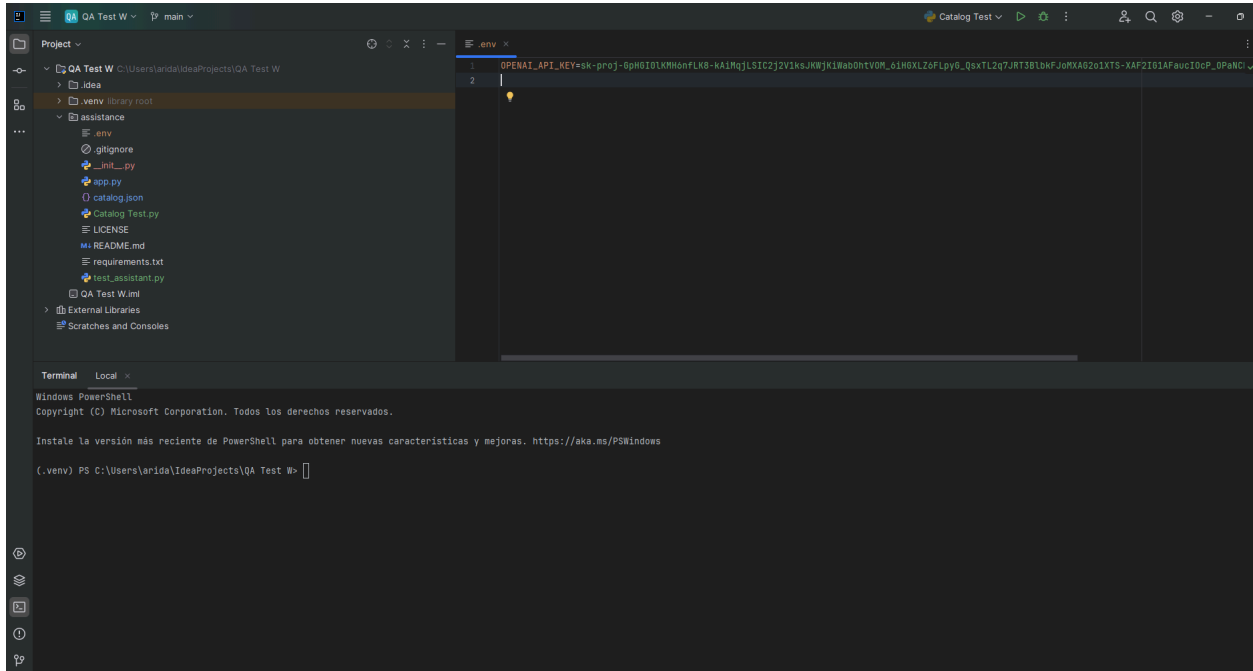
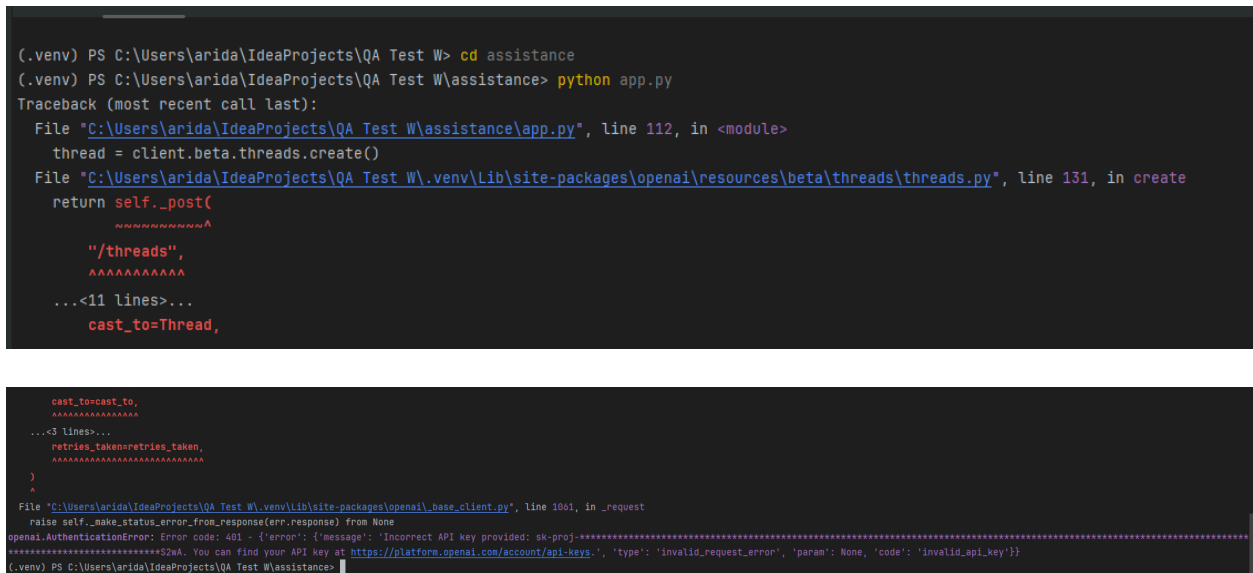


