Mini Project Report Job Portal with Smart Resume–Job Matching

Problem Statement:
Traditional job portals only rely on manual keyword searches to connect candidates with job openings. This often leads to irrelevant job recommendations and a poor hiring experience. There is a need for a smarter system that automatically analyzes resumes and matches candidates with the most relevant jobs using Artificial Intelligence (AI) and Machine Learning (ML) techniques.

Motivation:
Finding the right job or candidate is time-consuming for both applicants and recruiters. By leveraging AI, we can make this process faster, fairer, and more accurate. A smart resume–job matching system can save time by automating shortlisting, increase accuracy by matching skills and job descriptions, and improve user experience on the platform.

Objectives:
1. Design and develop a responsive web-based job portal.
2. Enable user registration for both candidates and recruiters.
3. Allow candidates to upload resumes and recruiters to post job openings.
4. Implement an AI-based matching algorithm that extracts skills and compares them.
5. Provide an admin dashboard to manage users, jobs, and matches.

Features:
For Candidates:
- Profile creation and resume upload.
- AI-based job recommendations.
- Apply for jobs directly.

For Recruiters:
- Post job openings.
- View AI-ranked candidate matches.
- Manage job listings.

For Admin:
- Manage users and postings.
- Monitor platform activity.

AI/ML Integration:
- Uses TF-IDF vectorization and cosine similarity for resume-job matching.

Dataset:
Dataset Name: Resume Dataset
Source: Kaggle
Link: https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset
Description: Contains labeled resumes with job categories such as Data Science, HR, Sales, etc. Used for training and testing skill-job similarity models.

Technologies Used:
Frontend: HTML, CSS, JavaScript, React, TailwindCSS
Backend: FastAPI
Database: PostgreSQL
AI/ML: Scikit-learn (TF-IDF, Cosine Similarity)
Libraries: Pandas, NLTK, Joblib

System Workflow:
1. Candidate uploads resume → text extraction and database storage.
2. Recruiter posts job details.

3. ML model computes cosine similarity between resume and job description.
4. Recommendations displayed to users.

Expected Output:
- Fully functional job portal.
- Automatic job recommendations.
- Recruiter dashboards with ranked candidates.

Scope & Future Enhancements:
- Integrate advanced NLP models (e.g., BERT).
- Add email/job alert notifications.
- Include chatbot support.
- Expand with company ratings and interview scheduling.

References:
- Kaggle Resume Dataset: https://www.kaggle.com/datasets/gauravduttakiit/resume-dataset
- Scikit-learn TF-IDF Docs:
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html