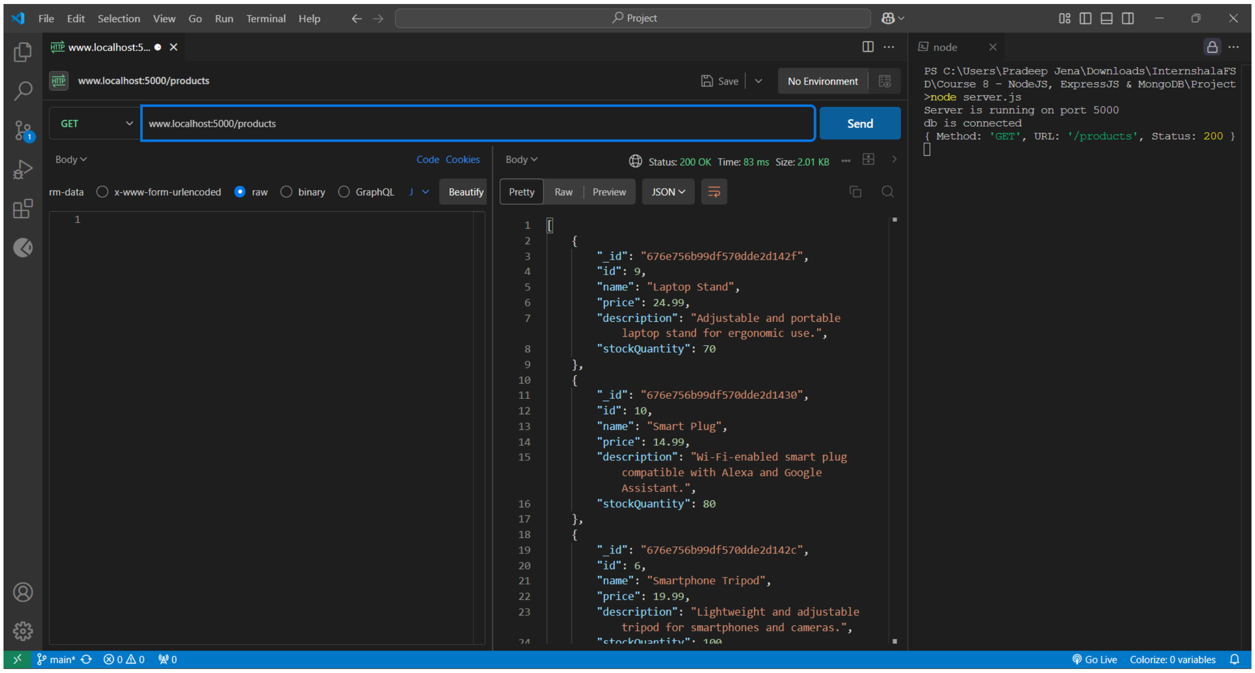
Routes:

1) **Method**: 'GET' & **URL**: '/products' & **Status**: 200 OK



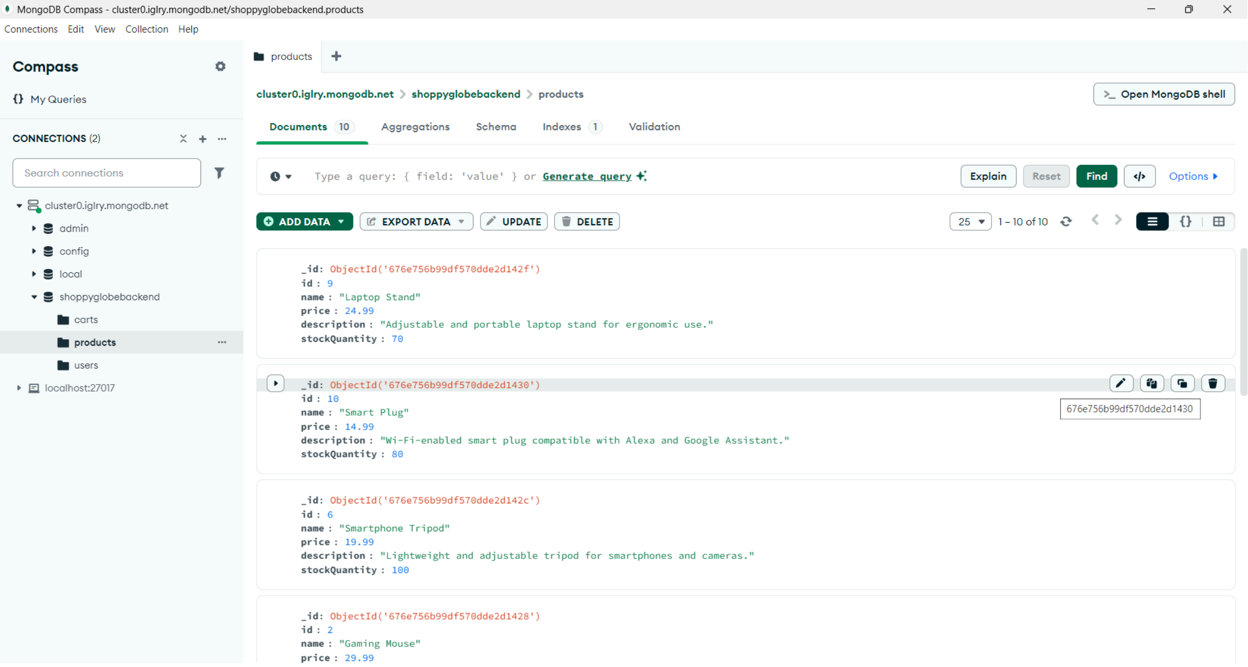
www.localhost:5000/products

GET www.localhost:5000/products

Status: 200 OK Time: 83 ms Size: 2.01 KB

```
[
    {
        "_id": "676e756b99df570dde2d142f",
        "id": 9,
        "name": "Laptop Stand",
        "price": 24.99,
        "description": "Adjustable and portable
          laptop stand for ergonomic use.",
        "stockQuantity": 70
    },
    {
        "_id": "676e756b99df570dde2d1430",
        "id": 10,
        "name": "Smart Plug",
        "price": 14.99,
        "description": "Wi-Fi-enabled smart plug
          compatible with Alexa and Google
          Assistant.",
        "stockQuantity": 80
    },
    {
        "_id": "676e756b99df570dde2d142c",
        "id": 6,
        "name": "Smartphone Tripod",
        "price": 19.99,
        "description": "Lightweight and adjustable
          tripod for smartphones and cameras.",
        "stockQuantity": 100
```

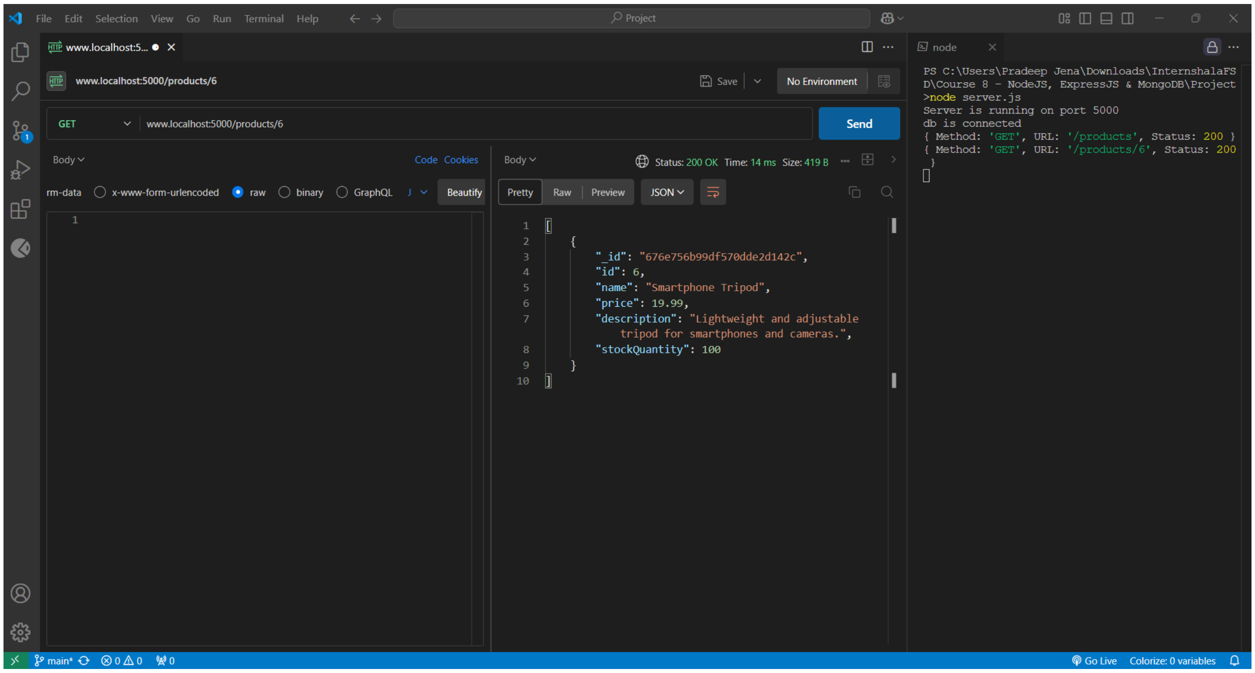
```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
```



```
_id: ObjectId('676e756b99df570dde2d142f')
id : 9
name : "Laptop Stand"
price : 24.99
description : "Adjustable and portable laptop stand for ergonomic use."
stockQuantity : 70

_id: ObjectId('676e756b99df570dde2d1430')
id : 10
name : "Smart Plug"
price : 14.99
description : "Wi-Fi-enabled smart plug compatible with Alexa and Google Assistant."
stockQuantity : 80

_id: ObjectId('676e756b99df570dde2d142c')
id : 6
name : "Smartphone Tripod"
price : 19.99
description : "Lightweight and adjustable tripod for smartphones and cameras."
stockQuantity : 100

_id: ObjectId('676e756b99df570dde2d1428')
id : 2
name : "Gaming Mouse"
price : 29.99
```

676e756b99df570dde2d1430

2) **Method**: 'GET' & **URL**: '/products/6' & **Status**: 200 OK



www.localhost:5000/products/6

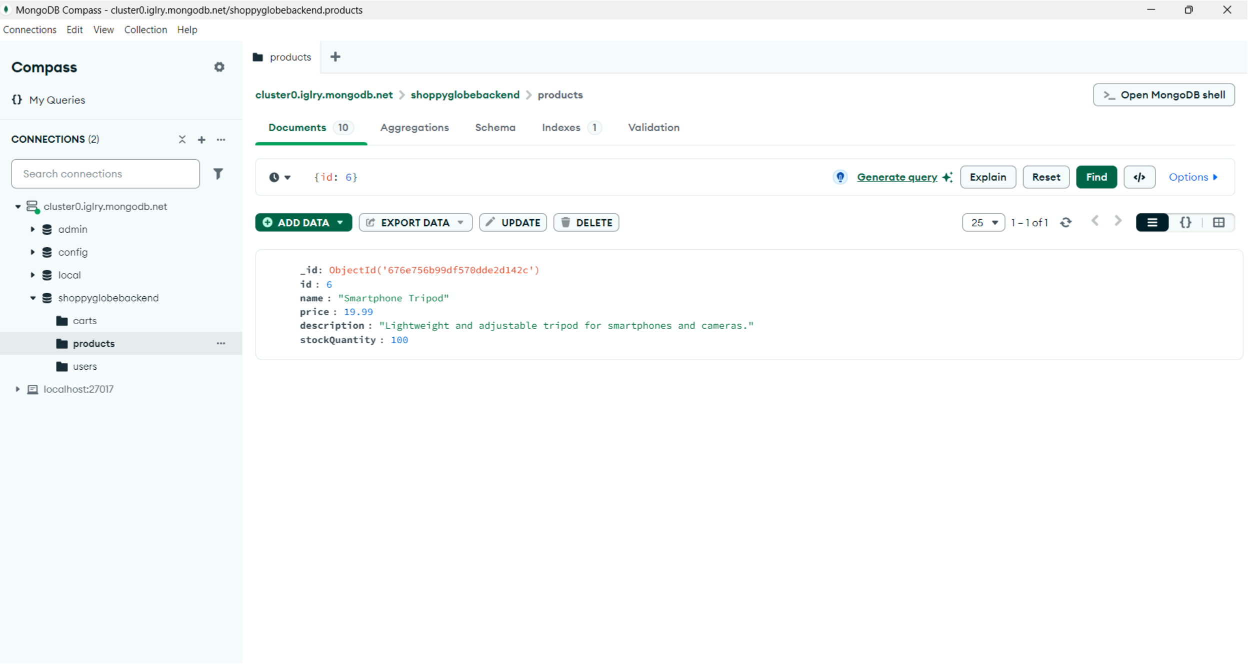
GET www.localhost:5000/products/6

○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

Status: 200 OK Time: 14 ms Size: 419 B

```
[
    {
        "_id": "676e756b99df570dde2d142c",
        "id": 6,
        "name": "Smartphone Tripod",
        "price": 19.99,
        "description": "Lightweight and adjustable
            tripod for smartphones and cameras.",
        "stockQuantity": 100
    }
]
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
```



