent building .env in docker: https://dotenvx.com/prebuild)
Server running on port 4500
Shoppyglobe-backend main
Go Live
Search
ENG IN
1:28 PM 7/22/2025



2. Read document

GET /cart

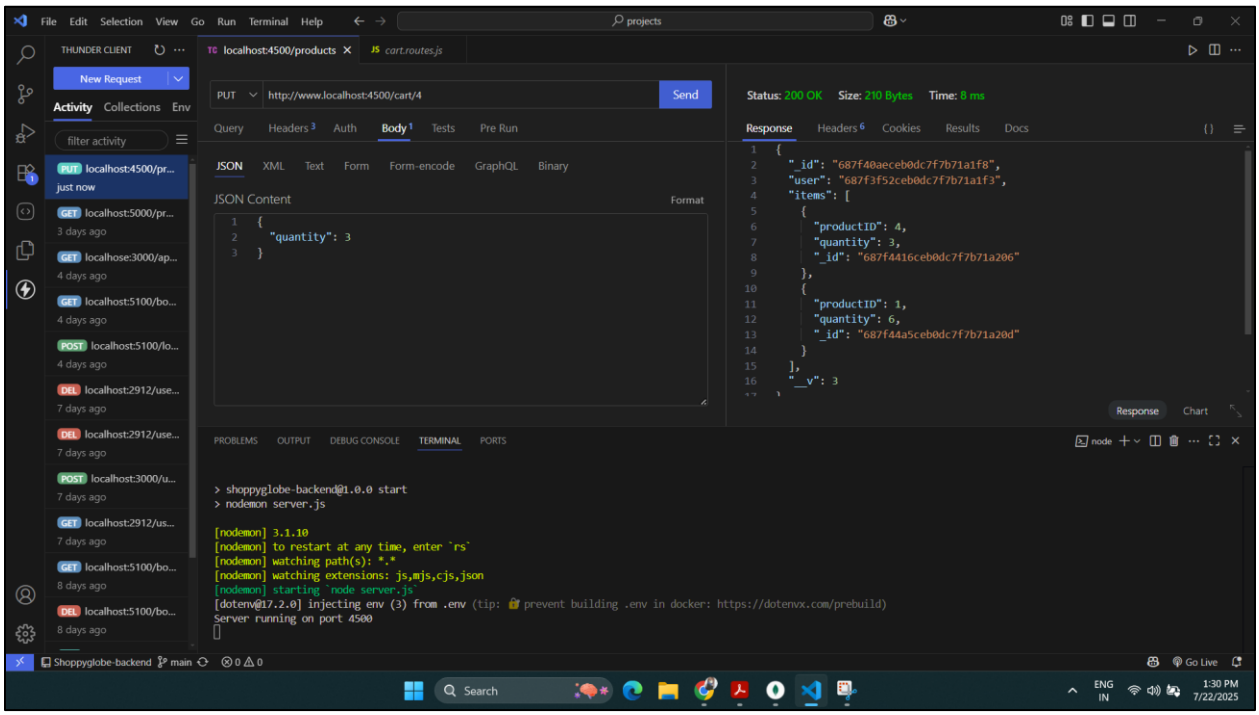
