MAJOR-1 PROJECT

Midsem Report

For

<u>Chatbot using OPENAI</u>

Submitted By

Specialization	SAP ID	Name
B.Tech CSE – AIML (NH)	500083955	Milind Sharma
B.Tech CSE – AIML (NH)	500082497	Rahul Mehta
B.Tech CSE – AIML (NH)	500082578	Rahul B Nair
B.Tech CSE – AIML (NH)	500082920	Priyank Singh

Department of Informatics

School of Computer Science

UNIVERSITY OF PETROLEUM & ENERGY STUDIES,

DEHRADUN- 248007. Uttarakhand

Priyanka

Ms. Priyanka Singh
Project Guide

Mr. Anil Kumar
Cluster Head

Midsem Report

<u>Project Title</u>: Chatbot using OPENAI

Content

1. Abstract

This project presents the development of a cutting-edge AI chatbot that leverages the seamless integration of state-of-the-art technologies. By combining the linguistic prowess of Lang Chain, the contextual comprehension of Gradio, insights obtained from the PlayHT Handbook, and the potent OpenAI GPT-3.5 model, this chatbot exhibits remarkable adaptability and responsiveness.

In addition to generating human-like text-based responses, the chatbot expands its capabilities to deliver high-fidelity audio feedback, elevating the user experience. This fusion highlights the immense potential of contemporary AI technologies, indicating the rise of conversational agents proficient in delivering precise, valuable, and engaging interactions across both textual and auditory modalities.

2. Introduction

At the forefront of technological innovation, we present an AI chatbot meticulously engineered through the integration of cutting-edge technologies. This fusion encompasses the linguistic prowess of Lang Chain, the contextual understanding of Gradio, insights obtained from the PlayHT Handbook, and the potent OpenAI GPT-3.5 model.

This convergence empowers the chatbot with remarkable adaptability, enabling versatile and responsive interactions. In a pioneering departure from tradition, our chatbot transcends text-based responses by offering high-fidelity audio feedback. This enhancement elevates user experiences, providing dynamic auditory interactions that promote accessibility and engagement.

This synthesis of innovation underscores the immense potential of contemporary AI technologies, ushering in a new era of precise, valuable, and engaging interactions that blend textual and auditory modalities. Throughout this report, we delve into AI’s transformative impact, heralding an era of interactive and inclusive conversations.

3. Literature Review

The project builds upon existing research in the fields of natural language processing and conversational AI. It incorporates one of the latest advancements in language models, such as GPT- 3.5, which have demonstrated remarkable proficiency in generating human-like text. Additionally, the integration of text-to-speech (TTS) technology via the PlayHT API aligns with the trend of making AI-driven interactions more immersive and accessible.

- Building Upon Existing Research: The project leverages prior NLP and conversational AI research, avoiding redundant work and expediting chatbot development.

- Incorporating Advanced Language Models: The project's key innovation is the use of the state-of-the-art OpenAI GPT-3.5 model, renowned for its human-like text generation capabilities.

- Remarkable Proficiency in Text Generation: GPT-3.5's exceptional text generation skills enable the chatbot to engage in context-aware, user-friendly conversations.

- Integration of Text-to-Speech (TTS) Technology: The project integrates TTS technology via PlayHT API, enhancing the chatbot's accessibility and user engagement through audio feedback.

- Immersive and Accessible AI Interactions: The project aligns with industry trends by offering immersive AI interactions in both text and audio formats, ensuring inclusivity and user satisfaction.

4. Problem Statement

The primary problem addressed by this project is the need for an advanced chatbot capable of delivering dynamic responses that include both text and audio components.

Conventional chatbots often provide only text-based interactions, limiting their ability to engage users effectively, especially in scenarios where audio feedback is preferred or necessary. This project seeks to bridge this gap by creating a chatbot that can seamlessly switch between text and audio responses.

5. Objectives:

The key objectives of the project are as follows:

- Develop a chatbot using the OpenAI GPT-3.5 model for text-based conversation.
- Integrate the PlayHT API to convert text responses into audio format.
- Provide users with a multi-modal conversational experience, including text and audio interactions.
- Enhance user engagement and accessibility by offering audio responses, particularly for individuals with visual impairments.
- Enable the publication of the chatbot for wider usage, potentially as an interactive webservice.

