SHOPPYGLOBE BACKEND - API TESTING DOCUMENTATION

1. Introduction

This document presents the complete Thunder Client testing of the backend API built for ShoppyGlobe, an e-commerce platform.
All major REST API endpoints - Auth, Products, Cart - were tested with correct data, incorrect data, validation failures, and error-handling scenarios.

2. Authentication API Testing

2.1 POST /register - User Registration

Description:
Creates a new user by storing name, email, and hashed password in MongoDB.

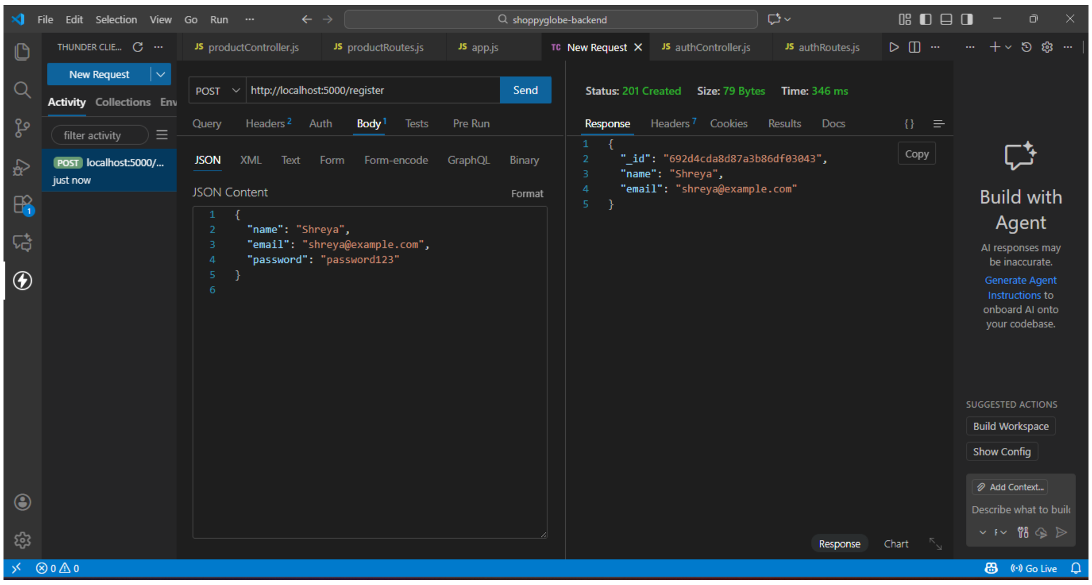
Valid Input:

- Unique email
- Password ≥ 6 characters

Expected Behavior:

- Status 201
- Returns user ID, name, email

Screenshot:



POST http://localhost:5000/register

```
{
  "name": "Shreya",
  "email": "shreya@example.com",
  "password": "password123"
}
```

Status: 201 Created   Size: 79 Bytes   Time: 346 ms

```
{
  "_id": "692d4cda8d87a3b86df03043",
  "name": "Shreya",
  "email": "shreya@example.com"
}
```

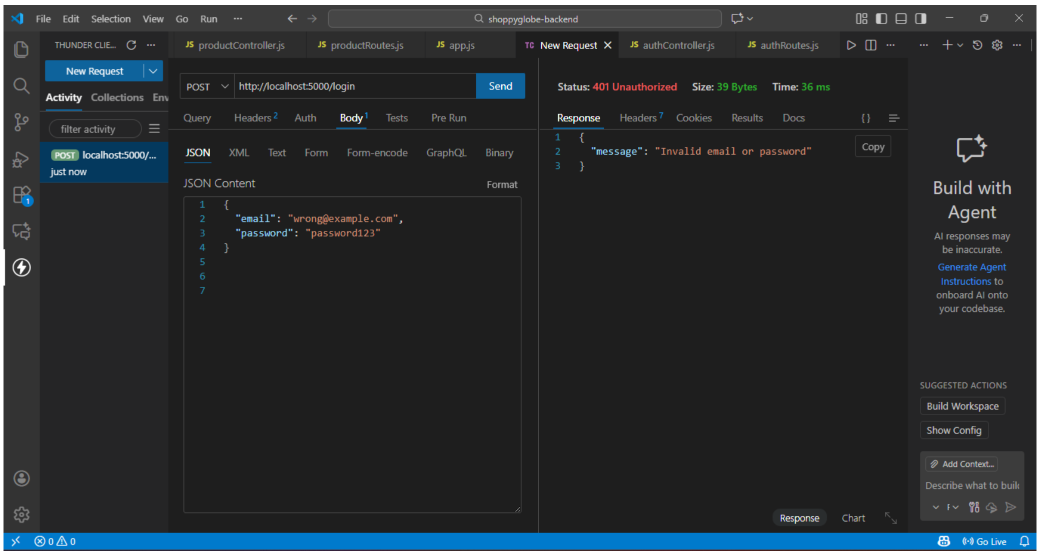
2.2 POST /login - Login With Invalid Email

Description:
Verifies authentication error when user enters an email that does not exist.

Expected Behavior:

- Status 401
- Message: *"Invalid email or password"*

Screenshot:



POST http://localhost:5000/login

JSON Content
```
{
  "email": "wrong@example.com",
  "password": "password123"
}
```

