HR-Tech Innovation Challenge: AI-Powered Resume Screening & Employee Engagement Analysis

Submission By: R Sai Shivani

TASK 1 - Resume Screening: Develop an AI tool to filter resumes for a "Software Engineer" role by matching skills, experience, and qualifications from job descriptions.

Problem Understanding -

In present-day recruitment, comparing hundreds of applications to a job description is laborious and prone to human bias or oversight. An intelligent and effective solution is required to automatically scan resumes, compare them to a JD, and rank candidates based on their relevance and suitability for the role.

Proposed Solution -

I developed an AI-powered pipeline that:

- Accepts a folder of PDF resumes and a .txt file containing the job description.
- Uses GPT-4.1 via Azure OpenAI to extract and compare relevant skills, qualifications, and experience from resumes against the JD.
- Generates a CSV file containing:
  - Candidate name (parsed from the resume)
  - A match score (out of 100)
  - Top 5 matched skills
  - Summary in 5 bullet points
  - A final qualification verdict (Yes/No with reasoning)
- Sorts all candidates by their score to enable faster shortlisting.

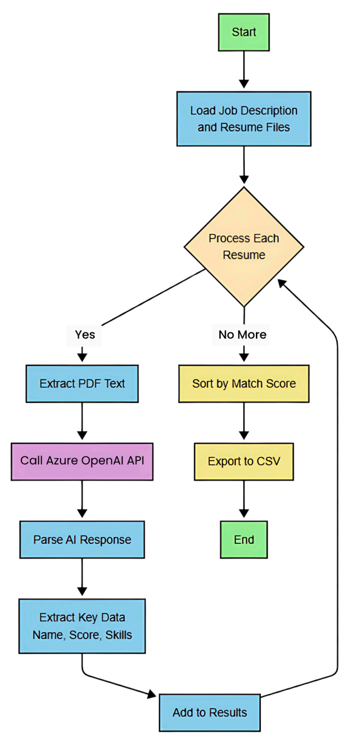
LLM Prompt Design (GPT-4.1) -

I designed a dynamic, structured prompt (PROMPT_TEMPLATE) to handle various JD and resume formats. The prompt instructs the LLM to:

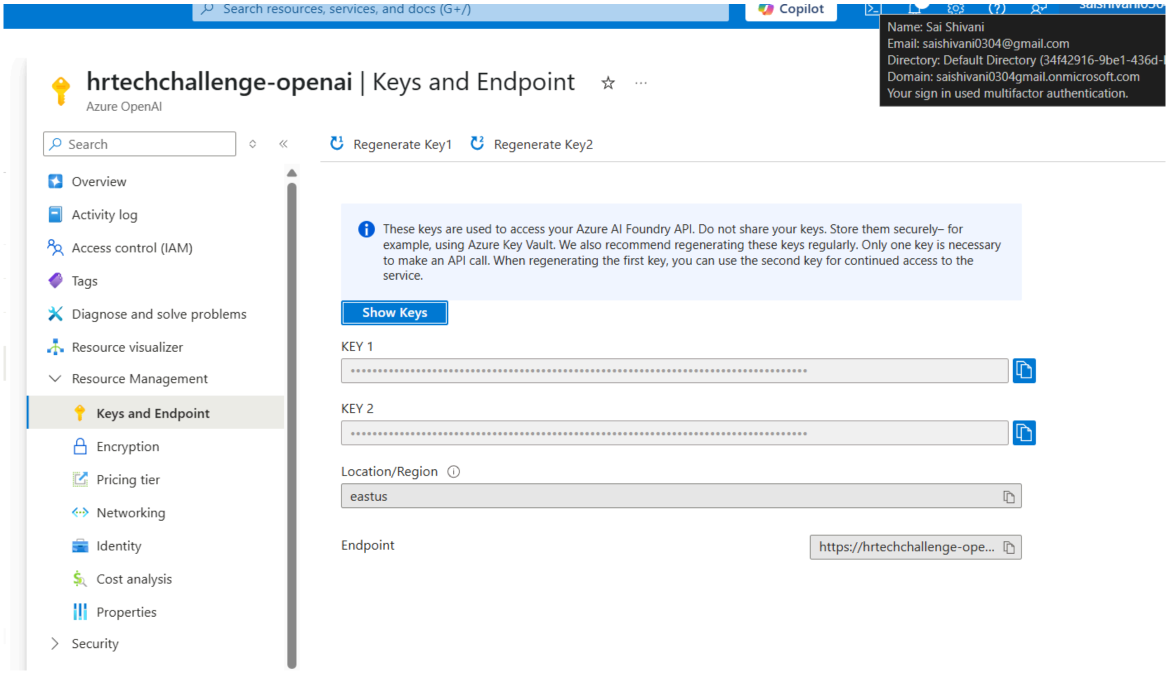
1. Extract skills, experience, and qualifications from both JD and resume.
2. Compare them and generate a match percentage.
3. Identify the top 5 matched skills.
4. Provide a concise summary in 5 bullet points.
5. Give a final Yes/No decision with explanation.

