Coding with AI

Alessandro Bocci

`name.surname@unipi.it`

Advanced Software Engineering (Lab)

26/09/2025

What will you do?

- Setting up your AI environment in VS Code with GitHub.
- Make the AI generate the code for a Python service.

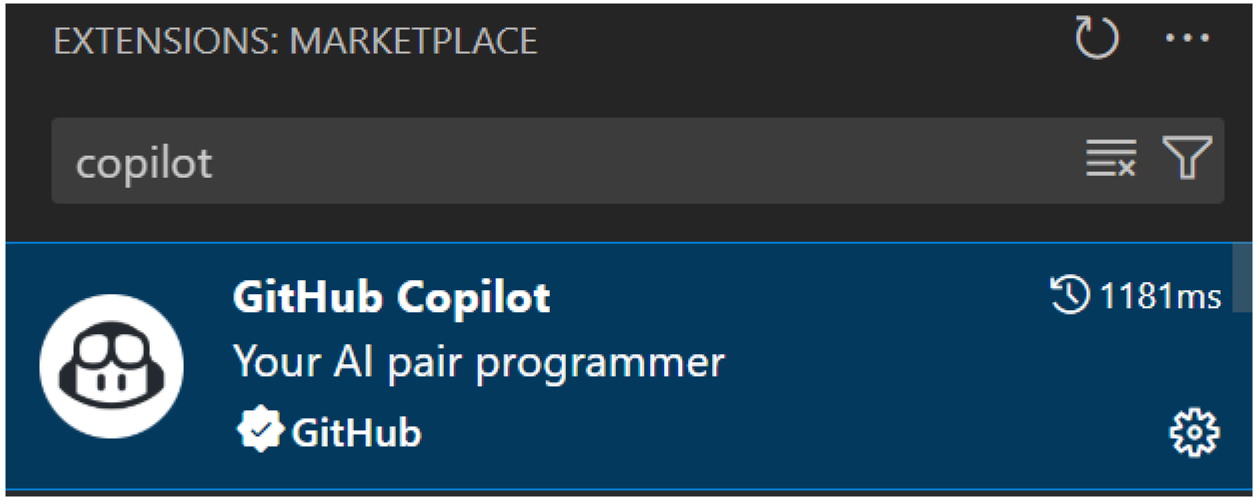
Checklist

- Download the material folder on the Moodle website.
- Install **Python 3.12** [https://www.python.org/]
- Install an **Visual Studio Code** [https://code.visualstudio.com/]
- Acquire **GitHub PRO** through Unipi [https://it.unipi.it/risorse/software/github-enterprise/]

Why Github Pro?

Because we need GitHub Copilot.

From the market place extension of Github PRO:



EXTENSIONS: MARKETPLACE

copilot

GitHub Copilot 1181ms

Your AI pair programmer

GitHub

Copilot

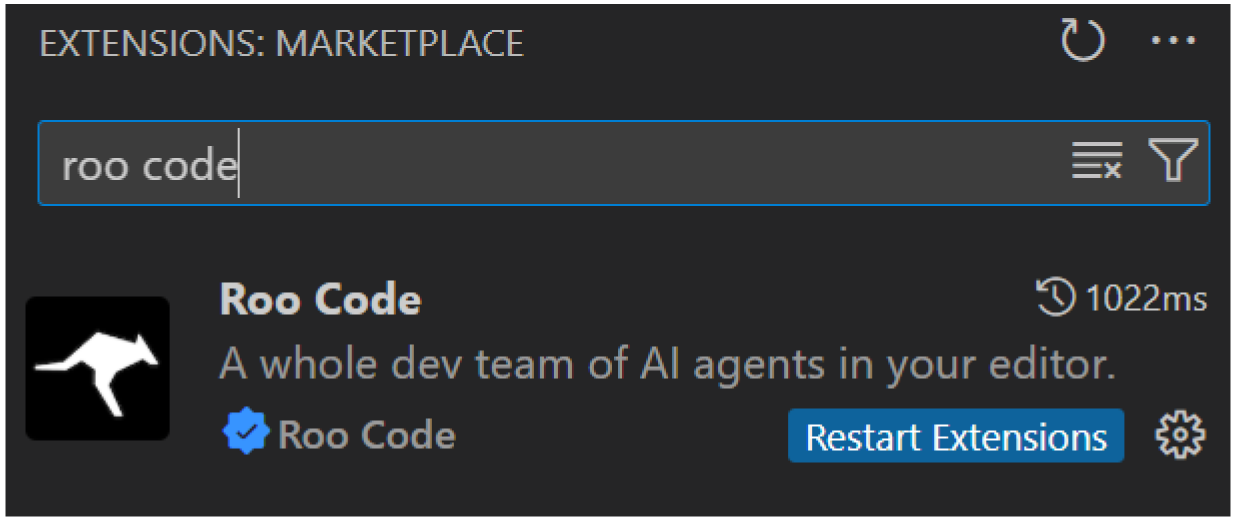
Set it up by linking your GitHub PRO account to Copilot in VS code.

https://code.visualstudio.com/docs/copilot/setup

If you do not have GitHub PRO, wait a bit: you will use ChatGPT (or another AI provider you want).

Install Roo Code plugin for VS code

If you have GitHub PRO:



What is Roo Code?

