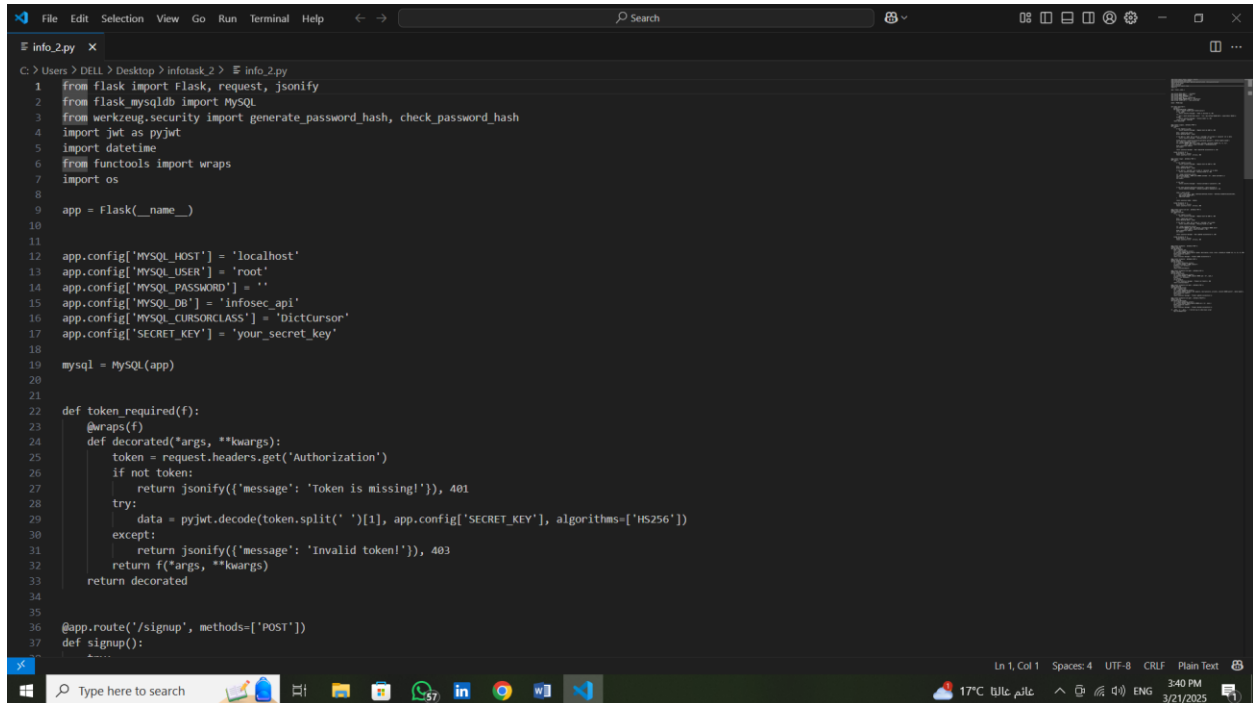


Name : **adham ahmed reda ahmed**

Id: **2205087**

In first we make the code python



```
File Edit Selection View Go Run Terminal Help
info_2.py
C:\Users\DELL\Desktop>infotask_2> info_2.py
1 from flask import Flask, request, jsonify
2 from flask_mysql import MySQL
3 from werkzeug.security import generate_password_hash, check_password_hash
4 import jwt as pyjwt
5 import datetime
6 from functools import wraps
7 import os
8
9 app = Flask(__name__)
10
11
12 app.config['MYSQL_HOST'] = 'localhost'
13 app.config['MYSQL_USER'] = 'root'
14 app.config['MYSQL_PASSWORD'] = ''
15 app.config['MYSQL_DB'] = 'infosec_api'
16 app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
17 app.config['SECRET_KEY'] = 'your_secret_key'
18
19 mysql = MySQL(app)
20
21
22 def token_required(f):
23     @wraps(f)
24     def decorated(*args, **kwargs):
25         token = request.headers.get('Authorization')
26         if not token:
27             return jsonify({'message': 'Token is missing!'}), 401
28         try:
29             data = pyjwt.decode(token.split(' ')[1], app.config['SECRET_KEY'], algorithms=['HS256'])
30         except:
31             return jsonify({'message': 'Invalid token!'}), 403
32         return f(*args, **kwargs)
33     return decorated
34
35
36 @app.route('/signup', methods=['POST'])
37 def signup():
38     ...
```

