1. GET /products - Get All Products

Description: Retrieves all products available in the database. Useful for displaying product listings on the website.

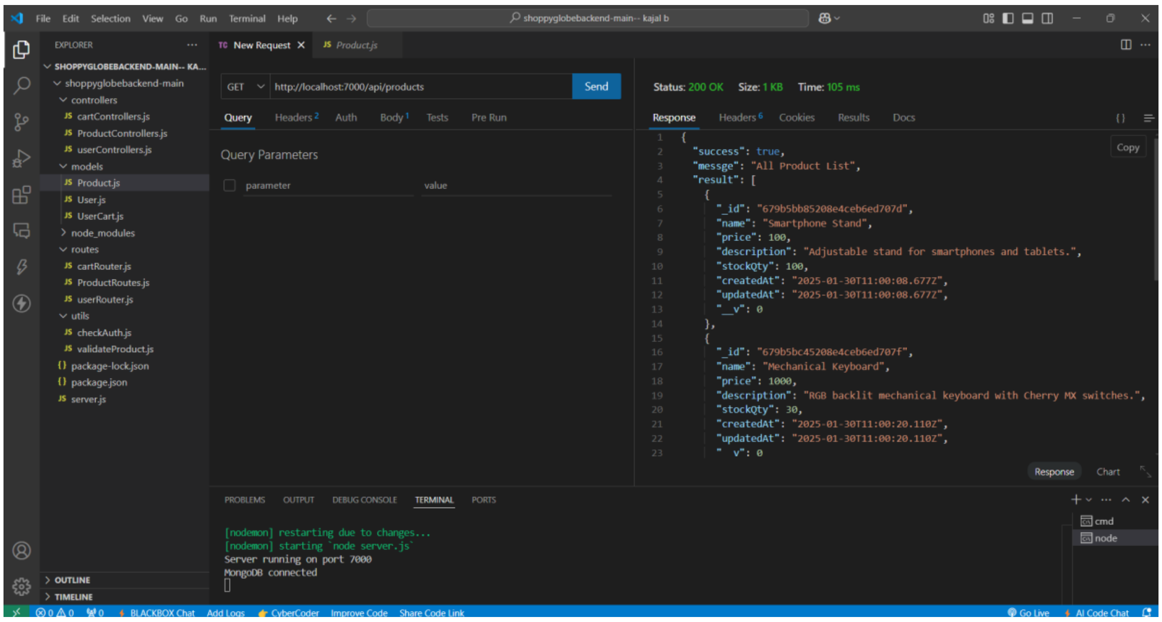
Request:

- **Method:** GET
- **URL:** /products

Example Request:

http://localhost:7000/api/products

ScreenShots:

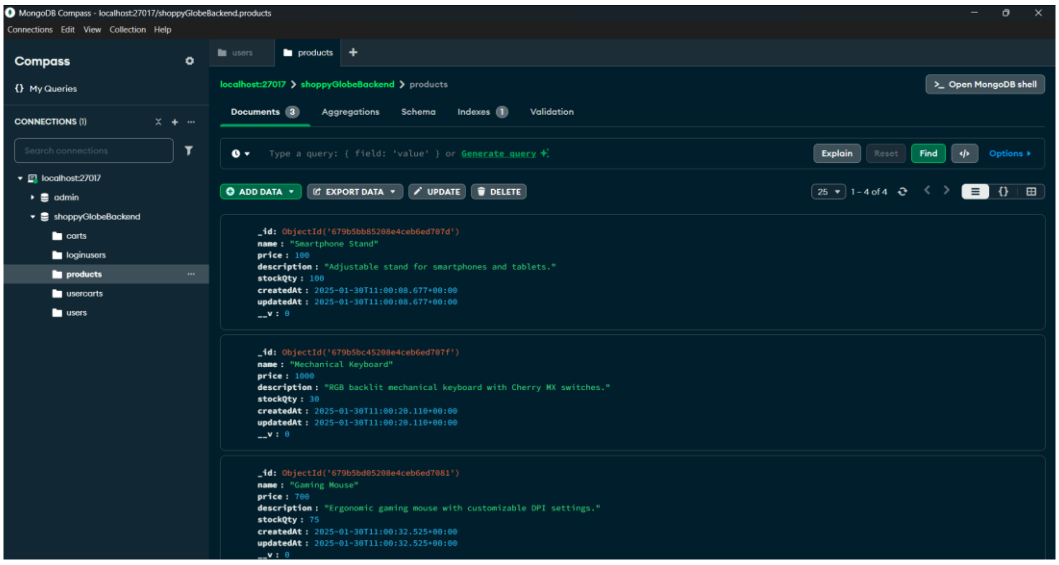


```
GET  http://localhost:7000/api/products  Send

Status: 200 OK   Size: 1 KB   Time: 105 ms
```

```
{
  "success": true,
  "messge": "All Product List",
  "result": [
    {
      "_id": "679b5bb85208e4ceb6ed707d",
      "name": "Smartphone Stand",
      "price": 100,
      "description": "Adjustable stand for smartphones and tablets.",
      "stockQty": 100,
      "createdAt": "2025-01-30T11:00:08.677Z",
      "updatedAt": "2025-01-30T11:00:08.677Z",
      "__v": 0
    },
    {
      "_id": "679b5bc45208e4ceb6ed707f",
      "name": "Mechanical Keyboard",
      "price": 1000,
      "description": "RGB backlit mechanical keyboard with Cherry MX switches.",
      "stockQty": 30,
      "createdAt": "2025-01-30T11:00:20.110Z",
      "updatedAt": "2025-01-30T11:00:20.110Z",
      "__v": 0
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server running on port 7000
MongoDB connected
```



```
localhost:27017 > shoppyGlobeBackend > products

_id: ObjectId('679b5bb85208e4ceb6ed707d')
name : "Smartphone Stand"
price : 100
description : "Adjustable stand for smartphones and tablets."
stockQty : 100
createdAt : 2025-01-30T11:00:08.677+00:00
updatedAt : 2025-01-30T11:00:08.677+00:00
__v : 0

_id: ObjectId('679b5bc45208e4ceb6ed707f')
name : "Mechanical Keyboard"
price : 1000
description : "RGB backlit mechanical keyboard with Cherry MX switches."
stockQty : 30
createdAt : 2025-01-30T11:00:20.110+00:00
updatedAt : 2025-01-30T11:00:20.110+00:00
__v : 0

_id: ObjectId('679b5bd05208e4ceb6ed7081')
name : "Gaming Mouse"
price : 700
description : "Ergonomic gaming mouse with customizable DPI settings."
stockQty : 75
createdAt : 2025-01-30T11:00:32.525+00:00
updatedAt : 2025-01-30T11:00:32.525+00:00
__v : 0
```

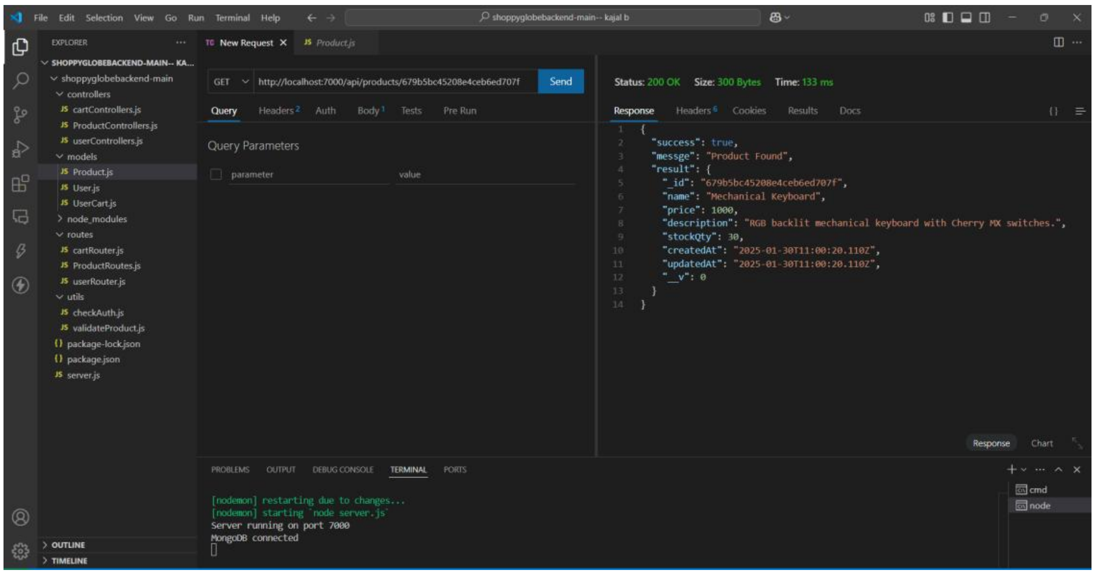
2. GET /products/:id - Get Product by ID

Description: Fetches a specific product using its unique ID. This is useful when viewing detailed product information.