Fetches all the items from the cart

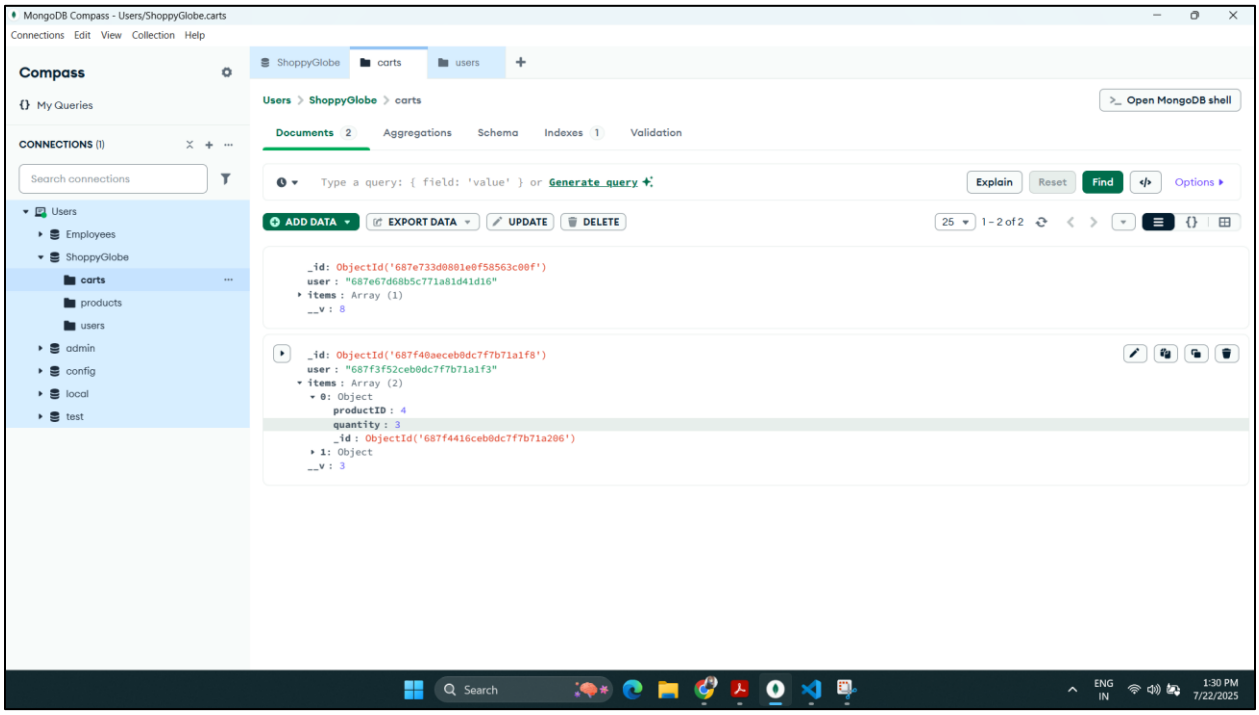


3. Update document

PUT /cart/:id

Edit the quantity of the specific cart item

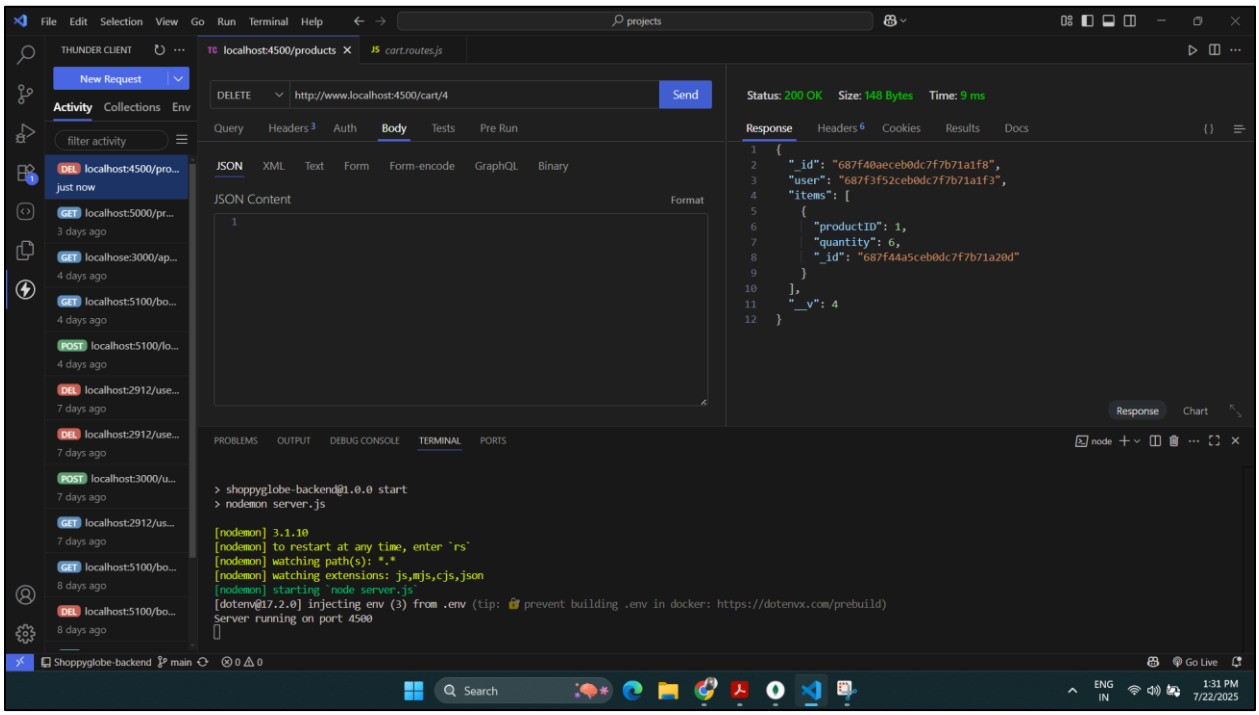


