Flask Joke APP

M.Junaid 109(4B)

Table of Contents

1. Introduction

- This Python code sets up a basic web application using the Flask framework. Its primary functionality is to display a simple home page and provide random jokes through the /joke endpoint. The application interacts with an external API (JokeAPI) to fetch jokes in real time and renders them as HTML content.

2. Libraries Used

- Flask: A micro web framework used for handling HTTP requests and defining routes in the web application.
- Requests: A Python library used to send HTTP requests to the external joke API and retrieve joke data.

3. Functions and Workflow

- Home Route (/)
    - Displays a welcome message.
    - Provides a hyperlink to the /joke route for accessing a random joke.

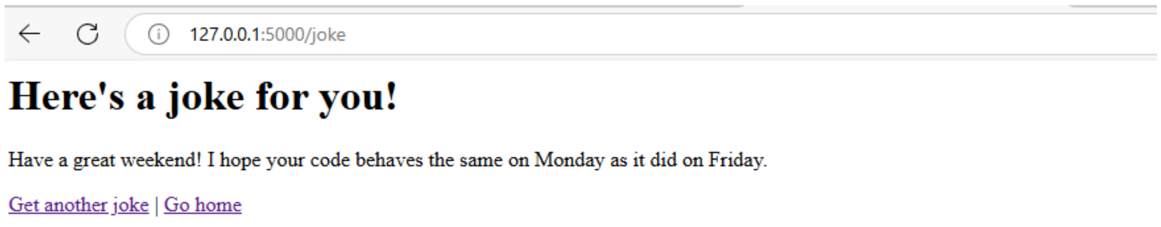
- Joke Route (/joke)

  - Sends a GET request to the JokeAPI using the URL: https://v2.jokeapi.dev/joke/Any?type=single
  - Checks if the request is successful and the response does not contain an error.
  - If successful:
  - Extracts the joke from the JSON response.
  - Displays the joke on a webpage along with options to get another joke or go back to the home page.
  - If unsuccessful:
  - Displays an error message: “Couldn't fetch a joke. Try again later!”
  - If an exception occurs (e.g., network failure), it returns: “Joke service unavailable. Try again later!”

4. Code Execution and Output

- When run, the app listens on localhost:5000.
- Visiting / displays the home page with a link to /joke.
- Visiting /joke shows a random joke retrieved from JokeAPI.
- HTML responses are dynamically generated using Python f-strings with embedded joke content.

5. Screenshot of Output



Here's a joke for you!

Have a great weekend! I hope your code behaves the same on Monday as it did on Friday.

Get another joke | Go home