Request:

- **Method:** GET
- **URL:** /products/:id

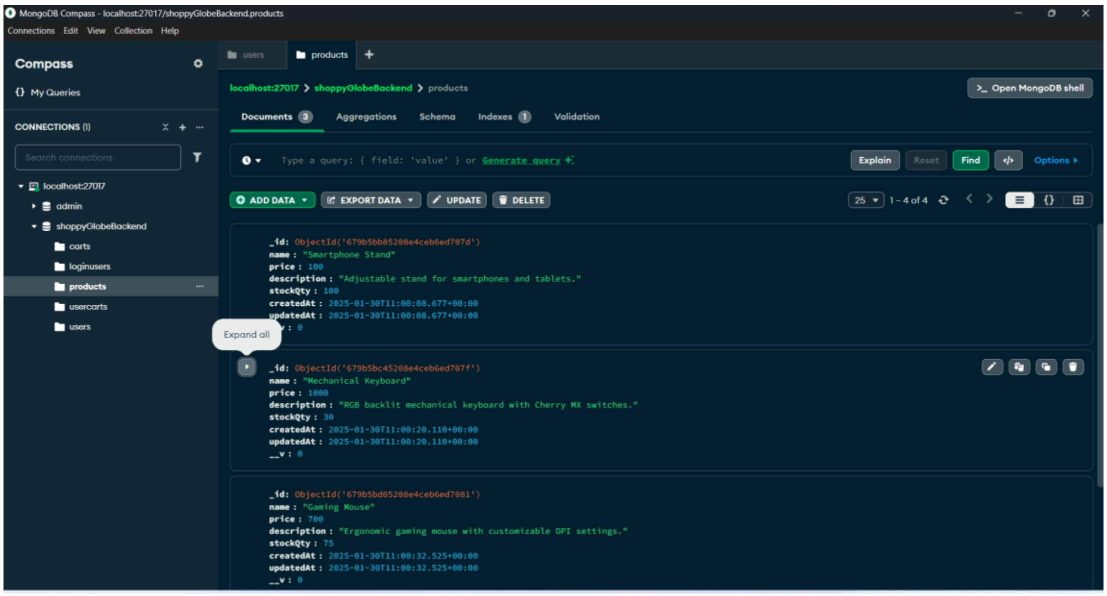
ScreenShots:



GET
http://localhost:7000/api/products/679b5bc45208e4ceb6ed707f
Send
Status: 200 OK Size: 300 Bytes Time: 133 ms
Query Parameters

```
{
  "success": true,
  "messge": "Product Found",
  "result": {
    "_id": "679b5bc45208e4ceb6ed707f",
    "name": "Mechanical Keyboard",
    "price": 1000,
    "description": "RGB backlit mechanical keyboard with Cherry MX switches.",
    "stockQty": 30,
    "createdAt": "2025-01-30T11:00:20.110Z",
    "updatedAt": "2025-01-30T11:00:20.110Z",
    "__v": 0
  }
}
```

```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server running on port 7000
MongoDB connected
```



MongoDB Compass - localhost:27017/shoppyGlobeBackend.products
localhost:27017 > shoppyGlobeBackend > products
Documents 3
Aggregations
Schema
Indexes 1
Validation
1 - 4 of 4

```
_id: ObjectId('679b5bb85208e4ceb6ed707d')
name : "Smartphone Stand"
price : 100
description : "Adjustable stand for smartphones and tablets."
stockQty : 100
createdAt : 2025-01-30T11:00:08.677+00:00
updatedAt : 2025-01-30T11:00:08.677+00:00
__v : 0

_id: ObjectId('679b5bc45208e4ceb6ed707f')
name : "Mechanical Keyboard"
price : 1000
description : "RGB backlit mechanical keyboard with Cherry MX switches."
stockQty : 30
createdAt : 2025-01-30T11:00:20.110+00:00
updatedAt : 2025-01-30T11:00:20.110+00:00
__v : 0

_id: ObjectId('679b5bd05208e4ceb6ed7081')
name : "Gaming Mouse"
price : 700
description : "Ergonomic gaming mouse with customizable DPI settings."
stockQty : 75
createdAt : 2025-01-30T11:00:32.525+00:00
updatedAt : 2025-01-30T11:00:32.525+00:00
__v : 0
```

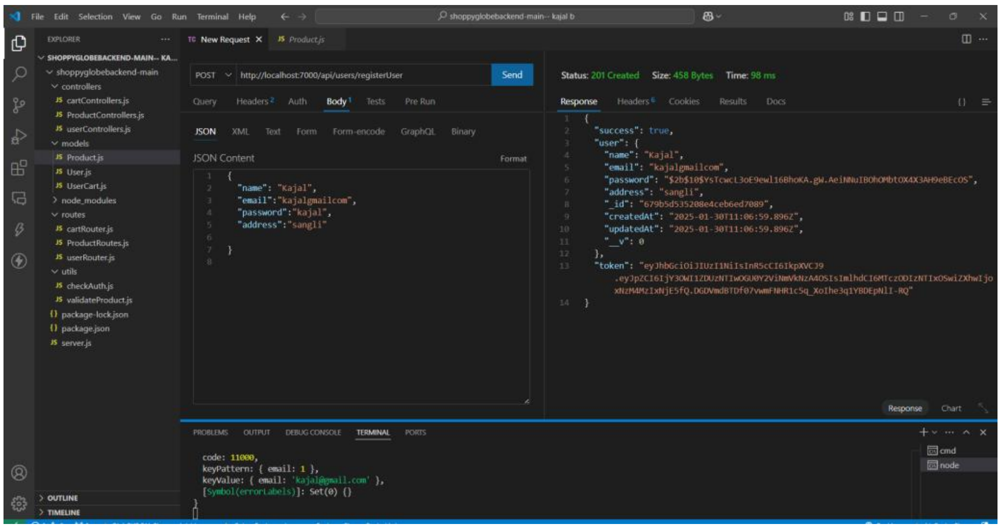
3. POST /register - Register a New User

Description: Allows new users to create an account by providing necessary credentials such as email and password.

Request:

- **Method:** POST
- **URL:** /registerUser

Screenshots:



POST http://localhost:7000/api/users/registerUser

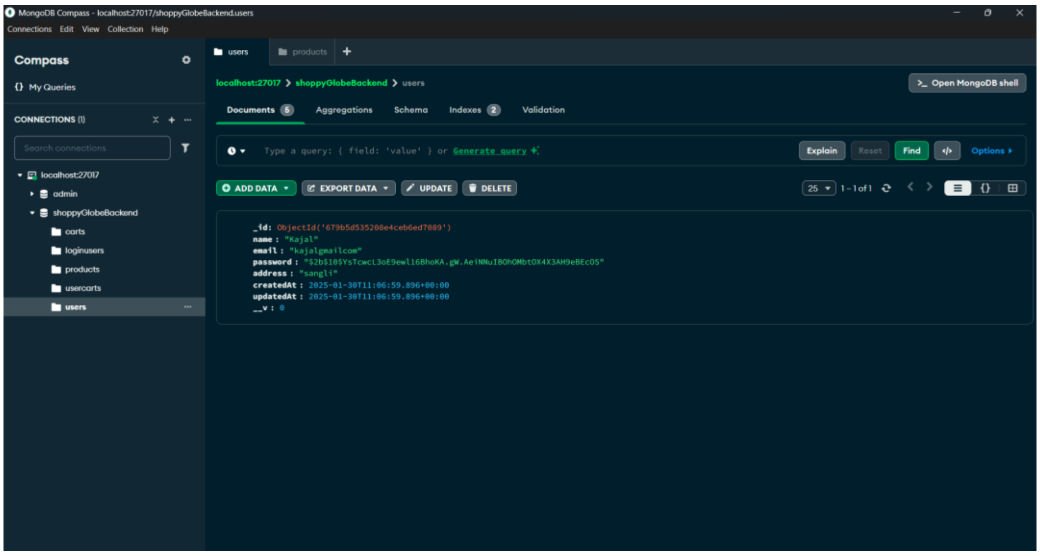
```json
{
  "name": "Kajal",
  "email":"kajalgmailcom",
  "password":"kajal",
  "address":"sangli"

}
```

Status: 201 Created   Size: 458 Bytes   Time: 98 ms

```json
{
  "success": true,
  "user": {
    "name": "Kajal",
    "email": "kajalgmailcom",
    "password": "$2b$10$YsTcwcL3oE9ewl16BhoKA.gW.AeiNNuIBOhOMbtOX4X3AH9eBEcOS",
    "address": "sangli",
    "_id": "679b5d535208e4ceb6ed7089",
    "createdAt": "2025-01-30T11:06:59.896Z",
    "updatedAt": "2025-01-30T11:06:59.896Z",
    "__v": 0
  },
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
      .eyJpZCI6IjY3OWI1ZDUzNTIwOGU0Y2ViNmVkNzA4OSIsImlhdCI6MTczODIzNTIxOSwiZXhwIjo
      xNzM4MzIxNjE5fQ.DGDVmdBTDf07vwmFNHR1c5q_XoIhe3q1YBDEpNlI-RQ"
}
```

```
  code: 11000,
  keyPattern: { email: 1 },
  keyValue: { email: 'kajal@gmail.com' },
  [Symbol(errorLabels)]: Set(0) {}
}
```

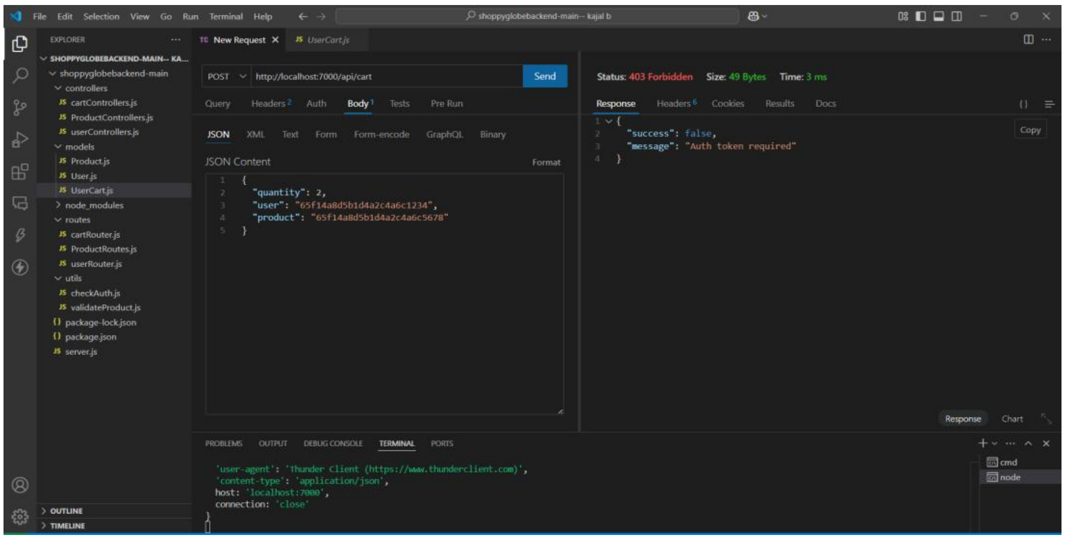


localhost:27017 > shoppyGlobeBackend > users

```
_id: ObjectId('679b5d535208e4ceb6ed7089')
name : "Kajal"
email : "kajalgmailcom"
password : "$2b$10$YsTcwcL3oE9ewl16BhoKA.gW.AeiNNuIBOhOMbtOX4X3AH9eBEcOS"
address : "sangli"
createdAt : 2025-01-30T11:06:59.896+00:00
updatedAt : 2025-01-30T11:06:59.896+00:00
__v : 0
```

Checking Authorization: Adding item to cart without login:

As shown in the below image, it will ask for token in backend.



