# Lesson 06 Demo 02

# Implementing Framework Migration Using Generative AI

**Objective:** To migrate the login functionality of a social media platform from Angular to Flask using a generative AI tool

**Tools required:** Visual Studio, Python, Node JS, Post man plugin and GitHub Copilot

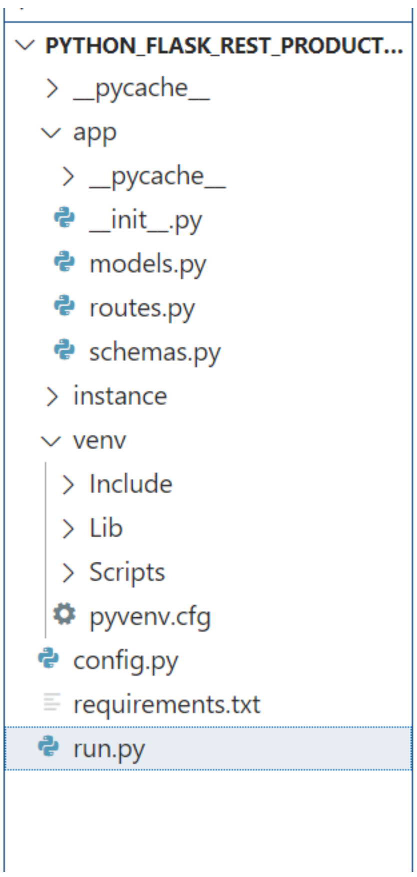
**Prerequisites:** installed python and node js in your machine

Steps to be followed:
1. Implement a Rest API using python with flask.
2. Migrate the Python Flask to express JS

## Step 1: Implement a Rest API using python with flask

1.1 Download the sample folder Python_Flask_Rest_Product_API and open this folder in VS code editor.

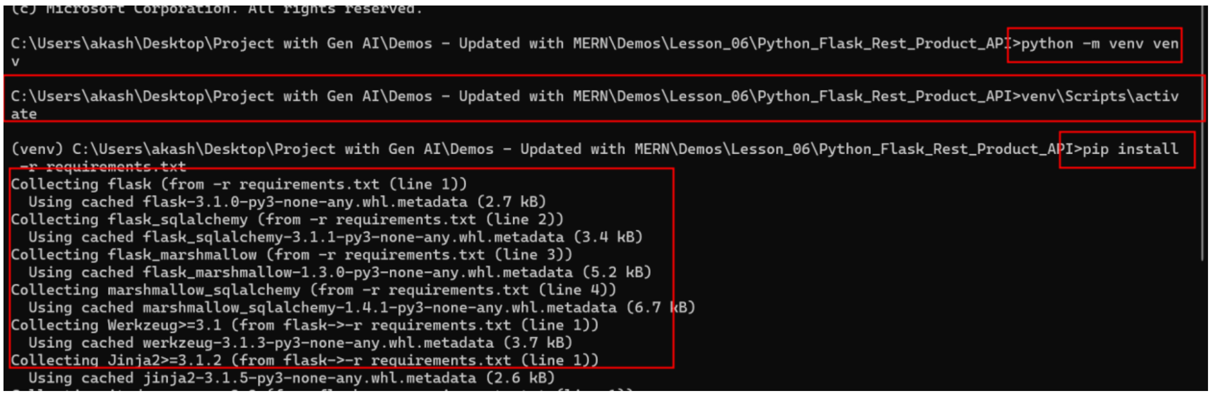
1.2 The project structure in VS Code



```
PYTHON_FLASK_REST_PRODUCT...
  __pycache__
  app
    __pycache__
    __init__.py
    models.py
    routes.py
    schemas.py
  instance
  venv
    Include
    Lib
    Scripts
    pyvenv.cfg
  config.py
  requirements.txt
  run.py
```

1.3 Installed required dependencies and create virtual environment

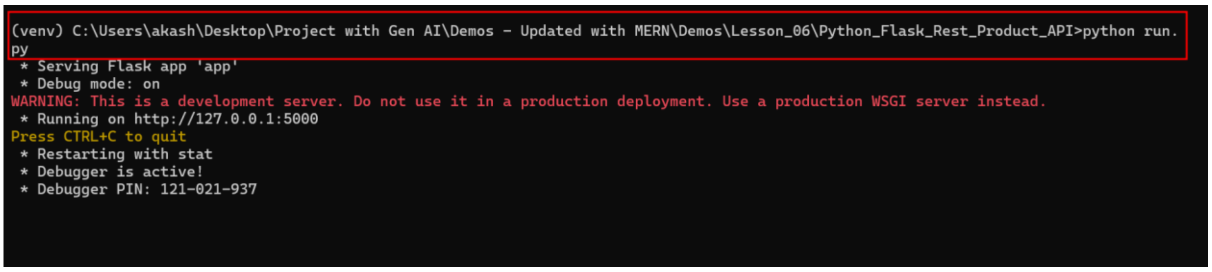
**python -m venv venv**
**On Non Window use : source venv/bin/activate #**
**On Windows use: venv\Scripts\activate**
**pip install -r requirements.txt**



```
(c) Microsoft Corporation. All rights reserved.

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API>python -m venv ven
v

C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API>venv\Scripts\activ
ate

(venv) C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API>pip install
 -r requirements.txt
Collecting flask (from -r requirements.txt (line 1))
  Using cached flask-3.1.0-py3-none-any.whl.metadata (2.7 kB)
Collecting flask_sqlalchemy (from -r requirements.txt (line 2))
  Using cached flask_sqlalchemy-3.1.1-py3-none-any.whl.metadata (3.4 kB)
Collecting flask_marshmallow (from -r requirements.txt (line 3))
  Using cached flask_marshmallow-1.3.0-py3-none-any.whl.metadata (5.2 kB)
Collecting marshmallow_sqlalchemy (from -r requirements.txt (line 4))
  Using cached marshmallow_sqlalchemy-1.4.1-py3-none-any.whl.metadata (6.7 kB)
Collecting Werkzeug>=3.1 (from flask->-r requirements.txt (line 1))
  Using cached werkzeug-3.1.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask->-r requirements.txt (line 1))
  Using cached jinja2-3.1.5-py3-none-any.whl.metadata (2.6 kB)
```

