CHATBOT USING NATURAL LANGUAGE PROCESSING (NLP)

PROJECT REPORT

Submitted by

RENU K S	**(20BCS015)**
UMESH KRISHNAN G	**(20BCS095)**
MOHANA PRIYAN A S	**(21BCS310)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Dr. MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY POLLACHI-642003

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2024

Dr.MAHALINGAM COLLEGE OF ENGINEERING AND TECHNOLOGY, POLLACHI -642003

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report, "CHATBOT USING NATURAL LANGUAGE PROCESSING" is the bonafide work of

RENU K S	(20BCS015)
UMESH KRISHNAN G	(20BCS095)
MOHANA PRIYAN A S	(21BCS310)

who carried out the project work under my supervision.

Ms. K.Radha
SUPERVISOR
Assistant Professor
Computer Science and Engineering
Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003

Dr. G.Anupriya
HEAD OF THE DEPARTMENT
Computer Science and Engineering
Dr. Mahalingam College of Engineering and Technology, Pollachi – 642003

Submitted for the Autonomous End Semester Examination Project Viva-voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

CHATBOT USING NATURAL LANGUAGE PROCESSING (NLP)

ABSTRACT

In an increasingly digitized world, the demand for intelligent conversational agents capable of understanding and responding to user queries has surged. This project addresses this demand by developing a sophisticated chatbot empowered with Natural Language Processing (NLP) and PDF document processing capabilities. Leveraging cutting-edge technologies such as NLP models and PDF parsing algorithms, the chatbot aims to provide users with a seamless and intuitive platform for engaging in text-based conversations and accessing information from PDF documents. The integration of advanced NLP techniques enables the chatbot to understand user queries, predict intents, and generate appropriate responses, while PDF document processing capabilities allow users to upload and inquire about specific information contained within PDF files.

Key features of the chatbot include the ability to extract text content from PDF documents, generate embeddings for semantic representation, and construct a FAISS index for efficient similarity search. The project's scope encompasses the development and implementation of these functionalities, aiming to create a versatile and user-friendly chatbot that enhances information retrieval and interaction experiences for users.

The project's significance lies in its contribution to the advancement of conversational AI applications and information retrieval systems. By harnessing the power of NLP and AI technologies, the chatbot empowers users with the tools they need to navigate the vast landscape of digital content effectively. Through systematic methodology, rigorous testing, and continuous monitoring, the project endeavours to create a chatbot that not only understands user queries but also retrieves accurate and relevant information from diverse sources, thereby enhancing user productivity and satisfaction in the digital age.

ACKNOWLEDGEMENT

First and foremost, we wish to express our deep unfathomable feeling, gratitude to our institution and our department for providing us a chance to fulfill our long-cherished dreams of becoming Computer Science Engineers.

We express our sincere thanks to our honorable Secretary **Dr. C. Ramaswamy** for providing us with required amenities.

We wish to express our hearty thanks to **Dr. P. Govindasamy,** Principal of our college, for his constant motivation and continual encouragement regarding our project work.

We are grateful to **Dr. G. Anupriya,** Head of the Department, Computer Science and Engineering, for her direction delivered at all times required. We also thank her for her tireless and meticulous efforts in bringing this project to its logical conclusion.

Our hearty thanks to our guide **Ms.K.Radha** for her constant support and guidance offered to us during the course of our project by being one among us and all the noble hearts that gave us immense encouragement towards the completion of our project.

We also thank our review panel members for their continuous support and guidance.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
HTML	Hypertext Markup Language
PDF	Portable Document Format
API	Application Programming Interface
JSON	JavaScript Object Notation
NLTK	Natural Language Toolkit
OS	Operating System
CPU	Central Processing Unit
LED	Light-Emitting Diode
UI	User Interface
AI	Artificial Intelligence

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

CHAPTER 1
INTRODUCTION

In the digital age, the demand for intelligent conversational agents capable of understanding and responding to user queries in natural language has surged. This project aims to address this demand by developing a sophisticated chatbot empowered with Natural Language Processing (NLP) and PDF document processing capabilities. With the proliferation of textual data and the prevalence of PDF documents containing valuable information, there is a growing need for chatbots capable of efficiently retrieving and presenting relevant information to users. By integrating cutting-edge technologies such as NLP models, PDF document parsing algorithms, and advanced AI systems, this project endeavors to create a versatile and user-friendly chatbot that enhances information retrieval and interaction experiences.

The chatbot's primary objective is to provide users with a seamless and intuitive platform for engaging in natural language conversations and accessing information from PDF documents. Leveraging NLP techniques, the chatbot can understand user queries, predict intents, and generate appropriate responses, facilitating meaningful interactions. Additionally, the integration of PDF document processing capabilities enables users to upload and inquire about specific information contained within PDF files, expanding the chatbot's utility beyond traditional text-based interactions. This project seeks to harness the power of AI and machine learning to create a chatbot that not only understands user queries but also retrieves accurate and relevant information from diverse sources, thereby enhancing user productivity and satisfaction.

As technological advancements continue to reshape the way we interact with information, the development of intelligent chatbots represents a significant milestone in the evolution of conversational AI. By combining state-of-the-art NLP models with PDF document processing techniques, this project aims to push the boundaries of what is possible in terms of information retrieval and user interaction. Through the creation of a versatile and adaptive chatbot, this project endeavors to empower users with the tools they need to navigate the vast landscape of digital content effectively.

1.1 Problem Statement

The objective is to develop an advanced chatbot utilizing Natural Language Processing (NLP) for versatile functionality, encompassing the handling of text-based conversations and processing PDF documents to effectively address user inquiries. The chatbot will seamlessly interact with users, understanding queries, and providing relevant responses drawn from both real-time conversations and indexed information within PDF files. Key features include the chatbot's ability to engage in natural language dialogue, parse and extract information from uploaded PDF documents, and deliver accurate responses to user questions through sophisticated question-answering mechanisms. Furthermore, contextual understanding will be prioritized, ensuring coherent responses based on ongoing conversations or the content extracted from PDFs, all within an intuitive and user-friendly interface.

The solution emphasizes efficiency, accuracy, and scalability. The chatbot will process PDF documents swiftly and accurately, indexing relevant information for future reference while maintaining robustness in performance and reliability, even when handling a high volume of user interactions. By combining advanced NLP techniques with seamless PDF document processing, the chatbot will offer users a comprehensive and efficient means to obtain information, fostering a user experience characterized by responsiveness, accuracy, and ease of use.