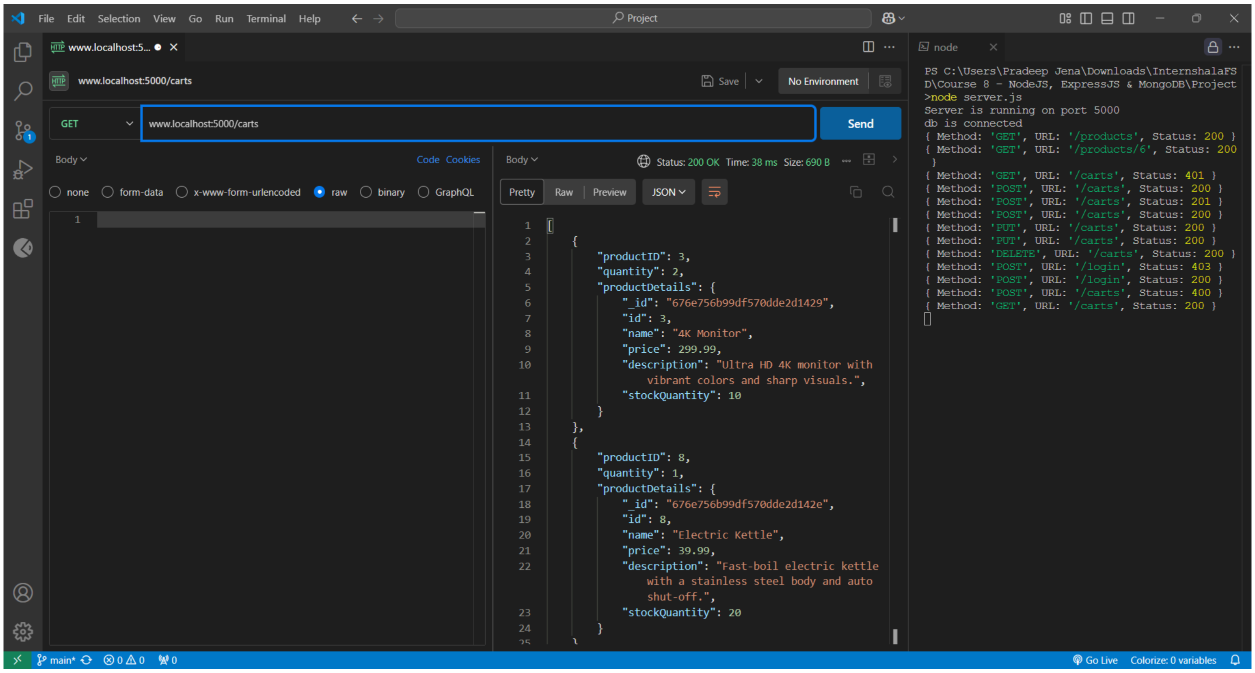
MongoDB Compass - cluster0.iglry.mongodb.net/shoppyglobebackend.products

cluster0.iglry.mongodb.net > shoppyglobebackend > products

```
{id: 6}
```

```
_id: ObjectId('676e756b99df570dde2d142c')
id : 6
name : "Smartphone Tripod"
price : 19.99
description : "Lightweight and adjustable tripod for smartphones and cameras."
stockQuantity : 100
```

3) **Method**: 'GET' & **URL**: '/carts' & **Status**: 200 OK



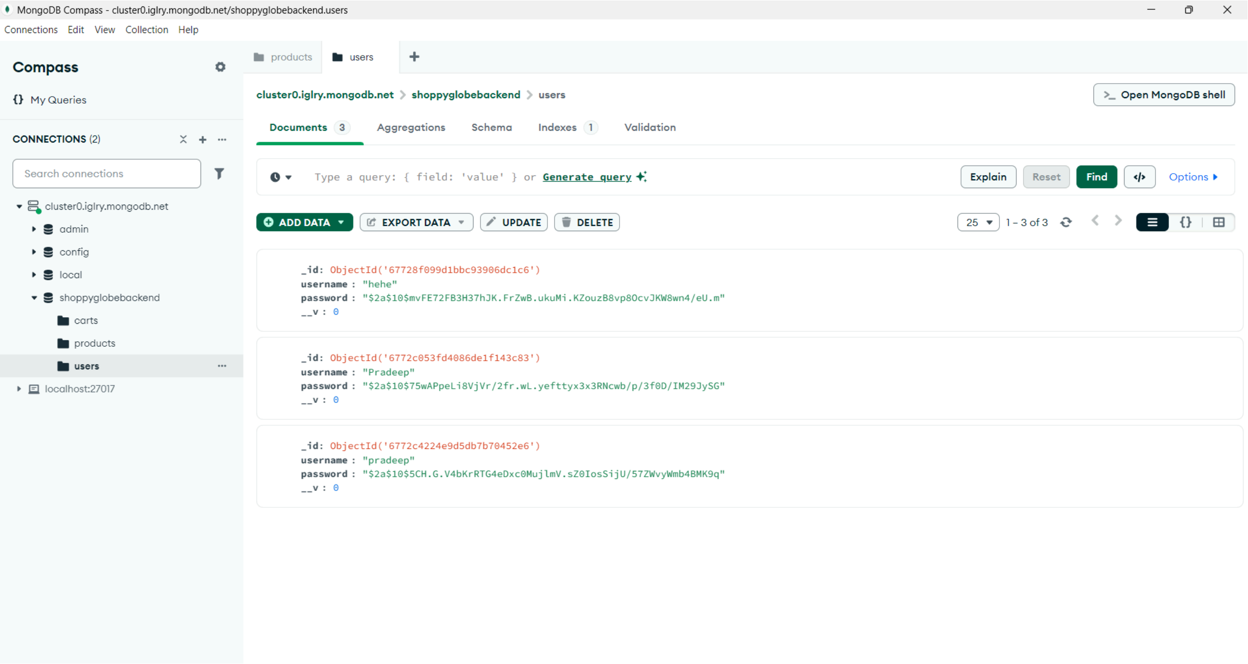
www.localhost:5000/carts

GET www.localhost:5000/carts

Status: 200 OK Time: 38 ms Size: 690 B

```
[
    {
        "productID": 3,
        "quantity": 2,
        "productDetails": {
            "_id": "676e756b99df570dde2d1429",
            "id": 3,
            "name": "4K Monitor",
            "price": 299.99,
            "description": "Ultra HD 4K monitor with
                vibrant colors and sharp visuals.",
            "stockQuantity": 10
        }
    },
    {
        "productID": 8,
        "quantity": 1,
        "productDetails": {
            "_id": "676e756b99df570dde2d142e",
            "id": 8,
            "name": "Electric Kettle",
            "price": 39.99,
            "description": "Fast-boil electric kettle
                with a stainless steel body and auto
                shut-off.",
            "stockQuantity": 20
        }
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
{ Method: 'GET', URL: '/carts', Status: 401 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'GET', URL: '/carts', Status: 200 }
```

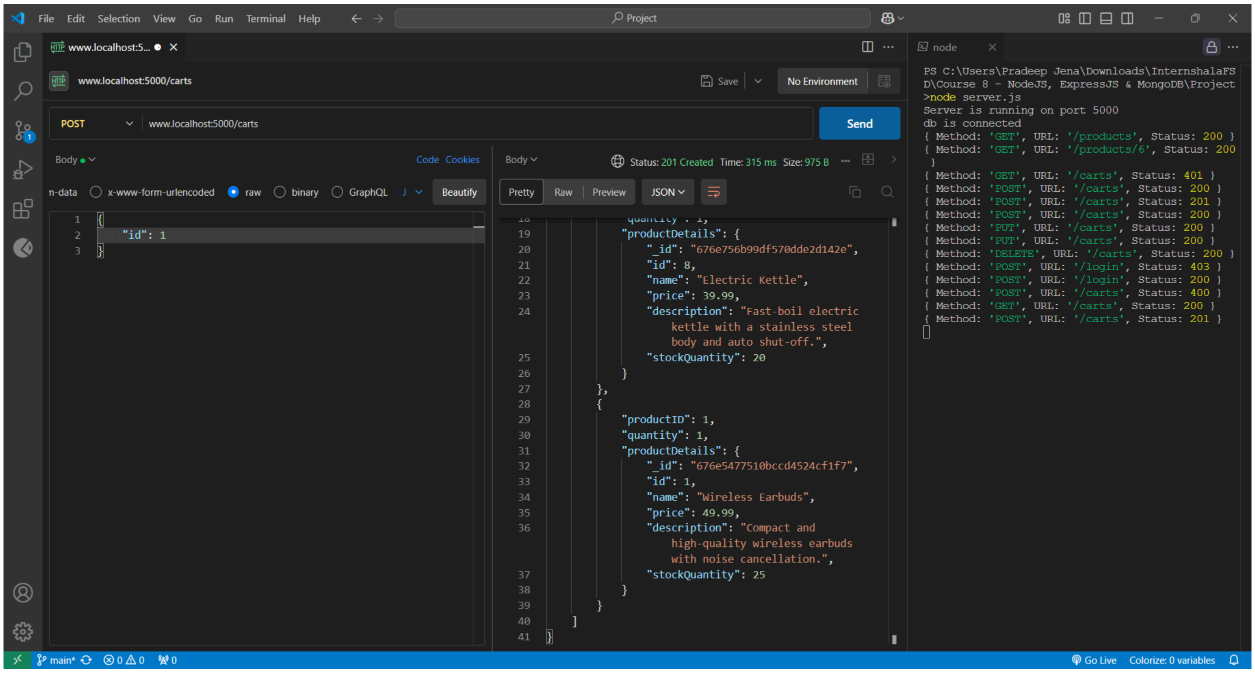


MongoDB Compass - cluster0.iglry.mongodb.net/shoppyglobebackend.users

cluster0.iglry.mongodb.net > shoppyglobebackend > users

```
_id: ObjectId('67728f099d1bbc93906dc1c6')
username : "hehe"
password : "$2a$10$mvFE72FB3H37hJK.FrZwB.ukuMi.KZouzB8vp8OcvJKW8wn4/eU.m"
__v : 0

_id: ObjectId('6772c053fd4086de1f143c83')
username : "Pradeep"
password : "$2a$10$75wAPpeLi8VjVr/2fr.wL.yefttyx3x3RNcwb/p/3f0D/IM29JySG"
__v : 0

_id: ObjectId('6772c4224e9d5db7b70452e6')
username : "pradeep"
password : "$2a$10$5CH.G.V4bKrRTG4eDxc0MujlmV.sZ0IosSijU/57ZWvyWmb4BMK9q"
__v : 0
```

4) **Method**: ‘POST’ & **URL**: '/carts' & **Status**: 201 Created



www.localhost:5000/carts

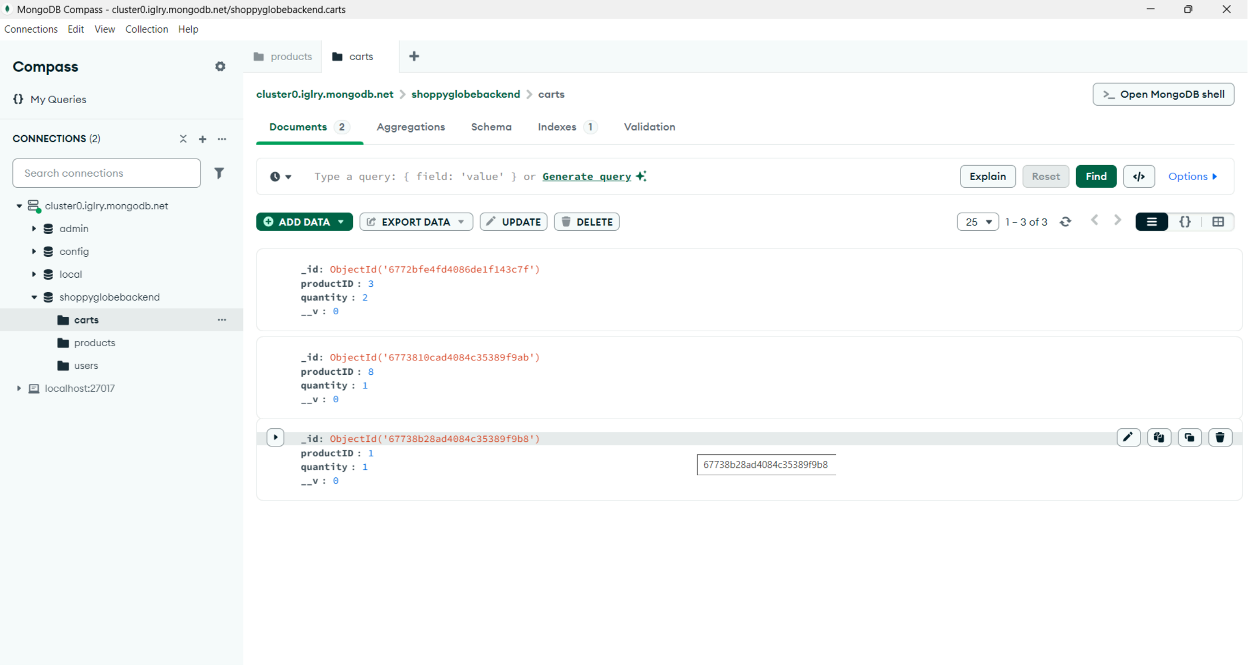
```
{
    "id": 1
}
```

Status: 201 Created Time: 315 ms Size: 975 B

```
                "quantity": 1,
                "productDetails": {
                    "_id": "676e756b99df570dde2d142e",
                    "id": 8,
                    "name": "Electric Kettle",
                    "price": 39.99,
                    "description": "Fast-boil electric
                      kettle with a stainless steel
                      body and auto shut-off.",
                    "stockQuantity": 20
                }
            },
            {
                "productID": 1,
                "quantity": 1,
                "productDetails": {
                    "_id": "676e5477510bccd4524cf1f7",
                    "id": 1,
                    "name": "Wireless Earbuds",
                    "price": 49.99,
                    "description": "Compact and
                      high-quality wireless earbuds
                      with noise cancellation.",
                    "stockQuantity": 25
                }
            }
        ]
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
{ Method: 'GET', URL: '/carts', Status: 401 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'GET', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
```