6. Methodology:

The methodology involves the following steps:

- Setting up environment variables for API keys and user IDs for OpenAI and PlayHT.
- Configuring the OpenAI GPT-3.5 model with desired parameters and a response template.
- Defining functions for generating audio from text using the PlayHT API.
- Creating a chat interface using Gradio to facilitate user interactions.
- Combining text response generation from the OpenAI model with audio generation from PlayHT to provide a multi-modal response.
- Enabling the publication of the chatbot code via the Hugging Face Hub for broader accessibility.

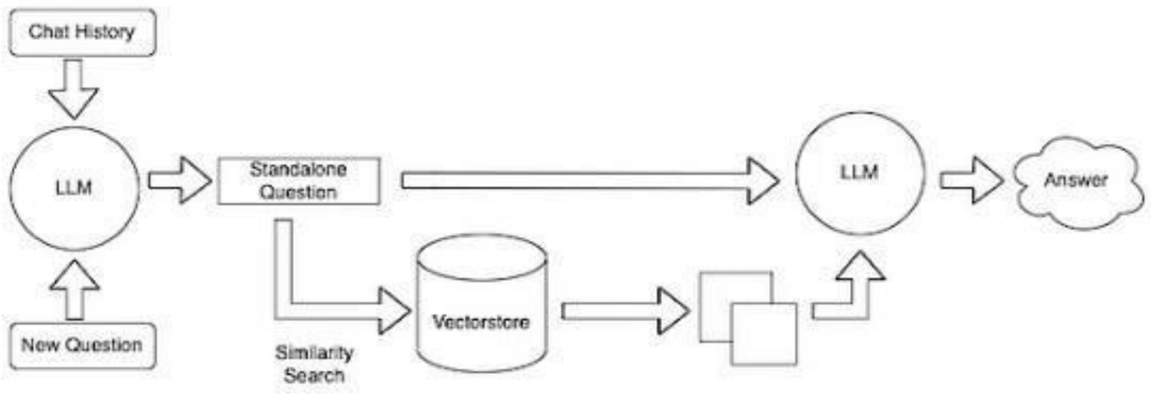
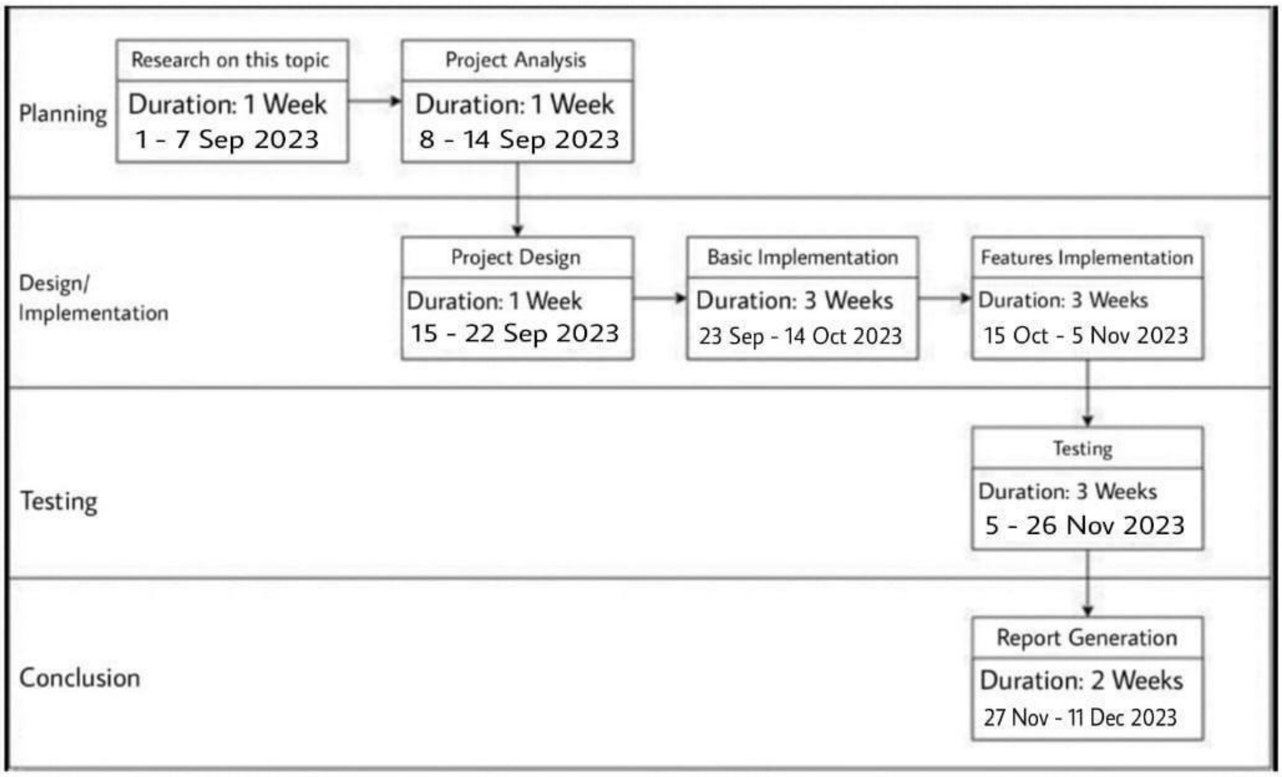
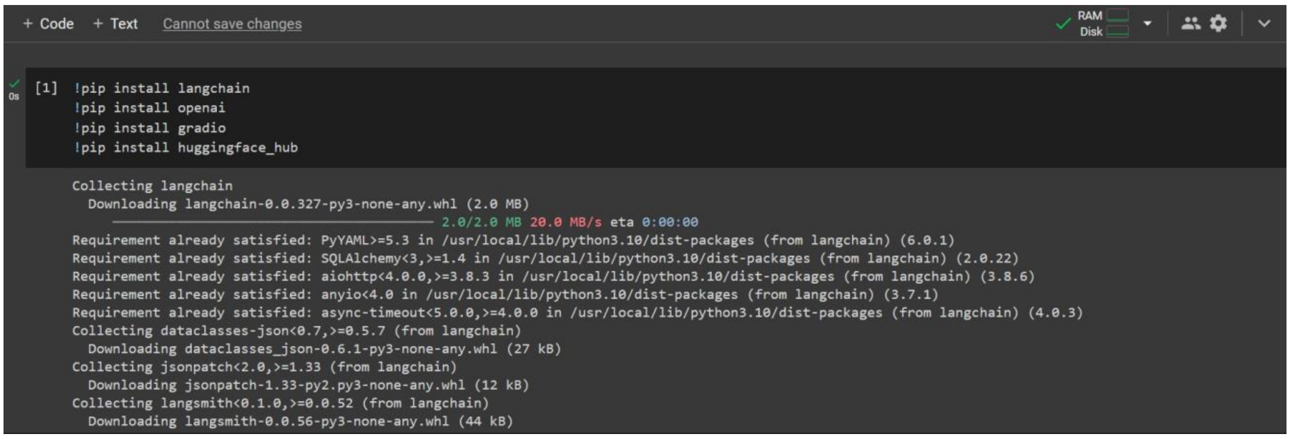


