API TESTING AND MONGODB SCREENSHOTS

**NOTE : Please use the given URL only as path for testing as these paths are used in the project , otherwise it will give error.**

1. **User Registration:**

* URL: http://localhost:5000/api/auth/register
* Method: POST
* **Body (JSON)**: {

"name": "Test User",

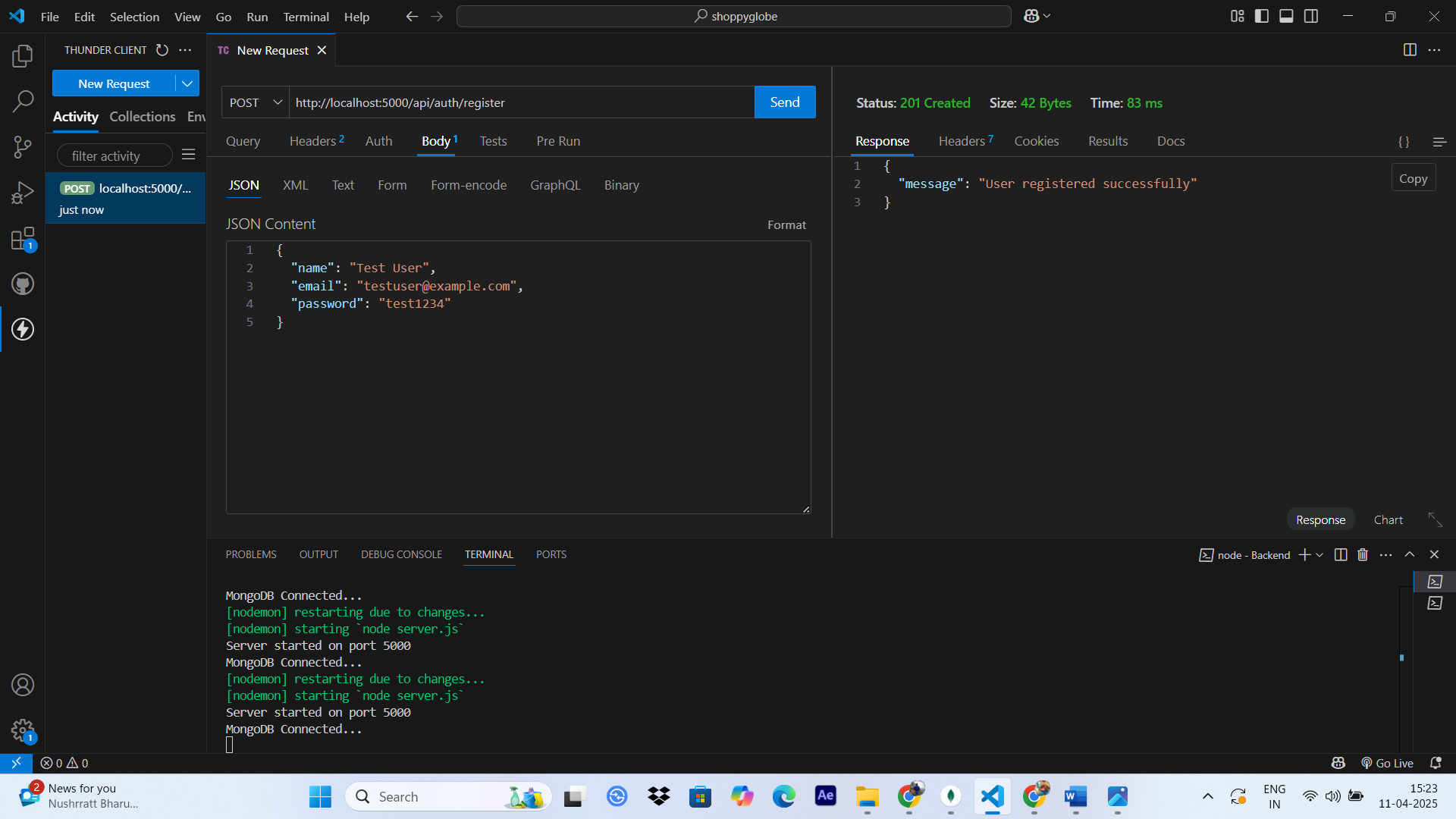
"email": "testuser@example.com",

"password": "test1234"

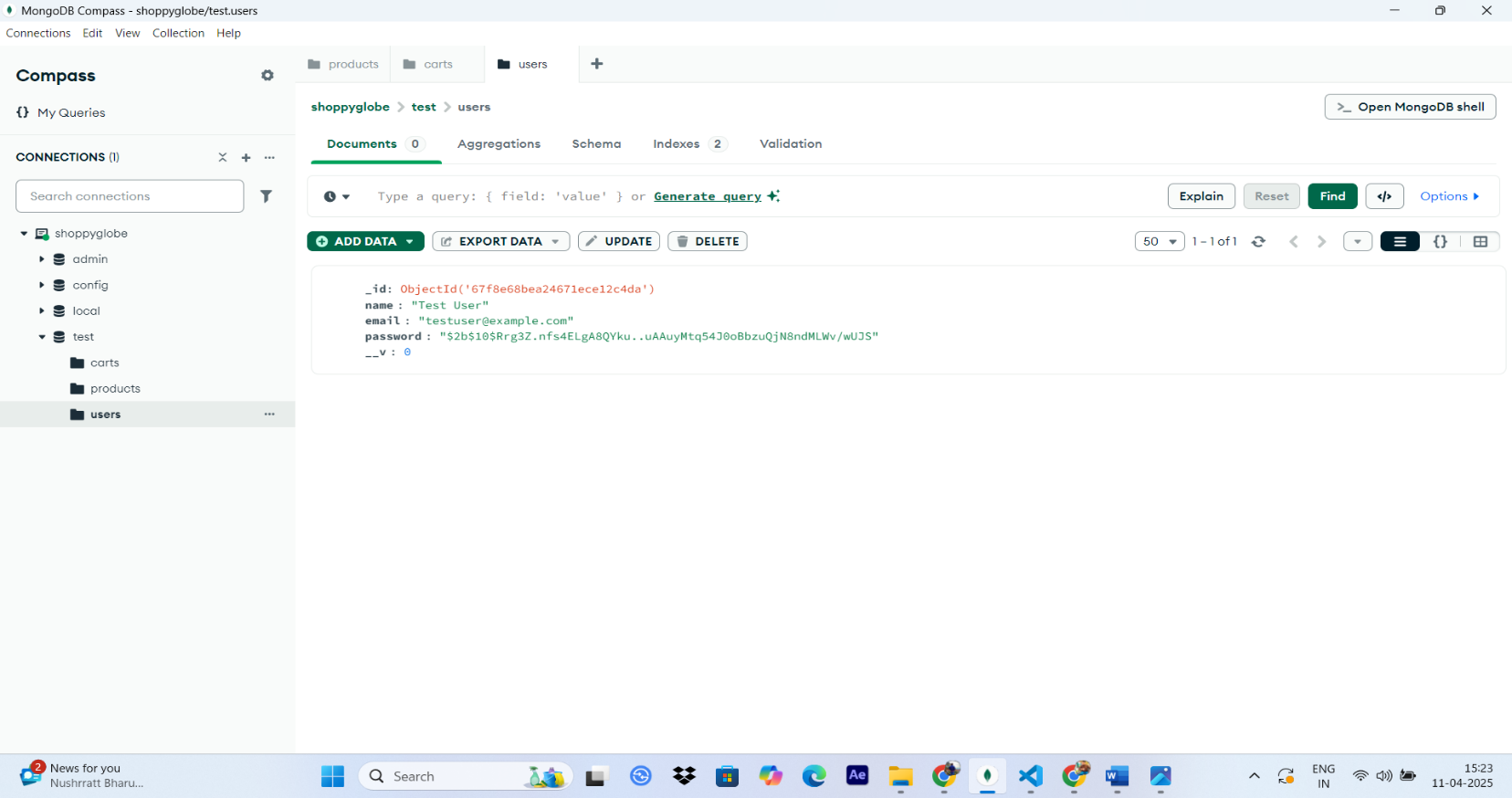
}

* **Expected Output:** User registered and added to collection users

**SCREENSHOTS:**

****

* **Once the User gets Registered a document is added in the Users collection in the database below is the screenshot**