Ok then I run the code ok then we open postman and search in <http://127.0.0.1:5000/>

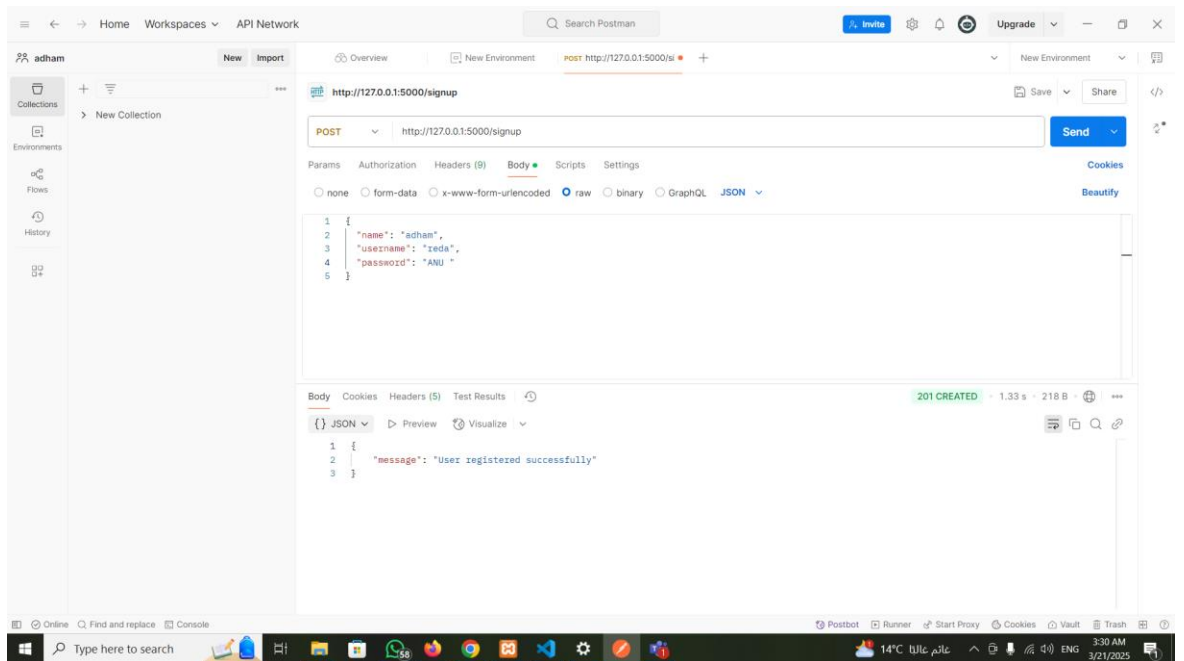
1-In first we signup and save the name and the hash password in sql

<http://127.0.0.1:5000/signup>

```
1 from werkzeug.security import generate_password_hash, check_password_hash
2 import jwt as jwt
3 import datetime
4 from functools import wraps
5 import os
6
7 app = Flask(__name__)
8
9 app.config['MYSQL_HOST'] = 'localhost'
10 app.config['MYSQL_USER'] = 'root'
11 app.config['MYSQL_PASSWORD'] = ''
12 app.config['MYSQL_DB'] = 'infotask_2'
13 app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
14 app.config['SECRET_KEY'] = 'your_secret_key'
15
16 mysql = MySQL(app)
```

File "C:\Users\DELL\Desktop\infotask_2\info_2.py", line 134, in add product
cur.execute("INSERT INTO products (pname, description, price, stock, created_at) VALUES (%s, %s, %s, %s, NOW())", (data['pname'], data['description'], data['price'], data['stock']))

TypeError: list indices must be integers or slices, not str
(venv) PS C:\Users\DELL\Desktop\infotask_2> & c:\Users\DELL\Desktop\infotask_2\venv\Scripts\python.exe c:\Users\DELL\Desktop\infotask_2\info_2.py
* Serving Flask app 'info_2'
* 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: 788-698-718
Received data: {'name': 'adham', 'username': 'reda', 'password': 'ANU'}
127.0.0.1 - - [21/Mar/2025 03:30:31] "POST /signup HTTP/1.1" 201 -



