

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

NAME : Lingananda.

REGISTER A NEW USER (POST /register)

Request:

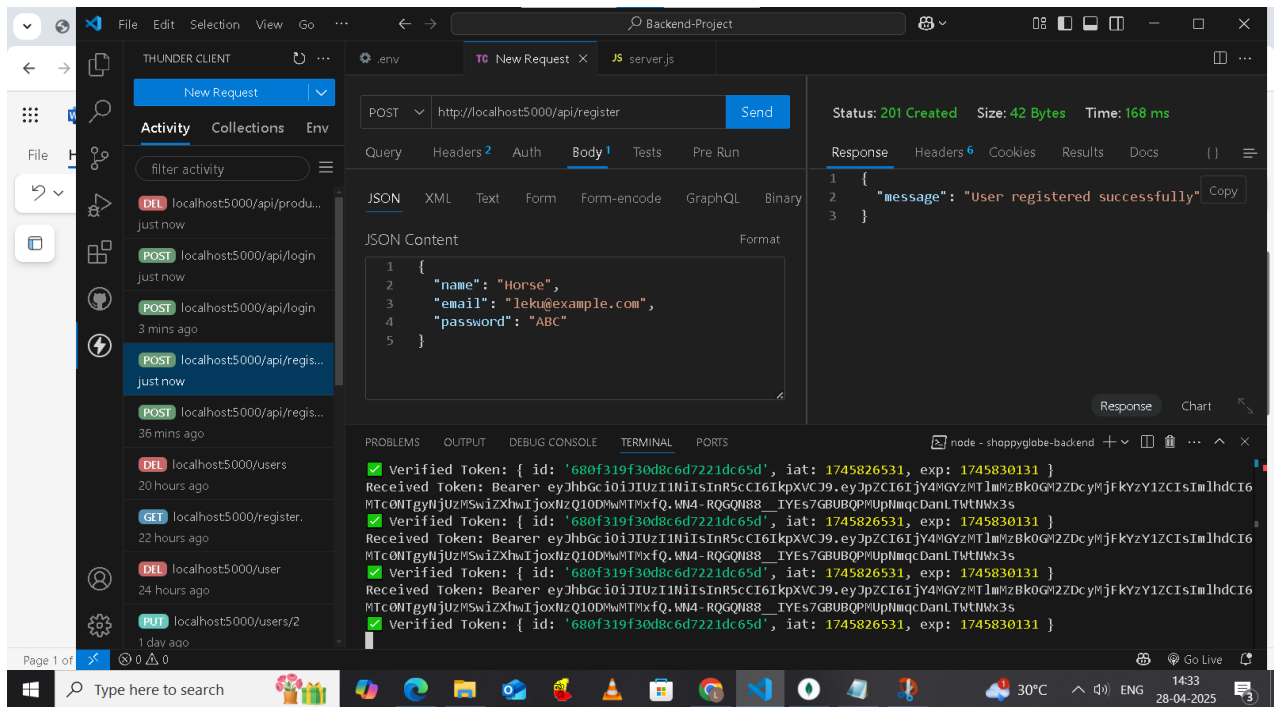
POST URL:

<http://localhost:5000/api/register>

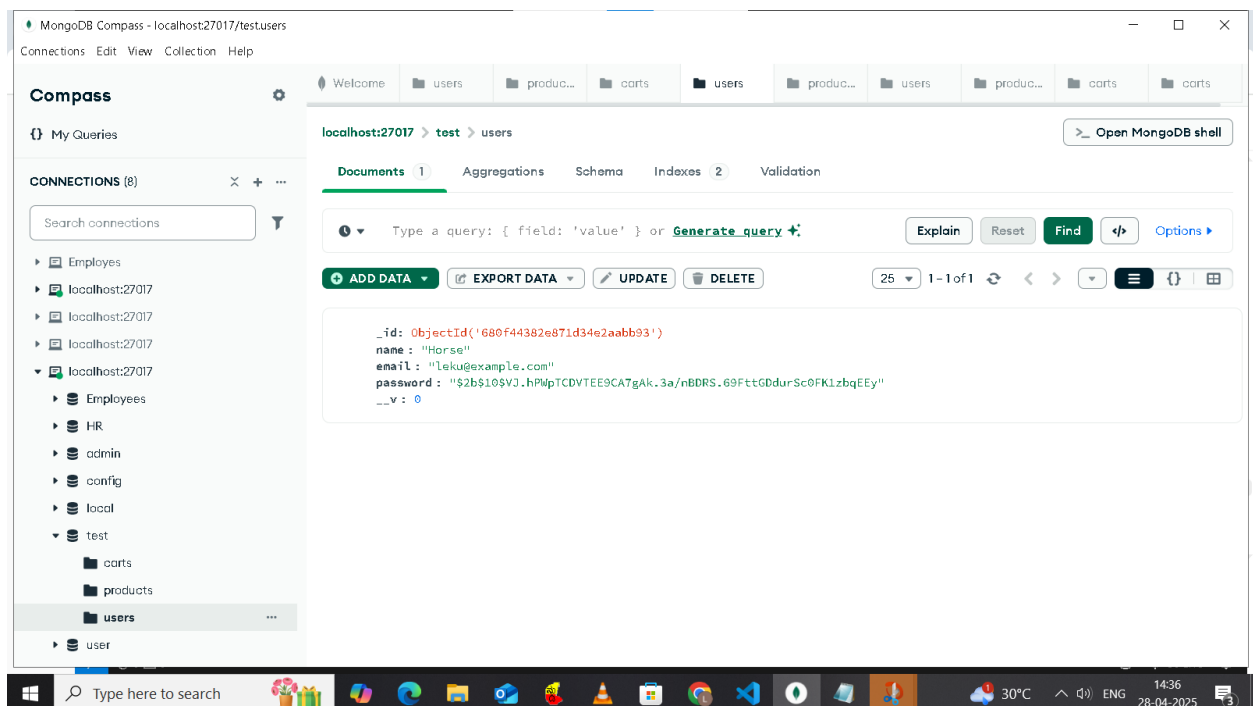
Go to the Body tab → Select raw → Choose JSON. Enter the JSON data:

```
{  
  "name": "Horse",  
  "email": "leku@gmail.com",  
  "password": "ABC"  
}
```

