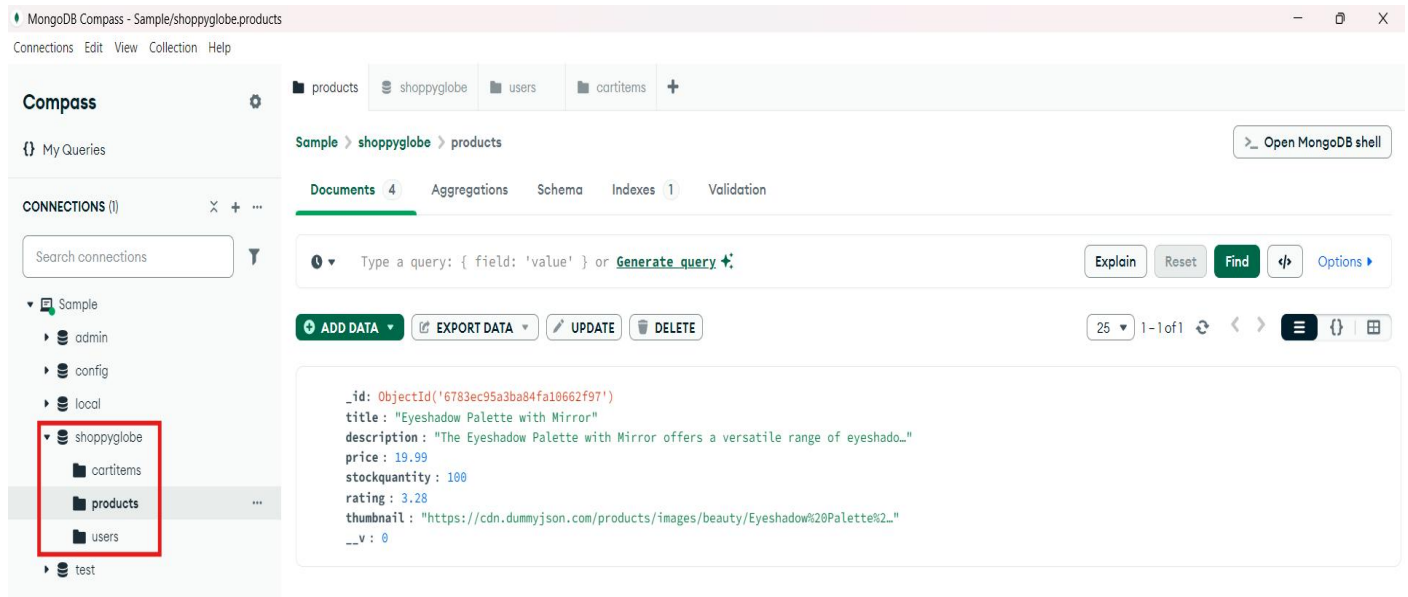


Shoppyglobe backend

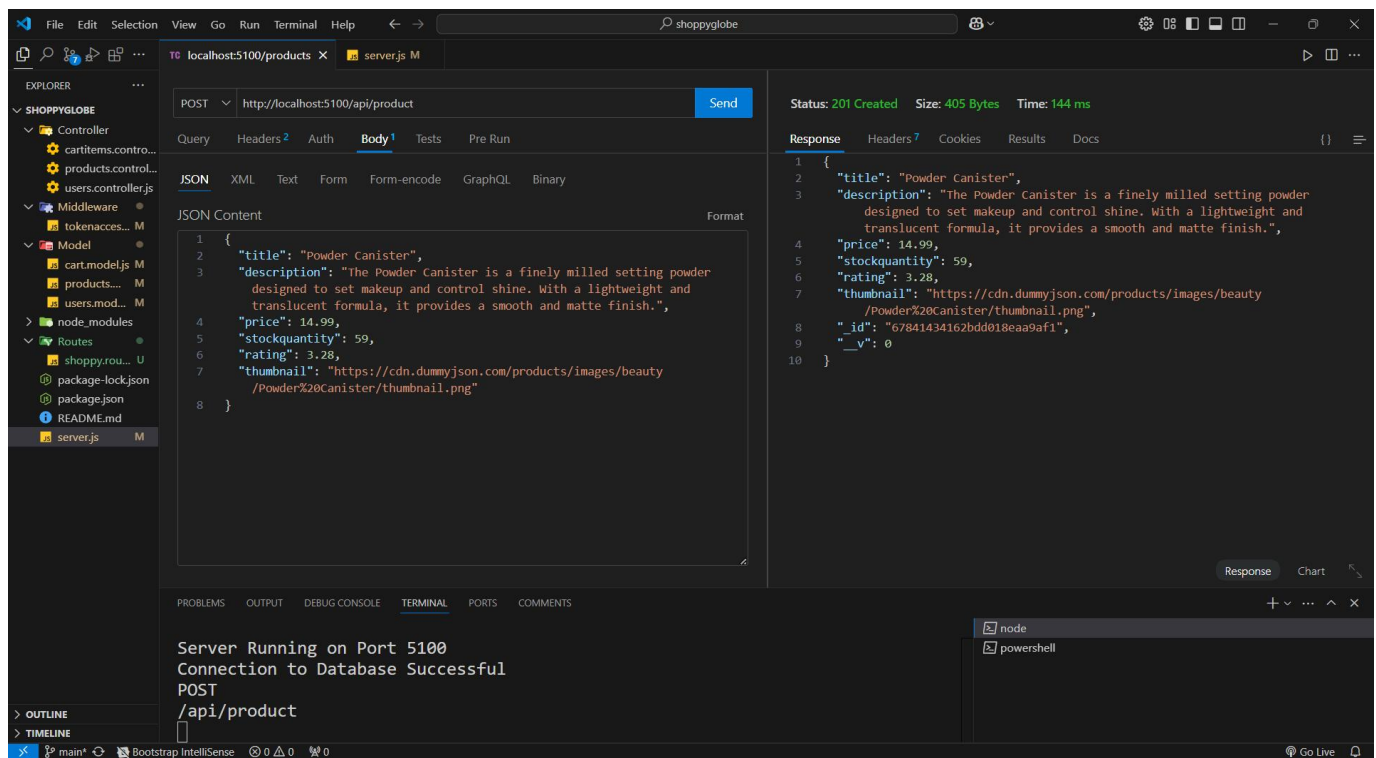
GITHUB LINK - https://github.com/Siddhesh-Dandekar/shoppyglobe_backend