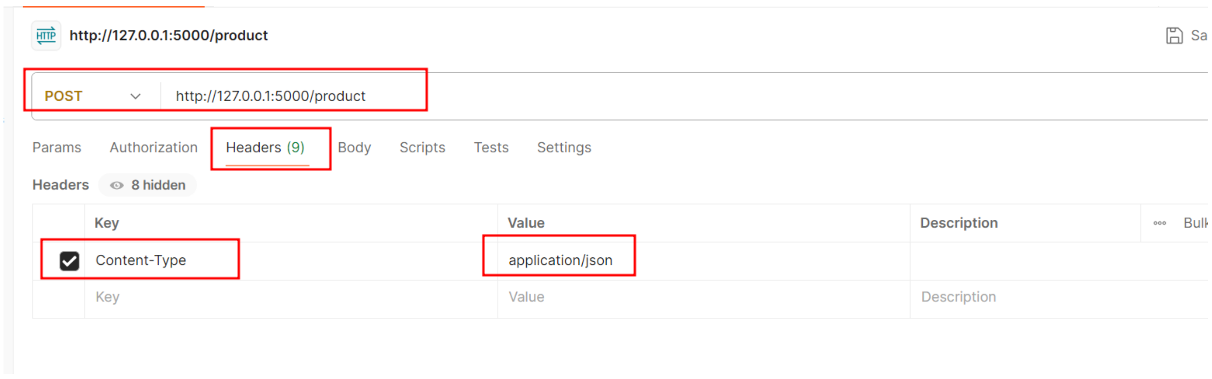
1.4 run the application

**python run.py**



```
(venv) C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API>python run.
py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 121-021-937
```

1.5 test end point using post man client plugin open the VS Code
1.6 Store Product details.



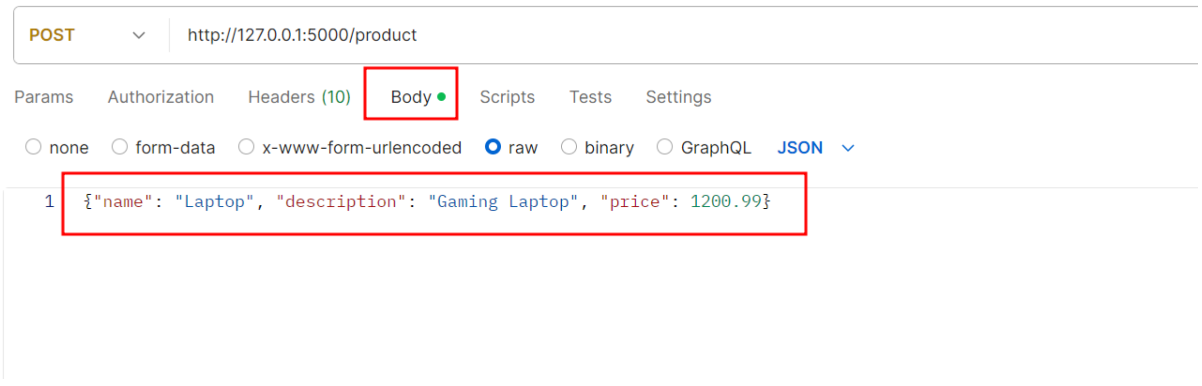
http://127.0.0.1:5000/product

POST http://127.0.0.1:5000/product

Headers (9)

Headers 8 hidden

| | Key | Value | Description |
|---|---|---|---|
| ✓ | Content-Type | application/json | |
| | Key | Value | Description |



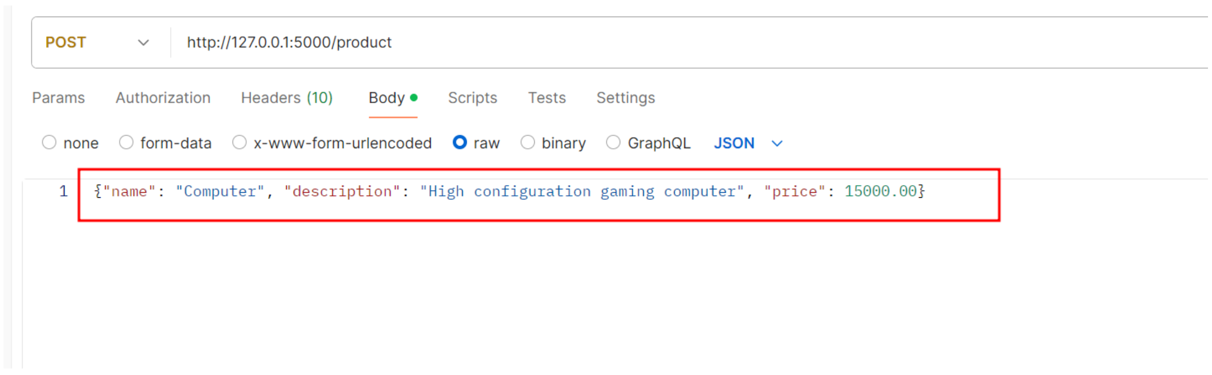
POST http://127.0.0.1:5000/product

Params | Authorization | Headers (10) | Body ● | Scripts | Tests | Settings

none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL | JSON

```
{"name": "Laptop", "description": "Gaming Laptop", "price": 1200.99}
```