2- And then we login to enter and take the token <http://127.0.0.1:5000/login>

The image shows two screenshots from a development environment. The top screenshot is from Postman, showing a POST request to `http://127.0.0.1:5000/login` with a JSON body: `{ "name": "adham", "username": "reda", "password": "ANU" }`. The response is a 200 OK status with a JSON body: `{ "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImYwIjZlY2F0LmAg3e1v-eZUnmR5wpuQ3ENLADMeB5Eh0wEzR60neIM" }`. The bottom screenshot is from VS Code, showing the `info_2.py` file and the terminal output. The terminal shows the server running on `http://127.0.0.1:5000` and receiving the login request, returning a 200 status.

```
from werkzeug.security import generate_password_hash, check_password_hash
import jwt as pyjwt
import datetime
from functools import wraps
import os

app = Flask(__name__)

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = ''
app.config['MYSQL_DB'] = 'infosec_api'
app.config['MYSQL_CURSORCLASS'] = 'DictCursor'
app.config['SECRET_KEY'] = 'your_secret_key'

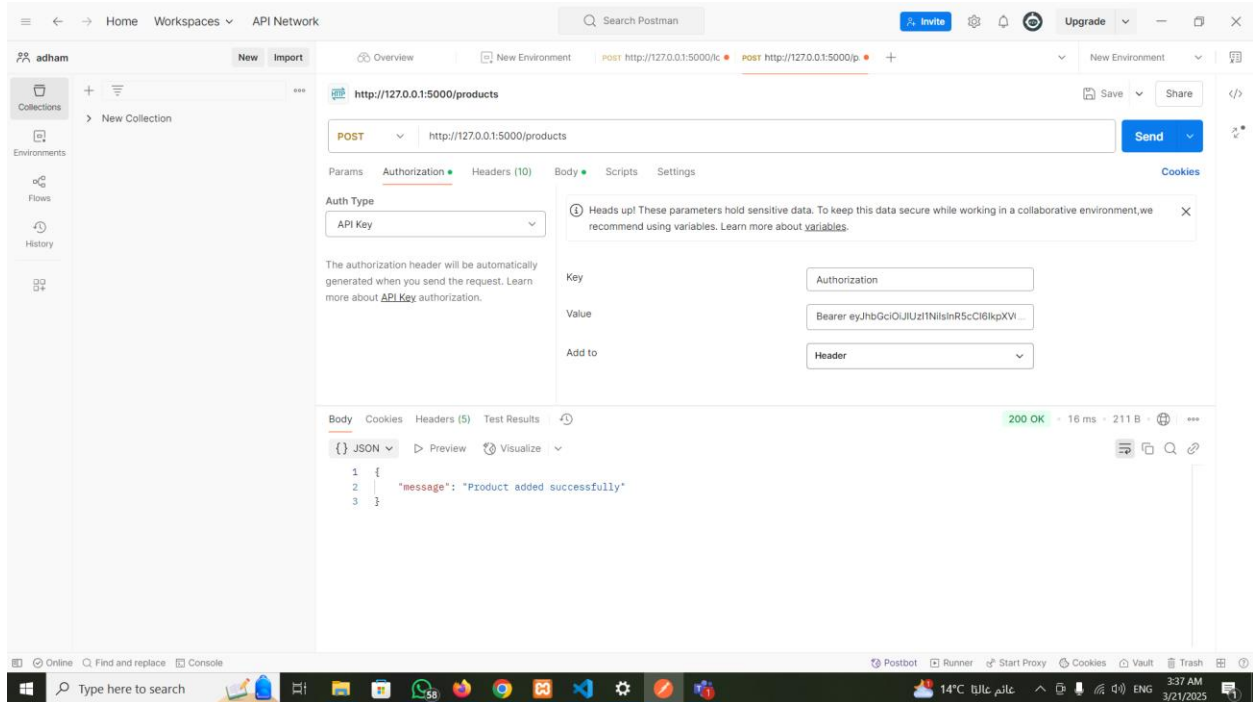
mysql = MySQL(app)

e'], data['stock']))

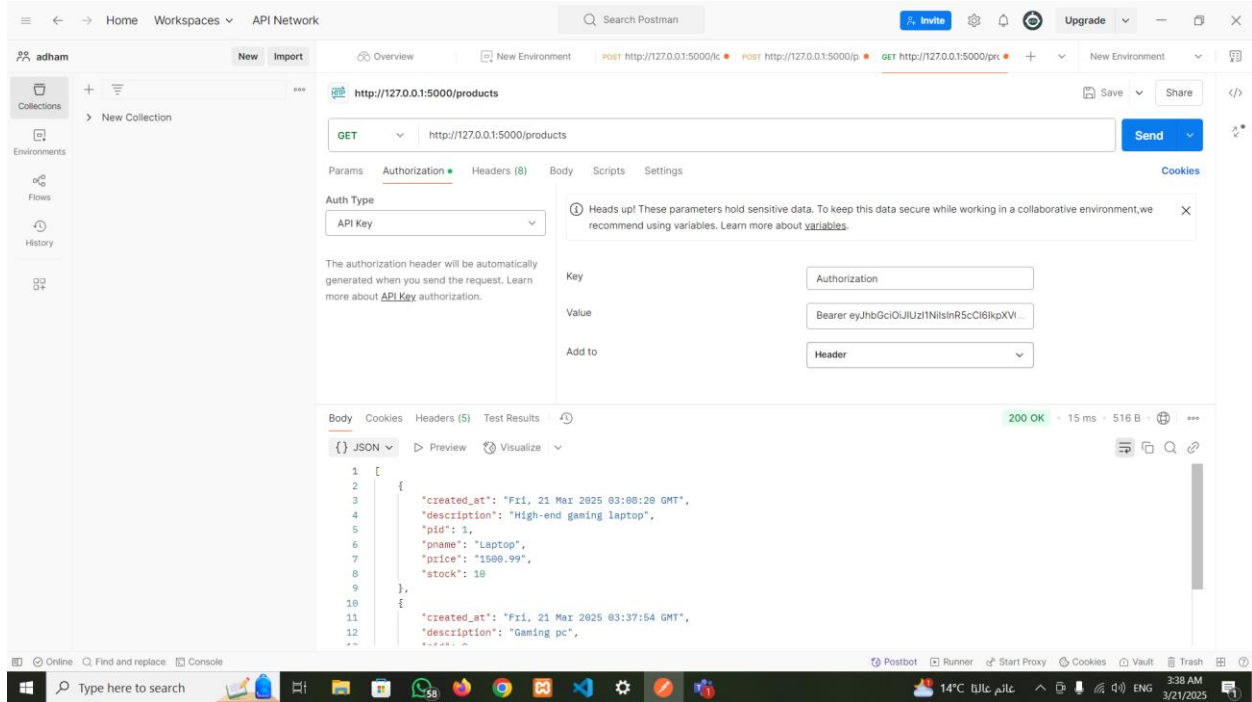
TypeError: list indices must be integers or slices, not str
(venv) ps c:/Users/DELL/Desktop/infotask_2> & c:/Users/DELL/Desktop/infotask_2/venv/Scripts/python.exe c:/Users/DELL/Desktop/infotask_2/info_2.py
* Serving Flask app "info_2"
* Debug mode: on
WARNING: This is a development server. Do not use 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: 788-698-718
Received Data: {'name': 'adham', 'username': 'reda', 'password': 'ANU '}
127.0.0.1 - - [21/Mar/2025 03:30:31] "POST /signup HTTP/1.1" 201 -
Received Data: {'name': 'adham', 'username': 'reda', 'password': 'ANU '}
127.0.0.1 - - [21/Mar/2025 03:31:06] "POST /login HTTP/1.1" 200 -
```

