**PROJECT PROPOSAL**

**MediBot**

**Focused on Checking Symptoms and Diagnosing Diseases**

**Course Instructor:**

Ms. Mamoona Akbar

**Group Members:**

* Abdullah Fayyaz (21L - 5208)
* Murtaza Ahmad (21L - 5173)

**Research Problem and Motivation:**

Accurate and timely disease diagnosis is a critical challenge in healthcare, especially when patients lack direct access to healthcare professionals. Healthcare systems are increasingly burdened by high volumes of data, making it difficult for professionals to provide timely diagnosis. MediBot aims to address these challenges by offering a conversational AI chatbot that collects vital patient information, analyzes symptoms, and suggests potential diseases and treatment plans. This project uses modern NLP techniques to deliver efficient, accurate, and personalized healthcare assistance to patients.

**NLP Techniques to be Used:**

1. **Fine-tuning OpenAI GPT-4o-mini**: The central element of MediBot will be the fine-tuning of OpenAI’s GPT-4o-mini on a custom dataset. The fine-tuning will be based on both positive and negative conversational examples to help the model understand how to handle patient input. Positive examples will focus on stage-wise conversations (e.g., moving from basic info to symptoms to diagnosis), while negative examples will involve users trying to jump between stages without answering the current stage questions.
2. **Prompt Engineering with Few-Shot Examples**: At each conversational stage, the chatbot will use prompt engineering, providing the model with few-shot examples to ensure that the generated responses are accurate, contextually appropriate, and relevant. These examples will guide the model in generating responses based on the user’s input at each stage (basic info, symptoms, family history, treatment plans, etc.).  
   * **Stage Analyzer**: The chatbot will use a stage analyzer to detect which stage of the conversation the user is in and respond accordingly. The analyzer will ensure the conversation follows the correct sequence (e.g., basic info → symptoms → treatment).
   * **Few-Shot Examples**: The chatbot will generate responses using few-shot examples tailored for each stage. These examples will ensure the chatbot responds to the patient’s input at the correct stage with appropriate medical information.
3. **Conversational AI**: Using **LangChain**, the chatbot will maintain the context of the conversation and manage the flow of dialogue. LangChain will help ensure the conversation progresses smoothly through the various stages. The chatbot will be able to handle interruptions or requests to skip stages by reminding the user to follow the flow or asking clarifying questions.

**Datasets and Tools to be Used:**

1. **Custom Dataset**: A custom dataset will be created for fine-tuning the model. This dataset will include **20-30 examples** of full conversations. Positive examples will reflect stage-wise conversations where the user answers all questions in order, while negative examples will feature users attempting to skip stages or jump ahead. This will allow the model to learn the appropriate sequence of dialogue and handle users who deviate from it.
2. **Tools**:  
   * **OpenAI GPT-4o-mini**: This pre-trained model will be fine-tuned with the custom dataset for handling medical conversations and disease diagnosis tasks.
   * **LangChain**: Used to manage the dialogue flow and ensure that the chatbot correctly handles different stages of the conversation.
   * **Streamlit**: A simple app framework will be used to build a **Streamlit demo app**, allowing you to showcase the chatbot’s functionality interactively. The app will provide users with a simple interface to engage with the MediBot.
   * **Python**: The entire project will be implemented in Python, leveraging the necessary libraries and APIs for model fine-tuning, prompt engineering, and app development.

**Evaluation Metrics:**

To evaluate MediBot’s effectiveness, the following metrics will be used:

1. **Accuracy**: Measures how accurately the chatbot identifies diseases and recommends treatment plans compared to expert healthcare professionals. Accuracy will be particularly important in assessing the chatbot’s ability to diagnose based on symptoms and medical history.
2. **F1-Score**: Used to evaluate the chatbot’s performance in extracting relevant information from the user (such as symptoms or family history). The F1-score will balance precision and recall to minimize both false positives and false negatives.
3. **BLEU/ROUGE**: These metrics will be used to evaluate the fluency and relevance of the chatbot’s responses, especially when generating treatment plans or summarizing diagnostic information. BLEU and ROUGE will compare generated responses to reference answers, helping ensure the chatbot remains consistent and provides high-quality information.
4. **Perplexity**: Measures how naturally the chatbot generates text. Lower perplexity indicates more fluent and coherent responses, which is important for maintaining a natural conversation with the patient.

**Project Implementation Flow:**

1. **Stage Detection**: The chatbot first detects which stage the user is in (e.g., basic info, symptoms, medical history, etc.). This will be achieved through a stage analyzer that classifies the user input based on the conversation flow.
2. **Response Generation**: Based on the detected stage, the chatbot will generate an appropriate response using fine-tuned GPT-4o-mini. The model will use the few-shot examples to understand how to respond at each stage.
3. **Ongoing Dialogue**: As the conversation progresses, the chatbot will maintain context, ensuring that the conversation flows smoothly from one stage to the next. If a user tries to skip a stage, the chatbot will ask clarifying questions to guide them back to the correct sequence.
4. **Final Diagnosis & Treatment Plan**: Once sufficient information is gathered, the chatbot will generate a final disease diagnosis and suggest a treatment plan. These responses will be based on the conversational history and fine-tuned model outputs.
5. **Streamlit Demo**: A **Streamlit app** will be developed for demo purposes, allowing users to interact with the chatbot and see its responses in real time. This will serve as a prototype for showcasing the chatbot’s potential in handling patient conversations and providing medical insights.

This updated proposal reflects the new details about your project, including the custom dataset, the use of **Streamlit** for the demo app, and the fine-tuning approach with **OpenAI GPT-4o-mini** using positive and negative examples for stage-wise conversations.