****

1. **User Login :**

* URL: http://localhost:5000/api/auth/login
* Method: POST
* **Body (JSON):**

{

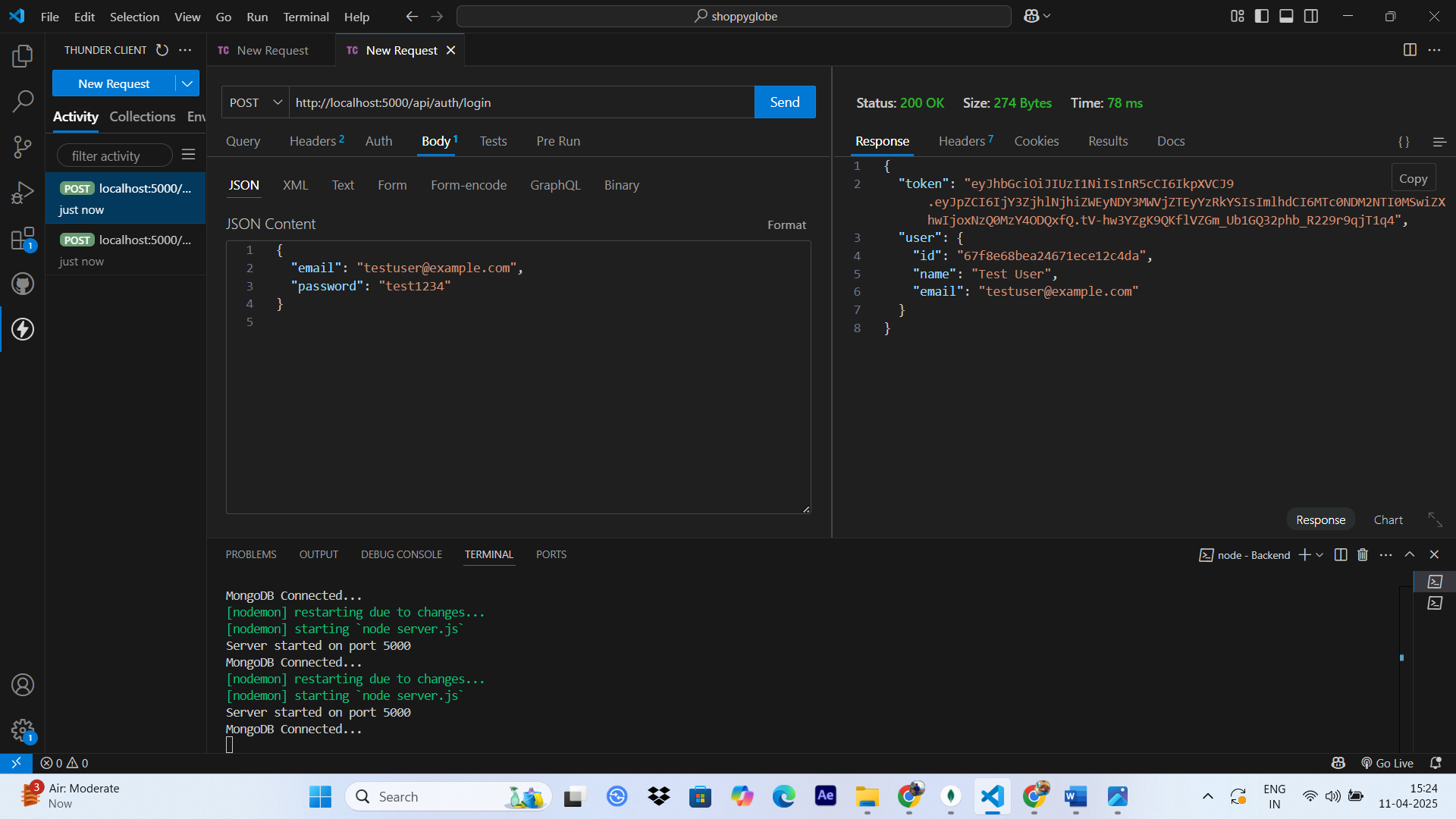
"email": "testuser@example.com",

"password": "test1234"

}

* **Expected Output:** As user login a token is generated as seen in the image now this token will be used in next steps as all other operations are protected .

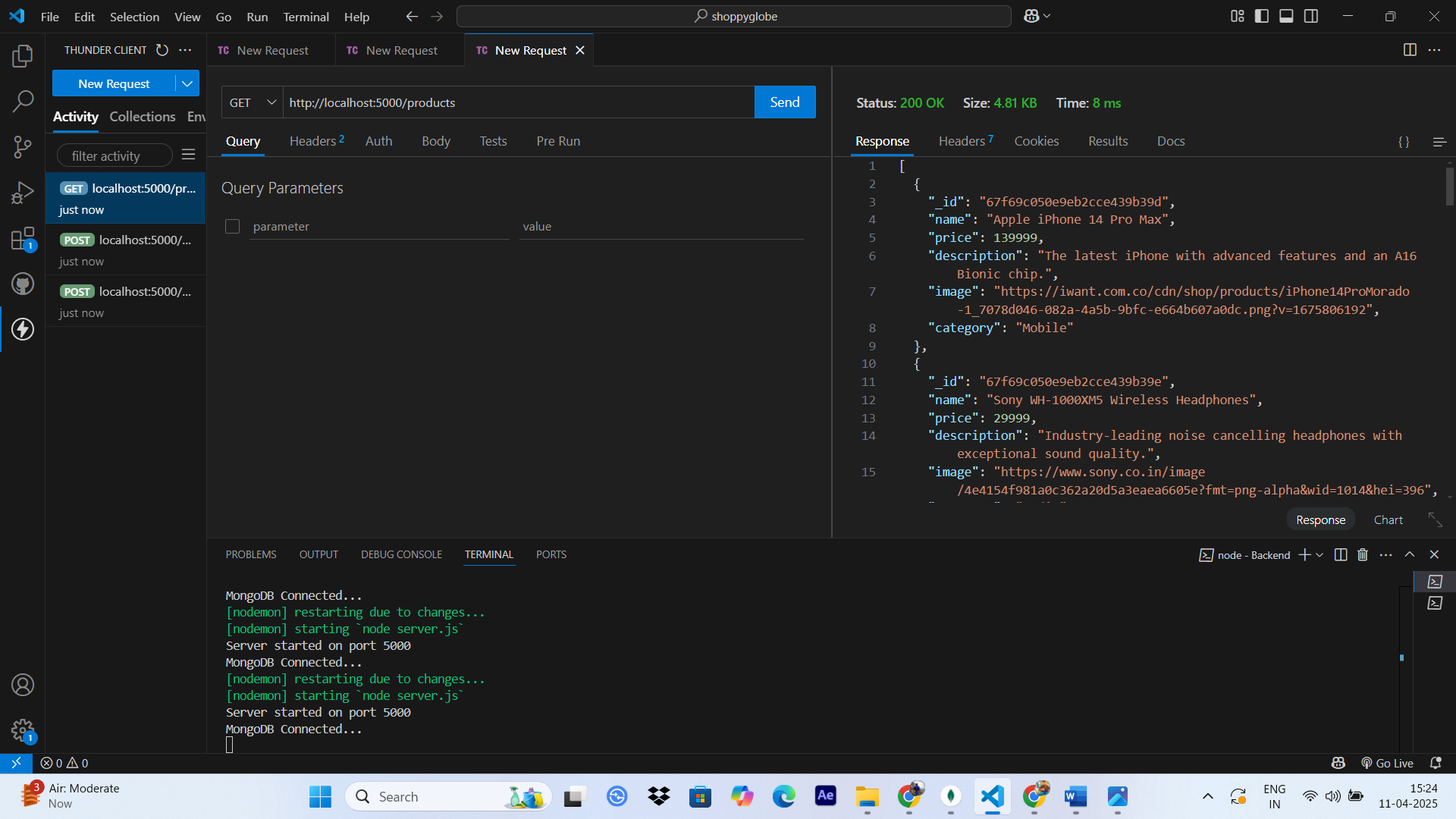
**SCREENSHOTS :**

****

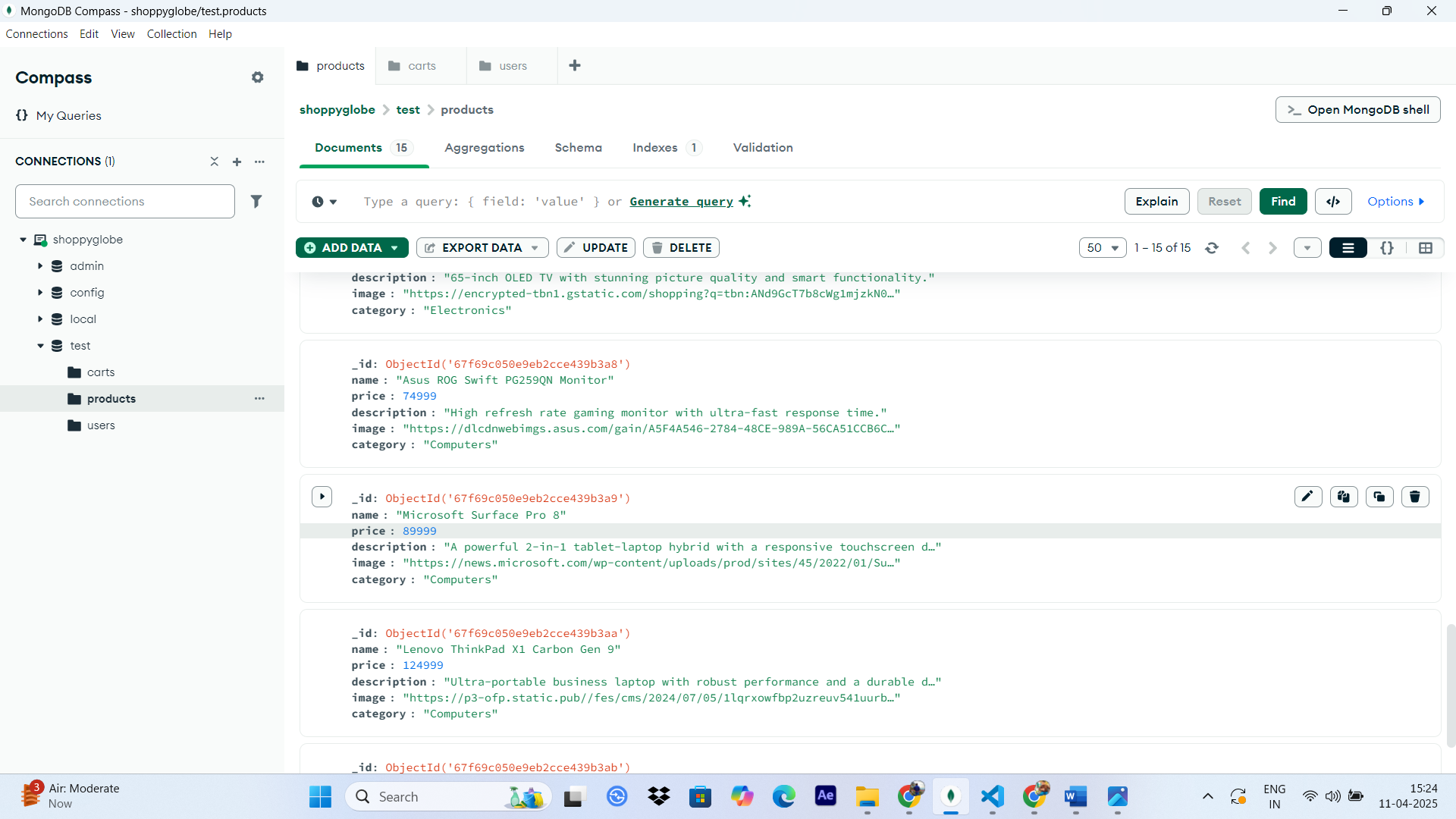
1. **GET All Products :**

* URL: http://localhost:5000/products
* Method: GET
* **Expected Output:** We get all the products in the products collection in Database as response.