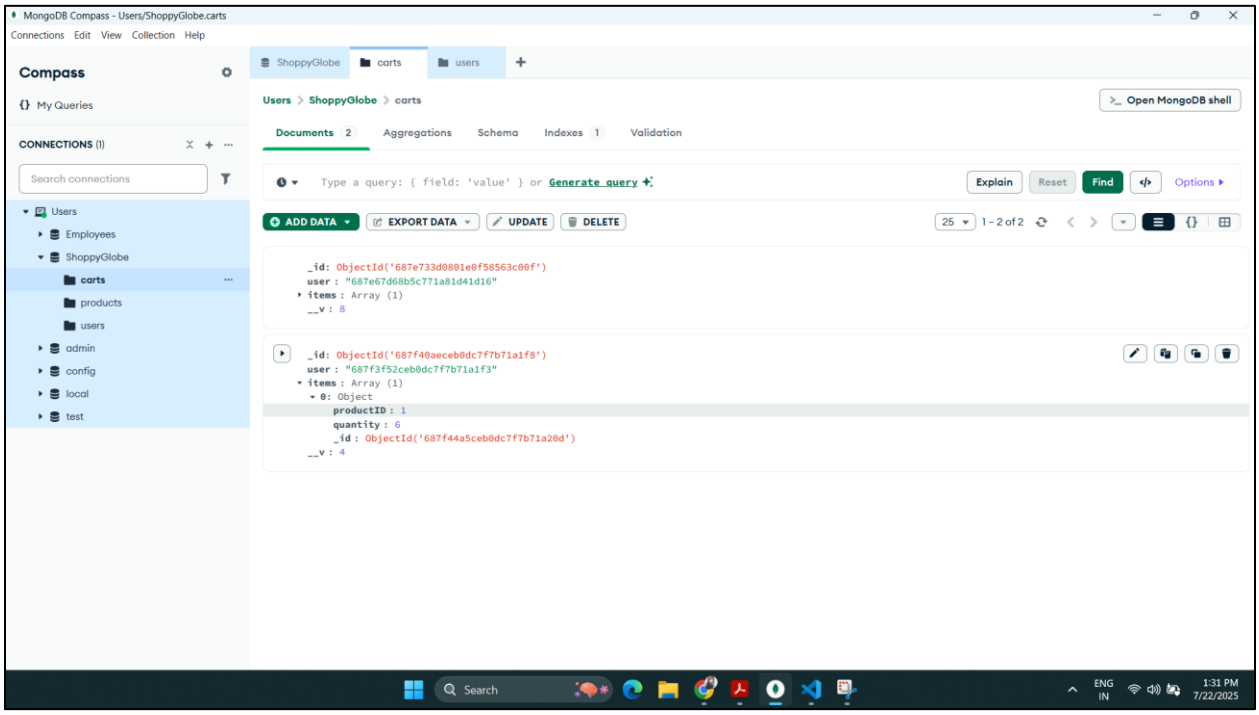


4. Delete document

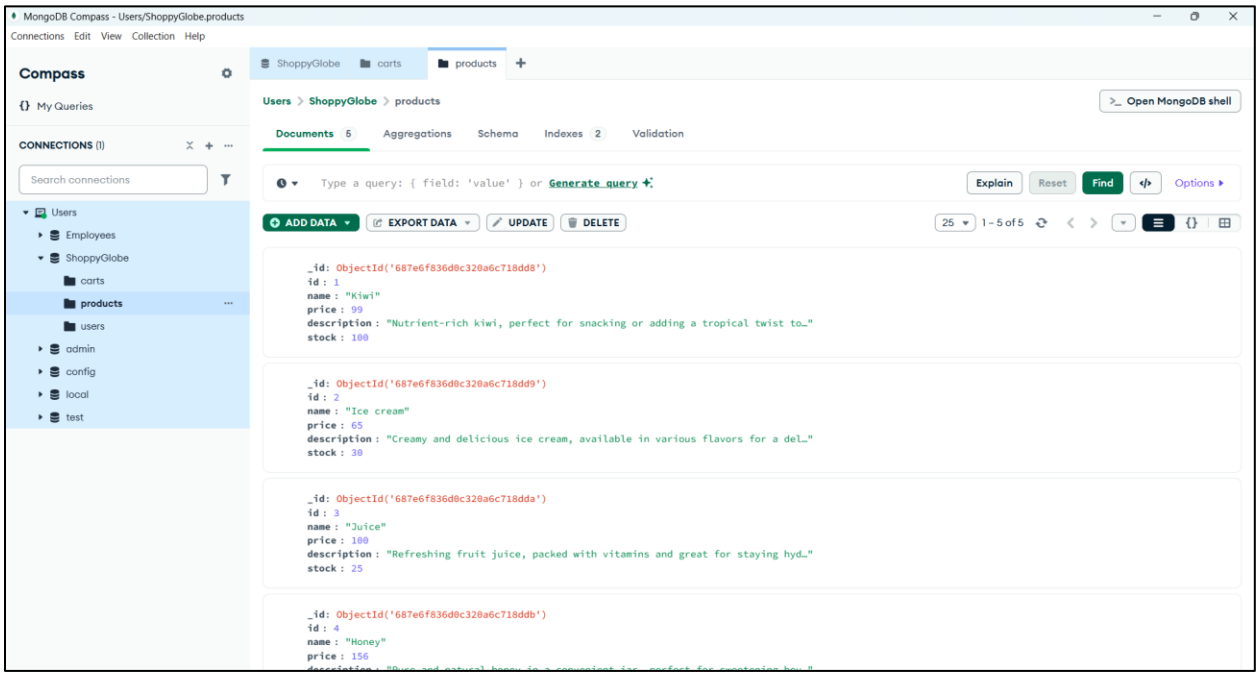
DELETE /cart/:id

Delete the particular item from the cart.





Products Collection:

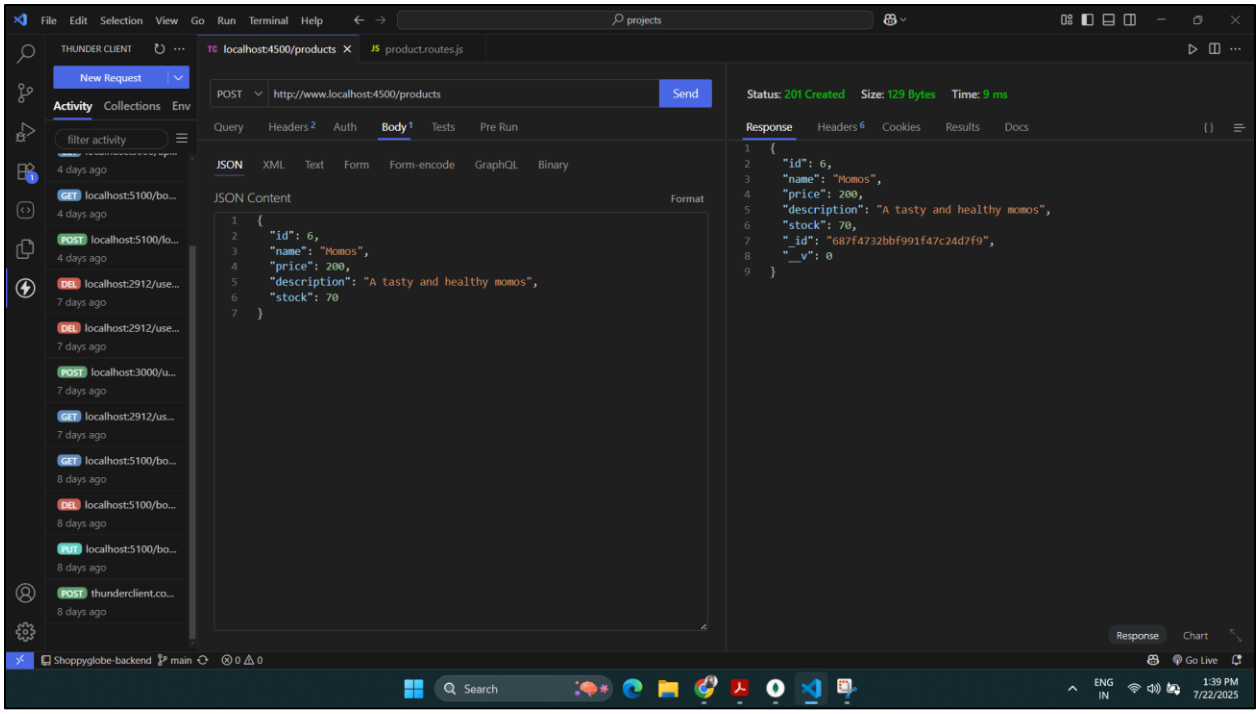


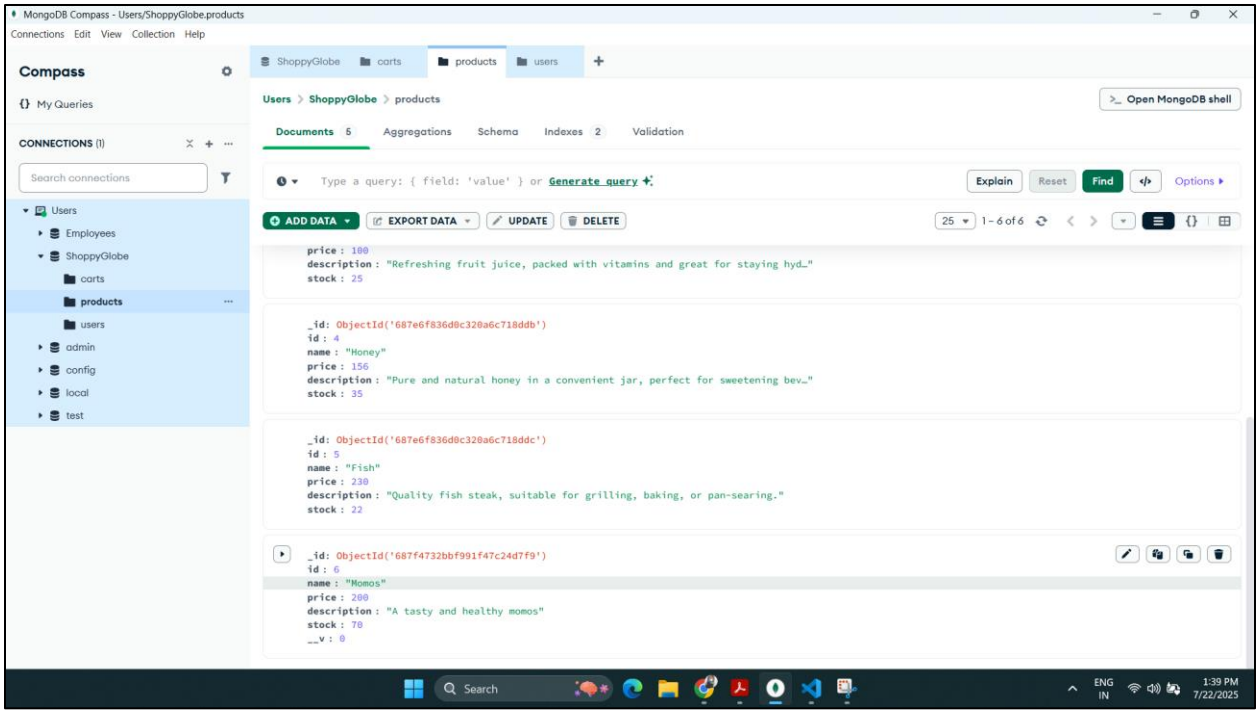
CRUD operation on Products:

1. Create document

POST /products

Create the new product

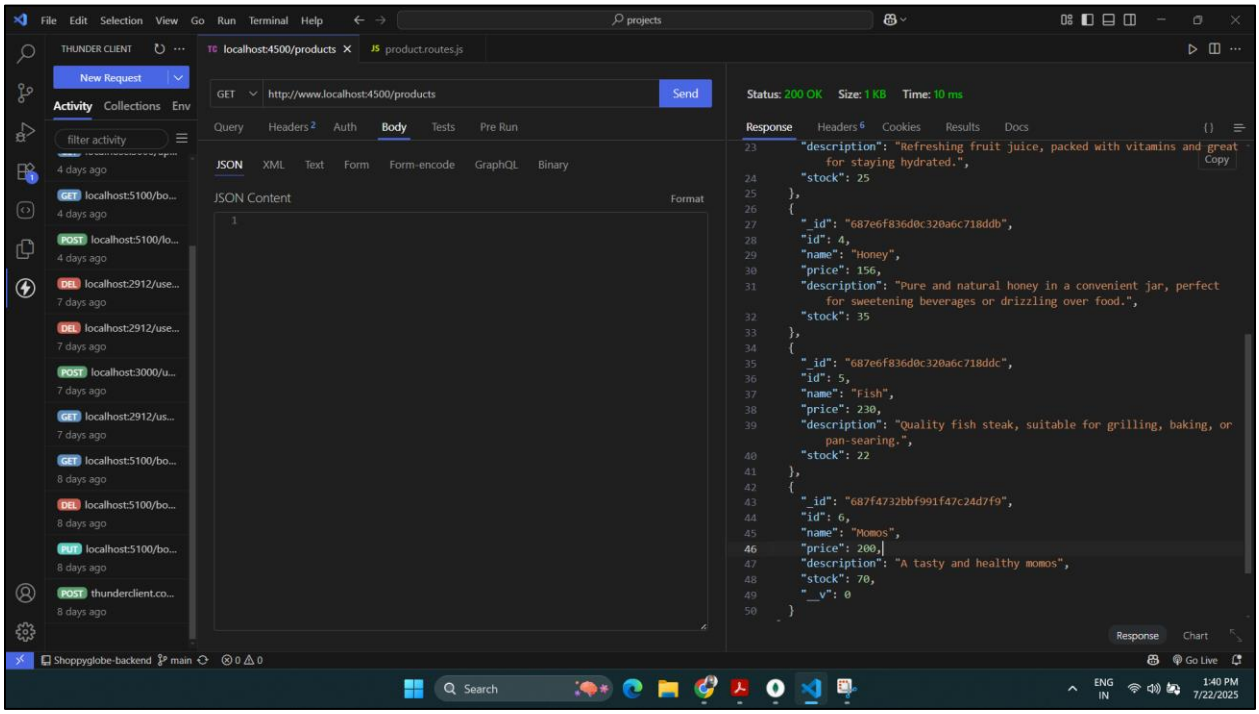




2. Read document

GET /products

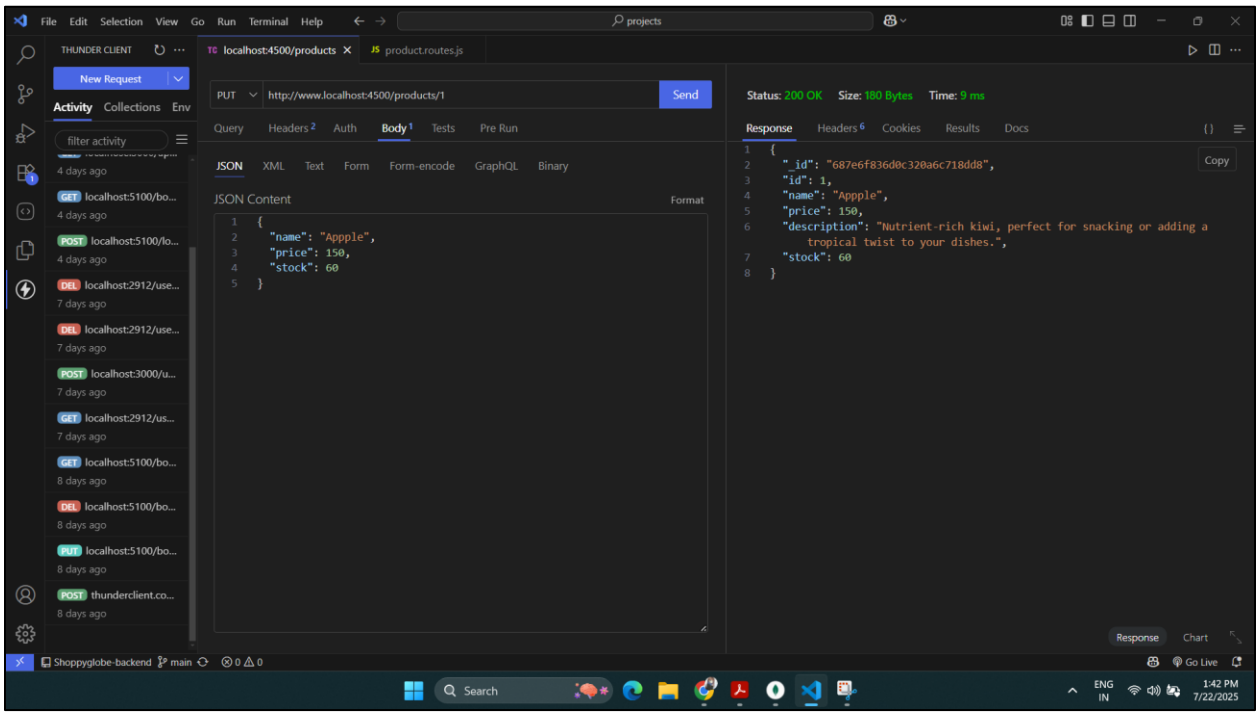
Fetches all the products from the cart

