Task 3

Question:
Document your solution to task 2. The documentation must _briefly_ explain how to get it up & running. The target audience for this technical documentation are imaginary fellow developers, who would have to productize, evolve, support and maintain it.

Answer: Please see the instructions below and please see the instructions on how to execute **client.py**, all the unit and E2E tests mentioned below.

Step1:
✓**IDE: VSCODE**

Step2:
✓**Install Dependencies:**

pip install pytest   `# pytest` – test framework used to discover and run tests.
pip install httpx   `#httpx` – used in your actual code and mocking (`httpx.get()`).
pip install fastapi
pip install uvicorn   **# Optional: for running the server (not for testing)**
pip install requests   **# Optional: only if used somewhere**
pip install pytest
pip install pytest-mock   **# Enables use of `monkeypatch` as a fixture**
pip install pytest-asyncio

Step 3:

✓**Execute "client.py" python file:**

☐ **Step-by-step Instructions:**

1. **Ensure prerequisites are met:**
    o   Your **FastAPI** server (likely at `http://localhost:8000/`) must be running. So, **run the command** on terminal : docker compose up -d --build
    o   The server should require **Basic Auth** with username **"user"** and password **"password".**
    o   Python 3.10+ is installed. (In my case it is 3.13)
    o   **httpx** is installed.
2. **Install httpx (if not already):**

    ```
    pip install httpx
    ```

3. **Run the script**

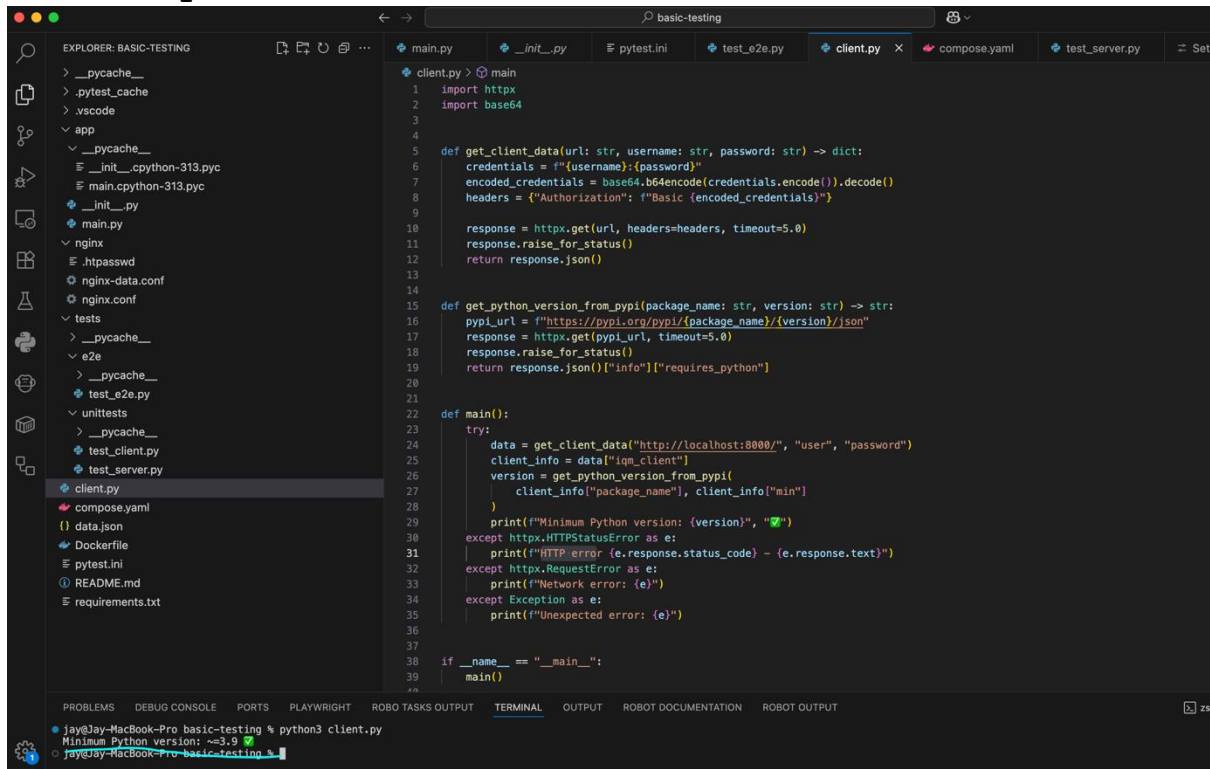   On your terminal, run the command:

```
Python3 client.py
```