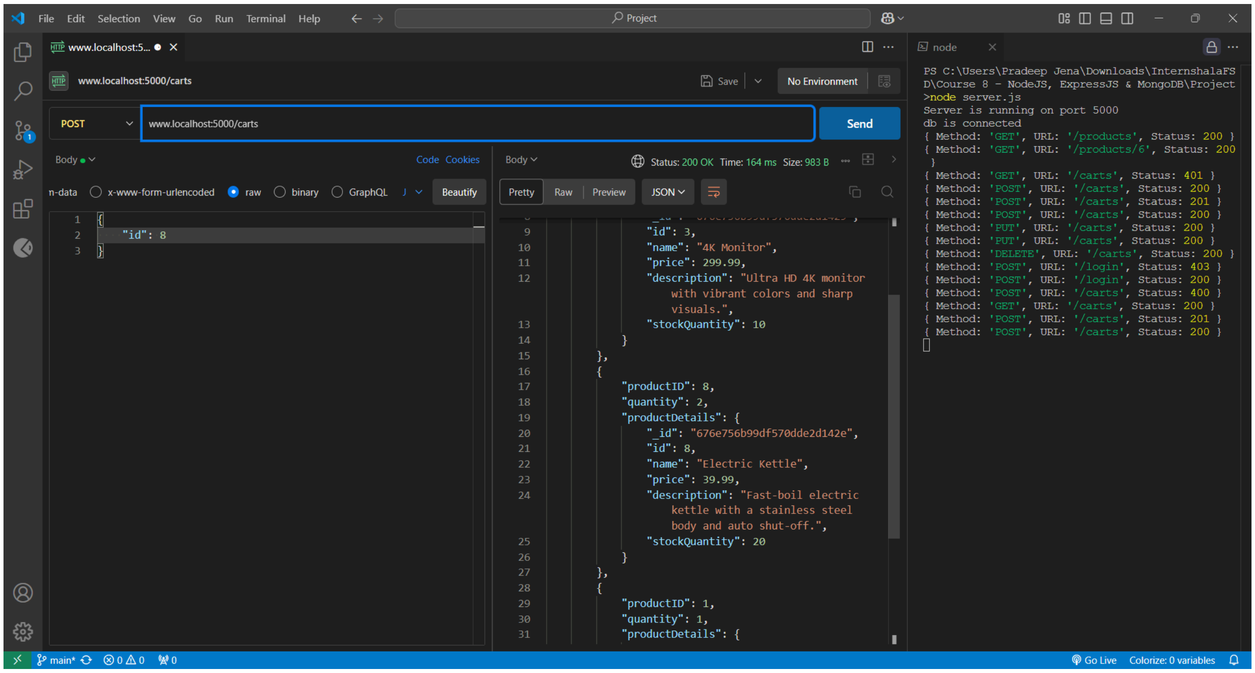
In case the product is not in the cart then it will add it to the cart.



```
_id: ObjectId('6772bfe4fd4086de1f143c7f')
productID : 3
quantity : 2
__v : 0

_id: ObjectId('6773810cad4084c35389f9ab')
productID : 8
quantity : 1
__v : 0

_id: ObjectId('67738b28ad4084c35389f9b8')
productID : 1
quantity : 1
__v : 0
```

67738b28ad4084c35389f9b8

5) **Method**: ‘POST’ & **URL**: '/carts' & **Status**: 200 OK



www.localhost:5000/carts

POST www.localhost:5000/carts

● raw

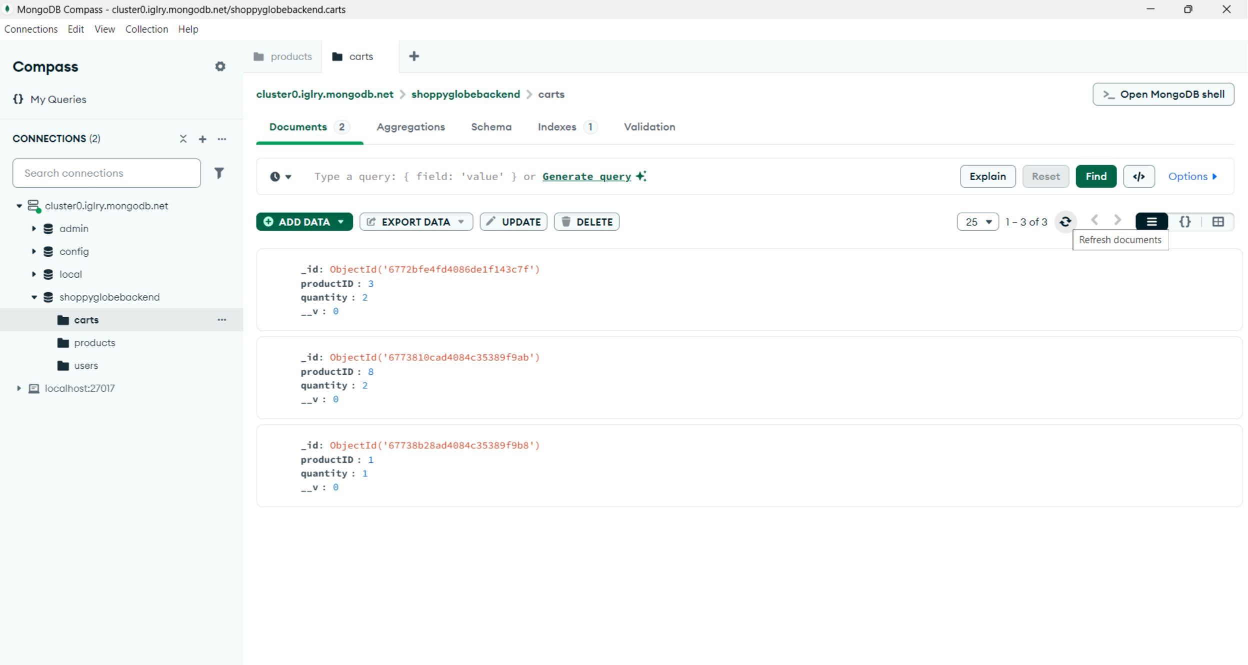
```
{
    "id": 8
}
```

Status: 200 OK Time: 164 ms Size: 983 B

```
                "id": 3,
                "name": "4K Monitor",
                "price": 299.99,
                "description": "Ultra HD 4K monitor
                    with vibrant colors and sharp
                    visuals.",
                "stockQuantity": 10
            }
        },
        {
            "productID": 8,
            "quantity": 2,
            "productDetails": {
                "_id": "676e756b99df570dde2d142e",
                "id": 8,
                "name": "Electric Kettle",
                "price": 39.99,
                "description": "Fast-boil electric
                    kettle with a stainless steel
                    body and auto shut-off.",
                "stockQuantity": 20
            }
        },
        {
            "productID": 1,
            "quantity": 1,
            "productDetails": {
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
{ Method: 'GET', URL: '/carts', Status: 401 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'GET', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
```

If the product is in the cart then, increase the quantity of the cart.

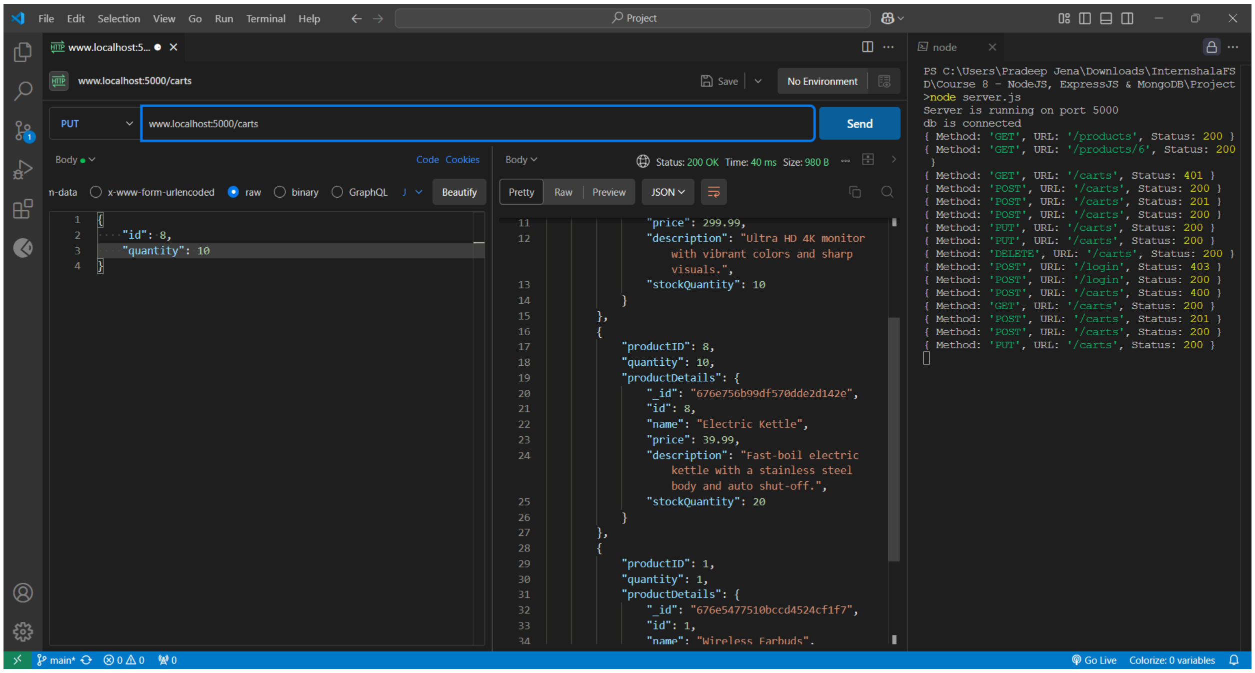


```
_id: ObjectId('6772bfe4fd4086de1f143c7f')
productID : 3
quantity : 2
__v : 0
```

```
_id: ObjectId('6773810cad4084c35389f9ab')
productID : 8
quantity : 2
__v : 0
```

```
_id: ObjectId('67738b28ad4084c35389f9b8')
productID : 1
quantity : 1
__v : 0
```

6) **Method**: ‘PUT’ & **URL**: '/carts' & **Status**: 200 OK



www.localhost:5000/carts

PUT www.localhost:5000/carts

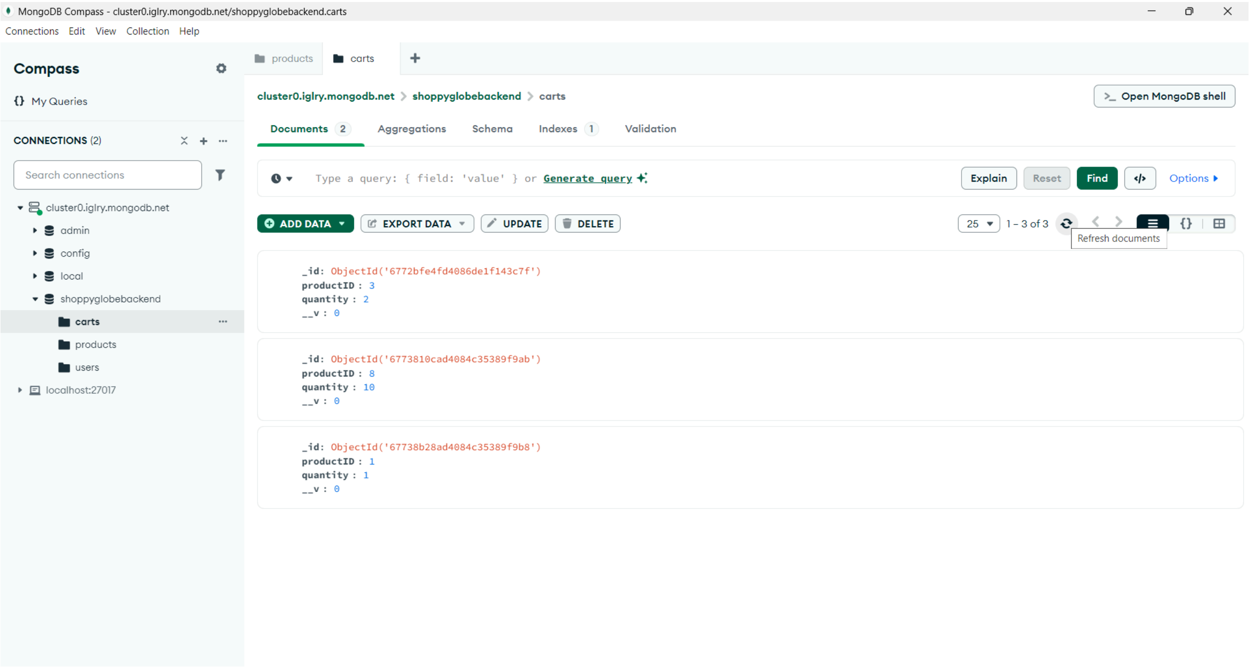
```
{
    "id": 8,
    "quantity": 10
}
```

Status: 200 OK Time: 40 ms Size: 980 B

```
                "price": 299.99,
                "description": "Ultra HD 4K monitor
                    with vibrant colors and sharp
                    visuals.",
                "stockQuantity": 10
            }
        },
        {
            "productID": 8,
            "quantity": 10,
            "productDetails": {
                "_id": "676e756b99df570dde2d142e",
                "id": 8,
                "name": "Electric Kettle",
                "price": 39.99,
                "description": "Fast-boil electric
                    kettle with a stainless steel
                    body and auto shut-off.",
                "stockQuantity": 20
            }
        },
        {
            "productID": 1,
            "quantity": 1,
            "productDetails": {
                "_id": "676e5477510bccd4524cf1f7",
                "id": 1,
                "name": "Wireless Earbuds",
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
{ Method: 'GET', URL: '/carts', Status: 401 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'GET', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
```