1.2 Objective

The project objective is to achieve several objectives: firstly, to develop a robust NLP model enabling the chatbot to comprehend and respond effectively to user queries in natural language; secondly, to implement efficient PDF document parsing and extraction algorithms, enabling the extraction of relevant information for indexing and retrieval; thirdly, to integrate a question-answering mechanism utilizing real-time conversation data and indexed PDF content for accurate responses; fourthly, to prioritize contextual understanding within conversations, ensuring coherence and relevance in responses; fifthly, to design an intuitive interface for seamless user interaction; sixthly, to optimize performance for efficiency and scalability; seventhly, to conduct thorough testing and validation for accuracy and user satisfaction

1.3 Scope and Structure

The project aims to develop a sophisticated chatbot equipped with Natural Language Processing (NLP) capabilities for text-based conversations and PDF document processing functionalities. The scope encompasses the integration of advanced NLP models and PDF parsing algorithms to enable the chatbot to understand user queries, retrieve information from PDF documents, and generate contextually relevant responses. The structure of the project report will include sections detailing the methodology used for chatbot development, implementation steps for each module, results showcasing the chatbot's performance and capabilities, and a conclusion summarizing the project's findings and potential implications. Additionally, the report will provide insights into the significance of the project in advancing conversational AI applications and enhancing user experiences in information retrieval.

CHAPTER 2
LITERATURE SURVEY

CHAPTER 2
LITERATURE SURVEY

2.1 D. Carlander-Reuterfelt, Á. Carrera, C. A. Iglesias, Ó. Araque, J. F. Sánchez Rada and S. Muñoz, "JAICOB: A Data Science Chatbot," in IEEE Access, vol. 8, pp. 180672-180680, 2020, doi: 10.1109/ACCESS.2020.3024795.

The paper "JAICOB: A Data Science Chatbot" introduces JAICOB, an innovative chatbot designed specifically for data science tasks. Developed by D. Carlander-Reuterfelt et al., JAICOB aims to assist users in navigating the complexities of data science, offering a user-friendly interface for querying and exploring datasets, performing data analyses, and obtaining insights.

Key features of JAICOB include its natural language processing capabilities, which allow users to interact with the chatbot using everyday language rather than requiring specialized commands or programming knowledge. The chatbot leverages advanced machine learning algorithms to understand and respond to user queries accurately.

The authors highlight JAICOB's versatility, noting its ability to handle a wide range of data science tasks, including data preprocessing, exploratory data analysis, and predictive modeling. Additionally, JAICOB is designed to facilitate collaborative data science projects by enabling users to share datasets, analyses, and findings with team members seamlessly.

Overall, "JAICOB: A Data Science Chatbot" represents a significant contribution to the field of data science by providing a powerful and accessible tool for researchers, analysts, and practitioners. Its intuitive interface and comprehensive functionality make it a valuable asset for anyone working with data, streamlining workflows and enhancing productivity.

2.2 **Mondal, M. Dey, D. Das, S. Nagpal and K. Garda, "Chatbot: An automated conversation system for the educational domain," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAINLP), Pattaya, Thailand, 2018, pp. 1-5, doi: 10.1109/iSAI-NLP.2018.8692927.**

The paper "Chatbot: An automated conversation system for the educational domain" by Mondal, Dey, Das, Nagpal, and Garda, presented at the 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing in Pattaya, Thailand, introduces a chatbot system tailored for the educational domain. The authors address the growing need for innovative educational technologies by proposing an automated conversation system capable of interacting with students and aiding in various educational contexts.

The chatbot system outlined in the paper leverages artificial intelligence and natural language processing techniques to engage with users in meaningful conversations. By understanding user queries and providing relevant responses, the chatbot aims to enhance learning experiences, support students in accessing educational resources, and facilitate knowledge acquisition.

Key components of the chatbot architecture include natural language understanding (NLU) modules for parsing user inputs, a knowledge base containing educational content and resources, and dialogue management algorithms for generating coherent responses. The system is designed to adapt to user preferences and learning styles, offering personalized assistance and guidance.

Overall, the paper contributes to the advancement of educational technology by introducing a novel chatbot system tailored to the specific needs of learners. By combining artificial intelligence and natural language processing techniques, the proposed system offers promising opportunities for enhancing student engagement, accessibility, and learning outcomes in the educational domain.

2.3 Miklosik, N. Evans and A. M. A. Qureshi, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review," in IEEE Access, vol.9,pp.106530-106539,2021,doi:10.1109/ACCESS.2021.3100885

The paper titled "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review" by Miklosik, N. Evans, and A.M.A. Qureshi, published in IEEE Access in 2021, provides a comprehensive analysis of the role of chatbots in digital business transformation. The study employs a systematic literature review methodology to gather and synthesize existing research on this topic.

The authors explore various aspects of chatbot utilization in digital business transformation, including their functionalities, benefits, challenges, and implementation strategies. Through an in-depth examination of relevant literature, they identify key trends, emerging technologies, and best practices associated with chatbot integration in business processes.

The findings highlight the versatility of chatbots in enhancing customer engagement, streamlining operations, and facilitating personalized interactions across diverse industry sectors. Moreover, the paper discusses the potential of chatbots to drive innovation, improve efficiency, and foster competitive advantage for organizations undergoing digital transformation initiatives.

However, the study also acknowledges certain limitations and challenges inherent in chatbot deployment, such as concerns regarding data privacy, user trust, and algorithmic biases. The authors emphasize the importance of addressing these issues through ongoing research, regulation, and ethical guidelines.

In conclusion, the systematic literature review conducted in this paper provides valuable insights into the evolving landscape of chatbot technology and its implications for digital business transformation. It serves as a comprehensive resource for researchers, practitioners, and decision-makers seeking to leverage chatbots effectively in driving organizational change and achieving strategic objectives.

2.4 N V. Socatiyanurak et al., "LAW-U: Legal Guidance Through Artificial Intelligence Chatbot for Sexual Violence Victims and Survivors," in IEEE Access, vol. 9, pp. 131440-131461, 2021, doi: 10.1109/ACCESS.2021.3113172

The paper titled "LAW-U: Legal Guidance Through Artificial Intelligence Chatbot for Sexual Violence Victims and Survivors" by V. Socatiyanurak et al., published in IEEE Access in 2021, introduces LAW-U, an innovative AI-based chatbot designed to provide legal guidance and support for victims and survivors of sexual violence. The chatbot aims to address the critical need for accessible and empathetic legal assistance in navigating complex legal processes related to sexual violence cases.

