### **Documentation:**

#### Introduction:

The provided code implements a chatbot application using Streamlit, a Python library for building web applications. The chatbot allows users to upload PDF or DOCX files, process them to extract text, and interact with the chatbot to ask questions related to the uploaded content.

#### Features:

* File Uploading: Users can upload PDF or DOCX files containing relevant information for the chatbot to process.
* Text Processing: The uploaded files are processed to extract text using PyPDF2 for PDF files and python-docx for DOCX files.
* Question Answering: Users can interact with the chatbot by asking questions related to the uploaded content. The chatbot utilizes Google Generative AI for generating responses based on the context and user queries.
* User Interface: The application provides a user-friendly interface using Streamlit, allowing users to upload files, interact with the chatbot, and view responses.

#### Architecture:

The architecture of the application consists of several components:

* Streamlit: The web application framework used to create the user interface and handle user interactions.
* PyPDF2: Library for extracting text from PDF documents.
* python-docx: Library for extracting text from DOCX documents.
* Google Generative AI: Utilized for generating responses to user queries based on the provided context.
* Langchain: A library providing functionality for text processing, including text splitting and question-answering chains.
* FAISS: A library for creating vector stores from text chunks to perform similarity searches.

#### Dependencies:

The application relies on the following Python libraries and dependencies:

* streamlit
* PyPDF2
* python-docx
* langchain
* langchain-google-genai
* google-generativeai
* langchain-community

The versions of these dependencies are specified in the requirements.txt file.

#### Usage:

* Setup: Clone the repository and install the dependencies using pip install -r requirements.txt.
* Run Application: Execute the app.py script using streamlit run app.py.
* Upload Files: Use the file uploader in the sidebar to upload PDF or DOCX files.
* Process Files: Click the "Submit & Process" button to extract text from the uploaded files and create a vector store.
* Interact with Chatbot: Input questions in the chat input box to receive responses from the chatbot.

#### Limitations:

* The chatbot's accuracy depends on the quality and relevance of the uploaded documents.
* Complex queries or questions may not always yield accurate responses.

#### Additional Information:

* For any issues or inquiries, please refer to the contact information provided in the README file.

### **Conclusion:**

The chatbot application provides a convenient way for users to interact with document content and receive relevant information through natural language queries. With its simple interface and powerful text processing capabilities, it serves as a useful tool for various information retrieval tasks.