Set the quantity of product 8 to 10

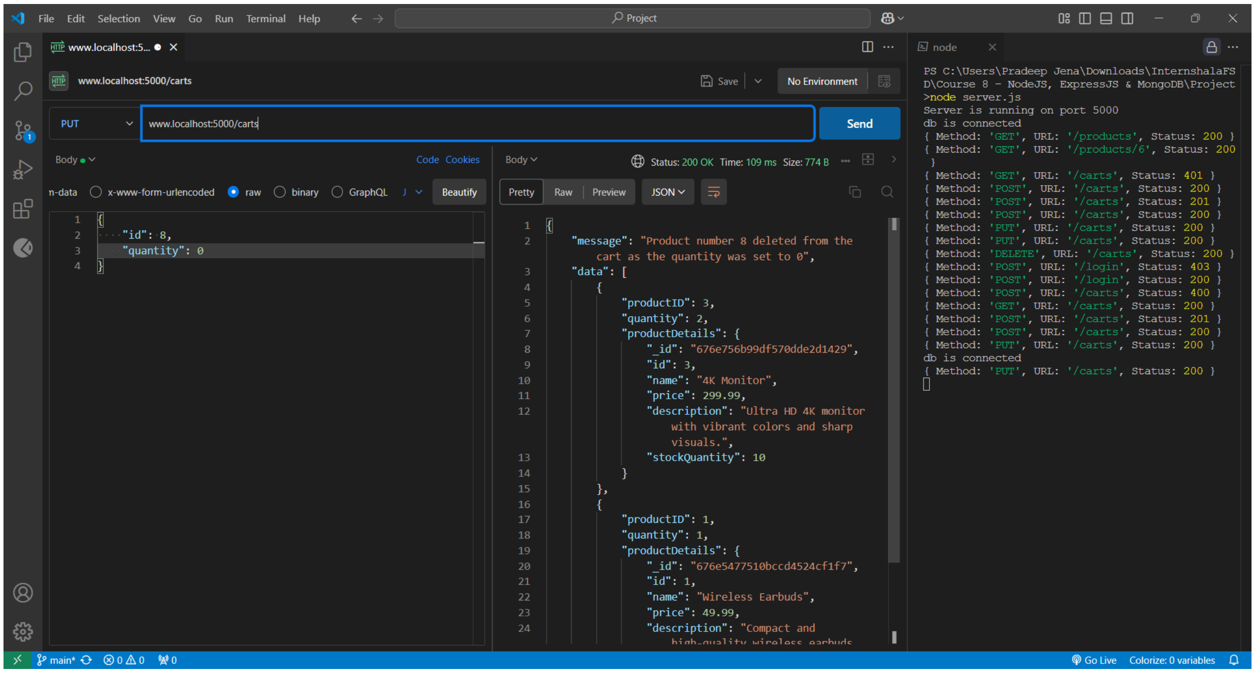


```
_id: ObjectId('6772bfe4fd4086de1f143c7f')
productID : 3
quantity : 2
__v : 0
```

```
_id: ObjectId('6773810cad4084c35389f9ab')
productID : 8
quantity : 10
__v : 0
```

```
_id: ObjectId('67738b28ad4084c35389f9b8')
productID : 1
quantity : 1
__v : 0
```

7) **Method**: ‘PUT’ & **URL**: '/carts' & **Status**: 200 OK



PUT www.localhost:5000/carts

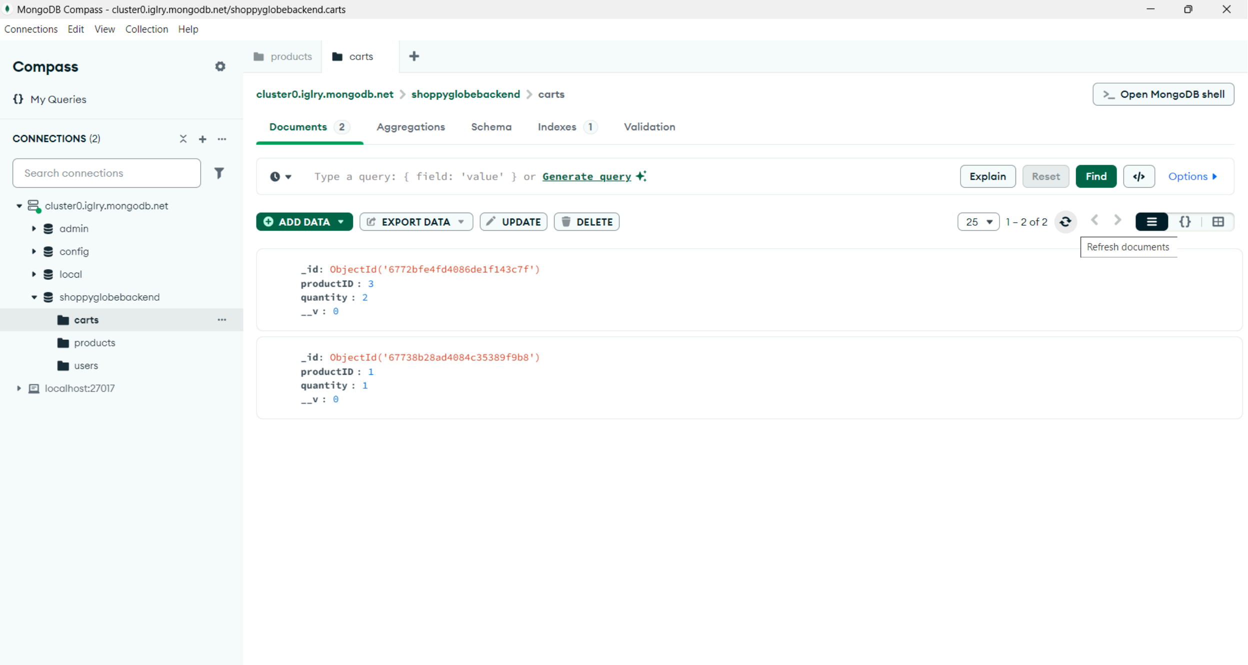
```
{
    "id": 8,
    "quantity": 0
}
```

Status: 200 OK Time: 109 ms Size: 774 B

```
{
    "message": "Product number 8 deleted from the
        cart as the quantity was set to 0",
    "data": [
        {
            "productID": 3,
            "quantity": 2,
            "productDetails": {
                "_id": "676e756b99df570dde2d1429",
                "id": 3,
                "name": "4K Monitor",
                "price": 299.99,
                "description": "Ultra HD 4K monitor
                    with vibrant colors and sharp
                    visuals.",
                "stockQuantity": 10
            }
        },
        {
            "productID": 1,
            "quantity": 1,
            "productDetails": {
                "_id": "676e5477510bccd4524cf1f7",
                "id": 1,
                "name": "Wireless Earbuds",
                "price": 49.99,
                "description": "Compact and
                    high-quality wireless earbuds
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
>node server.js
Server is running on port 5000
db is connected
{ Method: 'GET', URL: '/products', Status: 200 }
{ Method: 'GET', URL: '/products/6', Status: 200
 }
{ Method: 'GET', URL: '/carts', Status: 401 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'GET', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
db is connected
{ Method: 'PUT', URL: '/carts', Status: 200 }
```

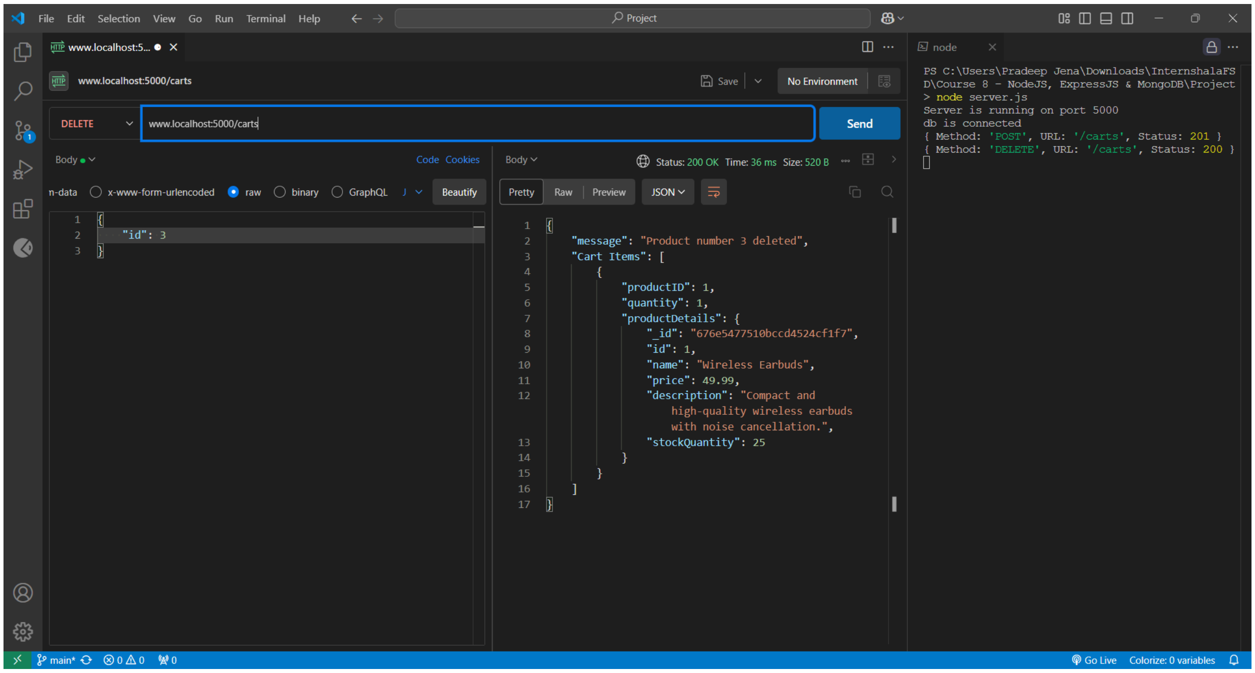
In case the product quantity is set to 0, it will automatically get deleted from the cart



```
_id: ObjectId('6772bfe4fd4086de1f143c7f')
productID : 3
quantity : 2
__v : 0
```

```
_id: ObjectId('67738b28ad4084c35389f9b8')
productID : 1
quantity : 1
__v : 0
```

8) **Method**: ‘DELETE’ & **URL**: '/carts' & **Status**: 200 OK



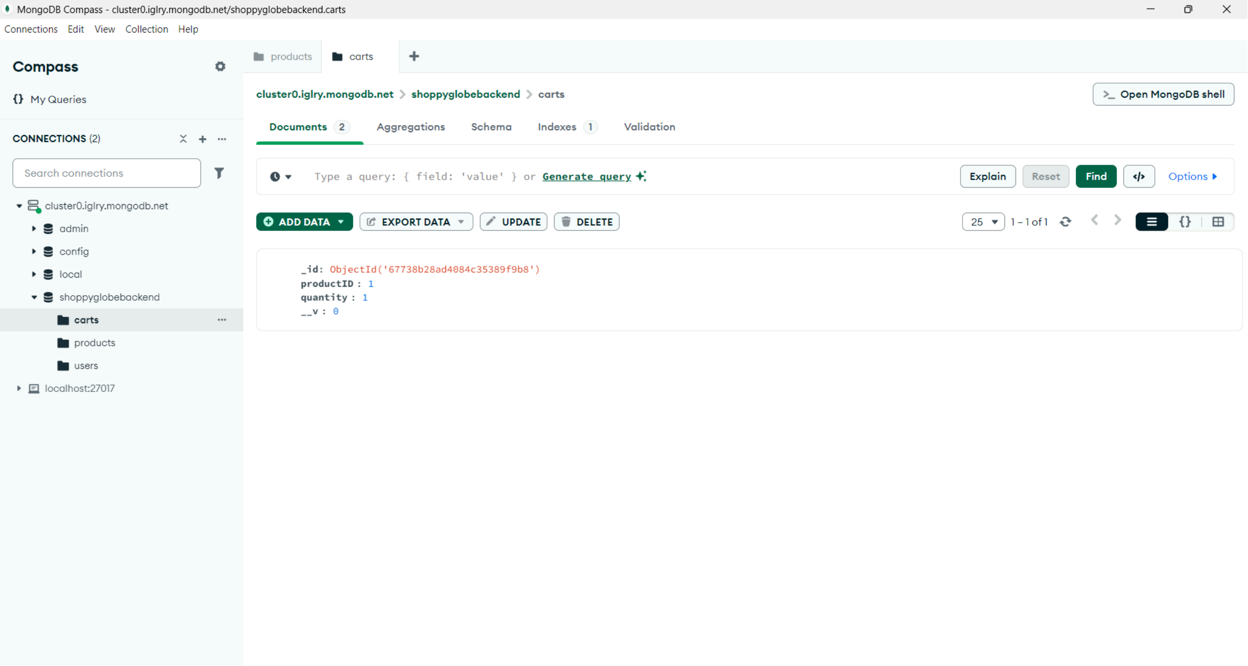
DELETE www.localhost:5000/carts

```
{
    "id": 3
}
```