Fig (1) A diagram of the process used to create a chatbot

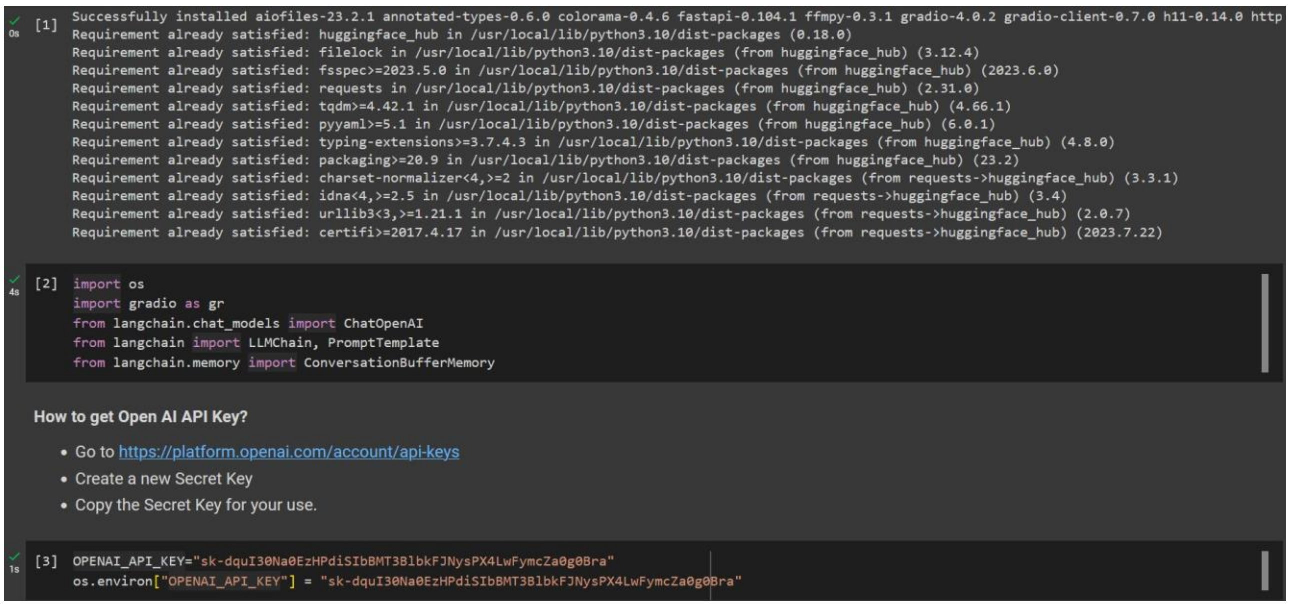
7.PERT Chart



8. Code:



```
[1] !pip install langchain
    !pip install openai
    !pip install gradio
    !pip install huggingface_hub

    Collecting langchain
      Downloading langchain-0.0.327-py3-none-any.whl (2.0 MB)
         ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 20.0 MB/s eta 0:00:00
    Requirement already satisfied: PyYAML>=5.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (6.0.1)
    Requirement already satisfied: SQLAlchemy<3,>=1.4 in /usr/local/lib/python3.10/dist-packages (from langchain) (2.0.22)
    Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.8.6)
    Requirement already satisfied: anyio<4.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (3.7.1)
    Requirement already satisfied: async-timeout<5.0.0,>=4.0.0 in /usr/local/lib/python3.10/dist-packages (from langchain) (4.0.3)
    Collecting dataclasses-json<0.7,>=0.5.7 (from langchain)
      Downloading dataclasses_json-0.6.1-py3-none-any.whl (27 kB)
    Collecting jsonpatch<2.0,>=1.33 (from langchain)
      Downloading jsonpatch-1.33-py2.py3-none-any.whl (12 kB)
    Collecting langsmith<0.1.0,>=0.0.52 (from langchain)
      Downloading langsmith-0.0.56-py3-none-any.whl (44 kB)
```