Store few products details



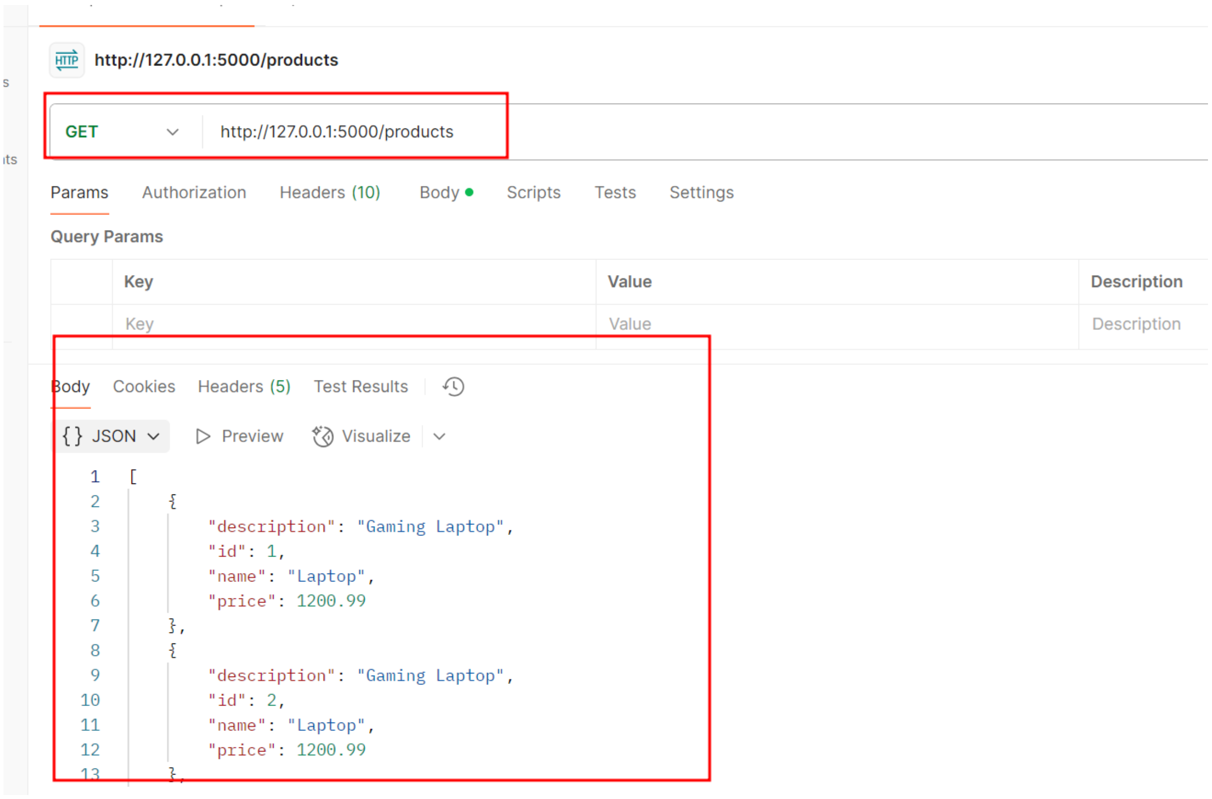
POST http://127.0.0.1:5000/product

Params | Authorization | Headers (10) | Body ● | Scripts | Tests | Settings

none | form-data | x-www-form-urlencoded | [✓ raw] | binary | GraphQL | JSON

```
{"name": "Computer", "description": "High configuration gaming computer", "price": 15000.00}
```

## 1.7 view all products



http://127.0.0.1:5000/products

GET http://127.0.0.1:5000/products

Params | Authorization | Headers (10) | Body ● | Scripts | Tests | Settings

Query Params

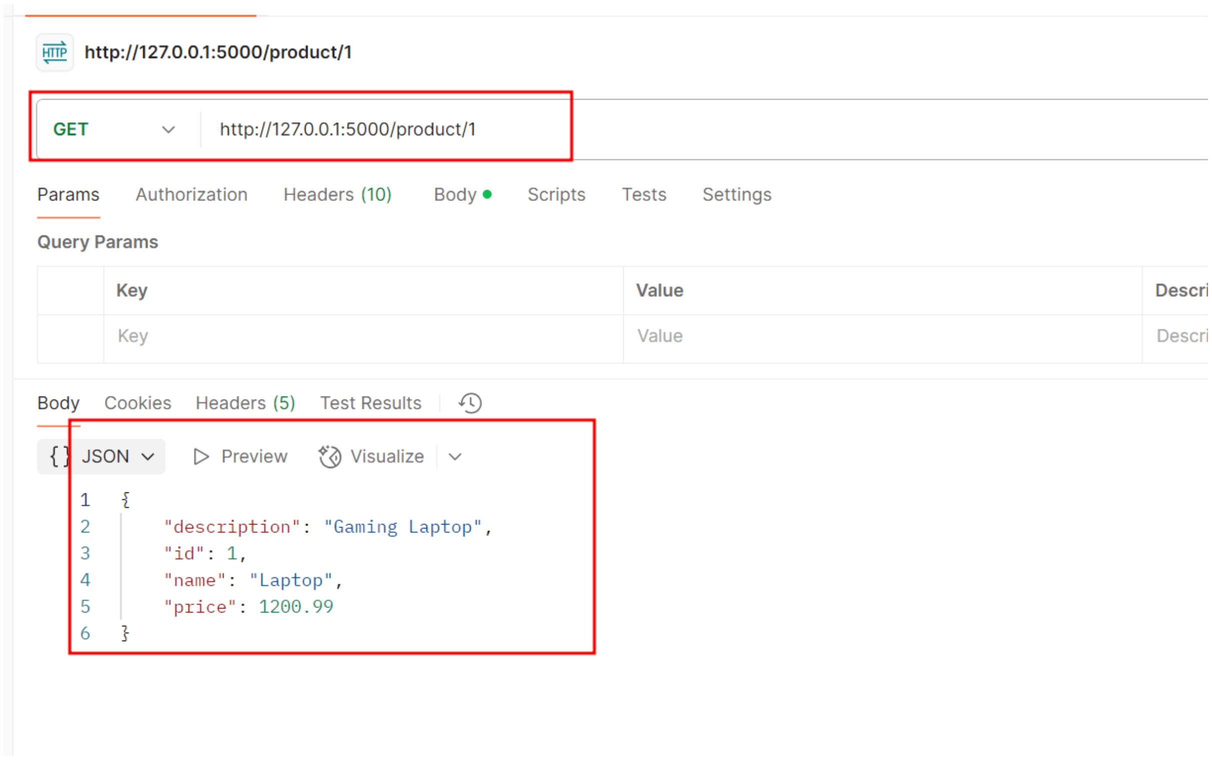
| | Key | Value | Description |
|---|---|---|---|
| | Key | Value | Description |

Body | Cookies | Headers (5) | Test Results

{ } JSON | Preview | Visualize

```
[
    {
        "description": "Gaming Laptop",
        "id": 1,
        "name": "Laptop",
        "price": 1200.99
    },
    {
        "description": "Gaming Laptop",
        "id": 2,
        "name": "Laptop",
        "price": 1200.99
    },
```