**SCREENSHOT :**

****

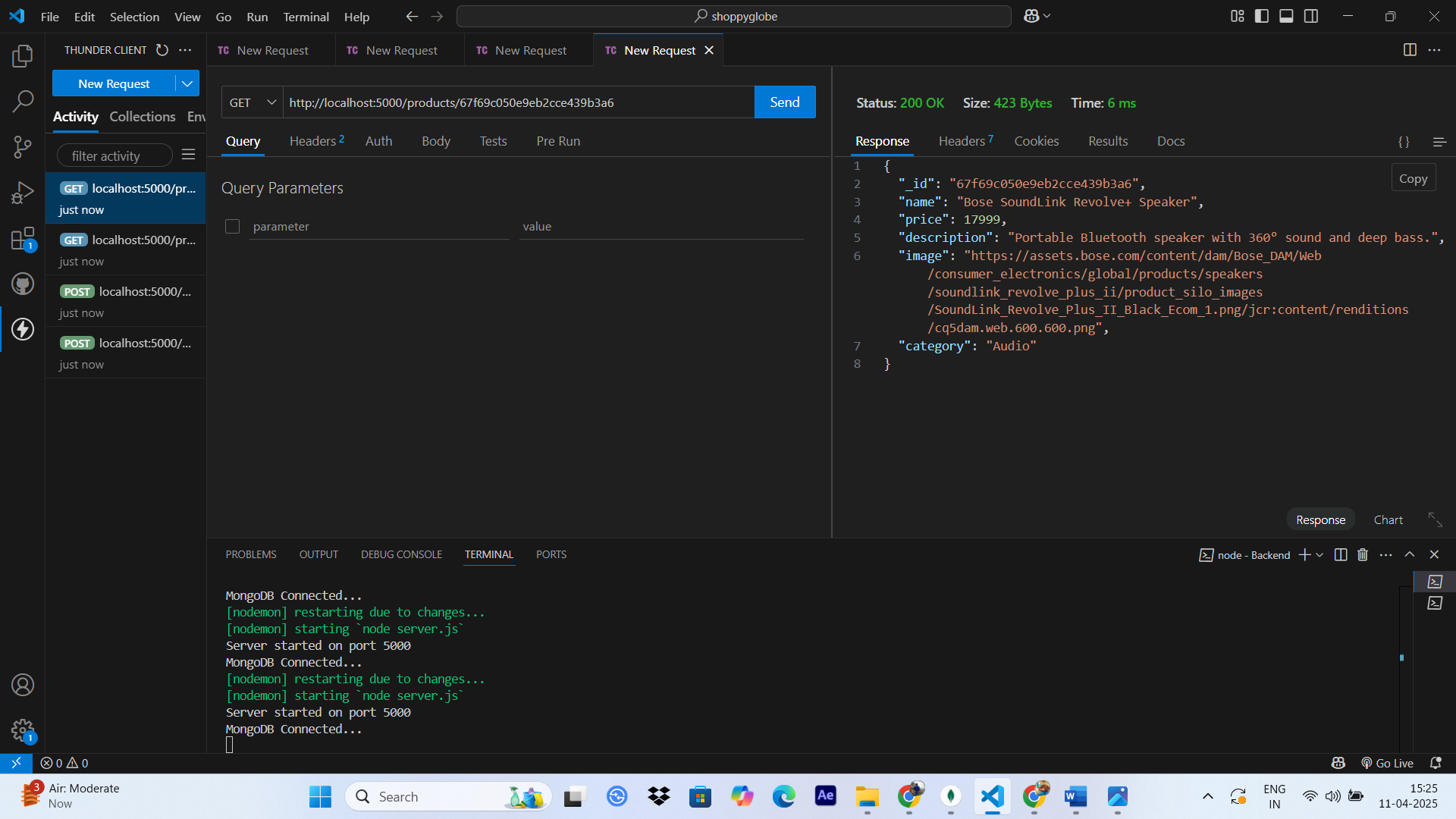
* Here is the Database screenshots showing all the products in the products collection which we fetched in above image using GET .



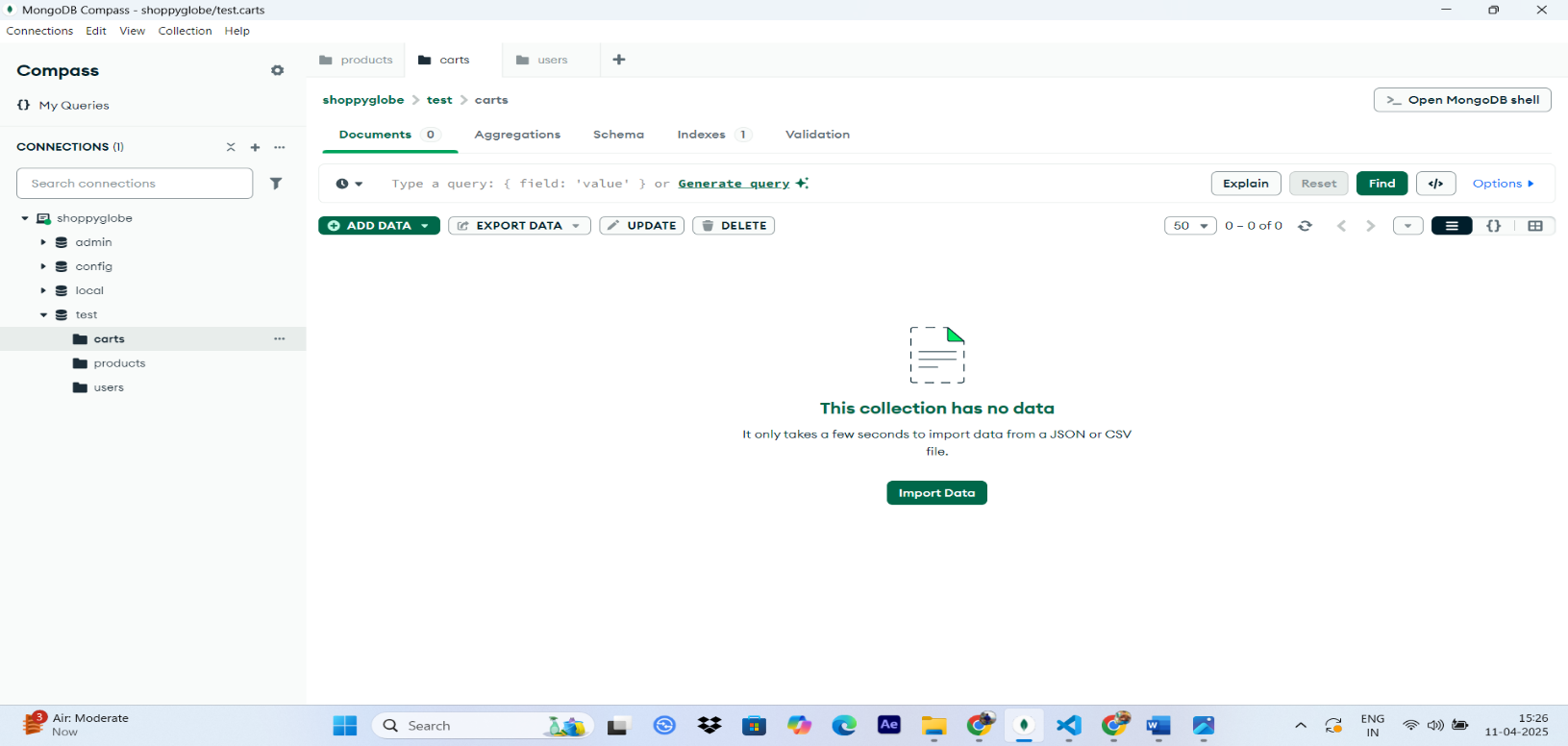
1. **GET Single Product By ID :**

* URL: http://localhost:5000/products/<product\_id>
* Method: GET
* **Expected Output:** We get the specific product of matching id from products collection in Database as response.

