DATA SLUSH ASSIGNMENT

Overview

DataSlush has engaged with a company called **Social Directory** (This is a fictitious company). Social Directory aims to streamline and diversify the hiring process for the creative industry, starting with social media creators as their core ICP (Ideal Customer Profile).

Customer Problem:

Creators receive thousands of applications from candidates with diverse backgrounds when they post a job. We are seeking a solution which ranks talent based on their creative fit, recommending the best talent to them.

Key Problems Addressed:

- Varying criteria among creators
- Specific needs for different production (video editing, design etc) and non-production (strategy, business development, operations etc) roles
- Less dependency on text-based matching
- Factoring in talent's personality and soft skills

Assignment Objective

- In this assignment, your core objective is to develop a recommendation system to score and recommend a list of Top 10 Candidates for the hiring creators from the provided list above.
-
- We are looking for your thought process to build a viable solution to serve real customers and not just a take home assignment.

Features

- Backend (Flask + Pandas):
  - Exposes REST API endpoints to fetch jobs and candidate recommendations.
  - Reads job/candidate data from CSV files.
  - Provides clean JSON responses for the frontend.

- Frontend (React + Vite):

  - Simple and responsive **Job Dashboard UI**.
  - Buttons to select different job roles.
  - Dynamic **charts** and **tables** displaying the top 10 candidates per job.
  - Dark theme UI for better visualization.

Tech Stack

Backend:

- Python
- Flask
- Flask-CORS

- Pandas

Frontend:
- React (Vite)
- JavaScript (ES6+)
- Chart.js / Recharts (for visualizations)

Project Structure

```
DATA_SLUSH_ASSIGNMENT/
|— backend_ds/
|  ├— app.py # Flask backend API
|  ├— recommender.py # Candidate recommender logic
|  ├— utils.py # Utility functions
|  ├— requirements.txt # Python dependencies
|
|— data_ds/
|  ├— jobs.json
|  ├— talent_samples.csv
|  ├— all_jobs_top10.csv # Top 10 candidates per job
|
|— job-dashboard/ # React frontend
|  ├— src/
|  |  ├— api.js # API calls to Flask backend
|  |  ├— App.jsx # Main React App
|  |  ├— pages/
|  |  |  ├— HomePage.jsx # Job Dashboard
|  |  |  ├— JobPage.jsx # Candidate details + chart
|  |  ├— components/
|  |  |  ├— CandidateChart.jsx
|  |  |  ├— CandidateTable.jsx
|  ├— public/
|  ├— package.json
```

Backend Setup (Flask)

```
cd ../backend_ds
pip install -r requirements.txt
py app.py
```

Frontend Setup (React + Vite)

```
cd ../job-dashboard
npm install
npm run dev
```