Status: 401 Unauthorized Size: 39 Bytes Time: 36 ms

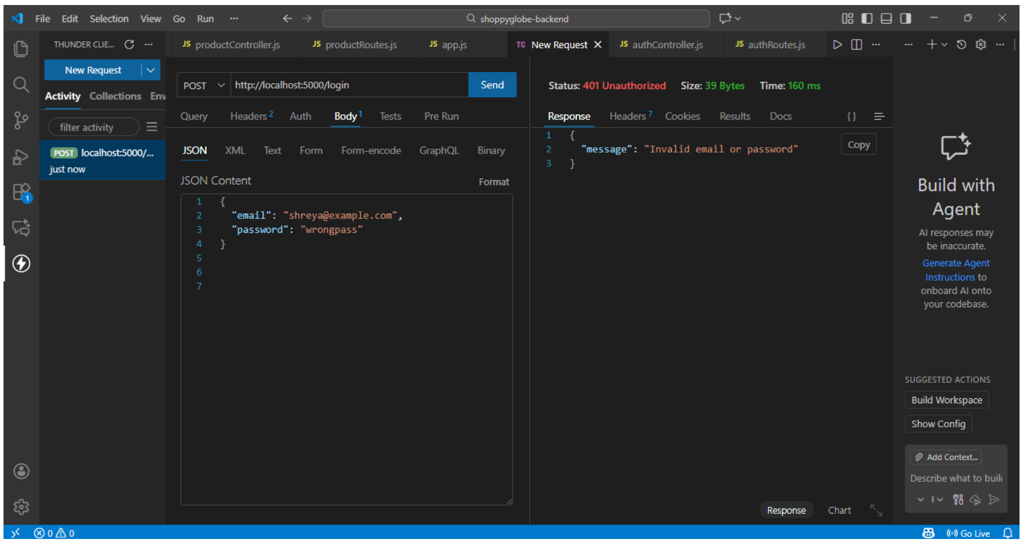
Response
```
{
  "message": "Invalid email or password"
}
```

2.3 POST /login - Login With Wrong Password

Description:
Checks behavior when email exists but password does not match.

Expected Behavior:

- Status 401
- Message: *"Invalid email or password"*



POST http://localhost:5000/login

JSON Content
```
{
  "email": "shreya@example.com",
  "password": "wrongpass"
}
```

Status: 401 Unauthorized Size: 39 Bytes Time: 160 ms

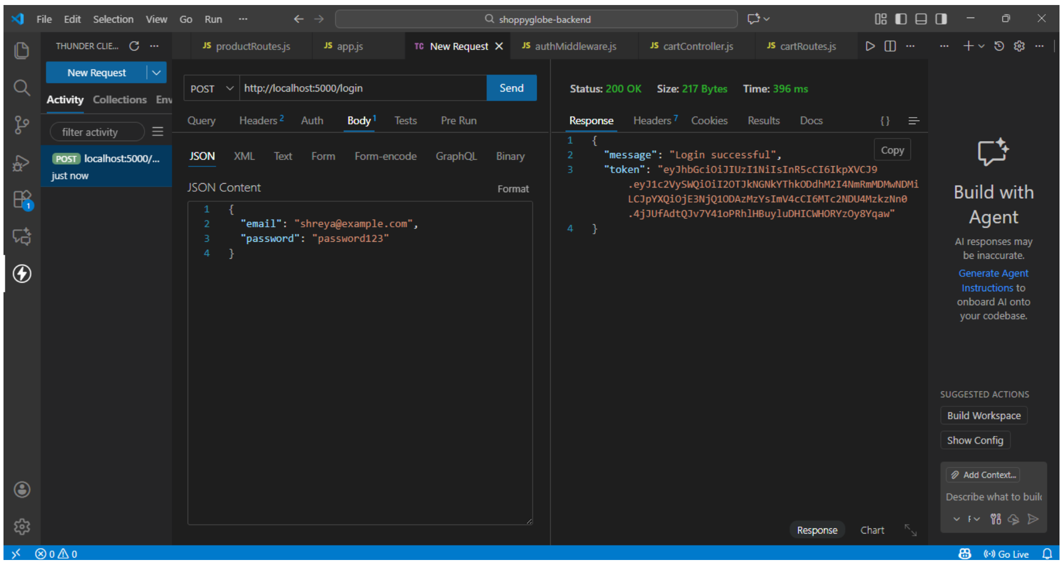
Response
```
{
  "message": "Invalid email or password"
}
```

2.4 POST /login - Successful Login

Description:
Generates JWT token for authenticated users.

Expected Behavior:

- Status 200
- Message: "Login successful"
- Returns valid JWT token



POST http://localhost:5000/login

Status: 200 OK   Size: 217 Bytes   Time: 396 ms

```
{
  "email": "shreya@example.com",
  "password": "password123"
}
```

```
{
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
      .eyJ1c2VySWQiOiI2OTJkNGNkYThkODdhM2I4NmRmMDMwNDMi
      LCJpYXQiOjE3NjQ1ODAzMzYsImV4cCI6MTc2NDU4MzkzNn0
      .4jJUfAdtQJv7Y41oPRhlHBuyluDHICWHORYzOy8Yqaw"
}
```

3. Product API Testing

3.1 POST /products - Add Product

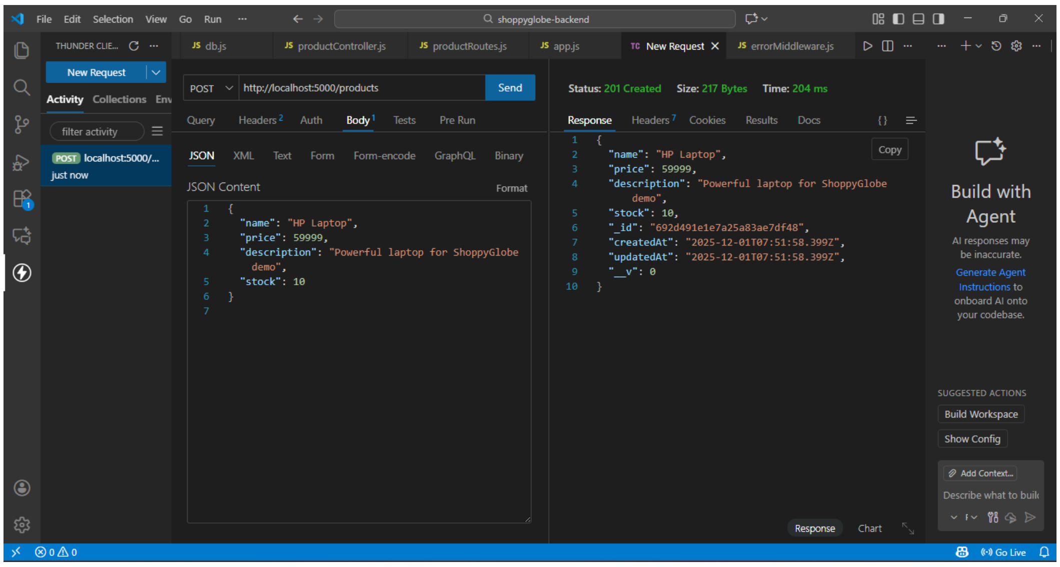
Description:
Creates a new product in the database.