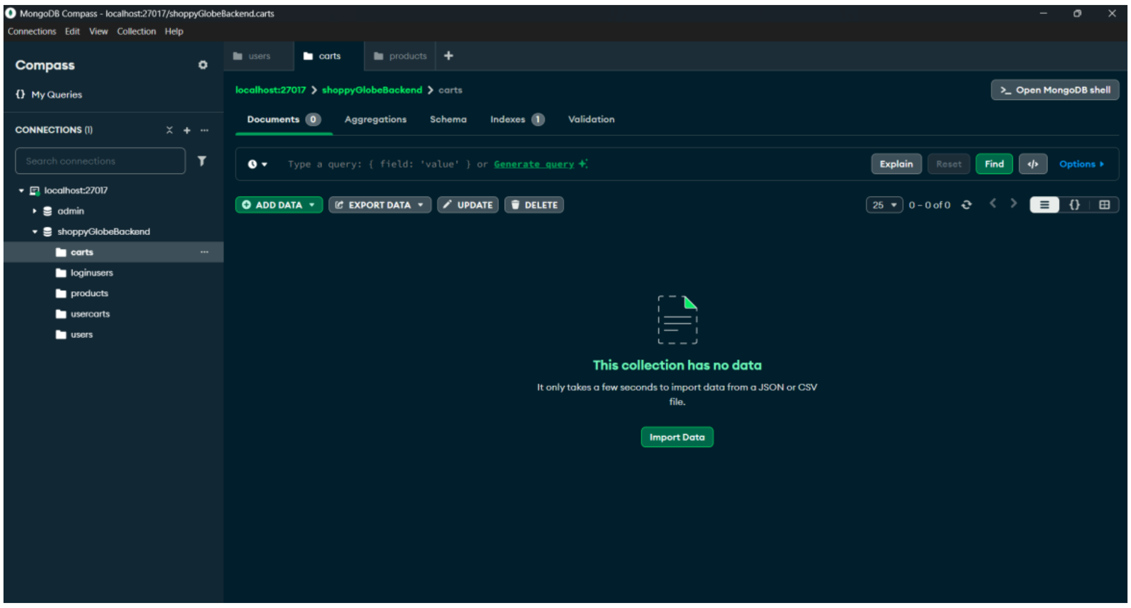
POST http://localhost:7000/api/cart

Status: 403 Forbidden Size: 49 Bytes Time: 3 ms

```
{
  "quantity": 2,
  "user": "65f14a8d5b1d4a2c4a6c1234",
  "product": "65f14a8d5b1d4a2c4a6c5678"
}
```

```
{
    "success": false,
    "message": "Auth token required"
}
```

```
  'user-agent': 'Thunder Client (https://www.thunderclient.com)',
  'content-type': 'application/json',
  host: 'localhost:7000',
  connection: 'close'
}
```



MongoDB Compass - localhost:27017/shoppyGlobeBackend.carts

localhost:27017 > shoppyGlobeBackend > carts

This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

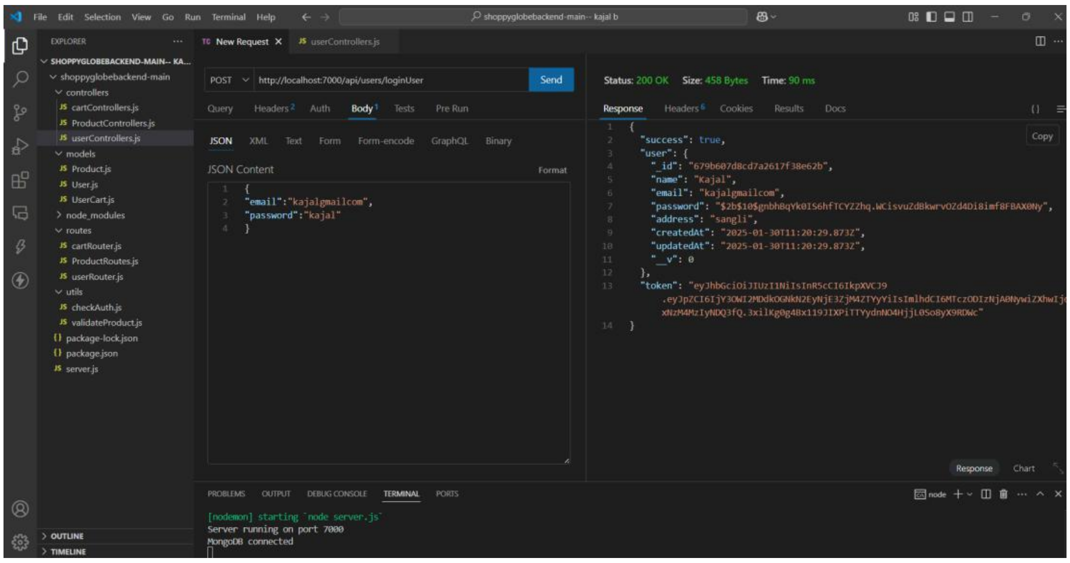
4. POST /login - User Login

Description: Authenticates a user using credentials and generates a token for secure access to protected routes.

Request:

- **Method:** POST
- **URL:** /loginUser

Screenshot:



POST http://localhost:7000/api/users/loginUser

```
{
"email":"kajalgmailcom",
"password":"kajal"
}
```

Status: 200 OK Size: 458 Bytes Time: 90 ms

```
{
  "success": true,
  "user": {
    "_id": "679b607d8cd7a2617f38e62b",
    "name": "Kajal",
    "email": "kajalgmailcom",
    "password": "$2b$10$gnbhBqYk0IS6hfTCYZZhq.WCisvuZdBkwrvOZd4Di8imf8FBAX0Ny",
    "address": "sangli",
    "createdAt": "2025-01-30T11:20:29.873Z",
    "updatedAt": "2025-01-30T11:20:29.873Z",
    "__v": 0
  },
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9
      .eyJpZCI6IjY3OWI2MDdkOGNkN2EyNjE3ZjM4ZTYyYiIsImlhdCI6MTczODIzNjA0NywiZXhwIj
      xNzM4MzIyNDQ3fQ.3xilKg0g4Bx119JIXPiTTYydnNO4HjjL0So8yX9RDWc"
}
```

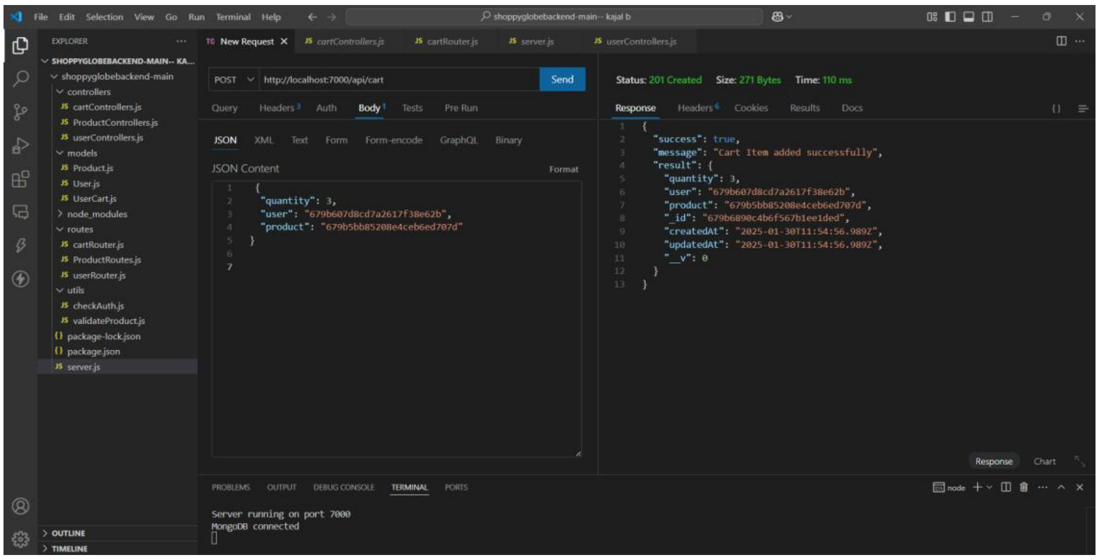
```
[nodemon] starting `node server.js`
Server running on port 7000
MongoDB connected
```

5. POST /cart - Add Item to Cart

Description: Allows a logged-in user to add items to their cart. If the item already exists, the quantity is updated.

