

app.get("/products",fetchproducts);

The screenshot shows the VS Code editor with the Thunder Client extension. The interface is divided into several panels:

- Left Panel (Activity):** Shows a list of requests. The first request is a GET request to `localhost:8086/` labeled "just now".
- Top Panel (Request Editor):** Shows the details of the selected request. The method is GET, and the URL is `http://localhost:8086/products`. The "Send" button is visible.
- Bottom Panel (Response):** Shows the response of the request. The status is 200 OK, the size is 8.49 KB, and the time is 170 ms. The response body is a JSON array of two product objects.

The response body is as follows:

```
[
  {
    "_id": "69525067e37dbba64e21f120",
    "title": "Essence Mascara Lash Princess",
    "price": 9.99,
    "image": "https://cdn.dummyjson.com/product-images/1/1.jpg",
    "discount": 7.17,
    "brand": "Essence",
    "returnPolicy": "",
    "warrantyInformation": "",
    "description": "Volumizing and lengthening mascara",
    "stock": 5,
    "minimumOrderQuantity": 24,
    "category": "beauty",
    "rating": "4.94",
    "discountPercentage": 7.17,
    "__v": 0
  },
  {
    "_id": "69525067e37dbba64e21f121",
    "title": "iPhone X",
    "price": 899,
    "image": "https://cdn.dummyjson.com/product-images/2/1.jpg",
    "discount": 17.04
  }
]
```

The bottom status bar shows the file explorer with `main*` and the output window with the message "Inline suggestions quota reached". The system tray at the bottom shows the time as 10:14 AM on 12/30/2025.

app.get("/products/:id",fetchproductbyid);

The screenshot displays the Thunder Client application interface. The top bar shows the file explorer with open files: `Shoppyglobe.model.js`, `Cart.model.js`, `shoppypyRoutes.js`, `localhost:8086/`, `User.model.js`, and `shoppyglobe.controller.js`.

The left sidebar, titled "THUNDER CLIENT", contains a "New Request" button and a list of recent requests. The "Activity" tab is selected, showing a list of requests with their methods, URLs, and timestamps.

The main workspace is divided into three panels:

- Request Panel:** Shows a GET request to `http://localhost:8086/products/69525067e37dbba64e21f12`. The "Send" button is visible.
- Query Panel:** Displays "Query Parameters" with a table for parameters and values.
- Response Panel:** Shows the response status as "200 OK", size as "369 Bytes", and time as "108 ms". The response body is a JSON object representing a product.

The response JSON is as follows:

```
1 {
2   "_id": "69525067e37dbba64e21f122",
3   "title": "Samsung Universe 9",
4   "price": 1249,
5   "image": "https://cdn.dummyjson.com/product-images/3/1.jpg",
6   "discount": 15.46,
7   "brand": "Samsung",
8   "returnPolicy": "",
9   "warrantyInformation": "",
10  "description": "Premium Samsung smartphone.",
11  "stock": 36,
12  "minimumOrderQuantity": 1,
13  "category": "smartphones",
14  "rating": "4.75",
15  "discountPercentage": 15.46,
16  "__v": 0
17 }
```

The bottom status bar shows the current project is "main*", there are 0 errors and 0 warnings, and a notification for "Inline suggestions quota reached". The system tray at the bottom indicates the time is 10:16 AM on 12/30/2025, with weather information for "Afternoon rain".

app.post("/register",verifyregisteruser,userRegister);

The screenshot displays the Thunder Client interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, and a search bar. The left sidebar shows the 'THUNDER CLIENT' with tabs for 'New Request', 'Activity', 'Collections', and 'Env'. The 'Activity' tab is active, showing a list of requests. The selected request is a POST to 'localhost:8086/' labeled 'just now'. The main panel shows the details of this request: the method is POST, the URL is 'http://localhost:8086/register', and the status is '201 Created' with a size of '42 Bytes' and time of '449 ms'. The 'Body' tab is selected, showing a JSON payload with user registration details. The 'Response' tab shows the server's response, which is a JSON object with a success message. The bottom status bar indicates 'main*' and '0 0 0' errors.