ROOCODE

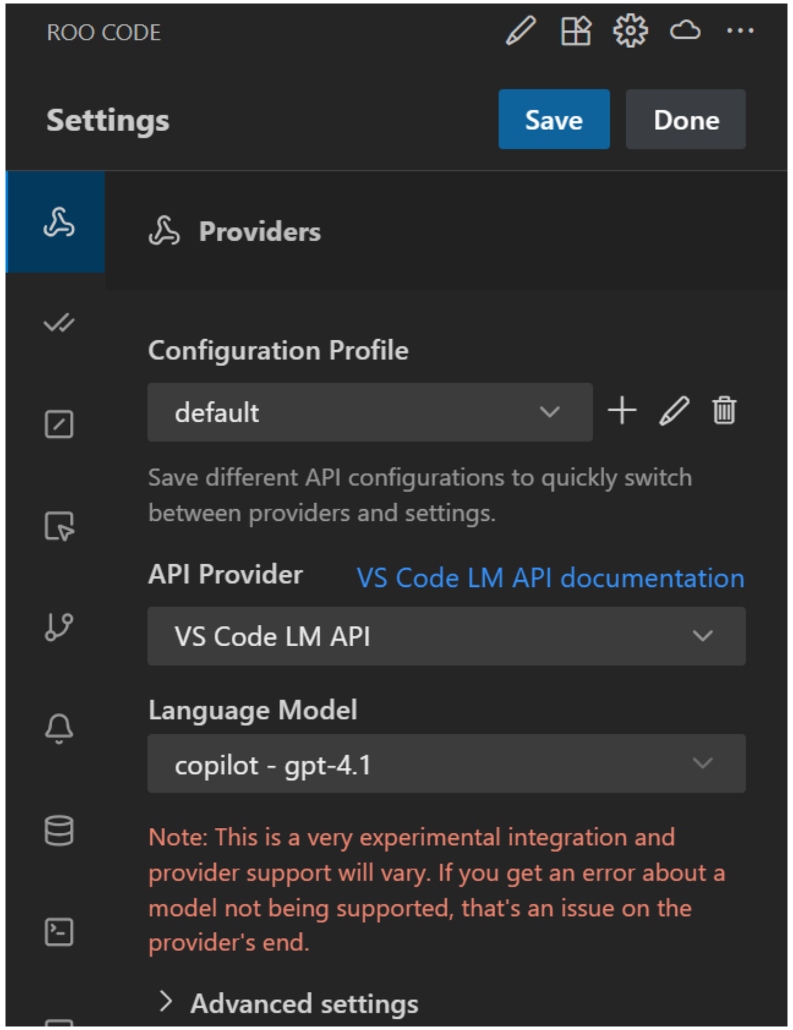
Roo Code is an AI-powered autonomous **coding agent** that lives in your editor.

It integrates with various AI models and providers, allowing you to connect to services like OpenAI, Google Gemini, Anthropic, OpenRouter, and local LLMs via platforms like LM Studio or Ollama.

We will use Copilot as an AI service provider for Roo Code to avoid paying for the service.

IN SUPREMÆ DIGNITATIS 1343

Setting up Roo Code



Settings

Providers

Configuration Profile

default

Save different API configurations to quickly switch between providers and settings.

API Provider VS Code LM API documentation

VS Code LM API

Language Model

copilot - gpt-4.1

Note: This is a very experimental integration and provider support will vary. If you get an error about a model not being supported, that's an issue on the provider's end.

Advanced settings

As Language Model you have plenty of choice.

But… the newest models have a limit of requests set by Github (around 200 per month).

Using gpt-4.1 currently is unlimeted. It performs worse than other models but for today is enough ☺

Now we are ready to ‘code’!

The task of today is develop a Python service using Flask.

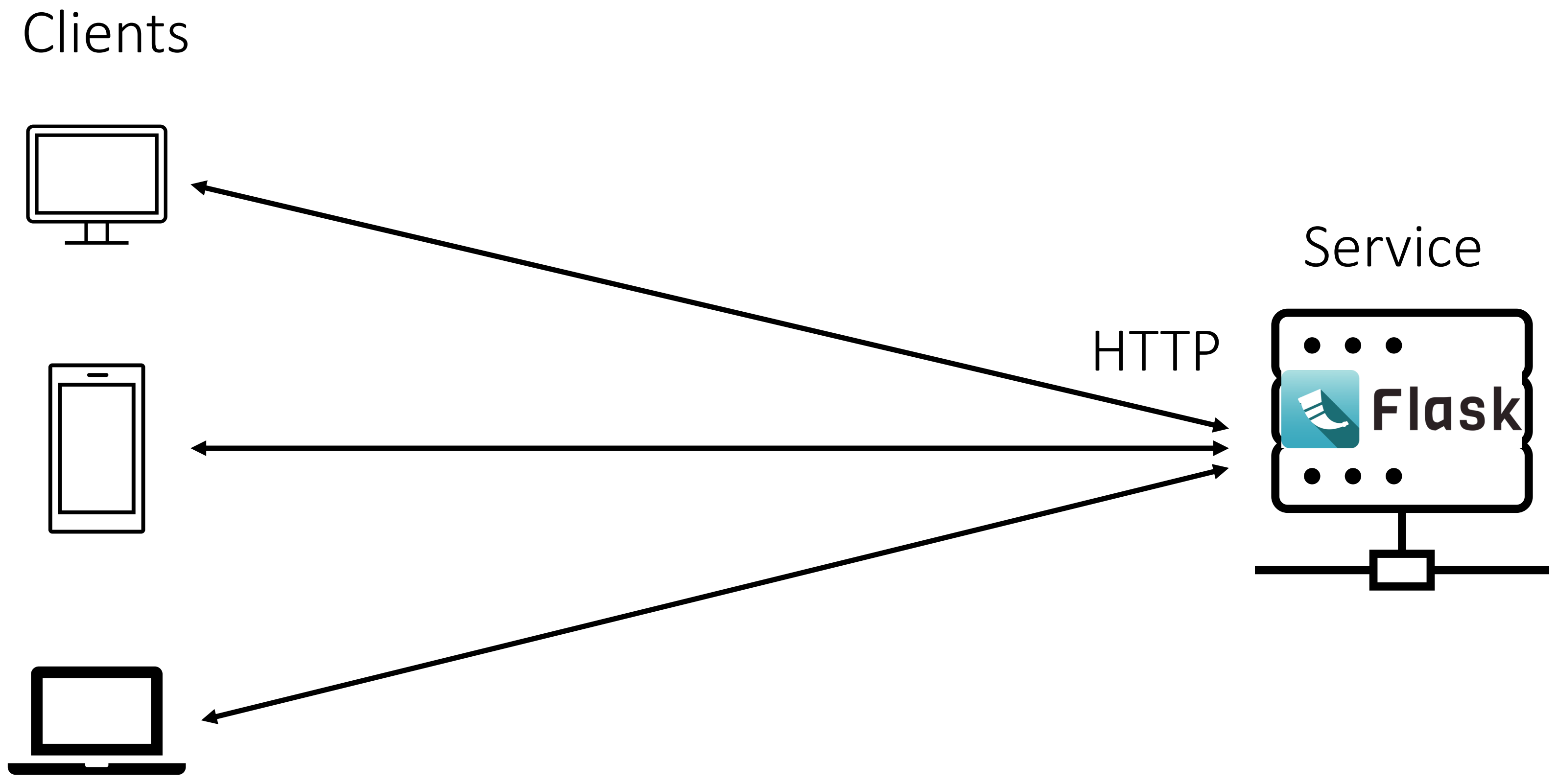
The service should perform operations from HTTP requests.

