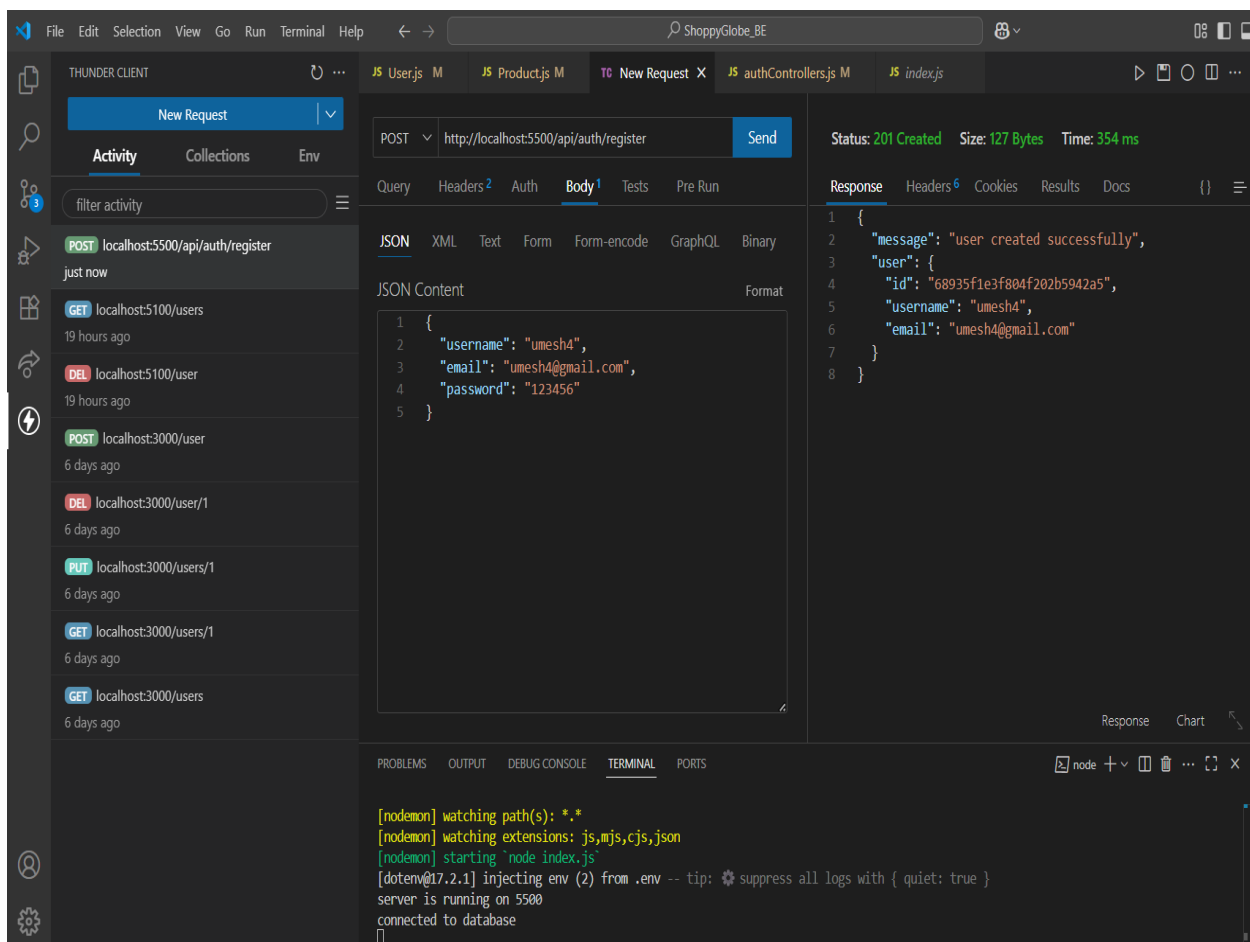


ShoppY Globe Backend

User using http method and jwt token and password encrypted

Post for register



The screenshot displays the Thunder Client interface with a new request configured for a POST method to the endpoint `http://localhost:5500/api/auth/register`. The request body is a JSON object containing the following data:

```
1 {
2   "username": "umesh4",
3   "email": "umesh4@gmail.com",
4   "password": "123456"
5 }
```

