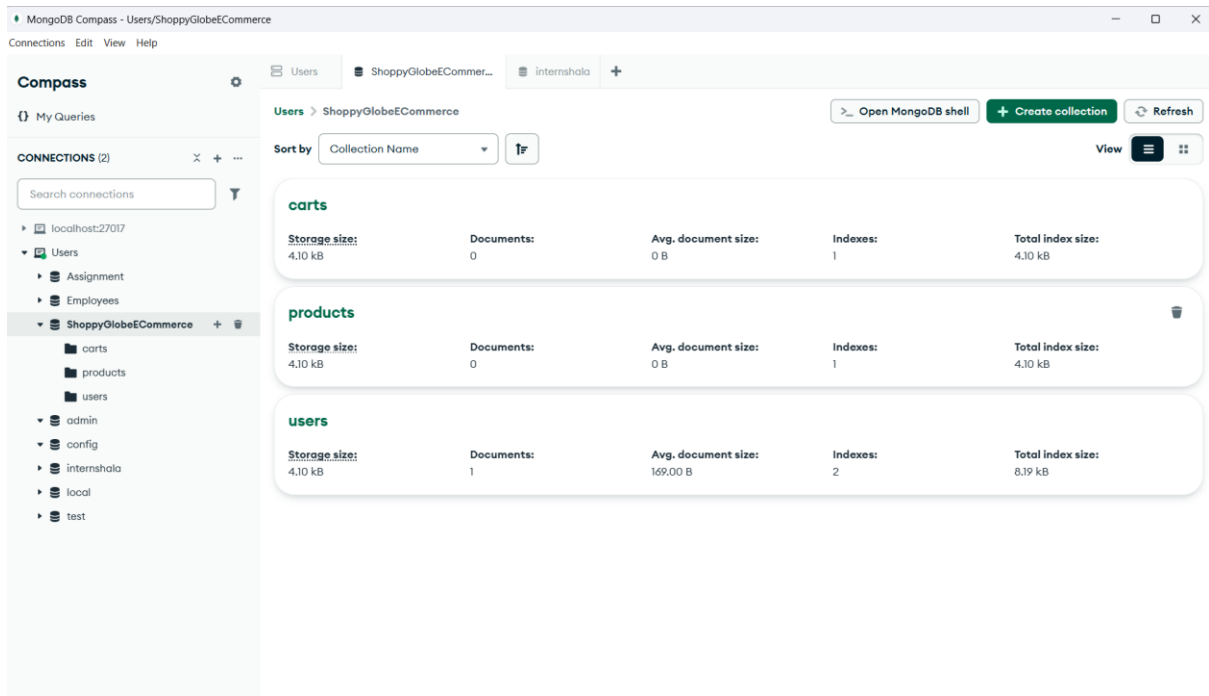


## ShoppYGlobe E-Commerce Project Backend

Here is the Screenshots

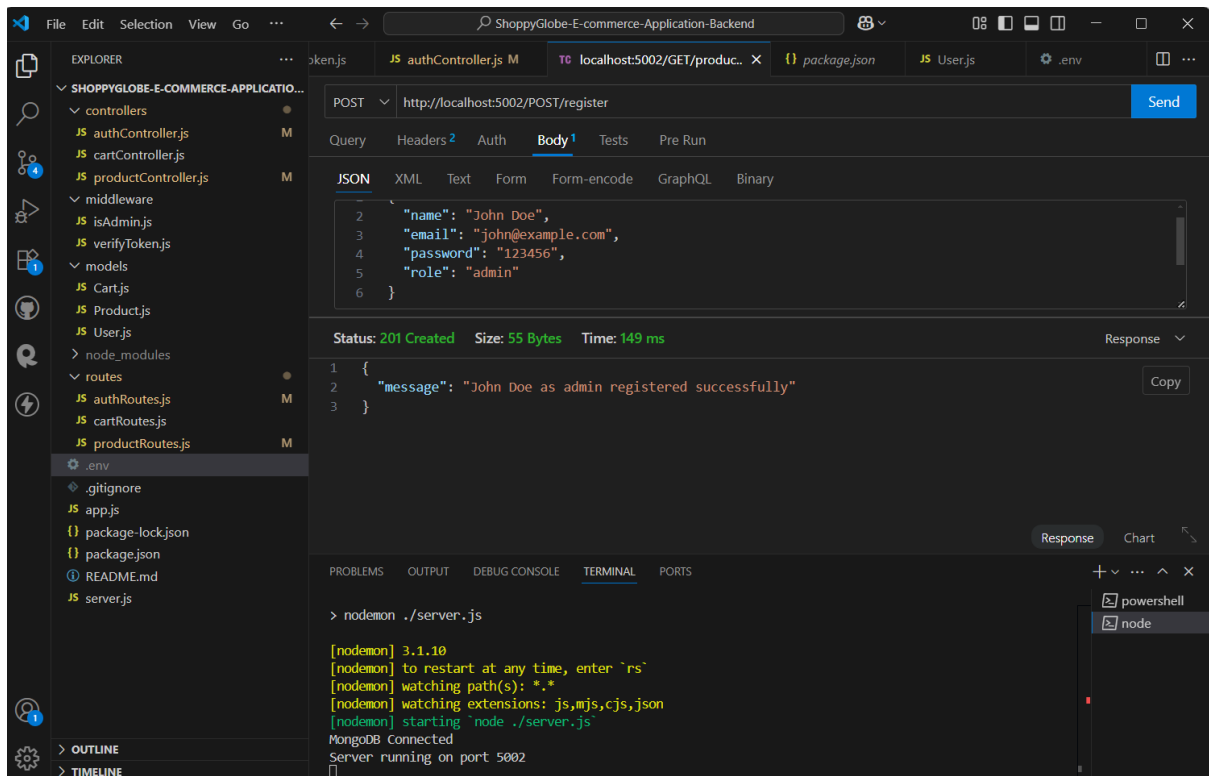
### 1. MongoDB Database Structure

This screenshot displays the ShoppYGlobe database with three main collections: `cart`, `products`, and `users`. These collections store the core data for the application.



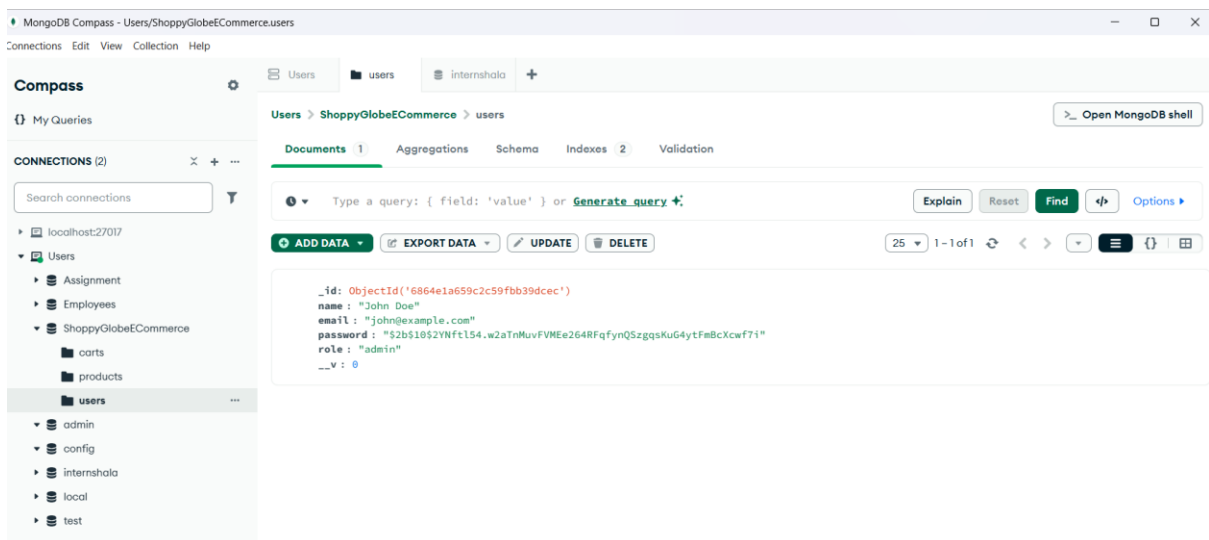
#### Register User – ThunderClient

Used a POST request to register a new user through ThunderClient. This API creates a user account with hashed password and role info.



