

LABORATORY REPORT

**Application Development Lab
(CS33002)**

B.Tech Program in ECSc

Submitted By

Name: Shreyaa Venkateswaran

Roll No: 2230120



**Kalinga Institute of Industrial Technology
(Deemed to be University)
Bhubaneswar, India**

Spring 2024-2025

Table of Content

Exp No.	Title	Date of Experiment	Date of Submission	Remarks
1.	Experiment 1: Build a resume using HTML/CSS	07-01-2025	14-01-2025	
2.	Experiment 2: Machine Learning for Cat and Dog Classification	15-01-2025	20-01-2025	
3.	Experiment 3: Regression Analysis for Stock Prediction	21-01-2025	27-01-2025	
4.	Experiment 4: Conversational Chatbot with Any Files	04-02-2025	09-02-2025	
5.				
6.				
7.				
8.				
9.	Open Ended 1			
10.	Open Ended 2			

Experiment Number	4
Experiment Title	Conversational Chatbot with Any Files
Date of Experiment	04-02-2025
Date of Submission	09-02-2025

1. Objective:

To build a chatbot capable of answering queries from an uploaded PDF/Word/Excel document.

2. Procedure:

1. Integrate open-source LLMs such as LLama or Gemma from Ollama
2. Develop a Flask backend to process the PDF/word/excel content.
3. Implement Natural Language Processing (NLP) to allow queries. You can use LLamaIndex or Langchain
4. Create a frontend to upload document files and interact with the chatbot, just like OpenAI interface
5. Provide an option to choose the LLM model from a dropdown list.
6. Display the chatbot responses on the webpage.

3. Code:

```
import os

import streamlit as st

from langchain_groq import ChatGroq

from langchain.text_splitter import RecursiveCharacterTextSplitter

from langchain.chains.combine_documents import create_stuff_documents_chain
```

```

from langchain_core.prompts import ChatPromptTemplate

from langchain.chains import create_retrieval_chain

from langchain_community.vectorstores import FAISS

from langchain_community.document_loaders import PyPDFDirectoryLoader

from langchain_google_genai import GoogleGenerativeAIEmbeddings

from dotenv import load_dotenv

load_dotenv()

## load the GROQ and Google API key from the .env file

groq_api_key = os.getenv("GROQ_API_KEY")

os.environ['GOOGLE_API_KEY']=os.getenv("GOOGLE_API_KEY")

st.title("PDF Reader ChatBot")

llm=ChatGroq(groq_api_key=groq_api_key,model_name="gemma
2-9b-it")

prompt=ChatPromptTemplate.from_template(
"""
Answer the questions based on the provided context only.

Please provide the most accurate response based on the question.

<context>

{context}

<context>

Questions: {input}
"""
)

```

```

def vector_embedding():
    if "vectors" not in st.session_state:

        st.session_state.embeddings=GoogleGenerativeAIEmbeddings(model="models/embedding-001")

        st.session_state.loader=PyPDFDirectoryLoader("./cholas")

        st.session_state.docs=st.session_state.loader.load()

        st.session_state.text_splitter=RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)

        st.session_state.final_documents=st.session_state.text_splitter.split_documents(st.session_state.docs)

        st.session_state.vectors=FAISS.from_documents(st.session_state.final_documents, st.session_state.embeddings)

        prompt1=st.text_input("Enter your Question")

        if st.button("Creating Vector Store"):

            vector_embedding()

            st.write("Vector Store DB is Ready")

            import time

            if prompt1:

                document_chain=create_stuff_documents_chain(llm,prompt)

                retriever=st.session_state.vectors.as_retriever()

                retrieval_chain=create_retrieval_chain(retriever, document_chain)

                start=time.process_time()

                response=retrieval_chain.invoke({'input':prompt1})

                st.write(response['answer'])

            #With a streamlit expander

            with st.expander("Document Similarity Search"):