Or, if you're inside a project folder:

```
Python3 path/to/client.py
```

☐ Output (if server returns expected data):

```
Minimum Python version: ~=3.9 ✓
```


✅Run the unit tests:

```
[jay@Jay-MacBook-Pro basic-testing % docker compose down
 [+] Running 4/4
  ⊞ Container data                     Removed                              0.3s
  ⊞ Container fastapi-application       Removed                              0.4s
  ⊞ Container nginx                     Removed                              0.3s
  ⊞ Network "basic-testing_default"    Removed                              0.1s
 jay@Jay-MacBook-Pro basic-testing %
```
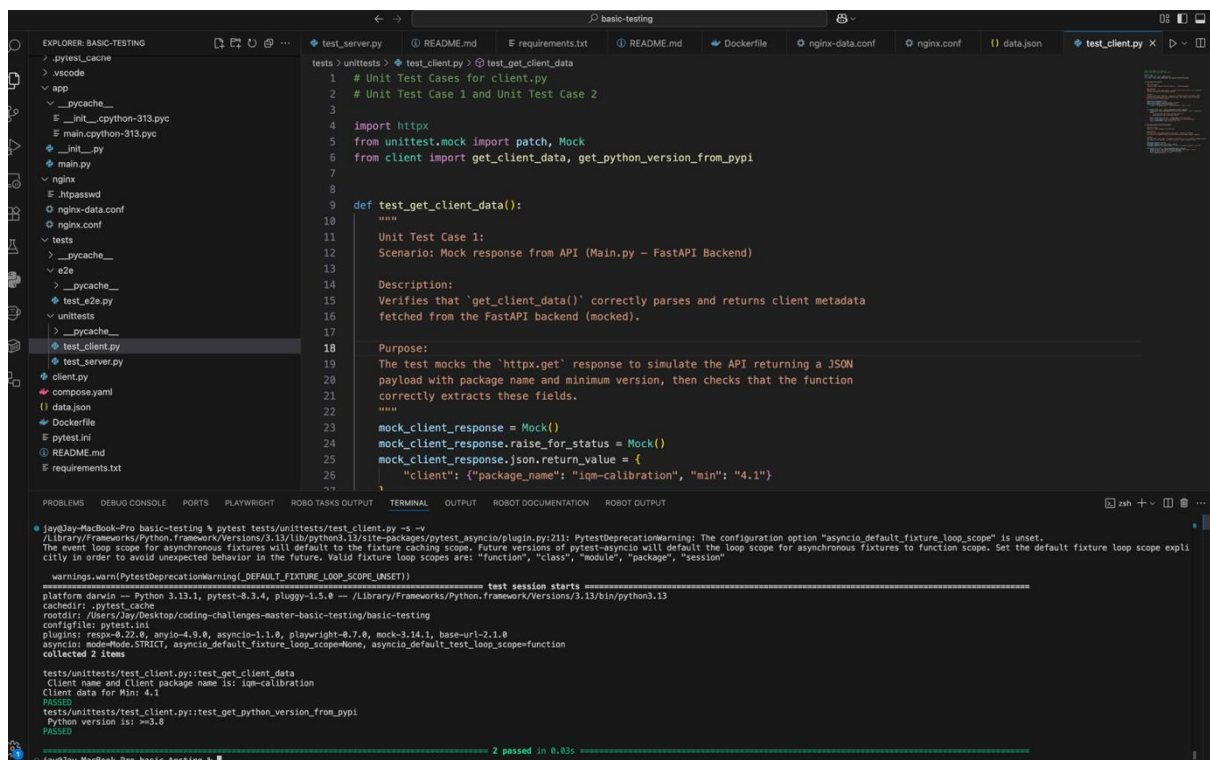
Note: Ensure, you do not run you FastAPI server (http://localhost:8000) as our unit tests are mocked and we don't need a real server.