## 1.8 Get particular product details



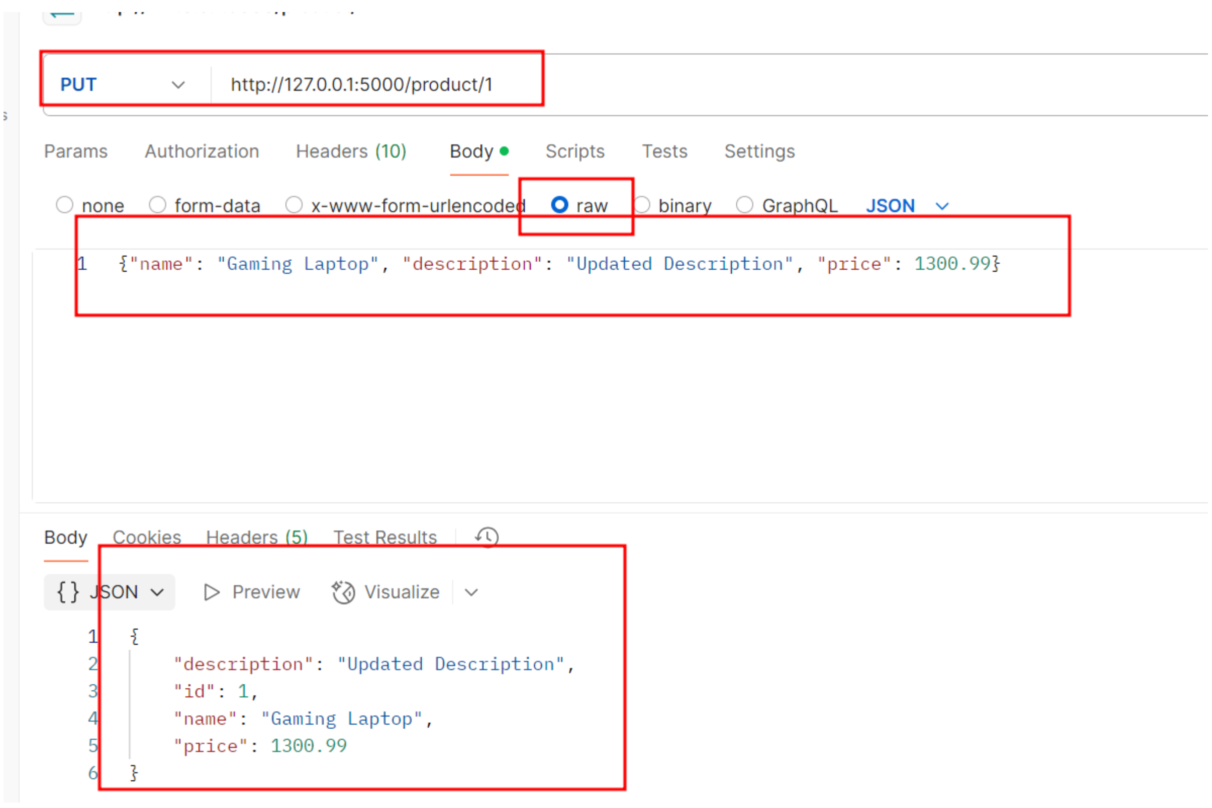
http://127.0.0.1:5000/product/1

GET http://127.0.0.1:5000/product/1

| Key | Value | Descri |
|---|---|---|
| Key | Value | Descri |

```json
{
    "description": "Gaming Laptop",
    "id": 1,
    "name": "Laptop",
    "price": 1200.99
}
```

## 1.9 update the product details



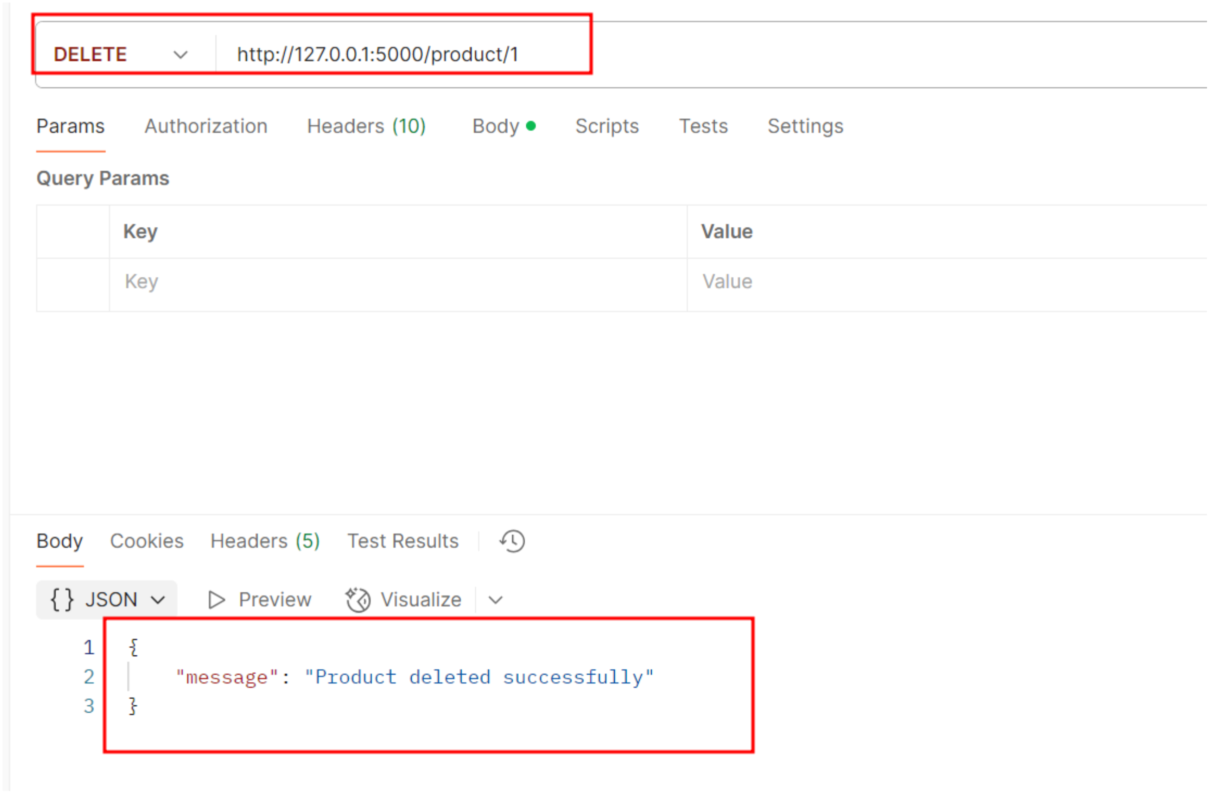
PUT http://127.0.0.1:5000/product/1

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```json
{"name": "Gaming Laptop", "description": "Updated Description", "price": 1300.99}
```

```json
{
    "description": "Updated Description",
    "id": 1,
    "name": "Gaming Laptop",
    "price": 1300.99
}
```

1.20 delete the product details



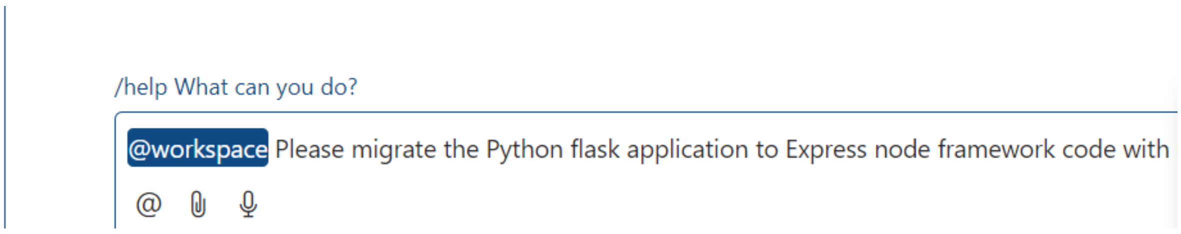
http://127.0.0.1:5000/product/1