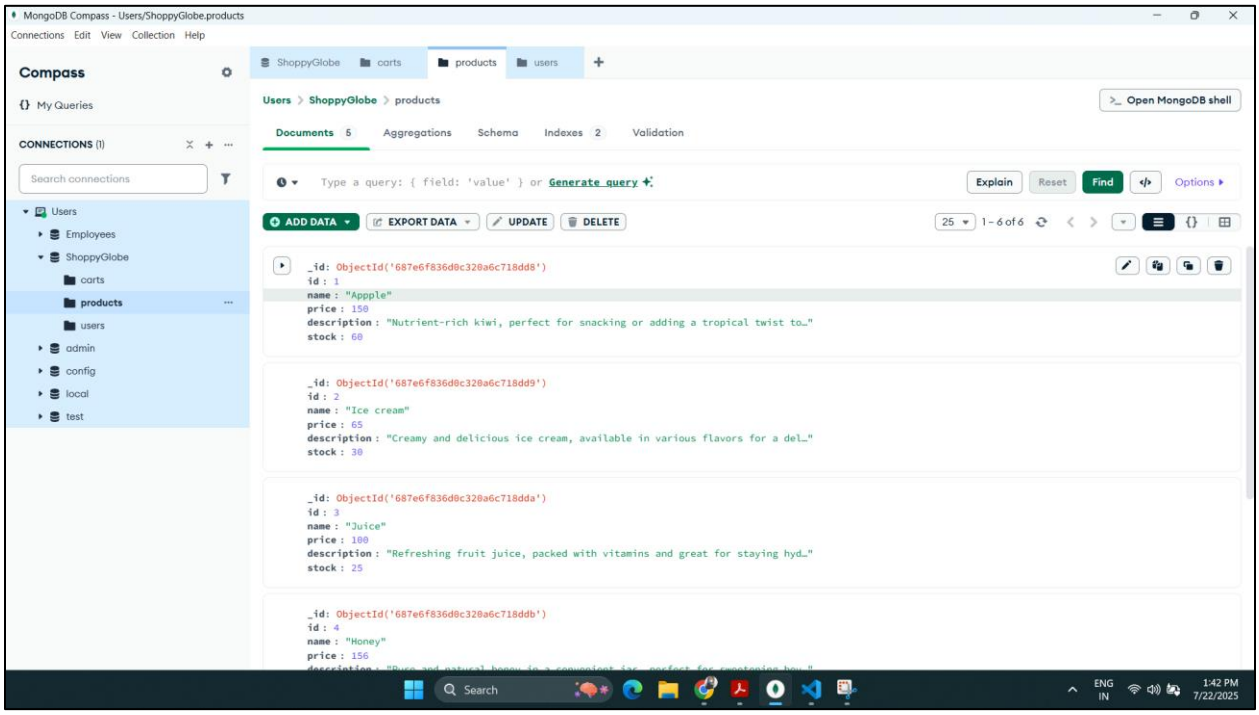


3. Update document

PUT /products/:id

Edit the details of the product

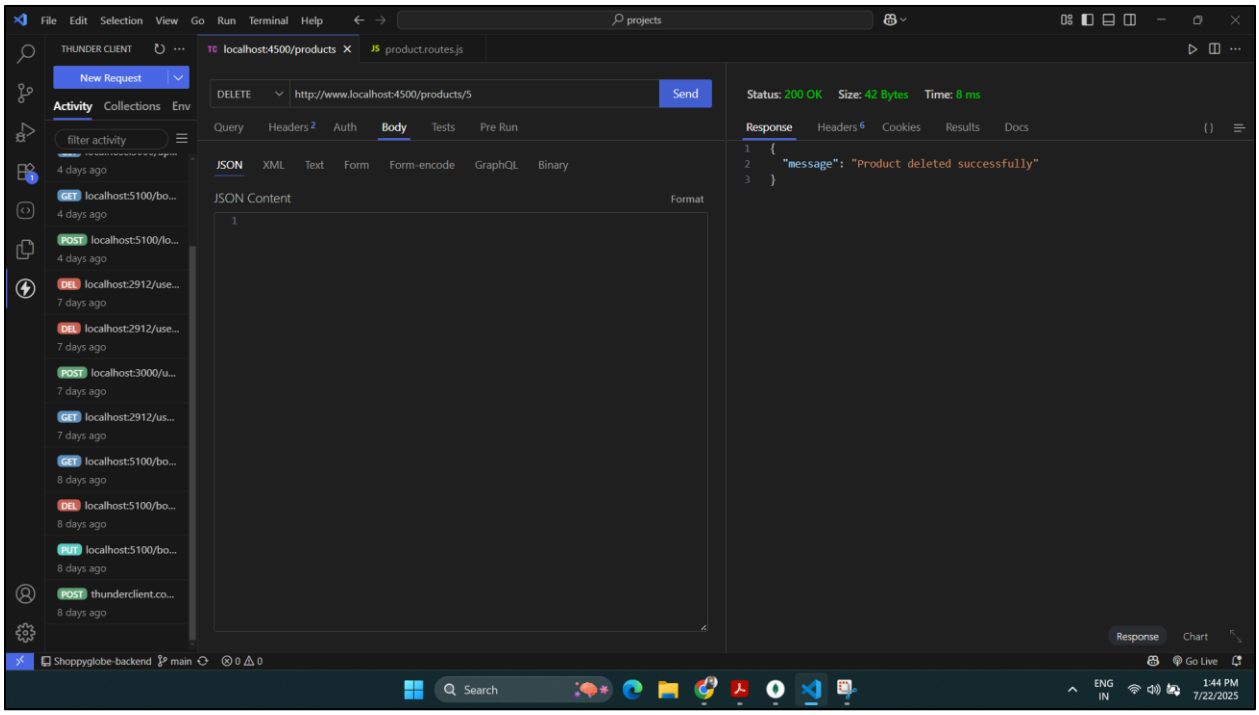




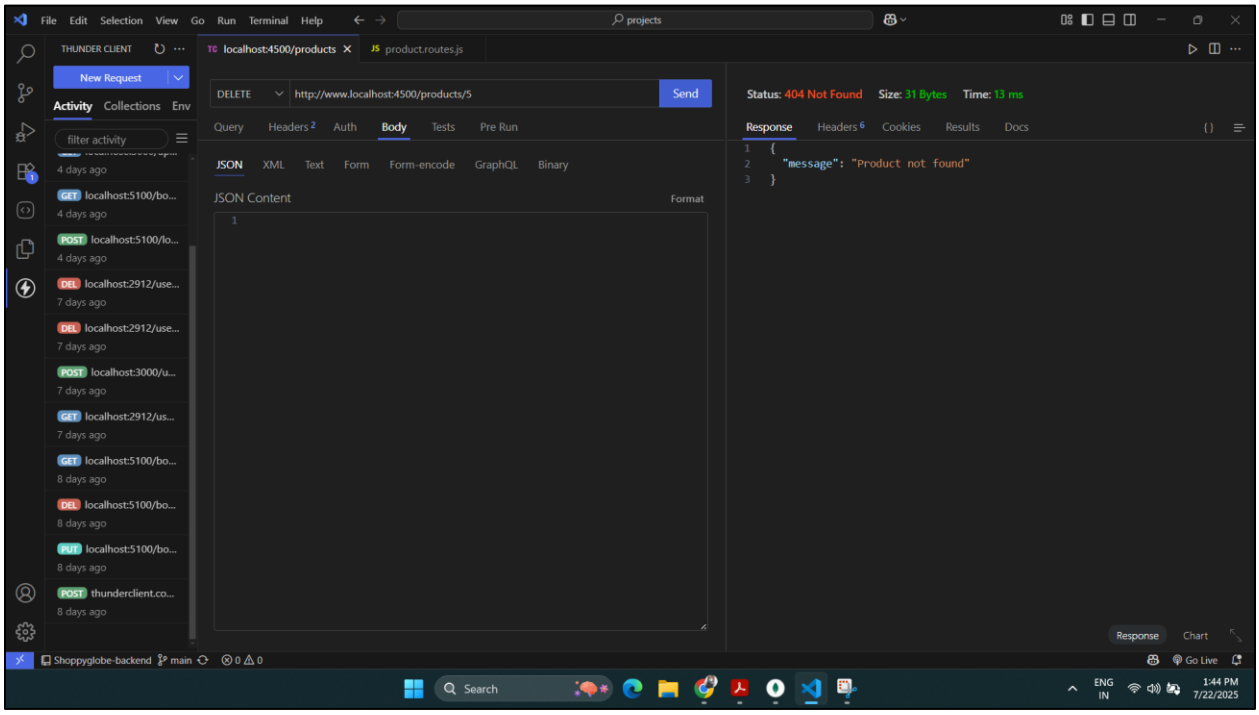
4. Delete document