Status: 200 OK Time: 36 ms Size: 520 B

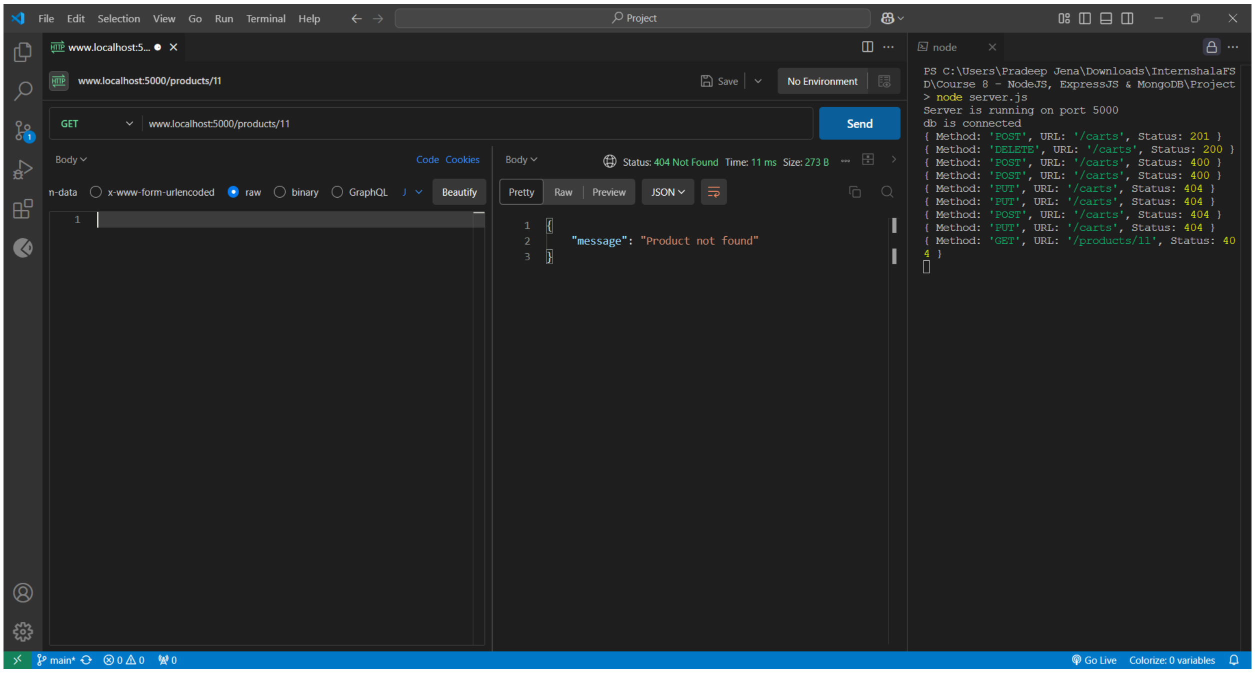
```
{
    "message": "Product number 3 deleted",
    "Cart Items": [
        {
            "productID": 1,
            "quantity": 1,
            "productDetails": {
                "_id": "676e5477510bccd4524cf1f7",
                "id": 1,
                "name": "Wireless Earbuds",
                "price": 49.99,
                "description": "Compact and
                    high-quality wireless earbuds
                    with noise cancellation.",
                "stockQuantity": 25
            }
        }
    ]
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
```



```
_id: ObjectId('67738b28ad4084c35389f9b8')
productID : 1
quantity : 1
__v : 0
```

Validation:



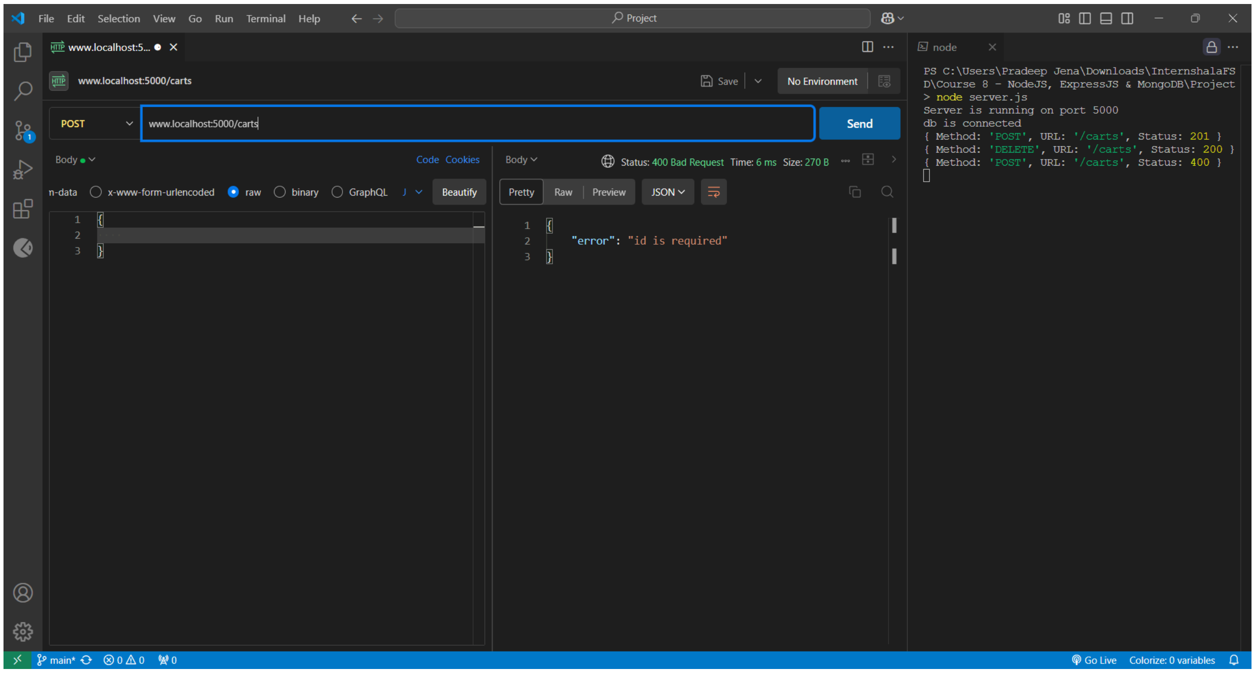
GET www.localhost:5000/products/11

Status: 404 Not Found Time: 11 ms Size: 273 B

```
{
  "message": "Product not found"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'POST', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'GET', URL: '/products/11', Status: 40
4 }
```

The given id of the products is not present



POST www.localhost:5000/carts

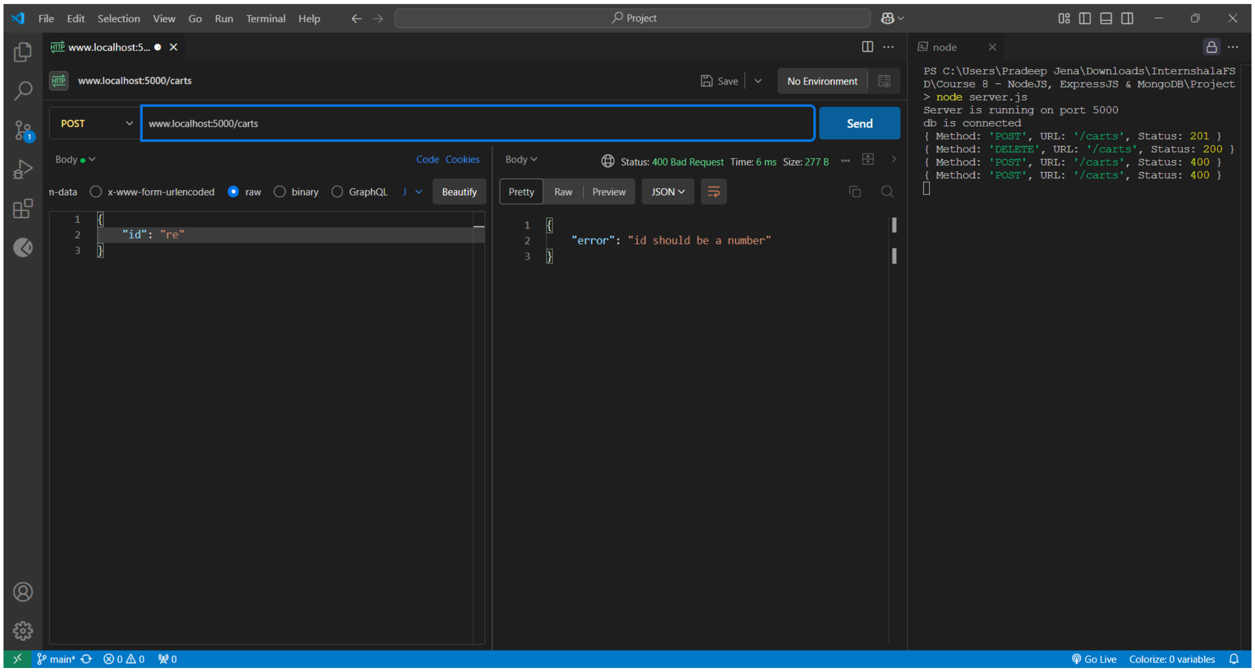
```
{

}
```

Status: 400 Bad Request Time: 6 ms Size: 270 B

```
{
  "error": "id is required"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
```

In case id is not provided for POST method.



POST www.localhost:5000/carts

form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

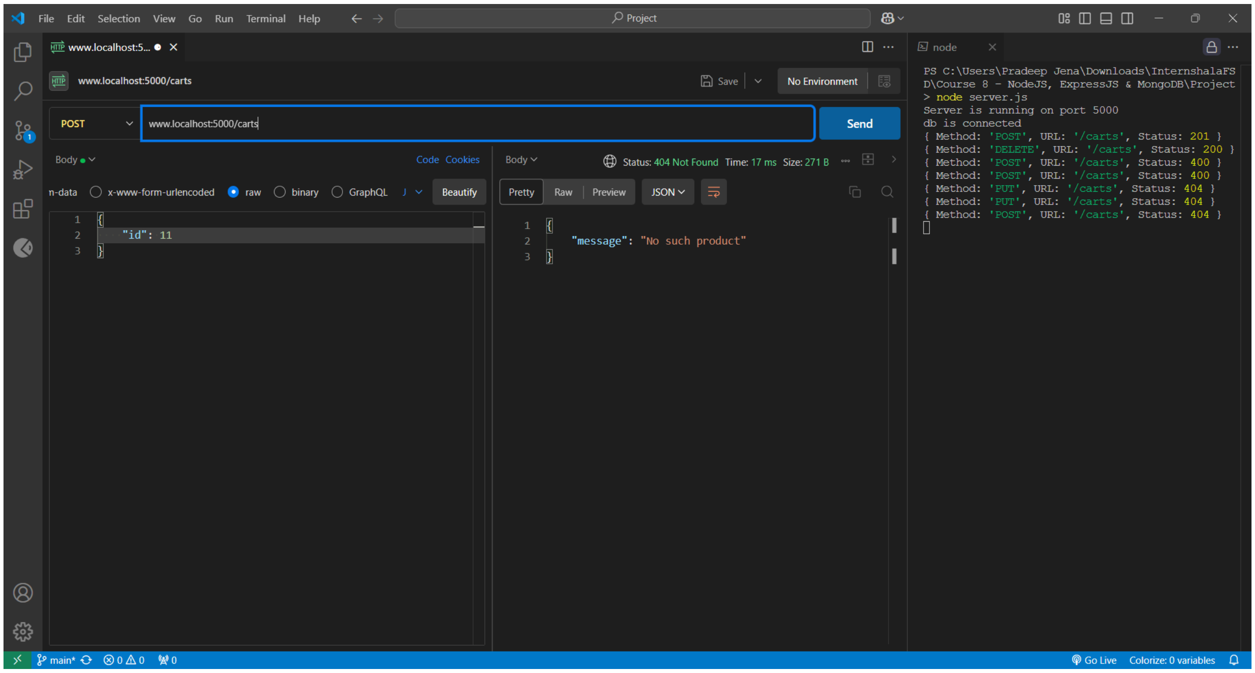
```
{
    "id": "re"
}
```

Status: 400 Bad Request Time: 6 ms Size: 277 B

```
{
    "error": "id should be a number"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
```

In case the given id is not a number for POST method..



POST www.localhost:5000/carts

form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

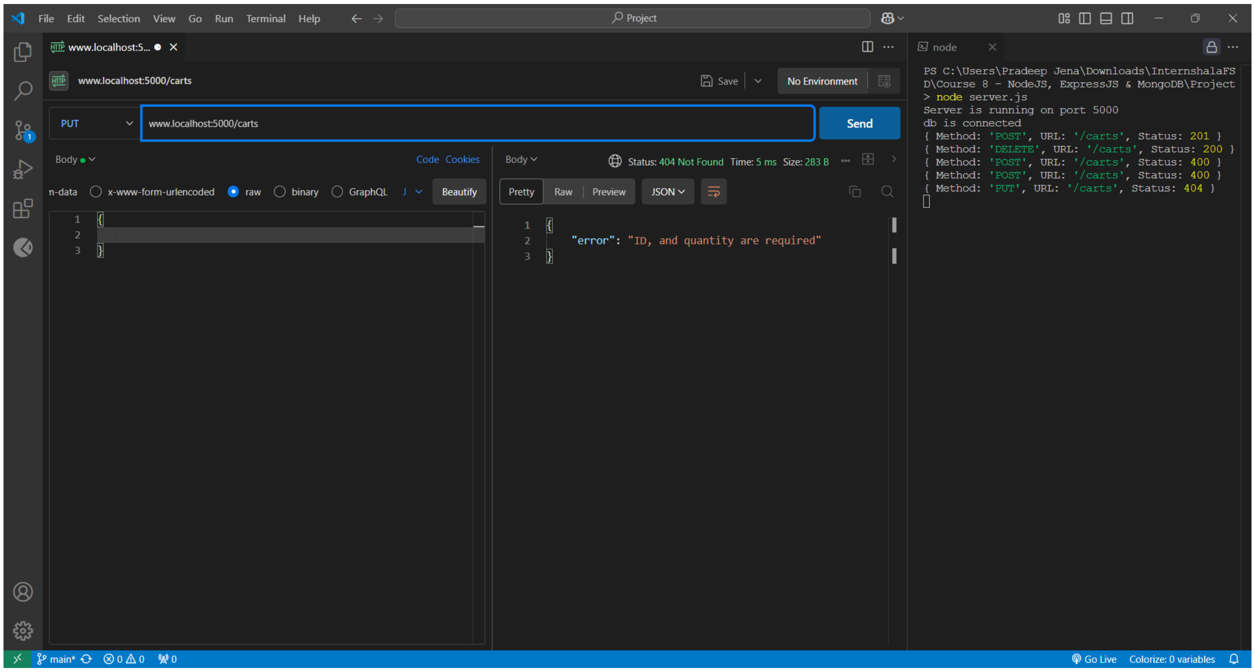
```
{
    "id": 11
}
```

Status: 404 Not Found Time: 17 ms Size: 271 B

```
{
    "message": "No such product"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'POST', URL: '/carts', Status: 404 }
```

In case, the id given is not present.