Screenshot



Screenshot Mongoddb



LOGIN THE USER (POST /login)

Request:

Make a POST request to:

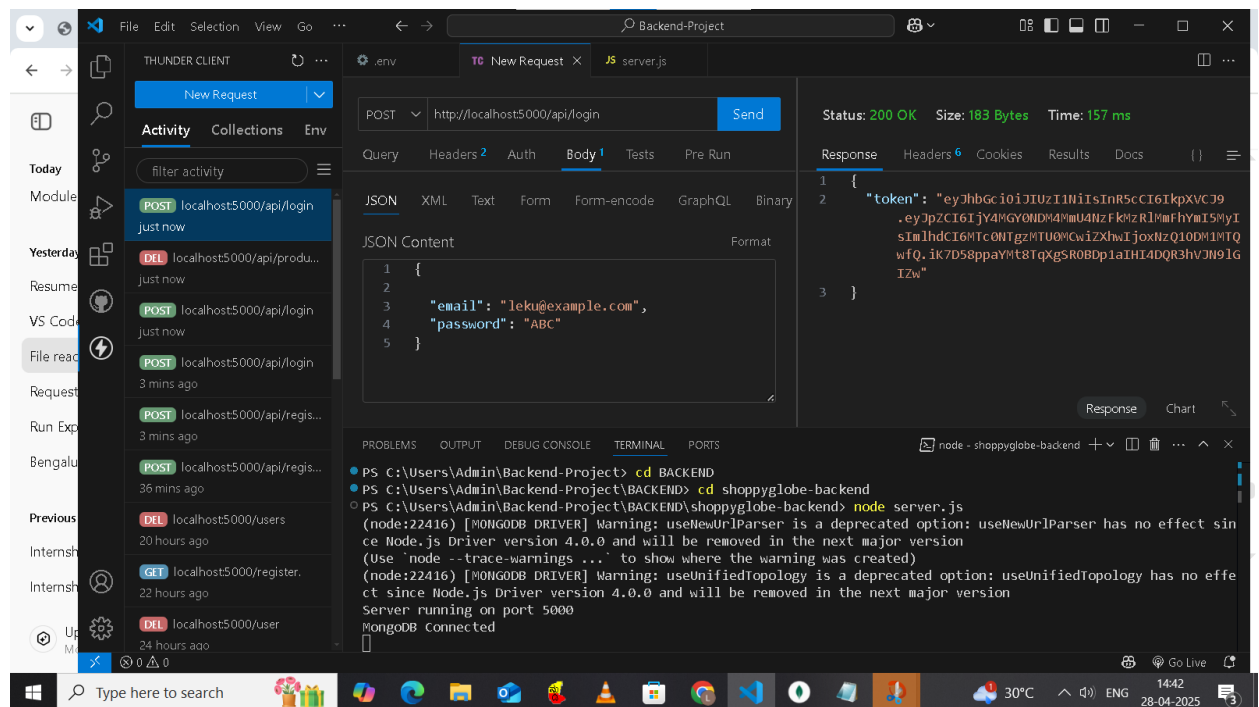
<http://localhost:5000/api/login>

Go to the Body tab → Select raw → Choose JSON.

Enter the JSON data:

```
{  
  "email": "leku@example.com",  
  "password": "ABC"  
}
```

Screenshot



The server verifies email and password. Generates a JWT token valid for 1 hour.

ADD A PRODUCT (POST /products)

Go to the Body tab → Select raw → Choose JSON. Enter the JSON data: <http://localhost:5000/api/products>