We will **not** write a web page!

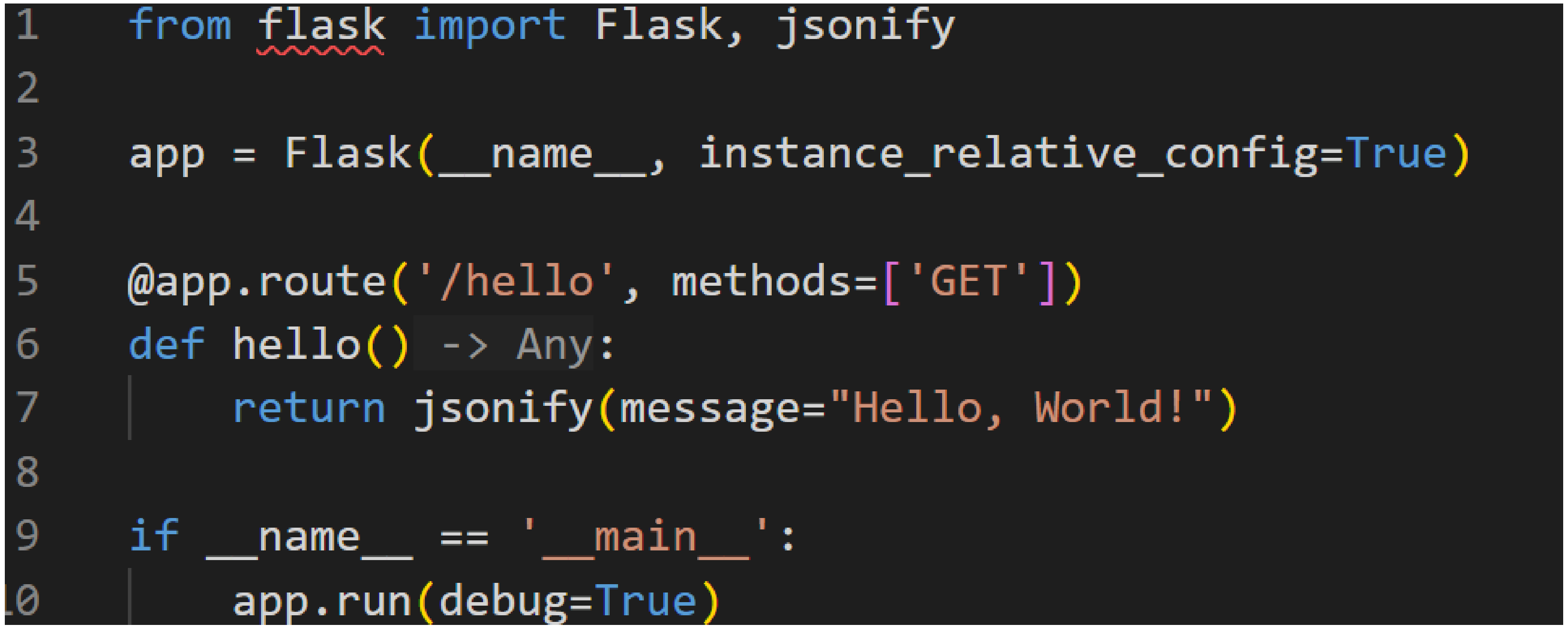
Download from the course website the `app.py` file to follow the example.

Let’s code a web service - Flask

Flask is a web application (service) framework written in Python.



Coding example

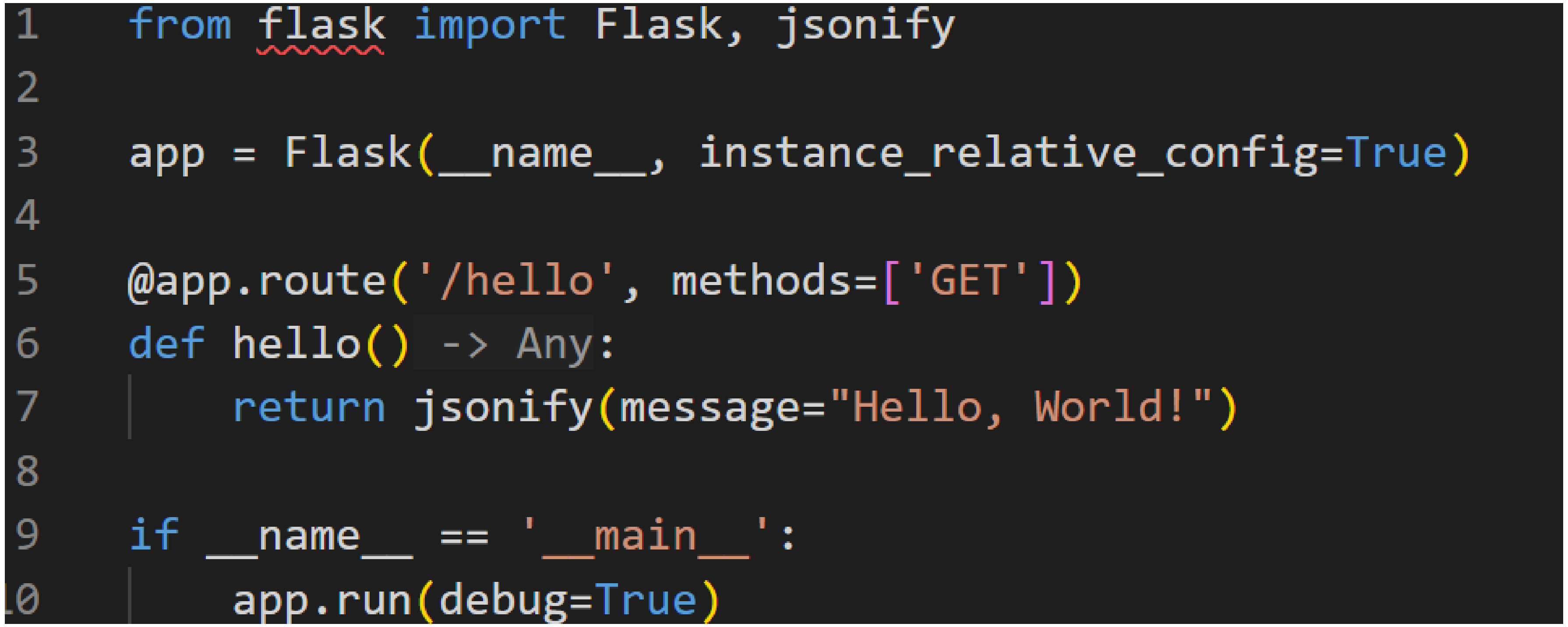


```python
from flask import Flask, jsonify

app = Flask(__name__, instance_relative_config=True)

@app.route('/hello', methods=['GET'])
def hello() -> Any:
    return jsonify(message="Hello, World!")

if __name__ == '__main__':
    app.run(debug=True)
```