TEST NOTES

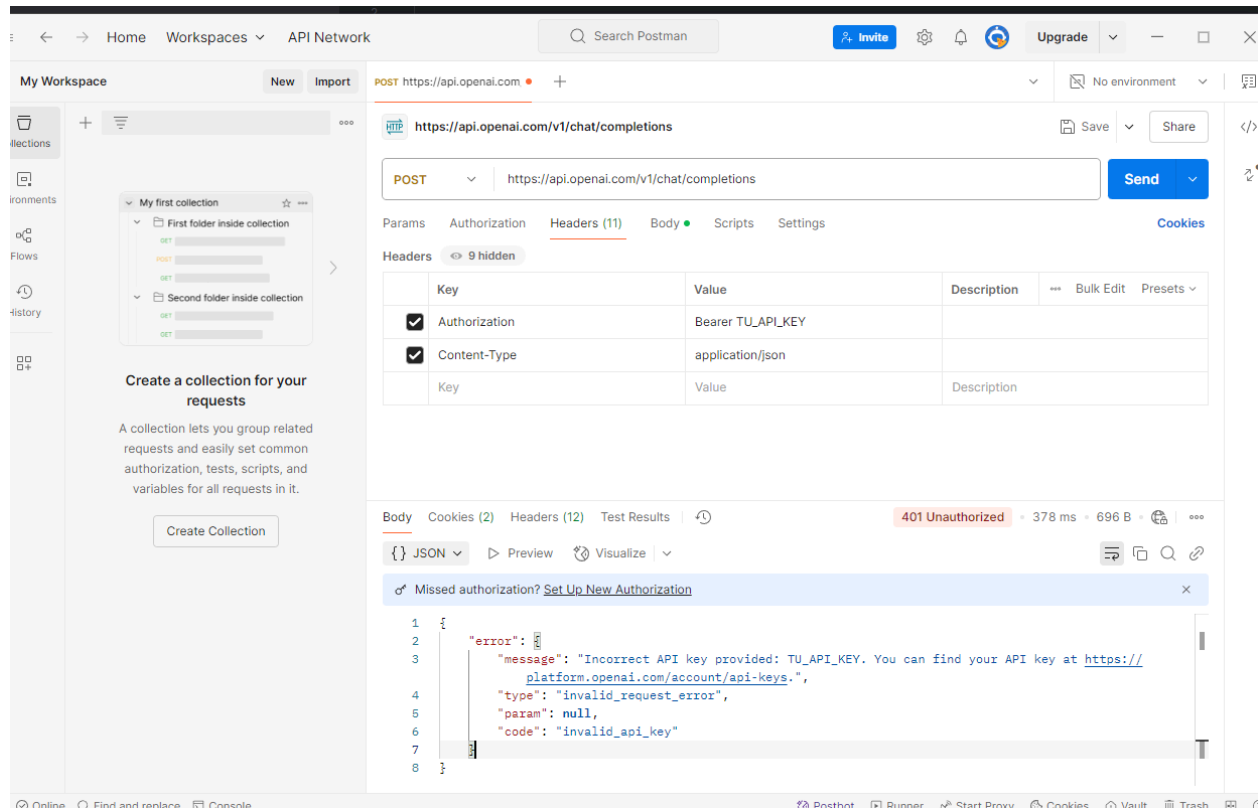
Repository cloned successfully with IntelliJ