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8086/ just now

POST localhost:8081/ 22 hours ago

PUT localhost:5081/ 4 days ago

GET localhost:8080 5 days ago

POST localhost:8888 1 month ago

POST http://localhost:8086/register Send

Status: 201 Created Size: 42 Bytes Time: 449 ms

Response Headers 6 Cookies Results Docs

```
1 {
2   "message": "User registered successfully"
3 }
```

Copy

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "firstName": "MOHD",
3   "middleName": "YASIR",
4   "lastName": "taploo",
5   "email": "yasir@gmail.com",
6   "mobileNumber": "7006588550",
7   "password": "Yasir@345"
8 }
```

main* 0 0 0

Type here to search

9°C Mostly cloudy 10:24 AM 12/30/2025

if we try to register the user from same email

The screenshot shows the Thunder Client interface with a dark theme. The left sidebar displays a list of requests, with the most recent one being a POST request to `localhost:8086/` made 1 minute ago. The main panel shows the details of this request, which is a POST to `http://localhost:8086/register`. The request body is in JSON format, containing the following data:

```
1 {
2   "firstName": "MOHD",
3   "middleName": "YASIR",
4   "lastName": "taploo",
5   "email": "yasir@gmail.com",
6   "mobileNumber": "7006588550",
7   "password": "Yasir@345"
8 }
```

The response status is **409 Conflict**, with a size of 34 Bytes and a time of 118 ms. The response body is:

```
1 {
2   "message": "email already exists"
3 }
```

The Windows taskbar at the bottom shows the system clock as 10:25 AM on 12/30/2025, with a weather forecast of 9°C and 'Mostly cloudy'.

app.post("/login",userLogin);

The screenshot displays the Thunder Client interface with a dark theme. The top menu bar includes File, Edit, Selection, View, Go, Run, and a search bar containing 'backendshoppyglobe'. The left sidebar, titled 'THUNDER CLIENT', features a 'New Request' button and tabs for 'Activity', 'Collections', and 'Env'. The 'Activity' tab is active, showing a list of requests. The selected request is a POST to 'localhost:8086/' labeled 'just now'. The main panel shows the details of this request: the method is POST, the URL is 'http://localhost:8086/login', and the 'Body' tab is selected. The body content is in JSON format, showing an email and password. The right panel displays the response status as '200 OK' with a size of '122 Bytes' and a time of '248 ms'. The response body is also in JSON format, containing a 'message' and a 'token'. A 'Copy' button is visible next to the response body.

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8086/ just now

POST localhost:8081/ 22 hours ago

PUT localhost:5081/ 4 days ago

GET localhost:8080 5 days ago

POST localhost:8888 1 month ago

shoppyRoutes.js localhost:8086/ localhost:8081/ localhost:8080 localhost:8888 User.model.js shoppyc

POST http://localhost:8086/login Send

Query Headers 3 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "email": "yasin@gmail.com",
3   "password": "Yasin@345"
4 }
```

Status: 200 OK Size: 122 Bytes Time: 248 ms

Response Headers 6 Cookies Results Docs {}

```
1 {
2   "message": "welcome user",
3   "token": "eyJhbGciOiJIUzI1NiJ9.eyJmZaXJAZ21haWwY29t
4   .pUgwh7cn0nJjQ0LnQpOWg9G2pFRkK9z3U1_LhPoybFA"
```

Copy

app.post("/login",userLogin);

The screenshot displays the Thunder Client interface with a dark theme. The top toolbar includes menus for File, Edit, Selection, View, Go, and Run. The main window is divided into three panes. The left pane, titled 'THUNDER CLIENT', shows a list of requests: a POST to localhost:8086/ (just now), a POST to localhost:8081/ (22 hours ago), a PUT to localhost:5081/ (4 days ago), a GET to localhost:8080 (5 days ago), and a POST to localhost:8888 (1 month ago). The middle pane shows the details of the selected POST request to http://localhost:8086/login. The 'Body' tab is active, displaying JSON content:

```
{  "email": "yasir@gmail.com",  "password": "Yasir@345"}
```

. The right pane shows the response with a status of 200 OK, size of 122 Bytes, and time of 248 ms. The response body is:

```
{  "message": "welcome user",  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQpInQ9.eWFZaXJAZ21haWwY29t.pUgwh7cn0nJjQ0LnQpOWg9G2pFRkK9z3U1_LhPoybFA"}
```

. The Windows taskbar at the bottom shows the search bar, several application icons, and system information including 9°C, Mostly cloudy, and the date 12/30/2025.