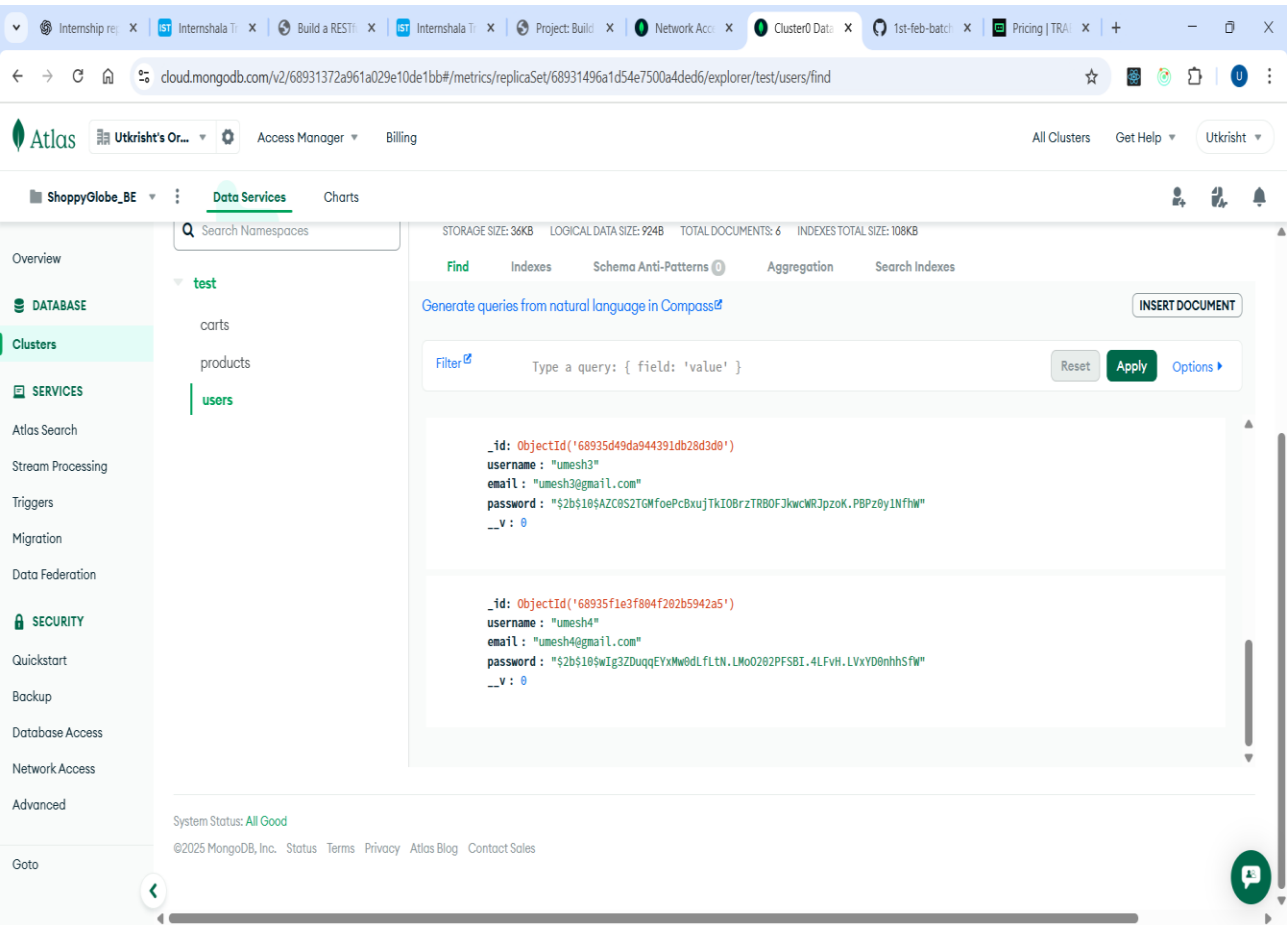
The response status is 201 Created, with a size of 127 Bytes and a time of 354 ms. The response body is a JSON object:

```
1 {
2   "message": "user created successfully",
3   "user": {
4     "id": "68935f1e3f804f202b5942a5",
5     "username": "umesh4",
6     "email": "umesh4@gmail.com"
7   }
8 }
```