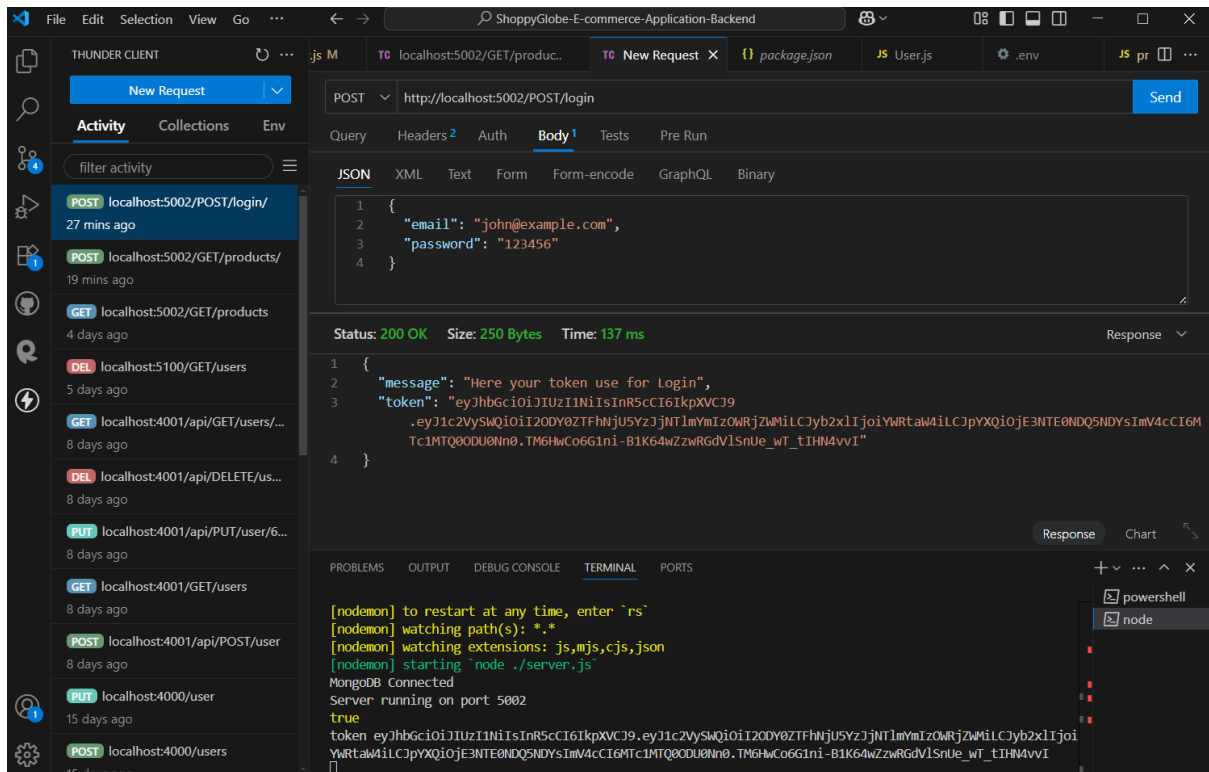
## MongoDB – Registered User Stored

After registration, the new user data is stored in the users collection in MongoDB, confirming backend and DB connectivity.



## Login User – ThunderClient

Used a POST request to login. A valid JWT token is generated in the response upon successful authentication.

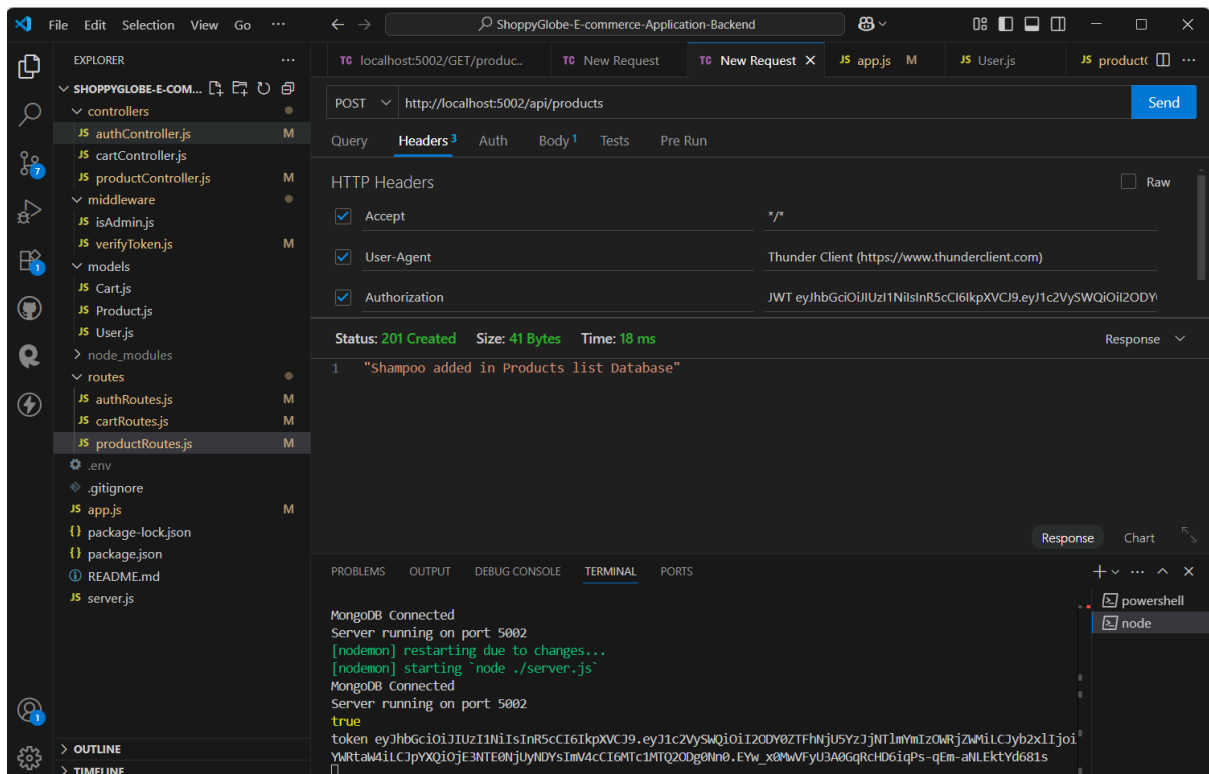


## Add Product – ThunderClient

Used POST /api/products route with token in headers. Product data like name, price, description, and stock is sent in request body.

