

Talent Matcher - Documentation

AI-Powered Employee Matching with LangChain + Streamlit

Version: 1.0 | Date: November 06, 2025

Author: Riju Ray

GitHub:

Overview

Talent Matcher is a **real-time AI-powered internal staffing tool** that matches project manager (PM) requests to the best available employees using **natural language processing** and **LLM-based ranking**.

- **Input:** Free-text PM request (e.g., "Need a senior Python dev with 5+ years in Fintech")
- **Output:** Top 5 ranked employees with **fit score (0-100)** and **2-sentence justification**

Built with:

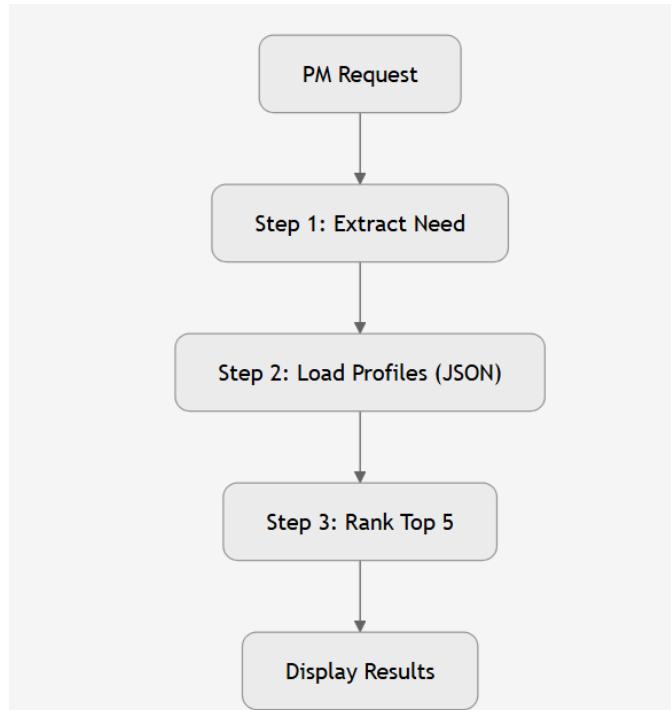
- **Streamlit** – Interactive UI
 - **LangChain (v0.2+)** – Modern chain-based LLM orchestration
 - **OpenRouter (Mistral 7B)** – Cost-effective, high-quality LLM
 - **Pandas** – Data handling
-

Features

Feature	Description
Natural Language Input	No forms — just type like you'd email a recruiter
3-Step AI Pipeline	1. Extract need → 2. Load profiles → 3. Rank top 5
Robust JSON Parsing	Handles model noise (<code><ss></code> , <code>[OUT]</code> , etc.)
Live Feedback	Spinners + extracted need + raw JSON debug

Feature	Description
Configurable	API key, CSV upload, model selection
Fast Prototyping	Limited to first 15 profiles (configurable)

How it Works (Pipeline)



Step 1: Extract Need

- **Prompt:** Summarize request into 3 bullets
- Output Example:
 - Role & seniority: Staff-level Python developer
 - Skills + years: Python (3+ years), Django preferred
 - Domain: Fintech

Step 2: Load Profiles

- Reads employee_skills_dataset.csv

- Limits to MAX_PROFILES = 15 for speed
- Returns **structured JSON** of candidate data

Step 3: Rank top 5

- **LLM Task:** Return **only** valid JSON array of top 5
 - **Scoring:** 0–100 based on role, skills, years, domain, availability
 - **Justification:** 2 clear sentences
-

Project Structure

```

talent-matcher/
|
├── app.py          # Main Streamlit app (single file)
├── employee_skills_dataset.csv # Required data
├── requirements.txt    # Python dependencies
├── README.md        # This file
└── .streamlit/
    └── config.toml   # Optional UI theme

```

CSV Data Format

Column	Type	Description
ID	int	Unique employee ID
Name	str	Full name
Role	str	Current role (e.g., "Data Scientist")
Seniority	str	"Junior", "Mid", "Senior", "Lead"
Skills	str	Comma-separated (e.g., "Python, SQL, AWS")
Years_Experience	int	Total years in tech
Last_Project_Domain	str	e.g., "Fintech", "Healthcare"

Column	Type	Description
Availability_Score	int	0–100 (higher = more available)
Bio	str	Short professional summary

Setup & Installation

1. Clone or Download

```
git clone https://github.com/yourname/talent-matcher.git  
cd talent-matcher
```

2. Install Dependencies

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

```
requirement.txt
```

```
streamlit
```

```
pandas
```

```
langchain-openai
```

```
openai
```

3. Add your Data

Place employee_skills_dataset.csv in the root folder or upload it via the app

4. Get API Key

- Go to <https://openrouter.ai/keys>
- Create a free API key
- Paste it in the Pycharm in the .env file as OPENROUTER_API_KEY = "your Key"

Running the App

```
streamlit run file_name.py
```

Default Test Query

Staff with Python skills in Fintech domain, at least 3 years experience

Configuration Options

Setting	Default
MODEL	"mistralai/mistral-7b-instruct"
MAX_PROFILES	15
temperature	0.1 (deterministic)
CSV Upload	Optional

Sample Output

Extracted Need

- Role & seniority: Staff-level Python developer
- Skills + years: Python (3+ years), Django preferred
- Domain: Fintech

Top 5 Matches

1. Alice | 92/100
Alice is a senior full-stack developer with strong Python skills...
2. Oscar | 90/100
Oscar is a senior backend developer with exceptional Python...
...