# Useless Facts Generator Project

### 1. Project Overview

The Useless Facts Generator is a web-based application that provides users with random, amusing facts. The application includes a "Fact of the Week" feature, displayed by default, as well as a button that allows users to retrieve new random facts on demand. This simple, interactive tool is built using Flask for the backend, a public facts API, and a responsive HTML/CSS/JavaScript frontend.

### 2. Project Structure

```

project/

├── app.py # Flask application with all routes

├── templates/

│ └── index.html # Main HTML page for display

└── static/

├── css/

│ └── style.css # Styling for the application

└── js/

└── script.js # JavaScript for dynamic fact generation

```

### 3. Features

- \*\*Fact of the Week\*\*: A random fact fetched upon server startup and displayed on the main page for the entire week.

- \*\*Random Fact Button\*\*: Allows users to retrieve new random facts via an API call without reloading the page.

- \*\*Error Handling\*\*: Provides user-friendly error messages if the fact retrieval fails.

### 4. Technology Stack

- \*\*Backend\*\*: Python (Flask), with API requests handled by `requests` library.

- \*\*Frontend\*\*: HTML, CSS, JavaScript

- \*\*External API\*\*: `https://uselessfacts.jsph.pl/random.json`

### 5. Installation and Setup

#### Step 1: Clone the Repository

```bash

git clone <repository-url>

cd <project-directory>

```

#### Step 2: Set Up a Virtual Environment (optional)

```bash

python -m venv venv

source venv/bin/activate # MacOS/Linux

venv\Scripts\activate # Windows

```

#### Step 3: Install Dependencies

```bash

pip install Flask requests

```

#### Step 4: Run the Application

```bash

python app.py

```

The application will be accessible at `http://127.0.0.1:5000`.

### 6. Code Overview

#### `app.py`

- Initializes Flask and defines routes for:

- \*\*`/index`\*\*: Displays the main page with "Fact of the Week" and "Generate Fact" button.

- \*\*`/generated-fact`\*\*: Returns a new random fact as JSON for asynchronous loading.

```python

from flask import Flask, render\_template, jsonify

import requests

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

# Function to fetch a random fact

def get\_fact\_of\_the\_week():

url = "https://uselessfacts.jsph.pl/random.json?language=en"

try:

response = requests.get(url)

return response.json().get("text", "No fact available.")

except requests.exceptions.RequestException:

return "Error retrieving the Fact of the Week."

fact\_of\_the\_week = get\_fact\_of\_the\_week()

@app.route("/index")

def home\_display():

return render\_template("index.html", fact\_of\_the\_week=fact\_of\_the\_week)

@app.route("/generated-fact")

def random\_fact():

url = "https://uselessfacts.jsph.pl/random.json?language=en"

try:

response = requests.get(url)

if response.status\_code == 200:

return jsonify(fact=response.json().get("text"))

return jsonify(fact="Error retrieving data"), 500

except requests.exceptions.RequestException:

return jsonify(fact="Error retrieving data"), 500

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

app.run(debug=True)

```

#### `index.html`

- Displays the "Fact of the Week" and a button to fetch new facts.

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Useless Facts Generator</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/style.css') }}">

</head>

<body>

<header>

<h1>Fact of the Week</h1>

</header>

<main>

<p>{{ fact\_of\_the\_week }}</p>

<button onclick="fetchRandomFact()">Generate Random Fact</button>

<p id="fact"></p>

</main>

<script src="{{ url\_for('static', filename='js/script.js') }}"></script>

</body>

</html>

```

#### `script.js`

- Fetches a new fact via the `/generated-fact` endpoint.

```javascript

async function fetchRandomFact() {

const response = await fetch("/generated-fact");

const data = await response.json();

document.getElementById("fact").innerText = data.fact;

}

```

### 7. Conclusion

The Useless Facts Generator is a simple, interactive application for delivering fun, random facts. The project combines a Flask backend with an HTML/JavaScript frontend, creating an engaging user experience with minimal setup and maintenance.