MONGODB DATABASE



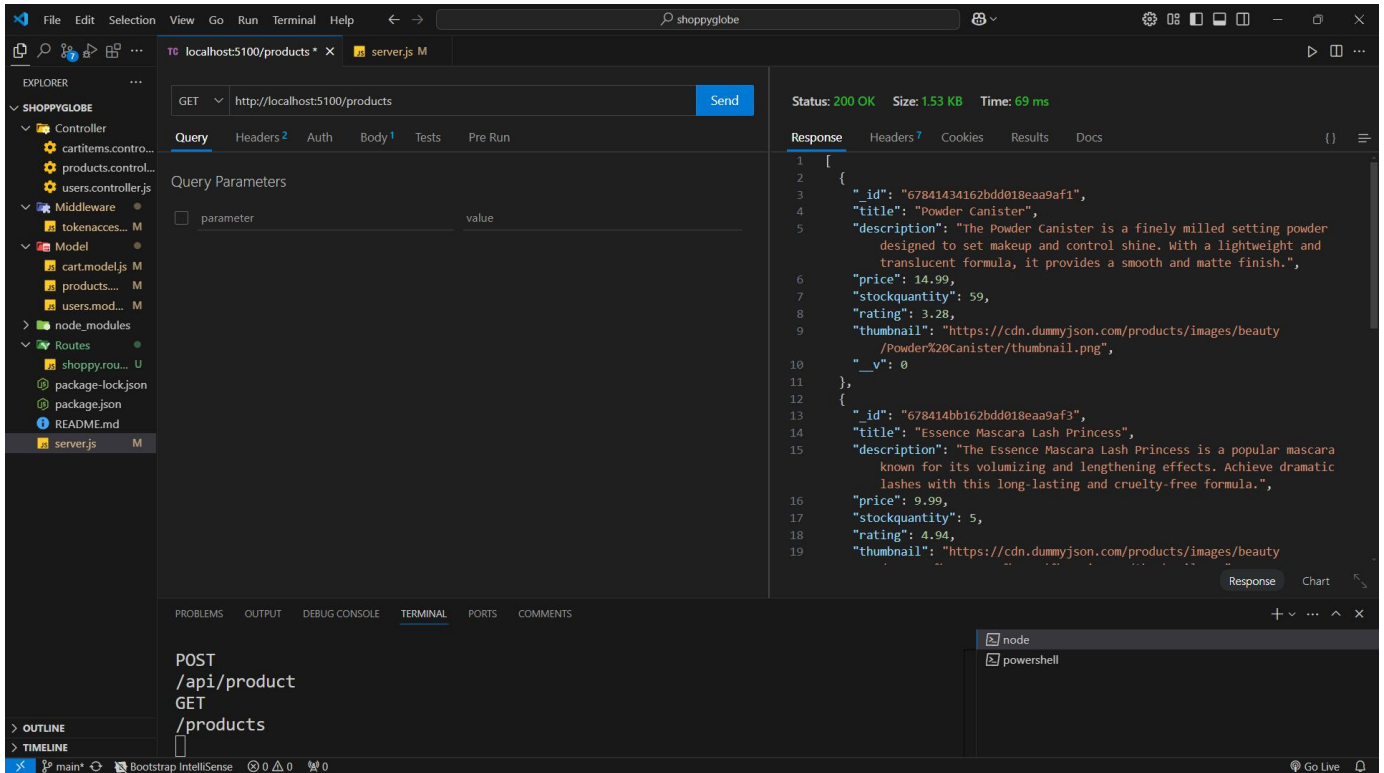
PRODUCT ROUTES

> ADD PRODUCTS ("/api/product") (WITHOUT TOKEN)



```
_id: ObjectId('67841434162bdd018eaa9af1')
title: "Powder Canister"
description: "The Powder Canister is a finely milled setting powder designed to set ..."
price: 14.99
stockquantity: 59
rating: 3.28
thumbnail: "https://cdn.dummyjson.com/products/images/beauty/Powder%20Canister/thu..."
__v: 0
```

> GET/products: Fetch a list of products from MongoDB.



The screenshot shows the VS Code interface with a REST client tab open for the URL `http://localhost:5100/products`. The request is a GET method. The response is a JSON array of two product objects, with a status of 200 OK, size of 1.53 KB, and time of 69 ms.