Input Example:
name, price, description, stock

Expected Behavior:

- Status 201
- Returns full product object

Screenshot:



POST http://localhost:5000/products Send

```
{
  "name": "HP Laptop",
  "price": 59999,
  "description": "Powerful laptop for ShoppyGlobe
    demo",
  "stock": 10
}
```

Status: 201 Created Size: 217 Bytes Time: 204 ms

```
{
  "name": "HP Laptop",
  "price": 59999,
  "description": "Powerful laptop for ShoppyGlobe
    demo",
  "stock": 10,
  "_id": "692d491e1e7a25a83ae7df48",
  "createdAt": "2025-12-01T07:51:58.399Z",
  "updatedAt": "2025-12-01T07:51:58.399Z",
  "__v": 0
}
```

3.2 GET /products - Empty Product List

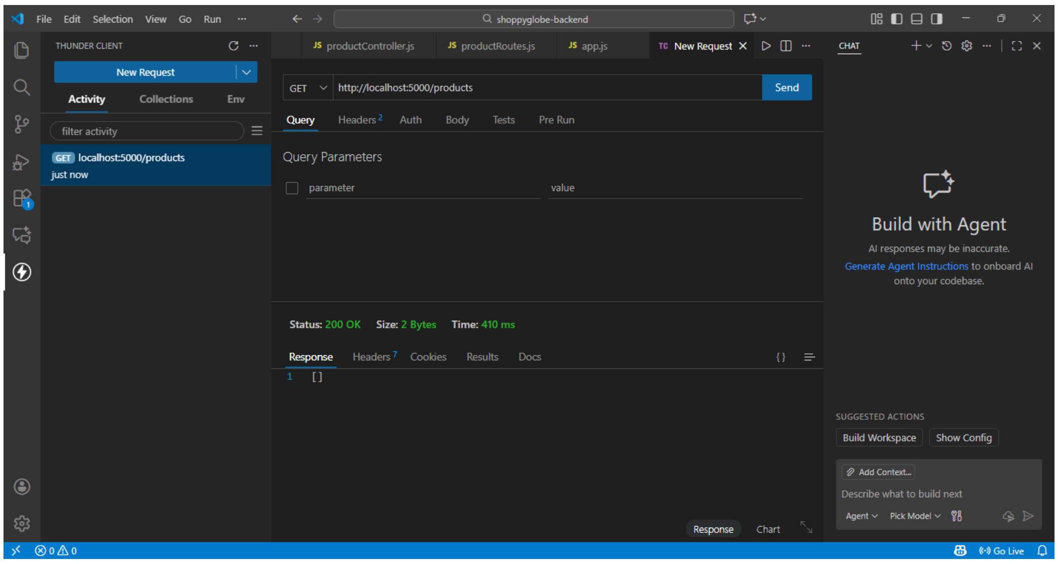
Description:

Checks default behavior when DB has no products initially.

Expected Behavior:

- Status 200
- Empty array []

Screenshot:



GET http://localhost:5000/products Send

Status: 200 OK Size: 2 Bytes Time: 410 ms

```
[]
```

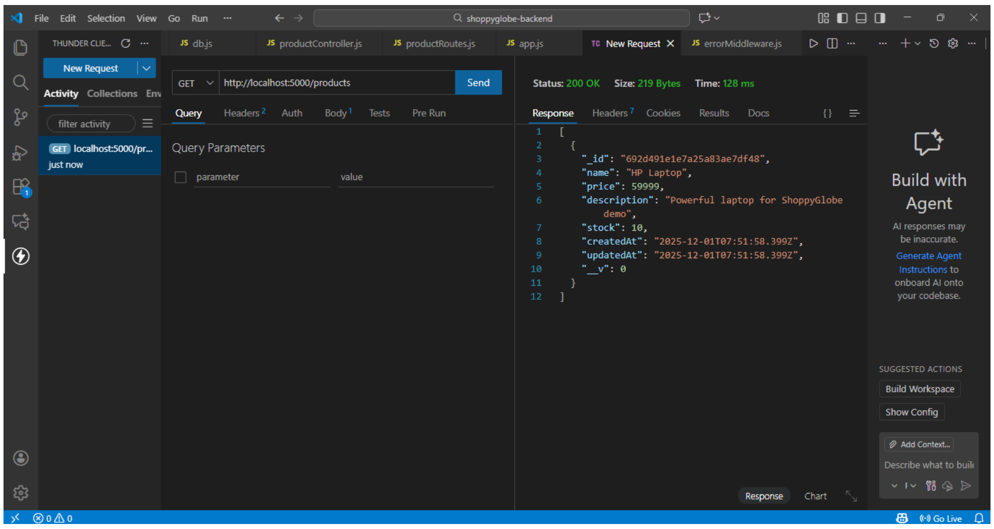
3.3 GET /products - List All Products After Insertion

Description:
Fetches all products now present in DB.

Expected Behavior:

- Status 200
- Returns array with product details

Screenshot:



GET http://localhost:5000/products

Status: 200 OK Size: 219 Bytes Time: 128 ms

```
[
  {
    "_id": "692d491e1e7a25a83ae7df48",
    "name": "HP Laptop",
    "price": 59999,
    "description": "Powerful laptop for ShoppyGlobe
      demo",
    "stock": 10,
    "createdAt": "2025-12-01T07:51:58.399Z",
    "updatedAt": "2025-12-01T07:51:58.399Z",
    "__v": 0
  }
]
```