LAW-U leverages artificial intelligence technology to offer personalized legal information, resources, and emotional support tailored to the individual needs of users. It provides a user-friendly interface for victims and survivors to access information on legal rights, reporting procedures, available support services, and self-care resources.

The development of LAW-U involved extensive research into legal frameworks, psychological principles, and user experience design to ensure its effectiveness and sensitivity to the unique needs of sexual violence survivors. The chatbot's capabilities include natural language processing for intuitive communication, real-time assistance, and a secure platform for maintaining user confidentiality.

The paper highlights the significance of LAW-U in addressing the barriers faced by sexual violence survivors in seeking legal assistance, including stigma, fear, and lack of awareness about their rights and options. By harnessing the power of AI technology, LAW-U aims to empower survivors to access timely and accurate legal information, thereby facilitating their journey towards justice and healing.

Overall, the paper underscores the importance of leveraging AI-driven solutions to enhance access to justice and support for vulnerable populations, such as sexual violence survivors. LAW-U represents a promising advancement in utilizing technology for social good and demonstrates the potential for AI chatbots to make a meaningful impact in addressing complex societal challenges.

2.5 A. Ait-Mlouk and L. Jiang, "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data," in IEEE Access, vol. 8, pp. 149220-149230, 2020, doi: 10.1109/ACCESS.2020.3016142.

The paper "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data" by A. Ait-Mlouk and L. Jiang, published in IEEE Access in 2020, presents KBot, a novel chatbot system designed to enhance natural language understanding over linked data using knowledge graphs. The research addresses the challenge of effectively leveraging linked data resources to improve chatbot performance, particularly in understanding complex user queries.

KBot integrates knowledge graphs into its architecture to enrich the chatbot's understanding of user queries by mapping natural language expressions to entities and relationships within the linked data. By harnessing the structured knowledge represented in knowledge graphs, KBot can interpret user queries more accurately and provide more relevant responses.

The paper discusses the technical details of KBot's architecture, including its components for natural language processing, entity recognition, relation extraction, and knowledge graph traversal. It also describes the algorithms and techniques employed to optimize query processing and improve the chatbot's responsiveness.

The experimental evaluation conducted by the authors demonstrates the effectiveness of KBot in understanding user queries over linked data compared to baseline methods. The results indicate significant improvements in both query interpretation accuracy and response relevance, highlighting the potential of knowledge graph-based approaches for enhancing chatbot performance.

Overall, the paper contributes to the advancement of natural language understanding in chatbot systems by introducing KBot, a novel approach that leverages knowledge graphs to facilitate more accurate and context-aware responses to user queries over linked data sources.

2.6 K S. Chakraborty et al., "An AI-Based Medical Chatbot Model for Infectious Disease Prediction," in IEEE Access, vol. 10, pp. 128469-128483, 2022, doi: 10.1109/ACCESS.2022.3227208.

The paper "An AI-Based Medical Chatbot Model for Infectious Disease Prediction" by S. Chakraborty et al., published in IEEE Access in 2022, presents a novel approach to infectious disease prediction utilizing artificial intelligence (AI) within a chatbot framework. The authors propose a chatbot model that leverages AI techniques to analyze user-provided medical data and predict the likelihood of infectious diseases. The model utilizes machine learning algorithms to process and interpret various types of medical data, such as symptoms, medical history, and demographic information, to generate accurate predictions.

Key features of the proposed chatbot model include its ability to adapt and improve over time through continuous learning from user interactions and updated medical data. The chatbot interface provides a user-friendly platform for individuals to input their medical information and receive personalized predictions regarding infectious diseases. The paper highlights the potential of such AI-based chatbots in assisting healthcare professionals in early disease detection, proactive intervention, and resource allocation.

Furthermore, the authors discuss the implementation details of the chatbot model, including data preprocessing, feature selection, and model training/validation techniques. They also evaluate the performance of the model using real-world medical datasets and compare it with existing prediction methods, demonstrating its effectiveness and reliability.

In conclusion, the paper presents a promising approach to infectious disease prediction through the integration of AI technology into a chatbot framework. The proposed model offers a user-friendly and efficient solution for early disease detection, with potential applications in telemedicine, public health surveillance, and patient management.

CHAPTER 3

EXISTING METHODOLOGY

CHAPTER 3
EXISTING METHODOLOGY

The existing system of a regular chatbot with basic keyword detection typically operates on a rule-based approach, where predefined keywords or phrases trigger corresponding responses. This chatbot relies on a static set of rules or patterns to recognize user input and generate appropriate replies. It lacks the ability to understand the nuances of natural language and cannot interpret user intents beyond the predefined keywords.

In this system, the chatbot's functionality is limited to recognizing specific keywords or phrases within the user's input and selecting pre-scripted responses associated with those keywords. As a result, interactions with the chatbot may feel scripted and impersonal, leading to a less engaging user experience. Additionally, without the capability for dynamic learning or adaptation, the chatbot's responses may become outdated or irrelevant over time. Overall, while this type of chatbot can provide basic assistance or information retrieval within the scope of its predefined rules, it falls short in delivering truly conversational experiences and may struggle to meet the evolving needs of users in complex or ambiguous scenarios.

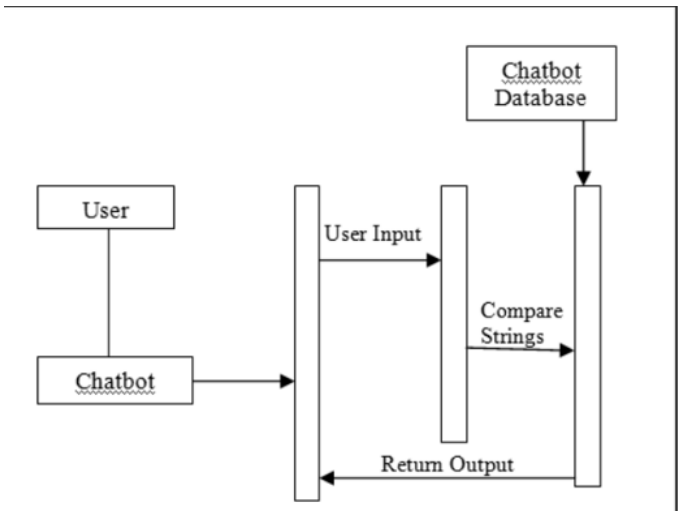

Fig 3.1 Existing Methodology

3.1 Drawbacks of the existing system

- **Limited Understanding and Lack of Context**: Chatbots relying solely on keyword detection struggle to grasp the **context** of a conversation. They recognize specific keywords or phrases but cannot understand the **nuances** or **intent** behind user queries.