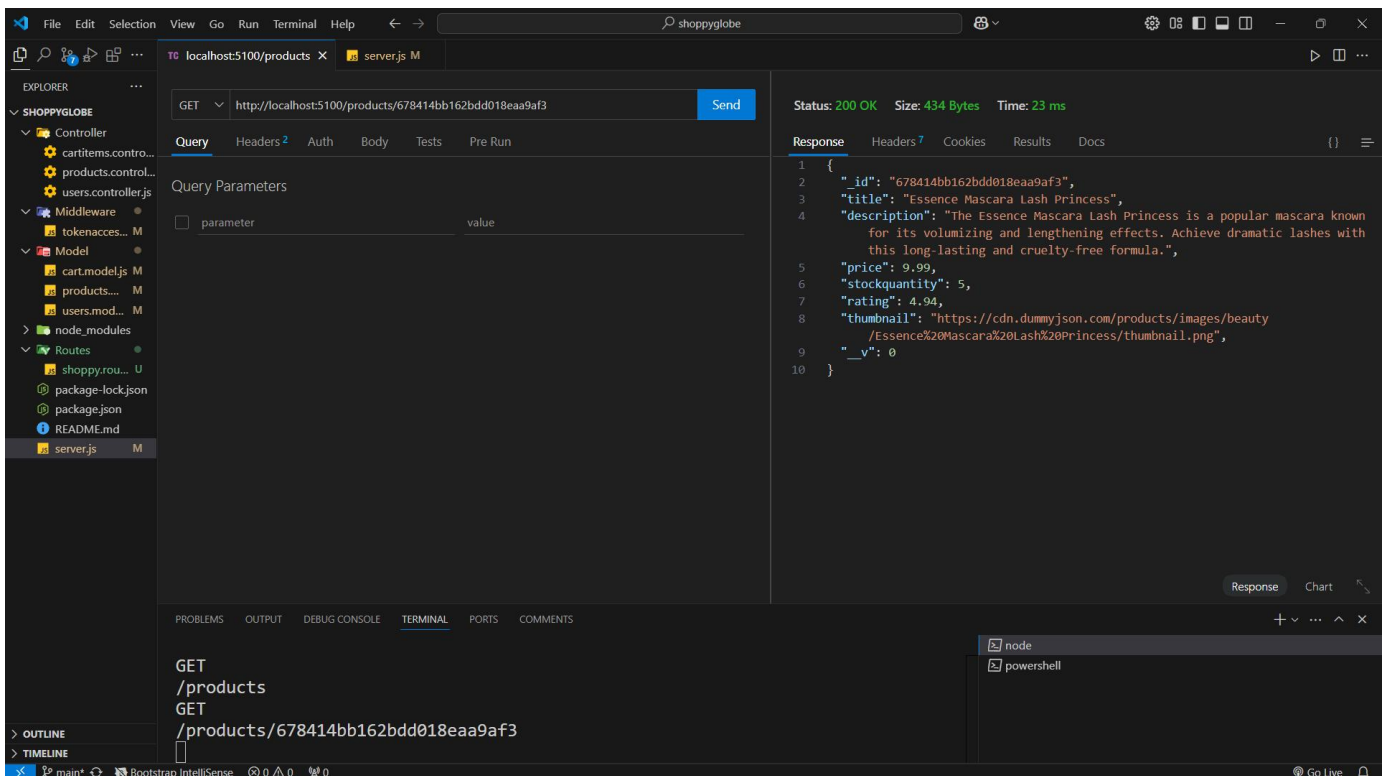
Query Parameters:

parameter	value
-----------	-------

Response:

```
[
  {
    "_id": "67841434162bdd018eaa9af1",
    "title": "Powder Canister",
    "description": "The Powder Canister is a finely milled setting powder designed to set makeup and control shine. With a lightweight and translucent formula, it provides a smooth and matte finish.",
    "price": 14.99,
    "stockquantity": 59,
    "rating": 3.28,
    "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Powder%20Canister/thumbnail.png",
    "_v": 0
  },
  {
    "_id": "678414bb162bdd018eaa9af3",
    "title": "Essence Mascara Lash Princess",
    "description": "The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve dramatic lashes with this long-lasting and cruelty-free formula.",
    "price": 9.99,
    "stockquantity": 5,
    "rating": 4.94,
    "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Essence%20Mascara%20Lash%20Princess/thumbnail.png",
    "_v": 0
  }
]
```

> GET/products/: Fetch details of a single product by its ID.



The screenshot shows the VS Code interface with a REST client tab open for the URL `http://localhost:5100/products/678414bb162bdd018eaa9af3`. The request is a GET method. The response is a single product object, with a status of 200 OK, size of 434 Bytes, and time of 23 ms.

Query Parameters:

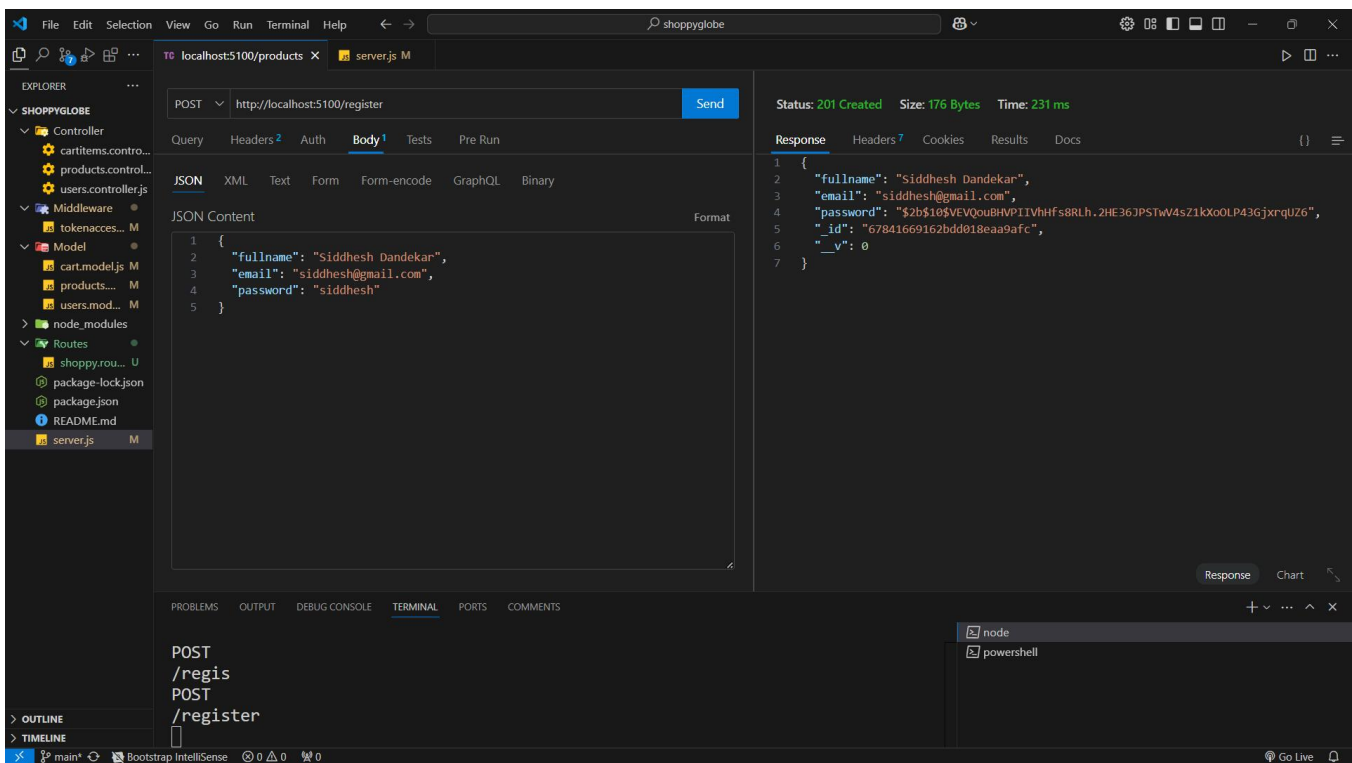
parameter	value
-----------	-------

Response:

```
{
  "_id": "678414bb162bdd018eaa9af3",
  "title": "Essence Mascara Lash Princess",
  "description": "The Essence Mascara Lash Princess is a popular mascara known for its volumizing and lengthening effects. Achieve dramatic lashes with this long-lasting and cruelty-free formula.",
  "price": 9.99,
  "stockquantity": 5,
  "rating": 4.94,
  "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Essence%20Mascara%20Lash%20Princess/thumbnail.png",
  "_v": 0
}
```

Create routes for User Registration and User Login.