Python code to generate an HTTP GET response to

`http://<service_host>/hello`

Coding example



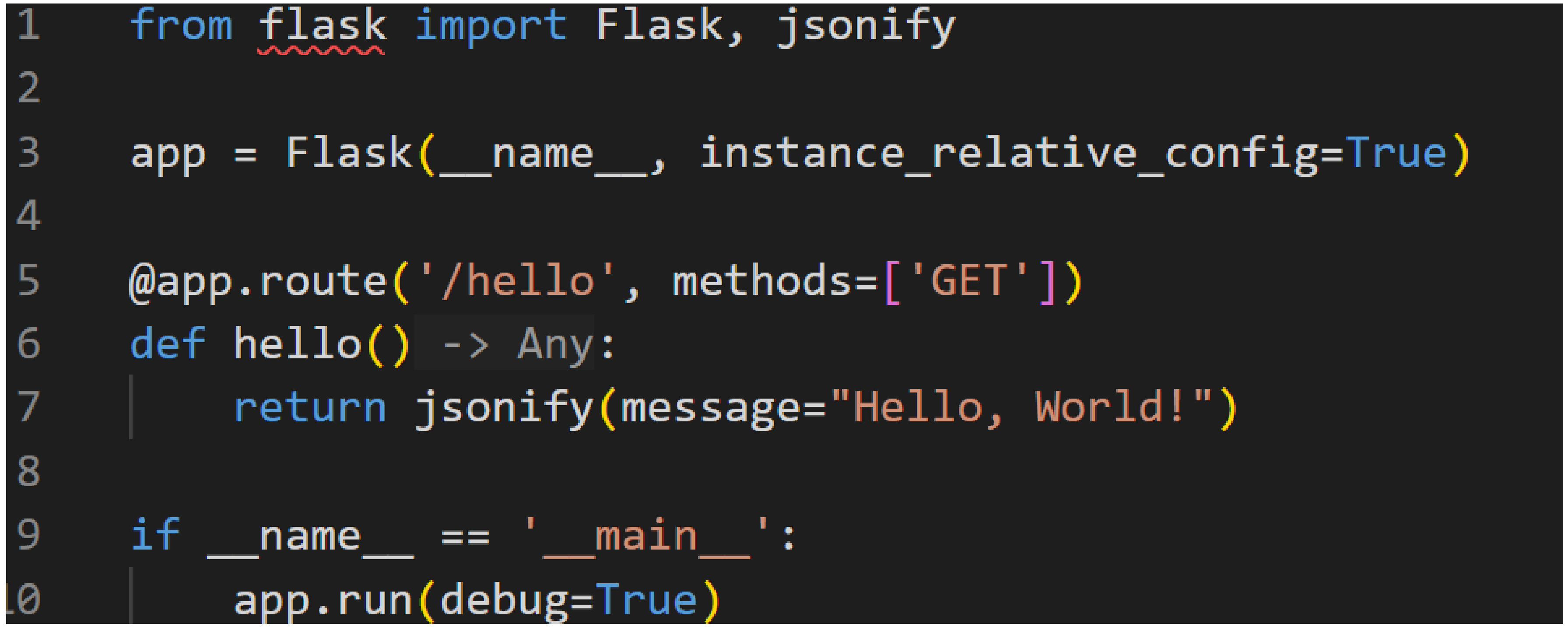
```python
from flask import Flask, jsonify

app = Flask(__name__, instance_relative_config=True)

@app.route('/hello', methods=['GET'])
def hello() -> Any:
    return jsonify(message="Hello, World!")

if __name__ == '__main__':
    app.run(debug=True)
```

Route: specifies the path (`/hello`) and can specify the HTTP method (default = GET). It is a Flask annotation.

A route is and endpoint part of the REST API of our service.