**Screenshot :**

****

* **Initially our Cart is empty :**

****

1. **POST Add to Cart:**

* URL: http://localhost:5000/cart
* Method: POST
* **Body (JSON):**

{

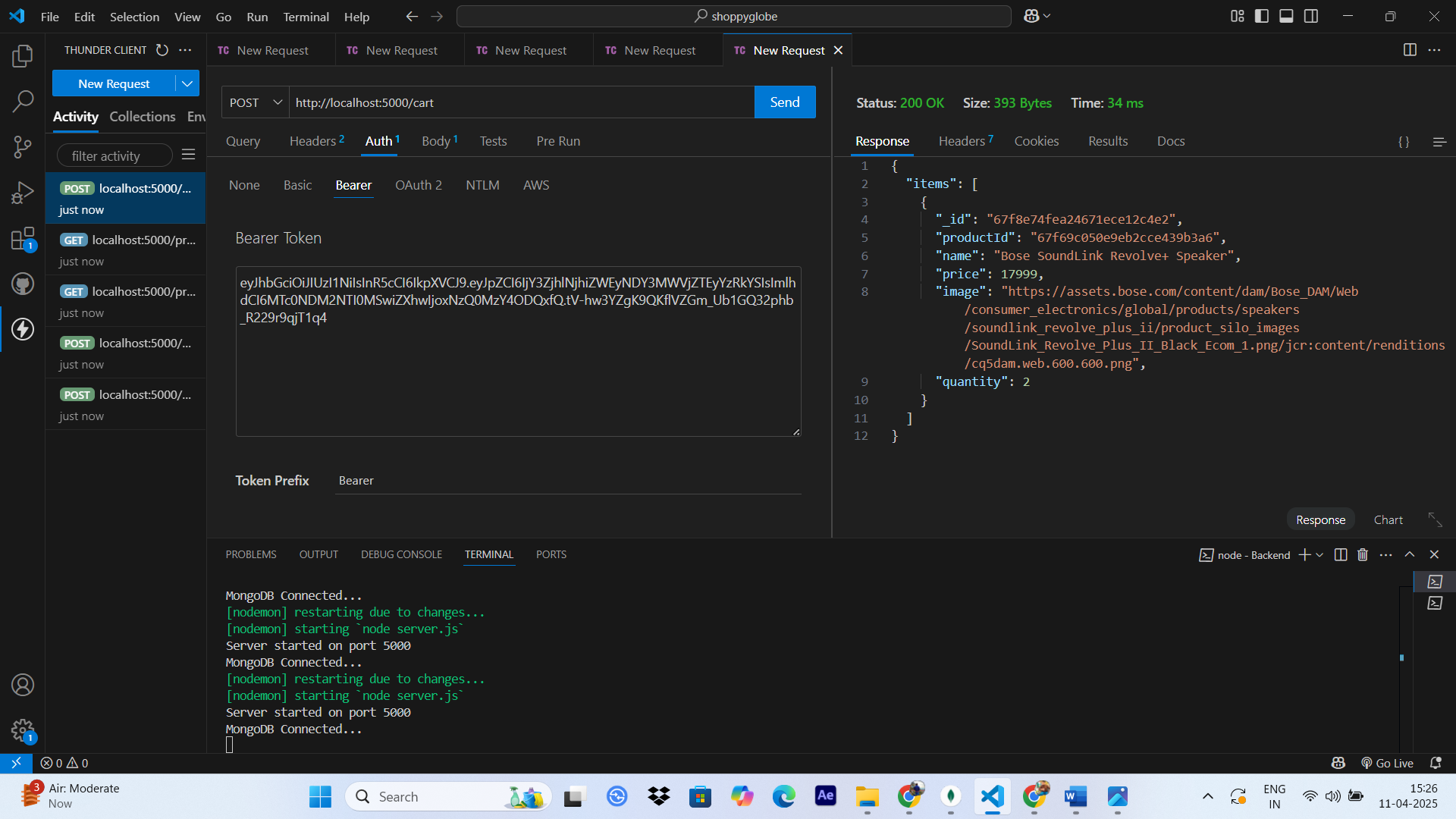
"productId": "<real\_product\_id>",

"quantity": 2

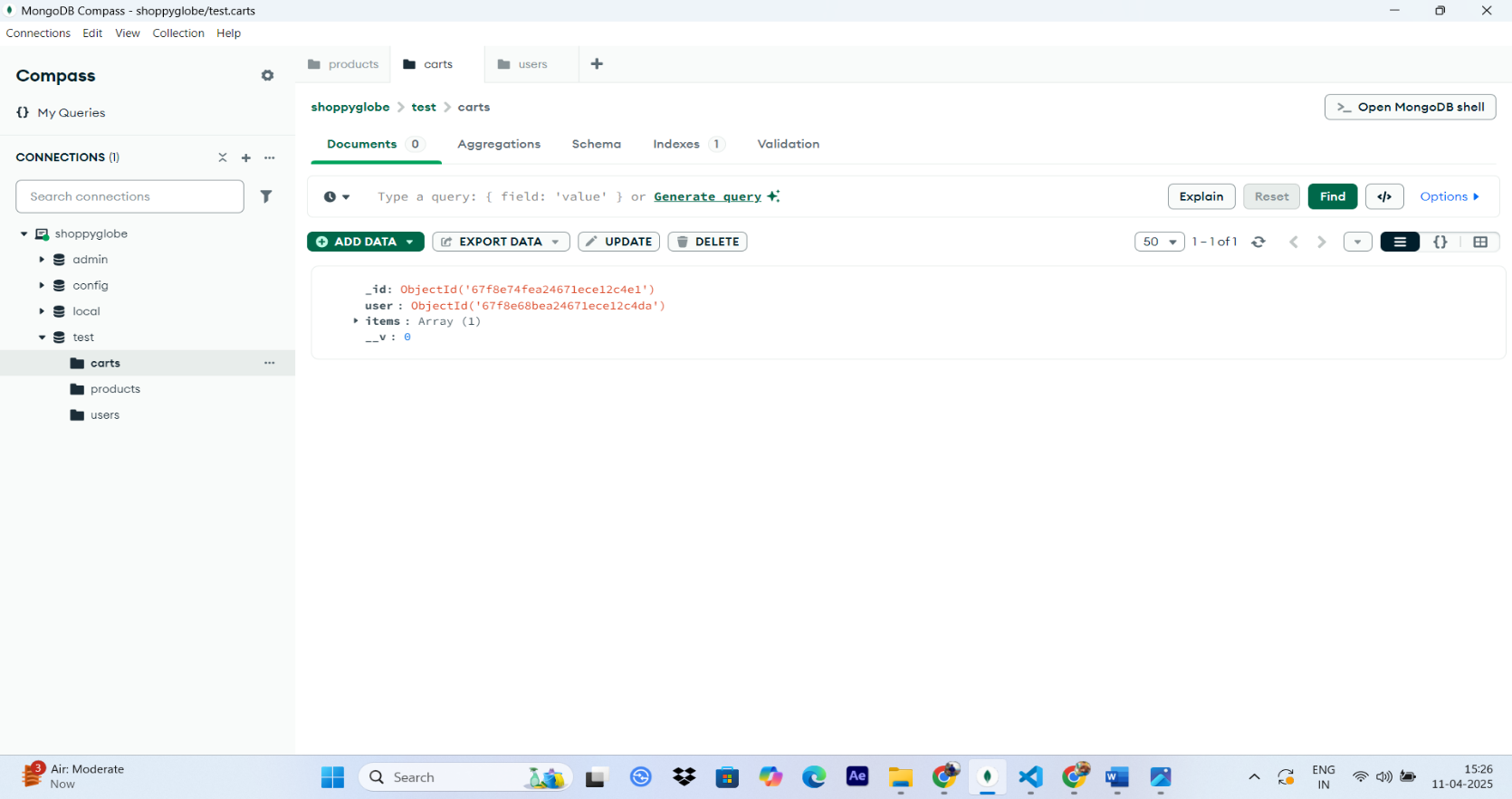
}

* AUTH TOKEN : IN Auth section of Thunder Client under Bearer add the token generated at the time of login.
* **Expected Output:** The specified product is added in the cart collection in Database.

**SCREENSHOT :**

****

* **The specified item is added to the Cart**

****

* **Cart collection is updated with the item in database.**

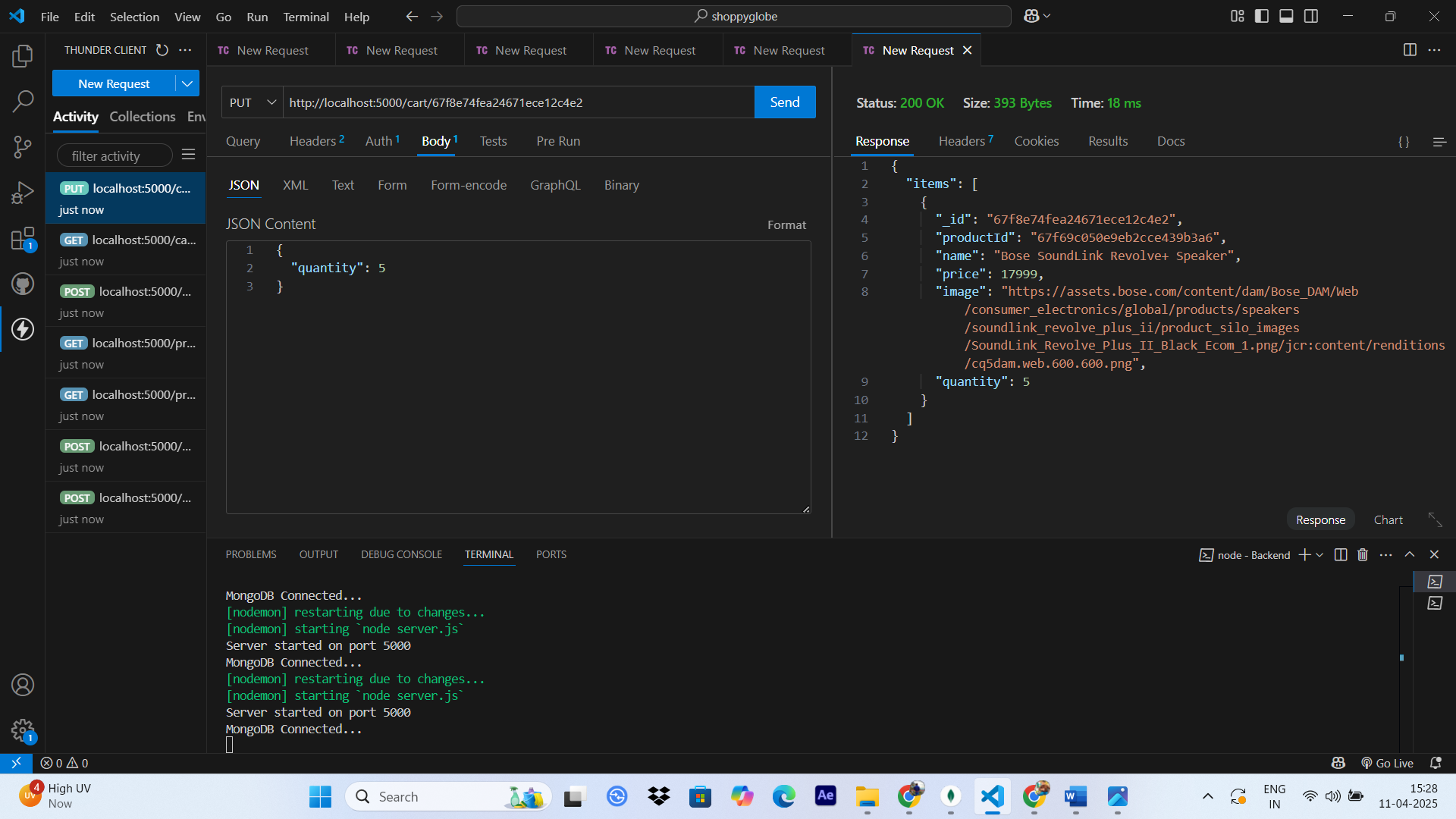
1. **PUT UPDATE Cart Item Quantity:**
   * URL: http://localhost:5000/cart/<cart\_item\_id>
   * Method: PUT
   * **BODY(JSON) :** {