But the API Key provided did not work, it had expired

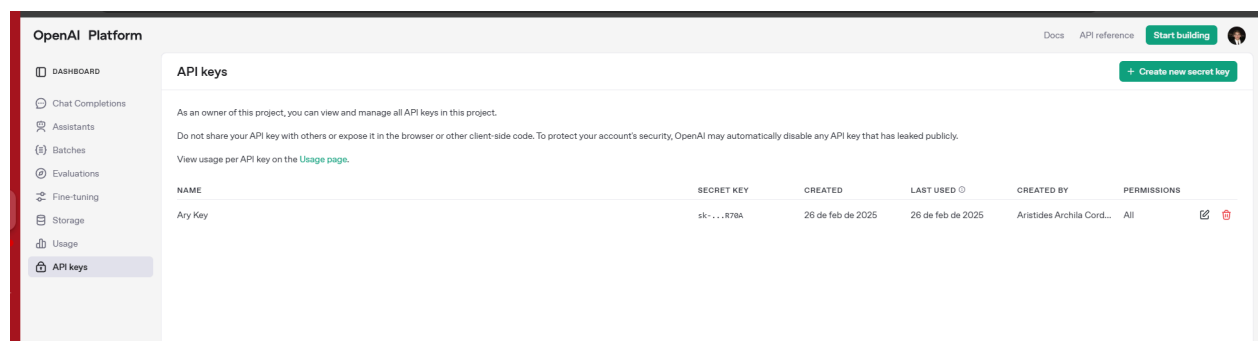


I proved the API with POSTMAN, and confirm that API was expire

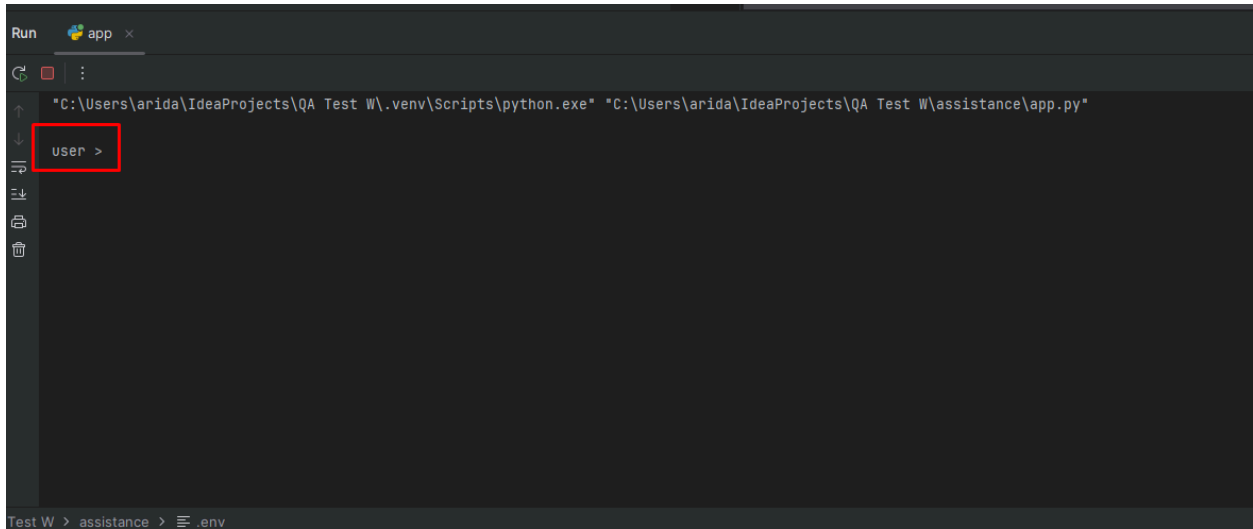


I was trying to contact you, but I couldn't, so I tried to continue the test like this.

I create a new API key with my operAI user and i used but this presented new problems