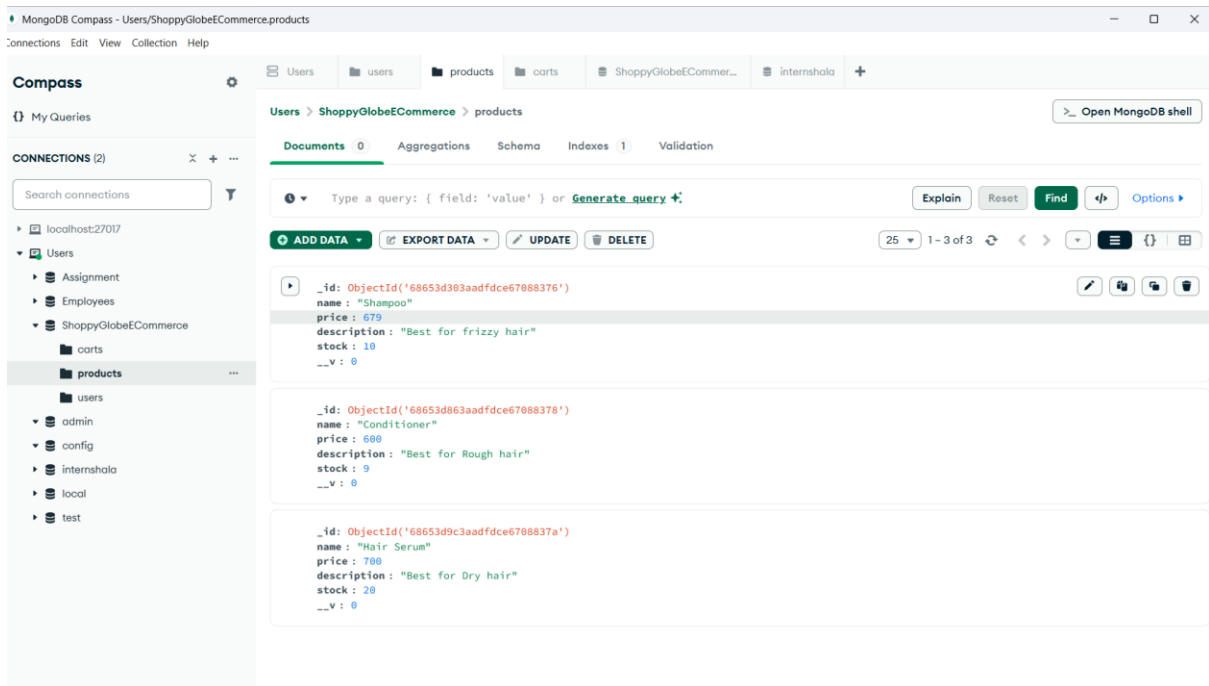
The Body is

```
{  
  "name": "Shampoo",  
  "price": 679,  
  "description": "Best for frizzy hair",  
  "stock": 10  
}
```