PUT www.localhost:5000/carts

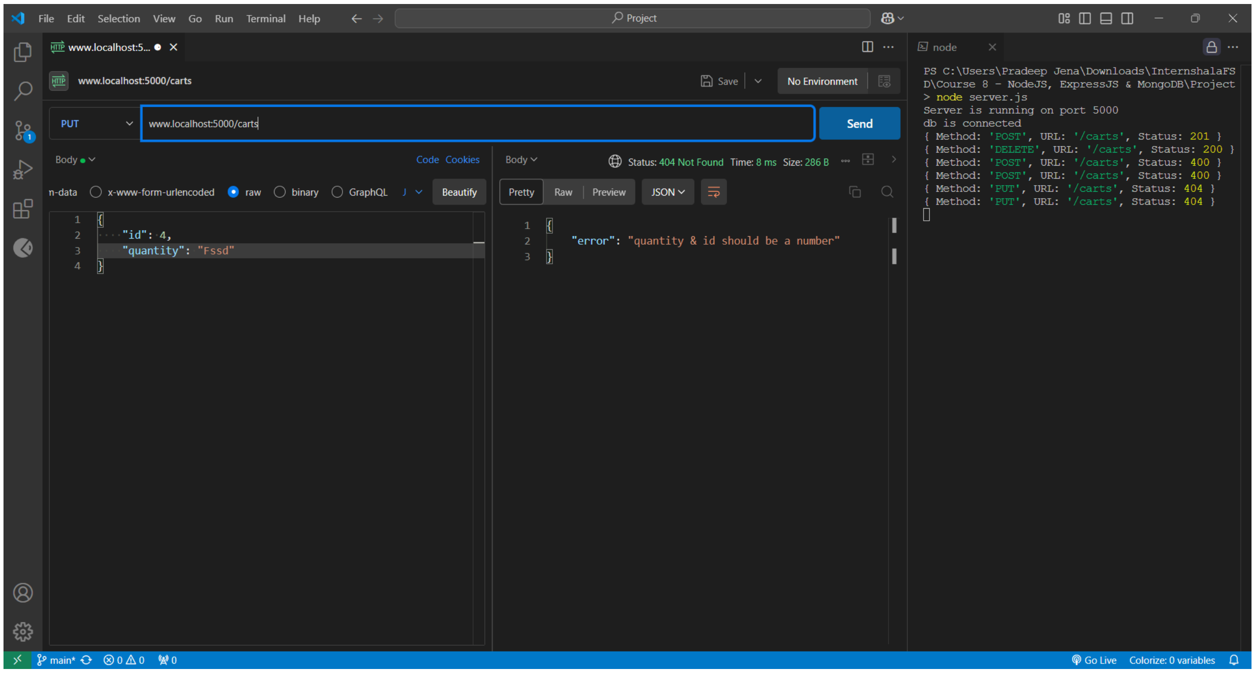
```
{

}
```

Status: 404 Not Found Time: 5 ms Size: 283 B

```
{
    "error": "ID, and quantity are required"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
```

If Id & Quantity is not provided for the PUT method.



PUT www.localhost:5000/carts

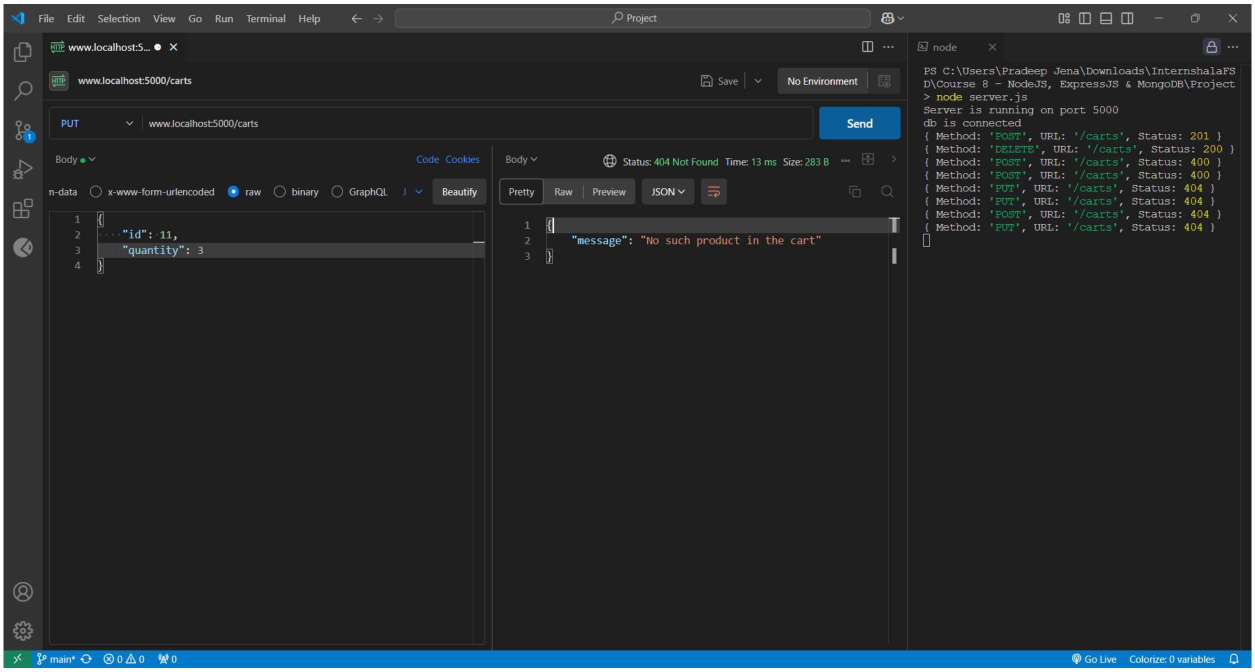
```
{
    "id": 4,
    "quantity": "Fssd"
}
```

Status: 404 Not Found Time: 8 ms Size: 286 B

```
{
    "error": "quantity & id should be a number"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
```

In case the given id & quantity are not numbers for PUT method.



PUT www.localhost:5000/carts

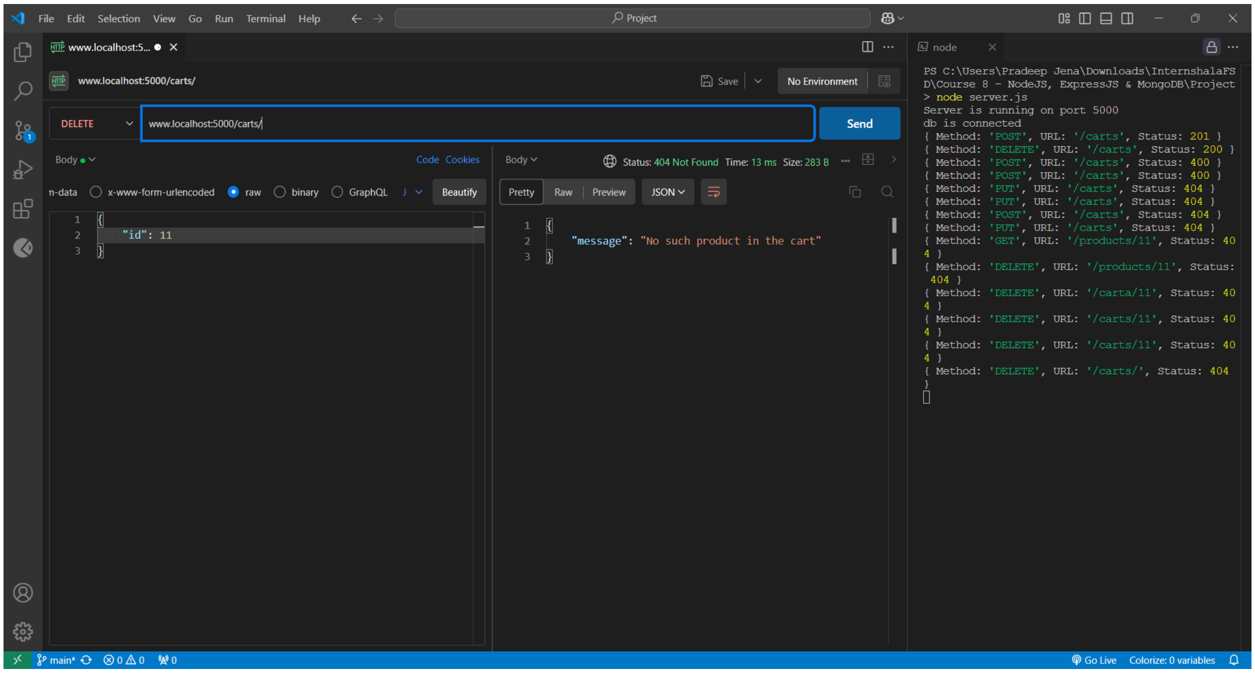
```
{
    "id": 11,
    "quantity": 3
}
```

Status: 404 Not Found Time: 13 ms Size: 283 B

```
{
    "message": "No such product in the cart"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'POST', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
```

The given id of the product is not present in the cart.



DELETE www.localhost:5000/carts/

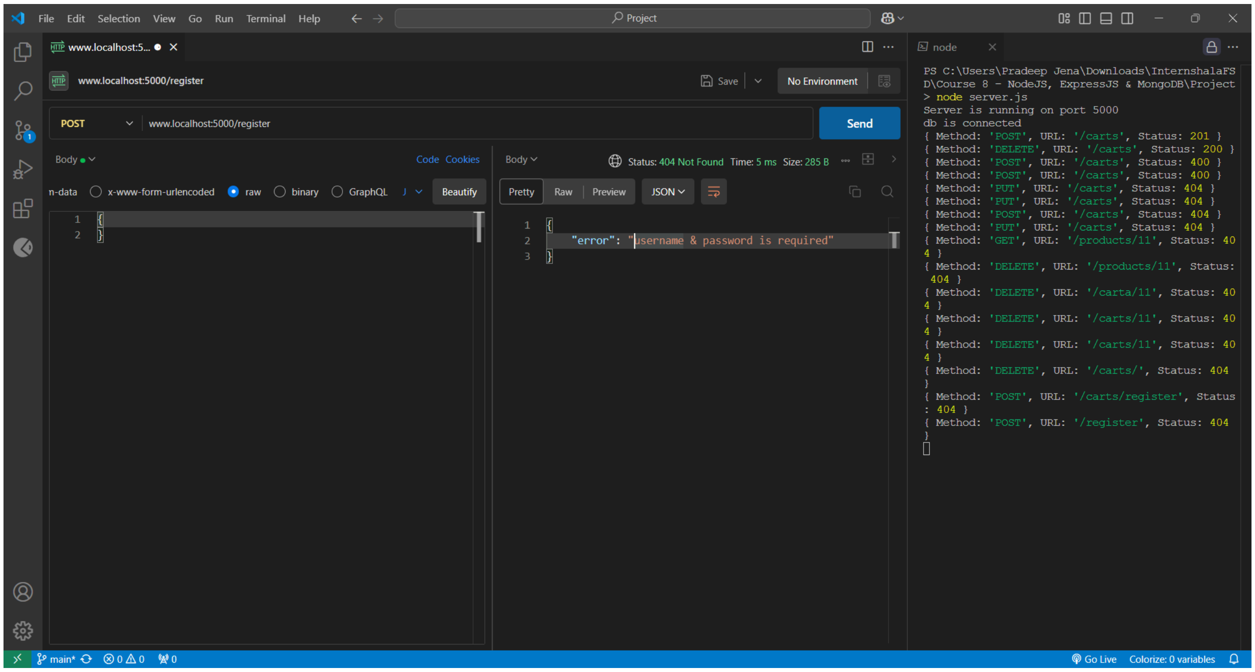
```
{
    "id": 11
}
```

Status: 404 Not Found Time: 13 ms Size: 283 B

```
{
    "message": "No such product in the cart"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'POST', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'GET', URL: '/products/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/products/11', Status:
 404 }
{ Method: 'DELETE', URL: '/carta/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/', Status: 404
}
```

The id you want to delete is not present in the cart.


File Edit Selection View Go Run Terminal Help
Project
Save
No Environment
Send
Code Cookies
n-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL
Beautify
Pretty Raw Preview JSON
Go Live Colorize: 0 variables
main*

www.localhost:5000/register

POST www.localhost:5000/register

Body

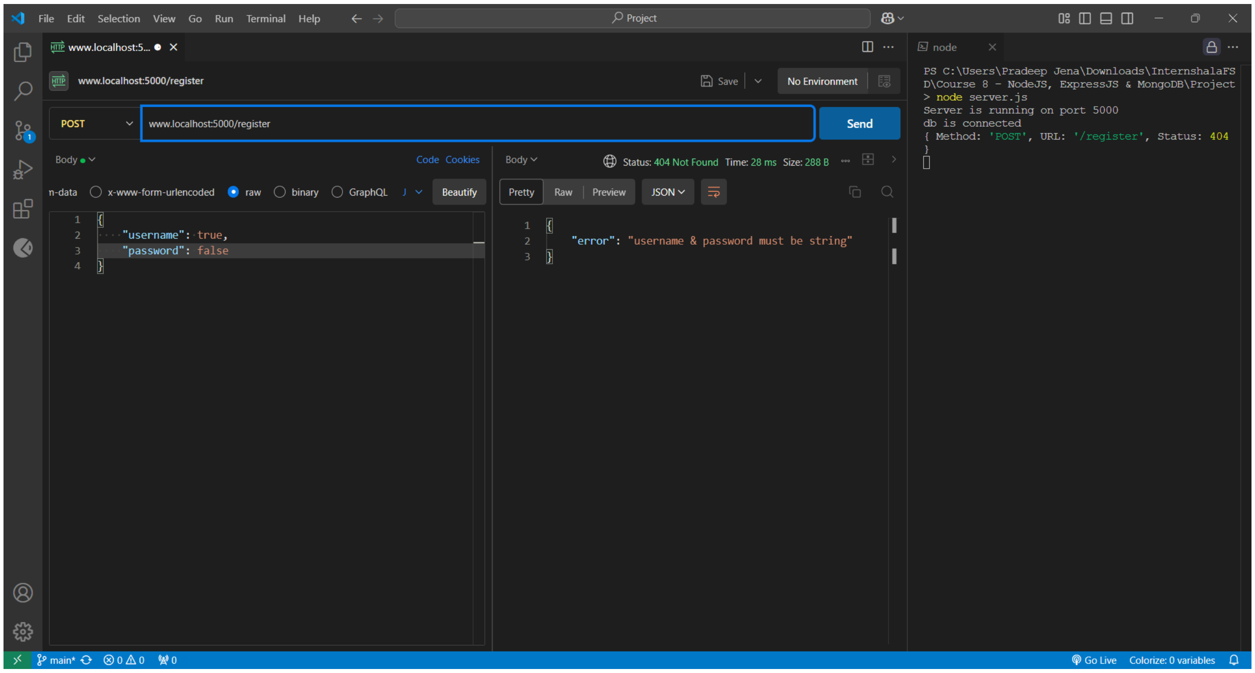
```
{
}
```

Status: 404 Not Found Time: 5 ms Size: 285 B

```
{
    "error": "username & password is required"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/carts', Status: 201 }
{ Method: 'DELETE', URL: '/carts', Status: 200 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'POST', URL: '/carts', Status: 400 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'POST', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'GET', URL: '/products/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/products/11', Status:
 404 }
{ Method: 'DELETE', URL: '/carta/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/11', Status: 40
4 }
{ Method: 'DELETE', URL: '/carts/', Status: 404
}
{ Method: 'POST', URL: '/carts/register', Status
: 404 }
{ Method: 'POST', URL: '/register', Status: 404
}
```

Username and password is required to register a new user in register/login POST method.


