Documentation: AI-Based Job & Internship Matching Engine

Project Overview

The objective of this project is to design and develop an **AI-powered system** that automatically collects job and internship listings from multiple sources and intelligently matches them with student profiles.
The system should notify students when a **highly relevant (perfect-fit)** job or internship becomes available.

What it does

- Pulls jobs from:
- Company portals
- LinkedIn
- Intern partners
- Matches students based on:
- Skills
- Location
- Salary expectations
- Company preference

Auto-alerts when a perfect-fit job appears.

Problem Statement

Students often miss suitable job or internship opportunities due to:

- Information scattered across multiple platforms
- Lack of personalized job recommendations
- No real-time alerts for relevant openings

This project aims to solve these problems using **data aggregation, filtering, and AI-based matching techniques**.

Key Features

1. Job & Internship Data Collection

The system should pull job/internship listings from:

- Company career portals
- LinkedIn job listings
- Partnered internship providers

(Initial phase can use sample datasets or APIs; real-time scraping is optional based on feasibility.)

2. Student Profile Management

Each student profile should contain:

- Skills (technical & non-technical)
- Preferred job location(s)
- Expected salary / stipend range
- Preferred companies or industries

3. AI-Based Matching Engine

The system should match students with jobs based on:

- Skill compatibility
- Location preference
- Salary/stipend expectations
- Company or industry preference

Matching should return a **match score or ranking** for each job.

4. Auto Alert System

- Automatically notify students when a **high match-score job/internship** is found
- Alerts can be simulated via:
    - Email (preferred)
    - Dashboard notifications
    - Console output (minimum requirement)

Technology Stack (Suggested)

(Intern may modify with approval)

- **Programming Language:** Python
- **Backend:** Flask / FastAPI
- **Database:** MySQL / MongoDB / SQLite
- **AI / ML:**
    - Scikit-learn (cosine similarity / basic ML models)
    - NLP for skill matching (TF-IDF / embeddings – optional)
- **Frontend (Optional):** HTML, CSS, Bootstrap
- **APIs / Data:** Sample job datasets / LinkedIn API (if accessible)