```

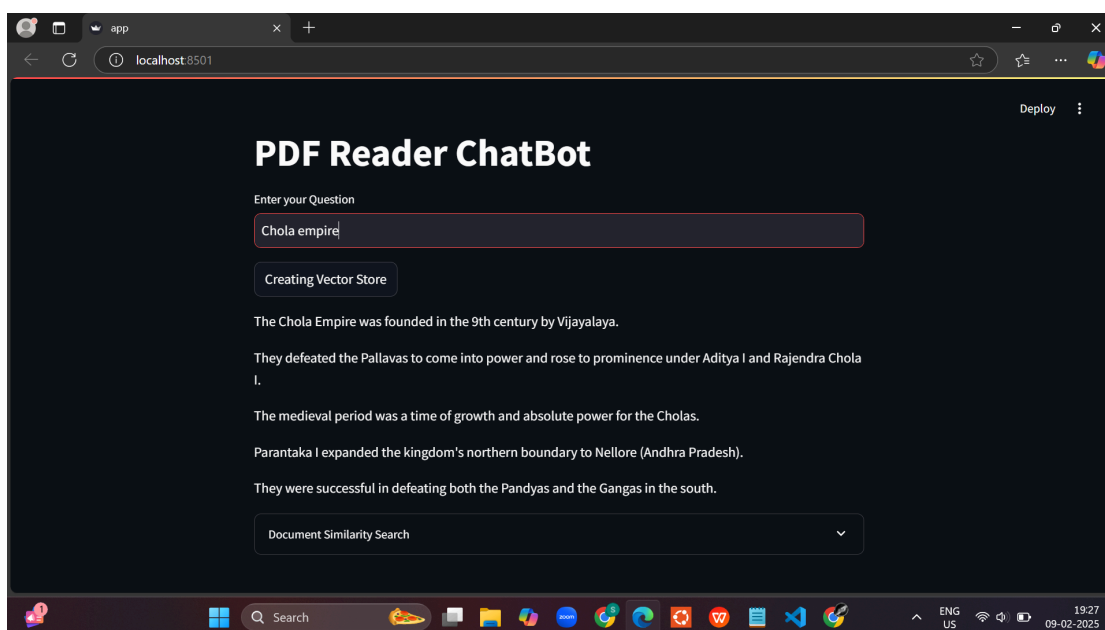
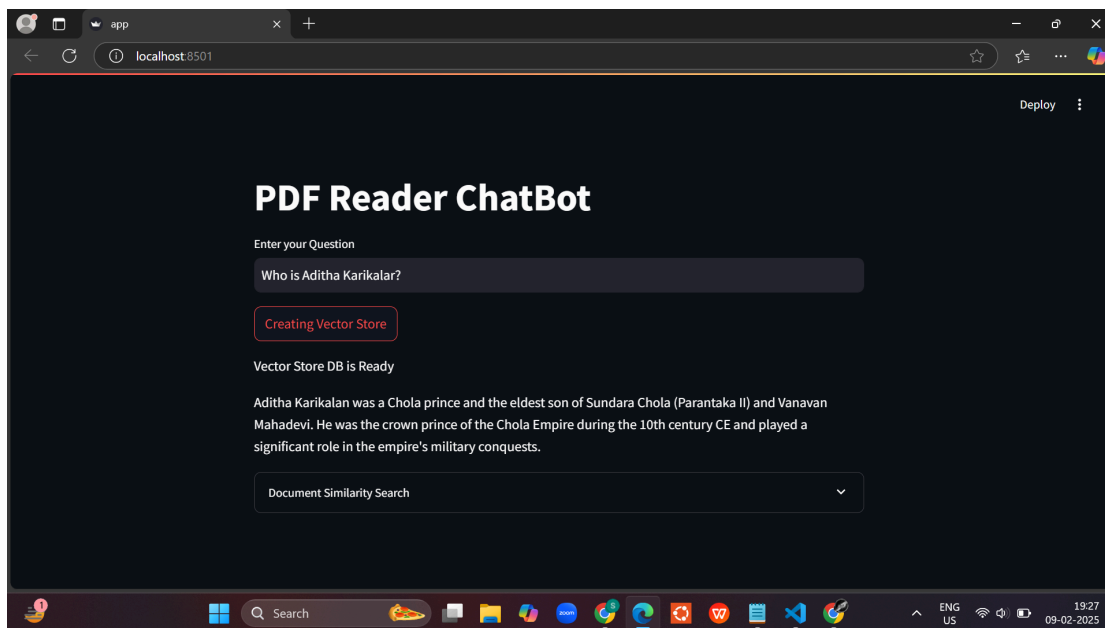
```
#Find relevant chunks
```

```
for i, doc in enumerate(response["context"]):
```

```
st.write(doc.page_content)
```

```
st.write("-----")
```

4. Results/Output:



5. Remarks:

Built a PDF reader chatbot that processes a PDF and answers any question from the user based on its content. The PDF reader chatbot uses the gemma2-9b-it model using GROQ API.
The vector indexing used is Langchain.

Website link: [PDF Reader ChatBOt](#)

GitHub link: [Github](#)

Shreyaa Venkateswaran

Signature of the Lab Coordinator