Request:

- **Method:** POST
- **URL:** /cart



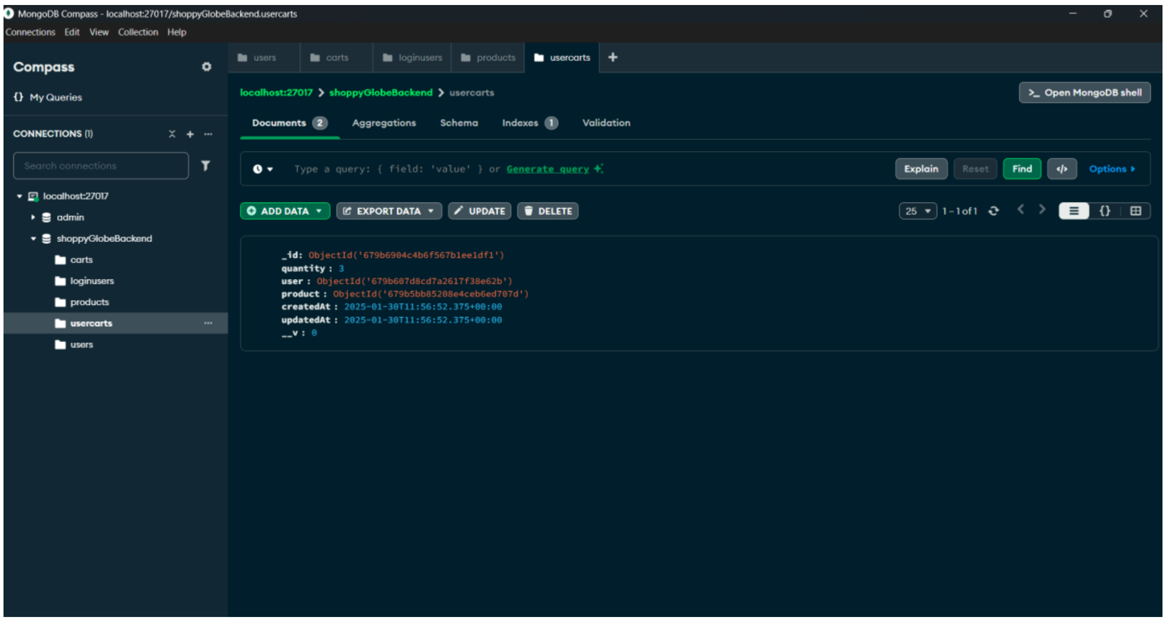
POST http://localhost:7000/api/cart

```
{
  "quantity": 3,
  "user": "679b607d8cd7a2617f38e62b",
  "product": "679b5bb85208e4ceb6ed707d"
}
```

Status: 201 Created   Size: 271 Bytes   Time: 110 ms

```
{
  "success": true,
  "message": "Cart Item added successfully",
  "result": {
    "quantity": 3,
    "user": "679b607d8cd7a2617f38e62b",
    "product": "679b5bb85208e4ceb6ed707d",
    "_id": "679b6890c4b6f567b1ee1ded",
    "createdAt": "2025-01-30T11:54:56.989Z",
    "updatedAt": "2025-01-30T11:54:56.989Z",
    "__v": 0
  }
}
```

```
Server running on port 7000
MongoDB connected
```



localhost:27017 > shoppyGlobeBackend > usercarts

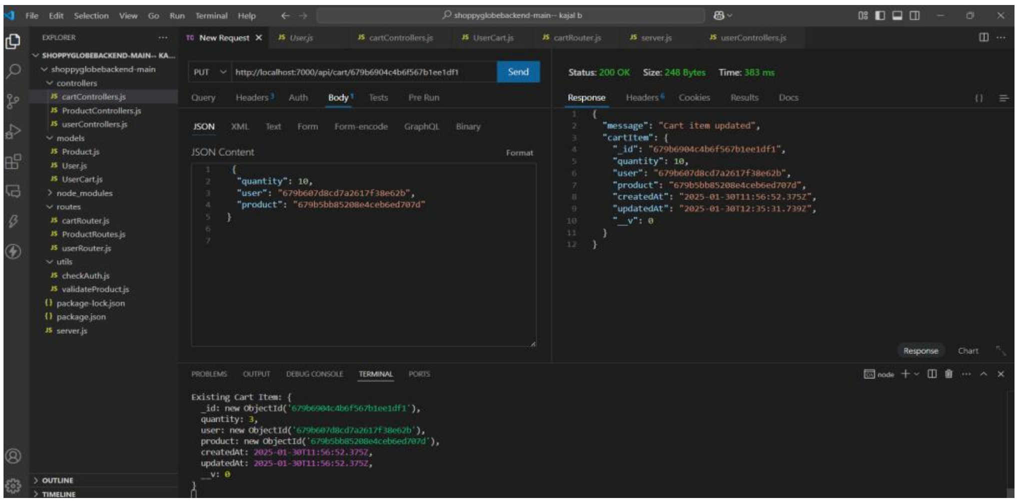
```
_id: ObjectId('679b6904c4b6f567b1ee1df1')
quantity : 3
user : ObjectId('679b607d8cd7a2617f38e62b')
product : ObjectId('679b5bb85208e4ceb6ed707d')
createdAt : 2025-01-30T11:56:52.375+00:00
updatedAt : 2025-01-30T11:56:52.375+00:00
__v : 0
```

6. PUT /cart/:id - Update Cart Item

Description: Modifies the quantity of an item in the cart. This helps users adjust their cart before checkout.

Request:

- **Method:** PUT
- **URL:** /cart/:id



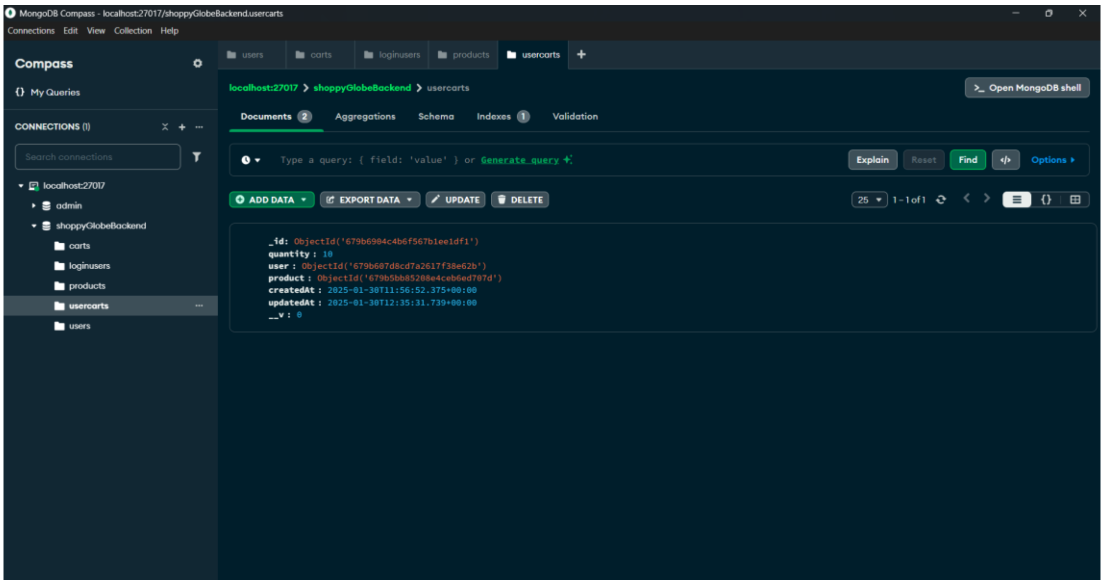
PUT http://localhost:7000/api/cart/679b6904c4b6f567b1ee1df1

Status: 200 OK Size: 248 Bytes Time: 383 ms

```
{
  "quantity": 10,
  "user": "679b607d8cd7a2617f38e62b",
  "product": "679b5bb85208e4ceb6ed707d"
}
```

```
{
  "message": "Cart item updated",
  "cartItem": {
    "_id": "679b6904c4b6f567b1ee1df1",
    "quantity": 10,
    "user": "679b607d8cd7a2617f38e62b",
    "product": "679b5bb85208e4ceb6ed707d",
    "createdAt": "2025-01-30T11:56:52.375Z",
    "updatedAt": "2025-01-30T12:35:31.739Z",
    "__v": 0
  }
}
```

```
Existing Cart Item: {
  _id: new ObjectId('679b6904c4b6f567b1ee1df1'),
  quantity: 3,
  user: new ObjectId('679b607d8cd7a2617f38e62b'),
  product: new ObjectId('679b5bb85208e4ceb6ed707d'),
  createdAt: 2025-01-30T11:56:52.375Z,
  updatedAt: 2025-01-30T11:56:52.375Z,
  __v: 0
}
```

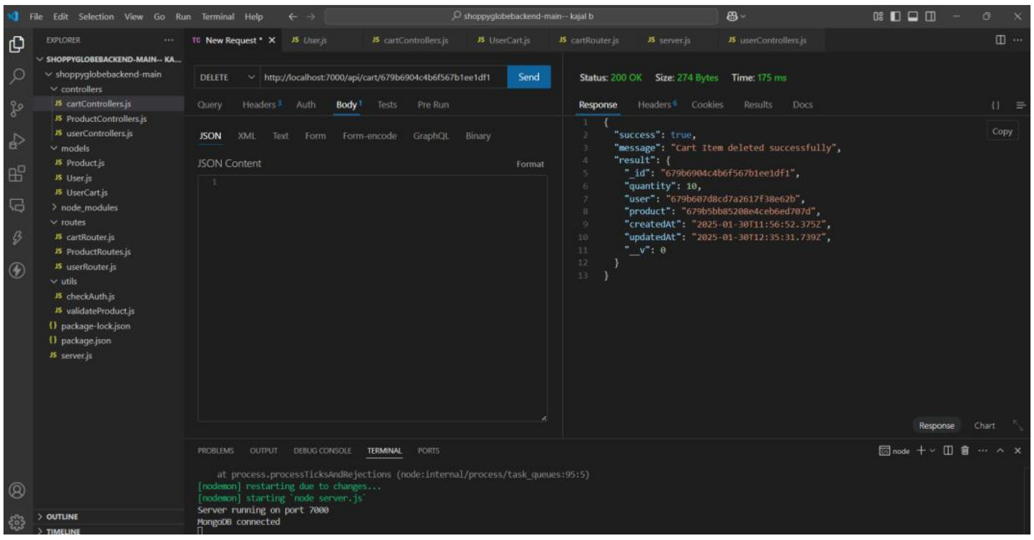


MongoDB Compass - localhost:27017/shoppyGlobeBackend.usercarts

```
_id: ObjectId('679b6904c4b6f567b1ee1df1')
quantity : 10
user : ObjectId('679b607d8cd7a2617f38e62b')
product : ObjectId('679b5bb85208e4ceb6ed707d')
createdAt : 2025-01-30T11:56:52.375+00:00
updatedAt : 2025-01-30T12:35:31.739+00:00
__v : 0
```

7. DELETE /cart/:id - Delete Cart Item

Description: Removes a specific item from the cart. Useful for users who change their purchase decisions.

