



TalentScout Hiring Assistant Chatbot

🔔 Project Overview

TalentScout Hiring Assistant is an intelligent, LLM-powered chatbot designed to assist a fictional recruitment agency (TalentScout) in the **initial screening of technology candidates**. The chatbot interacts with candidates, gathers essential information, asks them to declare their tech stack, and dynamically generates technical interview questions tailored to their skills.

This project demonstrates practical usage of **Large Language Models (LLMs)**, **prompt engineering**, and **context-aware conversational design** using **Streamlit**.



😢 Features

- Professional greeting and onboarding
 - Step-by-step candidate information collection
 - Tech stack declaration (languages, frameworks, databases, tools)
 - Automatic generation of 3–5 technical questions per technology
 - Context-aware conversation flow using session state
 - Graceful exit handling (`exit`, `quit`, `bye`)
 - Privacy-friendly design (no persistent storage)
-



Tech Stack

- **Programming Language:** Python
 - **Frontend:** Streamlit
 - **LLM:** OpenAI GPT (can be replaced with LLaMA or other models)
 - **State Management:** Streamlit Session State
-



👂 Project Structure

```
TalentScout-Hiring-Assistant/
|
├── app.py          # Main Streamlit application
├── README.md       # Project documentation
├── requirements.txt # Dependencies
└── .env            # API keys (not committed)
```

⌚ Installation & Setup

Clone the Repository

```
git clone https://github.com/your-username/talentscout-hiring-assistant.git  
cd talentscout-hiring-assistant
```

🤔 Create Virtual Environment (Optional)

```
python -m venv venv  
source venv/bin/activate # Windows: venv\Scripts\activate
```

😊 Install Dependencies

```
pip install -r requirements.txt
```

🛠️ Add Environment Variables

Create a `.env` file:

```
OPENAI_API_KEY=your_api_key_here
```

😊 Run the Application

```
streamlit run app.py
```

🧠 Prompt Design Strategy

System Prompt

Defines the chatbot's role strictly as a **hiring assistant**, preventing deviation from recruitment tasks.

Information Gathering Prompts

Designed to collect candidate details **step-by-step**, ensuring clarity and reduced cognitive load.

Technical Question Generation Prompt

Uses the declared tech stack to generate **role-relevant, multi-difficulty interview questions**.

Data Privacy & Compliance

- No personal data is stored permanently
 - All data is session-based and cleared on exit
 - API keys are securely stored using environment variables
 - GDPR-friendly simulated data handling
-

Challenges & Solutions

Challenge	Solution
Maintaining conversation context	Used Streamlit session state
Preventing off-topic responses	Strict conversation flow & prompts
Handling unexpected input	Fallback messages & exit keywords

Future Enhancements

- Sentiment analysis during conversation
 - Multilingual support
 - Resume upload & parsing
 - Database integration for recruiters
 - Role-based difficulty adjustment
-

Demo

A short video walkthrough (Loom) demonstrates: - Candidate onboarding - Tech stack input - Dynamic question generation

Author

Developed by **[Your Name]** as part of an LLM-based Hiring Assistant assignment.

 This project is suitable for academic submission, interviews, and portfolio demonstration.