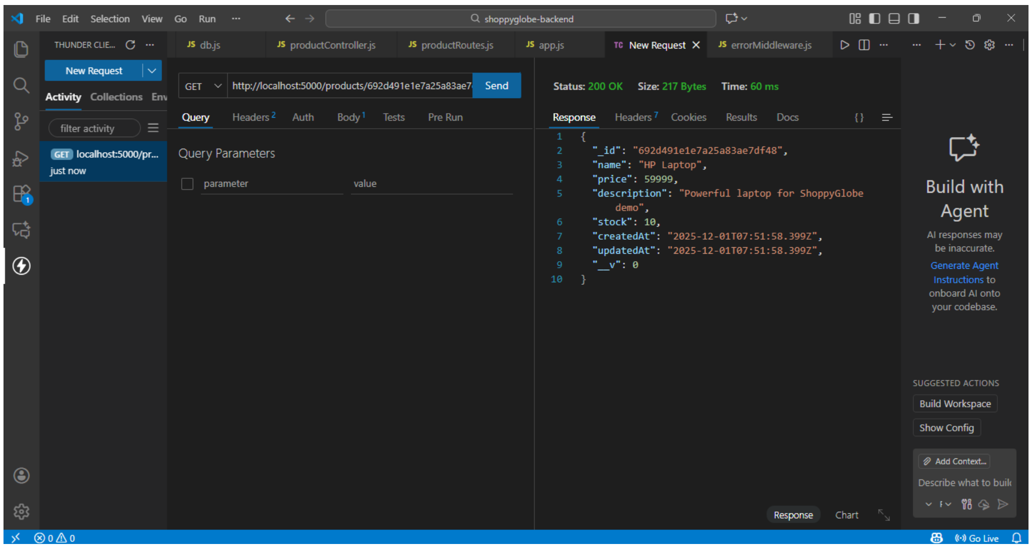
3.4 GET /products/:id - Valid ID

Description:
Fetches single product by ObjectId.

Expected Behavior:

- Status 200
- Returns single product

Screenshot:



GET http://localhost:5000/products/692d491e1e7a25a83ae7

Status: 200 OK Size: 217 Bytes Time: 60 ms

```
{
  "_id": "692d491e1e7a25a83ae7df48",
  "name": "HP Laptop",
  "price": 59999,
  "description": "Powerful laptop for ShoppyGlobe
    demo",
  "stock": 10,
  "createdAt": "2025-12-01T07:51:58.399Z",
  "updatedAt": "2025-12-01T07:51:58.399Z",
  "__v": 0
}
```

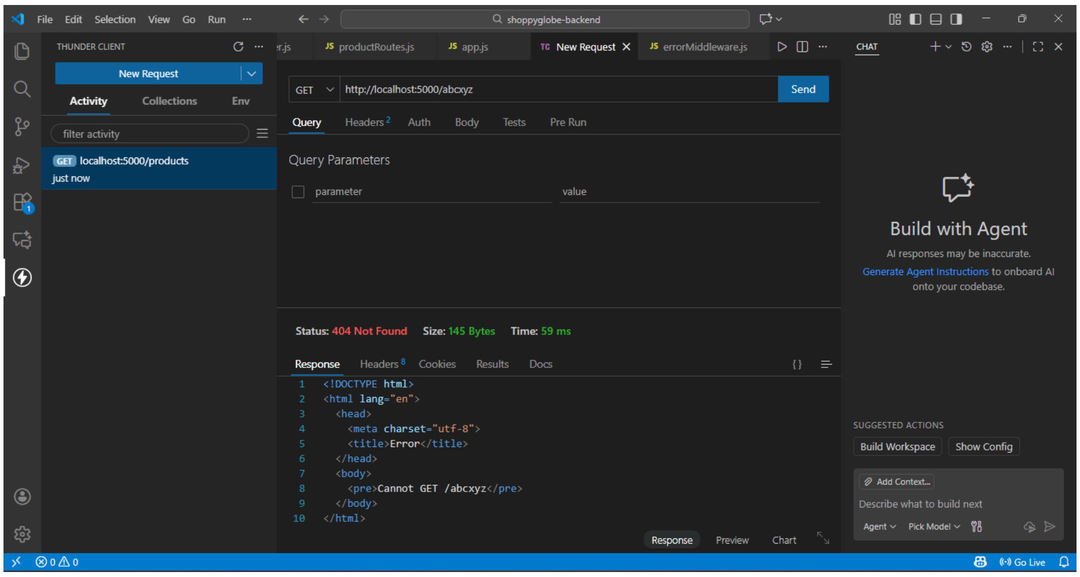
3.5 GET /products/:id - Invalid Route

Description:
Tests Express default 404 response.

Expected Behavior:

- Status 404
- “Cannot GET /abcxyz”

Screenshot:



GET http://localhost:5000/abcxyz Send

Status: 404 Not Found Size: 145 Bytes Time: 59 ms

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Error</title>
  </head>
  <body>
    <pre>Cannot GET /abcxyz</pre>
  </body>
</html>
```

AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

Describe what to build next

<u>4. Cart API Testing</u>

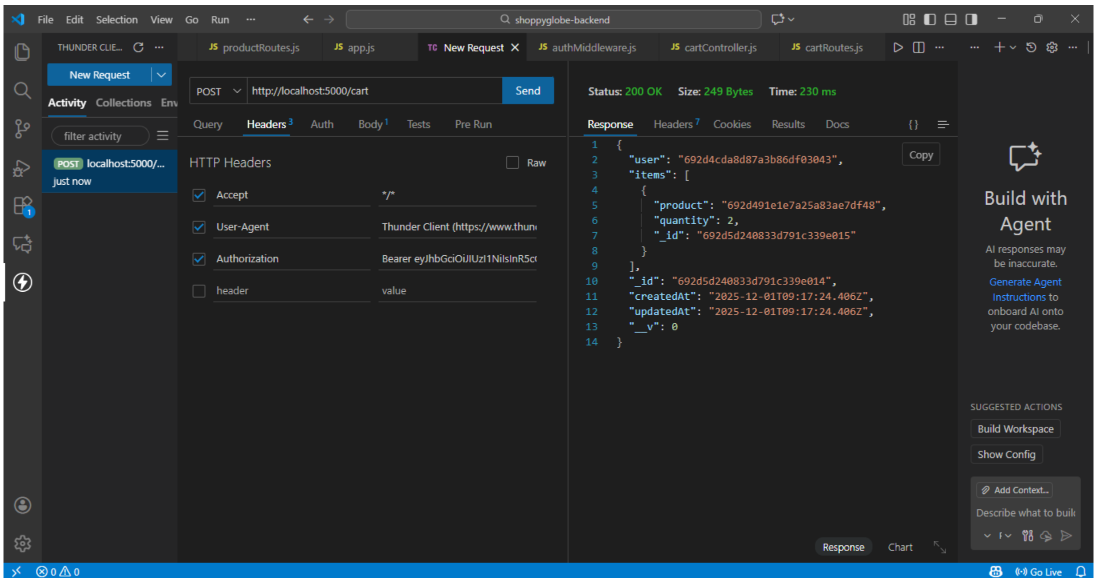
4.1 POST /cart - Add Item to Cart (Valid Token)

Description:
Adds selected product to user’s cart.

Expected Behavior:

- Status 200
- Returns cart with items array

Screenshot:



POST http://localhost:5000/cart

Status: 200 OK Size: 249 Bytes Time: 230 ms

HTTP Headers

✓ Accept	*/*
✓ User-Agent	Thunder Client (https://www.thun
✓ Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5c
header	value

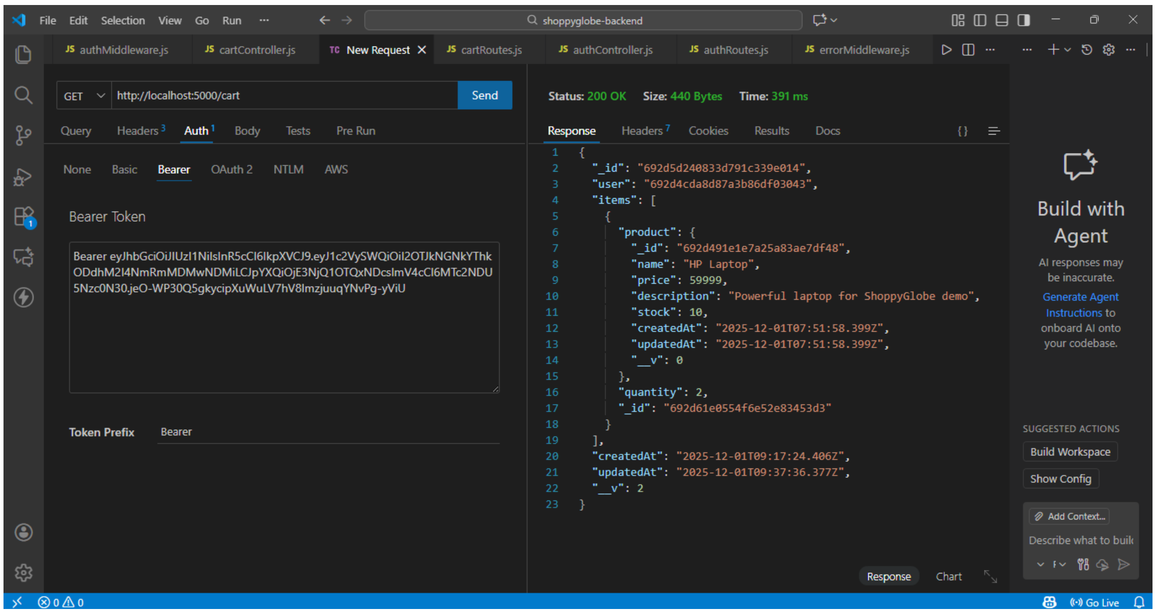
```
{
  "user": "692d4cda8d87a3b86df03043",
  "items": [
    {
      "product": "692d491e1e7a25a83ae7df48",
      "quantity": 2,
      "_id": "692d5d240833d791c339e015"
    }
  ],
  "_id": "692d5d240833d791c339e014",
  "createdAt": "2025-12-01T09:17:24.406Z",
  "updatedAt": "2025-12-01T09:17:24.406Z",
  "__v": 0
}
```

4.2 GET /cart - Fetch Cart (Valid Token)

Description:
Retrieves the current authenticated user's cart. Returns all products, quantities, and cart metadata.

Expected Behavior:

- Status 200
- Returns cart document with items[], product details, and timestamps



GET http://localhost:5000/cart

Auth: Bearer

Bearer Token

```
Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2OTJkNGNkYThkODdhM2I4NmRmMDMwNDMiLCJpYXQiOjE3NjQ1OTQxNDcsImV4cCI6MTc2NDU5NzcON30.jeO-WP30Q5gkycipXuWuLV7hV8ImzjuuqYNvPg-yViU
```

Token Prefix: Bearer

Status: 200 OK Size: 440 Bytes Time: 391 ms

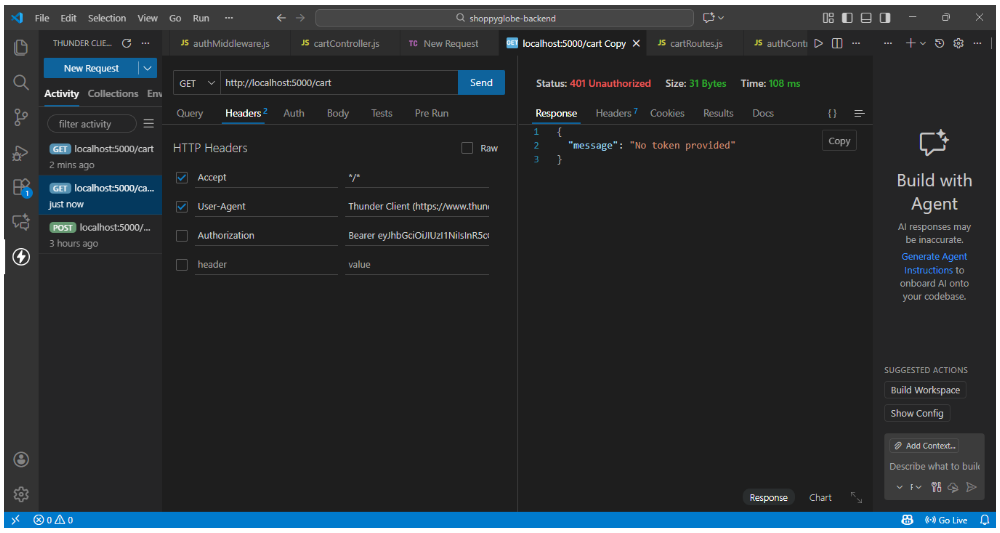
```
{
  "_id": "692d5d240833d791c339e014",
  "user": "692d4cda8d87a3b86df03043",
  "items": [
    {
      "product": {
        "_id": "692d491e1e7a25a83ae7df48",
        "name": "HP Laptop",
        "price": 59999,
        "description": "Powerful laptop for ShoppyGlobe demo",
        "stock": 10,
        "createdAt": "2025-12-01T07:51:58.399Z",
        "updatedAt": "2025-12-01T07:51:58.399Z",
        "__v": 0
      },
      "quantity": 2,
      "_id": "692d61e0554f6e52e83453d3"
    }
  ],
  "createdAt": "2025-12-01T09:17:24.406Z",
  "updatedAt": "2025-12-01T09:37:36.377Z",
  "__v": 2
}
```

4.3 GET /cart - Missing Token (Unauthorized)

Description:
Checks API behavior when no JWT token is passed in the Authorization header.

Expected Behavior:

- Status 401
- Message: "No token provided"



GET http://localhost:5000/cart

HTTP Headers

✓	Accept	*/*
✓	User-Agent	Thunder Client (https://www.thun
	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5c
	header	value

Status: 401 Unauthorized Size: 31 Bytes Time: 108 ms