✅**Run the unit test for client.py:**

- ✅**Test name**: **test_client.py**
- **Test Description** is described inside the tests as comments explaining manual steps.
- On your terminal, Run the unit test with command: **pytest tests/unittests/test_client.py -s -v**

Test Output:

☑️**Run the unit test for main.py (FastAPI Server):**

- ☑️**Test name**: test_server.py
- **Test Description** is described inside the tests as comments explaining manual steps.
- On your terminal, Run the unit test with command: **pytest tests/unittests/test_server.py -s -v**

Test Output:

✅Run the ==E2E or Integration tests==:

✅ **Pre-requisite:**

- Your **FastAPI** server (likely at http://localhost:8000/) must be up and running. So, please **run the command** on terminal: **docker compose up -d --build**

```
[jay@Jay-MacBook-Pro basic-testing % docker compose down
[+] Running 4/4
 ⋮ Container fastapi-application    Removed                                                              0.4s
 ⋮ Container nginx                  Removed                                                              0.3s
 ⋮ Container data                   Removed                                                              0.3s
 ⋮ Network "basic-testing_default"  Removed                                                              0.1s
[jay@Jay-MacBook-Pro basic-testing % docker compose up -d --build
[+] Building 4.4s (14/14) FINISHED
 => [internal] load build definition from Dockerfile                                                     0.0s
 => => transferring dockerfile: 79B                                                                      0.0s
 => [internal] load .dockerignore                                                                        0.0s
 => => transferring context: 2B                                                                          0.0s
 => resolve image config for docker.io/docker/dockerfile:1.4                                             2.1s
 => CACHED docker-image://docker.io/docker/dockerfile:1.4@sha256:9ba7531bd80fb0a858632727cf7a112fbfd19b17e94c4e84ced81e24ef1a0dbc   0.0s
 => [internal] load .dockerignore                                                                        0.0s
 => [internal] load build definition from Dockerfile                                                     0.0s
 => [internal] load metadata for docker.io/tiangolo/uvicorn-gunicorn-fastapi:python3.9-slim             1.9s