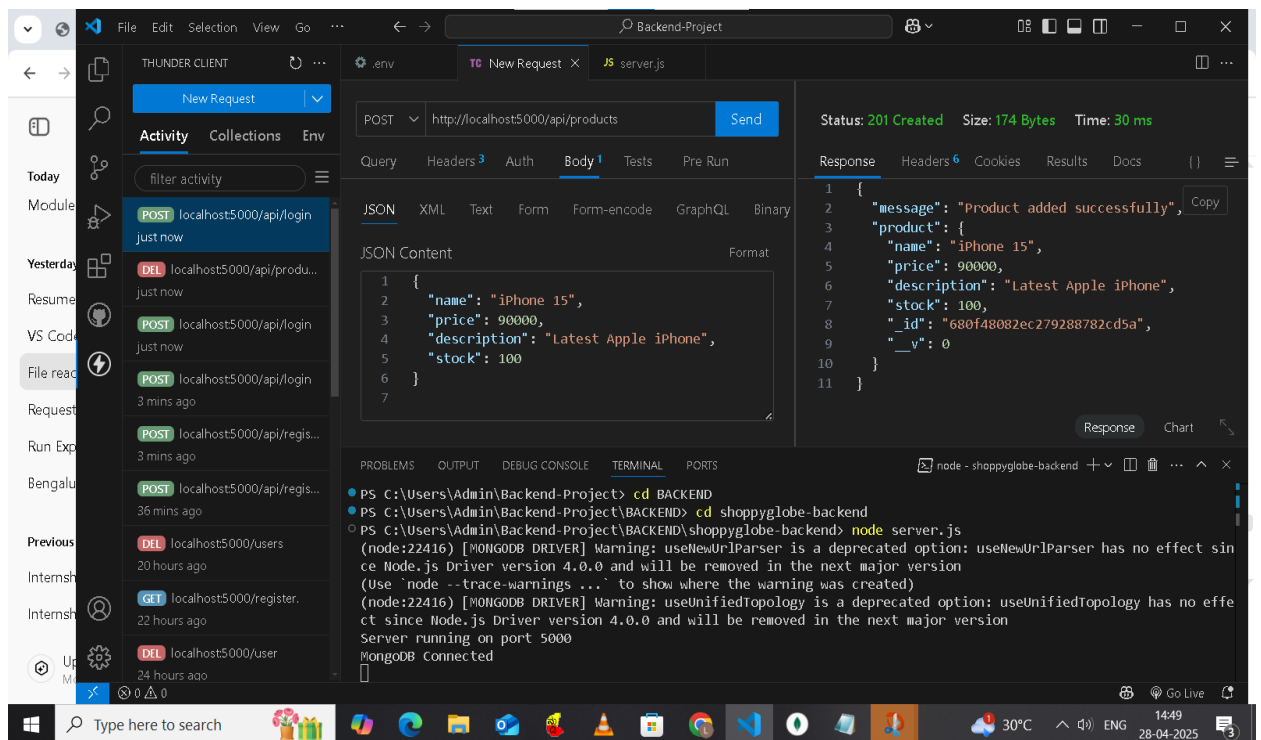
Request Body:

json

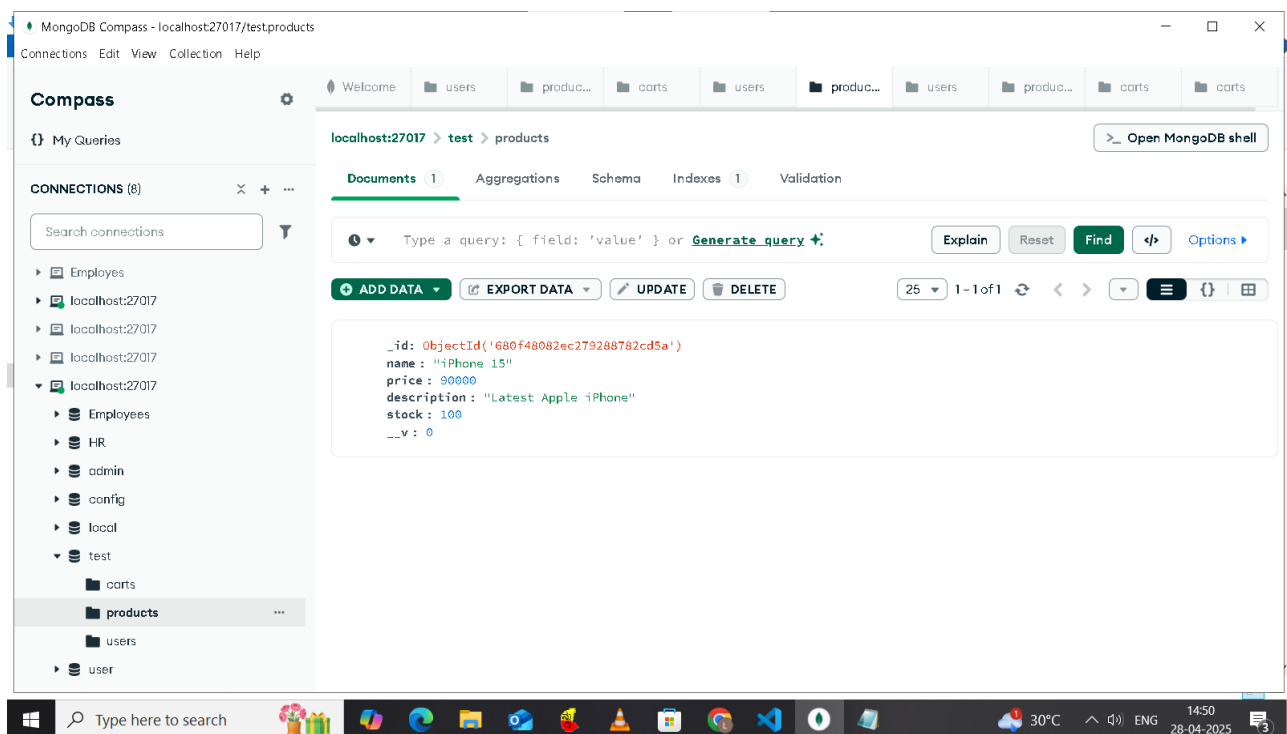
CopyEdit

```
{
  "name": "iPhone 15",
  "price": 90000,
  "description": "Latest Apple iPhone",
  "stock": 100
}
```

Screenshot



ScreenshotMongodb



GET ALL PRODUCTS (GET /products)

Request:

Make a GET request to:

<http://localhost:5000/api/products>

Processing:

- Fetches all products stored in MongoDB.