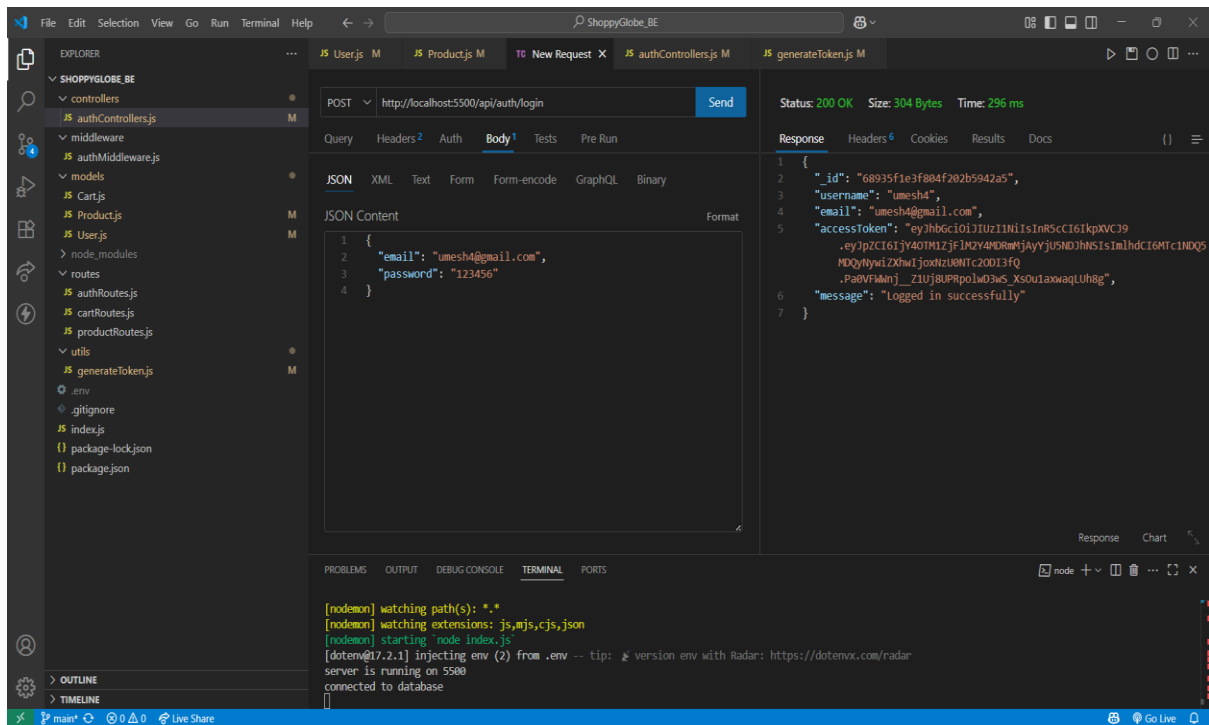
The bottom panel shows the terminal output, indicating that the server is running on port 5500 and connected to the database.

```
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node index.js'
[dotenv@17.2.1] injecting env (2) from .env -- tip: suppress all logs with { quiet: true }
server is running on 5500
connected to database
```