File Edit Selection View Go Run Terminal Help
Project
Save
No Environment
Send
Code Cookies
n-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL
Beautify
Pretty Raw Preview JSON
Go Live Colorize: 0 variables
main*

www.localhost:5000/register

POST www.localhost:5000/register

Body

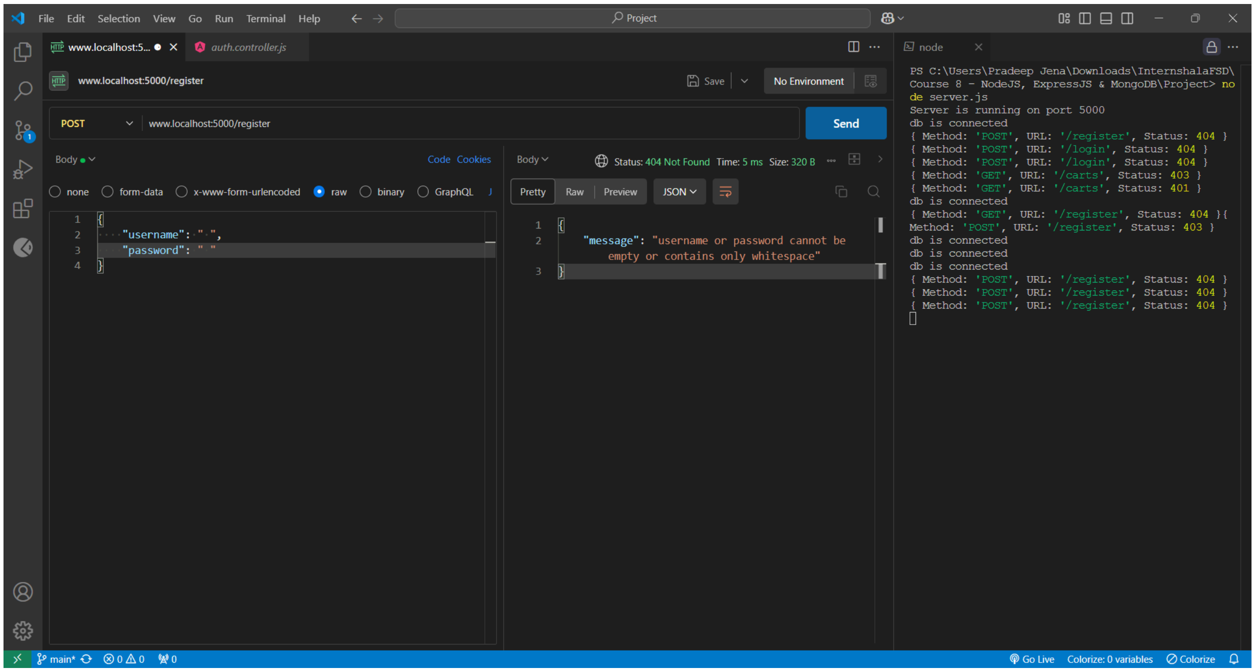
```
{
    "username": true,
    "password": false
}
```

Status: 404 Not Found Time: 28 ms Size: 288 B

```
{
    "error": "username & password must be string"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404
}
```

Username & password need to be string in register/login POST method.


www.localhost:5000/register
Save
No Environment
Send
Status: 404 Not Found Time: 5 ms Size: 320 B
Pretty
Raw
Preview
JSON
node

POST www.localhost:5000/register

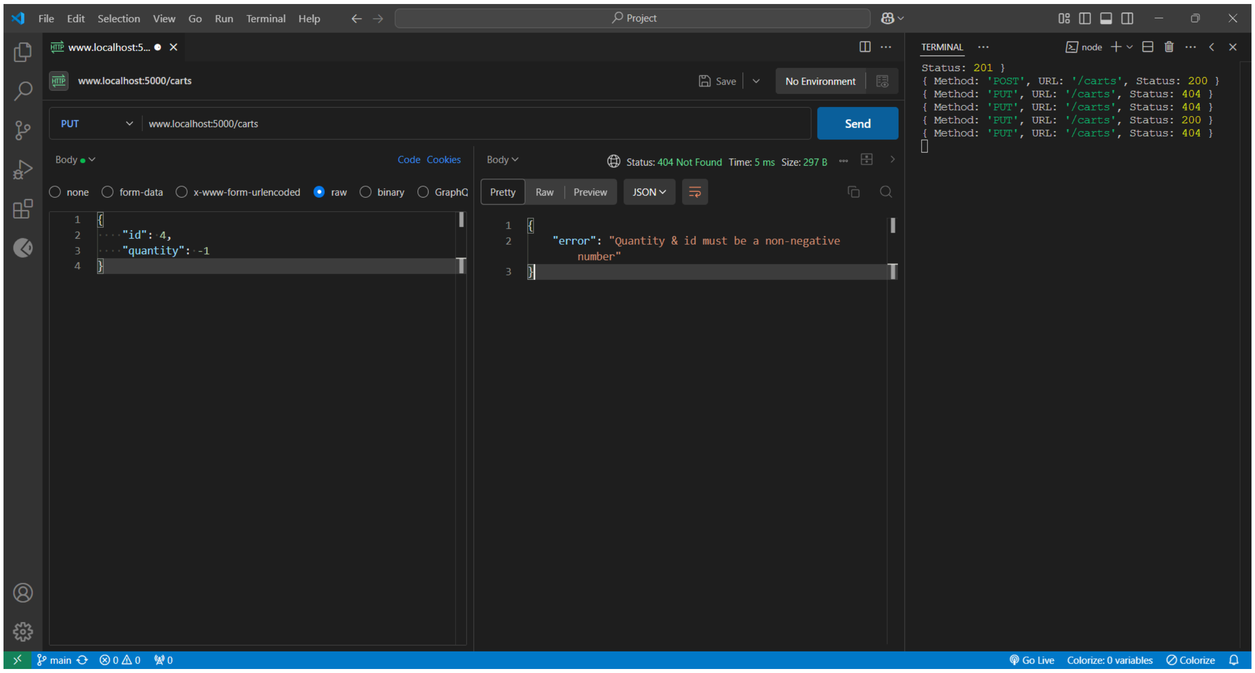
Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

```
{
    "username": " ",
    "password": " "
}
```

```
{
    "message": "username or password cannot be empty or contains only whitespace"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFSD\Course 8 - NodeJS, ExpressJS & MongoDB\Project> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }{
Method: 'POST', URL: '/register', Status: 403 }
db is connected
db is connected
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
```

If you try to register/login whitespace.



PUT www.localhost:5000/carts

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQ

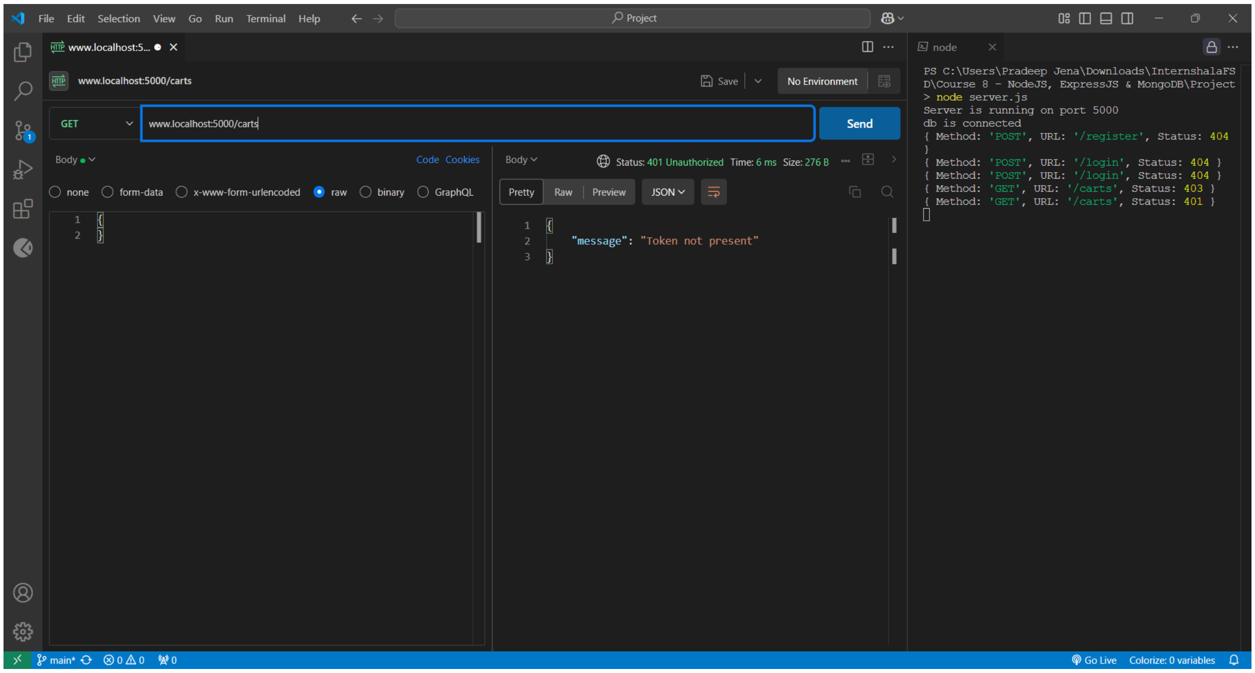
```
{
    "id": 4,
    "quantity": -1
}
```

```
{
    "error": "Quantity & id must be a non-negative number"
}
```

```
Status: 201 }
{ Method: 'POST', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
{ Method: 'PUT', URL: '/carts', Status: 200 }
{ Method: 'PUT', URL: '/carts', Status: 404 }
```

In case trying to access a product using -ve number

Authentication:



GET www.localhost:5000/carts

Body: raw

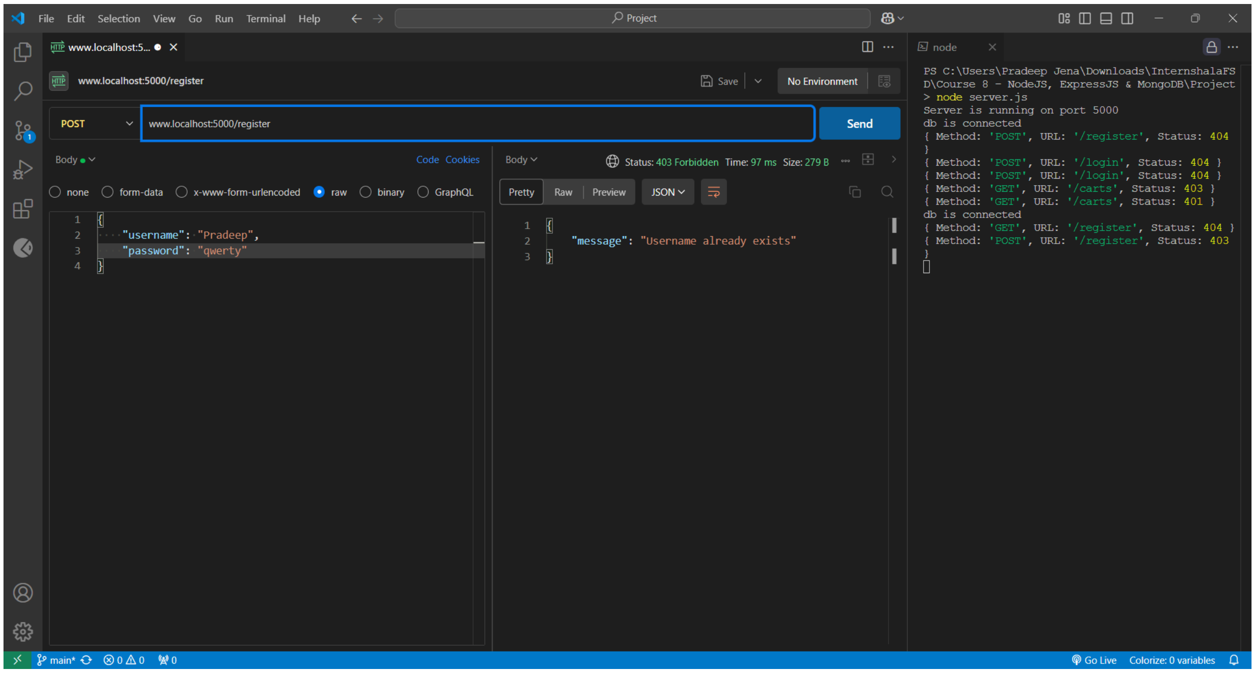
```
{
}
```

Status: 401 Unauthorized Time: 6 ms Size: 276 B

```
{
    "message": "Token not present"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404
}
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
```

Won’t be able to access the cart items until provided by JWT token.