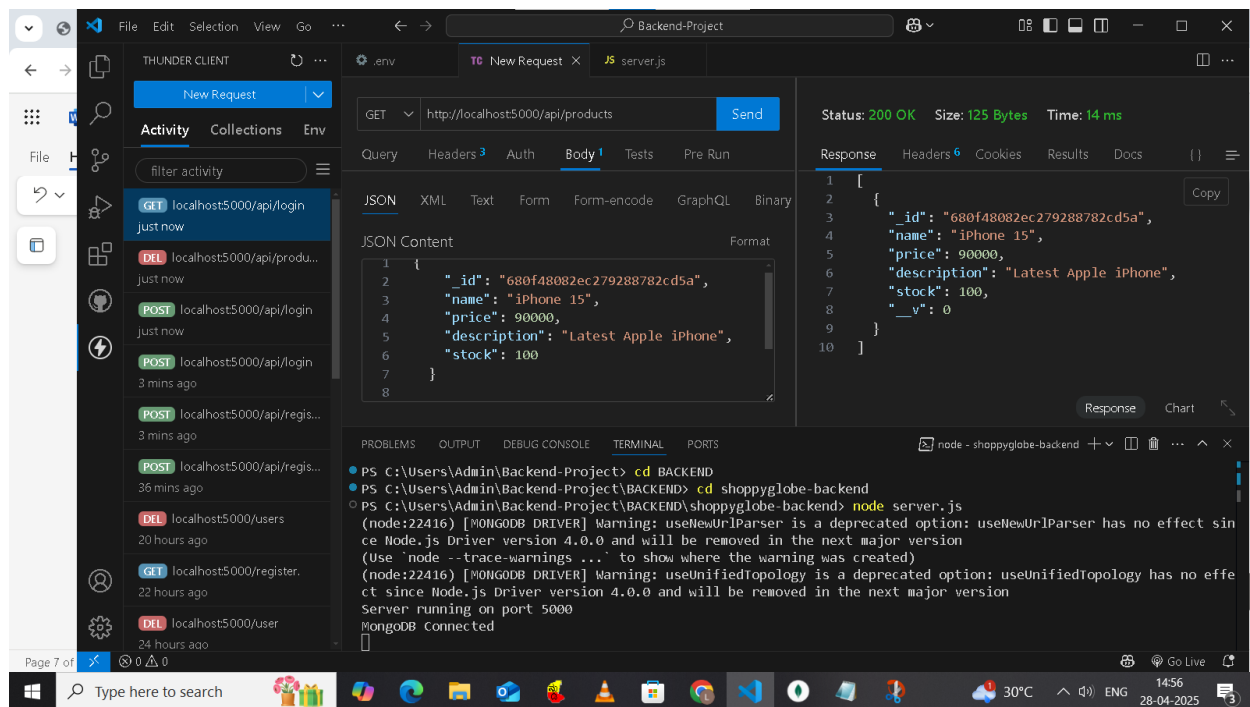
Expected Response:

BODY Json

```
[
  {
    "_id": "680f48082ec279288782cd5a ",
    "name": "iPhone 15",
    "price": 90000,
    "description": "Latest Apple iPhone",
    "stock": 100
  }
]
```

Confirms that the Product database and API are working.

Screenshot



GET A SINGLE PRODUCT (GET /products/:id)

Request:

Make a GET request to:

<http://localhost:5000/api/products/680f48082ec279288782cd5a>

Replace **:id** with actual Product ID.