- **Emotional Insensitivity**: These chatbots operate like **code-driven machines**. They do not possess the capacity to **empathize** or respond to emotional cues.

- **Fixed Responses**: Chatbots with basic keyword detection generate **predefined responses**. These answers are static and do not evolve based on user interactions.

- **Maintenance Overhead**: Regular chatbots require **constant maintenance** to stay relevant. As new keywords emerge or user queries evolve, the chatbot's database must be updated.

- **Transactional Focus**: Basic chatbots excel at handling **transactional interactions**. They can provide straightforward answers or guide users to specific resources. However, they struggle in **extended, natural conversations** where context matters more than isolated keywords.

- **Lack of Human Touch**: Users often seek a **personalized experience** when interacting with chatbots. Basic keyword-based bots lack the ability to mimic human conversation effectively.

CHAPTER 4

PROPOSED METHODOLOGY

CHAPTER 4
PROPOSED METHODOLOGY

The methodology for developing the chatbot with text chat and PDF document processing functionalities involves several steps. Initially, we configure the Flask application for handling file uploads and defining routes for user interaction. Upon receiving user queries, the chatbot utilizes a pre-trained NLP model to predict intents and generate appropriate responses. For PDF document processing, we employ PyPDFLoader to extract content from uploaded files, followed by embedding generation using Hugging Face's MiniLM-L6-v2 model. A FAISS index is then created to enable efficient similarity search. When users input queries related to PDF content, the chatbot retrieves relevant pages based on similarity search results. It then leverages Google Generative AI, specifically the Gemini Pro model, to generate responses using the extracted context and user queries. The generated responses are returned to the user, facilitating seamless interaction. Additionally, the methodology includes preprocessing steps such as text cleaning, bag-of-words representation, and intent prediction using a trained neural network model.

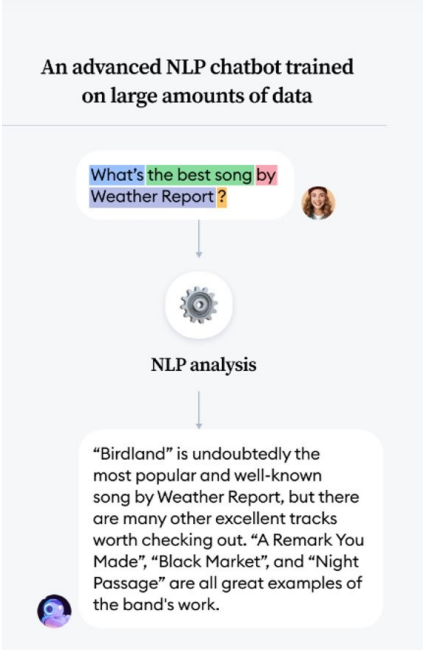

Fig 4.1 Working of proposed model.

4.1 Module Description

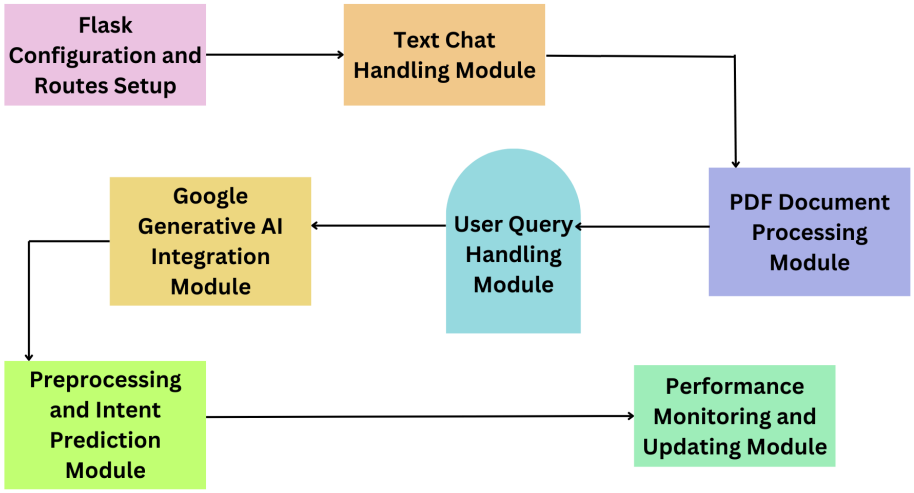

Fig 4.1.1 Modules of proposed model

4.1.1 Flask Configuration and Routes Setup

Flask configuration involves setting up the web application framework to handle file uploads and define routes for user interaction. This includes configuring the Flask application to specify the directory for file uploads, defining allowed file formats, and setting up routes to handle different types of user interactions, such as sending text queries or uploading PDF documents. These routes are responsible for receiving user requests, processing them, and returning appropriate responses. Flask's routing system allows for easy mapping of URLs to Python functions, enabling efficient handling of user interactions within the application.

4.1.2 Text Chat Handling Module

Text chat handling involves implementing functions to receive user queries and process them using a pre-trained NLP model. This module utilizes NLTK (Natural Language Toolkit) for text preprocessing tasks such as tokenization and lemmatization to prepare the user queries for analysis. The pre-trained NLP model is used to predict the intents behind the user queries and generate appropriate responses based on predefined intents stored in a JSON file. These responses are then returned to the user, providing a seamless conversational experience.

4.1.3 PDF Document Processing Module

The PDF document processing module is responsible for extracting content from uploaded PDF files and preparing it for analysis. PyPDFLoader is utilized to extract text content from PDF files and split them into individual pages. Hugging Face's MiniLM-L6-v2 model is then employed to generate embeddings for each page, which are vector representations capturing the semantic meaning of the text. These embeddings are used to construct a FAISS index, enabling efficient similarity search based on user queries related to PDF content.

4.1.4 User Query Handling Module

This module handles user queries related to PDF content by searching for relevant pages using the FAISS index constructed in the PDF document processing module. It receives user queries, searches for similar pages based on the embeddings generated from PDF content and retrieves the most relevant pages as a response. The retrieved pages are then used to generate context for the Google Generative AI model, which assists in generating responses based on the provided input text.

4.1.5 Google Generative AI Integration Module

Integrating Google Generative AI involves utilizing the Gemini Pro model to generate responses based on the input text provided, which includes the extracted context from PDF documents and the user query. The Google Generative AI model is invoked to generate responses, leveraging advanced natural language generation capabilities to provide coherent and contextually relevant answers to user queries. The generated responses are then returned to the user, completing the conversational loop and providing a seamless user experience.