> Register New user to Access CART items



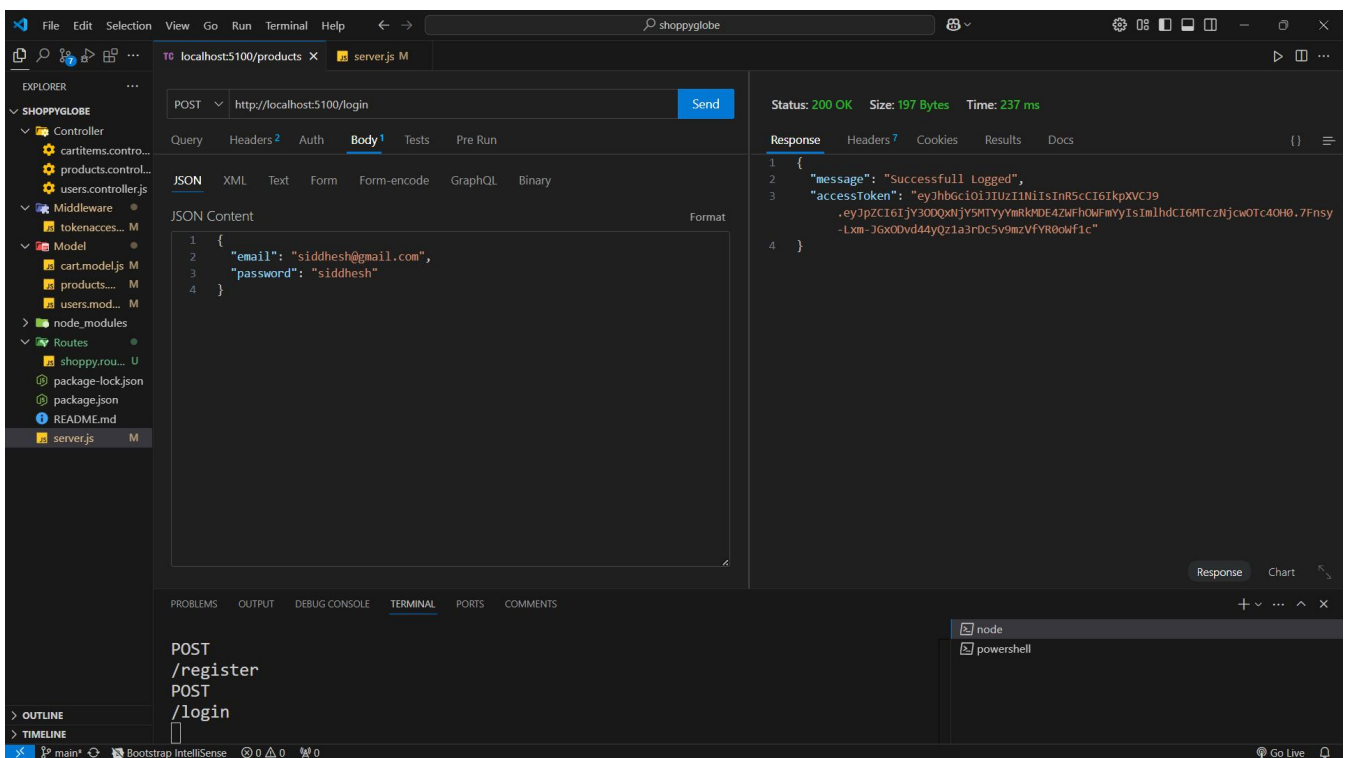
The screenshot shows the VS Code interface with a REST client request to `http://localhost:5100/register`. The request body is a JSON object:

```
1 {
2   "fullname": "Siddhesh Dandekar",
3   "email": "siddhesh@gmail.com",
4   "password": "siddhesh",
5 }
```

The response is a 201 Created status with a JSON object:

```
1 {
2   "fullname": "Siddhesh Dandekar",
3   "email": "siddhesh@gmail.com",
4   "password": "$2b$10$VEVQouBHVPIIVhHfs8RLh.2HE36JPSTw4sZ1kXo0LP43GjxrQz6",
5   "id": "67841669162bdd018eaa9afc",
6   "_v": 0
7 }
```

> Login with registered Credential to get Access Token



The screenshot shows the VS Code interface with a REST client request to `http://localhost:5100/login`. The request body is a JSON object:

```
1 {
2   "email": "siddhesh@gmail.com",
3   "password": "siddhesh"
4 }
```

The response is a 200 OK status with a JSON object:

```
1 {
2   "message": "Successfull Logged",
3   "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY3ODQxYjY5MTYyYmRkNDk0ZmFhZmFmYyIsIm1hdCI6MTczNjc4OTc4OH0.7Fnsy-Lxm-J6x00vd44yQz1a3r0c5v9mzVFYR0dwf1c"
4 }
```

> Using token to add product into cart

The screenshot shows the Thunder Client interface. The request is a POST to `http://localhost:5100/cart`. The Headers tab is selected, showing the following headers:

- `Accept`: `*/*`
- `User-Agent`: `Thunder Client (https://www.thunderclient.com)`
- `Authorization`: `JWT eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCkiOiJjY3ODQxLjY5MTYyYmRkNDc4ZWFiOWFmYyIsIm1hdCI6MTczNjcwOTc4OHB0.7Fnsy-LxmI-JGxODVd44yQz1a3rDc5v9mzVfYR0wMf1c`
- `header`: `value`

The Response tab shows a successful response with status 200 OK, size 197 Bytes, and time 237 ms. The response body is:

```
{
  "message": "Successful Logged",
  "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCkiOiJjY3ODQxLjY5MTYyYmRkNDc4ZWFiOWFmYyIsIm1hdCI6MTczNjcwOTc4OHB0.7Fnsy-LxmI-JGxODVd44yQz1a3rDc5v9mzVfYR0wMf1c"
}
```

PROTECTED CART ROUTES

> POST/cart: Add a product to the shopping cart.