`http://<service_host>/hello`

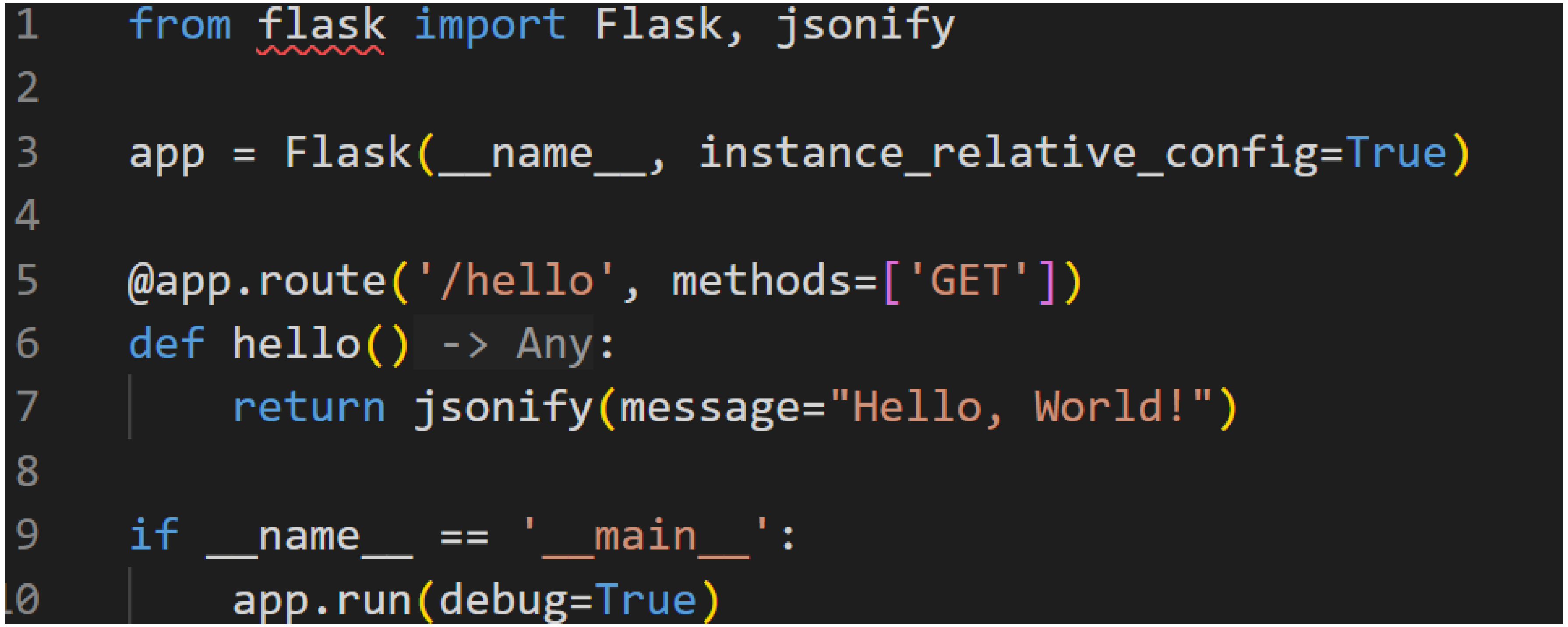
Coding example



```python
from flask import Flask, jsonify

app = Flask(__name__, instance_relative_config=True)

@app.route('/hello', methods=['GET'])
def hello() -> Any:
    return jsonify(message="Hello, World!")

if __name__ == '__main__':
    app.run(debug=True)
```

The annotated function is called when arrives a request for the route.

Coding example



```python
from flask import Flask, jsonify

app = Flask(__name__, instance_relative_config=True)

@app.route('/hello', methods=['GET'])
def hello() -> Any:
    return jsonify(message="Hello, World!")

if __name__ == '__main__':
    app.run(debug=True)
```

A JSON containing the message is generated and sent back to the requester.

How to run a Flask service

From the terminal (inside an activated venv):

- Install the service requirements `pip install flask`
- Run the service `flask run --host=0.0.0.0 --port=5005`

  Now the service will be waiting on *any* network interface (0.0.0.0) at *port* 5005.

  The terminal will log the events of the service, to close the service and take control of the terminal ctrl+C (Command+C).

  If the 5005 port is used by another service of your system, you can change it.

- Open a browser and visit `http://localhost:5005/hello`

You should see the result of the execution of the code.

By default, you have to call the Python file `app.py`.

Now it’s your turn

Use generative AI (Roo Code, ChatGPT, whatever) to create a Flask service that implements a simple calculator.

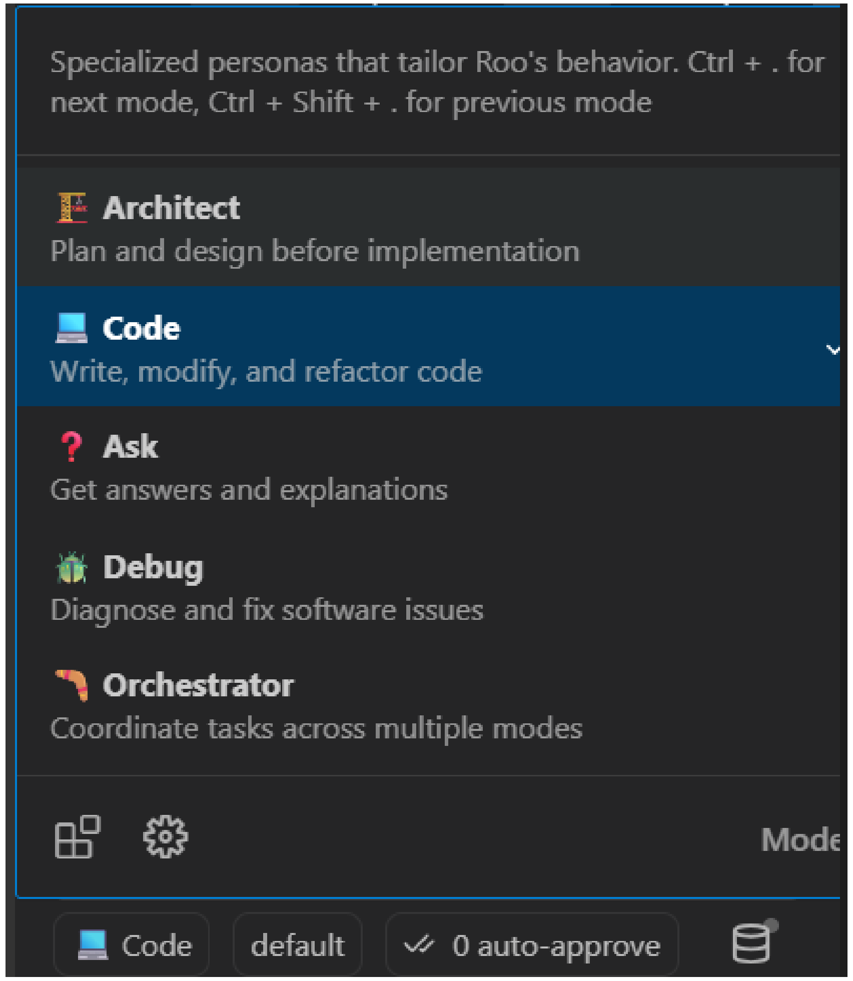
It should have the operations to:

- Add two numbers.
- Subtract two numbers.
- Multiply two numbers.
- Divide two numbers.