4.1.6 Preprocessing and Intent Prediction Module

Preprocessing and intent prediction module preprocesses user queries by cleaning them and generating a bag-of-words representation. This involves tokenizing the user queries, lemmatizing the tokens to normalize them, and converting them into a numerical representation suitable for input into the neural network model. The trained neural network model then predicts the intents behind the user queries, selecting appropriate

responses from a predefined set of intents stored in a JSON file. These responses are returned to the user, facilitating meaningful interactions and enhancing the overall user experience.

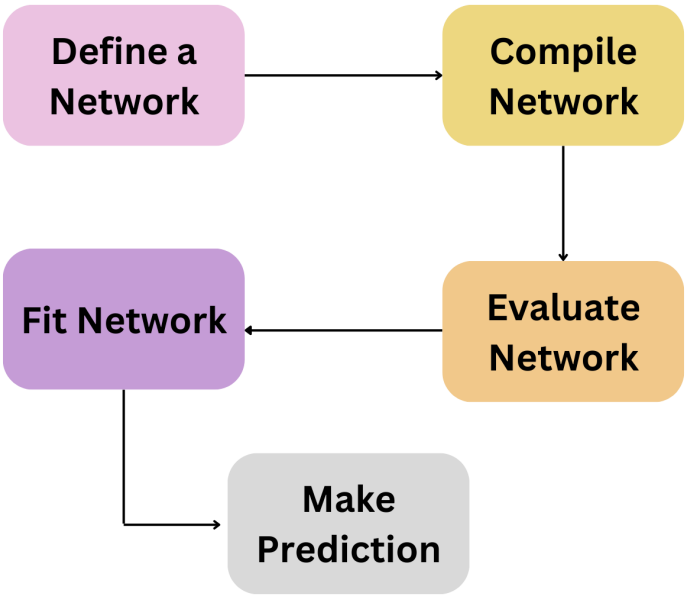

Fig 4.1.6.1 Building of Keras Model

4.1.7 Performance Monitoring and Updating Module

The performance monitoring and updating module continuously monitors the chatbot's performance, collecting user feedback to identify areas for improvement. This involves tracking various metrics such as response accuracy, user satisfaction, and system efficiency. Based on the feedback received, the chatbot's functionalities and configurations are updated to address any identified issues or shortcomings, ensuring optimal performance and user satisfaction. This iterative process of monitoring and updating ensures that the chatbot remains effective and responsive to user needs, providing a high-quality conversational experience over time.

4.2 System Specification

4.2.1 Software Specification

- System: 8-core CPU
- Storage: 256GB SSD
- Monitor: 13.3-inch (diagonal) LED-backlit
- Input Devices: Keyboard, Mouse
- Ram: 8 GB

4.2.2 Hardware Specification

- Operating system: macOS
- Coding Language: Python 3.9
- Web Framework: Flask

4.3 Advantages of proposed system

- **Seamless User Interaction**: Users can interact with the chatbot naturally, either through text conversations or by uploading PDF documents, providing a seamless and intuitive experience.
- **Efficient Information Retrieval:** The chatbot efficiently processes PDF documents, extracting relevant information and enabling users to obtain answers to their queries quickly and accurately.
- **Enhanced User Engagement:** By integrating Google Generative AI, the chatbot can generate contextually relevant responses based on user queries and extracted content from PDF documents, enhancing user engagement and satisfaction.
- **Versatile Functionality**: The project provides a versatile solution capable of handling various types of user interactions, including text-based conversations and document processing, making it suitable for a wide range of applications and use cases.
- **Automatic Response Generation:** Leveraging advanced NLP and generative AI models, the chatbot can automatically generate responses to user queries, reducing the need for manual intervention and improving efficiency.

CHAPTER 5
IMPLEMENTATION AND RESULTS

CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

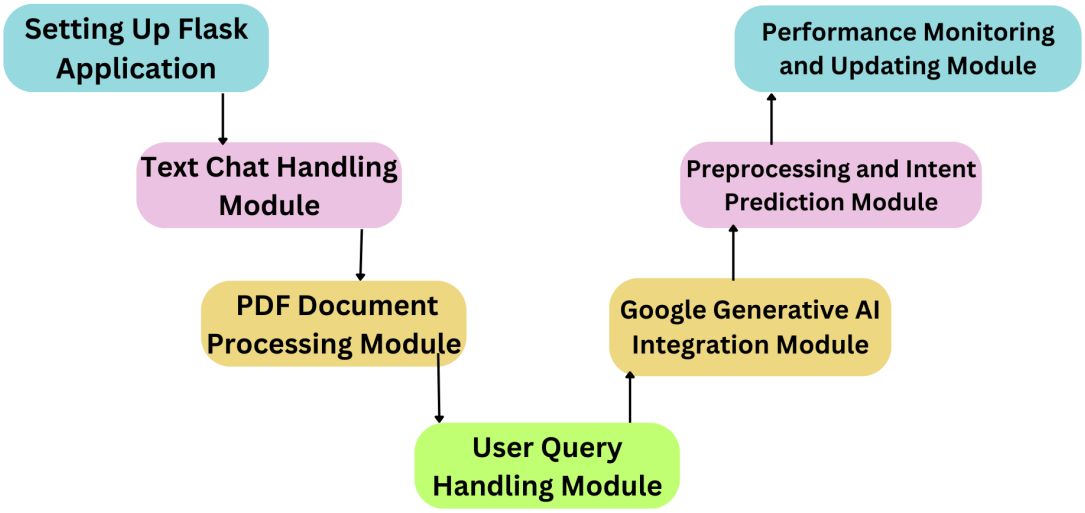

Fig 5.1.1 Implementation flow

5.1.1 Setting Up Flask Application

- Install Flask and necessary dependencies.
- Create a new Flask application.
- Configure file upload folder and allowed file formats.
- Set up routes for handling user interactions, such as text chat, PDF document processing, and user query handling.

5.1.2 Text Chat Handling Module

- Implement a function to receive user queries from the frontend.
- Preprocess user queries using NLTK for tokenization and lemmatization.
- Utilize a pre-trained NLP model to predict intents behind user queries.
- Map predicted intents to appropriate responses stored in a JSON file.
- Return the generated response to the frontend for display to the user.

5.1.3 PDF Document Processing Module

- Implement a function to handle PDF file uploads from the frontend.
- Extract text content from uploaded PDF files using PyPDFLoader.
- Generate embeddings for each page of the PDF using Hugging Face's MiniLM-L6-v2 model.
- Construct a FAISS index from the generated embeddings to enable efficient similarity search.