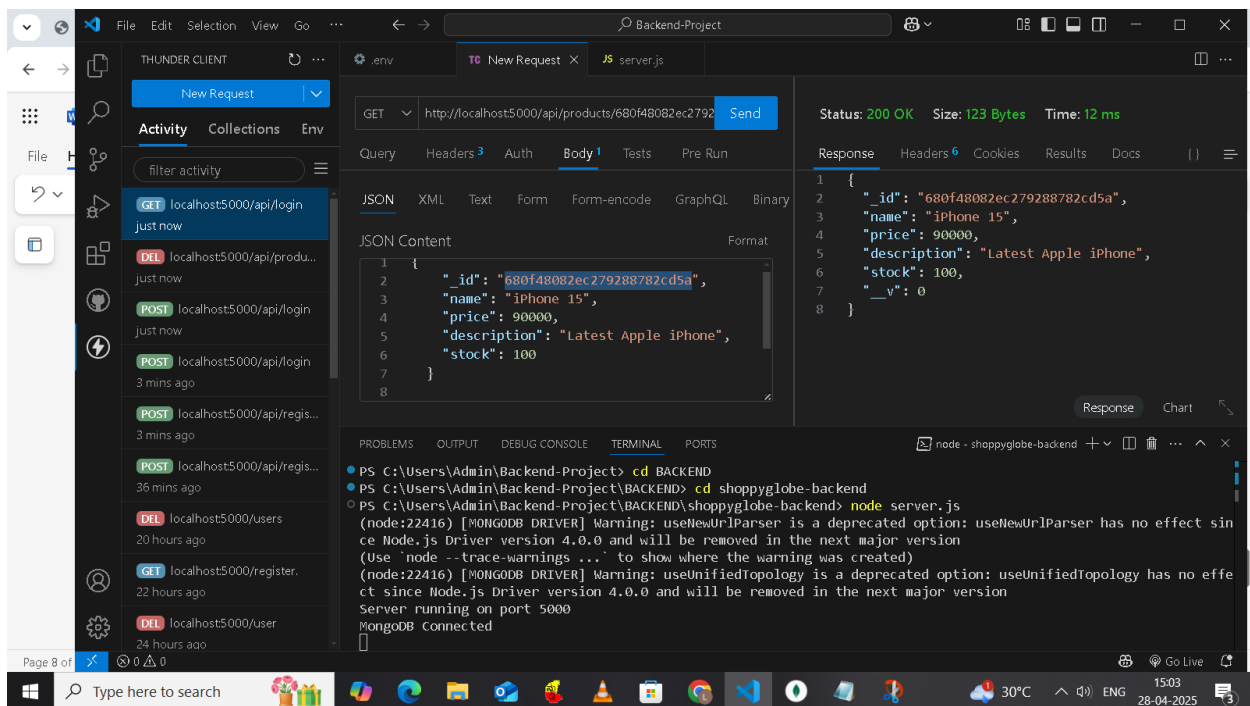
Processing:

- Server fetches product details by ObjectId.

Response:

```
{  
  
  "_id": "680f48082ec279288782cd5a",  
  
  "name": "iPhone 15",  
  
  "price": 90000,  
  
  "description": "Latest Apple iPhone",  
  
  "stock": 100}
```

Screenshot



ADD PRODUCT TO CART (POST /cart)

Request:

Make a POST request to:

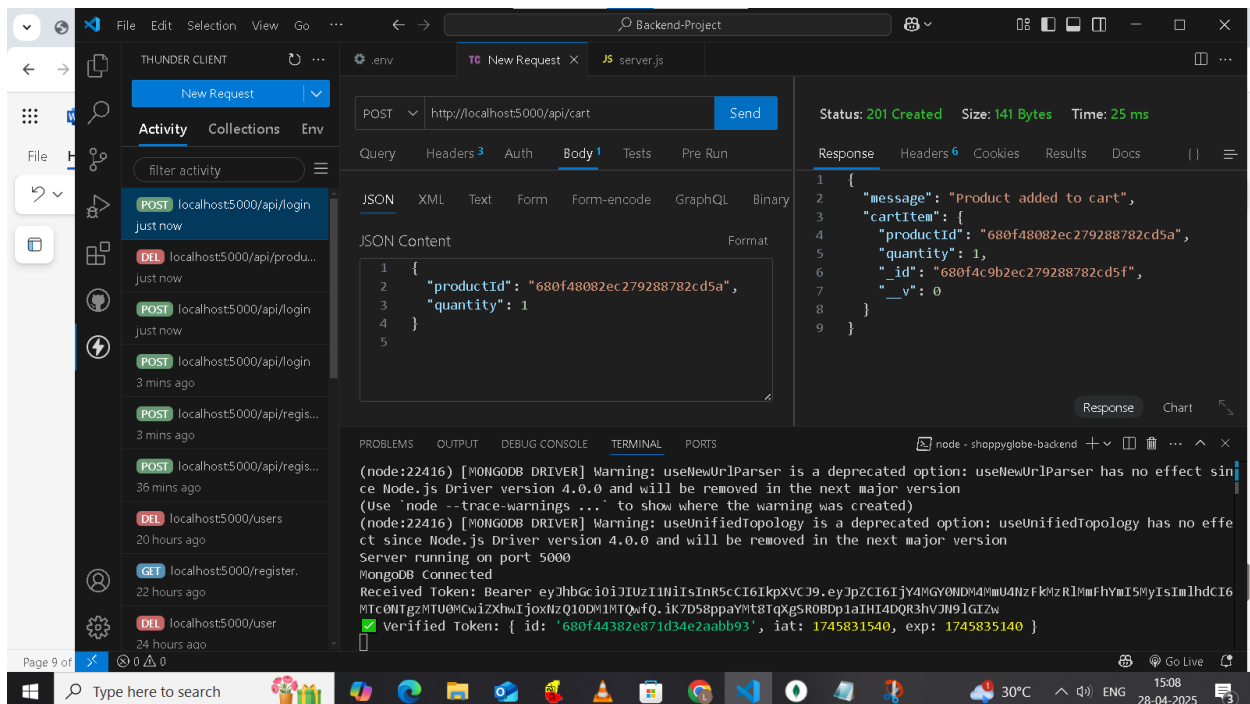
<http://localhost:5000/api/cart>

Authorization: Bearer Token Required

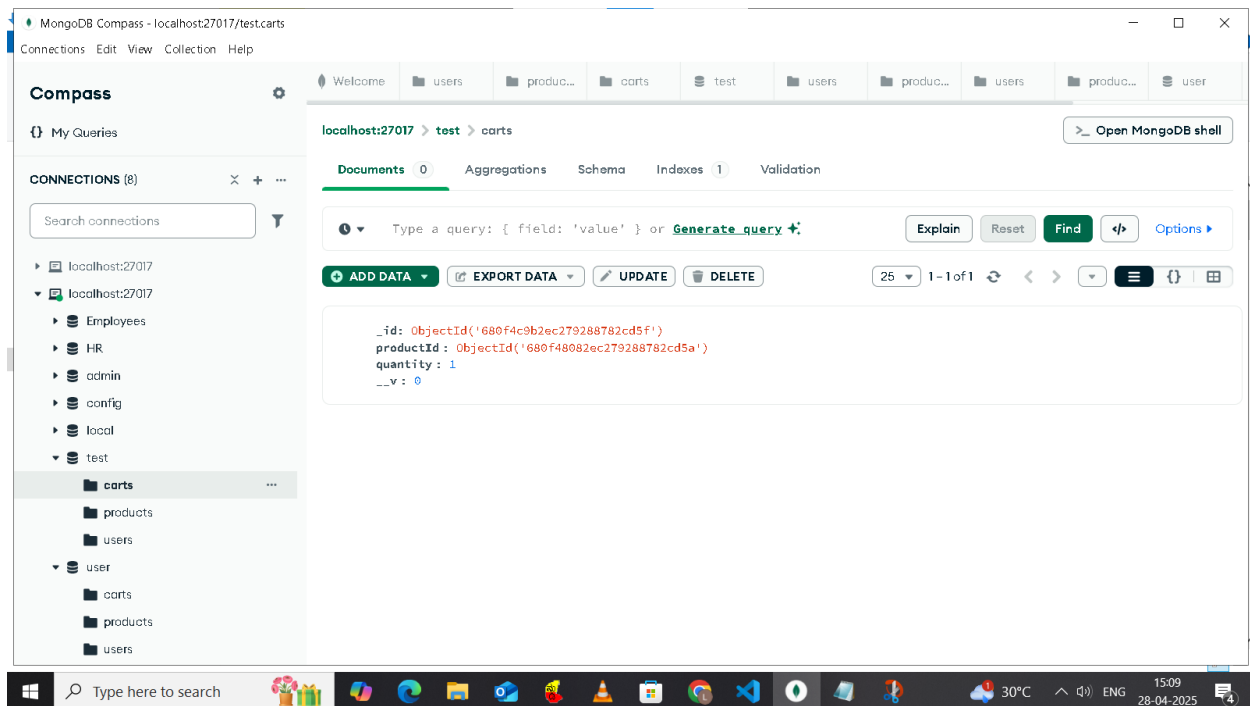
Request Body:

```
json
CopyEdit
{
  "productId": "680f48082ec279288782cd5a",
  "quantity": 1
}
```

Screenshot



Screenshot Mongoddb



Processing:

- Checks product availability.
- Adds the selected product to the Cart collection.

UPDATE CART ITEM QUANTITY (PUT /cart/:id)

Request:

Make a PUT request to:

<http://localhost:5000/api/cart/680f4c9b2ec279288782cd5f>

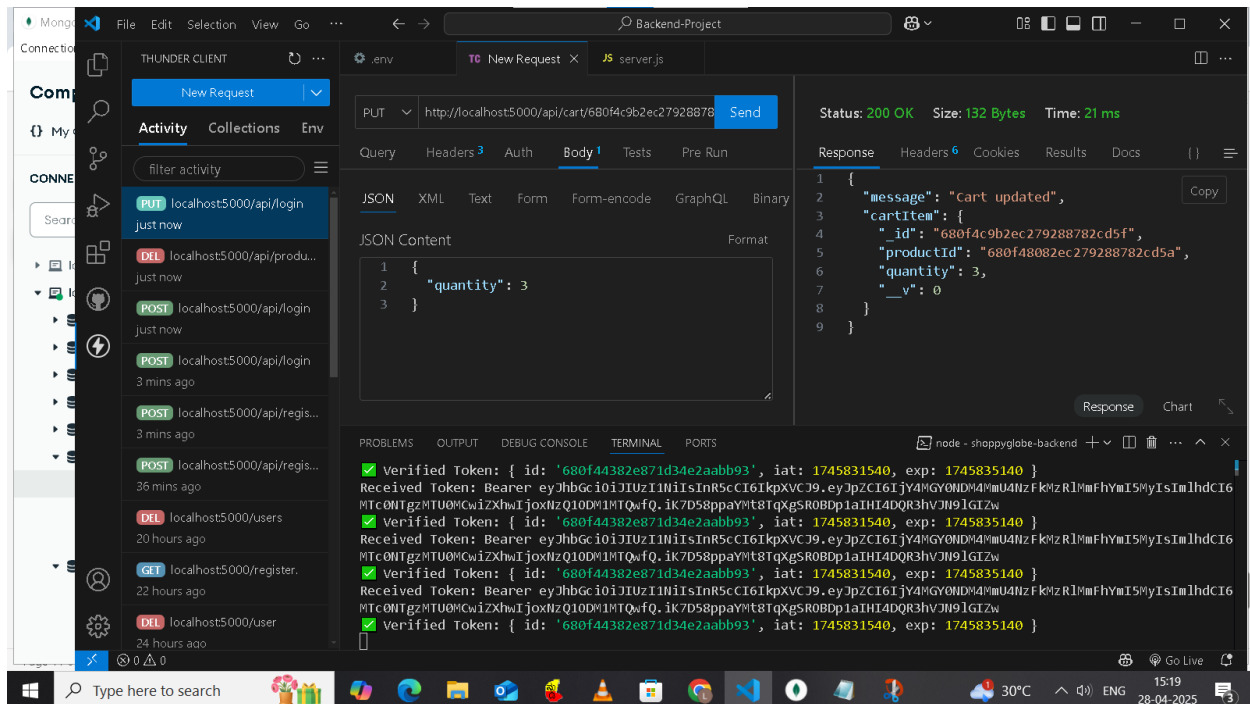
Authorization: Bearer Token Required

Request Body:

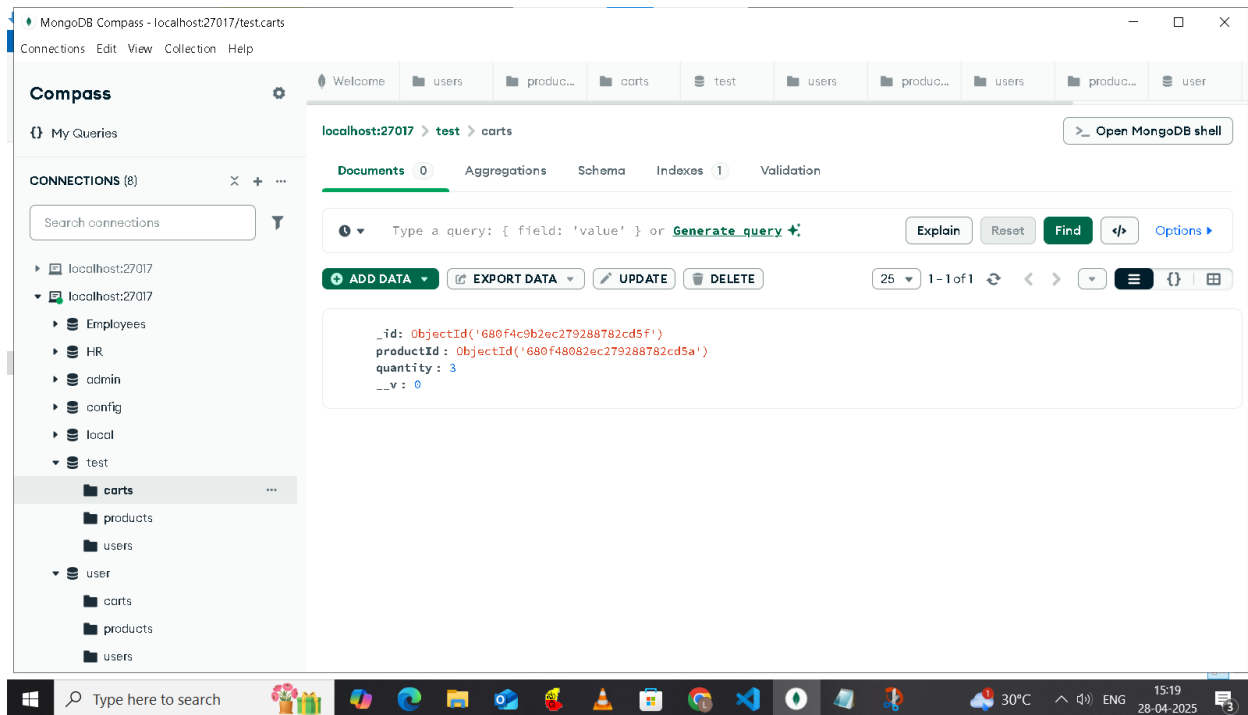
json
CopyEdit

```
{  
  "quantity": 3  
}
```

Screenshot



ScreenshotMongodb



Processing:

- Updates the quantity of the specific cart item.

DELETE A CART ITEM (DELETE /cart/:id)

