**Documentation of Chatbot for Byggjobs**

**Overview**

This documentation provides a detailed overview of the Chatbot application built using Flask, LangChain, and Google APIs. The chatbot is capable of answering questions based on content from PDF documents and recommending relevant YouTube videos.

**Table of Contents**

* Prerequisites
* Setup and Installation
* Code Structure
* Endpoints
* Classes and Methods
* Running the Application

**Prerequisites**

Before running the application, ensure you have the following:

* Python 3.8 or higher
* Flask
* LangChain
* Google API Client
* OpenAI API Key
* YouTube API Key
* Knowledge base documents to be loaded and processed

**Setup and Installation**

1. Install the required dependencies using the requirements file
2. Place PDF Documents - Ensure your PDF documents are located in the same directory as the script or provide the correct path.

**Endpoints**

**GET /**

Serves the main chat interface.

**POST /ask**

Endpoint to handle the user's questions and return answers along with recommended YouTube videos.

* Request JSON:

{

"question": "Your question here"

}

* Response JSON:

{

"answer": "Answer from the chatbot",

"videos": [

{

"title": "Video Title",

"description": "Video Description",

"thumbnail": "Thumbnail URL",

"video\_id": "Video ID",

"channel\_title": "Channel Title"

},

...

]

}

**Classes and Methods**

**“Chatbot”**

‘**\_\_init\_\_(self, openai\_api\_key, youtube\_api\_key, pdf\_paths, persist\_directory=".")**’

Initializes the Chatbot instance with the provided API keys and PDF paths. It sets up the OpenAI and YouTube API keys, loads and splits PDF documents, initializes embeddings, creates a vector database, and sets up conversational retrieval chain.

‘**fetch\_videos\_from\_youtube(self, query)**’

Fetches videos from YouTube based on the provided query.

**Parameters:**

* query (str): The search query for YouTube.

**Returns:**

* videos (list): A list of dictionaries containing video details.

ask\_question(self, query)

Answers the user's question using the conversational retrieval chain and fetches related YouTube videos.

**Parameters:**

* query (str): The user's question.

**Returns:**

* answer (str): The answer from the PDF QA model.
* videos (list): A list of recommended YouTube videos.

**Running the Application**

1. **Set API Keys:**

Replace placeholders in the code with your actual OpenAI and YouTube API keys.

1. **Run the Application:**

Once all the code is in structure launch the flask application by calling the code file – ‘python app.py’ and access the link visible In the terminal.

1. **Access the Application:**

Open a web browser and navigate to <http://192.168.0.186:5000/> to access the chat interface.

**Conclusion**

This documentation provides a comprehensive guide to setting up and running the Chatbot application. By following the steps outlined, you should be able to deploy and interact with the chatbot, leveraging the power of PDF, document or knowledge retrieval and YouTube video recommendations.

**Usage example: A screenshot of a web page

Description automatically generated**