FSD Lab Program 7 (Rest API in Postman)

Title : Create Rest API and GraphQL the allows the user to access and manage the products.

Problem description:. Set up a basic server that responds with "Hello, Express!" at the root endpoint (GET /). Implement endpoints for a Product resource: GET : Returns a list of products. POST Adds a new product. GET /:id: Returns details of a specific product. PUT /:id: Updates an existing product. DELETE /:id: Deletes a product. Add middleware to log requests to the console. Use express.json() to parse incoming JSON payloads.

Method: Install Express (npm install express). Use apollo server for GraphQL API.
Write endpoints and test the API using Postman.

Step 1: Setup Project

```
mkdir product-api
cd product-api
npm init -y
npm install express apollo-server-express graphql cors
code . (It will open the folder in VS Code)
Create a New File server.js in product-api folder
```

Step 2: server.js

```
import express from "express";
import cors from "cors";
import products from "./product.js"; //  Import products array from file product.js

const app = express();
app.use(cors());
app.use(express.json());

// Middleware to log requests
app.use((req, res, next) => {
  console.log(req.method, req.url);
  next();
});

// REST API
app.get("/", (req, res) => res.send("Hello, Express!"));

app.get("/products", (req, res) => res.json(products));

app.get("/products/:id", (req, res) => {
  const p = products.find(x => x.id == req.params.id);
  p ? res.json(p) : res.status(404).send("Not Found");
});

app.post("/addproducts", (req, res) => {
  const newP = { id: Date.now(), ...req.body };
  products.push(newP);
  res.json(newP);
});
```

```
app.put("/updateproducts/:id", (req, res) => {
  const i = products.findIndex(p => p.id == req.params.id);
  if (i >= 0) products[i] = { ...products[i], ...req.body };
  res.json(products[i]);
});

app.delete("/deleteproducts/:id", (req, res) => {
 const index = products.findIndex(p => p.id == req.params.id);
 if (index >= 0) {
   products.splice(index, 1); // removes 1 element from that index
   res.send("Deleted successfully");
 } else {
   res.status(404).send("Product not found");
 }
});

// Start the server
const PORT = 5000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

Step 3: Change package.json

```
{
  "name": "expresjs",
  "version": "1.0.0",
  "main": "server.js",
  "type": "module",
  "scripts": {
    "start": "node server.js"
  },
  "dependencies": {
    "express": "^5.1.0"
  }
}
```

Step 4: create new file product.js in product-api folder

```
let products = [
  { id: 1, name: "Laptop", price: 1000 },
  { id: 2, name: "Phone", price: 500 }
];

export default products;
```

Step 5: Run Server

```
node server.js
```

O/P: Server running on port 5000

Step 6: Install Postman

1. Visit the Postman website and download the application for your operating system (Windows, Mac, or Linux).
2. Install Postman by following the prompts

Next click on new there select http after selecting http here you can see the all apis like get, get/id, put/id, post, delete/id…etc

Step 7 - How to Use Product REST API in Postman

1. Get all products

- **URL:** http://localhost:5000/products
- In Postman:
  - Choose **GET** method.
  - Enter the URL.
  - Click **Send**.
- You will see all products in the response.

---

2. Add a new product

- **URL:** http://localhost:5000/addproducts
- In Postman:
  - Choose **POST** method.
  - Go to **Body → raw → JSON** (select JSON from the dropdown).
  - Enter the new product data, for example:

```
{
  "id": 3,
  "name": "Tablet",
  "price": 700
}
```

- Click **Send**.
- You will see the new product returned in the response.

---

3. Update a product

- **URL:** http://localhost:5000/updateproducts/2
- In Postman:
  - Choose **PUT** method.
  - Go to **Body → raw → JSON →** select **JSON**.
  - Enter updated product data, for example:

```
{
  "name": "iPhone 16",
  "price": 14000
}
```

- Click **Send**.
- You will see the updated product returned.

---

4. Get a product by ID

- **URL:** http://localhost:5000/products/2
- In Postman:
  - Choose **GET** method.
  - Enter the URL with the product ID.
  - Click **Send**.
- You will see the product with that ID.

---

5. Delete a product

- **URL:** http://localhost:5000/deleteproducts/1
- In Postman:
  - Choose **DELETE** method.
  - Enter the URL with the product ID.
  - Click **Send**.
- The product will be deleted and you will see a confirmation message.