# **FastAPI Joke Generator Documentation**

## **Problem Description**

## and display random jokes from the backend API. The goal is to provide an interactive and user-friendly interface for joke retrieval.

## **Allowed Operations**

The frontend supports the following operations:

* Click a button to fetch a joke from the backend.
* Display the fetched joke dynamically on the webpage.
* Show a loading message while fetching the joke.

## **Approach**

To implement the frontend, we use HTML, CSS, and JavaScript:

1. **HTML:** Provides the structure, including a button and a display area for the joke.
2. **CSS:** Styles the page for a clean and responsive look.
3. **JavaScript:** Handles API requests and dynamically updates the content.

The frontend interacts with the FastAPI backend to fetch and display jokes efficiently.

## **Explanation**

### **How It Works**

1. User clicks the **Get a Joke** button.
2. JavaScript sends a request to http://127.0.0.1:8000/joke.
3. The response is processed and displayed on the webpage.
4. A loading message appears while the joke is being fetched.

### **Key Components**

* **HTML:** Defines the page structure, including a title, button, and joke display area.
* **CSS:** Enhances the UI with styling for layout and button design.
* **JavaScript:** Fetches jokes from the API and updates the page dynamically.

### **Running the Frontend**

1. Start the FastAPI server.
2. Open index.html in a browser.
3. Click the **Get a Joke** button to retrieve and display a joke.

## **Expected Output**

* When the **Get a Joke** button is clicked, a joke is fetched and displayed.
* If the joke API returns a single-line joke, it is shown directly.
* If the joke API returns a two-part joke, both parts are displayed.
* If an error occurs, a failure message is displayed.