```
{
    "message": "Product deleted successfully"
}
```

## Step 2: Migrate the Python Flask to Express Node framework

2.1 Prompt GitHub Copilot in my-angular-app workspace to migrate from the Angular framework into the Python Flask

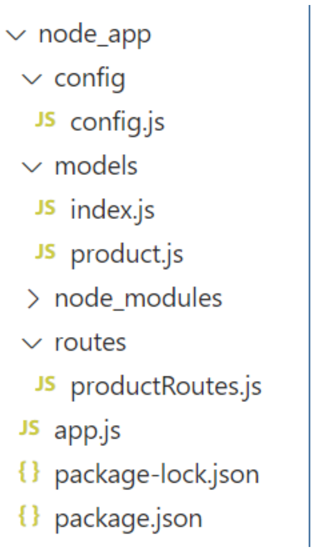
**Please migrate the Python flask application to Express node framework code with step by step explanation with sqllite database**


/help What can you do?

@workspace Please migrate the Python flask application to Express node framework code with

2.2 Here you can see the prompt message with project structure with code. Follow all those steps to make the complete code.

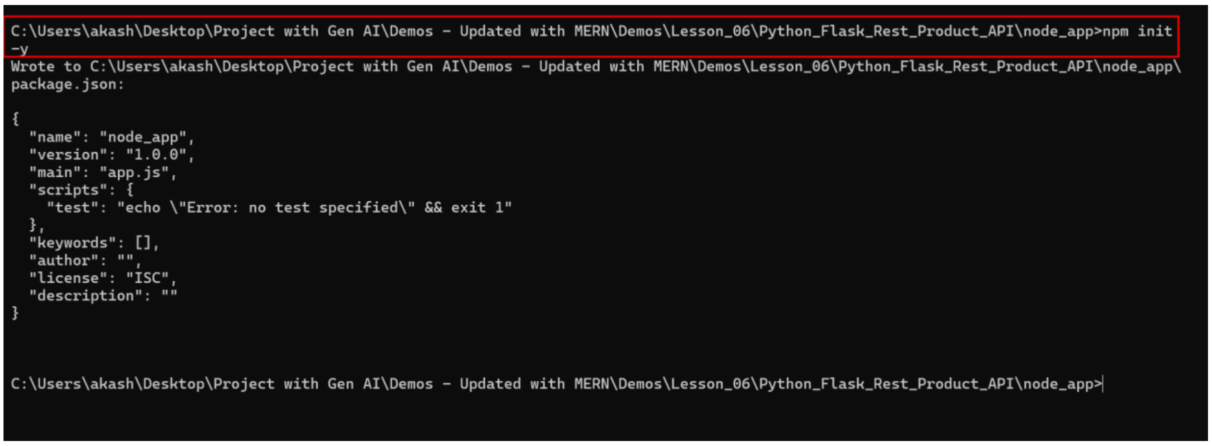
2.3 After project structure ready



```
node_app
  config
    config.js
  models
    index.js
    product.js
  node_modules
  routes
    productRoutes.js
  app.js
  package-lock.json
  package.json
```

2.4 Initialize the Node.js Project

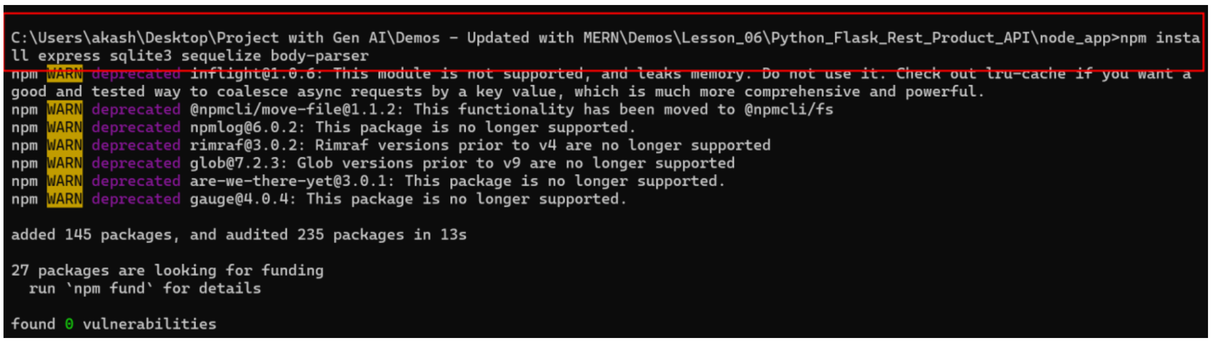
Make sure the terminal open inside node_app folder



```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API\node_app>npm init
-y
Wrote to C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API\node_app\
package.json:

{
  "name": "node_app",
  "version": "1.0.0",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}


C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API\node_app>
```

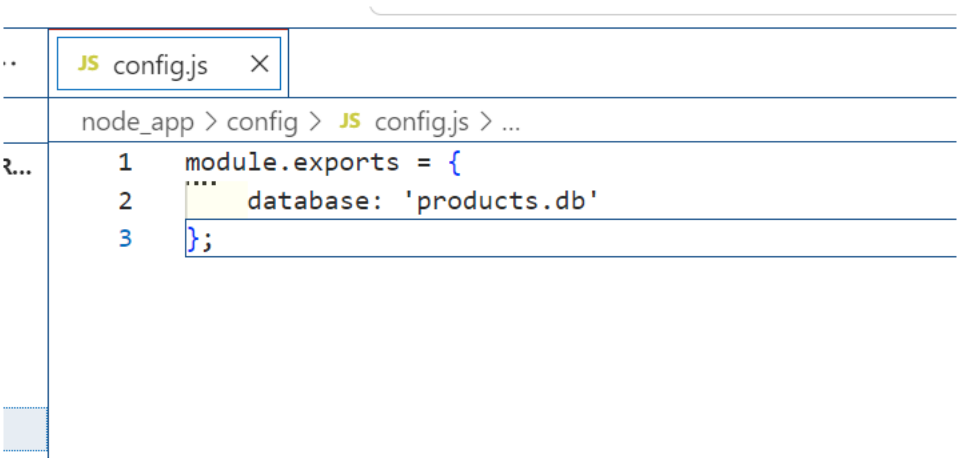
2.5 Install the necessary dependencies:

**npm install express sqlite3 sequelize body-parser**



```
C:\Users\akash\Desktop\Project with Gen AI\Demos - Updated with MERN\Demos\Lesson_06\Python_Flask_Rest_Product_API\node_app>npm insta
ll express sqlite3 sequelize body-parser
npm WARN deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a
good and tested way to coalesce async requests by a key value, which is much more comprehensive and powerful.
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated npmlog@6.0.2: This package is no longer supported.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated glob@7.2.3: Glob versions prior to v9 are no longer supported
npm WARN deprecated are-we-there-yet@3.0.1: This package is no longer supported.
npm WARN deprecated gauge@4.0.4: This package is no longer supported.