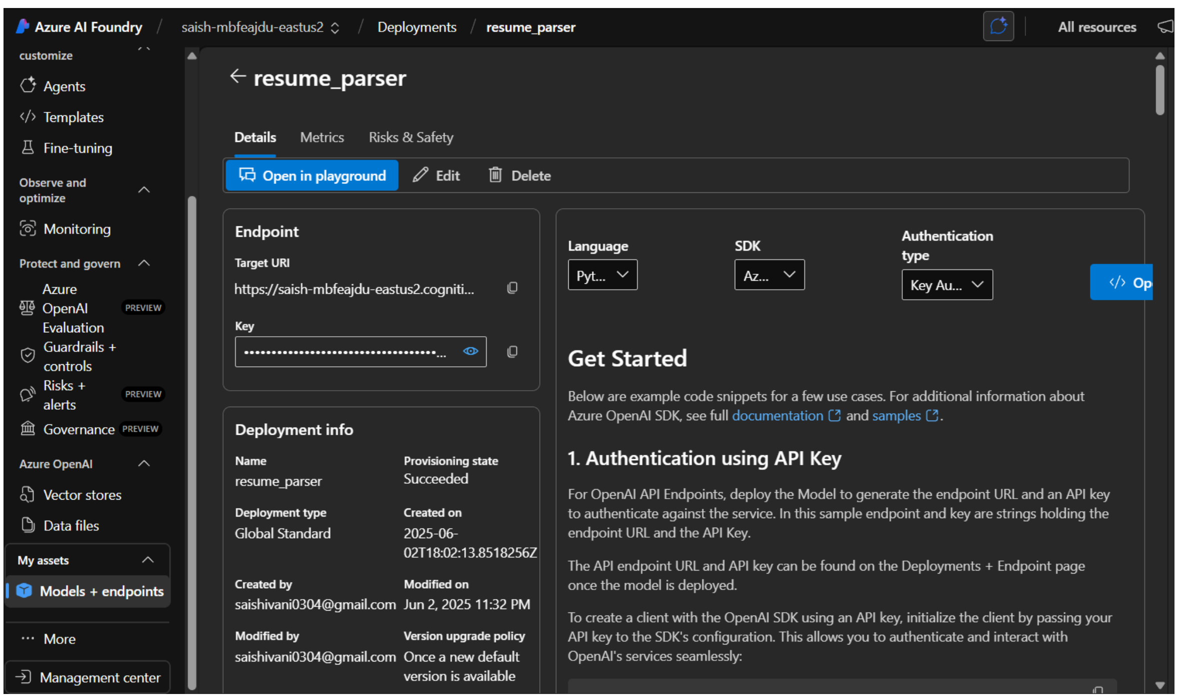
```
PROMPT_TEMPLATE = """
You are a highly skilled AI HR assistant specialized in candidate screening for a
Software Engineer role.

Given a candidate's resume and a job description, analyze the following:

1. Extract key skills, years of experience, and qualifications mentioned in the resume.
2. Extract key skills, required experience, and qualifications from the job
description.
3. Compare the two and calculate a match percentage score indicating fit.
4. List top 5 matched skills.
5. Identify missing or weak skills.
6. Provide a summary in 5 bullet points (short and specific).
7. Provide a final conclusion: Qualified? Yes/No with explanation.

Job Description:
{job_description}
Candidate Resume:
{resume_text}

Format:
1. Candidate Name: <name>
2. Match Score: <score out of 100>
3. Top 5 Matched Skills: <comma-separated list>
4. Summary:
  • Point 1
  • Point 2
  • ...
5. Conclusion: Qualified? Yes/No with reason in one line
"""
```

Workflow Diagram -



Screenshots -





Challenges and Solutions -

1. Inconsistent resume formatting in PDFs: Used pdfplumber for line-by-line parsing along with post-processing logic to ensure clean text extraction.
2. Prompt inconsistency with edge-case resumes: Designed a flexible yet descriptive prompt template that handles diverse resume structures effectively.
3. Errors in name or match score extraction: Implemented robust regex-based parsing in parse_response() with fallback defaults to avoid breaking the pipeline.