> PRODUCT ADDED IN CART SUCCESSFULLY

The screenshot shows the VS Code editor with a REST client request in `server.js`. The request is a POST to `http://localhost:5100/products`. The Body tab is selected, showing the following JSON content:

```
{
  "productid": "67841434162bdd018eaa9af1"
}
```

The Response tab shows a successful response with status 201 Created, size 229 Bytes, and time 72 ms. The response body is:

```
{
  "productid": "67841434162bdd018eaa9af1",
  "title": "Powder Canister",
  "price": 14.99,
  "quantity": 1,
  "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Powder%20Canister/thumbnail.png",
  "_id": "6784178a162bdd018eaa9b02",
  "__v": 0
}
```

The Explorer sidebar shows the project structure, including the `server.js` file. The Terminal panel at the bottom shows the command `node` being executed.

> TRYING AGAIN TO ADD SAME PRODUCT

The screenshot shows the VS Code interface with a REST client request to `http://localhost:5100/cart` using the POST method. The request body is a JSON object: `{ "productid": "67841434162bdd018eaa9af1" }`. The response status is **409 Conflict**, with a size of 43 Bytes and a time of 34 ms. The response body is: `{ "message": "Product Already added in cart" }`. The Explorer panel on the left shows the project structure for 'SHOPPYGLOBE', including files like `cartitems.c...`, `products.control...`, `users.controller.js`, `Middleware`, `tokenacces...`, `Model`, `cart.model.js`, `products...`, `users.mod...`, `node_modules`, `Routes`, `shoppy.rou...`, `package-lock.json`, `package.json`, `README.md`, and `server.js`. The Terminal panel at the bottom shows the command `GET /products/67841434162bdd018eaa9af1` and `POST /cart`. The Output panel on the right shows the command `node` and `powershell`.

> PUT/cart/ : Update the quantity of a product in the cart.

The screenshot shows the VS Code interface with a REST client request to `http://localhost:5100/cart/67841434162bdd018eaa9af1` using the PUT method. The request body is a JSON object: `{ "quantity": "3" }`. The response status is **200 OK**, with a size of 229 Bytes and a time of 160 ms. The response body is: `{ "_id": "6784178a162bdd018eaa9b02", "productid": "67841434162bdd018eaa9af1", "title": "Powder Canister", "price": 14.99, "quantity": 3, "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Powder%20Canister/thumbnail.png", "__v": 0 }`. The Explorer panel on the left shows the project structure for 'SHOPPYGLOBE', including files like `cartitems.c...`, `products.control...`, `users.controller.js`, `Middleware`, `tokenacces...`, `Model`, `cart.model.js`, `products...`, `users.mod...`, `node_modules`, `Routes`, `shoppy.rou...`, `package-lock.json`, `package.json`, `README.md`, and `server.js`. The Terminal panel at the bottom shows the command `Server Running on Port 5100`, `Connection to Database Successful`, and `PUT /cart/67841434162bdd018eaa9af1`. The Output panel on the right shows the command `node` and `powershell`.

> DELETE/cart/ : Remove a product from the cart.

