Only a few of the 38 new files contain the Flask web interface. The README describes how to run it:

39 ## Flask App

40 A simple Flask web app `app.py` allows interactive generation of questions using `gpt-4-0314`...

42 ```bash

43 python app.py

44 ```

45 Open `http://localhost:5000` in a browser ...

46 The app displays each question with its answer and evaluation metrics...

47 The app also attempts to compile or parse the submitted code to detect syntax errors...

requirements.txt lists flask as a dependency:

1 openai

2 replicate

3 anthropic

4 matplotlib

5 flask

The main Flask logic is in app.py:

1 """Simple Flask app to generate questions from code using GPT-4."""

...

8 from flask import Flask, request, render\_template

...

13 app = Flask(\_\_name\_\_)

14 llm = LLM("gpt-4-0314")

...

58 @app.route("/", methods=["GET", "POST"])

59 def index():

...

87 if \_\_name\_\_ == "\_\_main\_\_":

88 app.run(debug=True)

The app uses two HTML templates stored under templates/:

* templates/index.html – form for uploading or pasting code
* templates/results.html – displays question–answer pairs and their scores

Example snippet from templates/index.html:

1 <!doctype html>

...

8 <form method="post" enctype="multipart/form-data">

9 <label>Programming Language:

10 <select name="language">

...

24 <button type="submit">Generate</button>

Example snippet from templates/results.html:

7 <h1>Generated Questions and Answers</h1>

...

12 {% if code\_error %}

13 <p style="color:orange">Code analysis: {{ code\_error }}</p>

...

18 <em>A{{ loop.index }}: {{ qa.answer }}</em>

...

These pieces form the Flask app’s file structure:

app.py # Flask application entry point

templates/

index.html # upload/paste form

results.html # displays questions and scores

src/

llm.py # LLM wrapper used by the app

evaluator.py # evaluates generated questions

prompt.py # helper for building prompts

All these files live in the repository root (/workspace/LLMs). The src/ modules provide helper functions but are not Flask‑specific. The core Flask components are app.py and the templates directory.

- - -

Can you provide an imaginary view/visuals/images/UI of what would the Flask App look like?

Below is a conceptual “mock screenshot” showing how the app’s main form is structured based on the index.html template. The interface is intentionally simple:

+-------------------------------------------------+

| Code Q&A Generator |

+-------------------------------------------------+

| Programming Language: [ Python ▼ ] |

| Number of questions: [ 5 ] |

| |

| Paste code: |

| [-------------------------------------------] |

| | | |

| | <textarea rows="10" cols="80"> | |

| | | |

| [-------------------------------------------] |

| Or upload file: [ Choose File ] |

| |

| [ Generate ] |

+-------------------------------------------------+

*Relevant lines:* The form elements come directly from templates/index.html lines 8–24, which include the language dropdown, count input, textarea for pasting code, file upload option, and submit button

.

When the user submits code, the results page lists the generated questions and evaluation metrics:

+-------------------------------------------------+

| Generated Questions and Answers |

| ← Back |

| |

| Q1: <question text> |

| A1: <answer text> |

| Relevance: 0.89 |

| Clarity and Coherence: 0.92 |

| Conciseness: 0.87 |

| Coverage: 0.84 |

| |

| [Additional question–answer pairs...] |

+-------------------------------------------------+

*Relevant lines:* This layout corresponds to templates/results.html lines 7–27, where each question/answer pair is displayed with its associated scores

.

The README explains how to run this app at http://localhost:5000, noting that it shows the metrics (Relevance, Clarity and Coherence, Conciseness, Coverage) and also reports any syntax problems detected in the uploaded code

.

This should give you a clear picture of the minimal UI for the Flask application.