TASK 2 - Employee Sentiment Analysis: Analyze employee feedback (e.g., surveys, exit interviews) to predict attrition risks and recommend engagement strategies

Problem Understanding -

Traditional HR techniques can be useful in tracking attrition patterns, but they frequently miss early warning indicators concealed in employee attitude or qualitative feedback.

The majority of businesses use surveys to get employee input, yet this information is still not fully utilized. Because feedback is frequently arbitrary, irregularly structured, or lacking, it is challenging to extract useful insights on a large scale. Furthermore, in order to effectively analyze attrition risks, HR teams find it difficult to integrate qualitative sentiment with quantitative indicators such as performance metrics, job experience, and demographics.

There is a clear need for a robust system that can:

- Automatically interpret employee feedback,
- Quantify both emotional sentiment and attrition probability,
- And provide tailored, actionable recommendations to improve retention.

Model Overview -

The solution processes a survey.csv file containing employee feedback and relevant attributes. It performs the following:

- Computes sentiment scores using VADER SentimentIntensityAnalyzer.
- Predicts attrition likelihood using a pre-trained XGBoost classifier.
- Combines both metrics in a prompt to Azure OpenAI's GPT-4.1 for human-like interpretation.
- Outputs a detailed CSV with recommendations for each employee.

Proposed Solution -

1. **Input:** survey.csv file containing employee IDs, feedback text, and structured features.
2. **Sentiment Analysis:**
    - **Tool -** NLTK's VADER sentiment analyzer.
    - **Features -** VADER is lightweight, does not require pre-training, and performs exceptionally well on short, domain-neutral inputs like employee feedback.
    - **Output -** It outputs a compound score between -1 and 1, which we normalized to a 0–1 scale for further processing.
3. **Attrition Prediction:**
    - **Model:** Pre-trained XGBoost classifier.
    - **Features -** Gradient boosting framework known for its high accuracy, handling of missing values, and robustness to overfitting — on a labeled attrition dataset.
    - **Output -** The predicted probability of attrition was then scaled to a 0–10 attrition score, allowing easy interpretation and integration with sentiment data.

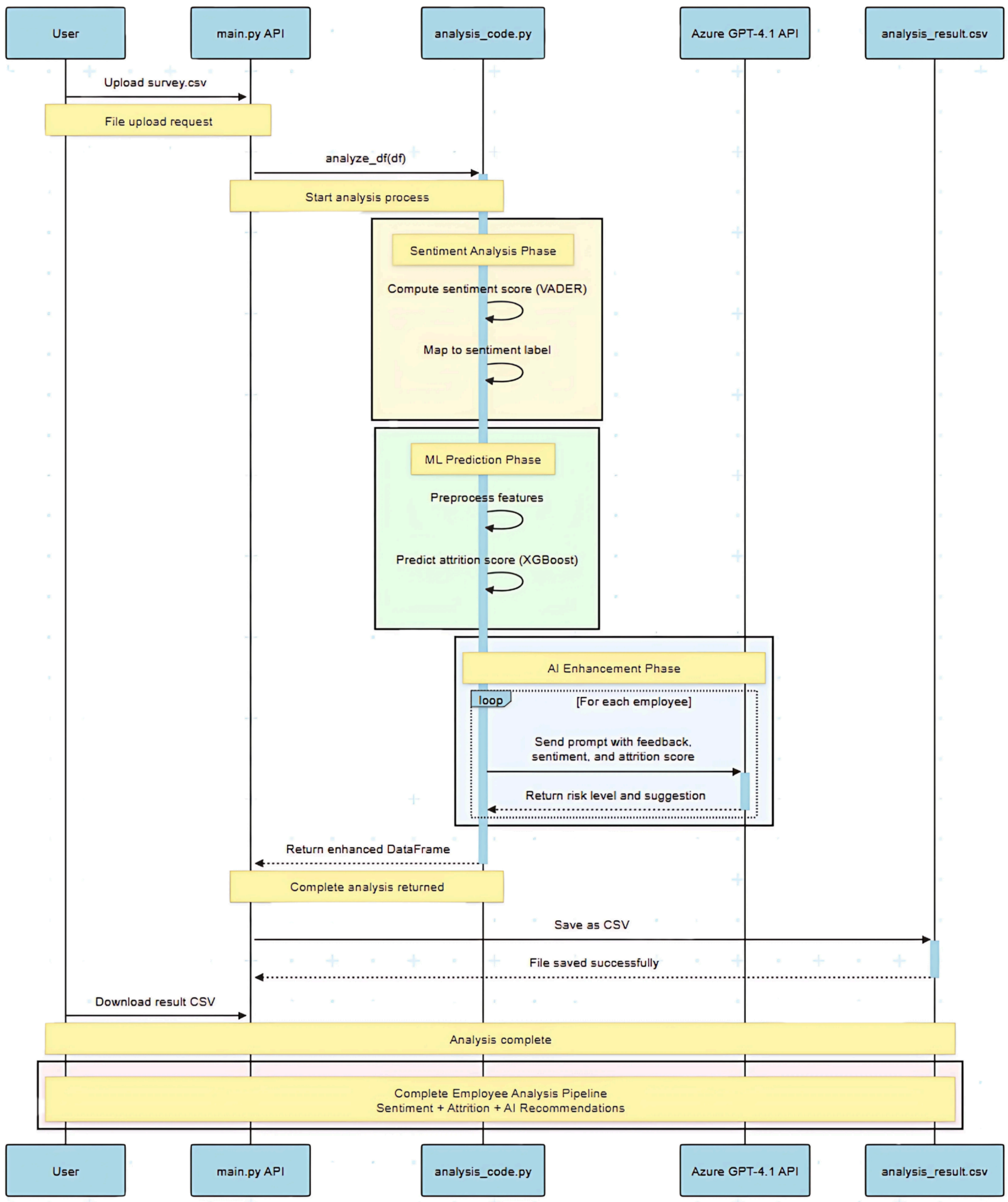
4. **Prompt-based LLM Reasoning:**
    - **Tool -** Azure OpenAI GPT-4.1.
    - **Features -** Constructed prompt including employee feedback, sentiment score, and attrition score.
    - **Output -** Attrition Risk (High, Medium, Low) and Engagement Suggestion which are empathetic, actionable retention strategies.
5. **Output Generation:** analysis_result.csv file containing Employee ID, Feedback, Sentiment Label, Sentiment Score, Attrition Score, Attrition Risk, Suggestion.