5.1.4 User Query Handling Module

- Implement a function to receive user queries related to PDF content.
- Search for relevant pages in the FAISS index based on user queries.
- Retrieve the most relevant pages as a response to the user query.

5.1.5 Google Generative AI Integration Module

- Implement a function to integrate Google Generative AI, specifically the Gemini Pro model.
- Prepare input text for the Generative AI model by combining extracted context from PDF documents and the user query.
- Invoke the Generative AI model to generate responses based on the provided input text.

5.1.6 Preprocessing and Intent Prediction Module

- Implement functions for preprocessing user queries, including cleaning and generating bag-of-words representation.
- Train a neural network model to predict intents behind user queries.
- Select appropriate responses from predefined intents based on predicted intents.

5.1.7 Performance Monitoring and Updating Module

- Implement mechanisms to continuously monitor the chatbot's performance, such as tracking response accuracy and user satisfaction.
- Collect user feedback to identify areas for improvement.
- Update the chatbot's functionalities and configurations based on user feedback to ensure optimal performance and user satisfaction.

5.2 RESULT

The project successfully implements a versatile chatbot capable of handling both text-based conversations and PDF document processing. Users can engage in natural language interactions with the chatbot, receiving accurate and informative responses to their queries. PDF documents can be uploaded and processed, with the chatbot efficiently extracting text content and providing targeted information retrieval through a FAISS index. Integration of Google Generative AI enhances the chatbot's conversational capabilities, generating contextually relevant responses. Preprocessing and intent prediction ensure accurate interpretation of user queries, while continuous performance monitoring and updates based on user feedback ensure optimal functionality and user satisfaction. Overall, the project delivers a seamless and efficient conversational experience, empowering users to obtain information seamlessly through both textual interactions and PDF document inquiries.

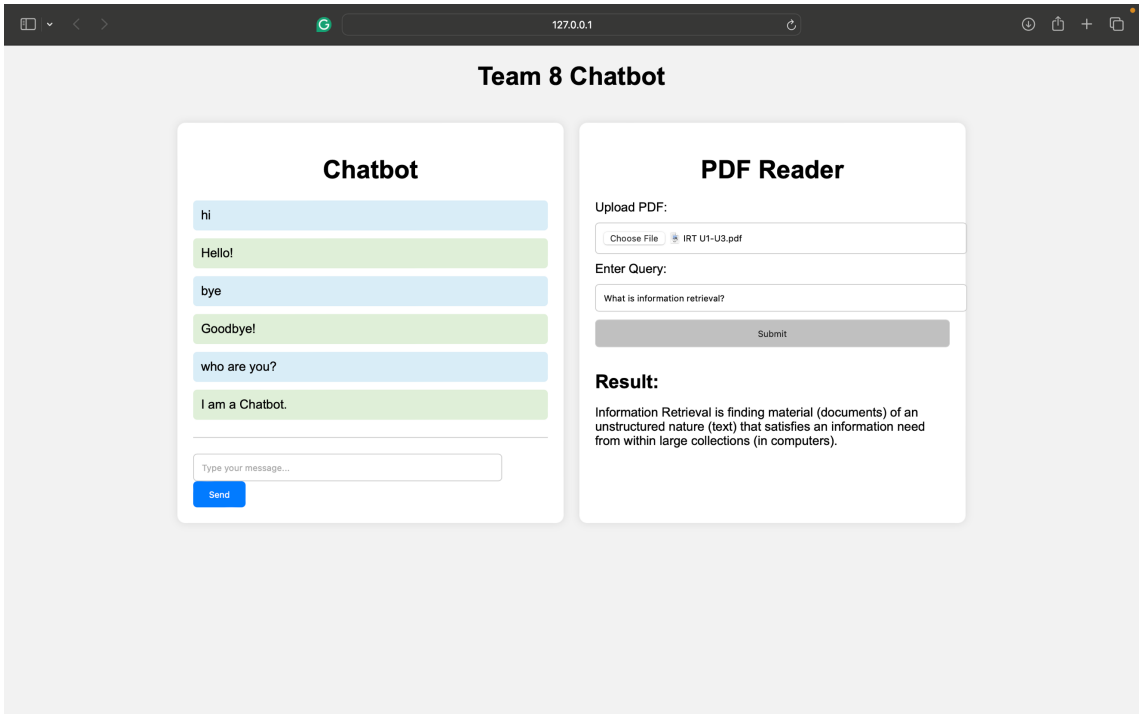

Fig 5.7.1 Sample Result of the Chatbot

CHAPTER 6
CONCLUSION AND FUTURE WORK

CHAPTER 6
CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

In conclusion, the developed chatbot demonstrates effective integration of advanced NLP, AI models, and PDF document processing techniques, offering users a seamless and efficient conversational experience. By accurately interpreting user queries, retrieving information from PDF documents, and generating contextually relevant responses, the chatbot enhances user engagement and satisfaction. Continuous monitoring and updates ensure optimal performance and adaptability to user needs over time. Overall, the project signifies the potential of leveraging modern technologies to create versatile and user-friendly solutions for information retrieval and interaction, contributing to the advancement of conversational AI applications.

6.2 FUTURE WORK

There are many future improvement aspects of this project such as,

1. Implementing the android version of the web application.
2. Adding additional functionalities like surveying the files in all formats and responding to the user queries based on the details in the files.

REFERENCES

[1] D. Carlander-Reuterfelt, Á. Carrera, C. A. Iglesias, Ó. Araque, J. F. Sánchez Rada and S. Muñoz, "JAICOB: A Data Science Chatbot," in IEEE Access, vol. 8, pp. 180672-180680, 2020, doi: 10.1109/ACCESS.2020.3024795.

[2] Mondal, M. Dey, D. Das, S. Nagpal and K. Garda, "Chatbot: An automated conversation system for the educational domain," 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAINLP), Pattaya, Thailand, 2018, pp. 1-5, doi: 10.1109/iSAI-NLP.2018.8692927.

[3] Miklosik, N. Evans and A. M. A. Qureshi, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 106530-106539, 2021, doi: 10.1109/ACCESS.2021.3100885.

[4] V. Socatiyanurak et al., "LAW-U: Legal Guidance Through Artificial Intelligence Chatbot for Sexual Violence Victims and Survivors," in IEEE Access, vol. 9, pp. 131440-131461, 2021, doi: 10.1109/ACCESS.2021.3113172.