Then I make the rest of task

3 Add product



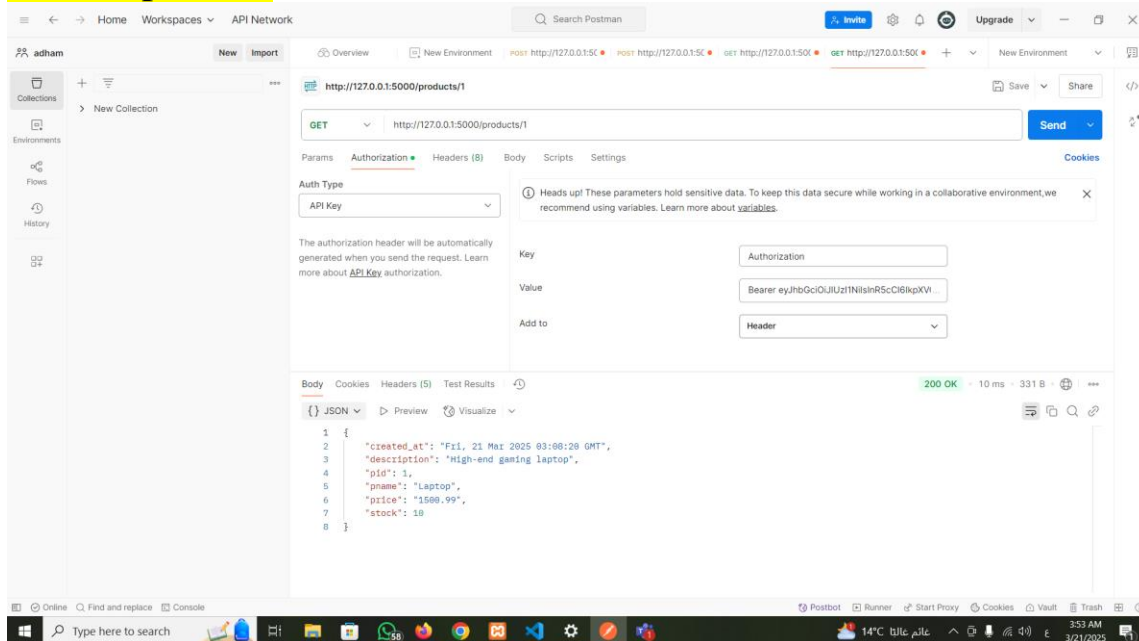
Retrieve product



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/products`. The request is configured with an API Key authorization header. The response is a 200 OK status with a response time of 15 ms and a body size of 516 B. The response body is a JSON array of two product objects.

```
{
  "created_at": "Fri, 21 Mar 2025 03:08:20 GMT",
  "description": "High-end gaming laptop",
  "pid": 1,
  "pname": "Laptop",
  "price": "1500.99",
  "stock": 10
},
{
  "created_at": "Fri, 21 Mar 2025 03:37:54 GMT",
  "description": "Gaming pc",
  "pid": 2,
  "pname": "PC",
  "price": "800.99",
  "stock": 5
}
```

Edit on product



The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:5000/products/1`. The request is configured with an API Key authorization header. The response is a 200 OK status with a response time of 10 ms and a body size of 331 B. The response body is a JSON object representing a single product.

```
{
  "created_at": "Fri, 21 Mar 2025 03:08:20 GMT",
  "description": "High-end gaming laptop",
  "pid": 1,
  "pname": "Laptop",
  "price": "1500.99",
  "stock": 10
}
```

Postman interface showing a GET request to `http://127.0.0.1:5000/products/2`. The request is authenticated using an API Key. The response is a 200 OK status with a JSON body:

```
{
  "created_at": "Fri, 21 Mar 2025 03:37:54 GMT",
  "description": "Gaming pc",
  "pid": 2,
  "pname": "pc",
  "price": "1111.56",
  "stock": 10
}
```

