Design & approach document

# Document search bot

**Objective**

The objective of this document is to outline the design and approach for the development of the Document Search Bot application. This document will provide a detailed overview of the application's functionality, architecture, and implementation plan.

**Scope:**

Users can upload several document formats (PDF, DOCX, TXT, XLSX, and PPTX) and use natural language queries to search through them using the Document Search Bot online application. PyPDF2, docx, and pandas will be used for document processing, LangChain for text processing, OpenAI for language models, and Streamlit for the front end of the application.

**Architecture Overview:**

* **Streamlit**: Used as the main framework for building the web application.
* **PyPDF2:** Library for PDF processing, likely used for extracting text from PDF documents.
* **langchain**: Text processing library, possibly used for various text processing tasks.
* **OpenAIEmbeddings:** Embeddings for text, possibly used for text representation and similarity calculations.
* **FAISS**: Vector store, possibly used for storing and querying vectors efficiently.
* **Question Answering (QA) Chain:** A chain model for question answering, loaded using load\_qa\_chain.
* **OpenAI LLMs**: Language model (LLMs) from OpenAI, possibly used for generating responses to user queries.
* **Pandas**: Library for data manipulation, possibly used for managing data within the application.
* **BytesIO**: Used for handling byte data, likely used for file manipulation.

**PPTX**: Library for working with PowerPoint files, possibly used for extracting text from PowerPoint documents.

* **OS**: Module for interacting with the operating system, likely used for file operations.
* **docx**: Library for working with Word documents, possibly used for extracting text from Word documents.

### **Document Approach:**

**User Types:**

1. **Admin**: In addition to being in the position of uploading and removing documents, they can post questions and get responses based on the documents they have uploaded.
2. **User**: Can ask queries and receive answers based on the uploaded documents.

**Document Processing:**

* Use PyPDF2 for processing PDF files to extract text.
* Use the docx library for processing Word documents (.docx) to extract text.
* Use pandas for processing Excel files (.xlsx) to extract text from each cell in the sheets.
* Use LangChain for text processing, including splitting text into chunks and generating embeddings.
* Use OpenAI for language models, such as question answering.

**Document Upload:**

* Users can upload documents of various formats (PDF, Word, Excel, PowerPoint, Text).
* There is a maximum file size limit enforced to ensure efficient processing (2MB).
* Uploaded documents are stored in a designated folder (UPLOAD\_FOLDER).

**Question Answering:**

* Use LangChain's question answering module for answering user queries based on the uploaded documents.
* Utilize the OpenAI library for language models to enhance the quality of responses.

**User Interface:**

* Use Streamlit to create a user-friendly interface.
* Include functionalities like uploading and deleting documents, asking queries, and displaying responses.

**Future Enhancements:**

1. Implement user authentication for distinguishing Admin and Normal Users.
2. Enhance search capabilities with advanced NLP techniques like semantic search.
3. Support for more file formats.
4. Improve error handling and edge case scenarios.
5. Implement caching mechanisms for improved performance, especially with larger documents.

### **Conclusion:**

With the help of the robust Document Search Bot, users may upload and use natural language queries to search across a wide range of documents. It makes use of Streamlit for the front end, offering a document management interface that is easy to use. LangChain handles text processing and embeddings, whereas PyPDF2, docx, and pandas manage text extraction from various document formats in the backend processing. Language models from OpenAI are utilized by the bot to deliver accurate responses to user inquiries. All things considered, the Document Search Bot improves user experience and simplifies the document search process.