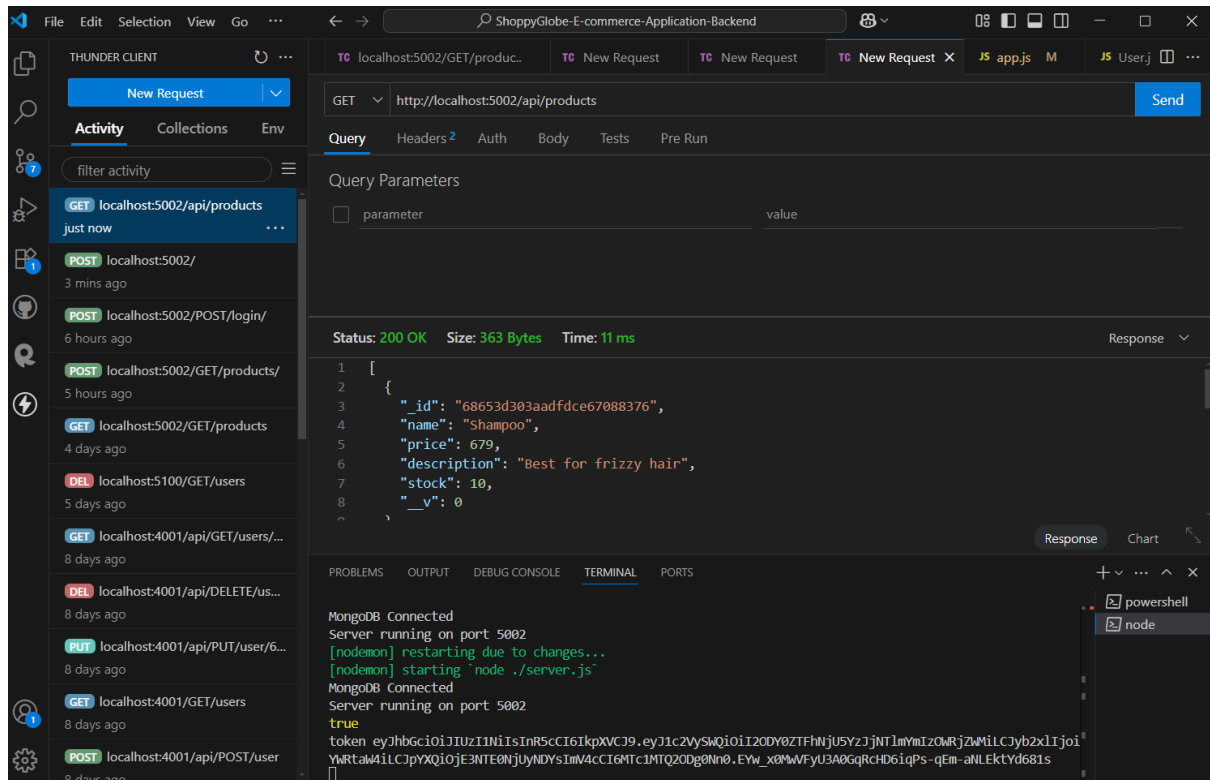
## MongoDB – Product Stored

The products collection now shows the newly added product “Shampoo”, confirming the create operation was successful.



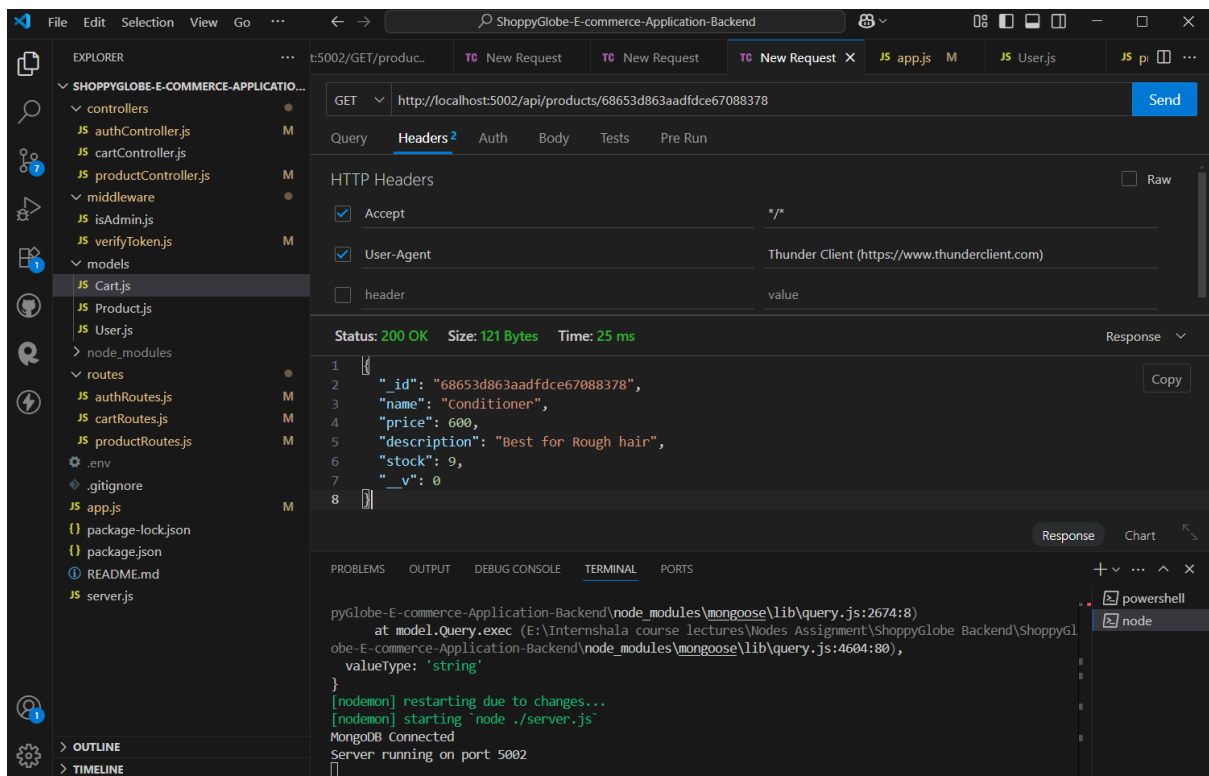
## Get All Products – ThunderClient

Sent a GET request to `/api/products` to retrieve the full list of products stored in the MongoDB collection.