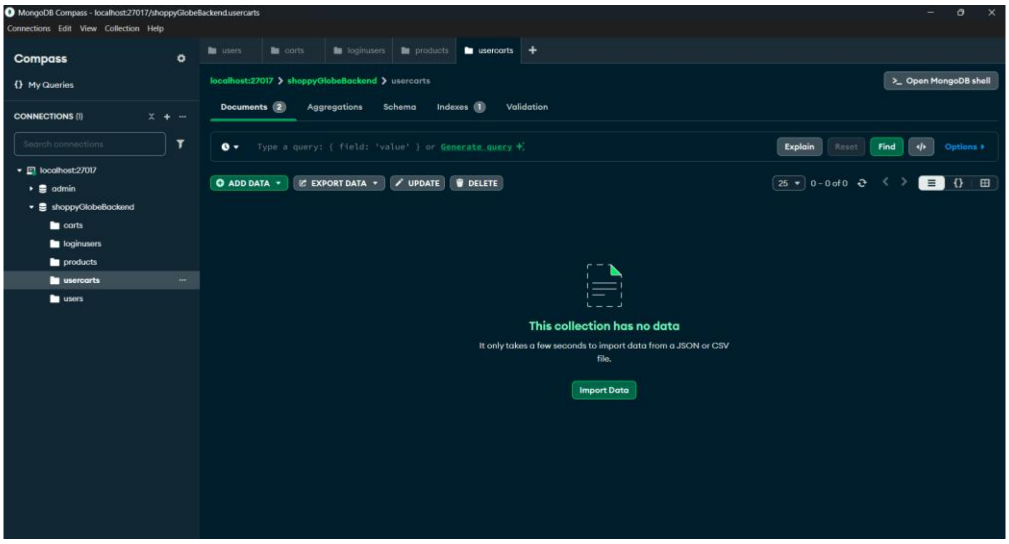


DELETE http://localhost:7000/api/cart/679b6904c4b6f567b1ee1df1

Status: 200 OK Size: 274 Bytes Time: 175 ms

```
{
  "success": true,
  "message": "Cart Item deleted successfully",
  "result": {
    "_id": "679b6904c4b6f567b1ee1df1",
    "quantity": 10,
    "user": "679b607d8cd7a2617f38e62b",
    "product": "679b5bb85208e4ceb6ed707d",
    "createdAt": "2025-01-30T11:56:52.375Z",
    "updatedAt": "2025-01-30T12:35:31.739Z",
    "__v": 0
  }
}
```

```
    at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server running on port 7000
MongoDB connected
```



MongoDB Compass - localhost:27017/shoppyGlobeBackend.usercarts

localhost:27017 > shoppyGlobeBackend > usercarts

This collection has no data

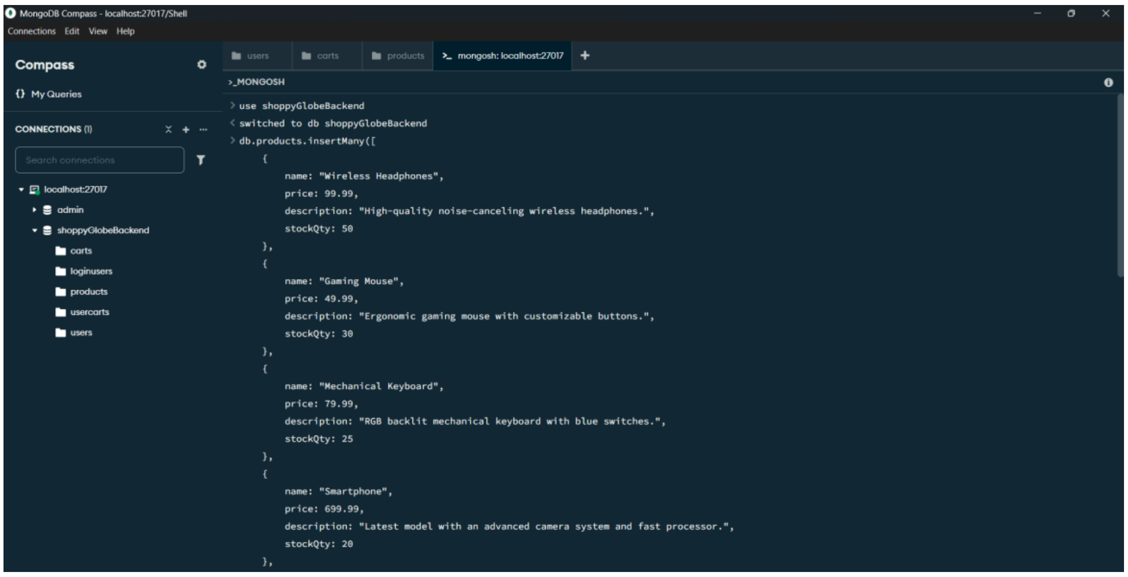
It only takes a few seconds to import data from a JSON or CSV file.

CRUD operations:

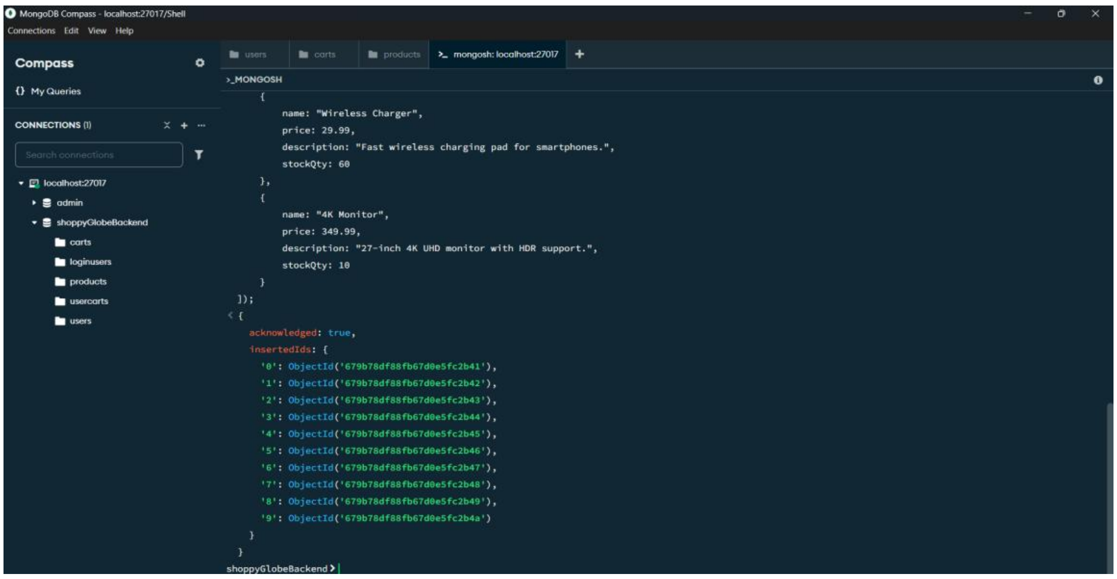
Inserting data to products collection :

By using

db.products.insertMany({ products });



```
> use shoppyGlobeBackend
< switched to db shoppyGlobeBackend
> db.products.insertMany([
      {
          name: "Wireless Headphones",
          price: 99.99,
          description: "High-quality noise-canceling wireless headphones.",
          stockQty: 50
      },
      {
          name: "Gaming Mouse",
          price: 49.99,
          description: "Ergonomic gaming mouse with customizable buttons.",
          stockQty: 30
      },
      {
          name: "Mechanical Keyboard",
          price: 79.99,
          description: "RGB backlit mechanical keyboard with blue switches.",
          stockQty: 25
      },
      {
          name: "Smartphone",
          price: 699.99,
          description: "Latest model with an advanced camera system and fast processor.",
          stockQty: 20
      },
```