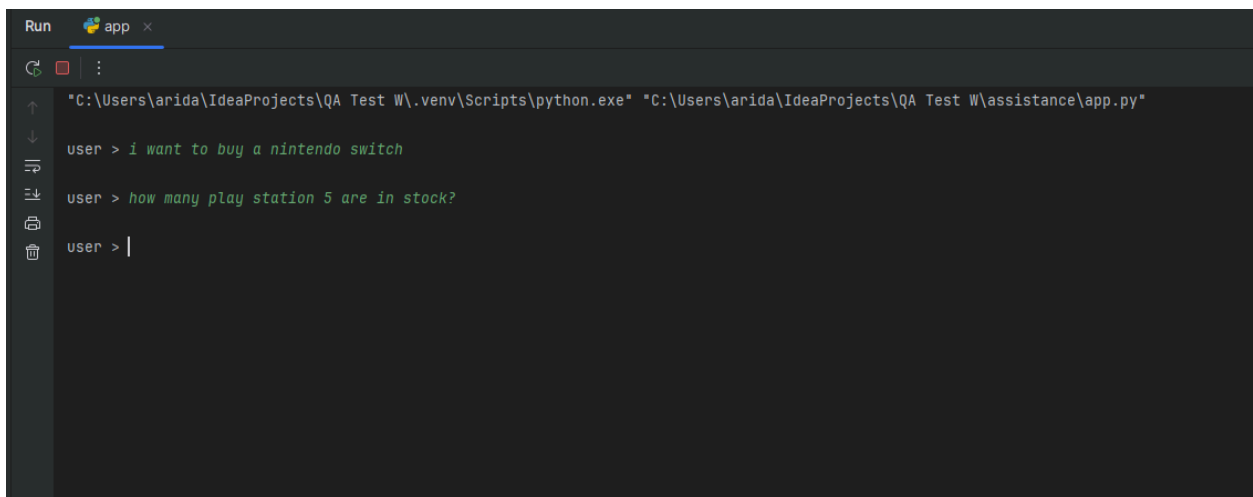


With that new key, I managed to get the program to start.

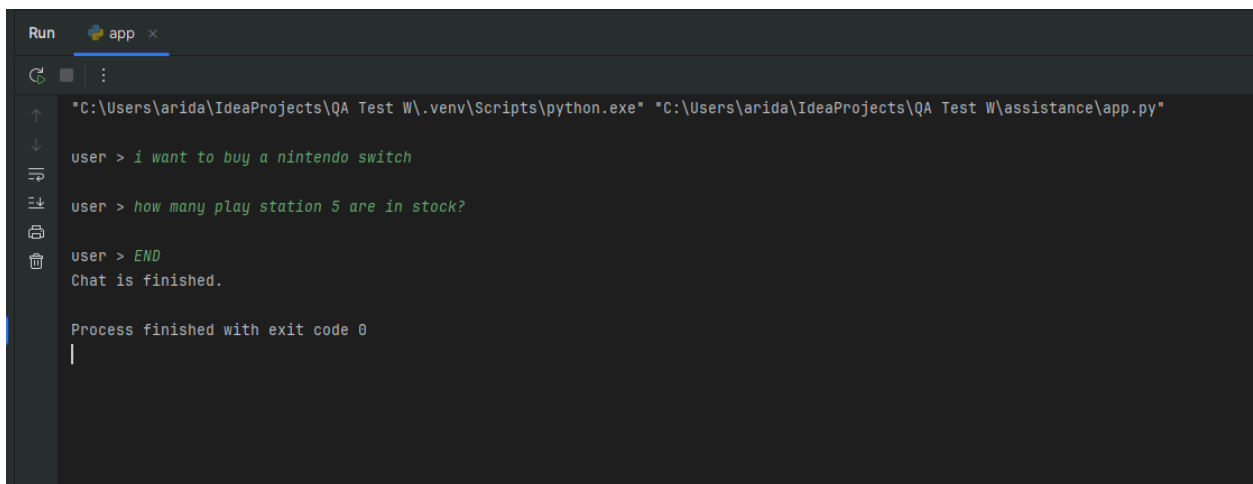


A screenshot of a terminal window titled "Run" with a tab labeled "app". The command bar shows the execution of a Python script: `"C:\Users\arida\IdeaProjects\QA Test W\.venv\Scripts\python.exe" "C:\Users\arida\IdeaProjects\QA Test W\assistance\app.py"`. Below the command bar, the prompt `user >` is visible and highlighted with a red rectangle. The terminal interface includes standard navigation icons on the left and a status bar at the bottom indicating the current directory as `Test W > assistance` and the active environment as `.env`.

But here new problems appeared, nothing I wrote got a response from the AI
im a smoke manual test to the AI but, the only command it the “END”



A screenshot of a terminal window showing three lines of user input. The first line is `user > i want to buy a nintendo switch`, the second is `user > how many play station 5 are in stock?`, and the third is `user > |` with a cursor. The terminal window has the same title and command bar as the first screenshot.