POST www.localhost:5000/register

Body: raw

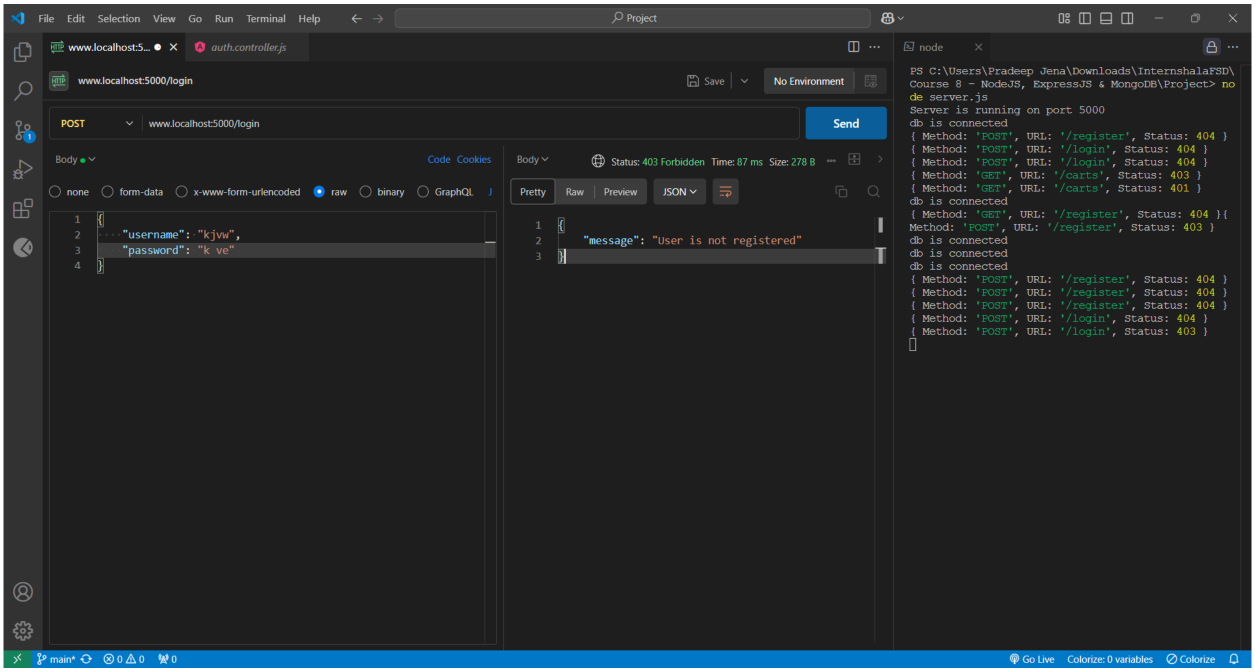
```
{
    "username": "Pradeep",
    "password": "qwerty"
}
```

Status: 403 Forbidden Time: 97 ms Size: 279 B

```
{
    "message": "Username already exists"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFS
D\Course 8 - NodeJS, ExpressJS & MongoDB\Project
> node server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404
}
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 403
}
```

If you try to register a username which is already there.



www.localhost:5000/login

POST www.localhost:5000/login

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

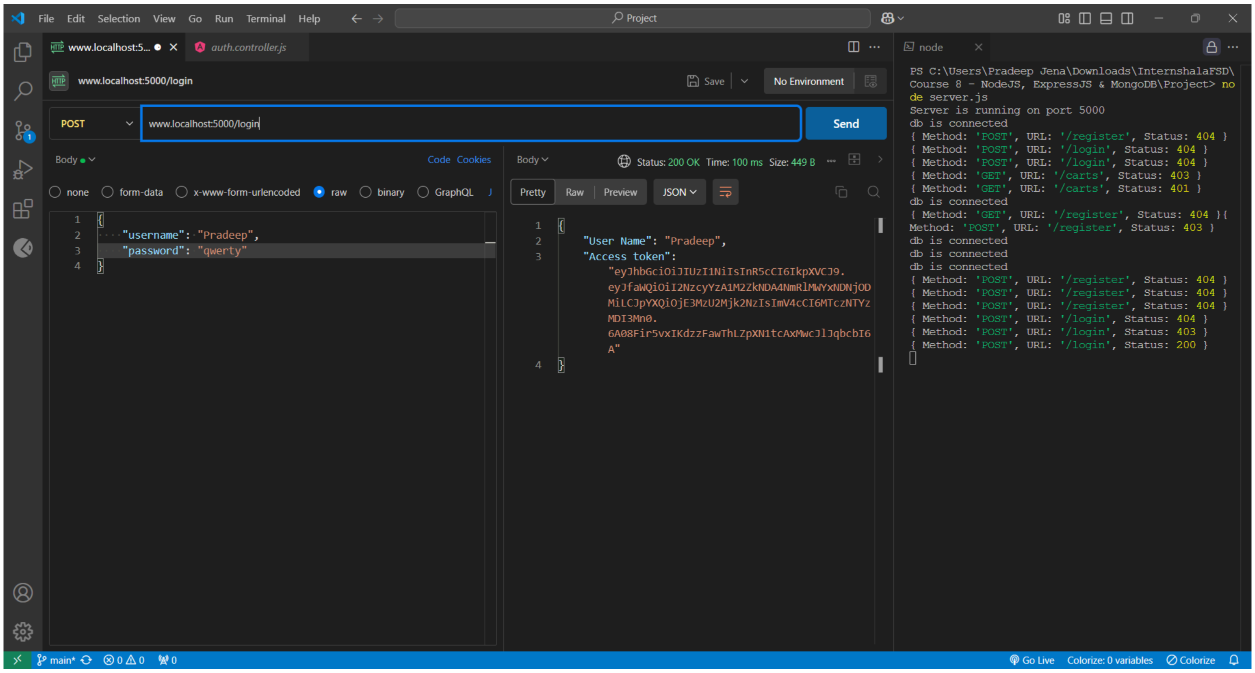
```
{
    "username": "kjvw",
    "password": "k ve"
}
```

Status: 403 Forbidden Time: 87 ms Size: 278 B

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

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFSD\
Course 8 - NodeJS, ExpressJS & MongoDB\Project> no
de server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }{
Method: 'POST', URL: '/register', Status: 403 }
db is connected
db is connected
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 403 }
```

Trying to login a user which is not registered.



www.localhost:5000/login

POST www.localhost:5000/login

Body: ○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

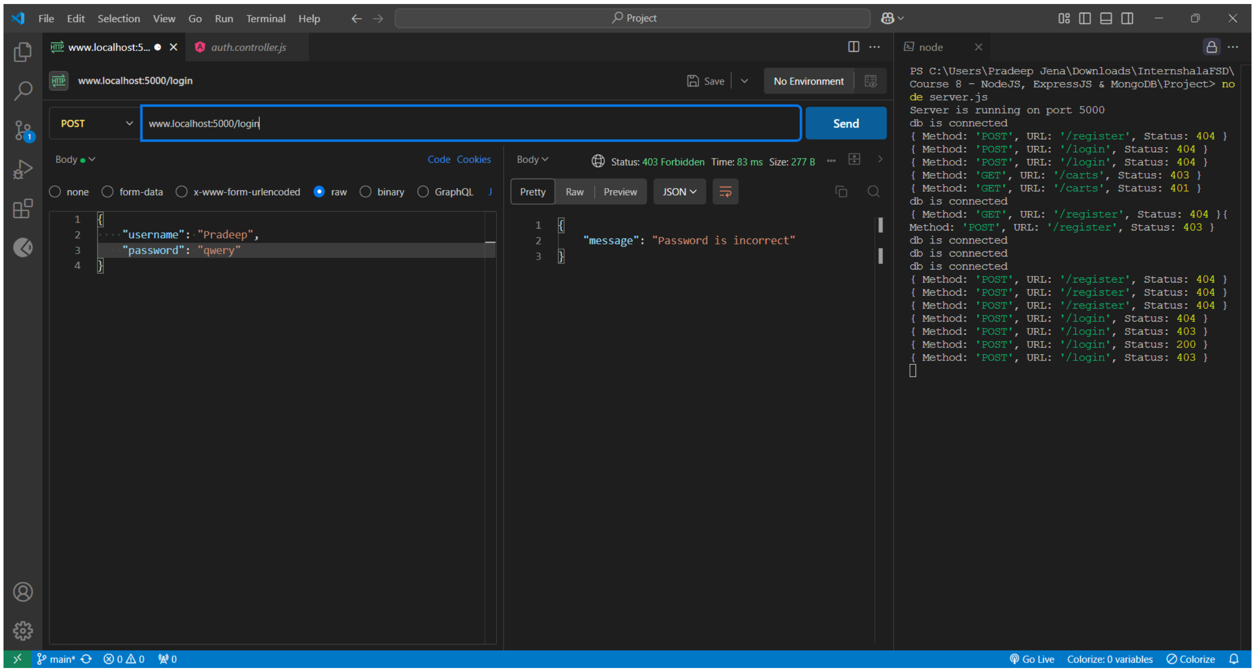
```
{
    "username": "Pradeep",
    "password": "qwerty"
}
```

Status: 200 OK Time: 100 ms Size: 449 B

```
{
    "User Name": "Pradeep",
    "Access token":
        "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJfaWQiOiI2NzcyYzA1M2ZkNDA4NmRlMWYxNDNjOD
        MiLCJpYXQiOjE3MzU2Mjk2NzIsImV4cCI6MTczNTYz
        MDI3Mn0.
        6A08Fir5vxIKdzzFawThLZpXN1tcAxMwcJlJqbcbI6
        A"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFSD\
Course 8 - NodeJS, ExpressJS & MongoDB\Project> no
de server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }{
Method: 'POST', URL: '/register', Status: 403 }
db is connected
db is connected
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
```

Login user with correct password



www.localhost:5000/login

POST www.localhost:5000/login

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

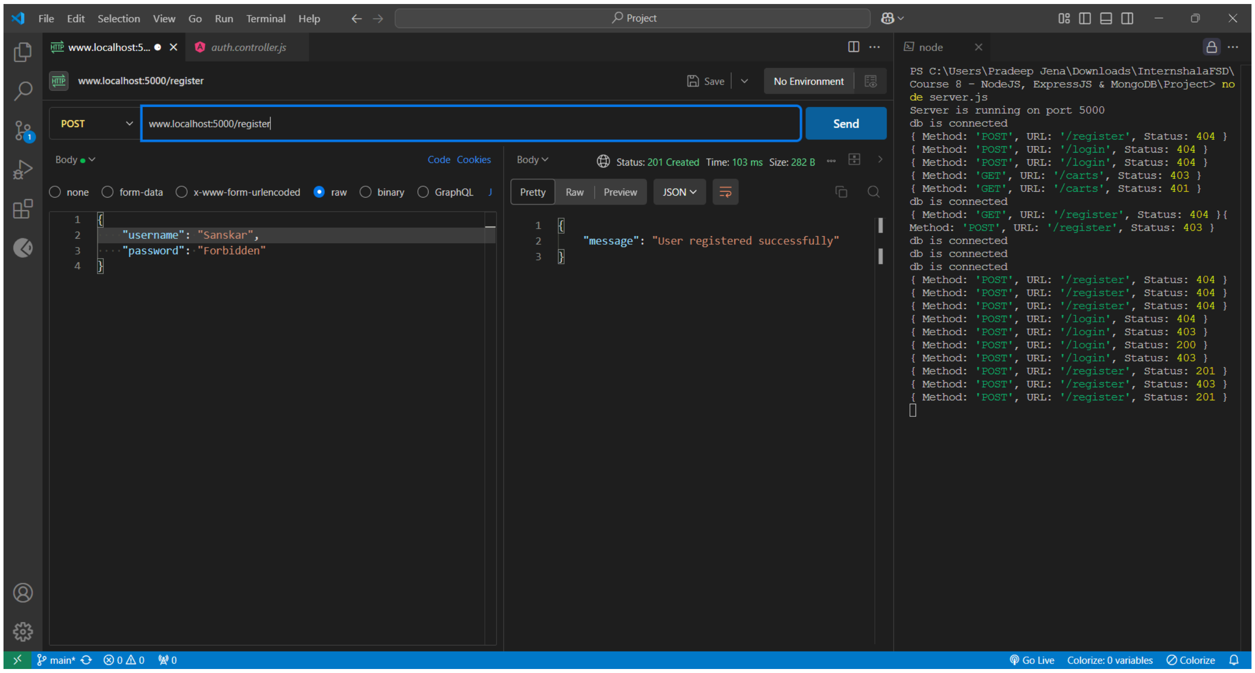
```
{
    "username": "Pradeep",
    "password": "qwery"
}
```

Status: 403 Forbidden Time: 83 ms Size: 277 B

```
{
    "message": "Password is incorrect"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFSD\
Course 8 - NodeJS, ExpressJS & MongoDB\Project> no
de server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }{
Method: 'POST', URL: '/register', Status: 403 }
db is connected
db is connected
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
```

Trying to login with an incorrect password.



www.localhost:5000/register

POST www.localhost:5000/register

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL

```
{
    "username": "Sanskar",
    "password": "Forbidden"
}
```

Status: 201 Created Time: 103 ms Size: 282 B

```
{
    "message": "User registered successfully"
}
```

```
PS C:\Users\Pradeep Jena\Downloads\InternshalaFSD\
Course 8 - NodeJS, ExpressJS & MongoDB\Project> no
de server.js
Server is running on port 5000
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'GET', URL: '/carts', Status: 403 }
{ Method: 'GET', URL: '/carts', Status: 401 }
db is connected
{ Method: 'GET', URL: '/register', Status: 404 }{
Method: 'POST', URL: '/register', Status: 403 }
db is connected
db is connected
db is connected
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/register', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 404 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/login', Status: 200 }
{ Method: 'POST', URL: '/login', Status: 403 }
{ Method: 'POST', URL: '/register', Status: 201 }
{ Method: 'POST', URL: '/register', Status: 403 }
{ Method: 'POST', URL: '/register', Status: 201 }
```

Registering a new user