added 145 packages, and audited 235 packages in 13s

27 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

2.6 apply each file code the empty file



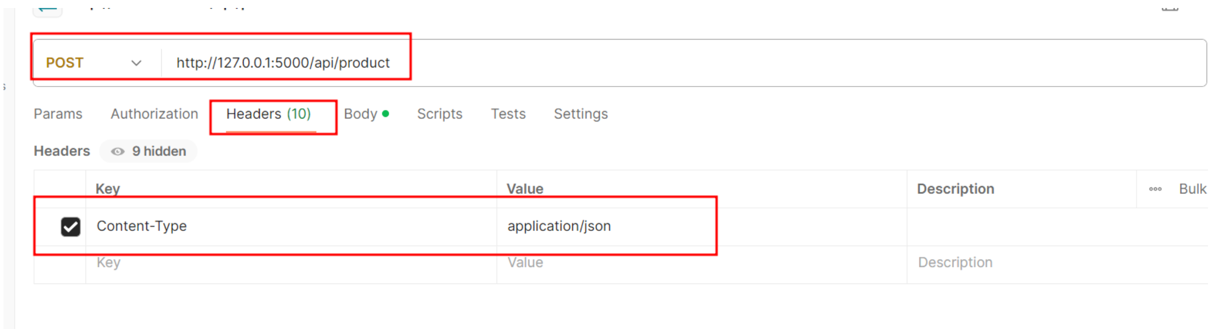
```
module.exports = {
    database: 'products.db'
};
```

2.7 after copied all code
2.8 you can run the application using below command as

**node app.js**

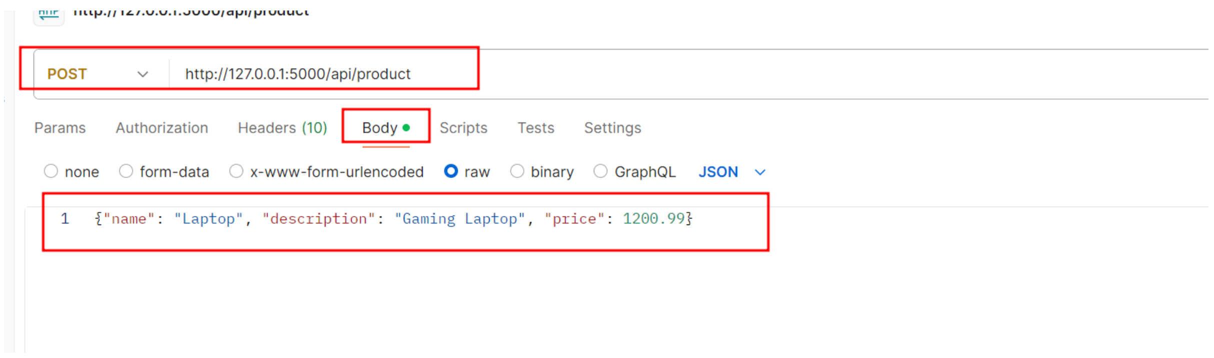
2.9 store the data

First set the header information



POST http://127.0.0.1:5000/api/product

| | Key | Value | Description |
|---|---|---|---|
| ✓ | Content-Type | application/json | |
| | Key | Value | Description |



http://127.0.0.1:5000/api/product

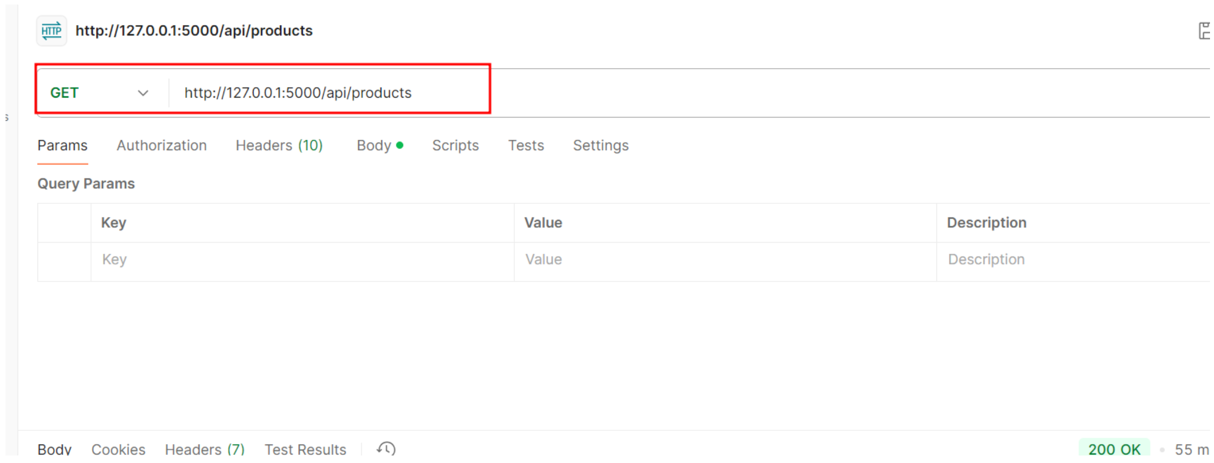
POST http://127.0.0.1:5000/api/product

Params Authorization Headers (10) Body ● Scripts Tests Settings

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON

```
{"name": "Laptop", "description": "Gaming Laptop", "price": 1200.99}
```

## 2.10 view all product data



http://127.0.0.1:5000/api/products

GET http://127.0.0.1:5000/api/products

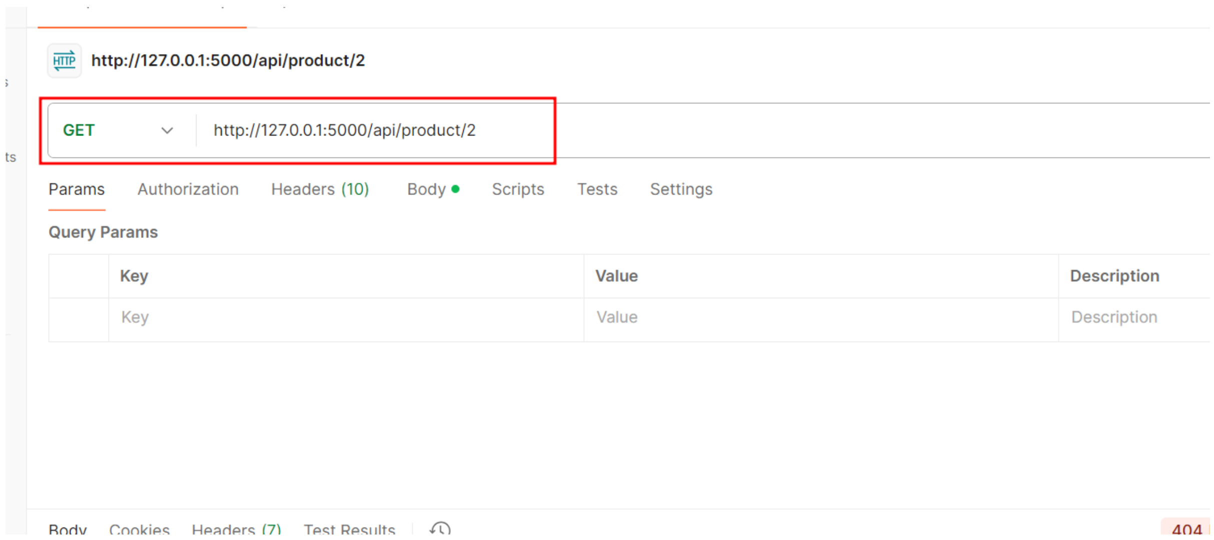
Params Authorization Headers (10) Body ● Scripts Tests Settings