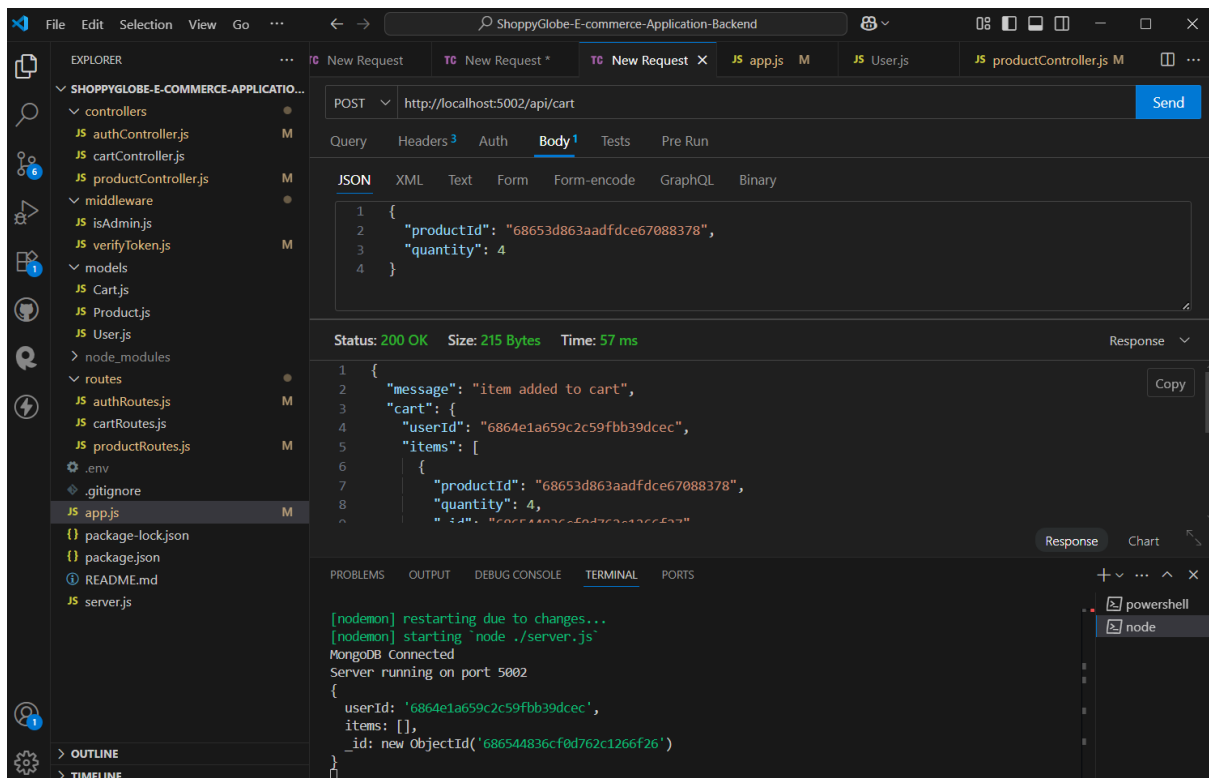
## Get Product by ID – ThunderClient

Used GET /api/products/:id to fetch a single product by its MongoDB ObjectId. This demonstrates the Read operation for individual entries.



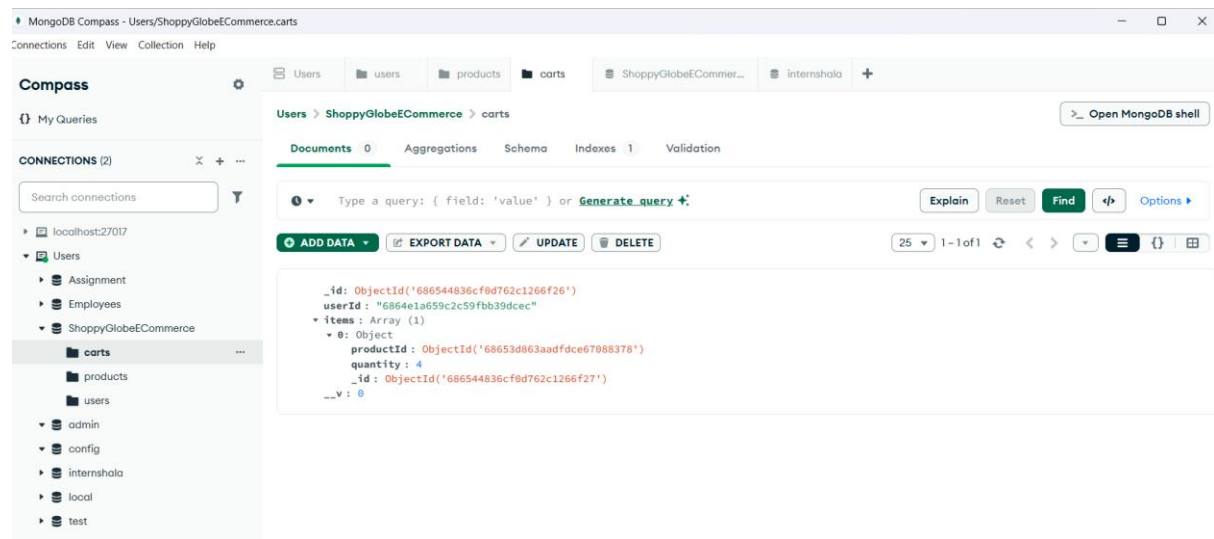
## Add to Cart – ThunderClient

Used POST /cart with product ID and quantity in the body. Token is passed in headers for authentication. Product added to cart successfully.