"quantity": 5

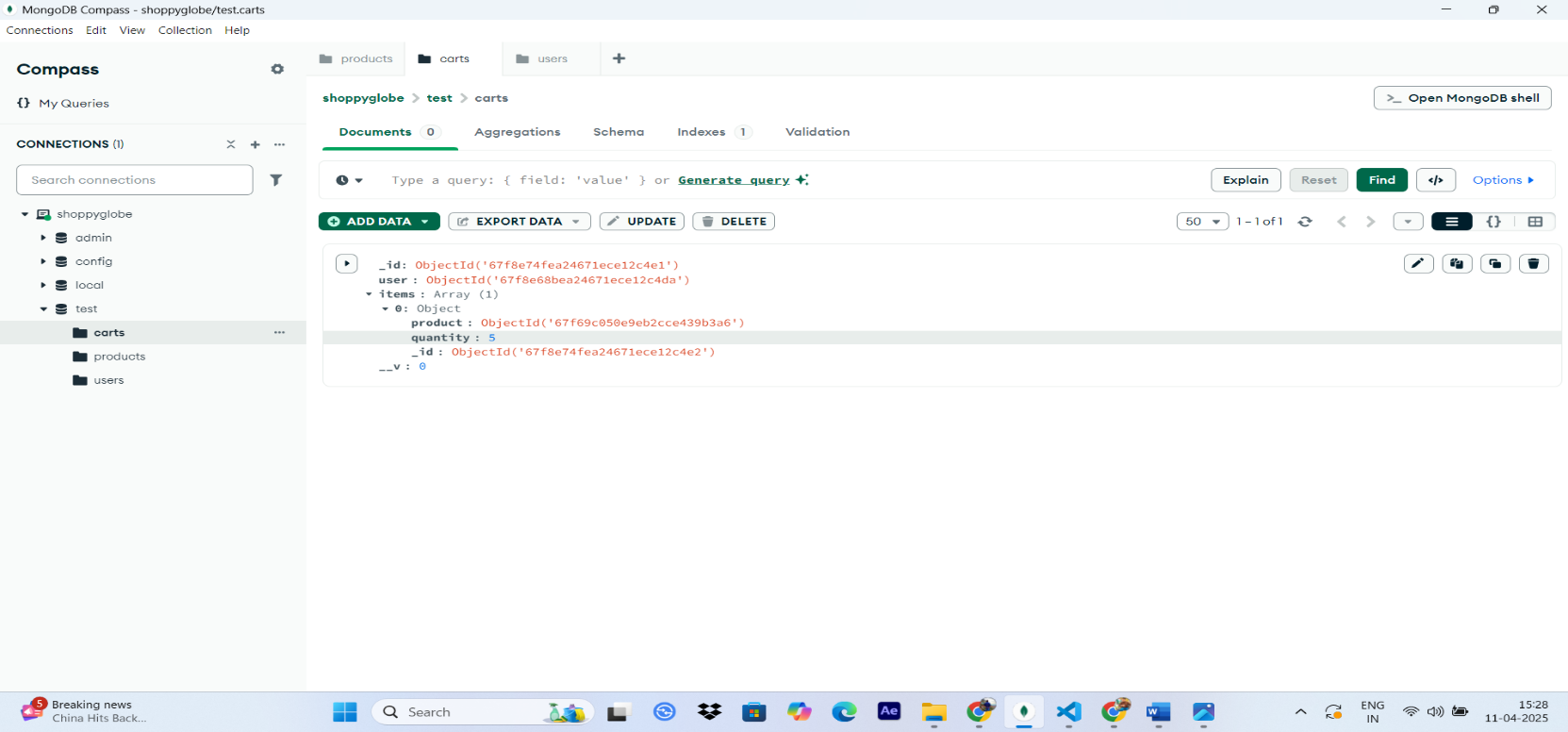
}

* AUTH TOKEN : IN Auth section of Thunder Client under Bearer add the token generated at the time of login.
  + **Expected Output**:Cart item quantity is updated in terms of quantity.