Query Params

| Key | Value | Description |
|---|---|---|
| Key | Value | Description |

Body Cookies Headers (7) Test Results 200 OK 55 m

## 2.11 view particular product data using pid



http://127.0.0.1:5000/api/product/2

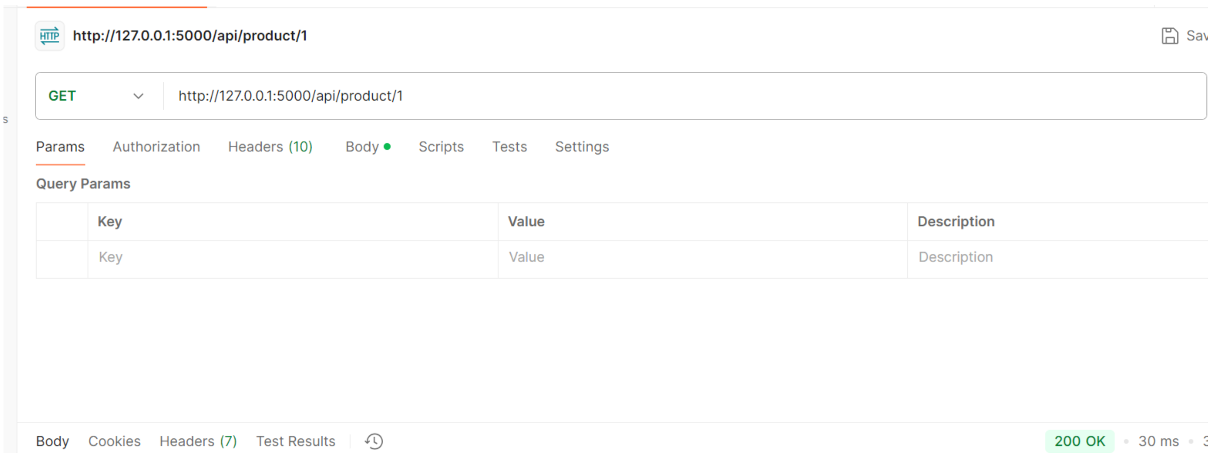
GET http://127.0.0.1:5000/api/product/2

Params Authorization Headers (10) Body ● Scripts Tests Settings

Query Params

| Key | Value | Description |
|---|---|---|
| Key | Value | Description |

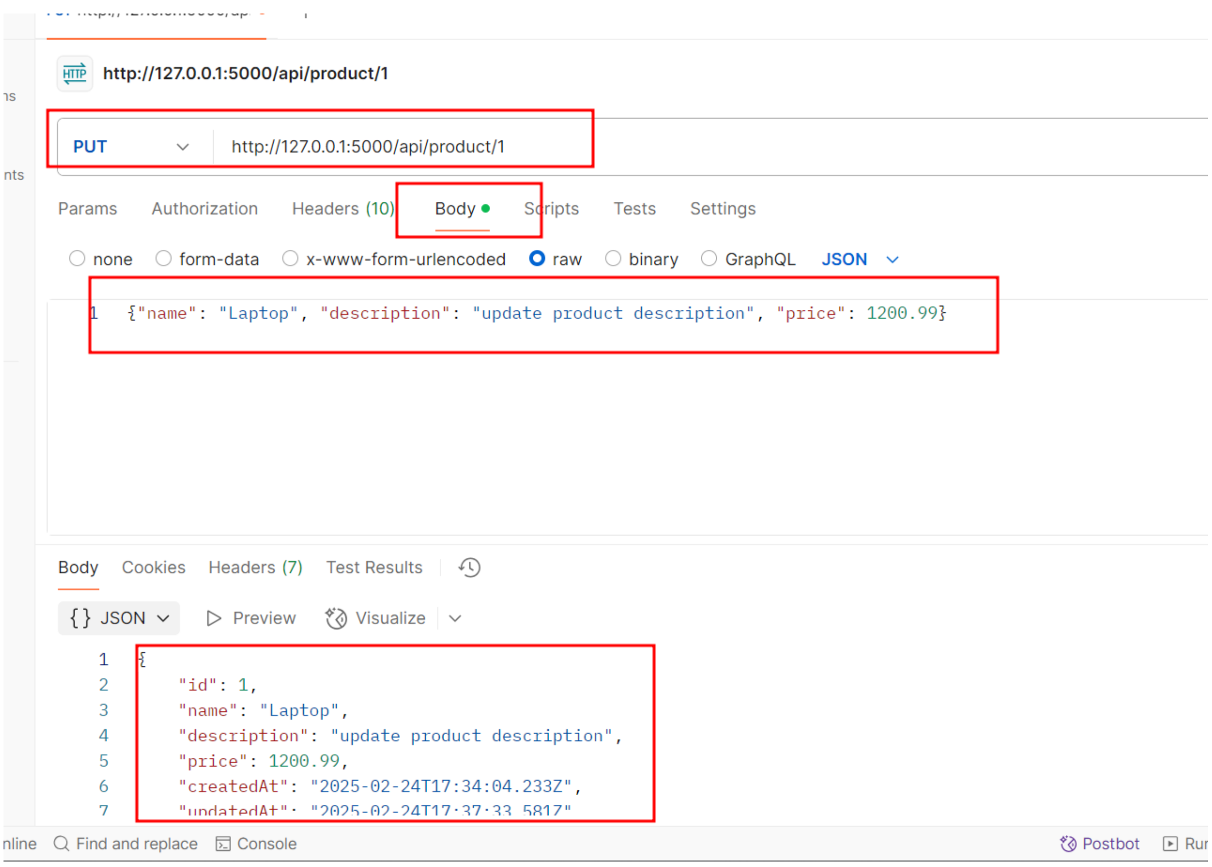
Body Cookies Headers (7) Test Results 404