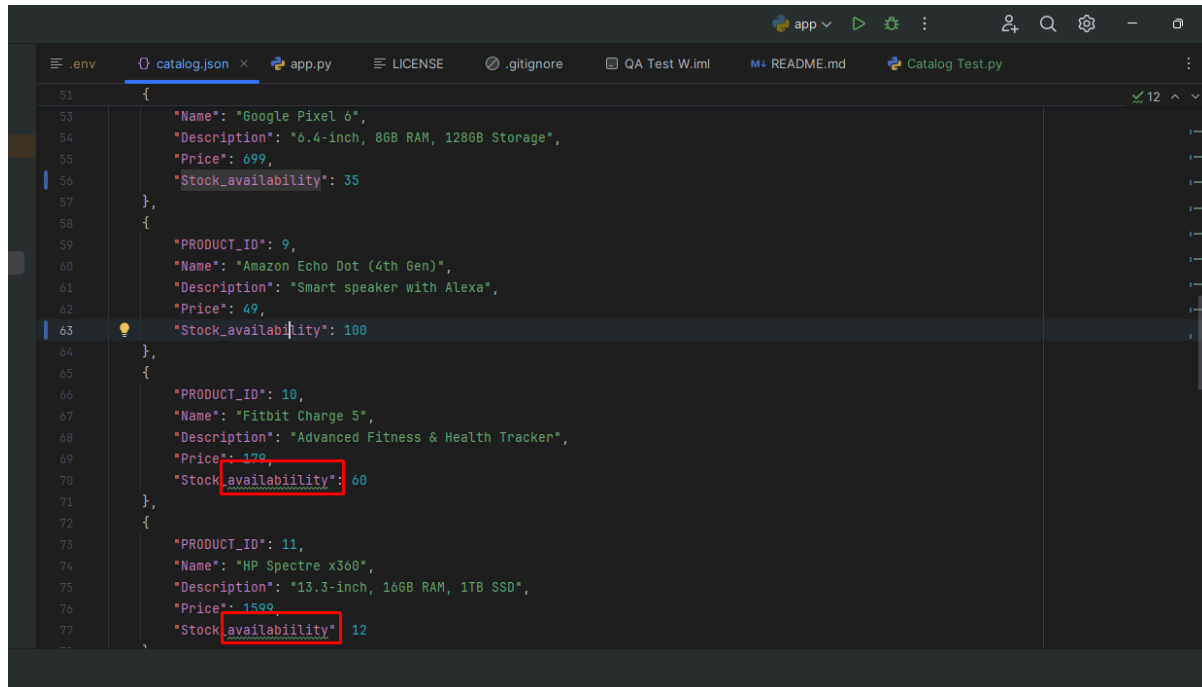


A screenshot of a terminal window showing the completion of the test. The third line of input is `user > END`. Below the input, the text `Chat is finished.` is displayed. At the bottom, a message states `Process finished with exit code 0`. The terminal window maintains the same title and command bar.

I try to chage the code to the test but that didnt work

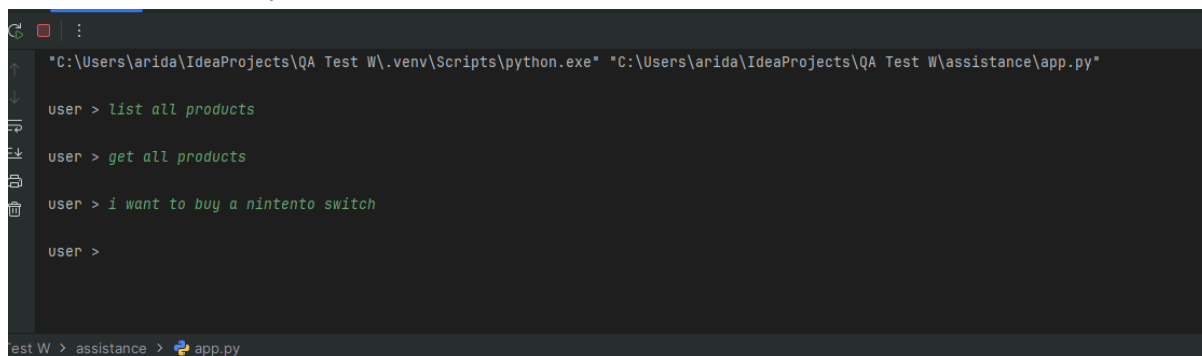
For example this change in "get_all_products"

After cloaned the repository i had a lot of problems with de API key, after i look a bug in the key "Stock_availability", it was written "Stock_availabiility"



```
51 {
52   "Name": "Google Pixel 6",
53   "Description": "6.4-inch, 8GB RAM, 128GB Storage",
54   "Price": 699,
55   "Stock_availability": 35
56 },
57 {
58   "PRODUCT_ID": 9,
59   "Name": "Amazon Echo Dot (4th Gen)",
60   "Description": "Smart speaker with Alexa",
61   "Price": 49,
62   "Stock_availability": 100
63 },
64 {
65   "PRODUCT_ID": 10,
66   "Name": "Fitbit Charge 5",
67   "Description": "Advanced Fitness & Health Tracker",
68   "Price": 179,
69   "Stock_availability": 60
70 },
71 {
72   "PRODUCT_ID": 11,
73   "Name": "HP Spectre x360",
74   "Description": "13.3-inch, 16GB RAM, 1TB SSD",
75   "Price": 1599,
76   "Stock_availability": 12
77 }
```