The screenshot shows the VS Code interface with the REST Client extension. The Explorer panel on the left shows the project structure for 'SHOPPYGLOBE'. The main editor area displays a REST Client request for a DELETE method to the URL 'http://localhost:5100/products/67841434162bdd018eaa9af1'. The 'Query' tab is active, showing 'Query Parameters' with a table for parameters. The 'Response' tab shows a successful response with a status of 200 OK, size of 229 Bytes, and time of 164 ms. The response body is a JSON object:

```
1 {
2   "_id": "6784178a162bdd018eaa9b02",
3   "productid": "67841434162bdd018eaa9af1",
4   "title": "Powder Canister",
5   "price": 14.99,
6   "quantity": 3,
7   "thumbnail": "https://cdn.dummyjson.com/products/images/beauty/Powder%20Canister/thumbnail.png",
8   "__v": 0
9 }
```

The Terminal panel at the bottom shows the server running on port 5100 and the connection to the database being successful. The command 'DELETE /cart/67841434162bdd018eaa9af1' is entered in the terminal.

ERROR HANDLING

The screenshot shows the VS Code interface with the REST Client extension. The Explorer panel on the left shows the project structure for 'SHOPPYGLOBE'. The main editor area displays a REST Client request for a GET method to the URL 'http://localhost:5100/products/67841434162bdd018eaa9af2'. The 'Body' tab is active, showing 'JSON Content' with a table for the JSON body. The 'Response' tab shows a 404 Not Found status with a size of 34 Bytes and time of 17 ms. The response body is a JSON object:

```
1 {
2   "message": "PRODUCT ID NOT FOUND"
3 }
```

The Terminal panel at the bottom shows the server running on port 5100 and the connection to the database being successful. The command 'GET /products/67841434162bdd018eaa9af1' is entered in the terminal.

> INVALID TOKEN

The image displays two screenshots of the Visual Studio Code interface, specifically the REST Client extension, used for testing API endpoints. Both screenshots show a project named 'shoppyglobe' with a file explorer on the left containing files like 'Controller', 'cartitems.c...', 'products.control...', 'users.controller.js', 'Middleware', 'tokenacces...', 'Model', 'cart.model.js', 'products...', 'users.mod...', 'node_modules', 'Routes', 'shoppy.rout...', 'package-lock.json', 'package.json', 'README.md', and 'server.js'.

Top Screenshot: A POST request to 'http://localhost:5100/cart' is shown. The 'Headers' tab is active, displaying the following headers: 'Accept' (application/*), 'User-Agent' (Thunder Client (https://www.thunderclient.com)), and 'Authorization' (JWT yJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI:). The 'Response' tab shows a 400 Bad Request status with a 27-byte response: { "message": "invalid token" }. The terminal at the bottom shows the command 'node' being executed.

Bottom Screenshot: A PUT request to 'http://localhost:5100/cart/67841434162bdd018eaa9af1' is shown. The 'Headers' tab is active, displaying the same headers as the top screenshot. The 'Response' tab shows a 400 Bad Request status with a 27-byte response: { "message": "invalid token" }. The terminal at the bottom shows the command 'node' being executed.

> ALREADY REGISTERED ACCOUNT

The screenshot shows a VS Code editor with a REST client interface. The request is a POST to `http://localhost:5100/register`. The request body is a JSON object:

```
1 {
2   "fullname": "siddhesh Dandekar",
3   "email": "siddhesh@gmail.com",
4   "password": "siddhesh"
5 }
```

The response is a 409 Conflict with a message: "ALREADY REGISTERED EMAIL FOUND".

The terminal shows the following output:

```
PUT
/cart/67841434162bdd018eaa9af1
POST
/register
[ ]
```


> WRONG CREDENTIAL

The image displays two screenshots of the Visual Studio Code interface, specifically the REST Client extension, showing a REST client request and its response for a login endpoint.

Top Screenshot:

- Request:** A POST request to `http://localhost:5100/login`. The request body is a JSON object:

```
{  "email": "siddhesh@gmail.com",  "password": "xyz"}
```
- Response:** The response status is **401 Unauthorized**. The response body is a JSON object:

```
{  "message": "Wrong Password"}
```

Bottom Screenshot:

- Request:** A POST request to `http://localhost:5100/login`. The request body is a JSON object:

```
{  "email": "siddhesh@gmail.com",  "password": "xyz"}
```
- Response:** The response status is **404 Not Found**. The response body is a JSON object:

```
{  "message": "User not registered"}
```

Both screenshots show the REST Client extension in VS Code, with the request and response details visible in the main editor area. The status bar at the bottom indicates the current file is `server.js` in the `shoppyglobe` project.