Roo Code

Open VS Code in a folder.
In the Roo Code tab, you can select the Mode from:

- Architect.
- Code.
- Ask.
- Debug.
- Orchestrator.



Specialized personas that tailor Roo's behavior. Ctrl + . for next mode, Ctrl + Shift + . for previous mode

Architect
Plan and design before implementation

Code
Write, modify, and refactor code

Ask
Get answers and explanations

Debug
Diagnose and fix software issues

Orchestrator
Coordinate tasks across multiple modes

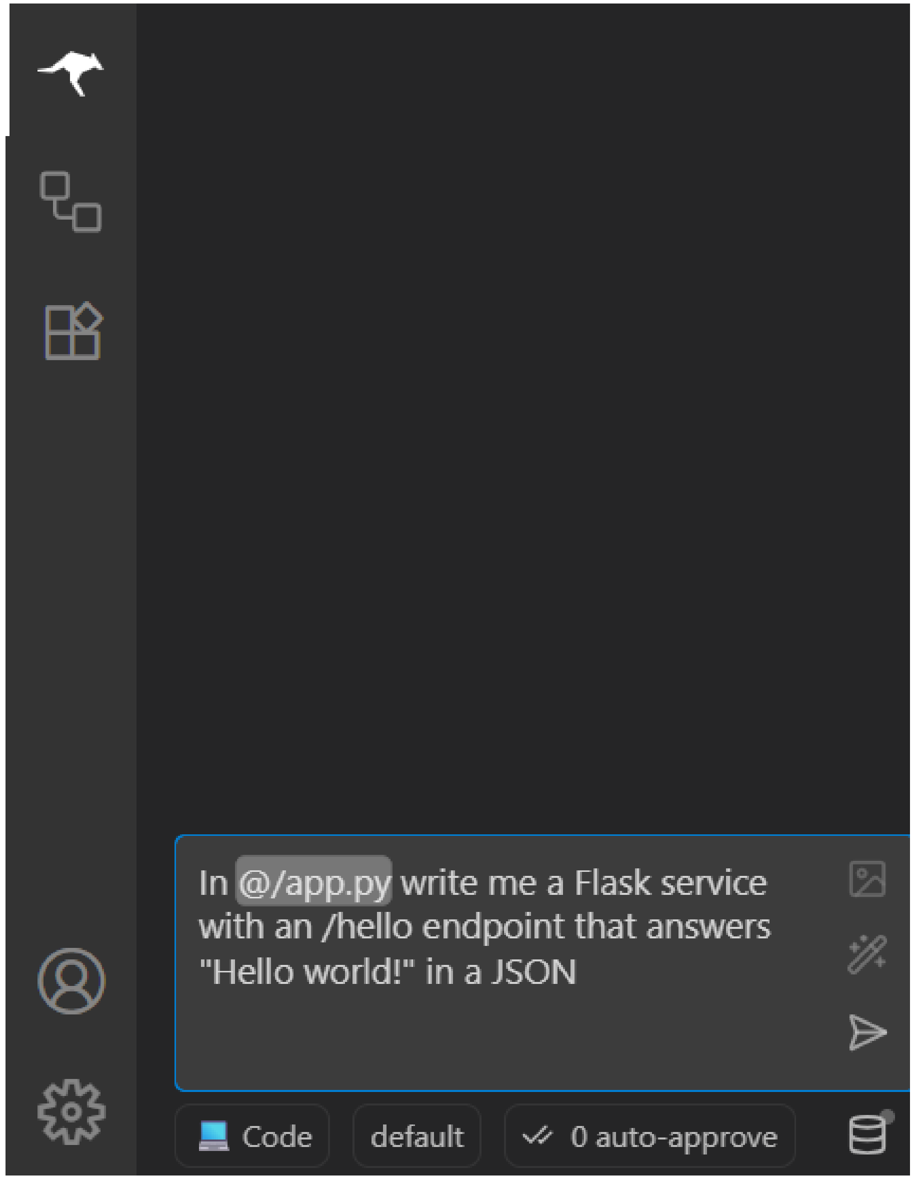
(more info: https://docs.roocode.com/basic-usage/using-modes)

Roo Code

You can write the prompt and it can:

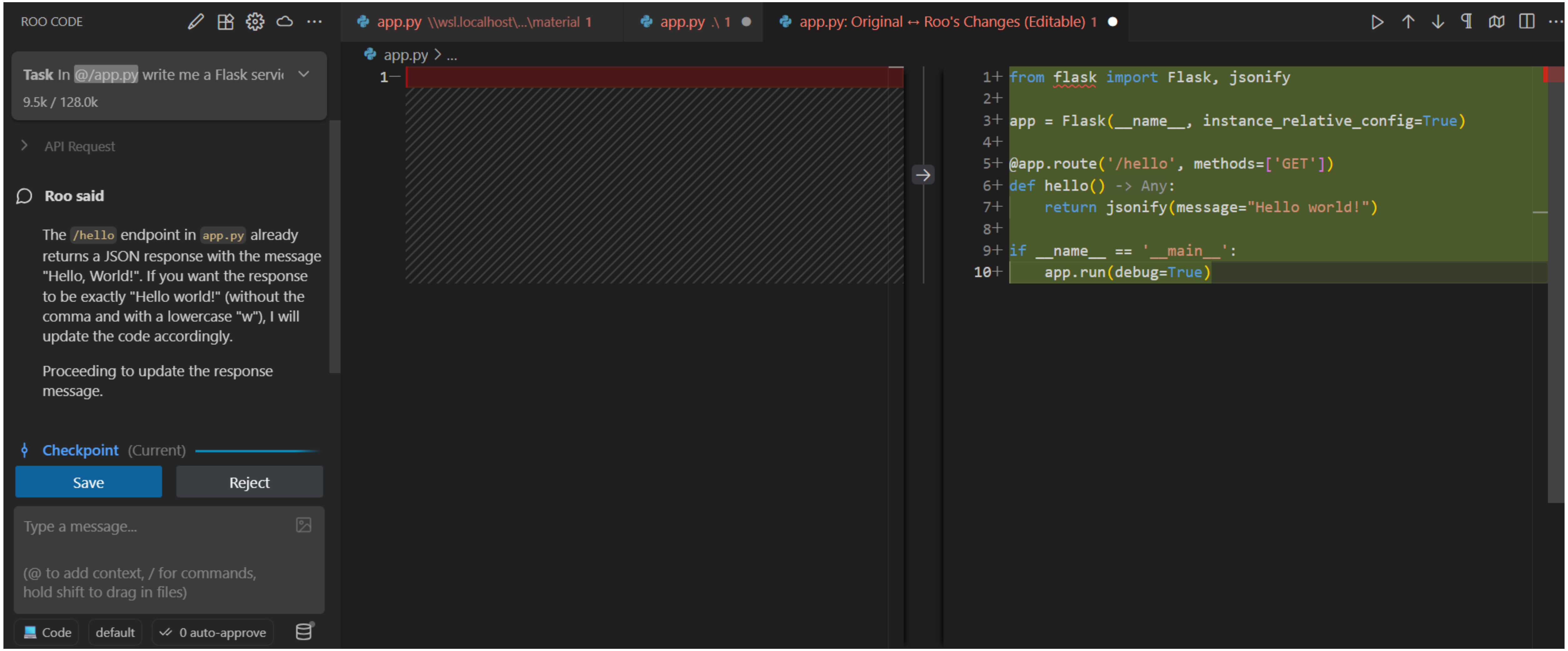
- Read the files in the folder, you can tag them with @.
- Write files.
- Change mode.
- Answer to your questions.
- Execute command in a terminal.
- Etc.