```
{
  "message": "No token provided"
}
```

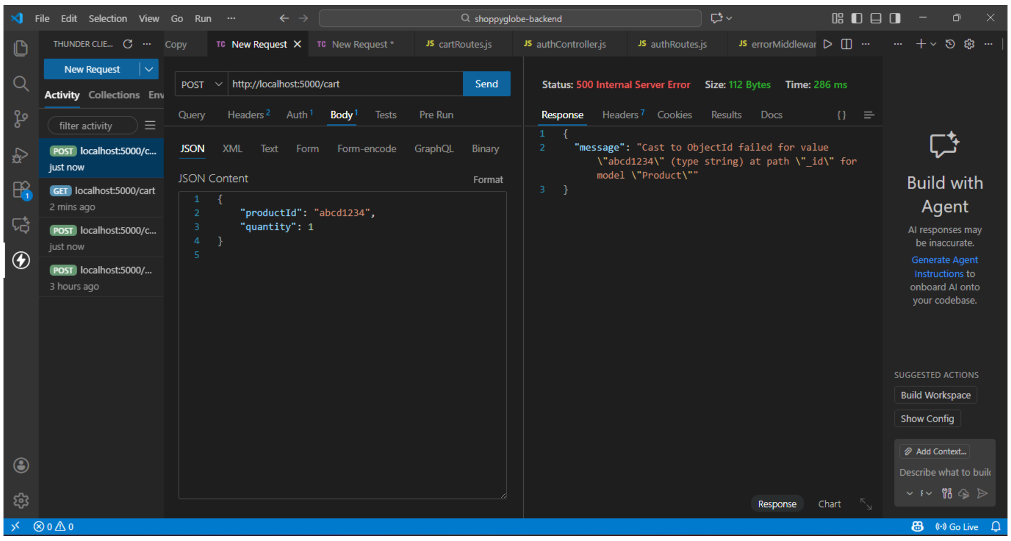
4.4 POST /cart – Invalid ProductId Format

Description:
Validates MongoDB ObjectId format. If productId is not a valid ObjectId, API must throw validation error.

Expected Behavior:

- Status 500 Internal Server Error
- Message: "Cast to ObjectId failed for value … at path '_id' for model 'Product'"

Screenshot:



POST http://localhost:5000/cart

JSON Content

```
{
    "productId": "abcd1234",
    "quantity": 1
}
```

Status: 500 Internal Server Error Size: 112 Bytes Time: 286 ms

```
{
  "message": "Cast to ObjectId failed for value \"abcd1234\" (type string) at path \"_id\" for model \"Product\""
}
```

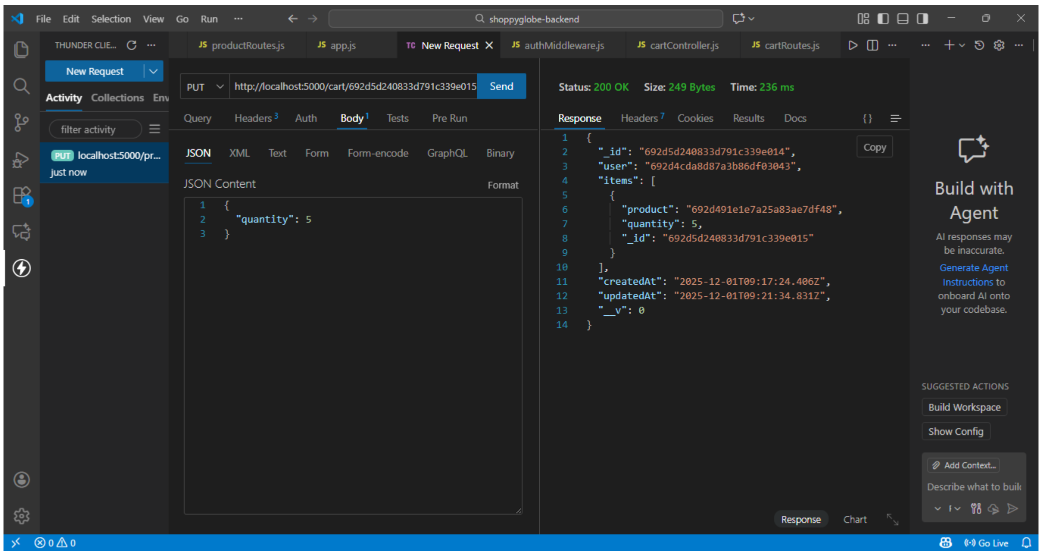
4.5 PUT /cart/:itemId - Update Quantity

Description:
Allows modifying quantity of an existing item.

Expected Behavior:

- Status 200
- Updated cart returned

Screenshot:



PUT http://localhost:5000/cart/692d5d240833d791c339e015