[5] A. Ait-Mlouk and L. Jiang, "KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding Over Linked Data," in IEEE Access, vol. 8, pp. 149220-149230, 2020, doi: 10.1109/ACCESS.2020.3016142.

[6] S. Chakraborty et al., "An AI-Based Medical Chatbot Model for Infectious Disease Prediction," in IEEE Access, vol. 10, pp. 128469-128483, 2022, doi: 10.1109/ACCESS.2022.3227208.

[7] Pro.Gholap P.S,Mr.Lohote Prathamesh Yashwant,Mr.Doke Sanket Santosh,Mr.Barve Omkar Arjun,Mr.Belhaware Akshay Anil, "AI CHATBOT FOR TRANSPORTATION SYSTEM", International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.9, Issue 10, pp.c908-c913, October 2021.

[8] S. García-Méndez, F. De Arriba-Pérez, F. J. González-Castaño, J. A. Regueiro-Janeiro and F. Gil-Castiñeira, "Entertainment Chatbot for the Digital Inclusion of Elderly People Without Abstraction Capabilities," in IEEE Access, vol. 9, pp. 75878-75891, 2021, doi: 10.1109/ACCESS.2021.3080837.

[9] Miklosik, N. Evans and A. M. A. Qureshi, "The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review," in IEEE Access, vol. 9, pp. 106530-106539, 2021, doi: 10.1109/ACCESS.2021.3100885.

[10] Oulin, A., Grave, E., Bojanowski, P., Mikolov, T., Bagherinezhad, M., & Others. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv: 1607.01759.

[11] QUOC-DAI LUONG TRAN, "Enhancing Conversational Model With Deep Reinforcement Learning and Adversarial Learning ", in IEEE Access, r 10.1109/ACCESS.2023.3297652,2023

[12] AND VAN-NAM HUYNH, "Conversational AI Unleashed: A Comprehensive Review of NLP-Powered Chatbot Platforms", in IEEE Access, Volume 7 Issue 3, ISSN: 2456-8880, 2023.

[13] Arun Iyenga, Amit Sheth, "Cognitive Services and Intelligent Chatbots: Current Perspectives and Special Issue Introduction ", in IEEE 1089-7801 ,2019.

APPENDIX A: SOURCE CODE

```
import os
from flask import Flask, render_template, request, jsonify
from werkzeug.utils import secure_filename
from langchain_community.document_loaders import PyPDFLoader
from langchain_community.embeddings import
HuggingFaceEmbeddings
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain_community.vectorstores import FAISS
import nltk
from nltk.stem import WordNetLemmatizer
import json
import pickle
import warnings
import numpy as np
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import SGD
import random

app = Flask(__name__)

# Configure file upload folder
UPLOAD_FOLDER = 'uploads'
ALLOWED_EXTENSIONS = {'pdf'}
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER

# Set Google API Key
os.environ["GOOGLE_API_KEY"]                                    =
"AIzaSyCmkCUuORIyV00QkWWMdzUbR3DQqv358bM"

# Load NLTK resources
nltk.download('punkt')
nltk.download('wordnet')

# Initialize WordNet Lemmatizer
lemmatizer = WordNetLemmatizer()

# Load intents
data_file = open('intents.json').read()
intents = json.loads(data_file)

# Load words and classes
```

```
words = pickle.load(open('words.pkl', 'rb'))
classes = pickle.load(open('classes.pkl', 'rb'))

# Load and compile the model
model = tf.keras.models.load_model('model.h5')

@app.route('/')
def home():
    return render_template('new.html')

@app.route('/get')
def get_bot_response():
    user_msg = request.args.get('msg')
    response = chatbot_response(user_msg)
    return str(response)

@app.route('/process_data', methods=['POST'])
def process_data():
    # Receive file from the user
    if 'file' not in request.files:
        return jsonify({'error': 'No file part'})
    file = request.files['file']
    if file.filename == '':
        return jsonify({'error': 'No selected file'})
    if file and allowed_file(file.filename):
        filename = secure_filename(file.filename)
        filepath  =  os.path.join(app.config['UPLOAD_FOLDER'],
filename)
        file.save(filepath)

        # Load PDF using PyPDFLoader
        loader = PyPDFLoader(filepath)
        pages = loader.load_and_split()

        # Load embeddings model
        embeddings  =  HuggingFaceEmbeddings(model_name="all-
MiniLM-L6-v2")

        # Create FAISS index from PDF documents
        db = FAISS.from_documents(pages, embeddings)

        # Receive input query from the user
        query = request.form['query']
```

```
        # Query for text
        docs = db.similarity_search(query)

        # Get page content from similar documents
        content = "\n".join([x.page_content for x in docs])

        # Prepare input text for generative AI
        qa_prompt = "Use the following pieces of context to
answer the user's question. If you don't know the answer, just
say that you don't know, don't try to make up an answer.-------
---------"
        input_text   =   f"{qa_prompt}\nContext:{content}\nUser
question:\n{query}"

        # Initialize Google Generative AI model
        llm = ChatGoogleGenerativeAI(model="gemini-pro")

        # Generate response using Google Generative AI
        result = llm.invoke(input_text)

        return jsonify({'result': result.content})
    else:
        return jsonify({'error': 'Invalid file format'})

def allowed_file(filename):
    return   '.'   in   filename   and   filename.rsplit('.',
1)[1].lower() in ALLOWED_EXTENSIONS

def clean_up_sentence(sentence):
    sentence_words = nltk.word_tokenize(sentence)
    sentence_words  =  [lemmatizer.lemmatize(word.lower())  for
word in sentence_words]
    return sentence_words

def bow(sentence, words, show_details=True):
    sentence_words = clean_up_sentence(sentence)
    bag = [0]*len(words)
    for s in sentence_words:
        for i,w in enumerate(words):
            if w == s:
                bag[i] = 1
                if show_details:
                    print("found in bag: %s" % w)
    return(np.array(bag))
```

```
def predict_class(sentence, model):
    p = bow(sentence, words,show_details=False)
    res = model.predict(np.array([p]))[0]
    ERROR_THRESHOLD = 0.25
    results = [[i,r] for i,r in enumerate(res) if
r>ERROR_THRESHOLD]
    results.sort(key=lambda x: x[1], reverse=True)
    return_list = []
    for r in results:
        return_list.append({"intent": classes[r[0]],
"probability": str(r[1])})
    return return_list

def getResponse(ints, intents_json):
    tag = ints[0]['intent']
    list_of_intents = intents_json['intents']
    for i in list_of_intents:
        if(i['tag']== tag):
            result = random.choice(i['responses'])
            break
    return result

def chatbot_response(text):
    ints = predict_class(text, model)
    res = getResponse(ints, intents)
    return res

if __name__ == '__main__':
    app.run(port=8000)
```