 => [builder 1/5] FROM docker.io/tiangolo/uvicorn-gunicorn-fastapi:python3.9-slim@sha256:ac5dc853b902c02a52a0d5da9666d29d73d16da04322bab99d79  0.0s
 => [internal] load build context                                                                       0.0s
 => => transferring context: 2.12kB                                                                      0.0s
 => CACHED [builder 2/5] WORKDIR /app                                                                    0.0s
 => CACHED [builder 3/5] COPY requirements.txt ./                                                        0.0s
 => CACHED [builder 4/5] RUN --mount=type=cache,target=/root/.cache/pip    pip install -r requirements.txt   0.0s
 => [builder 5/5] COPY ./app ./app                                                                       0.0s
 => exporting to image                                                                                   0.0s
 => => exporting layers                                                                                  0.0s
 => => writing image sha256:088dc1ba0bdb54f7be56c9c1e9d51792c02a8892ac76262fa547fc1a82776265            0.0s
 => => naming to docker.io/library/basic-testing_api                                                     0.0s
Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
[+] Running 4/4
 ⋮ Network "basic-testing_default"  Created
 ⋮ Container nginx                  Started
 ⋮ Container fastapi-application    Started
 ⋮ Container data                   Started
jay@Jay-MacBook-Pro basic-testing % docker ps -a
CONTAINER ID   IMAGE           COMMAND                  CREATED        STATUS         PORTS                    NAMES
b2d5f3e1d73e   nginx:alpine    "/docker-entrypoint.…"   6 minutes ago  Up 6 minutes   0.0.0.0:8000->80/tcp     nginx
d7dbc7a0c842   nginx:alpine    "/docker-entrypoint.…"   6 minutes ago  Up 6 minutes   0.0.0.0:8002->80/tcp     data
[4307ab35c9d4   basic-testing_api  "/start.sh"           6 minutes ago  Up 6 minutes   0.0.0.0:8001->8000/tcp   fastapi-application
jay@Jay-MacBook-Pro basic-testing % █
```
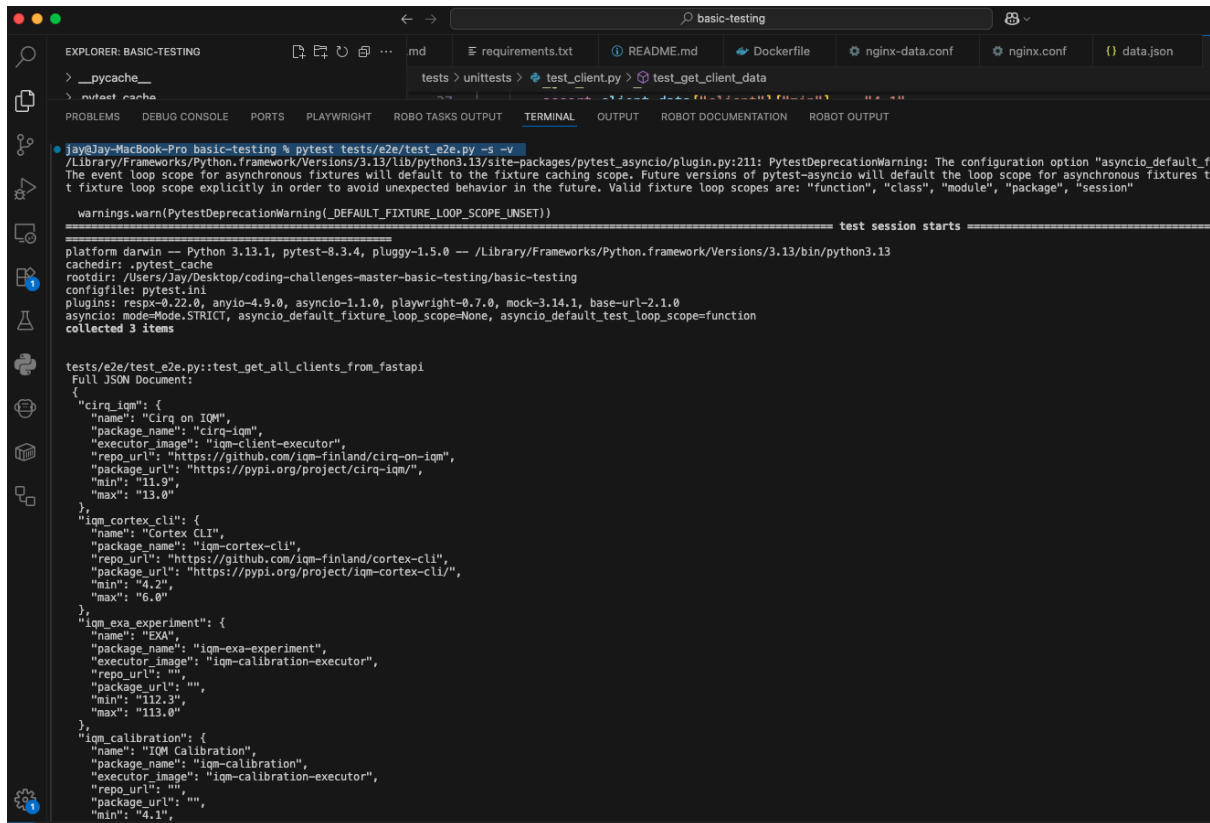
Once, your **FastAPI server** is up and running, open a new browser type **http://localhost:8000/** and type username as "user" and password as "password". Now, hit "Sign In" Button. You should see the web page displaying full **JSON document** like mentioned below.

```
←  →  C  ⓘ  localhost:8000
```