**SCREENSHOT :**

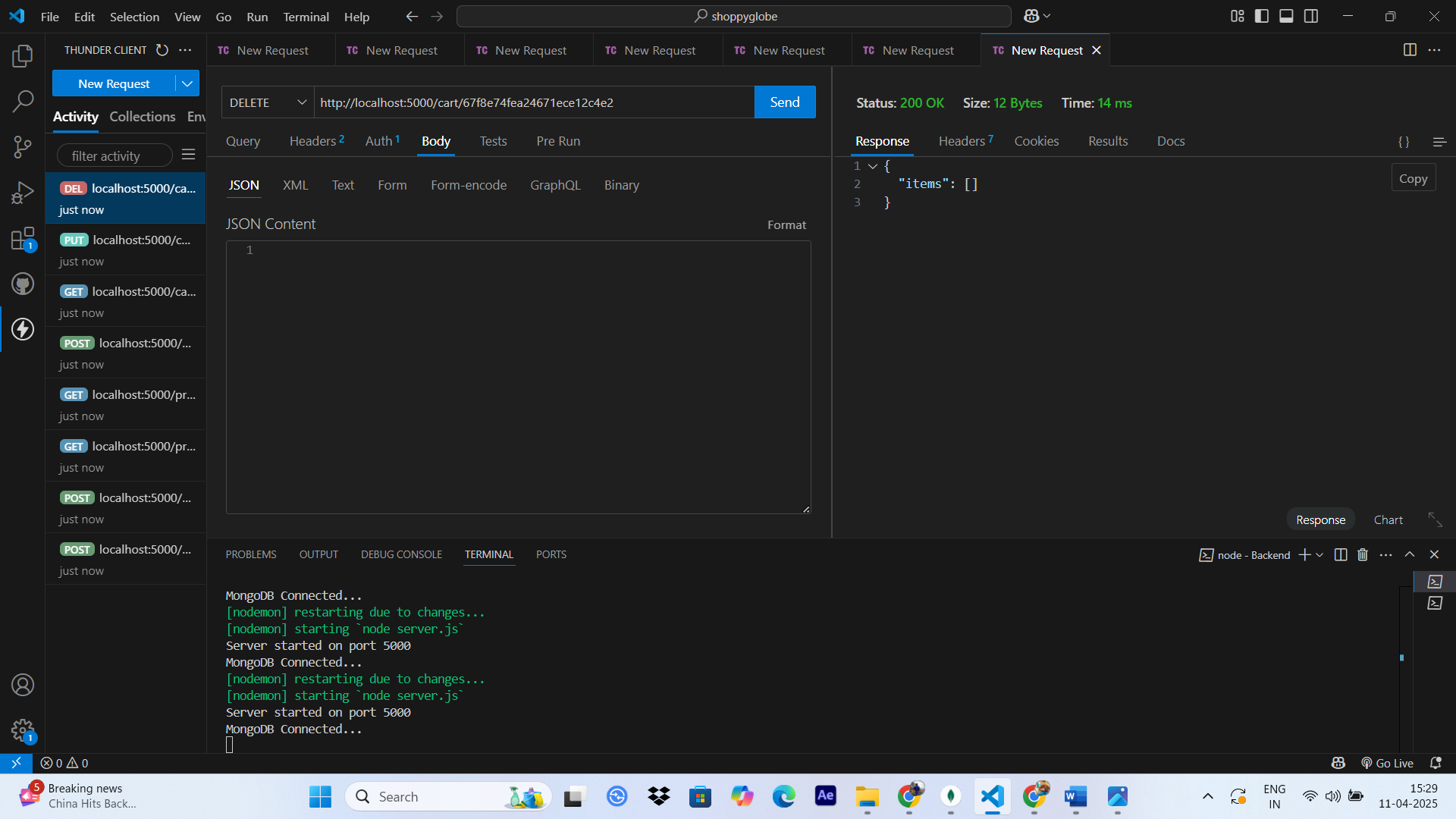
****

* **Quantity in Database is also updated**

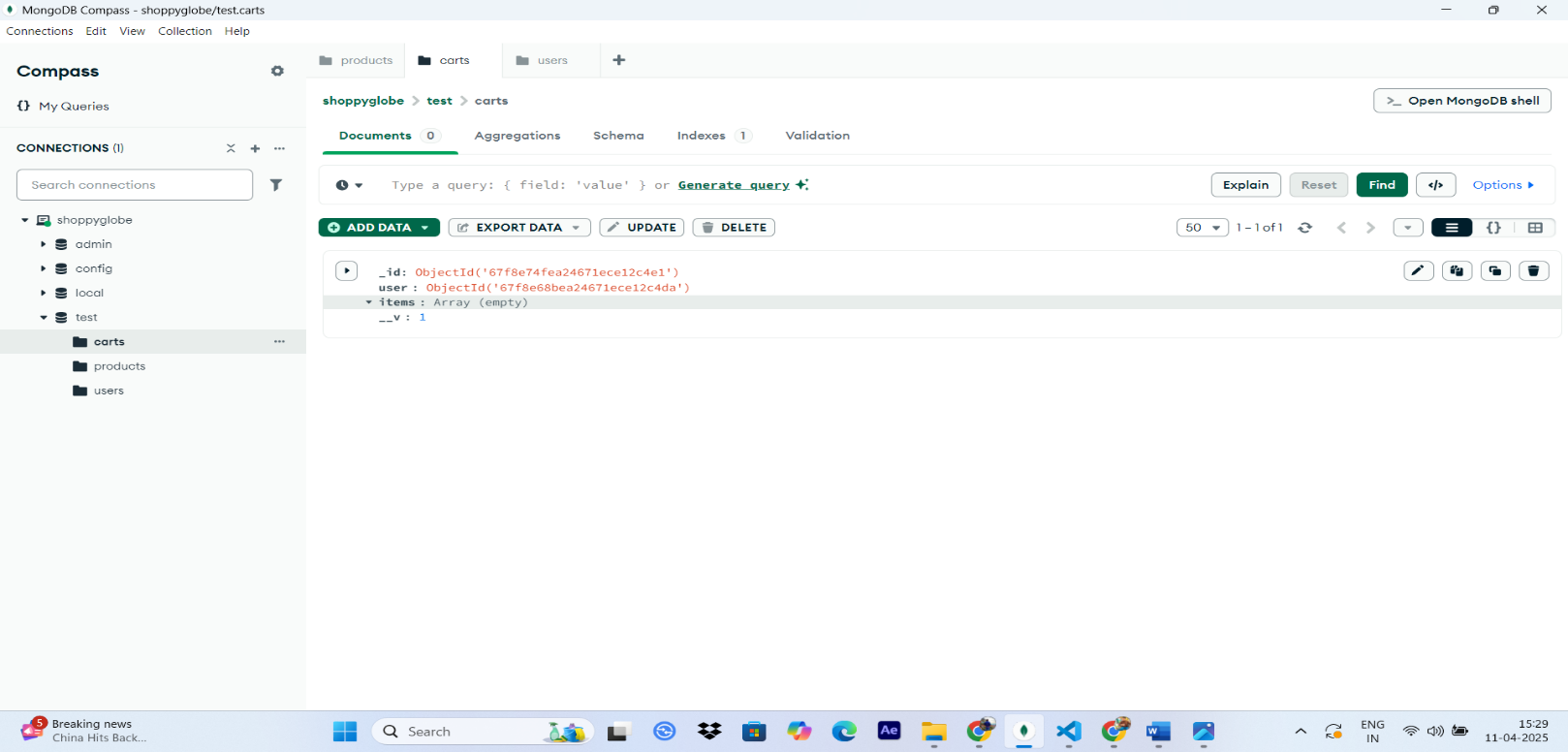
****

1. **DELETE Remove item from cart :**

* URL: http://localhost:5000/cart/<cart\_item\_id>
* Method: DELETE
* AUTH TOKEN : IN Auth section of Thunder Client under Bearer add the token generated at the time of login.
* **Expected Output:** The specified product will get deleted from the cart.

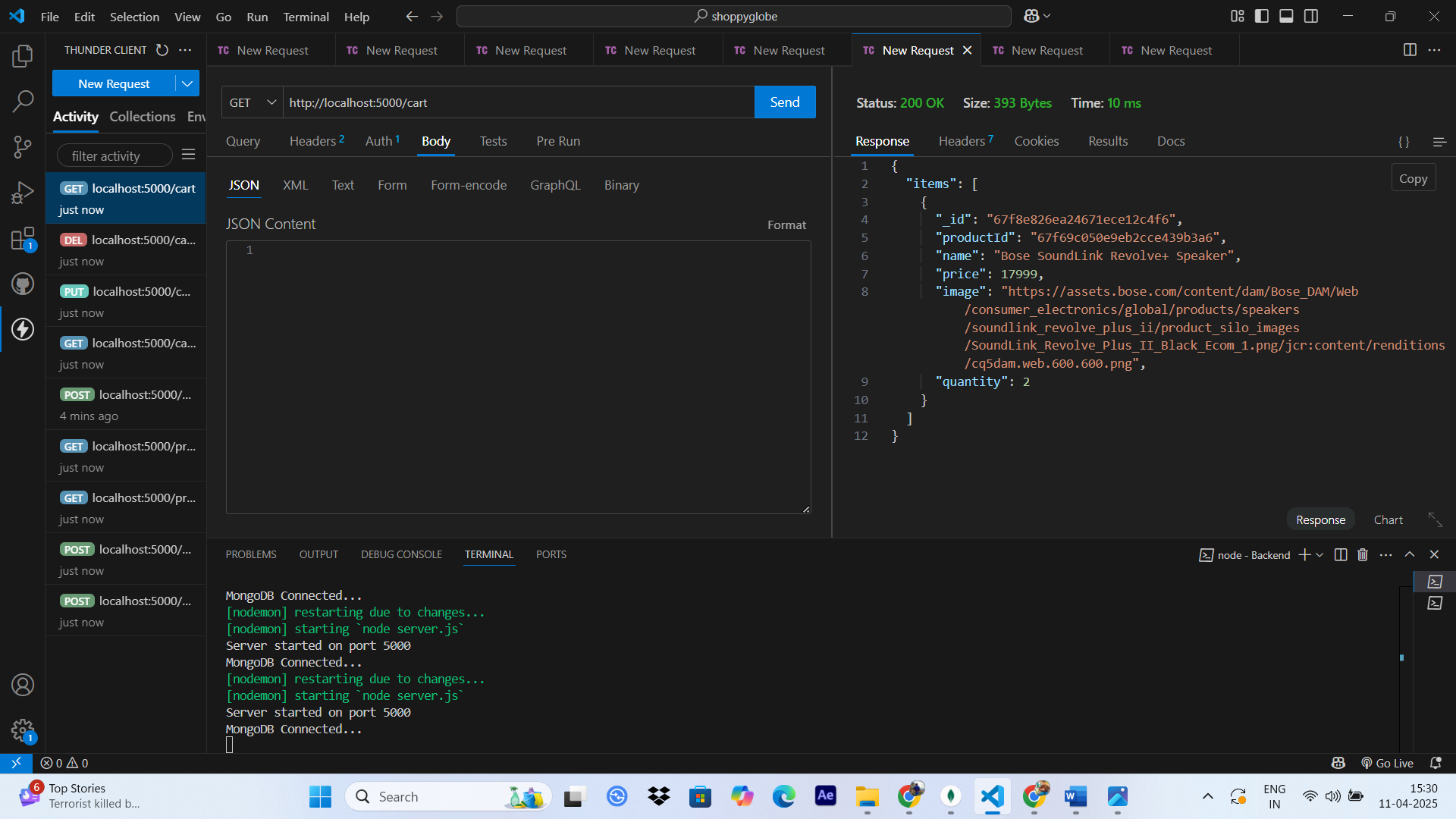
****

* **The Array of objects in Database gets Empty.**

****

1. **GET Cart Items:**

* URL: http://localhost:5000/cart
* Method: GET
* AUTH TOKEN : IN Auth section of Thunder Client under Bearer add the token generated at the time of login.
* **Expected Output:** The products in the cart will be available in the response.



1. **POST products in product collection:**

* URL: http://localhost:5000/products
* Method: POST
* BODY(JSON) : EXAMPLE : {

"name": "Apple iPhone 14 Pro Max",

"price": 139999,

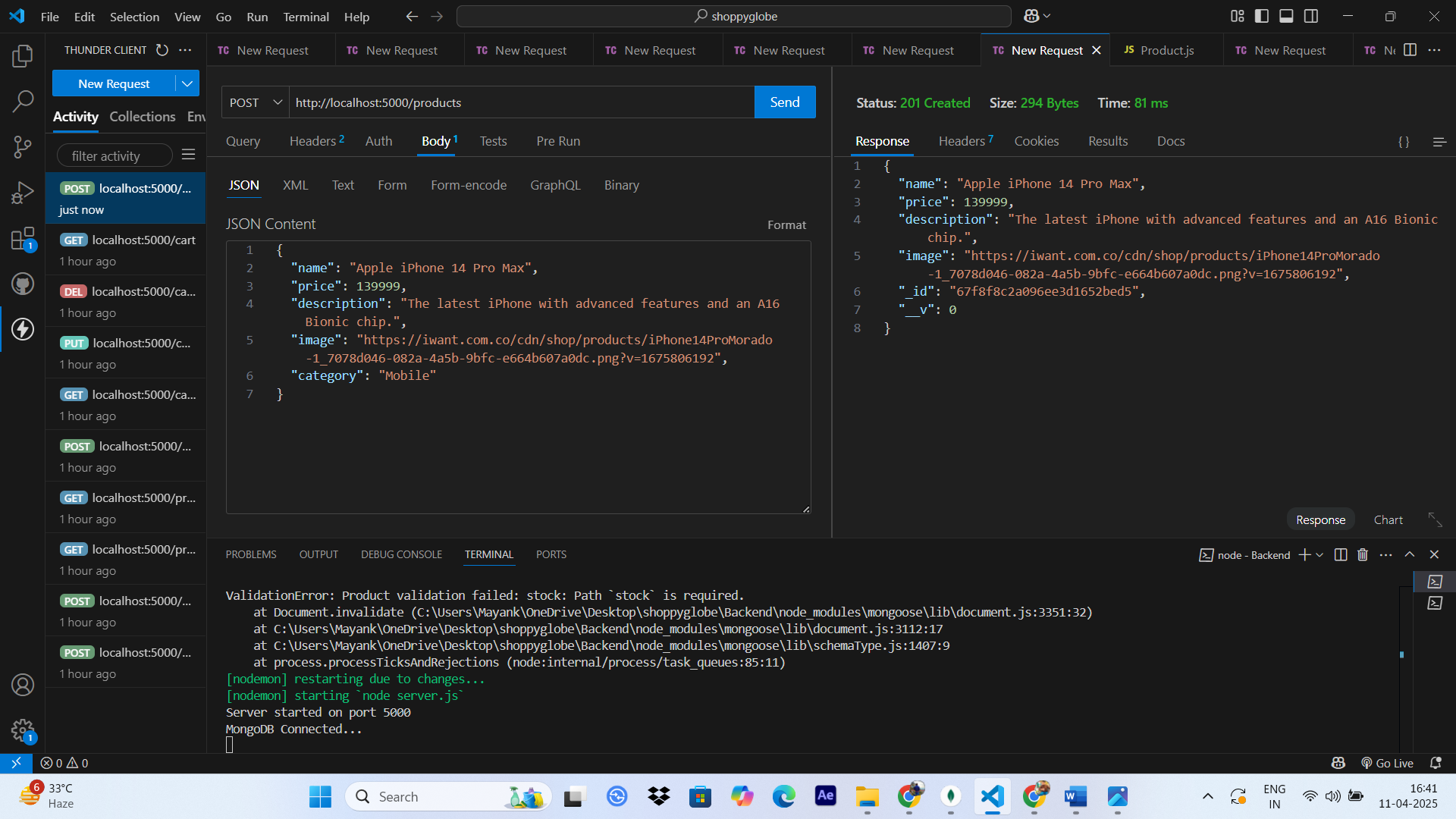
"description": "The latest iPhone with advanced features and an A16 Bionic chip.",

