**Step-by-step guide (with prompts you can paste)**

**1) Make a project folder**

Create a folder anywhere, e.g. **copilot\_flask\_scaffold**, and open it in VS Code.

**2) Create a virtual environment & install Flask**

Open VS Code Terminal:

python -m venv .venv

# Windows: .

.venv\Scripts\activate

# macOS/Linux:

source .venv/bin/activate

pip install Flask

pip freeze > requirements.txt

Create a file named **requirements.txt** (if pip freeze didn’t create it) and put:

Flask>=3.0

**3) Create app.py and let Copilot write the server**

Create app.py. Paste this prompt at the top, then **pause** for Copilot’s ghost text and **Tab** to accept:

**PROMPT (in app.py):**

# Basic Flask server

# - Create a Flask app

# - GET "/" should render templates/index.html

# - GET "/api/hello" should return JSON: {"message": "Hello from Flask!"}

# - Run with debug=True when executed directly

If nothing appears, type the first import to “prime” Copilot:

from flask import Flask, jsonify, render\_template

Pause again to let Copilot complete the app.

**4) Frontend files (index.html, app.js, style.css)**

Create folders **templates** and **static**.

**A) templates/index.html**

Paste this at the top, **pause**, then accept Copilot’s HTML:

**PROMPT:**

<!-- Minimal homepage

- Title: "Flask + Copilot"

- H1 heading

- Button with id="btnHello" labeled "Call API"

- <pre id="output"></pre> to show results

- link to /static/style.css and script /static/app.js

-->

Make sure it includes:

<link rel="stylesheet" href="/static/style.css">

<script src="/static/app.js"></script>

**B) static/app.js**

Paste this, **pause**, accept:

**PROMPT:**

// When the page loads, add a click handler to #btnHello.

// On click: fetch("/api/hello"), parse JSON, pretty-print it into #output.

// Handle errors by showing the message in #output.

**C) static/style.css**

Paste this, **pause**, accept:

**PROMPT:**

/\* Simple, centered layout with readable font, max-width container, spacing,

and a basic button style. \*/

**4) Run your app**

python app.py

Open [**http://localhost:5000**](http://localhost:5000) → Click **“Call API”** → you should see JSON from /api/hello.

**5) Practice: add another API and wire it up**

In app.py, **under** your hello route:

**PROMPT:**

# Add a GET "/api/time" route that returns {"time": "<ISO timestamp>"}.

In templates/index.html:

**PROMPT:**

<!-- Add another button with id="btnTime" labeled "Current Time" next to Call API. -->

In static/app.js:

**PROMPT:**

// Add a click handler for #btnTime to fetch("/api/time") and show the JSON in #output.

Run again and test the new button.

**Copilot tips (when in doubt)**

* **Trigger suggestion:** Cmd/Ctrl+Shift+P → “GitHub Copilot: Generate Inline Suggestion”
* **Cycle options:** “Show Next/Previous Inline Suggestion”
* **Accept partially:** “Accept Next Word/Line”
* **Inline Chat:** Select code → “Copilot: Inline Chat” → ask: *“make this async and add error handling”*