AI/ML Intern Assignment:

Position: AI/ML Intern
Deadline: 48 Hours from Assignment Receipt
Submission: Provide all necessary submissions on Career page.

I. Assignment Overview

You are tasked with developing an intelligent Hiring Assistant chatbot for "TalentScout," a fictional recruitment agency specializing in technology placements. The chatbot should assist in the initial screening of candidates by gathering essential information and posing relevant technical questions based on the candidate's declared tech stack. This project will allow you to demonstrate your understanding of LLM’s

II. Purpose of Prompting

The candidate will design prompts that guide the underlying language model to:

1. **Gather Initial Candidate Information**: Collect essential details such as name, contact information, years of experience, and desired positions.
2. **Generate Technical Questions**: Based on the candidate’s specified tech stack (e.g., programming languages, frameworks, tools), generate relevant technical questions to assess their proficiency.
3. **Ensure Coherent and Context-Aware Interactions**: Maintain the flow of conversation and context to provide a seamless user experience.

III. Requirements

Functionality

- **User Interface**:
  - Develop a clean and intuitive UI using Streamlit or Gradio where candidates can interact with the chatbot.
- **Chatbot Capabilities**:
  - **Greeting**: The chatbot should greet candidates upon initiation and provide a brief overview of its purpose. The chatbot should also exit whenever a conversation-ending keyword is encountered.
  - **Information Gathering**:
    - Collect essential candidate details such as:
      - Full Name
      - Email Address
      - Phone Number
      - Years of Experience
      - Desired Position(s)
      - Current Location
      - Tech Stack
  - **Tech Stack Declaration**:
    - Prompt candidates to specify their tech stack, including programming languages, frameworks, databases, and tools they are proficient in.
  - **Technical Question Generation**:
    - Based on the declared tech stack, generate a set of 3-5 technical questions tailored to assess the candidate's proficiency in each specified technology.
    - Example: If a candidate lists Python and Django, generate questions related to Python programming and Django framework.
  - **Context Handling**:
    - Maintain the context of the conversation to handle follow-up questions and ensure a coherent flow.
  - **Fallback Mechanism**:
    - Provide meaningful responses when the chatbot does not understand the user input or when unexpected inputs are received.
    - It should not deviate from the Purpose.
  - **End Conversation**:
    - Gracefully conclude the conversation, thanking the candidate and informing them about the next steps.

Technical Specifications

- **Programming Language**: Python

- **Libraries & Tools**:
    - **Streamlit**: For developing the frontend interface.
    - **Large Language Models**: Utilize pre-trained models (e.g., GPT-3/4, llama, etc).
- **Deployment**:
    - Local deployment is acceptable.
    - Bonus points for Deploying on a cloud platform (e.g. AWS, GCP, etc) and providing a live demo link.

Prompt Engineering

- **Design Effective Prompts**:
    - Craft prompts that accurately gather candidate information and generate relevant technical questions based on the declared tech stack.
    - Ensure prompts are clear, concise, and guide the language model to produce desired outputs.
- **Optimize for Diverse Tech Stacks**:
    - Handle a variety of technologies and frameworks, ensuring questions are relevant and appropriately challenging.
- **Manage Sensitive Information**:
    - Implement techniques to handle and store sensitive candidate information securely and in compliance with data privacy best practices.

Data Handling

- **Simulated Data**:
    - Use simulated or anonymized data for any backend processes, such as storing candidate information and technical responses.
- **Data Privacy**:
    - Ensure that all candidate data is handled in compliance with data privacy standards (e.g., GDPR).

Documentation

- **README File**:
    - **Project Overview**: Brief description of the Hiring Assistant chatbot and its capabilities.
    - **Installation Instructions**: Detailed steps to set up and run the application locally.
    - **Usage Guide**: Clear and Concise Readme file.
    - **Technical Details**: Libraries used, model details, and architectural decisions.
    - **Prompt Design**: Explanation of how prompts were crafted to handle information gathering and technical question generation.
    - **Challenges & Solutions**: Discuss any challenges faced during development and how they were addressed.

Code Quality

- **Structure & Readability**:
  - Ensure the code is well-structured, modular, and follows best practices for readability and maintainability.
- **Documentation**:
  - Include comments and docstrings where necessary to explain complex logic and functions.
- **Version Control**:
  - Use Git for version control, with clear commit messages and a well-organized repository.

IV. Deliverables

Source Code

- A complete and well-documented codebase is uploaded to a public Git repository OR zip file.

Documentation

- Comprehensive README as outlined above.

Demo

- A live demo link, if Using a tool such as LOOM.
- Alternatively, a short video walkthrough demonstrating the chatbot's features and interactions along with the source code.

Optional Enhancements (Bonus Points)

- **Advanced Features**:
  - Integrate sentiment analysis to gauge candidate emotions during the conversation.
  - Implement multilingual support to interact with candidates in different languages.
  - Provide personalized responses based on user history or preferences.
- **UI Enhancements**:
  - Improve the aesthetic appeal and user experience of the Streamlit interface with custom styling and interactive elements.
- **Performance Optimization(Bonus)**:
  - Ensure the chatbot responds promptly and can handle multiple user inputs efficiently.

V. Evaluation Criteria

Technical Proficiency (40%)

- Correct implementation of hiring assistant functionalities.
- Effective use of LLM's tailored to the defined purpose.
- Quality, efficiency, and scalability of the code.

Problem-Solving & Critical Thinking (30%)

- Ability to design effective prompts for information gathering and technical question generation.
- Creative solutions to maintain context and manage conversation flow.
- Addressing challenges related to data handling and user interactions.

User Interface & Experience (15%)

- Ease of interaction and overall user experience tailored to a hiring context.

Documentation & Presentation (10%)

- Clarity and comprehensiveness of the README.
- Quality of the demo presentation.

Optional Enhancements (5%)

- Implementation of additional features or optimizations beyond the basic requirements.

VI. Submission Guidelines

- **Deadline**: 48 Hours
- **Submission Method**: Provide the Git repository link and demo link (if applicable) via Career Portal.
- **File Format**: Ensure all files are accessible and the repository is public or accessible with the provided link.

VII. Additional Resources

- **Streamlit Documentation**: https://docs.streamlit.io/
- **Prompt Engineering Guide**: https://www.promptingguide.ai/
- **GitHub Guides**: https://guides.github.com/
- **Data Privacy Best Practices**: https://gdpr.eu/

PG-AGI