GET http://127.0.0.1:5000/api/product/1

| Key | Value | Description |
|---|---|---|
| Key | Value | Description |

## 2.12 update the product details

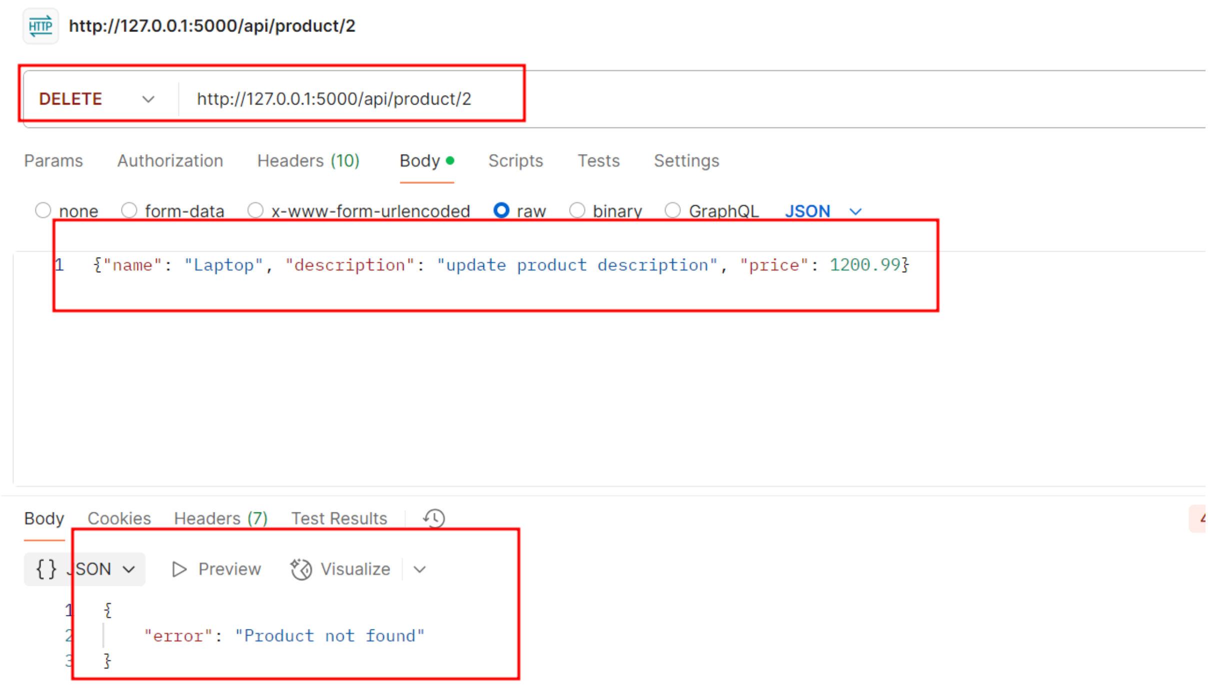


PUT http://127.0.0.1:5000/api/product/1

```
{"name": "Laptop", "description": "update product description", "price": 1200.99}
```

```
{
    "id": 1,
    "name": "Laptop",
    "description": "update product description",
    "price": 1200.99,
    "createdAt": "2025-02-24T17:34:04.233Z",
    "updatedAt": "2025-02-24T17:37:33.581Z"
```

## 2.13 Delete the product

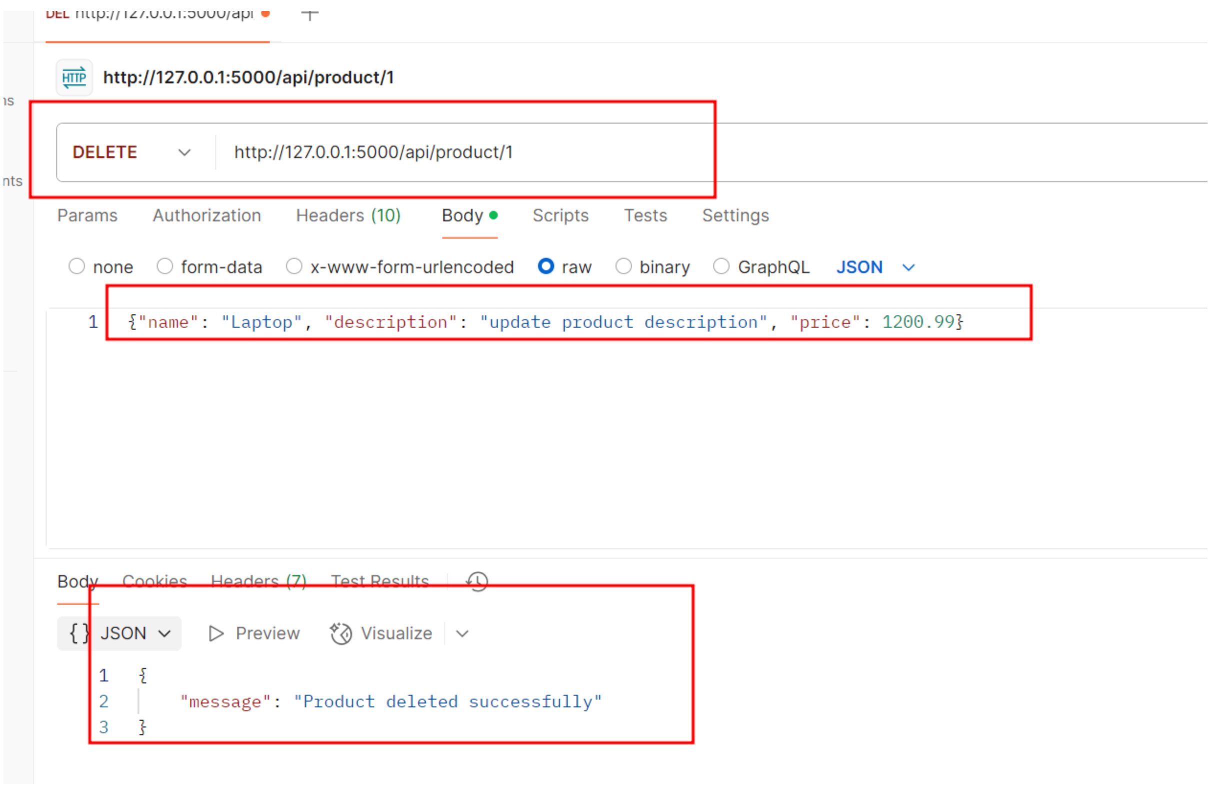


http://127.0.0.1:5000/api/product/2

DELETE http://127.0.0.1:5000/api/product/2

```
{"name": "Laptop", "description": "update product description", "price": 1200.99}
```

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



http://127.0.0.1:5000/api/product/1

DELETE http://127.0.0.1:5000/api/product/1

```
{"name": "Laptop", "description": "update product description", "price": 1200.99}
```

```
{
    "message": "Product deleted successfully"
}
```

By following the outlined steps, you have successfully migrated the python flask functionality node js express js framework using GitHub Copilot, ensuring a successful transition while maintaining the integrity and functionality of the REST API system.