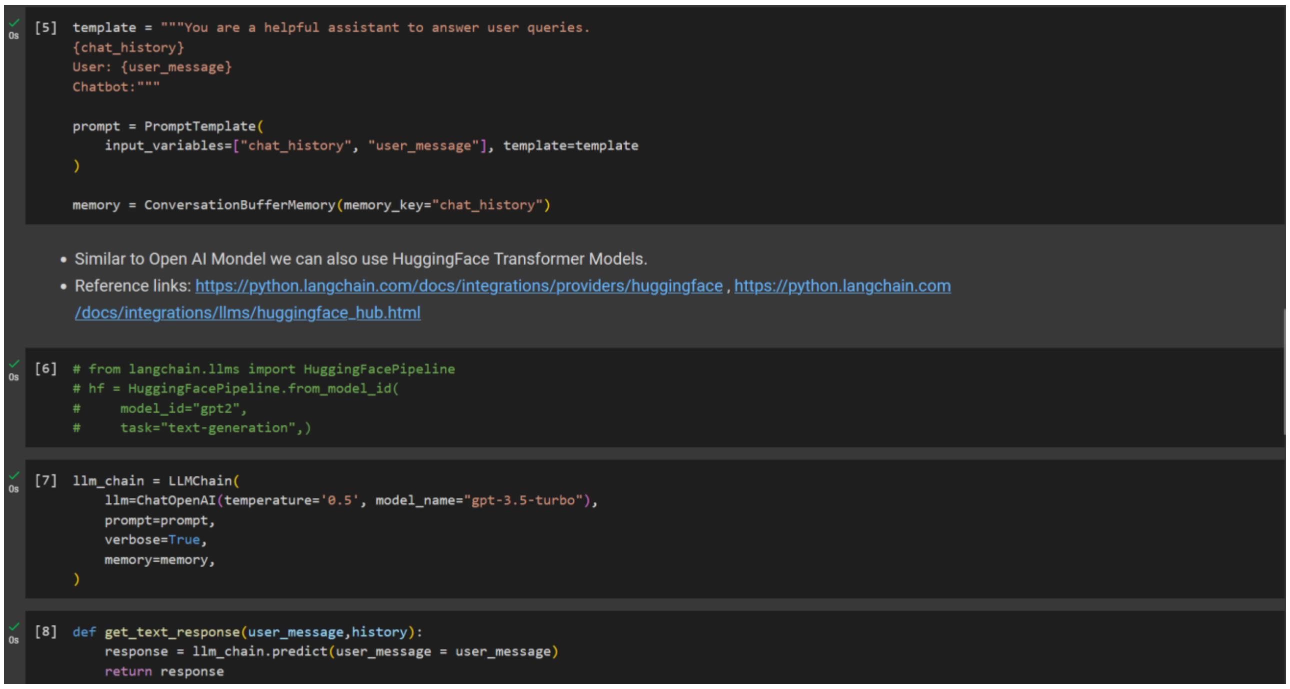
```
[1] Successfully installed aiofiles-23.2.1 annotated-types-0.6.0 colorama-0.4.6 fastapi-0.104.1 ffmpy-0.3.1 gradio-4.0.2 gradio-client-0.7.0 h11-0.14.0 http
    Requirement already satisfied: huggingface_hub in /usr/local/lib/python3.10/dist-packages (0.18.0)
    Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (3.12.4)
    Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (2023.6.0)
    Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (2.31.0)
    Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.66.1)
    Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (6.0.1)
    Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (4.8.0)
    Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.10/dist-packages (from huggingface_hub) (23.2)
    Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (3.3.1)
    Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (3.4)
    Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (2.0.7)
    Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface_hub) (2023.7.22)
```

```
[2] import os
    import gradio as gr
    from langchain.chat_models import ChatOpenAI
    from langchain import LLMChain, PromptTemplate
    from langchain.memory import ConversationBufferMemory
```

How to get Open AI API Key?

- Go to https://platform.openai.com/account/api-keys
- Create a new Secret Key
- Copy the Secret Key for your use.

```
[3] OPENAI_API_KEY="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"
    os.environ["OPENAI_API_KEY"] = "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"
```



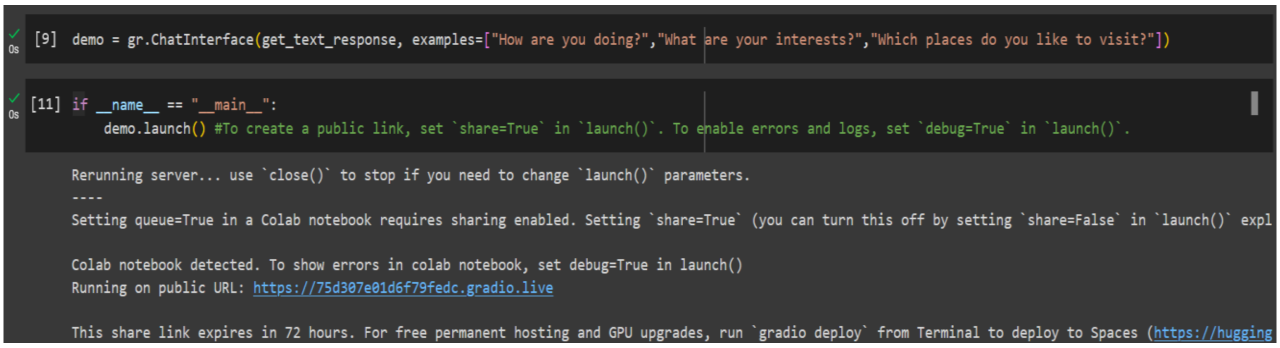
```
[5] template = """You are a helpful assistant to answer user queries.
    {chat_history}
    User: {user_message}
    Chatbot:"""

    prompt = PromptTemplate(
        input_variables=["chat_history", "user_message"], template=template
    )

    memory = ConversationBufferMemory(memory_key="chat_history")
```

