API Documentation

Used Version : Python 3.11

Overview

This API allows users to input a URL and a related question; the API will then fetch the content from the URL, process it, and return an answer to the question based on the content found on the page.

API Endpoint

- **URL**: /fetch-answer
- **Method**: POST
- **Content-Type**: application/json
- **Authorization**: None (Assuming no authentication is required for simplicity)

Request Format

To use this API, send a JSON object with the following properties:

```
{
  "url": "URL_TO_FETCH_CONTENT_FROM",
  "question": "QUESTION_ABOUT_THE_CONTENT"
}
```

- **url**: A string representing the URL from which the content will be fetched. This should be a valid HTTP or HTTPS URL.
- **question**: A string representing the question that needs to be answered based on the content of the URL.

Response Format

The API will respond with a JSON object containing the answer or an error message:

```
{
  "answer": "ANSWER_TO_THE_QUESTION"
}
```

```
{
  "error": "ERROR_DESCRIPTION"
}
```

Authentication

This version of the API does not use authentication. For production environments, it is recommended to implement at least token-based authentication to secure access to the API endpoints.

Input Processing

- The URL provided in the request is validated to be a well-formed URL.
- The content is fetched from the URL using the Python `requests` library.
- The HTML content is parsed using `BeautifulSoup` to extract textual content.
- The text is then chunked into manageable pieces for processing.

Response Generation

- The extracted text is processed using the LangChain library with OpenAI embeddings to generate an answer to the question.
- The response is generated by isolating the most relevant part of the processed answer, specifically filtering out any additional queries or extraneous information included in the output from the LangChain model.

Deployment Instructions

Local Deployment

1. **Prerequisites**: Ensure Python and pip are installed on your system.
2. **Install dependencies**:

   ```
   pip install Flask requests chromadb beautifulsoup4 python-dotenv langchain openai
   ```

3. **Set environment variables**:
   - Create a `.env` file in the root directory of your project and add the following line:

     ```
     OPENAI_API_KEY=your_openai_api_key_here
     ```

4. **Run the application**:

   ```
   python app.py
   ```