

TalentScout – AI-Powered Hiring Assistant

Project Documentation

1. Introduction

TalentScout is an **AI-driven interactive hiring assistant** designed to automate the **initial candidate screening process** for recruiters. It engages applicants through an **intuitive chat interface**, collects essential details, validates them, and evaluates their technical knowledge dynamically. The assistant leverages **Natural Language Processing (NLP)** for intelligent scoring and provides an overall summary for recruiters.

2. Objectives

- Automate the **pre-screening process** for recruitment.
 - Collect **accurate candidate details** with input validation.
 - Assess candidate **technical knowledge** using **semantic similarity**.
 - Provide **AI-based scoring on a 20-point scale** for better candidate ranking.
-

3. Key Features

- **Interactive Chat Flow** – Conversational interface using **Gradio**.
- **Input Validation** – Name, email, and phone number checks.
- **Dynamic Questioning** – Based on candidate's chosen skill (**Python or Java**).
- **Follow-up Questions** – Checks for **specific libraries or frameworks** knowledge.

- **AI Scoring** – Uses **Sentence Transformers** to compute semantic similarity, scaled to **20 points**.
 - **Final Summary** – Displays score and recruiter-friendly output.
-

4. Technology Stack

- **Programming Language:** Python
 - **Framework:** Gradio
 - **NLP Model:** Sentence Transformers for semantic similarity
 - **Deployment:** Local system or Google Colab
-

5. System Workflow

1. **Start Session** – Candidate initiates chat.
2. **Collect Candidate Details:**
 - Full Name
 - Email (validated with regex)
 - Contact Number (validated with regex)
3. **Skill Selection** – Candidate chooses **Python** or **Java**.
4. **Dynamic Questions:**
 - Based on the chosen skill.
 - Includes follow-up on **specific libraries**.
5. **Answer Evaluation:**
 - Compute semantic similarity with expected keywords.
 - Assign **score out of 20**.

6. Final Result:

- Display total score and brief evaluation summary.
-

6. Scoring System

- Each question is scored based on **semantic similarity** using the `SentenceTransformer` model.
 - Scores normalized to **0–20**:
 - **Excellent (18–20)** – Highly relevant, detailed answer.
 - **Good (14–17)** – Covers most points.
 - **Average (10–13)** – Basic understanding.
 - **Poor (<10)** – Weak or irrelevant answer.
 - Empty answers or irrelevant responses = **0**.
-

7. Installation Guide

Step 1: Clone Repository

```
Git clone https://github.com/hibasaudha/Talent-Scout_Assistant-  
cd TalentScout
```

Step 2: Install Dependencies

```
pip install gradio sentence-transformers
```

Step 3: Run the App

```
python app.py
```

8. Code Structure

```
TalentScout/
|
├── app.py          # Main Gradio App
├── requirements.txt # Required Python packages
├── README.md        # Project overview
└── docs/            # Documentation folder
```

9. Future Enhancements

- Add **resume upload** feature for parsing candidate CVs.
 - Support **multiple job roles and categories**.
 - Enable **voice-based interactions**.
 - Integrate **database for storing candidate results**.
 - Deploy as a **web application** (Streamlit / Hugging Face Spaces).
-

10. Limitations

- Requires internet for model loading (if not cached).
 - Currently supports **Python and Java only**.
 - Basic keyword-based evaluation for now (semantic similarity only).
-

11. Example Questions

Python Candidate:

- What is Python used for?
- Can you explain OOP in Python?
- Which libraries are you most familiar with?

Java Candidate:

- Explain JVM, JDK, and JRE.
- What are Java Collections?
- Which frameworks do you commonly use?