- Similar to Open AI Mondel we can also use HuggingFace Transformer Models.
- Reference links: https://python.langchain.com/docs/integrations/providers/huggingface , https://python.langchain.com/docs/integrations/llms/huggingface_hub.html

```
[6] # from langchain.llms import HuggingFacePipeline
    # hf = HuggingFacePipeline.from_model_id(
    #     model_id="gpt2",
    #     task="text-generation",)
```

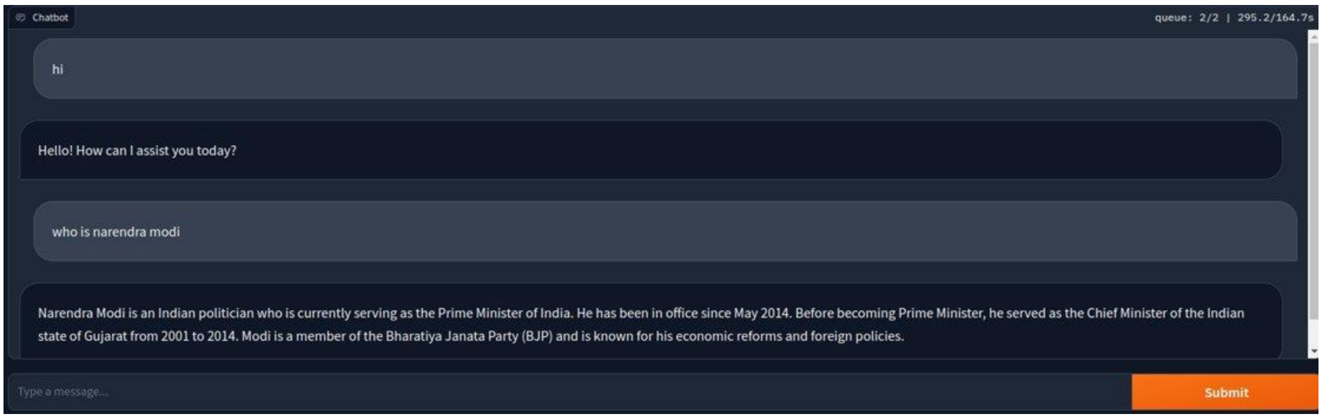
```
[7] llm_chain = LLMChain(
        llm=ChatOpenAI(temperature='0.5', model_name="gpt-3.5-turbo"),
        prompt=prompt,
        verbose=True,
        memory=memory,
    )
```

```
[8] def get_text_response(user_message,history):
        response = llm_chain.predict(user_message = user_message)
        return response
```



```
[9]  demo = gr.ChatInterface(get_text_response, examples=["How are you doing?","What are your interests?","Which places do you like to visit?"])

[11] if __name__ == "__main__":
         demo.launch() #To create a public link, set `share=True` in `launch()`. To enable errors and logs, set `debug=True` in `launch()`.

     Rerunning server... use `close()` to stop if you need to change `launch()` parameters.
     ----
     Setting queue=True in a Colab notebook requires sharing enabled. Setting `share=True` (you can turn this off by setting `share=False` in `launch()` expl

     Colab notebook detected. To show errors in colab notebook, set debug=True in launch()
     Running on public URL: https://75d307e01d6f79fedc.gradio.live

     This share link expires in 72 hours. For free permanent hosting and GPU upgrades, run `gradio deploy` from Terminal to deploy to Spaces (https://hugging
```

Implementation:



Chatbot

queue: 2/2 | 295.2/164.7s

hi

Hello! How can I assist you today?

who is narendra modi

Narendra Modi is an Indian politician who is currently serving as the Prime Minister of India. He has been in office since May 2014. Before becoming Prime Minister, he served as the Chief Minister of the Indian state of Gujarat from 2001 to 2014. Modi is a member of the Bharatiya Janata Party (BJP) and is known for his economic reforms and foreign policies.