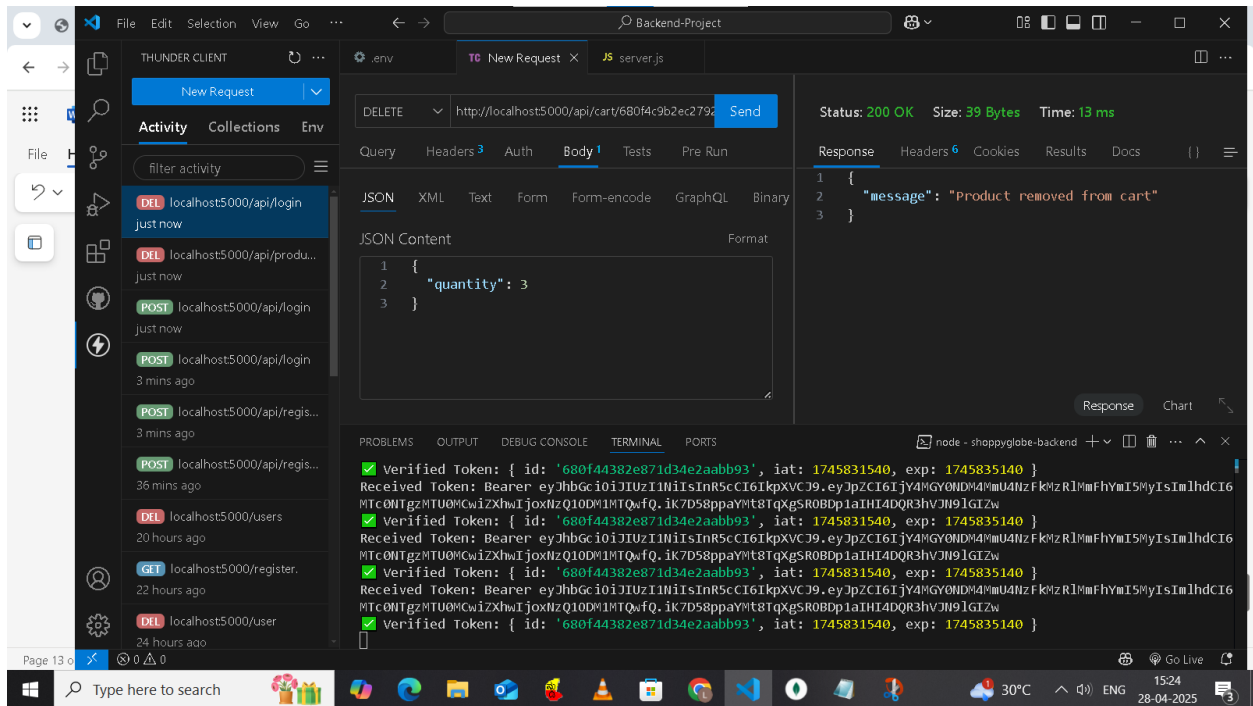
Request:

Make a DELETE request to:

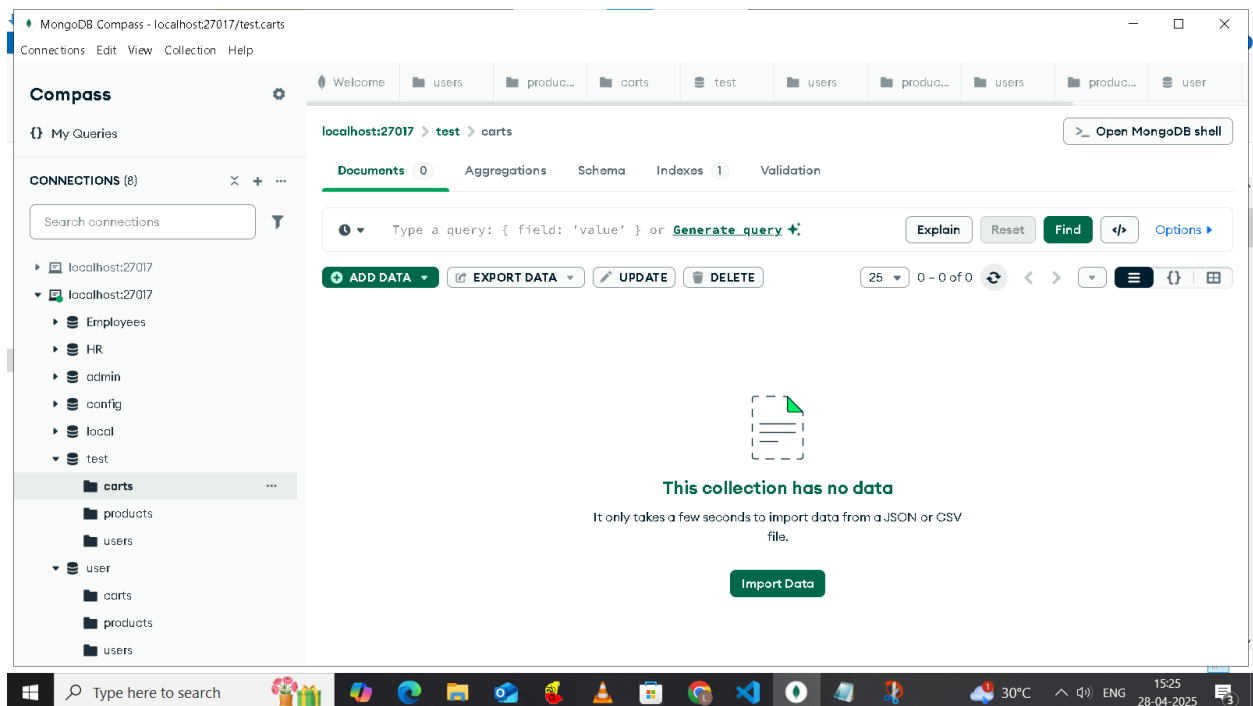
<http://localhost:5000/api/cart/680f4c9b2ec279288782cd5f>

Authorization: Bearer Token Required

Screenshot



Screenshotmongodb



Processing:

- Deletes the specific item from the cart.

SUMMARY

Register a New User:

- User details saved securely with hashed password. **Login and Authentication:**
- Secure JWT-based authentication for protected routes.

Product Management:

- Add new products.
- View all products or single product by ID.

Cart Management:

- Add products to cart.
- Update quantity.
- Delete products from the cart.

MongoDB Storage:

- All users, products, and cart items are stored and managed inside MongoDB database.

Thunder Client:

- All operations tested through Thunder Client in VS Code.
- API working correctly with appropriate status codes.