```
{
  "quantity": 5
}
```

Status: 200 OK Size: 249 Bytes Time: 236 ms

```
{
  "_id": "692d5d240833d791c339e014",
  "user": "692d4cda8d87a3b86df03043",
  "items": [
    {
      "product": "692d491e1e7a25a83ae7df48",
      "quantity": 5,
      "_id": "692d5d240833d791c339e015"
    }
  ],
  "createdAt": "2025-12-01T09:17:24.406Z",
  "updatedAt": "2025-12-01T09:21:34.831Z",
  "__v": 0
}
```

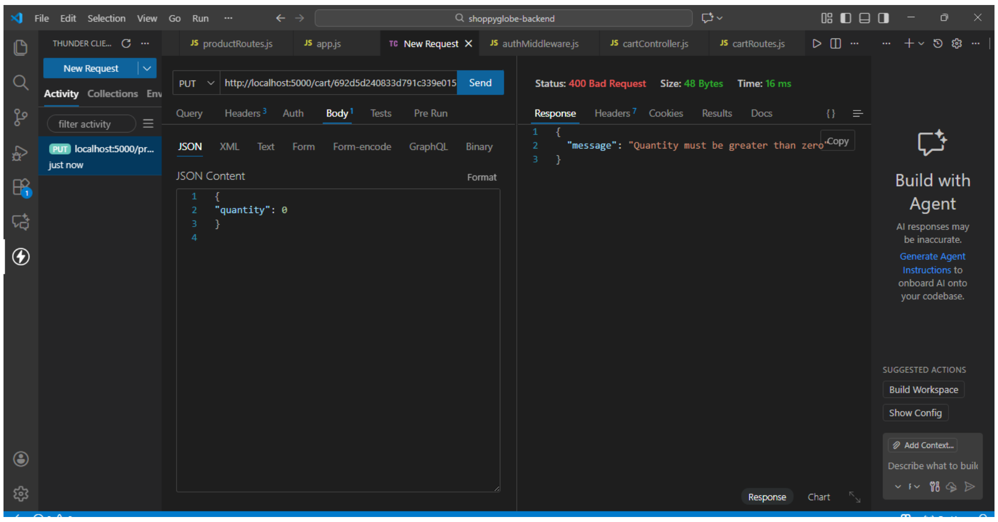
4.6 PUT /cart/:itemId - Quantity ≤ 0 Error

Description:
Validation to prevent invalid quantities.

Expected Behavior:

- Status 400
- Message: "Quantity must be greater than zero"

Screenshot:



PUT http://localhost:5000/cart/692d5d240833d791c339e015

```
{
"quantity": 0
}
```

Status: 400 Bad Request Size: 48 Bytes Time: 16 ms

```
{
  "message": "Quantity must be greater than zero"
}
```

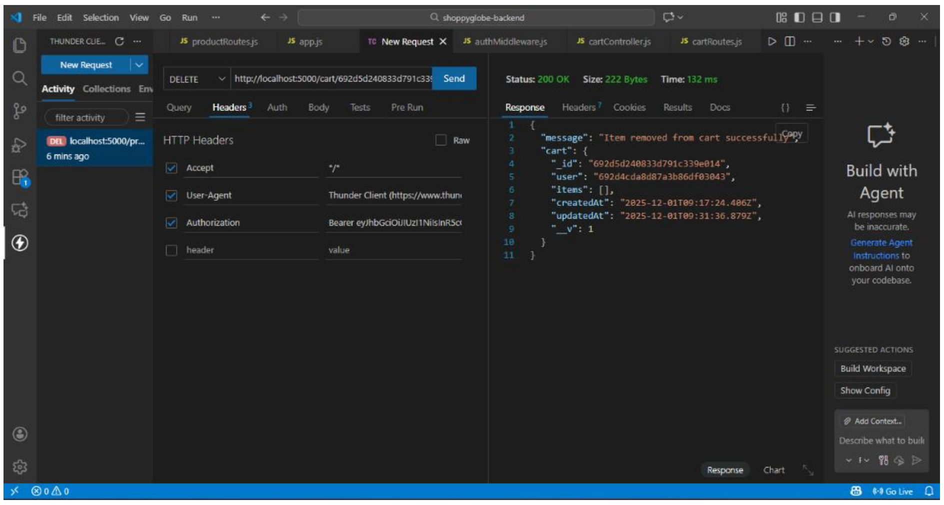
4.7 DELETE /cart/:itemId - Remove Item

Description:
Removes one cart item by its subdocument ID.

Expected Behavior:

- Status 200
- item removed confirmation
- Empty cart (if no items left)

Screenshot:



DELETE http://localhost:5000/cart/692d5d240833d791c33

Status: 200 OK Size: 222 Bytes Time: 132 ms

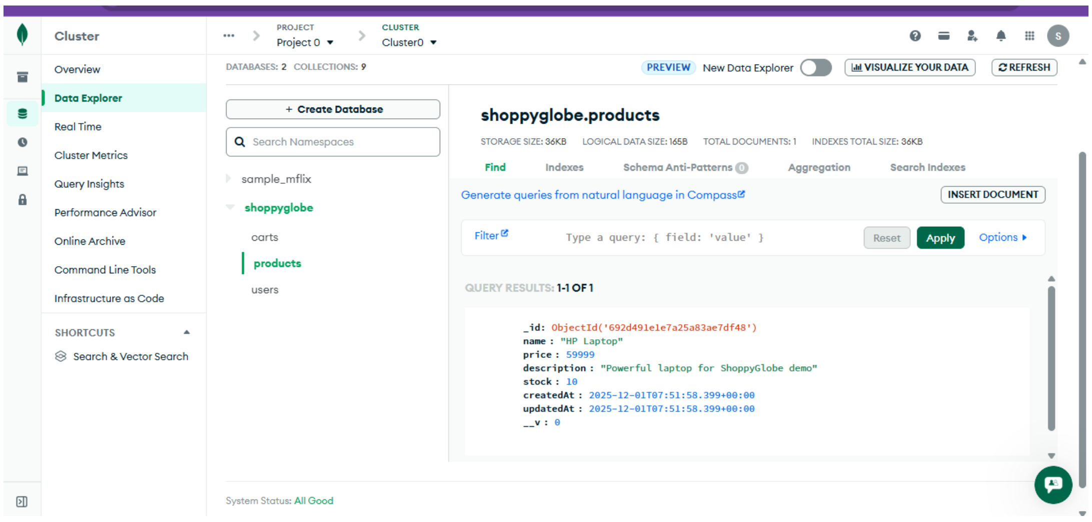
	HTTP Headers	
✓	Accept	*/*
✓	User-Agent	Thunder Client (https://www.thun
✓	Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5c
	header	value