In general it ask for your approval to the operations but you can set the auto approval for some of them.



In @/app.py write me a Flask service with an /hello endpoint that answers "Hello world!" in a JSON

Roo Code


ROO CODE
app.py \\wsl.localhost\...\material 1
app.py .\ 1
app.py: Original ↔ Roo's Changes (Editable) 1
app.py > ...
API Request
Save
Reject
Code
default
0 auto-approve

Task In @/app.py write me a Flask servi

9.5k / 128.0k

Roo said

The `/hello` endpoint in `app.py` already returns a JSON response with the message "Hello, World!". If you want the response to be exactly "Hello world!" (without the comma and with a lowercase "w"), I will update the code accordingly.

Proceeding to update the response message.

Checkpoint (Current)

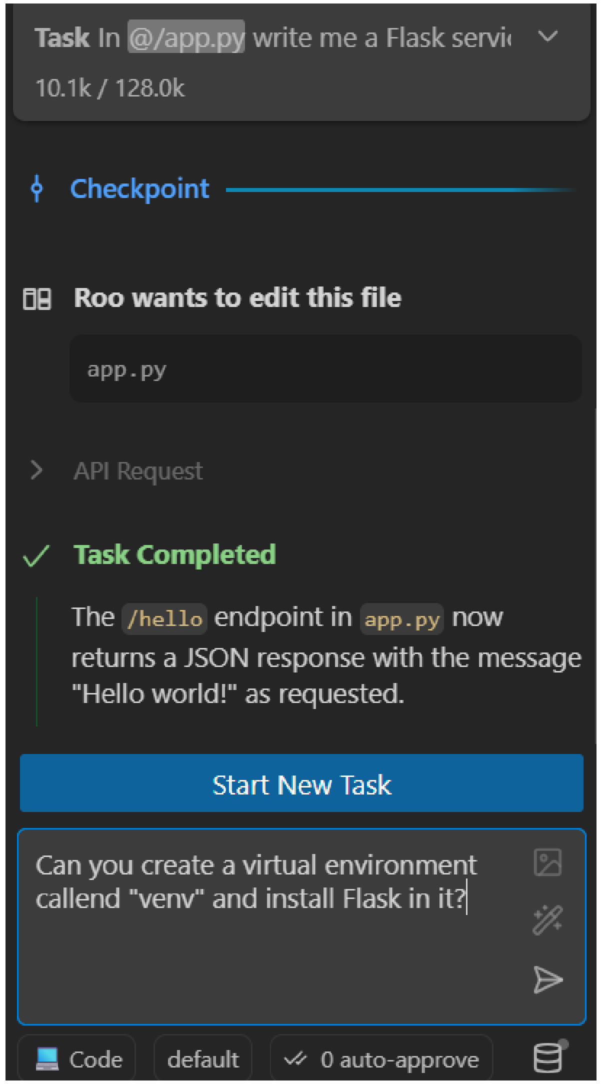
Type a message...

(@ to add context, / for commands, hold shift to drag in files)

```python
from flask import Flask, jsonify

app = Flask(__name__, instance_relative_config=True)

@app.route('/hello', methods=['GET'])
def hello() -> Any:
    return jsonify(message="Hello world!")

if __name__ == '__main__':
    app.run(debug=True)
```

You see the write proposal: on the left the current (empty) file and on the right what Roo Code wants to write. Press ‘Save’ to approve.