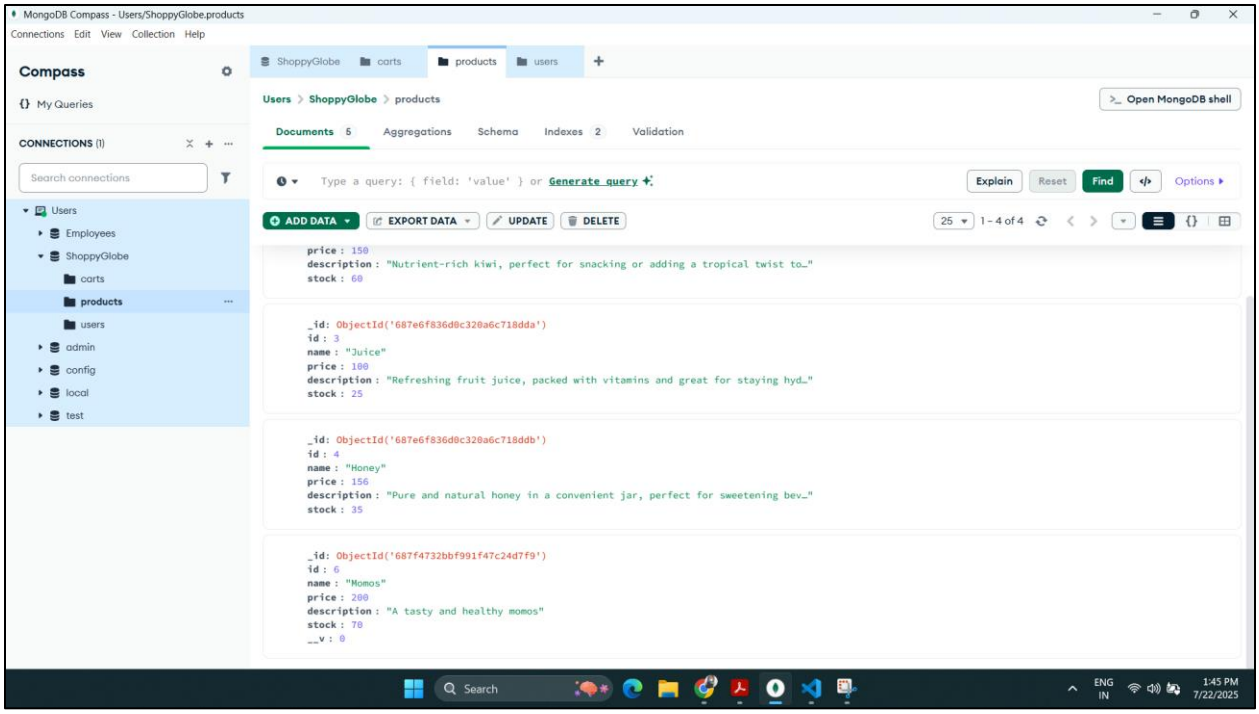
DELETE /products/:id

Deletes the product from DB

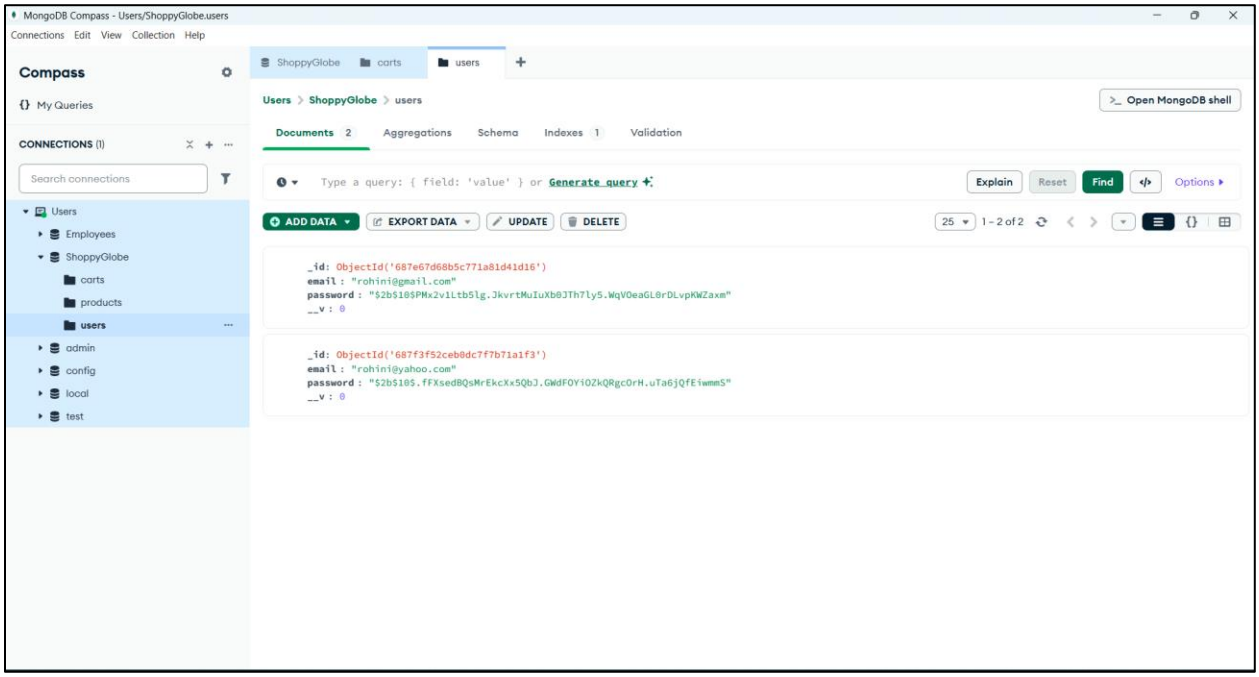


If trying to delete the product that is not available in DB, it returns error.



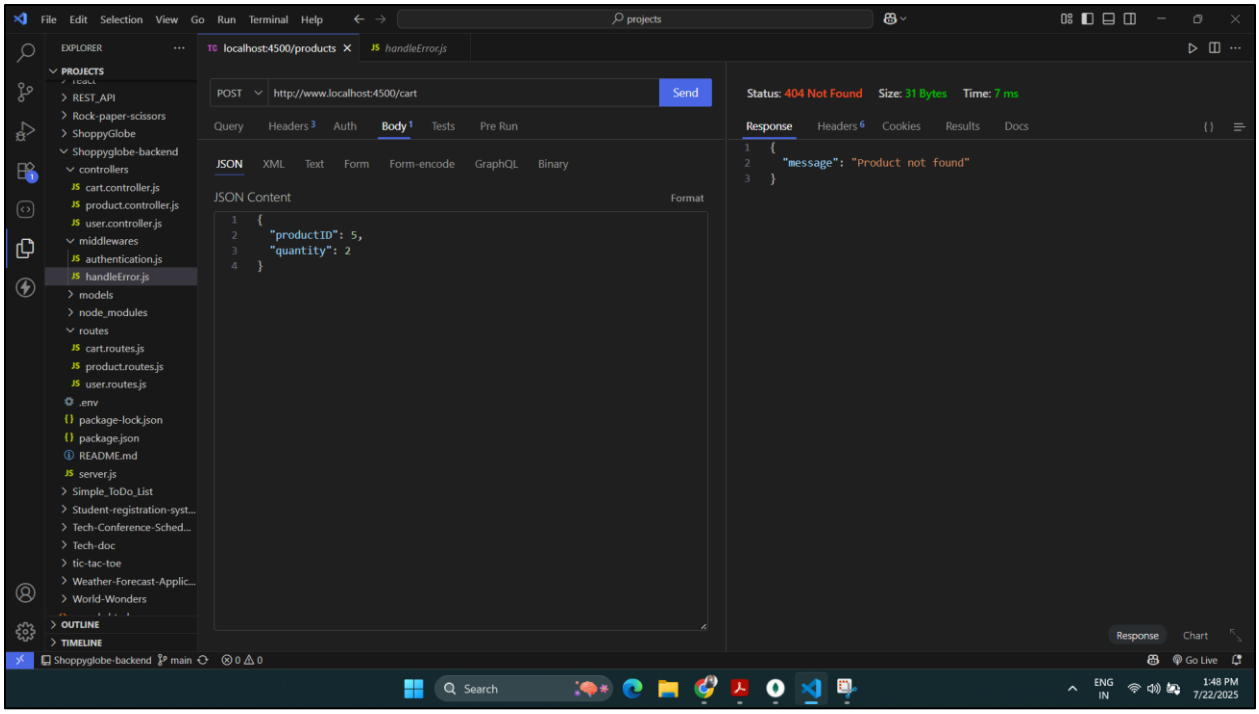


Users Collection:



Cart Validation:

The API call checks if the product is available before adding it to cart. If not present, then it shows an error message.



The product Id - 5 is not present in the DB: