CODEBOOSTERS TECH - NODE JS & GRAPHQL PROJECT LIST

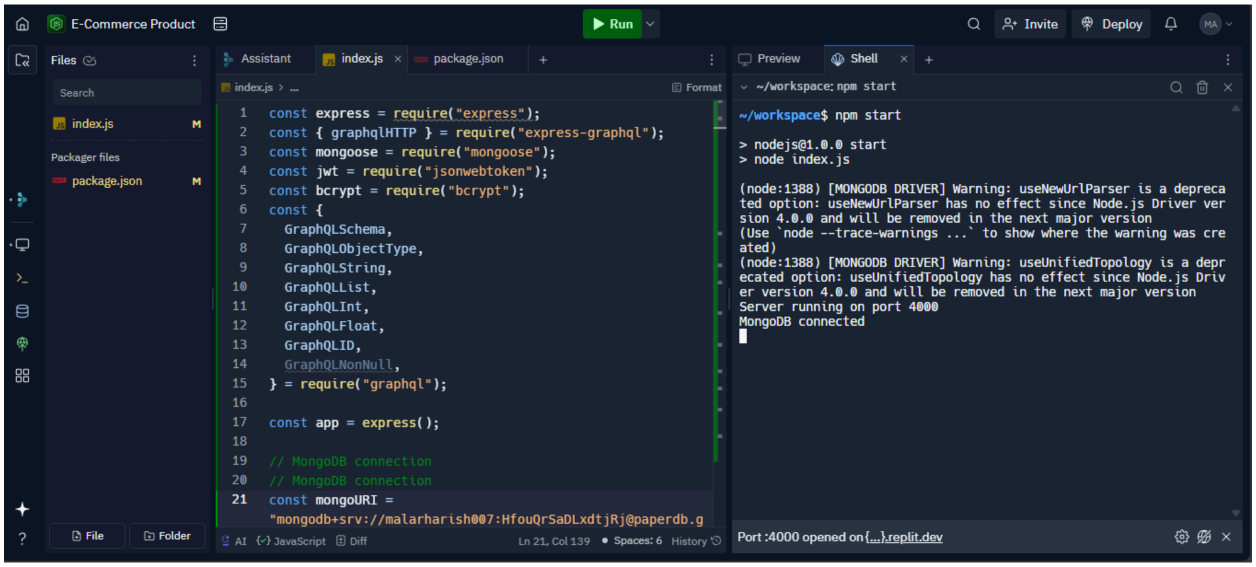
1. E-Commerce Product Catalog API with GraphQL
   Project Objective:
   The goal of this project is to design and implement a robust product catalog API for an e
   commerce platform using GraphQL and Node.js. This project aims to provide a flexible and
   scalable API to manage and interact with product data, enabling customers to browse
   products, apply filters, and sort the catalog dynamically.
   Detailed Objectives:
2. Set up a GraphQL Server: Use Node.js with Express.js to set up a GraphQL server.
   Students will learn how to configure and implement the server to handle various queries
   and mutations.
3. Design GraphQL Schema: Students will design a schema that defines product types
   (e.g., name, description, price, category), and allow filtering and sorting. The schema will
   also define mutations for adding, updating, and deleting products.
4. Pagination and Filtering: Implement pagination to allow users to request a subset of
   products and avoid overwhelming the system with large queries. Students will create
   f
   iltering options for price range, category, and other attributes like rating or brand.

5. Integration with MongoDB/MySQL: Students will learn how to integrate a database
(MongoDB for flexibility or MySQL for relational data) to persist product data. This will
include designing tables/collections for products, categories, and other related entities.
6. GraphQL Queries and Mutations:
Queries: Implement queries for searching, listing, and viewing product details.
Students will learn how to handle complex queries with sorting and filtering.
Mutations: Implement mutations for adding, updating, and deleting products from
the catalog. This will involve interacting with the database and returning appropriate
success or failure responses.
7. Error Handling: Implement robust error handling in GraphQL to ensure that users
receive meaningful error messages when something goes wrong (e.g., when a product

   cannot be found or deleted).
8. Authentication and Authorization (Optional): Integrate user authentication (e.g.,
admin users who can add or modify products) to control access to certain mutations,
ensuring only authorized users can make changes to the catalog.

OUTPUT

1. MongoDB Connected



```javascript
const express = require("express");
const { graphqlHTTP } = require("express-graphql");
const mongoose = require("mongoose");
const jwt = require("jsonwebtoken");
const bcrypt = require("bcrypt");
const {
  GraphQLSchema,
  GraphQLObjectType,
  GraphQLString,
  GraphQLList,
  GraphQLInt,
  GraphQLFloat,
  GraphQLID,
  GraphQLNonNull,
} = require("graphql");

const app = express();

// MongoDB connection
// MongoDB connection
const mongoURI =
"mongodb+srv://malarharish007:HfouQrSaDLxdtjRj@paperdb.g
```

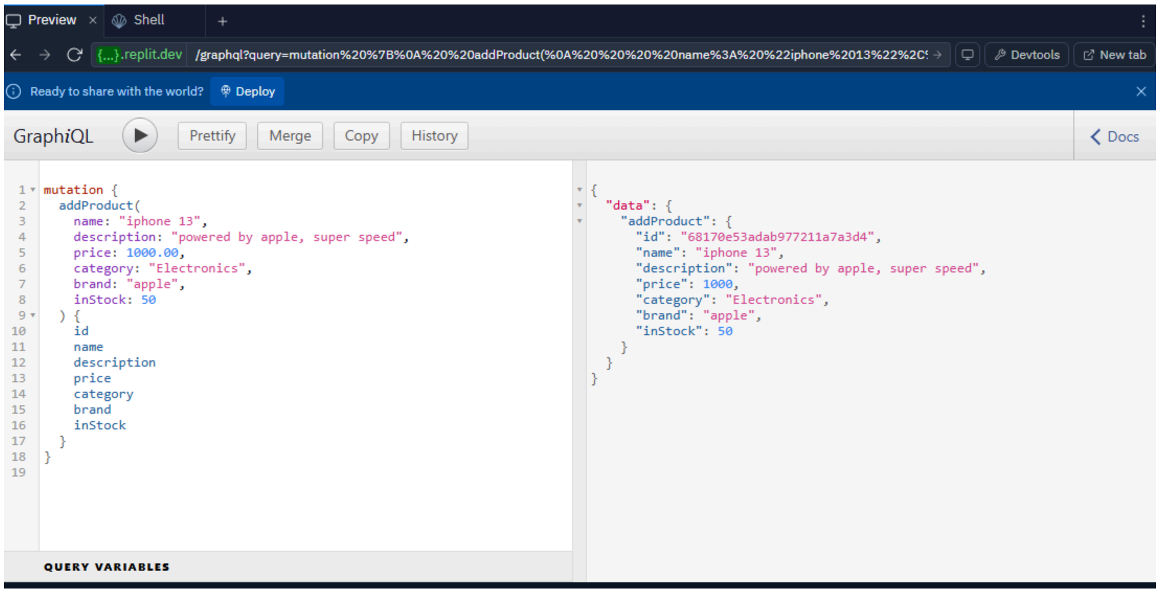
```
~/workspace$ npm start

> nodejs@1.0.0 start
> node index.js

(node:1388) [MONGODB DRIVER] Warning: useNewUrlParser is a depreca
ted option: useNewUrlParser has no effect since Node.js Driver ver
sion 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was cre
ated)
(node:1388) [MONGODB DRIVER] Warning: useUnifiedTopology is a depr
ecated option: useUnifiedTopology has no effect since Node.js Driv
er version 4.0.0 and will be removed in the next major version
Server running on port 4000
MongoDB connected
```

Port :4000 opened on {...}.replit.dev

2. add Product

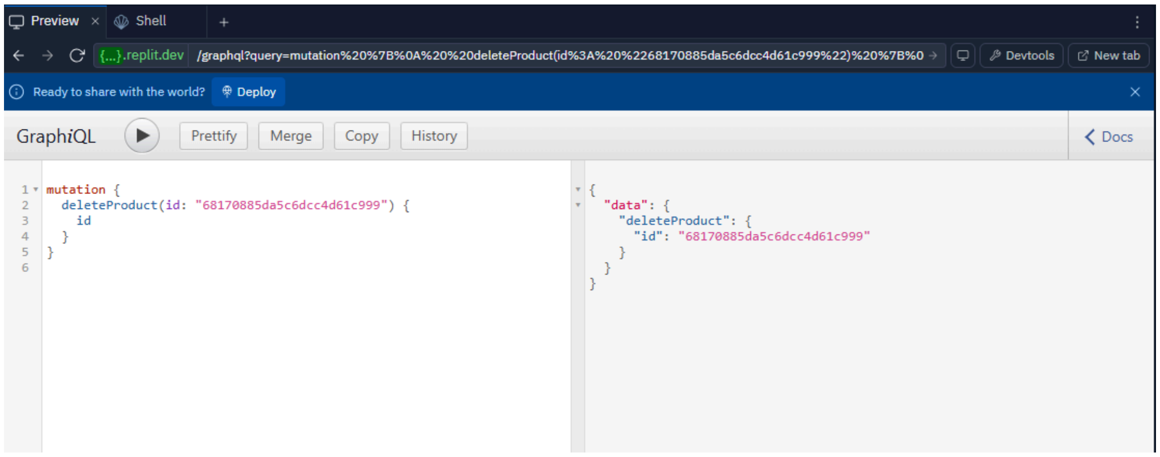


{...}.replit.dev /graphql?query=mutation%20%7B%0A%20%20addProduct(%0A%20%20%20%20name%3A%20%22iphone%2013%22%2C

```graphql
mutation {
  addProduct(
    name: "iphone 13",
    description: "powered by apple, super speed",
    price: 1000.00,
    category: "Electronics",
    brand: "apple",
    inStock: 50
  ) {
    id
    name
    description
    price
    category
    brand
    inStock
  }
}
```

```json
{
  "data": {
    "addProduct": {
      "id": "68170e53adab977211a7a3d4",
      "name": "iphone 13",
      "description": "powered by apple, super speed",
      "price": 1000,
      "category": "Electronics",
      "brand": "apple",
      "inStock": 50
    }
  }
}
```

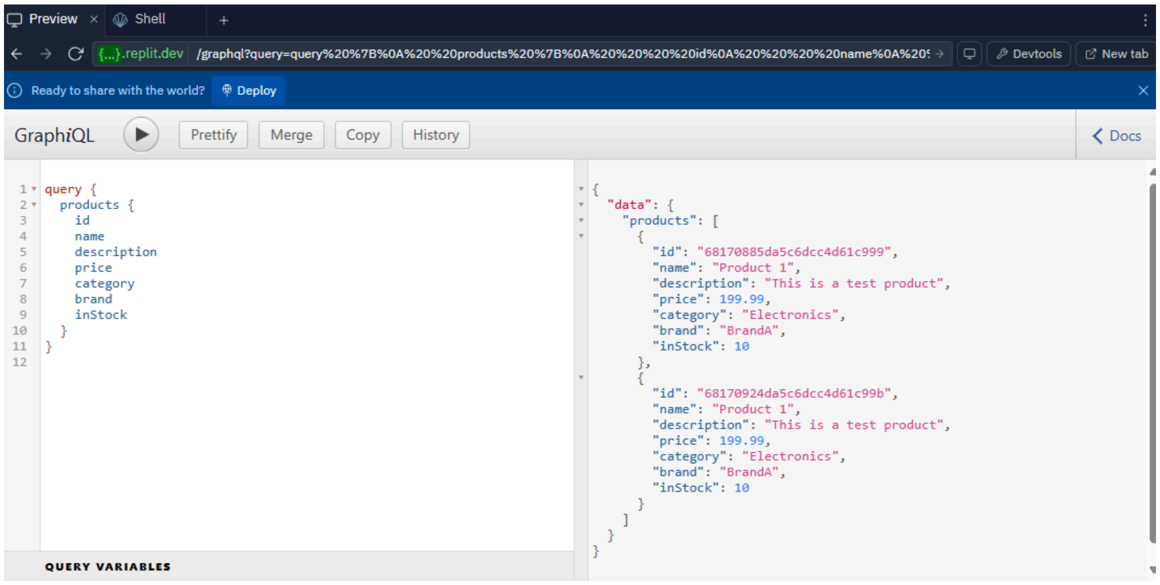
3. Delete a product by id


Preview
Shell
{...}.replit.dev
/graphql?query=mutation%20%7B%0A%20%20deleteProduct(id%3A%20%2268170885da5c6dcc4d61c999%22)%20%7B%0
Devtools
New tab
Ready to share with the world?
Deploy
GraphiQL
Prettify
Merge
Copy
History
Docs

```graphql
mutation {
  deleteProduct(id: "68170885da5c6dcc4d61c999") {
    id
  }
}
```

```json
{
  "data": {
    "deleteProduct": {
      "id": "68170885da5c6dcc4d61c999"
    }
  }
}
```

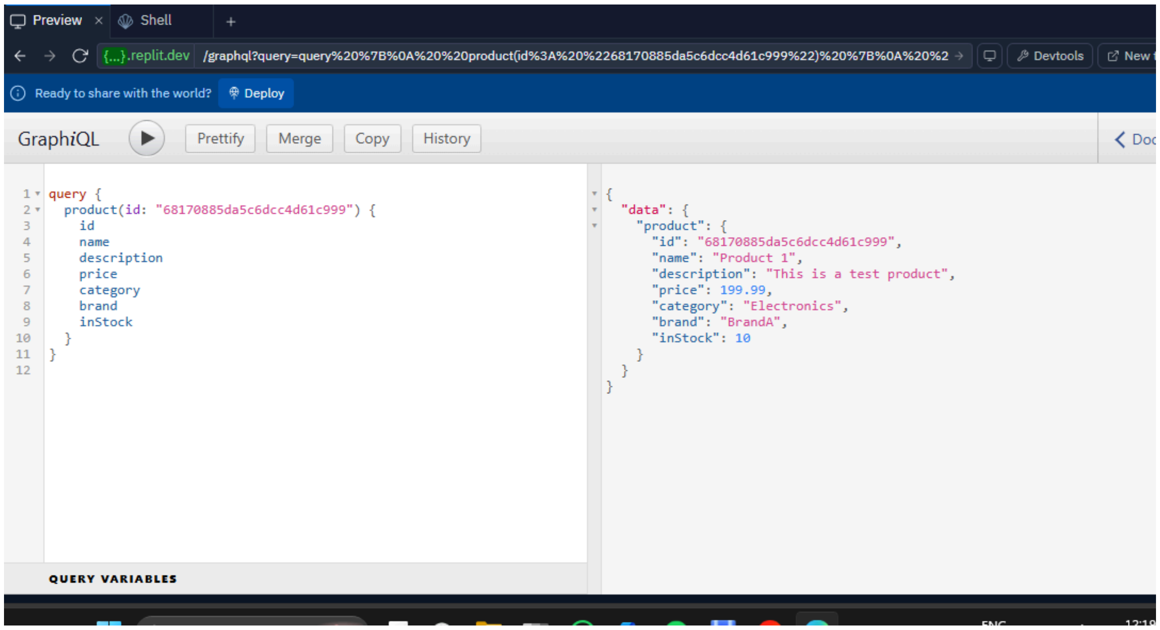
4. get all products


Preview
Shell
{...}.replit.dev
/graphql?query=query%20%7B%0A%20%20products%20%7B%0A%20%20%20%20id%0A%20%20%20%20name%0A%20
Devtools
New tab
Ready to share with the world?
Deploy
GraphiQL
Prettify
Merge
Copy
History
Docs
QUERY VARIABLES

```graphql
query {
  products {
    id
    name
    description
    price
    category
    brand
    inStock
  }
}
```

```json
{
  "data": {
    "products": [
      {
        "id": "68170885da5c6dcc4d61c999",
        "name": "Product 1",
        "description": "This is a test product",
        "price": 199.99,
        "category": "Electronics",
        "brand": "BrandA",
        "inStock": 10
      },
      {
        "id": "68170924da5c6dcc4d61c99b",
        "name": "Product 1",
        "description": "This is a test product",
        "price": 199.99,
        "category": "Electronics",
        "brand": "BrandA",
        "inStock": 10
      }
    ]
  }
}
```

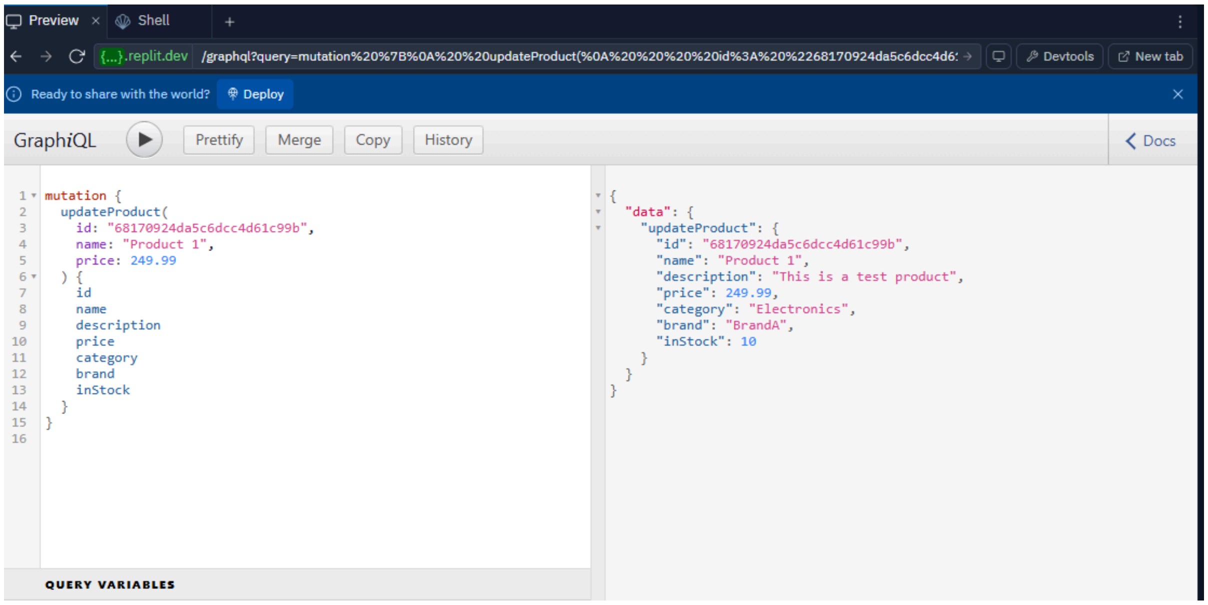
5. get product by id


Preview
Shell
{...}.replit.dev /graphql?query=query%20%7B%0A%20%20product(id%3A%20%2268170885da5c6dcc4d61c999%22)%20%7B%0A%20%2
Devtools
Ready to share with the world?
Deploy
GraphiQL
Prettify
Merge
Copy
History
QUERY VARIABLES

```
query {
  product(id: "68170885da5c6dcc4d61c999") {
    id
    name
    description
    price
    category
    brand
    inStock
  }
}
```

```
{
  "data": {
    "product": {
      "id": "68170885da5c6dcc4d61c999",
      "name": "Product 1",
      "description": "This is a test product",
      "price": 199.99,
      "category": "Electronics",
      "brand": "BrandA",
      "inStock": 10
    }
  }
}
```

6. update product by id


Preview
Shell
{...}.replit.dev /graphql?query=mutation%20%7B%0A%20%20updateProduct(%0A%20%20%20%20id%3A%20%2268170924da5c6dcc4d6
Devtools
New tab
Ready to share with the world?
Deploy
GraphiQL
Prettify
Merge
Copy
History
Docs
QUERY VARIABLES

```
mutation {
  updateProduct(
    id: "68170924da5c6dcc4d61c99b",
    name: "Product 1",
    price: 249.99
  ) {
    id
    name
    description
    price
    category
    brand
    inStock
  }
}
```

```
{
  "data": {
    "updateProduct": {
      "id": "68170924da5c6dcc4d61c99b",
      "name": "Product 1",
      "description": "This is a test product",
      "price": 249.99,
      "category": "Electronics",
      "brand": "BrandA",
      "inStock": 10
    }
  }
}
```