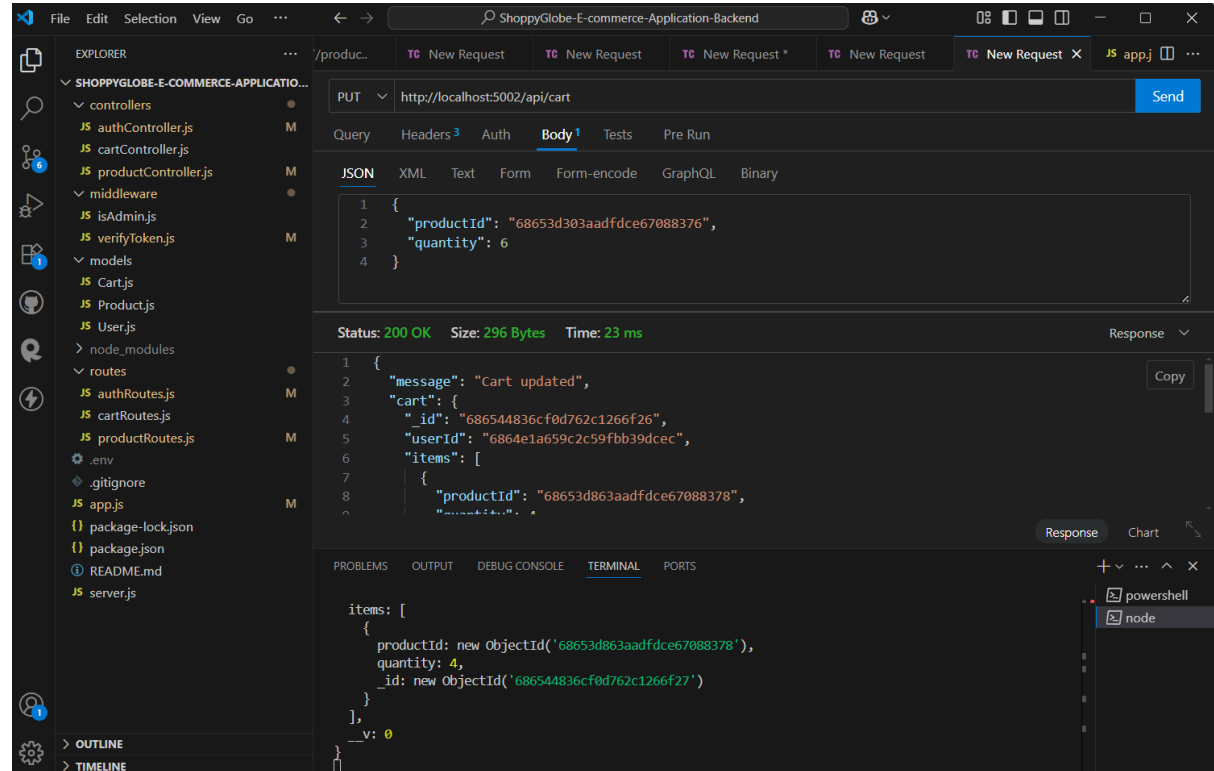
## MongoDB – Cart Data Stored

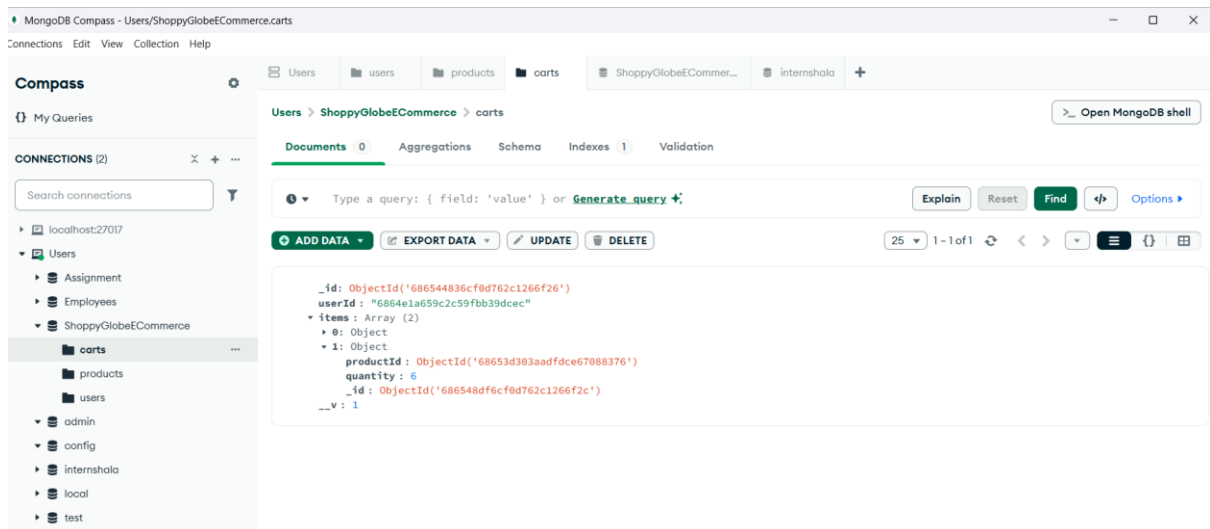
The cart collection reflects the user's cart with the selected product ID and quantity mapped to their user ID.