Sign in

http://localhost:8000

Username | user
Password | ••••••••

Cancel   Sign In
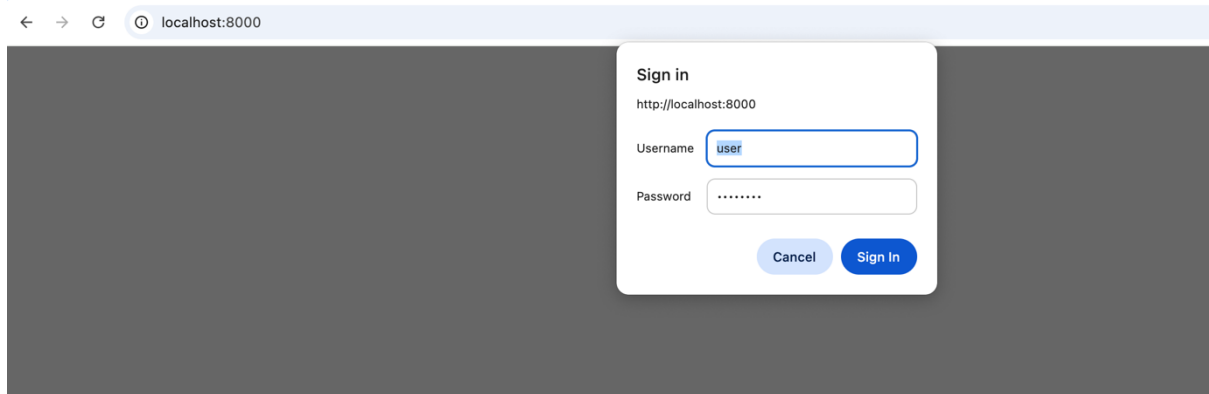
```
←  →  C  ⓘ  localhost:8000
```

Pretty print ☑

{
    "cirq_iqm": {
        "name": "Cirq on IQM",
        "package_name": "cirq-iqm",
        "executor_image": "iqm-client-executor",
        "repo_url": "https://github.com/iqm-finland/cirq-on-iqm",
        "package_url": "https://pypi.org/project/cirq-iqm/",
        "min": "11.9",
        "max": "13.0"
    },
    "iqm_cortex_cli": {
        "name": "Cortex CLI",
        "package_name": "iqm-cortex-cli",
        "repo_url": "https://github.com/iqm-finland/cortex-cli",
        "package_url": "https://pypi.org/project/iqm-cortex-cli/",
        "min": "4.2",
        "max": "6.0"
    },
    "iqm_exa_experiment": {
        "name": "EXA",
        "package_name": "iqm-exa-experiment",
        "executor_image": "iqm-calibration-executor",
        "repo_url": "",
        "package_url": "",
        "min": "112.3",
        "max": "113.0"
    },
    "iqm_calibration": {
        "name": "IQM Calibration",
        "package_name": "iqm-calibration",
        "executor_image": "iqm-calibration-executor",
        "repo_url": "",
        "package_url": "",
        "min": "4.1",
        "max": "4.2"
    },
    "iqm_client": {
        "name": "IQM client",
        "package_name": "iqm-client",
        "executor_image": "iqm-client-executor",
        "repo_url": "https://github.com/iqm-finland/iqm-client",
        "package_url": "https://pypi.org/project/iqm-client/",
        "min": "13.2",
        "max": "15.0"
    },
    "qiskit_iqm": {
        "name": "Qiskit on IQM",
        "package_name": "qiskit-iqm",
        "executor_image": "iqm-client-executor",
        "repo_url": "https://github.com/iqm-finland/qiskit-on-iqm",
        "package_url": "https://pypi.org/project/qiskit-iqm/",
        "min": "10.9",
        "max": "12.0"
    }
}
```

- **Test name:** `test_e2e.py`
- **Test Description** is described inside the tests as comments explaining manual steps.
- On your terminal, Run the Integration (OR) E2E tests with command:
  **pytest tests/e2e/test_e2e.py -s -v  (OR) PYTHONPATH=. pytest -s -v tests/e2e/test_e2e.py -s -v**

# Test Output:

```
jay@Jay-MacBook-Pro basic-testing % pytest tests/e2e/test_e2e.py -s -v
/Library/Frameworks/Python.framework/Versions/3.13/lib/python3.13/site-packages/pytest_asyncio/plugin.py:211: PytestDeprecationWarning: The configuration option "asyncio_default_f
The event loop scope for asynchronous fixtures will default to the fixture caching scope. Future versions of pytest-asyncio will default the loop scope for asynchronous fixtures to
t fixture loop scope explicitly in order to avoid unexpected behavior in the future. Valid fixture loop scopes are: "function", "class", "module", "package", "session"

  warnings.warn(PytestDeprecationWarning(_DEFAULT_FIXTURE_LOOP_SCOPE_UNSET))