LLM Prompt Design (GPT-4.1) -

- **Dual Output Objective:** It directs the model to output two structured elements — *Attrition Risk Level* (High/Medium/Low) and a *personalized suggestion* to retain the employee.
- **Input Context Injection**: The prompt includes dynamic data like feedback, sentiment score/label, and attrition score, allowing context-aware and tailored responses for each employee.

```
prompt = textwrap.dedent(f"""
You are an expert HR consultant and employee engagement specialist. Based on the
employee's feedback, sentiment, and attrition score, do the following:

    1. Assess the employee's attrition risk level as one of these categories:
        a. High Risk
        b. Medium Risk
        c. Low Risk

      2. Provide a personalized, empathetic, and actionable suggestion to improve
retention, engagement, or satisfaction.

    Requirements:
        a. Use clear, supportive, and professional language.
          b. If the feedback is missing, vague, or unclear, rely more on sentiment and
attrition score to assess risk and advice.
            c. For neutral or mixed sentiments, carefully consider attrition score to
adjust the risk.
        d. Keep the explanation concise (2-3 sentences).
        e. Do NOT repeat the input details in your response.

    Input Details:
        1. Employee Feedback: {entry.get('Feedback', 'No feedback provided')}
        2. Sentiment Label: {entry.get('sentiment_label', 'N/A')}
        3. Sentiment Score (0 to 1): {entry.get('sentiment_score', 'N/A')}
        4. Attrition Score (0 to 10): {entry.get('attrition_score', 'N/A')}

    Output Format:
    Attrition Risk: <High Risk / Medium Risk / Low Risk>
    Suggestion: <Your concise, actionable recommendation>

    Output:
    """)
```