DataBase Update

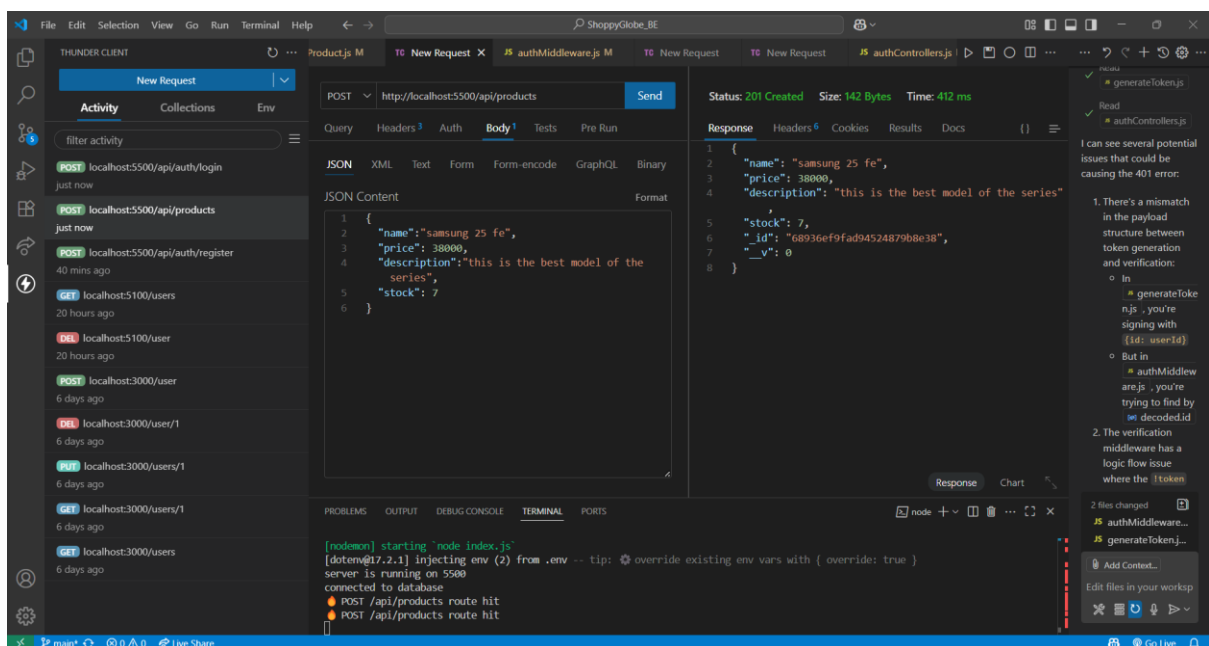


Post for Login

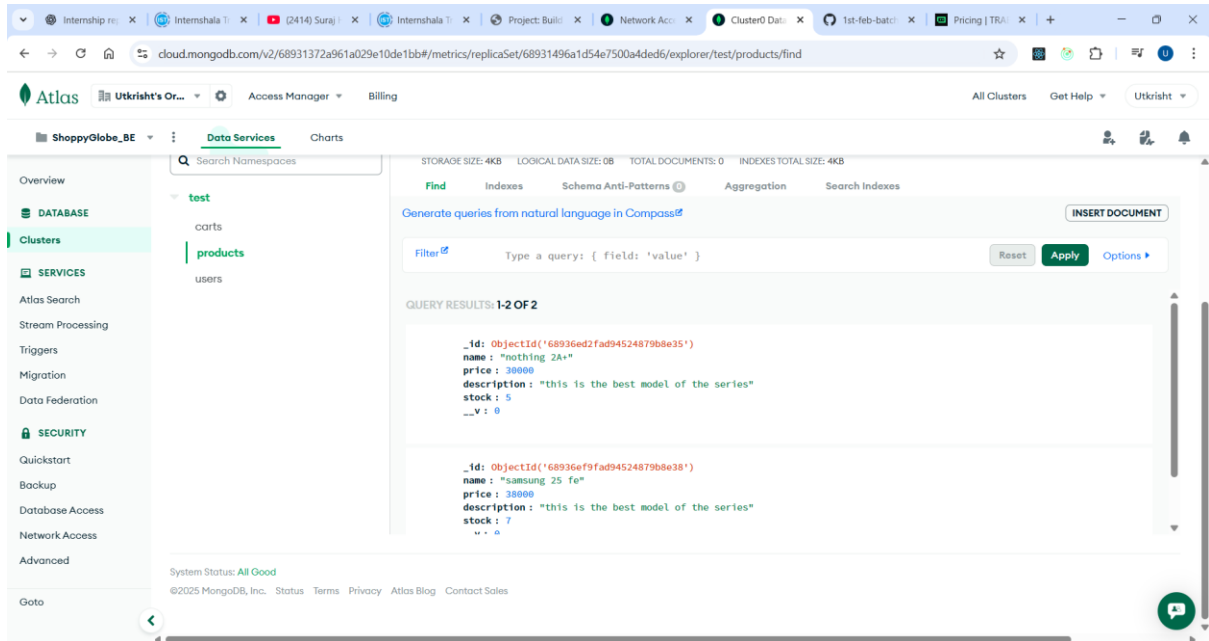


PRODUCT PAGE

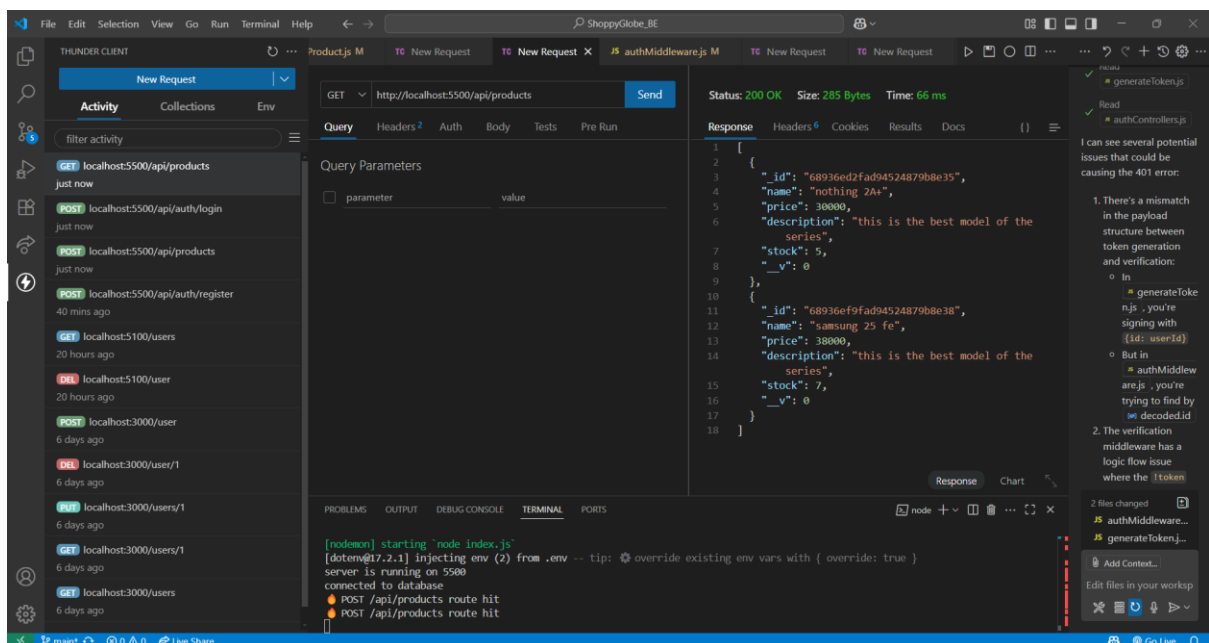
Post Method



DataBase Update



GET Method without Id



GET Method with ID

The screenshot displays the Thunder Client interface with a GET request to `http://localhost:5500/api/products/68936ef9ad94524879b8e38`. The response is a 200 OK status with a size of 142 Bytes and a time of 217 ms. The response body is a JSON object:

```
{
  "_id": "68936ef9ad94524879b8e38",
  "name": "samsung 25 fe",
  "price": 38000,
  "description": "this is the best model of the series",
  "stock": 7,
  "_v": 0
}
```

