

CHATBOT DOCUMENTATION

INTRODUCTION

THIS DOCUMENTATION PROVIDES AN OVERVIEW OF A CHATBOT DESIGNED TO INTERACT WITH PDF FILES AND ANSWER USER QUESTIONS. IT'S BUILT USING PYTHON, STREAMLIT, GOOGLE GENERATIVE AI, AND OTHER LIBRARIES.

LIBRARIES USED

- **STREAMLIT:** A PYTHON FRAMEWORK FOR CREATING WEB APPLICATIONS.
- **PYPDF2:** A LIBRARY FOR WORKING WITH PDF FILES.
- **LANGCHAIN:** A LIBRARY FOR NATURAL LANGUAGE PROCESSING TASKS.
- **GOOGLE GENERATIVE AI:** A LIBRARY FOR ACCESSING GOOGLE'S GENERATIVE AI MODELS.
- **PYTHON-DOTENV:** A LIBRARY FOR MANAGING ENVIRONMENT VARIABLES.
- **CHROMADB:** A DATABASE FOR MANAGING CHROME EXTENSIONS.
- **FAISS-CPU:** A LIBRARY FOR EFFICIENT SIMILARITY SEARCH AND CLUSTERING.

CODE OVERVIEW

IMPORTING LIBRARIES

```
import streamlit as st
```

```
from PyPDF2 import PdfReader
```

```
from langchain.text_splitter import RecursiveCharacterTextSplitter
```

```
import os
```

```
from langchain_google_genai import GoogleGenerativeAIEmbeddings
```

```
from langchain.vectorstores import FAISS
```

```
from langchain_google_genai import ChatGoogleGenerativeAI
```

```
from langchain.chains.question_answering import load_qa_chain
```

FROM LANGCHAIN.PROMPTS IMPORT PROMPTTEMPLATE

FROM DOTENV IMPORT LOAD_DOTENV

THE CHATBOT UTILIZES VARIOUS PYTHON LIBRARIES FOR ITS FUNCTIONALITIES.

LOADING ENVIRONMENT VARIABLES

ENVIRONMENT VARIABLES ARE USED FOR CONFIGURATION, SUCH AS THE **GOOGLE API** KEY.

FUNCTIONS

SEVERAL FUNCTIONS ARE DEFINED TO HANDLE TASKS LIKE RETRIEVING TEXT FROM **PDF** FILES AND GENERATING RESPONSES TO USER QUERIES.

STREAMLIT INTERFACE

THE CHATBOT PROVIDES A USER-FRIENDLY INTERFACE POWERED BY **STREAMLIT**, ALLOWING USERS TO UPLOAD **PDF** FILES AND ASK QUESTIONS.

USAGE

1. **UPLOAD PDF** FILES USING THE FILE UPLOADER.
2. TYPE QUESTIONS IN THE TEXT INPUT FIELD.
3. CLICK "**SUBMIT & PROCESS**" TO PROCESS THE UPLOADED **PDF** FILES AND RECEIVE ANSWERS.

SYSTEM REQUIREMENTS

ENSURE THAT THE SYSTEM WHERE THE CHATBOT WILL BE DEPLOYED MEETS THE FOLLOWING REQUIREMENTS:

- **OPERATING SYSTEM:** COMPATIBLE WITH **WINDOWS**, **MACOS**, OR **LINUX**.
- **PYTHON INSTALLATION:** **PYTHON 3.X** INSTALLED WITH THE NECESSARY LIBRARIES.
- **RAM:** **MINIMUM OF 16GB RAM** RECOMMENDED FOR SMOOTH PERFORMANCE.
- **STORAGE:** ADEQUATE DISK SPACE FOR STORING **PYTHON CODE**, **LIBRARIES**, AND **UPLOADED FILES**.

- **WEB BROWSER:** COMPATIBLE WEB BROWSER FOR ACCESSING THE STREAMLIT INTERFACE.
- **INTERNET CONNECTION:** STABLE INTERNET CONNECTION FOR ACCESSING EXTERNAL APIs AND SERVICES.
- **FIREWALL AND SECURITY SETTINGS:** ADJUSTMENTS TO FIREWALL AND SECURITY SETTINGS TO ALLOW NECESSARY TRAFFIC FOR THE CHATBOT APPLICATION.

DEPLOYMENT CONSIDERATIONS

- **CLOUD DEPLOYMENT:** CONSIDER DEPLOYING ON AWS, GOOGLE CLOUD PLATFORM, OR MICROSOFT AZURE FOR SCALABILITY.
- **CONTAINERIZATION:** USE DOCKER FOR PACKAGING AND PORTABILITY.
- **CONTINUOUS INTEGRATION/CONTINUOUS DEPLOYMENT (CI/CD):** IMPLEMENT CI/CD PIPELINES FOR AUTOMATION AND UPDATES.

FUNCTIONS

GET_PDF_TEXT: RETRIEVES TEXT FROM PDF FILES.

GET_TEXT_CHUNKS: SPLITS TEXT INTO SMALLER CHUNKS.

GET_VECTOR_STORE: CREATES A VECTOR STORE FOR TEXT CHUNKS.

GET_CONVERSATIONAL_CHAIN: SETS UP A CONVERSATIONAL CHAIN FOR QUESTION ANSWERING.

USER_INPUT: HANDLES USER INPUT AND GENERATES RESPONSES.

MAIN: MAIN FUNCTION TO RUN THE CHATBOT.

USAGE

1. **UPLOAD PDF FILES** USING THE FILE UPLOADER.
2. **ASK QUESTIONS** IN THE TEXT INPUT FIELD.
3. **CLICK "SUBMIT & PROCESS"** TO PROCESS THE PDF FILES AND GET ANSWERS.