================================================================================ test session starts ================================================================================
platform darwin -- Python 3.13.1, pytest-8.3.4, pluggy-1.5.0 -- /Library/Frameworks/Python.framework/Versions/3.13/bin/python3.13
cachedir: .pytest_cache
rootdir: /Users/Jay/Desktop/coding-challenges-master-basic-testing/basic-testing
configfile: pytest.ini
plugins: respx-0.22.0, anyio-4.9.0, asyncio-1.1.0, playwright-0.7.0, mock-3.14.1, base-url-2.1.0
asyncio: mode=Mode.STRICT, asyncio_default_fixture_loop_scope=None, asyncio_default_test_loop_scope=function
collected 3 items

tests/e2e/test_e2e.py::test_get_all_clients_from_fastapi
  Full JSON Document:
  {
    "cirq_iqm": {
      "name": "Cirq on IQM",
      "package_name": "cirq-iqm",
      "executor_image": "iqm-client-executor",
      "repo_url": "https://github.com/iqm-finland/cirq-on-iqm",
      "package_url": "https://pypi.org/project/cirq-iqm/",
      "min": "11.9",
      "max": "13.0"
    },
    "iqm_cortex_cli": {
      "name": "Cortex CLI",
      "package_name": "iqm-cortex-cli",
      "repo_url": "https://github.com/iqm-finland/cortex-cli",
      "package_url": "https://pypi.org/project/iqm-cortex-cli/",
      "min": "4.2",
      "max": "6.0"
    },
    "iqm_exa_experiment": {
      "name": "EXA",
      "package_name": "iqm-exa-experiment",
      "executor_image": "iqm-calibration-executor",
      "repo_url": "",
      "package_url": "",
      "min": "112.3",
      "max": "113.0"
    },
    "iqm_calibration": {
      "name": "IQM Calibration",
      "package_name": "iqm-calibration",
      "executor_image": "iqm-calibration-executor",
      "repo_url": "",
      "package_url": "",
      "min": "4.1",
```

```
11  AUTH = ("user", "password")
12
13
14  @pytest.mark.asyncio
15  async def test_get_all_clients_from_fastapi():
16      """
17      E2E Test Case 1:
18      Scenario: Request all client metadata from FastAPI through NGINX
19
20      Description:
21      This test checks if the FastAPI server correctly returns all available client metadata when accessed at the root endpoint
22      It also verifies that the response:
23          * Has HTTP 200 status code
24          * Includes expected keys (iqm_client and its package_name)
25          * Matches exactly with the expected data defined in data.json
26
27      Purpose:
28      To ensure the FastAPI server returns the full and correct client metadata structure as configured.
29      """
30      async with httpx.AsyncClient(base_url=BASE_URL, auth=AUTH) as client:
31          # Load the expected data from data.json
32          with open("data.json") as f:
33              expected_data = json.load(f)
34
```

```
      "executor_image": "iqm-client-executor",
      "repo_url": "https://github.com/iqm-finland/iqm-client",
      "package_url": "https://pypi.org/project/iqm-client/",
      "min": "13.2",
      "max": "15.0"
    },
    "qiskit_iqm": {
      "name": "Qiskit on IQM",
      "package_name": "qiskit-iqm",
      "executor_image": "iqm-client-executor",
      "repo_url": "https://github.com/iqm-finland/qiskit-on-iqm",
      "package_url": "https://pypi.org/project/qiskit-iqm/",
      "min": "10.9",
      "max": "12.0"
    }
  }

 All clients fetched successfully. ✅
PASSED
tests/e2e/test_e2e.py::test_get_specific_client_metadata
 Specific client metadata (iqm_calibration) fetched successfully. ✅
PASSED
tests/e2e/test_e2e.py::test_client_fastapi_interaction
 client.py successfully fetched metadata from FastAPI. ✅
PASSED

============================= 3 passed in 0.68s =============================
jay@Jay-MacBook-Pro basic-testing %
```