This is a critical pivot point for Group 6. As an expert strategist and Python developer, I have analyzed your constraints, the competitive landscape of the other groups, and the limitations of your dataset.

Here is the hard truth: **Your current idea is not feasible.**

Your current objective is *"matching users to roles that best fit their skills"*. Without job descriptions, you cannot extract the necessary keywords (e.g., "Python," "Stakeholder Management," "Agile") to perform skills matching. Relying solely on Job Titles for skills matching is notoriously inaccurate and will result in a low-quality product.

Furthermore, Group 9 is already building a "Skills Gap Analysis & Personalized Career Path Engine", and Group 6 would struggle to compete with them given your data handicap.

### The Strategic Pivot: "Alternative Data" for Market Intelligence

Given that you have **~1 million rows** but no descriptions, your strength lies in the **volume and metadata** (Company, Title, Salary, Date). You should treat job postings not as "ads for candidates," but as **signals of corporate strategy.**

I recommend pivoting to a **Corporate Strategy & "Signal" Dashboard**.

In the finance and strategy world, job postings are considered "Alternative Data." If a bank suddenly posts 50 jobs for "AI Engineers," they are signaling a digital transformation before they announce it in their earnings call.

#### Proposal: "The Corporate Signal Dashboard" (Market Intelligence)

* **The Business Problem:** Investors, B2B Sales Teams, and Competitive Analysts need to know which companies are expanding, pivoting, or struggling *before* it hits the news.
* **Target User:**
  + **Private Equity / Investors:** Looking for growing companies (hiring spikes).
  + **B2B Sales Leads:** Looking for companies with budget (high salary spend).
  + **Competitor Analysts:** "Why is Competitor X hiring 20 Sales staff in Changi?"
* **The Solution:** A Streamlit dashboard that visualizes "Hiring Momentum" and "Strategic Focus" by aggregating Job Titles and Salaries.

### Why this wins (The Gap Analysis)

* **Group 7** is doing *Industry* level analysis. You will go deeper to the *Company* level.
* **Group 2** is doing *Salary* benchmarking. You are using salary only as a proxy for "Budget/Spend," not for HR benchmarking.
* **Group 1** is doing *Government* policy gaps. You are focused on *Private Sector* competition.

### Technical Implementation Plan (Python & Streamlit)

Since we cannot parse text for skills, we will use **Regex-based Categorization** on Job Title to create "Roles."

#### 1. Data Processing Strategy (The "Title Cleaner")

You need to map messy titles to standardized "Departments" to see where companies are spending money.

* **Logic:** Create a Python function to tag titles based on keywords.
  + *If "Sales", "Business Development", "Account" in Title -> Tag: "Revenue Generating"*
  + *If "Engineer", "Developer", "Data" in Title -> Tag: "R&D / Tech"*
  + *If "HR", "Admin", "Finance" in Title -> Tag: "Back Office"*
* **Metric:** Calculate "Share of Voice." If 60% of Company X's posts are "R&D," they are building product. If 60% are "Sales," they are pushing for revenue.

#### 2. Key Streamlit Views

* **View 1: The "Mover & Shaker" Radar:**
  + *Chart:* Scatter plot. X-axis = Total Postings (Volume), Y-axis = Average Salary (Quality).
  + *Insight:* High Volume + High Salary = Aggressive Growth (High Value Target). High Volume + Low Salary = High Churn / Low Quality.
* **View 2: Strategy Decoder (Company Drill-down):**
  + *Input:* Select 2 Companies (e.g., DBS vs. UOB).
  + *Chart:* Stacked Bar Chart of "Hiring Category" (Tech vs. Sales vs. Ops) over time.
  + *Insight:* "DBS is hiring 2x more Tech roles than UOB this quarter."
* **View 3: The "Budget Estimator":**
  + *Logic:* $\text{Estimated Spend} = \sum (\text{Min Salary} \times \text{Posting Count})$.
  + *Insight:* Which company has the deepest pockets right now?

### Data Dictionary Assumption & Code Snippet

Assuming your CSV has job\_title, company\_name, salary\_max, and date.

Python

import pandas as pd  
import streamlit as st  
  
# Example Function to Categorize Strategy based on Title only  
def categorize\_role(title):  
 title = str(title).lower()  
 if any(x in title for x in ['sales', 'business dev', 'account', 'marketing']):  
 return 'Growth/Revenue'  
 elif any(x in title for x in ['engineer', 'developer', 'data', 'tech', 'software']):  
 return 'R&D/Tech'  
 elif any(x in title for x in ['manager', 'director', 'lead', 'vp', 'head']):  
 return 'Management'  
 else:  
 return 'Operations/Support'  
  
# Load Data (Simulated)  
df = pd.read\_csv('your\_data.csv')  
  
# Apply Strategy Tag  
df['Strategy\_Tag'] = df['job\_title'].apply(categorize\_role)  
  
# Insight: Calculate "Strategic Focus" by Company  
company\_focus = df.groupby(['company\_name', 'Strategy\_Tag']).size().unstack().fillna(0)  
company\_focus['Total'] = company\_focus.sum(axis=1)  
  
# Sort by biggest hirers  
top\_companies = company\_focus.sort\_values('Total', ascending=False).head(10)  
  
st.title("Corporate Strategy Decoder")  
st.bar\_chart(top\_companies.drop(columns=['Total']))  
st.write("This chart reveals what departments top companies are aggressively funding.")

### Summary of the New Direction

1. **Project Name:** "Talent Radar: Corporate Strategy Decoder"
2. **User:** Investors and Competitor Intelligence Analysts.
3. **Value:** Inferring corporate strategy and budget allocation from hiring metadata.
4. **Feasibility:** **High.** Completely bypasses the need for job descriptions.

**Would you like me to refine the "Strategy Tagging" logic for specific industries (e.g., Tech vs. Finance) to make the categorization more robust?**