Roo Code



Task In @/app.py write me a Flask servi

10.1k / 128.0k

Checkpoint

Roo wants to edit this file

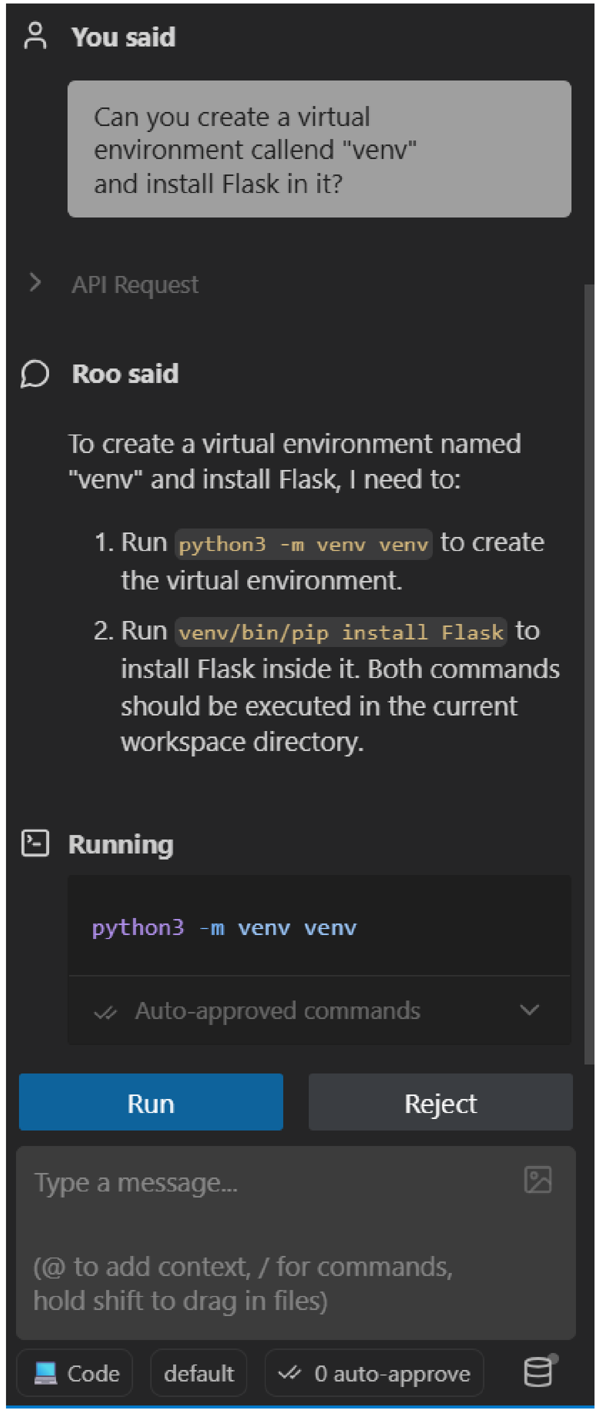
```
app.py
```

API Request

Task Completed

The `/hello` endpoint in `app.py` now returns a JSON response with the message "Hello world!" as requested.

Can you create a virtual environment callend "venv" and install Flask in it?



You said

Can you create a virtual environment callend "venv" and install Flask in it?

API Request

Roo said

To create a virtual environment named "venv" and install Flask, I need to:

1. Run `python3 -m venv venv` to create the virtual environment.
2. Run `venv/bin/pip install Flask` to install Flask inside it. Both commands should be executed in the current workspace directory.

Running

```
python3 -m venv venv
```

If you use the WSL, you need to open the folder and VS Code in the WSL terminal to run commands.

From the folder
`code .`

Roo Code

You can also ask to make the HTTP requests to the service to test it. In general, it uses curl.

The full docs are at https://docs.roocode.com/

Play with it to discover its features!

Hint: Try to start in Architect mode.

Roo Code

Remember: always check before approve write or executing!

You can use auto-approve for the reading.

No Roo Code?

If you cannot access GitHub Pro use another AI service provider.

They are not integrated with VS code, so you have to copy and paste the code and run the commands by yourself.

Bonus work

Add new endpoints to the REST API

- `/upper` which given a string returns it in a JSON, all in uppercase.
- `/lower` which given a string returns it in a JSON, all in lowercase.
- `/concat` which given two strings returns in a JSON the concatenation of them.
- `/reduce` which takes the operator (one of add, sub, mul, div, concat) and a list string representing a list, apply the operator to all the elements giving the result. For instance, with `add` and `[2,1,3,4]` it returns a JSON containing 10, meaning 2+1+3+4.
- `/crash` which terminates the service execution after responding to the client with info about the host and the port of the service.
- `/last` which returns a string representing the last operation requested with success in a fixed format.

What are the problems?

- Requirements: My words were ambiguous, and this impacts your prompting.
- ‘Does it work?’: How can you understand if the code produced is correct in terms of expected behaviour, bugs, vulnerabilities, etc.?
- Can AI code everything?
- Is the code extendible easily?
- Sometimes it ‘get stuck’ and goes in a loop of updates (context poisoning).
- Environmental and economic impacts.

Requirements

- Customers, colleagues, and people in general do not always give a clear view of what the software should be.
- When you are using AI prompting, there are a lot of things to know to obtain better results.
- Prompt engineering is a topic that has received attention lately, but it is unstable given rapid the evolution of AI models.

Does it work?

- First of all, who evaluate it should be expert of the domain of the software.
- Testing is the way to understand if a large, generated code base behave as expected.
- Testing can be automated, but not fully: creating meaningfull tests and interpret the results is currently hard for an AI model.

Can AI code everything?

- Today our task with simple.
- In general, the more complex the domain, the more mistakes it makes.
- As before, you need an expert to understand if it allucinates or inserts small bugs.

Is the code extendible easily?

In general, the AI model looks to satisfy your prompt.

The code generated is tailored for it.

When you add a feature to the software, it happens that it try to rewrite large parts of the code base.

All because it didn’t know that the feature has to be added in a second moment.

Think of large codebases if this is affordable…

Context poisoing

Most of those AI models (LLMs) work on a context window with the previous interaction of the chat.

This bring two problems:

- A space limit of what it ‘remembers’ of what you ask it.
- If there are errors in some output, it will ‘remember’ the errors.

This brings a loop of trial and error with your prompts.

In general, you have to reset and restart the task.

More info tailored with Roo Code: https://docs.roocode.com/advanced-usage/context-poisoning

Environmental impact

Two words: energy consumption.

As of September 2025, AI data centers are on track to account for nearly half of all data center power consumption, projected to reach an astounding 23 gigawatts (GW). A figure equivalent to twice the total energy consumption of the Netherlands

More info at: https://www.technologyreview.com/2025/05/20/1116327/ai-energy-usage-climate-footprint-big-tech/

Economical impact

Today, we exploited GitHub PRO to use Roo Code otherwise, we had to pay for the tokens to AI service providers.

And we used an old GPT model.

The better code will be written by those who can afford better models.

What will happen in the future?

Hard to say, the tecnology is running wild.

What we saw today in few months will be different.

Currently, the demand for junior developers is decreasing.

They will not be fully substituted, but one person will do the work of 5/10 people.

What will happen in the future?

For those reasons, probably (advanced) software engineering will grow in importance.

- Requirements elicitation and specification.
- Understand software architectures and APIs.
- Structured testing.
- Security analysis.

Lab take away

- ❑ What is a web service.
- ❑ How to exploit AI to generate simple code.
- ❑ What are the issues in using AI for coding.

Project take away

- ❑ You will be allowed to use AI for the project.
- ❑ You will have to document in the report the use, issued and results of using it.
- ❑ You will not be allowed to use AI in lab tests.