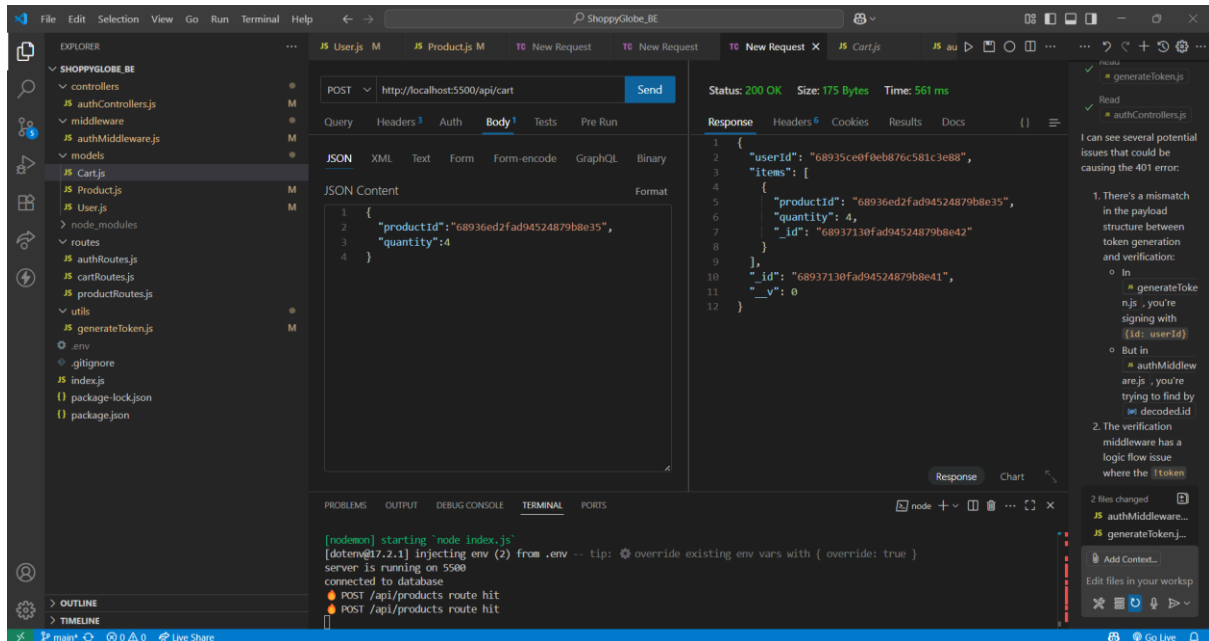
The left sidebar shows a list of recent requests, including GET requests to `localhost:5500/api/products` and `localhost:5100/users`, and POST requests to `localhost:5500/api/auth/login` and `localhost:5500/api/auth/register`. The bottom terminal shows the following output:

```
[nodemon] starting 'node index.js'
[dotenv@17.2.1] injecting env (2) from .env -- tip: override existing env vars with { override: true }
server is running on 5500
connected to database
POST /api/products route hit
POST /api/products route hit
```

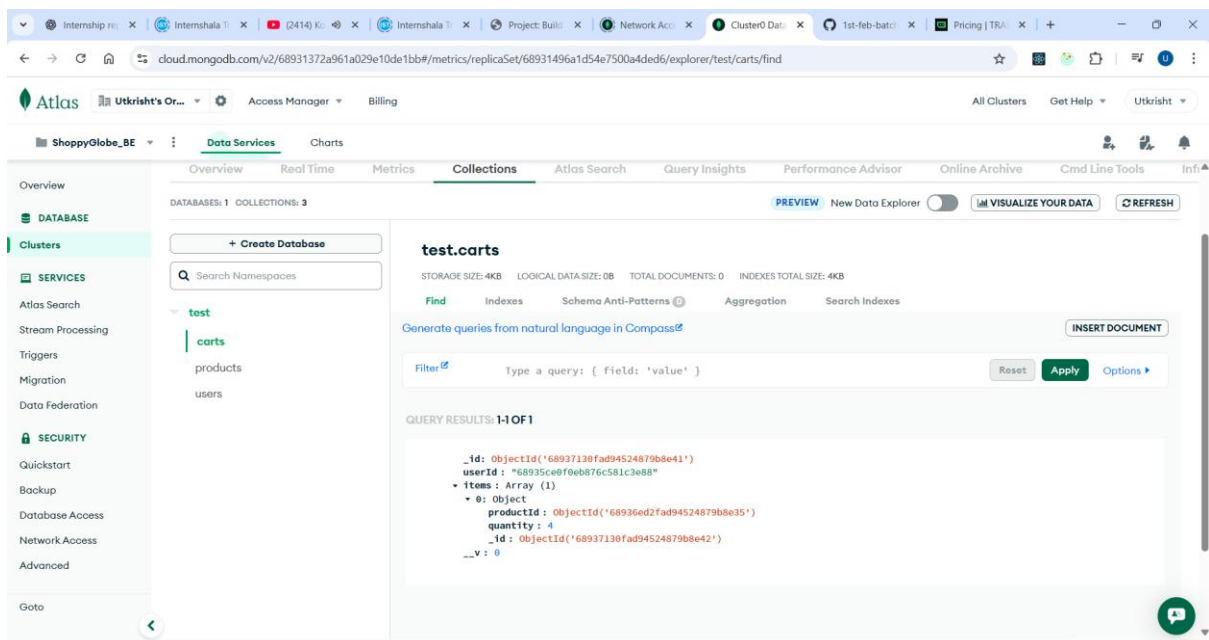
The right sidebar shows a list of files changed, including `generateToken.js`, `authControllers.js`, and `authMiddleware.js`. It also contains a note about a mismatch in the payload structure between token generation and verification, and a note about a logic flow issue in the verification middleware.

