**ASSIGNMENT**

The goal of this project is to create a robust and flexible API for managing product data. The API will provide various endpoints to perform operations such as retrieving all products, adding a new product, updating an existing product, and deleting a product. These endpoints adhere to standard RESTful principles, offering a clear and intuitive interface for interacting with the product data.

The database used here is the in-memory data store i.e., **products.json file**.

**Steps to run the server: -**

* Type **cd node\_typescript assignment**
* Before running the API, type **npm install** to install all the required package
* Then type **npm run dev** to run the application

**Endpoints:**

1. **GET /api/products:**

* Retrieve a list of all products - **http://localhost:5000/api/products**
* Provides a comprehensive view of the available products in the system.

1. **GET /api/products/:id:**

* Retrieve a specific product by its unique identifier (ID) - **http://localhost:5000/api/products/<product\_id>**

NOTE – Replace the <product\_id> with the actual id (like 1, 2, etc)

* Allows clients to access detailed information about a particular product.

1. **POST /api/products:**

* Add a new product to the system - **http://localhost:5000/api/products**
* Requires clients to submit product details such as name, price, and category for creation.
* Automatically assigns a unique identifier to the new product.

Example: -

{

    "name": "Book",

    "price": 699.99,

    "category": "Books"

}

1. **PUT /api/products/:id:**

* Update an existing product based on its ID - **http://localhost:5000/api/products/<product\_id>**

NOTE – Replace the <product\_id> with the actual id (like 1, 2, etc)

* Enables clients to modify product information, such as name, price, or category.

Example: -

{

  "name": "Updated Book",

    "price": 699.99,

    "category": "Updated Books"

}

1. **DELETE /api/products/:id:**

* Delete a product using its unique identifier (ID) - **http://localhost:5000/api/products/<product\_id>**
* NOTE – Replace the <product\_id> with the actual id (like 1, 2, etc)
* Removes the specified product from the system.

**Features:**

* **Validation**:
* Ensure that the data provided during product creation and updates adheres to predefined validation rules.
* Handle validation errors gracefully and provide meaningful error messages.
* **Auto-incremented IDs:**
* Automatically assign unique identifiers to newly created products, eliminating the need for clients to specify an ID during creation.
* **Error Handling:**
* Implement robust error handling to gracefully manage unexpected scenarios, providing informative responses to clients.
* **Middleware Logging:**
* Implement middleware to log the timestamp, HTTP method, and requested URL for every incoming request to the API.

This project aims to deliver a reliable and user-friendly API for managing product data, offering a seamless experience for developers interacting with the system. The API design adheres to industry best practices and provides comprehensive functionality for handling various CRUD (Create, Read, Update, Delete) operations on product entities.

**Steps to do unit testing:**

* Stop the running server
* Type **npm test**