File Edit Selection View Go Run ...

backendshoppyglobe

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8086/ just now

POST localhost:8081/ 22 hours ago

PUT localhost:5081/ 4 days ago

GET localhost:8080 5 days ago

POST localhost:8888 1 month ago

POST http://localhost:8086/login Send

Query Headers³ Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "email": "yasir@gmail.com",
3   "password": "Yasir@345"
4 }
```

Status: 200 OK Size: 122 Bytes Time: 248 ms

Response Headers⁶ Cookies Results Docs

```
1 {
2   "message": "welcome user",
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXZWQpInQ9.eWFZaXJAZ21haWwY29t.pUgwh7cn0nJjQ0LnQpOWg9G2pFRkK9z3U1_LhPoybFA"
4 }
```

Copy

main* 0 0

Type here to search

9°C Mostly cloudy 10:28 AM 12/30/2025

if i try wrong email

The screenshot shows the Thunder Client interface with a REST client tab open for `localhost:8086/login`. The request is a POST with a JSON body containing email and password. The response is a 404 Not Found status with a message indicating an invalid email.

Thunder Client Interface:

- Activity Panel:** Shows a list of requests. The most recent is a POST to `localhost:8086/` labeled "just now".
- Request Editor:**
 - Method:** POST
 - URL:** `http://localhost:8086/login`
 - Body:** JSON format with the following content:

```
1 {
2   "email": "xyz@gmail.com",
3   "password": "Yasin@345"
4 }
```
- Response Panel:**
 - Status:** 404 Not Found
 - Size:** 27 Bytes
 - Time:** 99 ms
 - Response Body:**

```
1 {
2   "message": "Invalid email"
3 }
```

System Tray: Shows the time as 10:29 AM on 12/30/2025, with weather information (9°C, Mostly cloudy) and system icons.

if i try wrong password

The screenshot shows the Thunder Client interface with a REST client request to `http://localhost:8086/login` via POST. The request body is a JSON object: `{ "email": "yasir@gmail.com", "password": "Yas5" }`. The response status is **401 Unauthorized** with a size of 30 Bytes and a time of 255 ms. The response body is a JSON object: `{ "message": "invalid password" }`.

Thunder Client Interface:

- Activity Panel (Left):** Lists recent requests. The top request is a POST to `localhost:8086/` labeled "just now".
- Request Editor (Center):**
 - Method:** POST
 - URL:** `http://localhost:8086/login`
 - Body:** JSON format with `"email": "yasir@gmail.com", "password": "Yas5"`
- Response Panel (Right):**
 - Status:** 401 Unauthorized
 - Size:** 30 Bytes
 - Time:** 255 ms
 - Response Body:** `{ "message": "invalid password" }`

System Taskbar (Bottom):

- Search bar: "Type here to search"
- Taskbar icons: File Explorer, Microsoft Edge, Visual Studio Code, etc.
- System tray: Weather (9°C Mostly cloudy), Date/Time (10:30 AM 12/30/2025), Language (ENG).

after login i can add products

The screenshot shows the VS Code interface with the Thunder Client extension. The main editor displays a REST client request and its response.

Request:

- Method: POST
- URL: `http://localhost:8086/addproducts`
- Body (JSON):

```
{  "title": "Samsung Galaxy S23",  "price": 74999,  "image": "https://example.com/images/galaxy-s23.jpg",  "discount": 5000,  "brand": "Samsung",  "returnPolicy": "7 days replacement only",  "warrantyInformation": "1 year manufacturer warranty",  "description": "Samsung Galaxy S23 with 8GB RAM and 128GB storage.",  "stock": 50,  "minimumOrderQuantity": 1,  "category": "Electronics",  "rating": "4.5",  "discountPercentage": 6.6}
```

Response:

- Status: 201 Created
- Size: 498 Bytes
- Time: 130 ms
- Response Body (JSON):