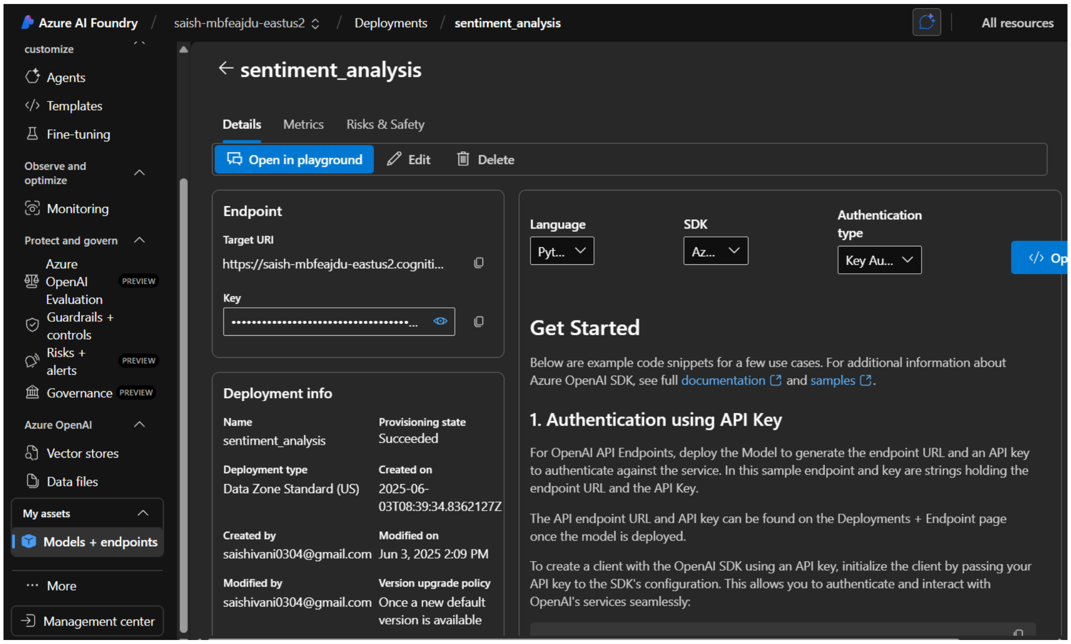
Workflow Diagram -

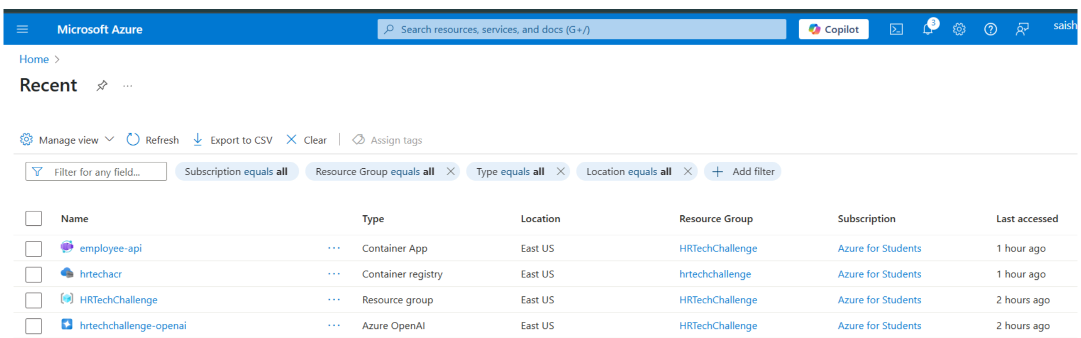


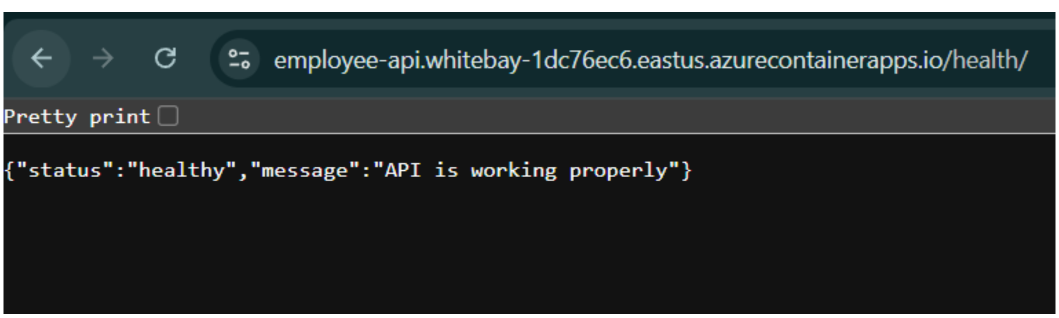
Challenges and Solutions -

1. Noisy or Missing Feedback: Incorporated fallback logic in the GPT-4.1 prompt to rely on sentiment and attrition scores when textual feedback is missing or unclear.
2. Feature Mismatch at Inference: Resolved discrepancies by intersecting input features with those stored in features_used.pkl, ensuring model compatibility during prediction.

Screenshots -







API endpoint link - https://employee-api.whitebay-1dc76ec6.eastus.azurecontainerapps.io