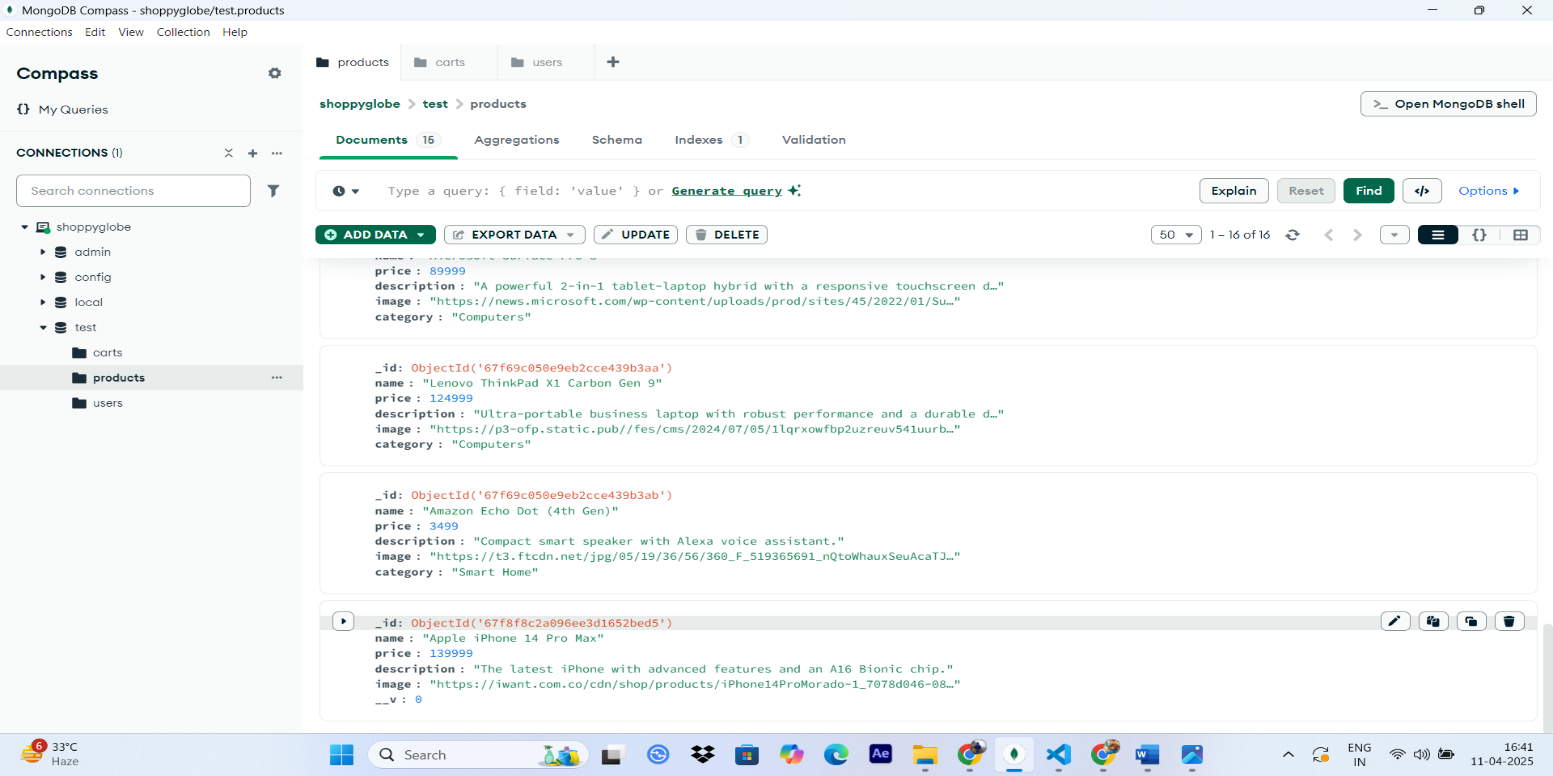
"image": "https://iwant.com.co/cdn/shop/products/iPhone14ProMorado-1\_7078d046-082a-4a5b-9bfc-e664b607a0dc.png?v=1675806192",

"category": "Mobile"

}

* **Expected Output:** The product will be added in the product collection.





* **Product is added in last.**

1. **PUT products in product collection:**

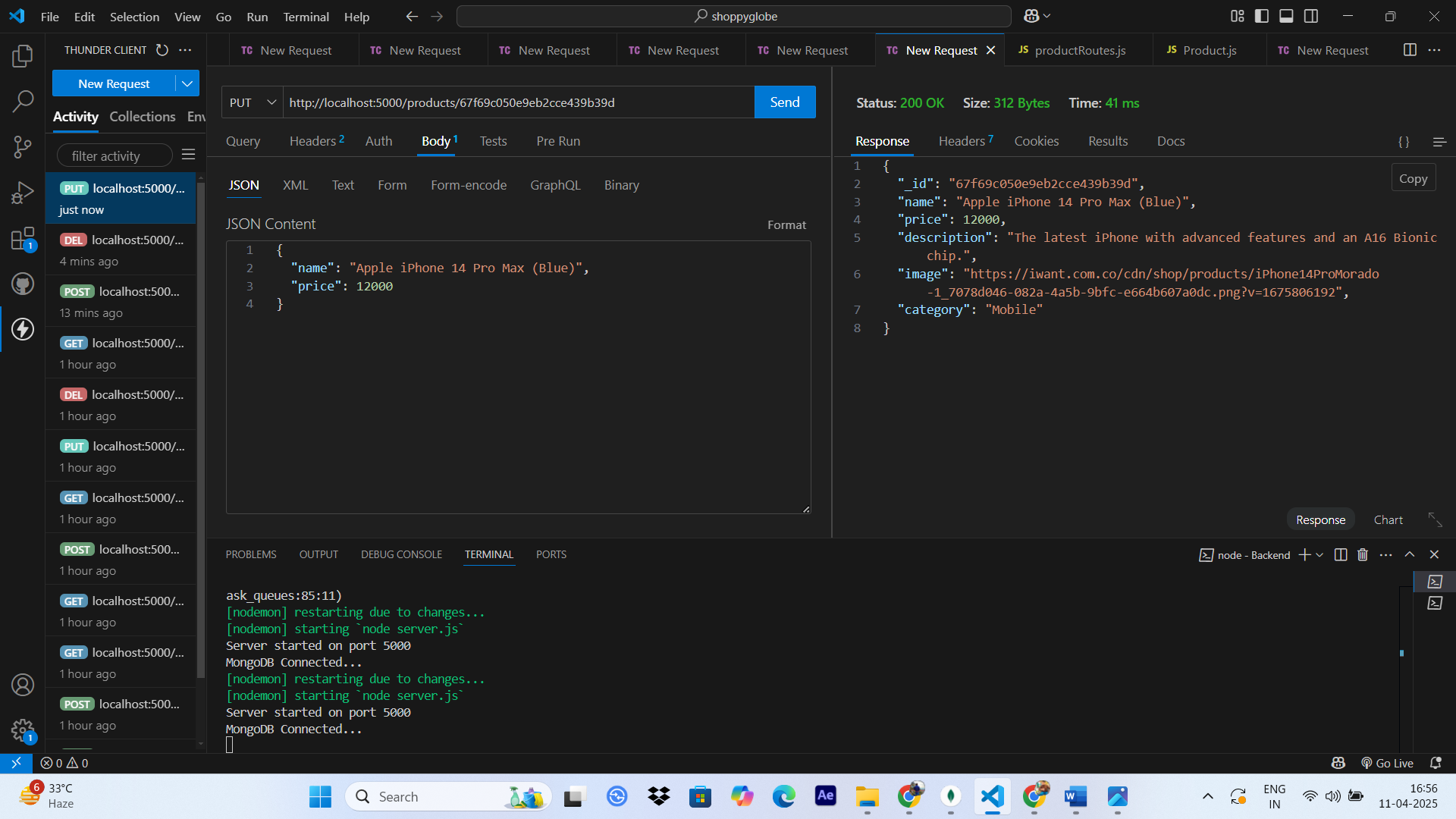
* URL: http://localhost:5000/products/<product\_id>
* Method: PUT
* BODY(JSON) : EXAMPLE : {

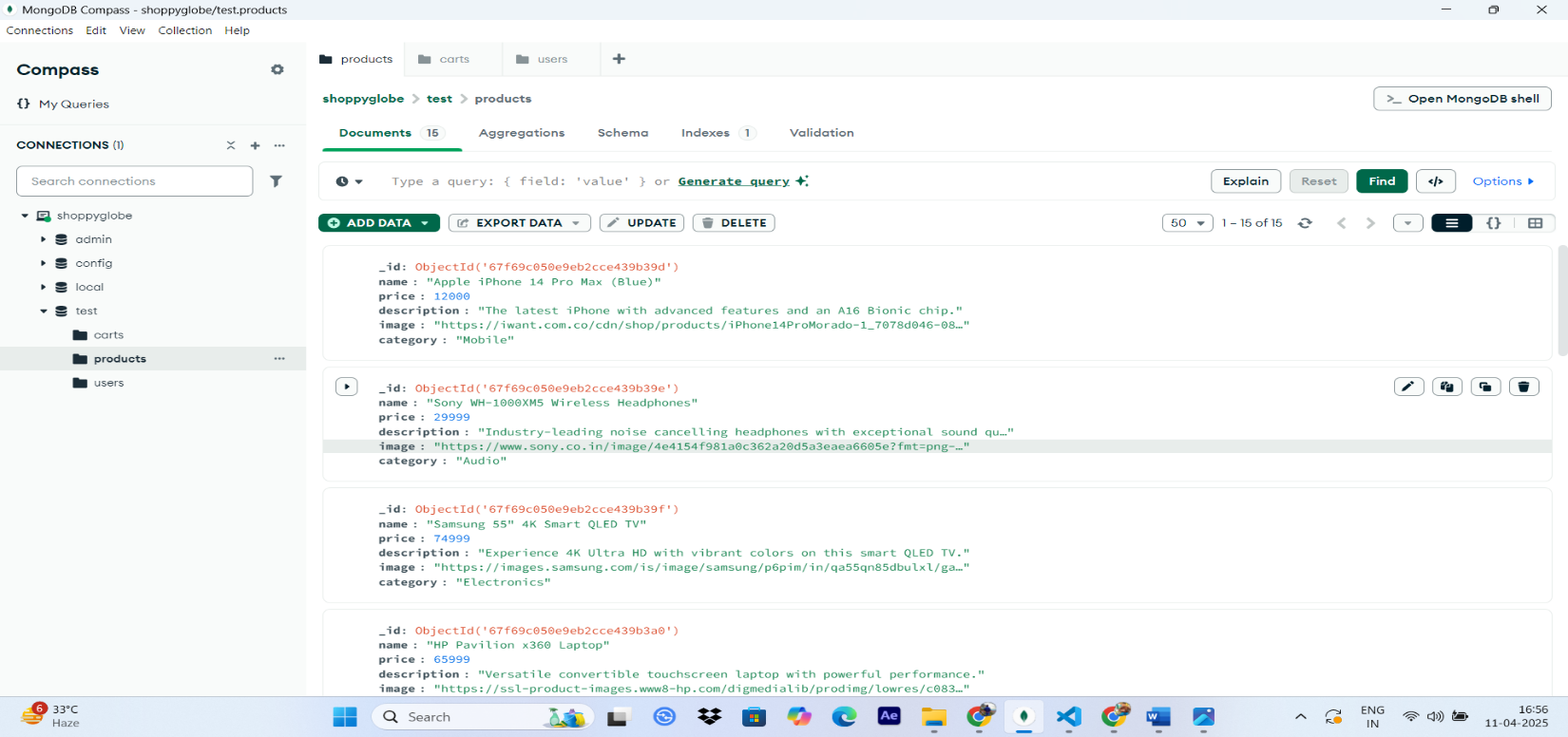
"name": "Apple iPhone 14 Pro Max(Blue)",