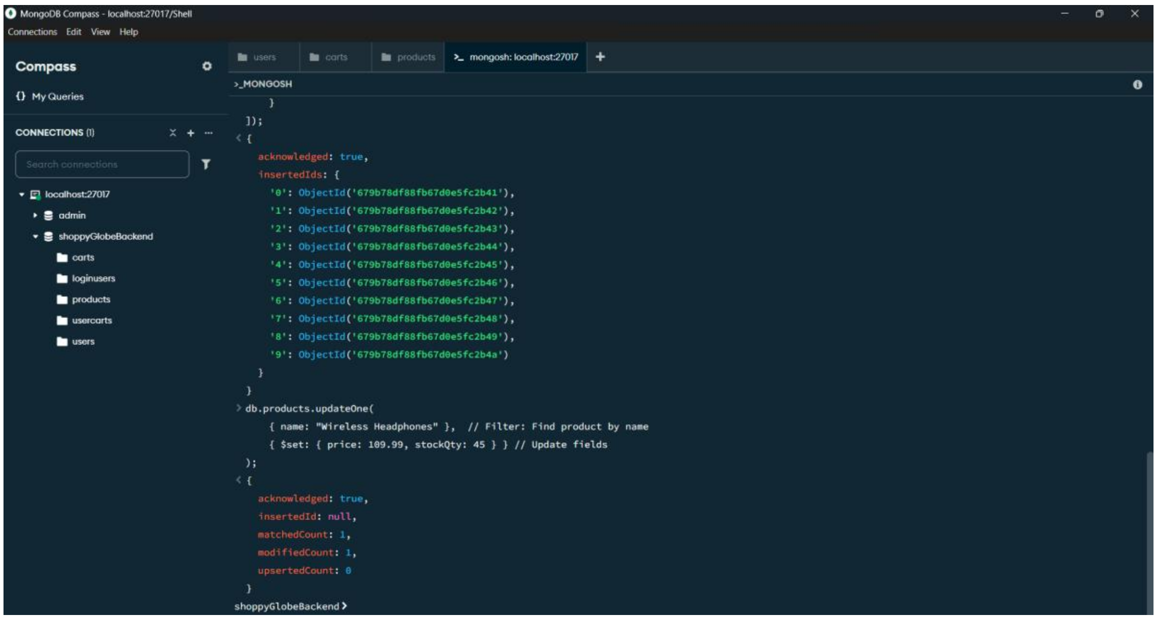


```
      {
          name: "Wireless Charger",
          price: 29.99,
          description: "Fast wireless charging pad for smartphones.",
          stockQty: 60
      },
      {
          name: "4K Monitor",
          price: 349.99,
          description: "27-inch 4K UHD monitor with HDR support.",
          stockQty: 10
      }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('679b78df88fb67d0e5fc2b41'),
      '1': ObjectId('679b78df88fb67d0e5fc2b42'),
      '2': ObjectId('679b78df88fb67d0e5fc2b43'),
      '3': ObjectId('679b78df88fb67d0e5fc2b44'),
      '4': ObjectId('679b78df88fb67d0e5fc2b45'),
      '5': ObjectId('679b78df88fb67d0e5fc2b46'),
      '6': ObjectId('679b78df88fb67d0e5fc2b47'),
      '7': ObjectId('679b78df88fb67d0e5fc2b48'),
      '8': ObjectId('679b78df88fb67d0e5fc2b49'),
      '9': ObjectId('679b78df88fb67d0e5fc2b4a')
    }
  }
shoppyGlobeBackend >
```

Updating products:

Command : db.products.updateOne(

```
   { name: "Wireless Headphones" },  // Filter: Find product by name
   { $set: { price: 109.99, stockQty: 45 } } // Update fields
);
```

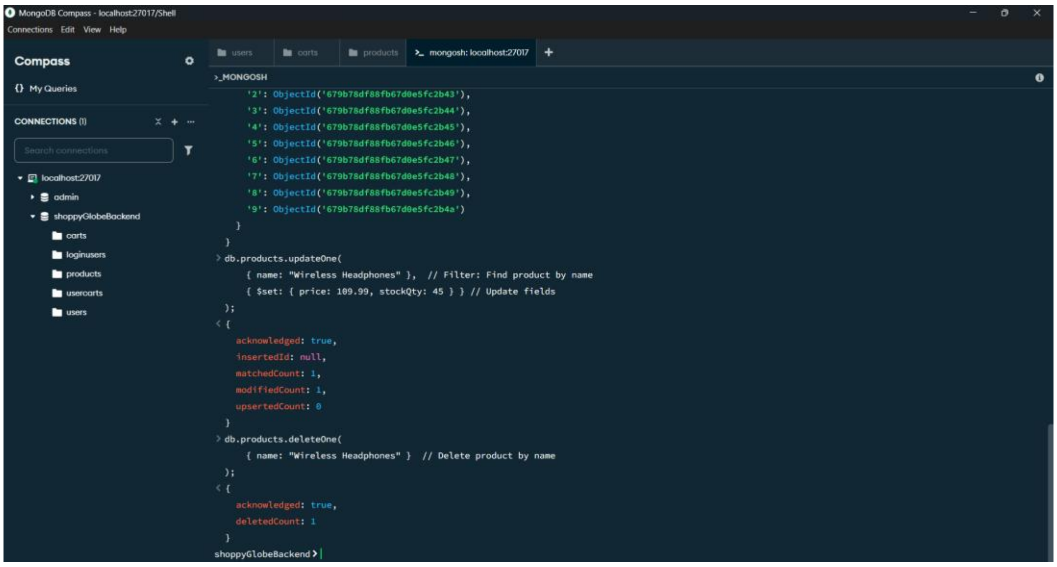


```
    }
  ]);
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('679b78df88fb67d0e5fc2b41'),
      '1': ObjectId('679b78df88fb67d0e5fc2b42'),
      '2': ObjectId('679b78df88fb67d0e5fc2b43'),
      '3': ObjectId('679b78df88fb67d0e5fc2b44'),
      '4': ObjectId('679b78df88fb67d0e5fc2b45'),
      '5': ObjectId('679b78df88fb67d0e5fc2b46'),
      '6': ObjectId('679b78df88fb67d0e5fc2b47'),
      '7': ObjectId('679b78df88fb67d0e5fc2b48'),
      '8': ObjectId('679b78df88fb67d0e5fc2b49'),
      '9': ObjectId('679b78df88fb67d0e5fc2b4a')
    }
  }
> db.products.updateOne(
      { name: "Wireless Headphones" },  // Filter: Find product by name
      { $set: { price: 109.99, stockQty: 45 } } // Update fields
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
shoppyGlobeBackend>
```

Deleting Item from product:

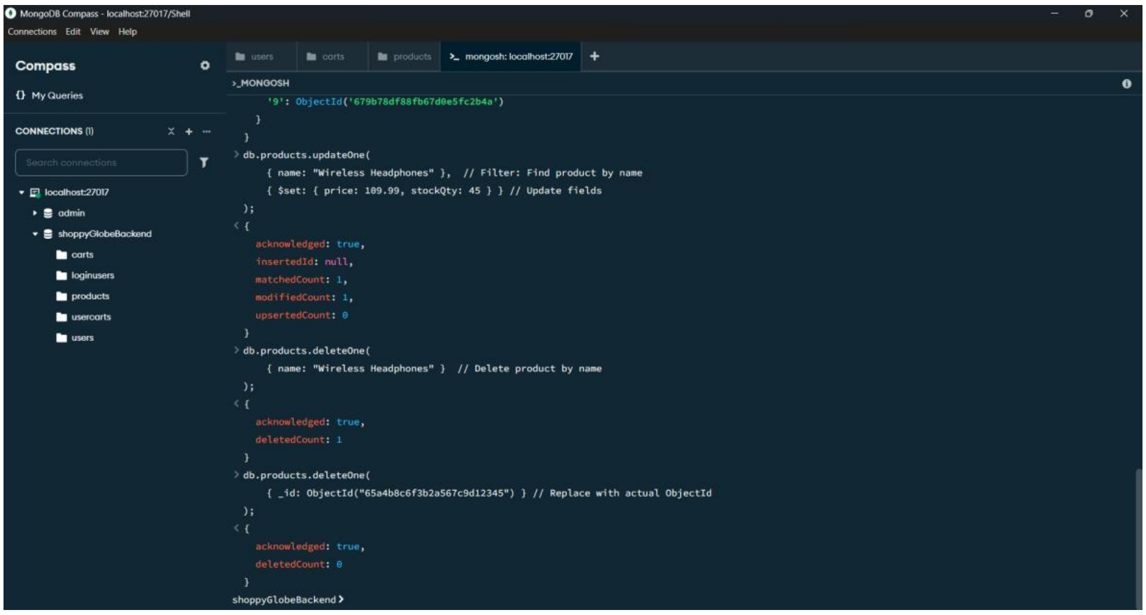
Command: db.products.deleteOne(

```
   { name: "Wireless Headphones" }  // Delete product by name
);
```



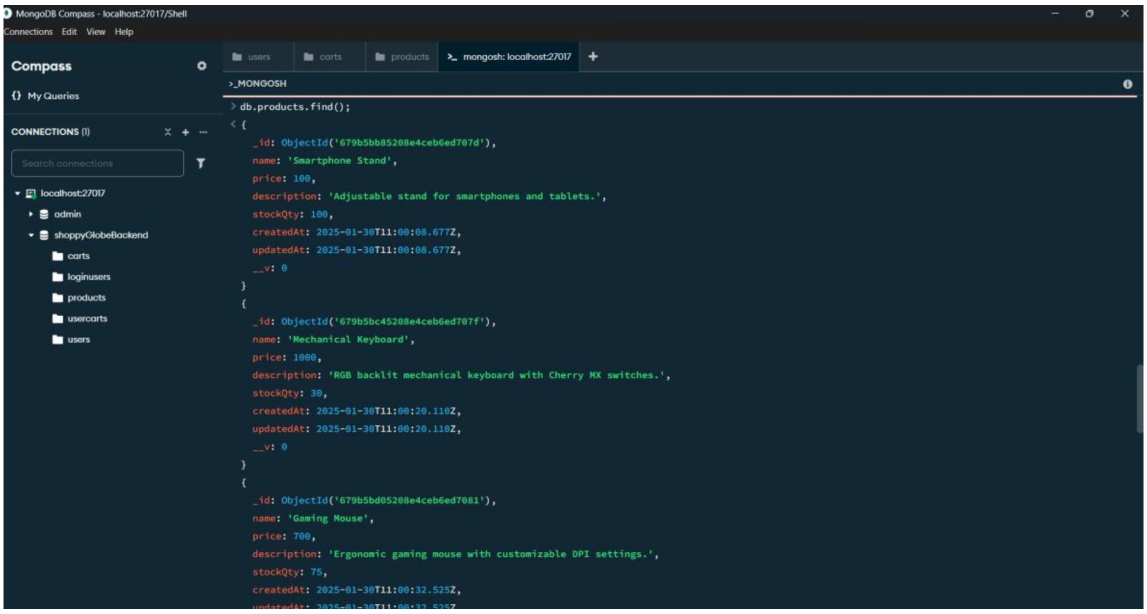
```
      '2': ObjectId('679b78df88fb67d0e5fc2b43'),
      '3': ObjectId('679b78df88fb67d0e5fc2b44'),
      '4': ObjectId('679b78df88fb67d0e5fc2b45'),
      '5': ObjectId('679b78df88fb67d0e5fc2b46'),
      '6': ObjectId('679b78df88fb67d0e5fc2b47'),
      '7': ObjectId('679b78df88fb67d0e5fc2b48'),
      '8': ObjectId('679b78df88fb67d0e5fc2b49'),
      '9': ObjectId('679b78df88fb67d0e5fc2b4a')
    }
  }
> db.products.updateOne(
      { name: "Wireless Headphones" },  // Filter: Find product by name
      { $set: { price: 109.99, stockQty: 45 } } // Update fields
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.products.deleteOne(
      { name: "Wireless Headphones" }  // Delete product by name
  );
< {
    acknowledged: true,
    deletedCount: 1
  }
shoppyGlobeBackend>
```

Note: for unique identification use id.