In the app.py code I have a problem using the assistance. Everything I wrote to him was not accepted and he only returned User >



```
"C:\Users\arida\IdeaProjects\QA Test W\.venv\Scripts\python.exe" "C:\Users\arida\IdeaProjects\QA Test W\assistance\app.py"

user > list all products

user > get all products

user > i want to buy a nintendo switch

user >
```

So I changed part of the code, because In the section where you define the tools, get_all_products has an error in the parameters. Currently it is defined like this

```
{
  "type": "function",
  "function": {
    "name": "get_all_products",
    "description": "Get all products",
```

```

    "parameters": {
      "type": "object",
      "properties": {
        "Name": {
          "description": "The name of the product",
          "type": "string"
        },
      },
    },
    "required": ["Name"],
    "additionalProperties": False
  },
}

```

For this: because get_all_products don't require parameters

```

{
  "type": "function",
  "function": {
    "name": "get_all_products",
    "description": "Get all products",
    "parameters": {
      "type": "object",
      "properties": {},
      "required": [],
      "additionalProperties": False
    },
  },
}

```

4. Testing Environment

- **Local:** Run in a virtual environment with the required dependencies installed.
- **Tool Simulation:** Use mocks or stubs to simulate the responses from the OpenAI API and the streams function, if necessary, for automated tests.

I use Pytest to simulate for the automation test and create a new python
 “test_assistance.py”

5. Test Cases

5.1. Catalog Loading

Objective:

Verify that the file *catalog.json* loads correctly and contains the expected products.

• Steps:

- Execute the script and check that the file opens without errors.
- Validate that the variable *catalog* contains a non-empty list and that each object has the fields: "Name", "Description", "Price", and "Stock_availability".

• Expected Outcome:

The catalog loads correctly and the products can be iterated over without errors.

i wrote the next script to test that.

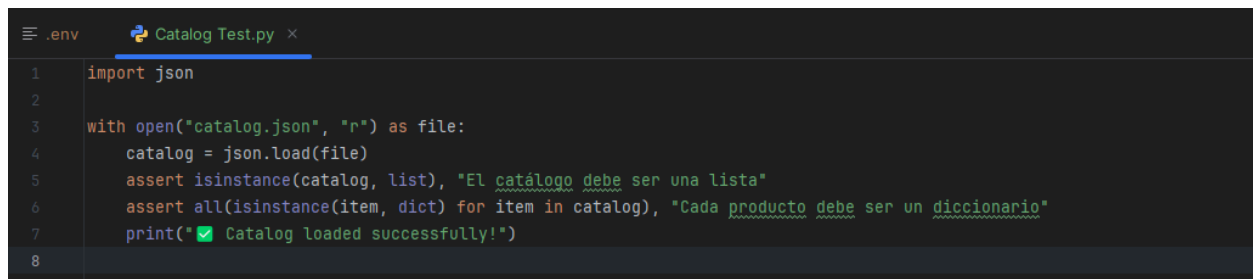
I imported *json*, which is a module for working with JSON files.

I opened the *catalog.json* file in read mode ("r") and loaded it into the *catalog* variable.

I opened the *catalog.json* file in read mode ("r") and loaded it into the *catalog* variable.

I checked that each item in the catalog is a dictionary (product).


and if everything is ok, I printed a success message.



```
.env Catalog Test.py x
1 import json
2
3 with open("catalog.json", "r") as file:
4     catalog = json.load(file)
5     assert isinstance(catalog, list), "El catálogo debe ser una lista"
6     assert all(isinstance(item, dict) for item in catalog), "Cada producto debe ser un diccionario"
7     print("✅ Catalog loaded successfully!")
8
```

and it works successfully

Run Catalog Test x

  :

```
"C:\Users\arida\IdeaProjects\QA Test W\.venv\Scripts\python.exe" "C:\Users\arida\IdeaProjects\QA Test W\assistance\Catalog Test.py"
```

```
✓ Catalog loaded successfully!
```

```
Process finished with exit code 0
```