HTTP Methods for user Cart

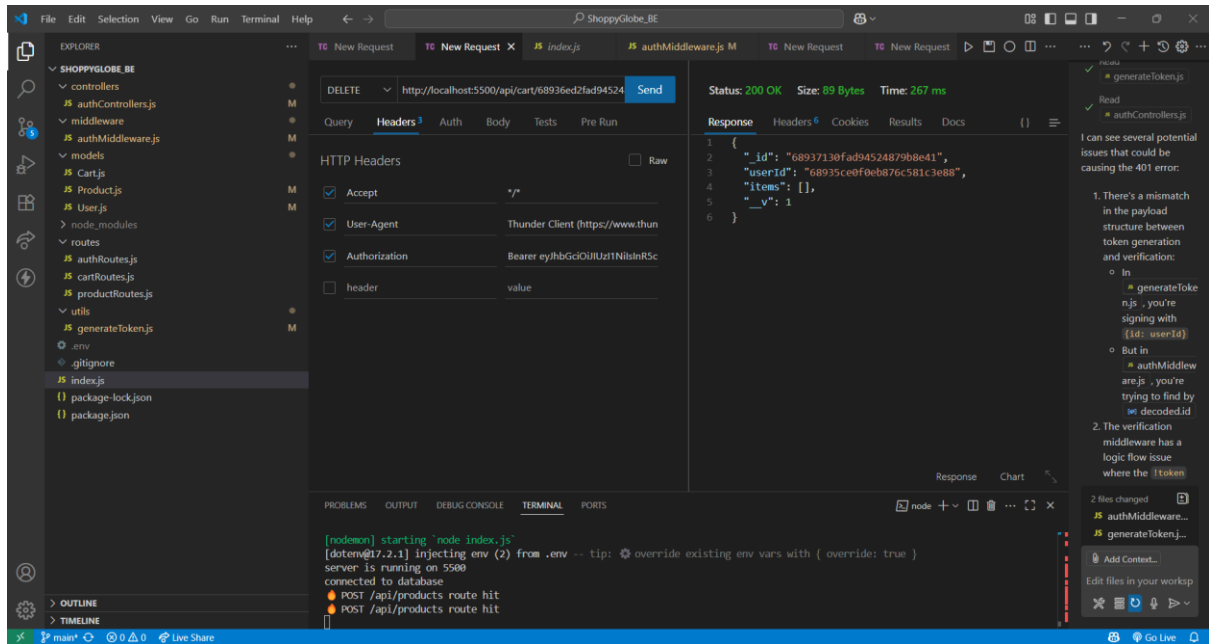
1. Post using product id in cart



DataBase Update



Delete Method for cart



Data Base Update(array is empty now)