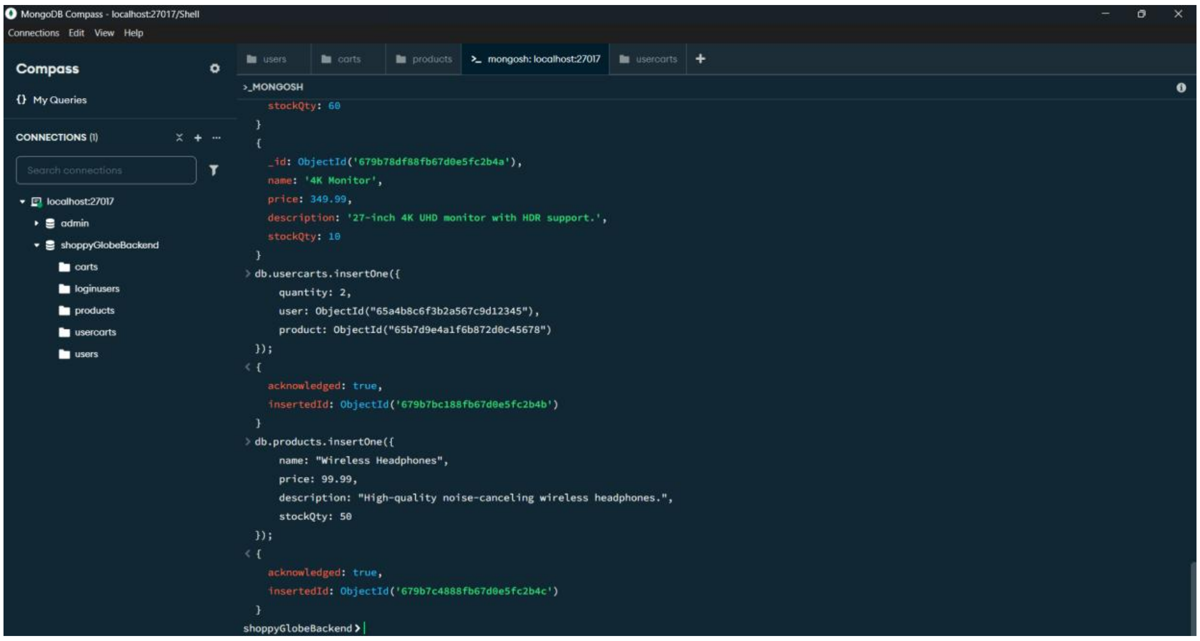
```
      '9': ObjectId('679b78df88fb67d0e5fc2b4a')
    }
  }
> db.products.updateOne(
      { name: "Wireless Headphones" },  // Filter: Find product by name
      { $set: { price: 109.99, stockQty: 45 } } // Update fields
  );
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.products.deleteOne(
      { name: "Wireless Headphones" }  // Delete product by name
  );
< {
    acknowledged: true,
    deletedCount: 1
  }
> db.products.deleteOne(
      { _id: ObjectId("65a4b8c6f3b2a567c9d12345") } // Replace with actual ObjectId
  );
< {
    acknowledged: true,
    deletedCount: 0
  }
shoppyGlobeBackend >
```

Read the data from products collection: Use db.products.find() → for all products.

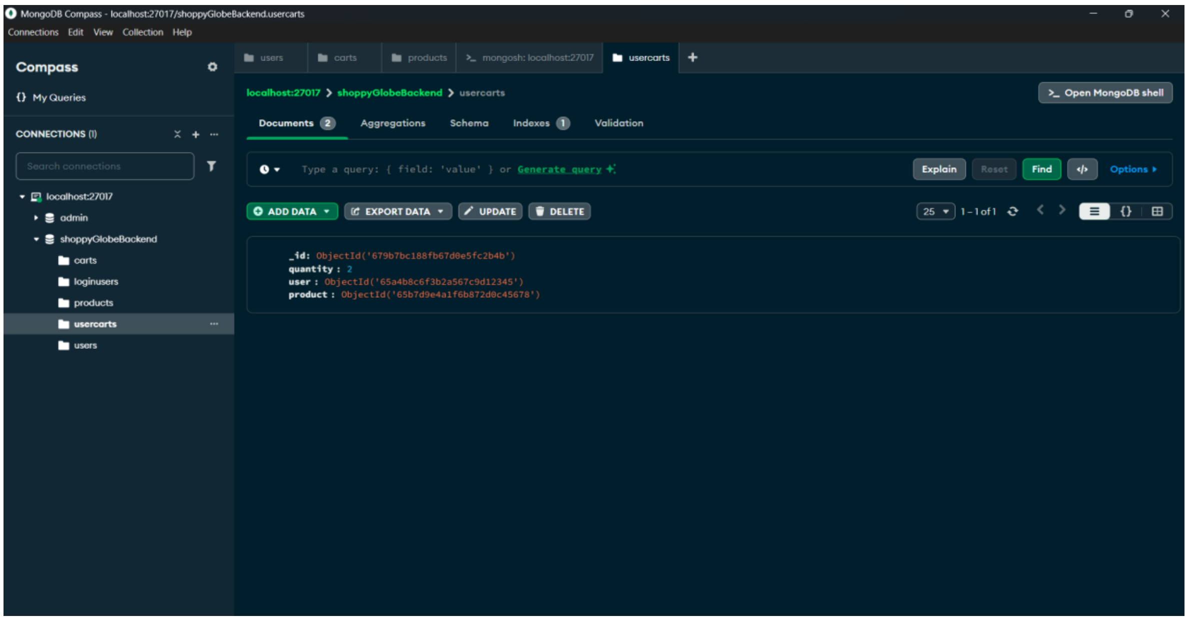


```
> db.products.find();
< {
    _id: ObjectId('679b5bb85208e4ceb6ed707d'),
    name: 'Smartphone Stand',
    price: 100,
    description: 'Adjustable stand for smartphones and tablets.',
    stockQty: 100,
    createdAt: 2025-01-30T11:00:08.677Z,
    updatedAt: 2025-01-30T11:00:08.677Z,
    __v: 0
  }
  {
    _id: ObjectId('679b5bc45208e4ceb6ed707f'),
    name: 'Mechanical Keyboard',
    price: 1000,
    description: 'RGB backlit mechanical keyboard with Cherry MX switches.',
    stockQty: 30,
    createdAt: 2025-01-30T11:00:20.110Z,
    updatedAt: 2025-01-30T11:00:20.110Z,
    __v: 0
  }
  {
    _id: ObjectId('679b5bd05208e4ceb6ed7081'),
    name: 'Gaming Mouse',
    price: 700,
    description: 'Ergonomic gaming mouse with customizable DPI settings.',
    stockQty: 75,
    createdAt: 2025-01-30T11:00:32.525Z,
    updatedAt: 2025-01-30T11:00:32.525Z
```

Adding Item to Cart :

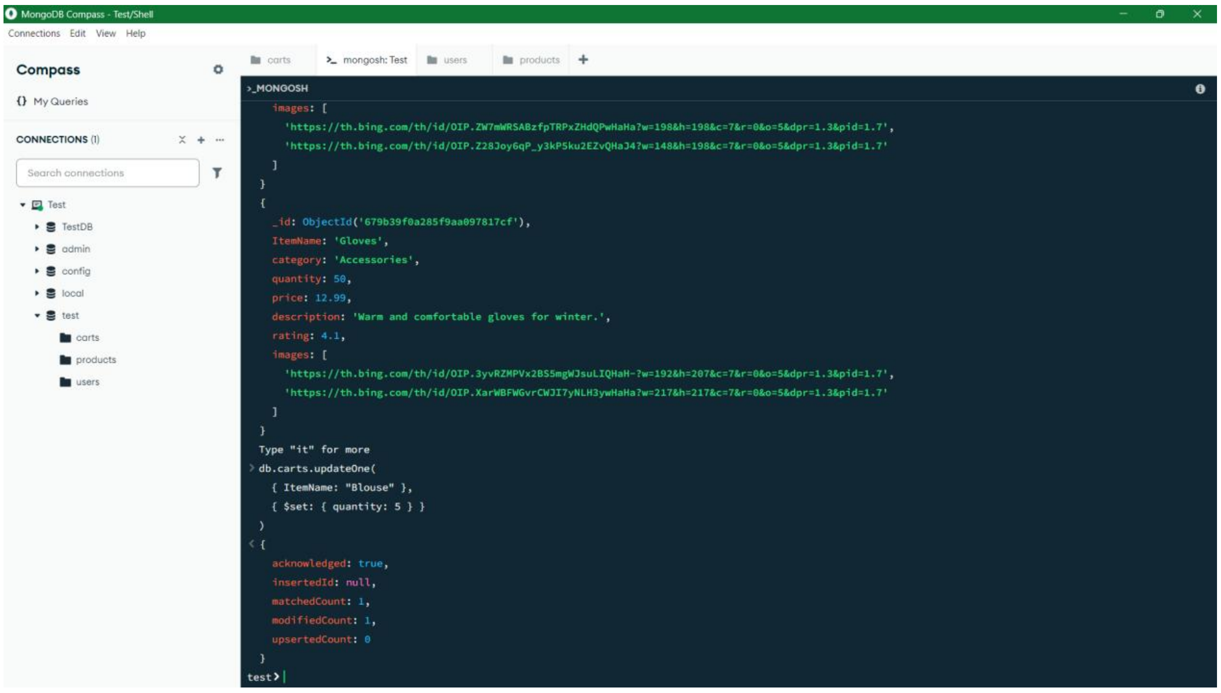


```
    stockQty: 60
  }
  {
    _id: ObjectId('679b78df88fb67d0e5fc2b4a'),
    name: '4K Monitor',
    price: 349.99,
    description: '27-inch 4K UHD monitor with HDR support.',
    stockQty: 10
  }
> db.usercarts.insertOne({
      quantity: 2,
      user: ObjectId("65a4b8c6f3b2a567c9d12345"),
      product: ObjectId("65b7d9e4a1f6b872d0c45678")
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7bc188fb67d0e5fc2b4b')
  }
> db.products.insertOne({
      name: "Wireless Headphones",
      price: 99.99,
      description: "High-quality noise-canceling wireless headphones.",
      stockQty: 50
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7c4888fb67d0e5fc2b4c')
  }
shoppyGlobeBackend >
```