## Update Cart – ThunderClient

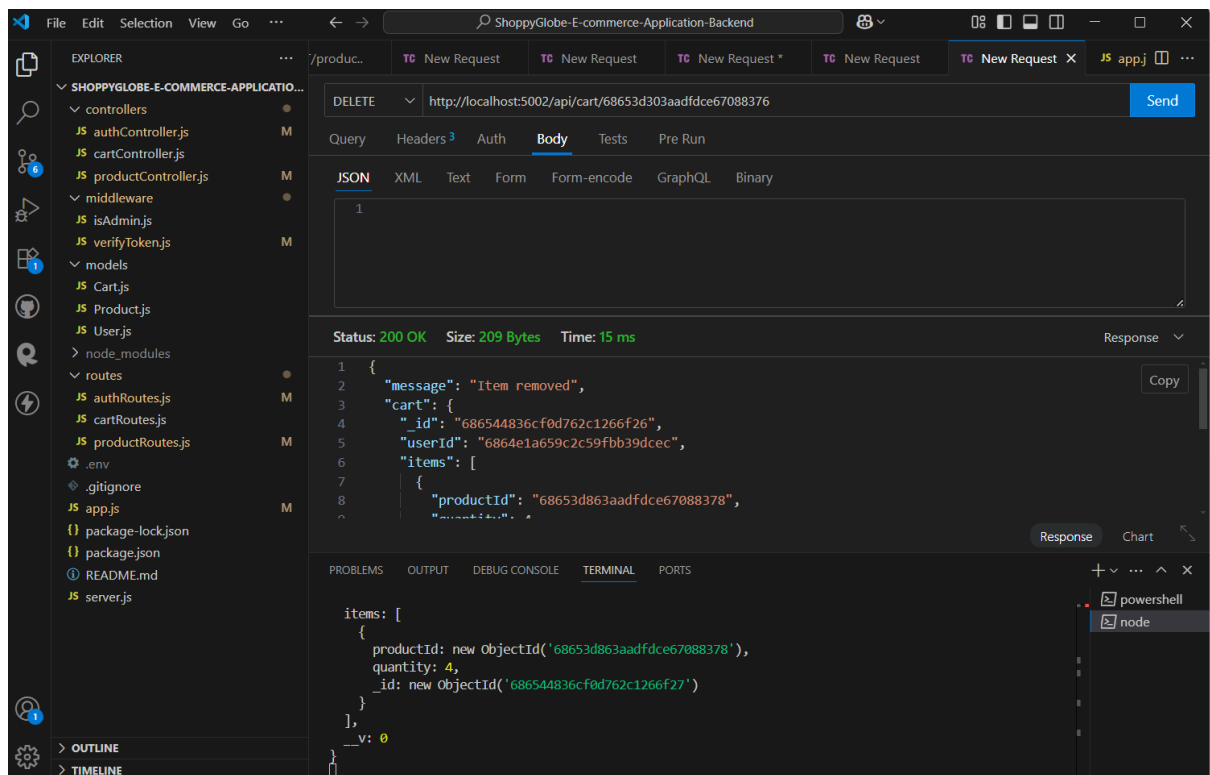
Sent a PUT /cart request with updated quantity in body. This modifies the quantity of an existing product in the user's cart.





## Delete Cart Item – ThunderClient

Used DELETE /cart/:productId route to remove a product from the cart. The cart is updated and reflects the changes.





MongoDB Compass - Users/ShopyGlobeECommerce.carts

ConnectionsEditViewCollectionHelp

Compass

{ } My Queries

CONNECTIONS (2)

Search connections

localhost:27017

Users

Assignment

Employees

ShopyGlobeECommerce

carts

products

users

admin

config

internshala

local

test

Users

users

products

carts

ShopyGlobeECommerce...

internshala

+

Users > ShopyGlobeECommerce > carts

Documents 0

Aggregations

Schema

Indexes 1

Validation

Type a query: { field: 'value' } or [Generate query](#)

ExplainResetFindOptions

ADD DATA

EXPORT DATA

UPDATE

DELETE

251 - 1 of 1

```
{
  "_id": ObjectId('686544836cf8d762c1266f26'),
  "userId": "6864e1a659c2c59fbb39dcec",
  "items": Array (1)
  0: Object
    productId: ObjectId('68653d863aadfde67088378')
    quantity: 4
    _id: ObjectId('686544836cf8d762c1266f27')
  __v: 2
}
```