**Project Name: PDF Chatbot Analyzer**

**Overview**

We’re building a web application where users can upload a PDF containing medical symptoms, and a chatbot will respond to queries about the symptoms by referencing the content of the PDF. This application involves a frontend for user interaction and a backend for processing the PDF and generating chatbot responses.

**Technologies Used**

* **Frontend**: HTML, CSS, JavaScript
* **Backend**: Python.
* **PDF Processing**: PyMuPDF.
* **Chatbot Logic**: Natural Language Processing (NLP) using Python.

(e.g. spaCy).

Purpose: Generate responses based on user queries and PDF content

* **Fitz module (PyMuPDF)**: Fitz is used for extracting all the text content in the pdf document.
* **Spacy module**: The extracted text is tokenized using spacy module, and then cleaned by:

1. Removing any punctuation marks.

2. Converting all the tokens into lower case,

3. Cleans all the empty spaces from the text,

4. Then lemmatises the whole text,

5. The cleaning of the text is done.

* **Google Generative AI**: Once the text data is cleaned, we feed it to Gemini, to read the whole text and answer from the text, we clearly add a constraint asking it to only answer from the text it is getting and not any irrelevant question.
* **Flask & flask\_cors**: We use flask module, for integrating this python functionality for the web application, we input the pdf document, using JS, since flask does not properly integrate with JS, we use flask\_cors for getting file from the user, and then pass it to the python file in the backend, where the text is extracted, cleaned and fed into Gemini.

**Conclusion**

This project integrates multiple technologies to create a functional and interactive web application. The frontend provides a user-friendly interface, while the backend handles the core logic of processing PDF content and generating chatbot responses based on the symptoms described in the PDF.