"price": 12000,

}

* **Expected Output:** The product will be edited in the product collection.

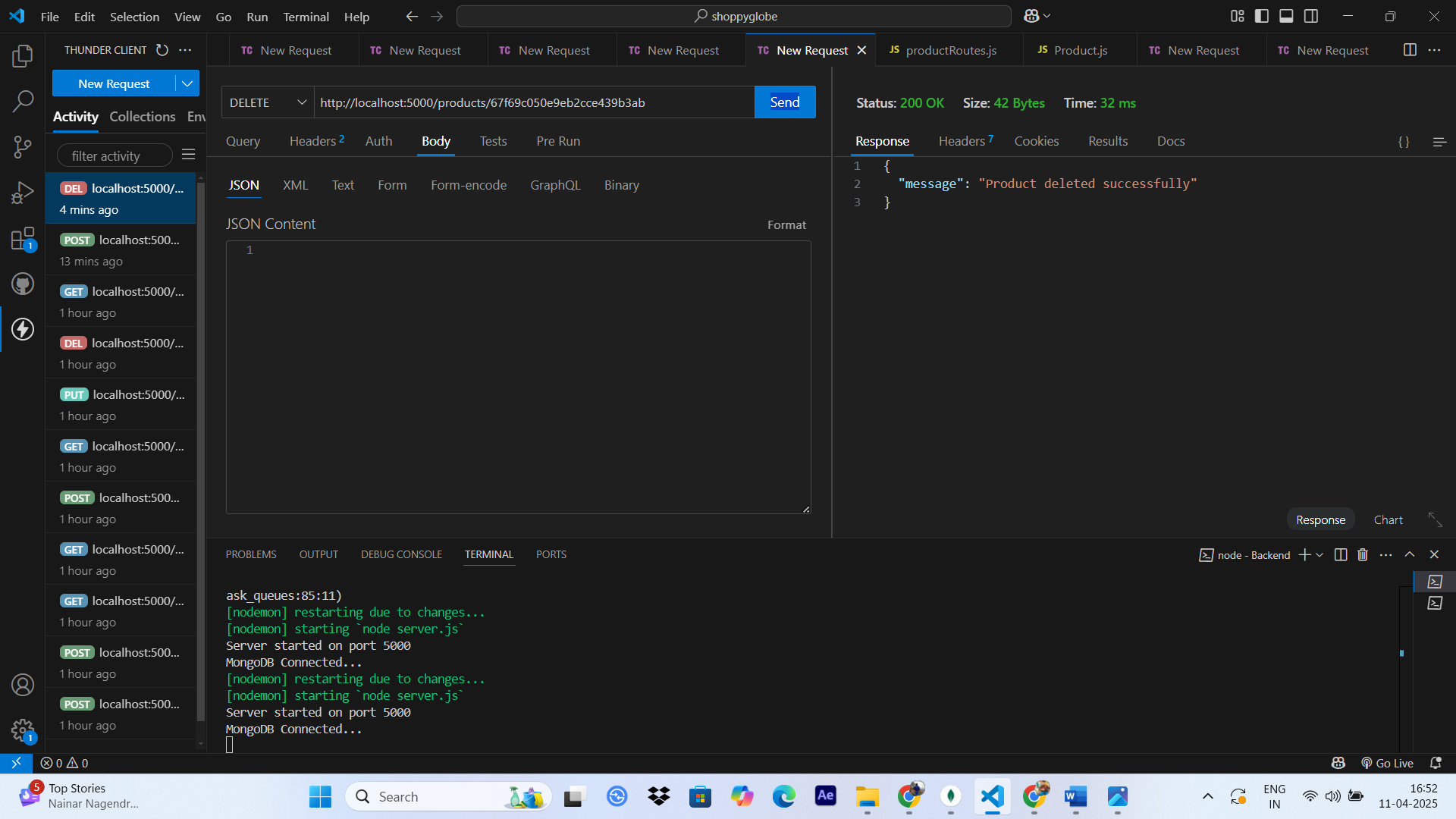


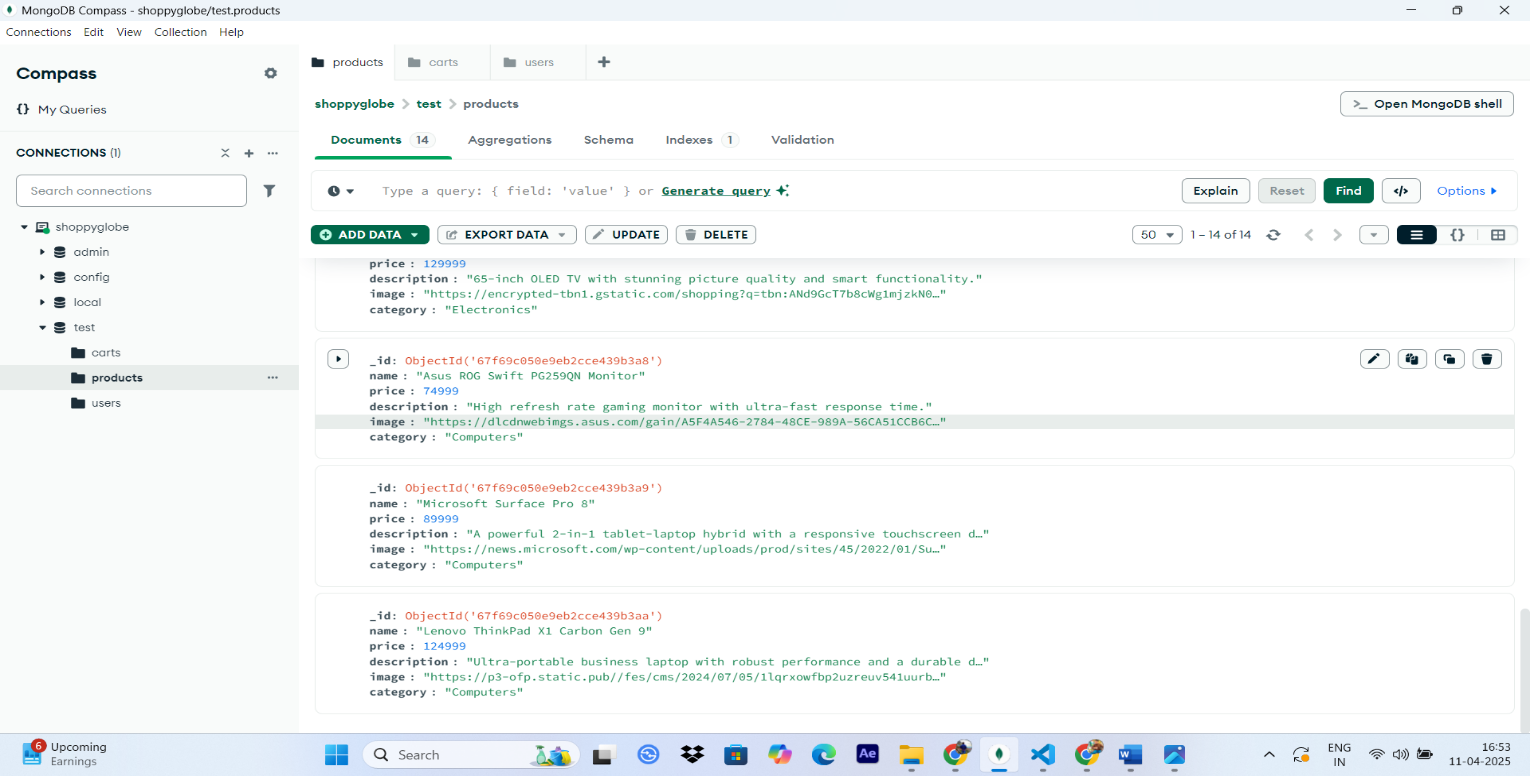


* Product details will get updated accordingly.

1. **DELETE Remove product from product collection :**

* URL: http://localhost:5000/products/<product\_id>
* Method: DELETE
* **Expected Output:** The specified product will get deleted from the collection.

****

****

* **The product successfully get deleted from the collection**

**(NOTE : HERE WE HAVE COVERED ALL CRUD OPERATIONS ON PRODCUTS , CART AND ALSO THE LOGIN AND REGISTER OPERATION)**

1. **Unauthorized Access (No Token):**