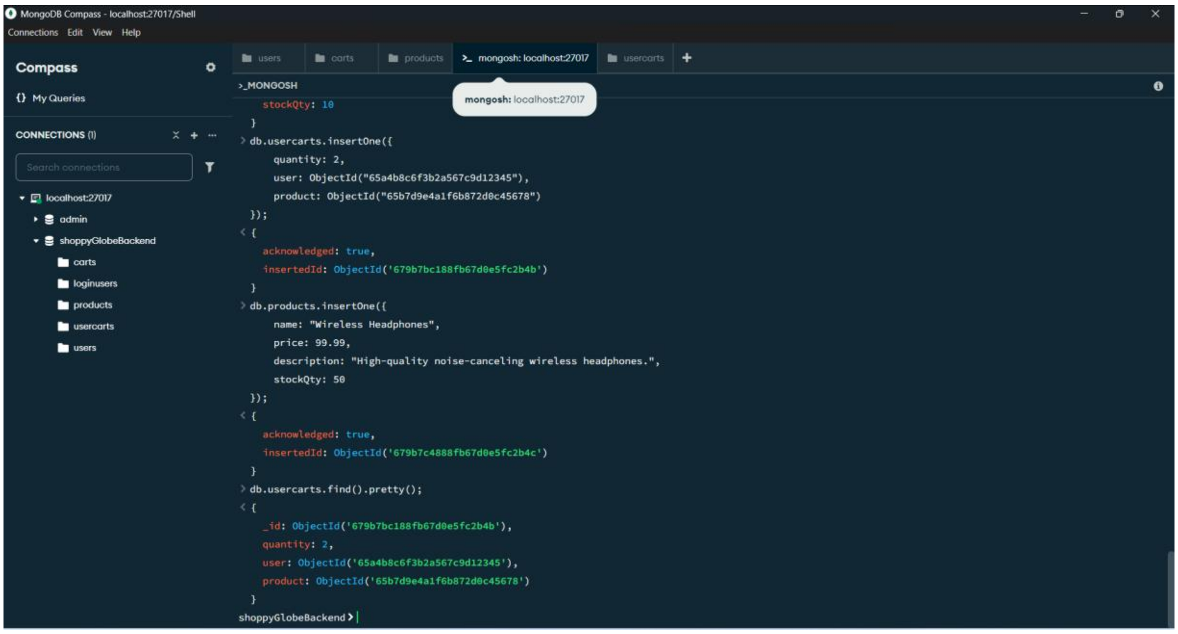
```
_id: ObjectId('679b7bc188fb67d0e5fc2b4b')
quantity : 2
user : ObjectId('65a4b8c6f3b2a567c9d12345')
product : ObjectId('65b7d9e4a1f6b872d0c45678')
```

Update cart:



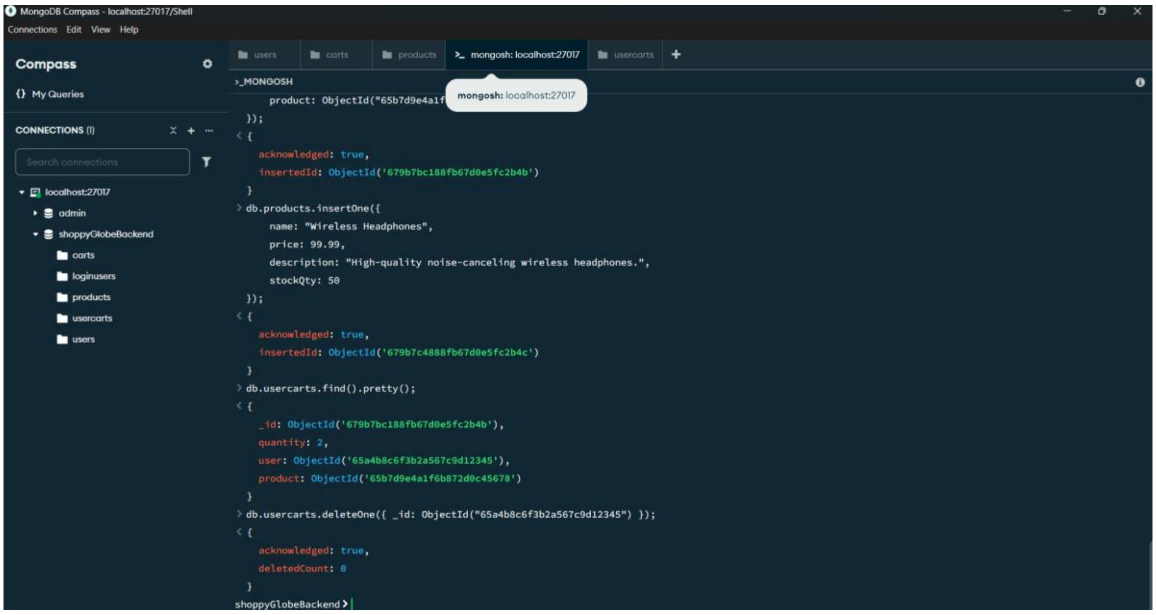
```
    images: [
      'https://th.bing.com/th/id/OIP.ZW7mWRSABzfpTRPxZHdQPwHaHa?w=198&h=198&c=7&r=0&o=5&dpr=1.3&pid=1.7',
      'https://th.bing.com/th/id/OIP.Z28Joy6qP_y3kP5ku2EZvQHaJ4?w=148&h=198&c=7&r=0&o=5&dpr=1.3&pid=1.7'
    ]
  }
  {
    _id: ObjectId('679b39f0a285f9aa097817cf'),
    ItemName: 'Gloves',
    category: 'Accessories',
    quantity: 50,
    price: 12.99,
    description: 'Warm and comfortable gloves for winter.',
    rating: 4.1,
    images: [
      'https://th.bing.com/th/id/OIP.3yvRZMPVx2BS5mgWJsuLIQHaH-?w=192&h=207&c=7&r=0&o=5&dpr=1.3&pid=1.7',
      'https://th.bing.com/th/id/OIP.XarWBFWGvrCWJI7yNLH3ywHaHa?w=217&h=217&c=7&r=0&o=5&dpr=1.3&pid=1.7'
    ]
  }
  Type "it" for more
> db.carts.updateOne(
    { ItemName: "Blouse" },
    { $set: { quantity: 5 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
test>
```

Read items from Cart:



```
    stockQty: 10
  }
> db.usercarts.insertOne({
      quantity: 2,
      user: ObjectId("65a4b8c6f3b2a567c9d12345"),
      product: ObjectId("65b7d9e4a1f6b872d0c45678")
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7bc188fb67d0e5fc2b4b')
  }
> db.products.insertOne({
      name: "Wireless Headphones",
      price: 99.99,
      description: "High-quality noise-canceling wireless headphones.",
      stockQty: 50
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7c4888fb67d0e5fc2b4c')
  }
> db.usercarts.find().pretty();
< {
    _id: ObjectId('679b7bc188fb67d0e5fc2b4b'),
    quantity: 2,
    user: ObjectId('65a4b8c6f3b2a567c9d12345'),
    product: ObjectId('65b7d9e4a1f6b872d0c45678')
  }
shoppyGlobeBackend>
```

Delete item from cart:



```
>_MONGOSH
    product: ObjectId("65b7d9e4a1f
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7bc188fb67d0e5fc2b4b')
  }
> db.products.insertOne({
    name: "Wireless Headphones",
    price: 99.99,
    description: "High-quality noise-canceling wireless headphones.",
    stockQty: 50
  });
< {
    acknowledged: true,
    insertedId: ObjectId('679b7c4888fb67d0e5fc2b4c')
  }
> db.usercarts.find().pretty();
< {
    _id: ObjectId('679b7bc188fb67d0e5fc2b4b'),
    quantity: 2,
    user: ObjectId('65a4b8c6f3b2a567c9d12345'),
    product: ObjectId('65b7d9e4a1f6b872d0c45678')
  }
> db.usercarts.deleteOne({ _id: ObjectId("65a4b8c6f3b2a567c9d12345") });
< {
    acknowledged: true,
    deletedCount: 0
  }
shoppyGlobeBackend >
```