

TalentScout-Hiring-Chatbot

An intelligent Hiring Assistant chatbot developed for TalentScout, a fictional recruitment agency specializing in technology placements. The chatbot streamlines technical evaluations by asking domain-specific interview questions, evaluating candidate responses, and analyzing sentiment and proficiency in real-time.

Features

- **Information Collection:**
Gathers full name, experience, contact info, and preferred tech stack.
- **LLM-Powered Question Generation:**
Dynamically generates technical questions based on tech stack (e.g., Python, React, Django).
- **Context-Aware Interaction:**
Maintains coherent flow using memory-aware prompt chaining.
- **Fallback & Error Handling:**
Handles ambiguous inputs gracefully without breaking flow.
- **Scoring System:**
Evaluate candidate responses with scoring logic or sentiment analysis.
- **Streamlit Interface:**
Clean and intuitive web-based chat UI.

Tech Stack

- Language: Python
- Libraries: LangChain, Streamlit, etc.
- Model: Groq LLMs.

Setup Instructions

Follow these steps to set up and run the chatbot locally.

1. Clone the Repository

```
git clone https://github.com/nishan129/TalentScout-Hiring-Chatbot.git
cd TalentScout-Hiring-Chatbot
```

2. Create a Virtual Environment You can use either uv, venv, or conda.:

```
uv venv
source .venv/bin/activate # On Windows: .venv\Scripts\activate
```

3. Install Dependencies:

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

4. Set up Environment Variables Create a .env file in the project root and add your GROQ API key:

```
GROQ_API_KEY=your_api_key_here
```

5. Run the Streamlit App

```
streamlit run main.py
```

Technologies Used

- Streamlit – UI Framework for ML apps
- LangChain – LLM orchestration and prompt management
- GROQ API – LLM inference engine (can switch to OpenAI, Anthropic, etc.)
- Python – Core programming language
- dotenv – For managing environment variables

Project Structure

TalentScout-Hiring-Chatbot/

```
|—— main.py
|—— src/
|       |—— bot/
|       |—— answer_bot/
|       |—— analysis_bot/
|       |—— optimize/
|       |—— experiments/
|       |—— utils/
|       |—— prompts/
|—— .env.example
|—— requirements.txt
|—— setup.py
|—— README.md
```

Optional Enhancements

- Add authentication and user session tracking
- Integrate with applicant tracking systems (ATS)
- Add support for voice input and feedback
- Save candidate session transcripts