```
{
  "message": "Item removed from cart successfully",
  "cart": {
    "_id": "692d5d240833d791c339e014",
    "user": "692d4cda8d87a3b86df03043",
    "items": [],
    "createdAt": "2025-12-01T09:17:24.406Z",
    "updatedAt": "2025-12-01T09:31:36.879Z",
    "__v": 1
  }
}
```

5. MongoDB Atlas Verification

5.1 products Collection

Description:
Verifies product was created correctly.



shoppyglobe.products

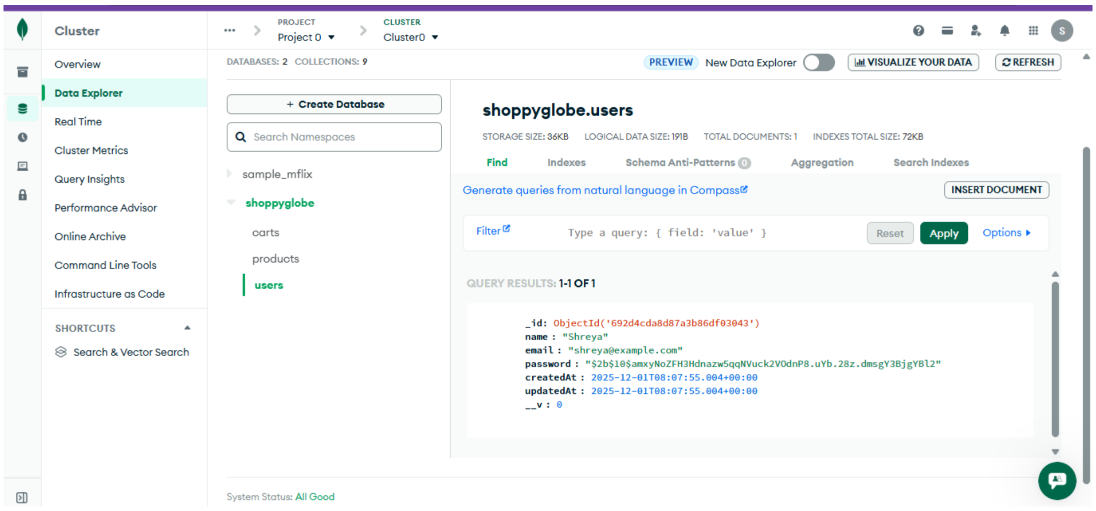
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 165B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('692d491e1e7a25a83ae7df48')
name : "HP Laptop"
price : 59999
description : "Powerful laptop for ShoppyGlobe demo"
stock : 10
createdAt : 2025-12-01T07:51:58.399+00:00
updatedAt : 2025-12-01T07:51:58.399+00:00
__v : 0
```

5.2 users Collection

Description:
Ensures user registration stored hashed passwords.

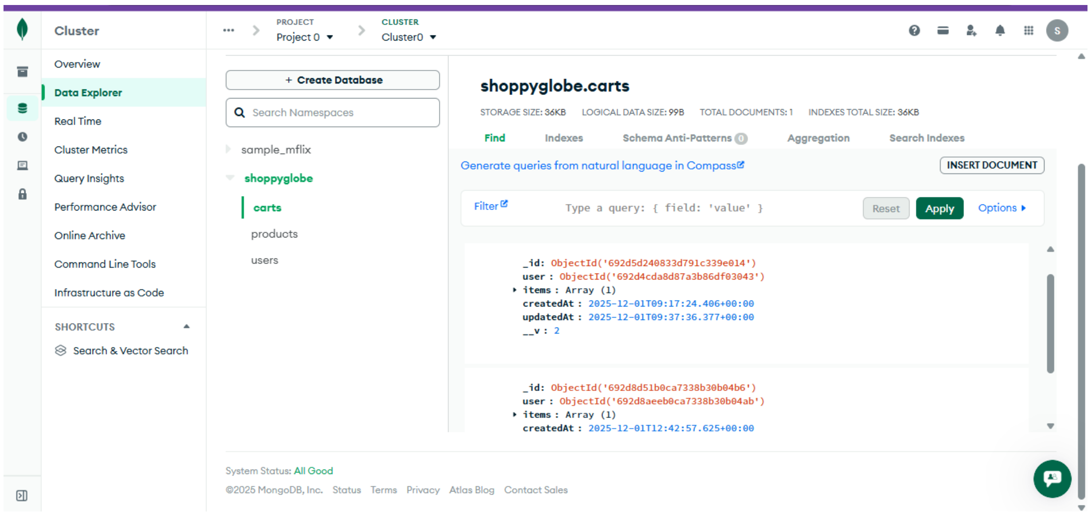


shoppyglobe.users

```
_id: ObjectId('692d4cda8d87a3b86df03043')
name : "Shreya"
email : "shreya@example.com"
password : "$2b$10$amxyNoZFH3Hdnazw5qqNVuck2VOdnP8.uYb.28z.dmsgY3BjgYBl2"
createdAt : 2025-12-01T08:07:55.004+00:00
updatedAt : 2025-12-01T08:07:55.004+00:00
__v : 0
```

5.3 carts Collection

Description:
Verifies cart creation + item details.



shoppyglobe.carts

```
  _id: ObjectId('692d5d240833d791c339e014')
  user : ObjectId('692d4cda8d87a3b86df03043')
▸ items : Array (1)
  createdAt : 2025-12-01T09:17:24.406+00:00
  updatedAt : 2025-12-01T09:37:36.377+00:00
  __v : 2

  _id: ObjectId('692d8d51b0ca7338b30b04b6')
  user : ObjectId('692d8aeeb0ca7338b30b04ab')
▸ items : Array (1)
  createdAt : 2025-12-01T12:42:57.625+00:00
```

6. Conclusion

All endpoints were tested thoroughly, covering:

- Success responses
- Token authentication
- Error handling
- Validation failures
- Database state verification

The API behaves consistently, returns appropriate messages, and supports secure CRUD operations.

Github Link: https://github.com/Shreyasri30/shoppyglobe-backend