Updated product

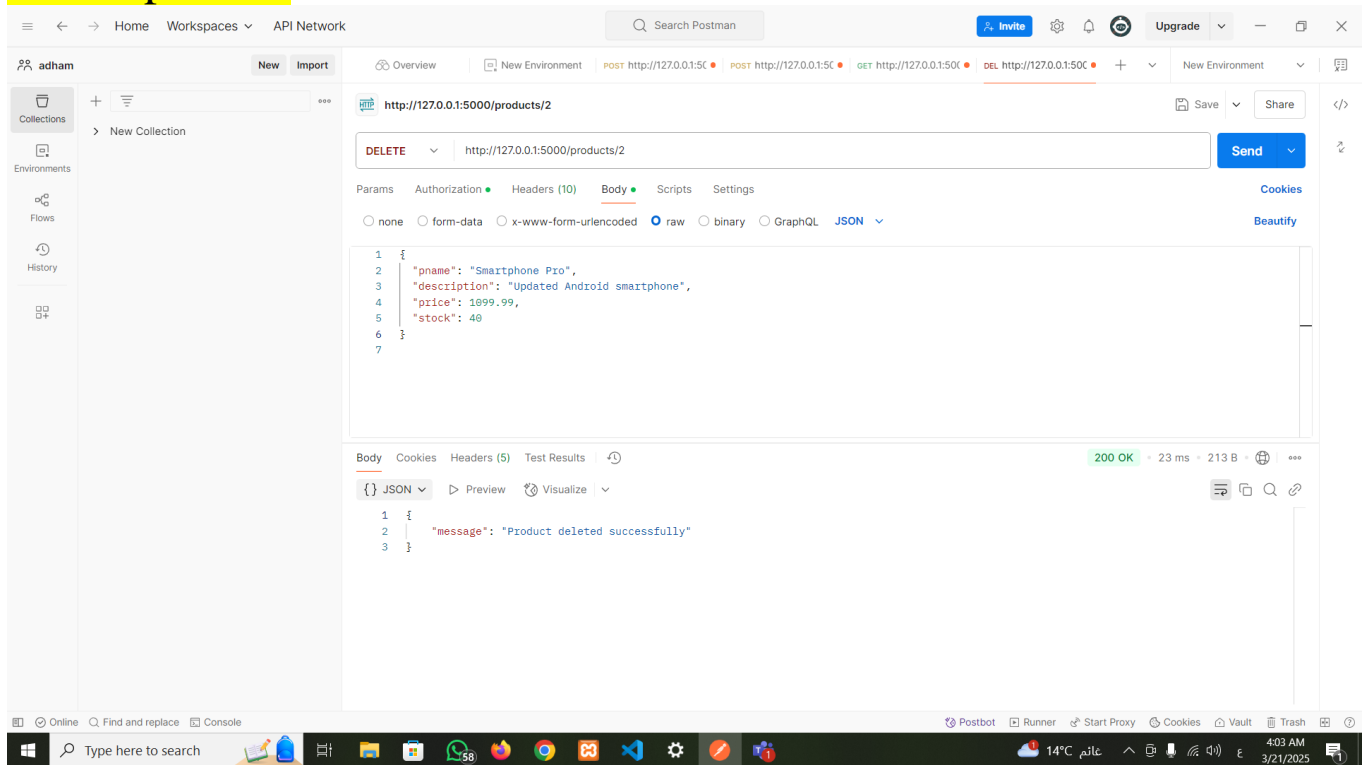
Postman interface showing a PUT request to `http://127.0.0.1:5000/products/2`. The request is authenticated using an API Key. The request body is a JSON object:

```
{
  "pname": "Smartphone Pro",
  "description": "Updated Android smartphone",
  "price": 1099.99,
  "stock": 40
}
```

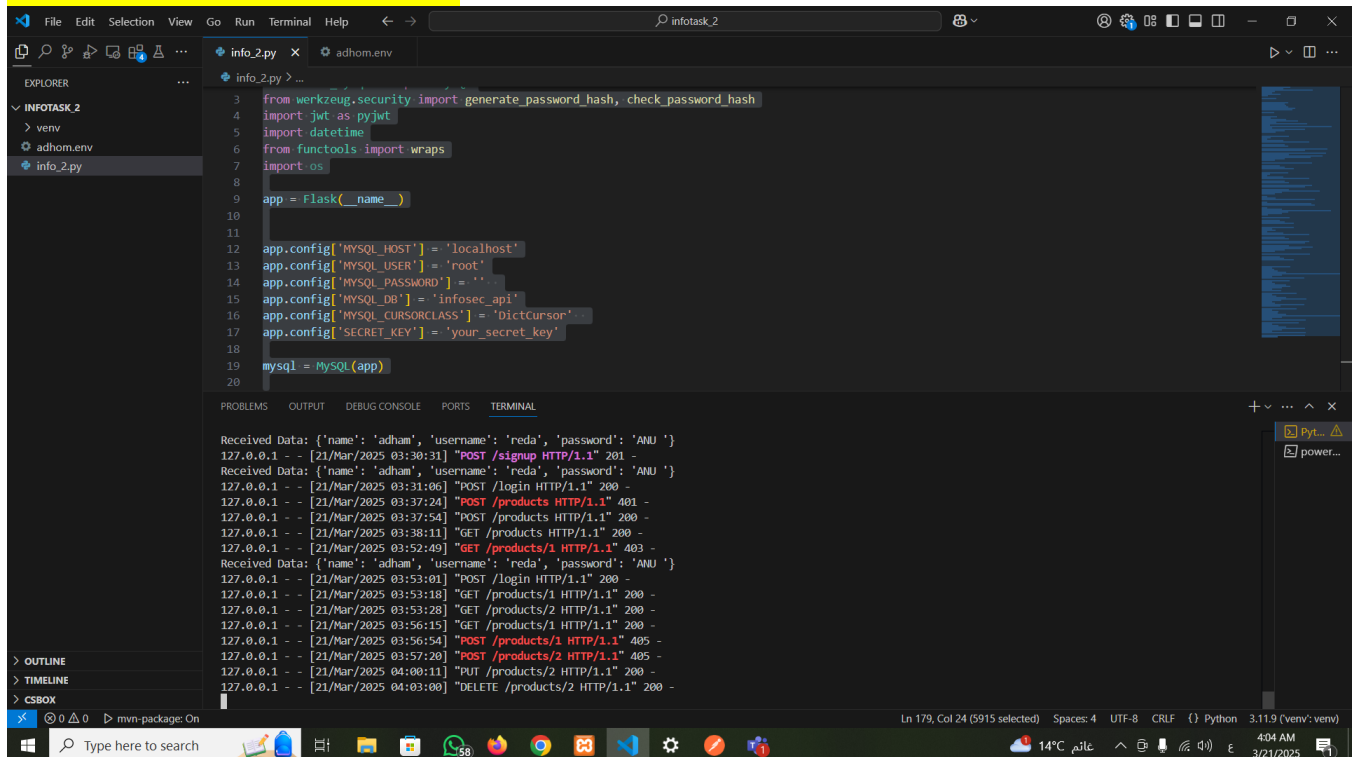
The response is a 200 OK status with a JSON body:

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

Delete product



This show that all task is done



localhost/phpmyadmin/index.php?route=/sql&pos=0&db=infosec_api&table=users

Database: infosec_api Table: users

Showing rows 0 - 2 (المجموع 3, استغرق الاستعلام 0.0003 ثانية)

SELECT * FROM `users`

جانبى [تحرير مُضمن] [تعديل] [شرح SQL] [أنشئ كود PHP] [تحديث]

عرض الكل | عدد الأسطر: 25 | تصفية الصفوف: | ابحث في هذا الجدول | Sort by key: لا شيء

id	name	username	password
1	John Doe	johndoe	...pbkdf2:sha256:1000000\$Ttdxu7EU18SB1nL\$cd70ecf9e24
2	adham	adhamahmed	...pbkdf2:sha256:1000000\$LELTqY8WcrWTTVeE\$0ee63923239
3	adham	reda	...pbkdf2:sha256:1000000\$J2smGTF8UqDQQA3m\$14f484d4fb9

تصدير | حذف | نسخ | تعديل | مع المحدد: | تحقق من الكل

عملیات على نتائج الاستعلام

طباعة | نسخ إلى الحافظة | تصدير | عرض الرسم البياني | إنشاء عرض

اجعل علامة مرجعية SQL-استعلام

اسمح لكل المستخدمين الوصول إلى هذه العلامة المرجعية

Label:

مرجعية SQL-استعلام

3:38 AM 3/21/2025 ENG 14°C عائم غاليتا

localhost / 127.0.0.1

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=infosec_api&table=products

Table: products Database: infosec_api 127.0.0.1

العمليات تصدير استيراد الإحصائيات تتبع القوائم

Showing rows 0 - 1 (المجموع 2, استغرق الاستعلام 0.0004 ثانية).

SELECT * FROM `products`

جانبى [تحرير مُضمن] [تعديل] [شرح SQL] [أنشئ كود PHP] [تحديث]

عرض الكل | عدد الأسطر: 25 | تصفية الصفوف: ابحث في هذا الجدول | Sort by key: لا شيء

created_at	stock	price	description	pname	pid	
03:08:20 2025-03-21 10	10	1500.99	High-end gaming laptop	Laptop 1	1	حذف نسخ تعديل
03:37:54 2025-03-21 10	10	1111.56	Gaming pc	pc 2	2	حذف نسخ تعديل

تصدير حذف نسخ تعديل مع المجدد: تحقق من الكل

Sort by key: لا شيء | تصفية الصفوف: ابحث في هذا الجدول | عدد الأسطر: 25

عمليات على نتائج الاستعلام

إظهار عرض الرسم البياني تصدير نسخ إلى الحافظة طباعة

اجعل علامة مرجعية SQL-استعلام

اسمح لكل المستخدمين الوصول إلى هذه العلامة المرجعية

اجعل علامة مرجعية SQL-استعلام

المعارض

phpMyAdmin

المفضلة

New

car_app

information_schema

infosec

infosec_api

New

products

users

mysql

performance_schema

phpmyadmin

re

test

testdb

travell

Type here to search

14°C عالم عالتا 3:38 AM 3/21/2025