```
{  "success": true,  "message": "Product added successfully",  "data": {    "title": "Samsung Galaxy S23",    "price": 74999,    "image": "https://example.com/images/galaxy-s23.jpg",    "discount": 5000,    "brand": "Samsung",    "returnPolicy": "7 days replacement only",    "warrantyInformation": "1 year manufacturer warranty",    "description": "Samsung Galaxy S23 with 8GB RAM and 128GB storage.",    "stock": 50,    "minimumOrderQuantity": 1,    "category": "Electronics",    "rating": "4.5",    "discountPercentage": 6.6,    "_id": "69535d031440bb0bfd54f859",    "__v": 0  }  }
```

The bottom status bar shows the current file is `main*` and the language is `ENG`. The system tray at the bottom indicates the temperature is 9°C, mostly cloudy, and the time is 10:33 AM on 12/30/2025.

after login i can delete products

The screenshot displays the Thunder Client interface with a dark theme. The top toolbar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a search bar containing 'backendshoppyglobe'. The left sidebar, titled 'THUNDER CLIENT', shows a list of requests under the 'Activity' tab. The selected request is a DELETE method to 'localhost:8086/' labeled 'just now'. The main panel shows the details of this request: the method is 'DELETE', the URL is 'http://localhost:8086/products', and the 'Body' tab is active, displaying a JSON object with an 'id' field. The right panel shows the response: 'Status: 200 OK', 'Size: 42 Bytes', and 'Time: 116 ms'. The response body is a JSON object with a 'message' field stating 'product deleted successfully'. The bottom status bar shows 'main*' and a search bar. The Windows taskbar at the very bottom includes icons for various applications and system information like '9°C Cloudy' and '10:37 AM 12/30/2025'.

File Edit Selection View Go Run ...

backendsoppyglobe

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

DEL localhost:8086/ just now

POST localhost:8081/ 22 hours ago

PUT localhost:5081/ 4 days ago

GET localhost/8080 5 days ago

POST localhost8888 1 month ago

DELETE http://localhost:8086/products Send

Query Headers³ Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "id": "69535e010d5e1dd867227511"
3 }
```

Status: 200 OK Size: 42 Bytes Time: 116 ms

Response Headers⁶ Cookies Results Docs

```
1 {
2   "message": "product deleted successfully"
3 }
```

Copy

main* 0 0 0

Type here to search

9°C Cloudy 10:37 AM 12/30/2025

after login i can access cart

The screenshot shows the Thunder Client interface with a REST client request configured. The request is a GET to `http://localhost:8086/cart`. The body is a JSON object with an `id` field. The response is a 404 Not Found status with a message "No items in cart".

Thunder Client Interface:

- Activity Panel (Left):** Shows a list of requests. The selected request is a GET to `localhost:8086/` labeled "just now".
- Request Editor (Middle):**
 - Method:** GET
 - URL:** `http://localhost:8086/cart`
 - Body:** JSON format with content:

```
{  "id": "69535e010d5e1dd867227511"}
```
- Response Panel (Right):**
 - Status:** 404 Not Found
 - Size:** 30 Bytes
 - Time:** 122 ms
 - Response Body:**

```
{  "message": "No items in cart"}
```

System Tray (Bottom): Includes a search bar, taskbar icons (Thunder Client, Dev C++, Chrome, VS Code), and system status (Rain today, 10:40 AM, 12/30/2025).

after login i can add an item to the cart

The screenshot displays the Visual Studio Code interface with a REST client extension. The main editor shows a POST request to `http://localhost:8086/cart` with a JSON body:

```
{
  "productid": "69536238564e8ad77f388834",
  "productQuantity": "20"
}
```

The response status is **201 Created** with a size of **151 Bytes** and a time of **231 ms**. The response body is:

```
{
  "cartitem": {
    "productid": "69536238564e8ad77f388834",
    "productQuantity": 20,
    "_id": "695364846af4e6de6e7187c2",
    "__v": 0
  },
  "message": "item added successfully"
}
```

On the right, the console shows a `SyntaxError: missing) after argument list` and a message from nodemon: `[nodemon] app crashed - waiting for file changes before starting...`. The status bar at the bottom indicates `main*` and `0 errors, 0 warnings`.

if i try to add the same item again in cart

THUNDER CLIENT

New Request

Activity Collections Env

filter activity

POST localhost:8086/
9 mins ago

POST localhost:8081/
22 hours ago

PUT localhost:5081/
4 days ago

GET localhost:8080
5 days ago

POST localhost:8888
1 month ago

localhost:8086/ X

localhost:8081/

localhost:8080

localhost:8888

User.model.js

shop

POST http://localhost:8086/cart Send

Query Headers Auth Body Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

1 {
2 "productid":"69535f860d5e1dd867227515",
3 "productQuantity":"20"
4 }
5

Status: 400 Bad Request Size: 37 Bytes Time: 67 ms

Response Headers Cookies Results Docs

1 {
2 "message": "Product already in cart"
3 }

Response Chart

node + -

js`
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Server is running on port 8086
Connected to MongoDB
yasin@gmail.com
i amjjjj
[nodemon] restarting due to changes...
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Server is running on port 8086
Connected to MongoDB
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Server is running on port 8086
Connected to MongoDB
yasin@gmail.com

update quantity of cart item by id

The screenshot displays the Thunder Client interface with a PUT request configured to update a cart item's quantity. The request is sent to `http://localhost:8086/cart/69536238564e8ad77f388834` with a JSON body containing the item ID and the new quantity of 80. The response is a 200 OK status with a message "item updated".