APPENDIX B: SNAP SHOTS

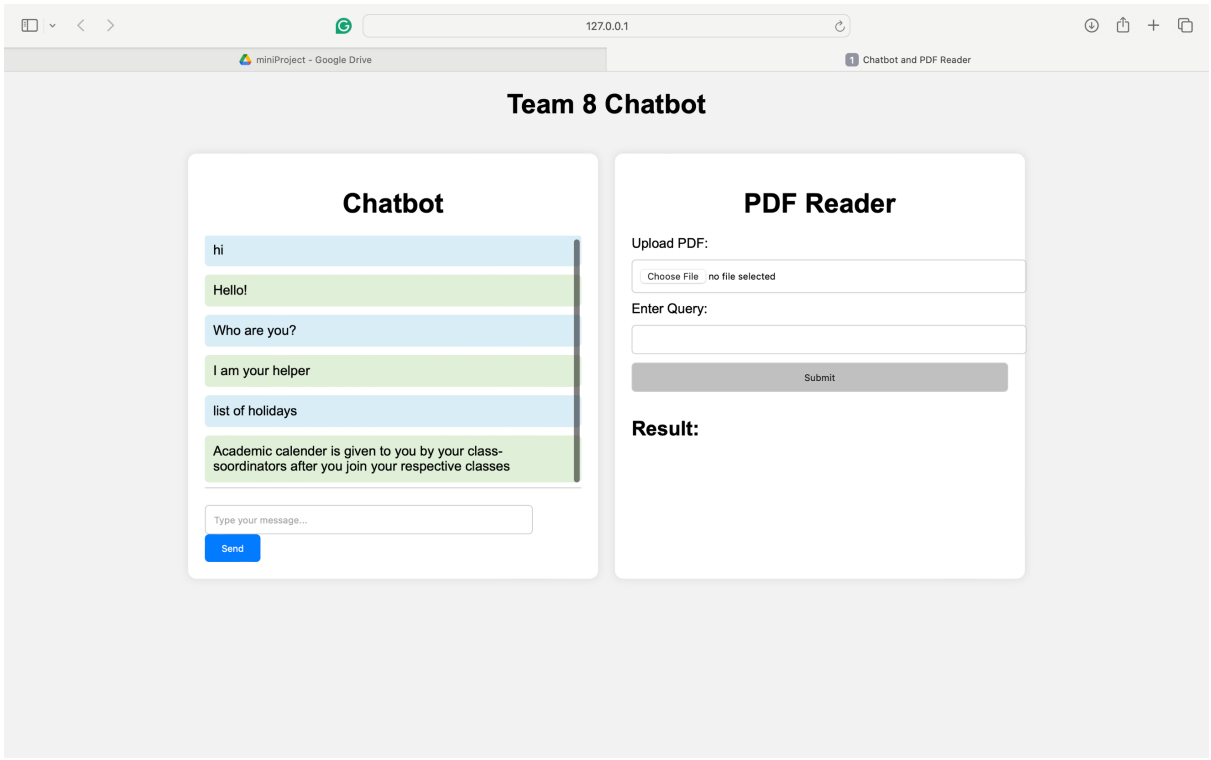

Fig B.1 Working image of Text based Chatbot

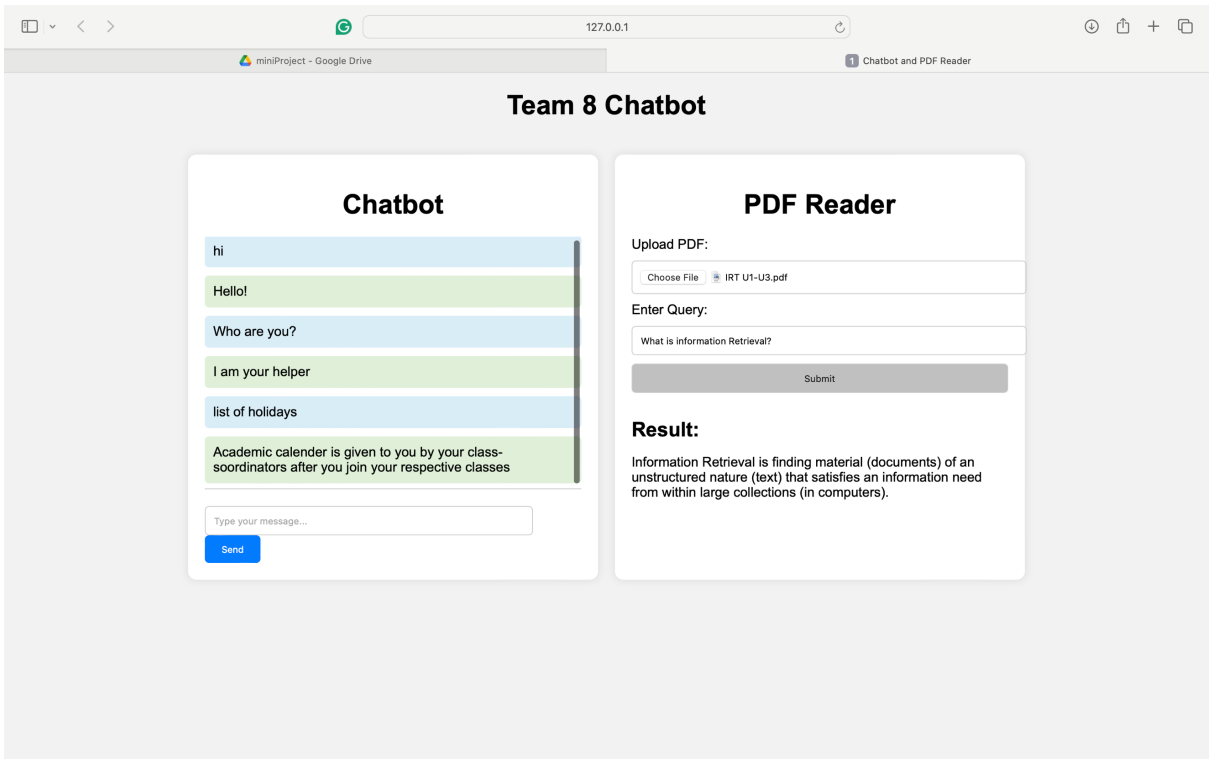

Fig B.2 Working of PDF based Chatbot