Thunder Client Interface:

- Activity Panel (Left):** Lists recent requests. The top entry is a PUT request to `localhost:8086/` labeled "just now".
- Request Editor (Center):**
 - Method:** PUT
 - URL:** `http://localhost:8086/cart/69536238564e8ad77f388834`
 - Body:** JSON format with content:

```
{
  "id": "69536238564e8ad77f388834",
  "quantity": "80"
}
```
 - Status:** 200 OK, Size: 26 Bytes, Time: 184 ms
 - Response:**

```
{
  "message": "item updated"
}
```
- Terminal (Right):** Shows logs from a Node.js application running on port 8086, including messages like "[nodemon] restarting due to changes..." and "Server is running on port 8086".

System Taskbar (Bottom): Includes the Windows search bar, taskbar icons for various applications, and system status information showing 10°C, Cloudy weather, and the time 11:40 AM on 12/30/2025.

if wrong id was sent then it will show the message accordingly

The image shows a screenshot of the Visual Studio Code editor. The main window displays a REST client request in the 'Body' tab. The request is a PUT method to the URL 'http://localhost:8086/cart'. The request body is a JSON object:

```
{  "id": "69536238564e8ad77f388824",  "quantity": "90"}
```

. The response is a 404 Not Found status with a message:

```
{  "message": "Cart item not found"}
```

. The left sidebar shows a list of requests, with the current one being a PUT request to localhost:8086/. The right sidebar shows a terminal window with the output of the request, including the status '404 Not Found' and the message 'Cart item not found'. The bottom status bar shows the file 'main*' and the text 'Type here to search'.

Data Explorer

- My Queries
- Data Modeling

CLUSTERS (1)

Search clusters

- Cluster0
 - admin
 - local
 - test
 - carts
 - products
 - users

Cluster0 > test

[View monitoring](#) [Visualize your data](#) [+ Create collection](#) [Refresh](#)

Collection name	Properties	Storage size	Documents	Avg. document size	Indexes	Total index size
carts	-	32.77 kB	0	0 B	1	32.77 kB
products	-	36.86 kB	27	368.00 B	1	36.86 kB
users	-	36.86 kB	1	225.00 B	1	36.86 kB

three collections created

ORGANIZATION Yasir's Org - 2025-11-19 PROJECT shoppyglobetwo

Data Explorer

- My Queries
- Data Modeling

CLUSTERS (1)

Search clusters

- Cluster0
 - admin
 - local
 - test
 - carts**
 - products
 - users

Cluster0 > test > carts

Documents 0 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Practice creating, reading, updating, and deleting documents efficiently. [GO TO SKILLS](#)

[ADD DATA](#) [UPDATE](#) [DELETE](#)

25 1-1 of 1

```
{
  "_id": ObjectId('69537dee4bc9f72491ed84b6'),
  "productid": ObjectId('69536238564e8ad77fd38894'),
  "productQuantity": 1,
  "__v": 0
}
```

cart collection

System Status: All Good

©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

IST Internshala Trainings

Project: Build APIs with Node

Cluster0 Data | Cloud: MongoDB

New Tab

cloud.mongodb.com/v2/69524271e41bf67355c9b5bf#/explorer/6952428f45800142cc175000/test/products/find

ORGANIZATION
Yasir's Org - 2025-11-19

PROJECT
shoppyglobetwo

?

Y

Data Explorer

My Queries

Data Modeling

CLUSTERS (1)

Search clusters

Cluster0

- admin
- local
- test
 - carts
 - products
 - users

Cluster0 > test > products

Documents 27 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

Practice creating, reading, updating, and deleting documents efficiently. [GO TO SKILLS](#)

ADD DATA UPDATE DELETE

25 1 - 25 of 27

product collection

View monitoring

Visualize Your Data

Explain

Reset

Find

Options

```
{
  "_id": ObjectId('69525067e37dbba64e21f128'),
  "title": "Essence Mascara Lash Princess",
  "price": 9.99,
  "image": "https://cdn.dummyjson.com/product-images/1/1.jpg",
  "discount": 7.17,
  "brand": "Essence",
  "returnPolicy": "",
  "warrantyInformation": "",
  "description": "Volumizing and lengthening mascara.",
  "stock": 5,
  "minimumOrderQuantity": 24,
  "category": "beauty",
  "rating": "4.94",
  "discountPercentage": 7.17,
  "__v": 0
}
```

System Status: All Good

©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Type here to search

10°C Cloudy 12:54 PM 12/30/2025

