

# Intelligent Job Recommendation Engine

## Project Proposal:

### 1. Executive Summary

The goal of this project is to create an intelligent system for recommending jobs that matches job seekers' talents with appropriate positions using natural language processing and machine learning. The technology will use machine learning methods to estimate job compatibility after analyzing resumes and job descriptions and extracting pertinent characteristics. This solves the inefficiency of conventional job matching and assists applicants in locating positions that fit their skills.

### 2. Project Objectives

#### Primary Objectives

- Extract and parse skills from resumes using NLP techniques
- Analyze job descriptions to identify key requirements
- Build and train machine learning models to predict job-candidate fit
- Create a functional prototype with a user interface

#### Secondary Objectives

- Evaluate at least 2-3 machine learning algorithms for performance comparison
- Implement skill matching and compatibility scoring
- Assess system accuracy through appropriate metrics.

### 3. Problem Statement

Information overload, keyword mismatches, and laborious manual searches make it difficult for job seekers to identify employment that match their talents. By using resume analysis to generate intelligent, data-driven job recommendations, an automated recommendation system can enhance this procedure.

### 4. Proposed Solution

#### System Architecture

The system will consist of three main components:

**Data Processing Module:** Accepts resume and job description inputs, extracts text, and performs cleaning and preprocessing.

**Feature Extraction Module:** Uses NLP to identify skills, experience level, job titles, and keywords from both resumes and job descriptions.

**Recommendation Engine:** Applies machine learning algorithms to calculate compatibility scores and rank job recommendations.

#### Machine Learning Approach

- **Algorithms:** Logistic Regression, Random Forest, or Neural Networks
- **Features:** TF-IDF vectorization, skill extraction, keyword matching
- **Evaluation Metrics:** Accuracy, Precision, Recall, F1-Score

## 5. Methodology (4 Phases)

### Phase 1 - Report Due: 31st October 2025

- Find and download datasets from Kaggle (resumes, job postings)
- Build a resume parser that can actually read and extract useful information
- Clean up the data so it's ready to work with
- Test it on a few sample resumes
- **Deliverable:** Report 1 - Data Collection & Resume Parser Implementation

### Phase 2 - Report Due: 15th November 2025

- Create training data with examples of good and bad job matches
- Build and train 2-3 different machine learning models
- See which one performs best
- Write down the results (accuracy, precision, all that good stuff)
- **Deliverable:** Report 2 - AI Model Development & Performance Analysis

### Phase 3 - Report Due: 30th November 2025

- Connect the resume parser to our best AI model
- Build a simple interface where people can upload resumes
- Make it show the top job matches with scores
- Make sure it actually works and is reasonably fast
- **Deliverable:** Report 3 - System Integration & User Interface

### Phase 4 - Report Due: 15th December 2025

- Test everything thoroughly
- Fix any bugs that come up
- Write up the documentation
- Get ready to show it off
- **Deliverable:** Report 4 - Testing, Optimization & Documentation

Final Submission: 12th January 2026

- Complete working system with all reports
- Full source code documentation
- Final presentation and demo
- **Deliverable:** Final Project Submission

## 6. Expected Outcomes

- A working prototype for job recommendations that analyzes resumes and finds matches
- ML model that has been trained and has reported accuracy metrics (target:  $\geq 75\%$ )
- Basic web interface showcasing essential features
- Analysis and comparison of several machine learning methods
- All project documentation

## 7. Technologies & Tools

Programming Language: Python

### **Libraries:**

- NLP: spaCy, NLTK, scikit-learn
- ML: scikit-learn, TensorFlow/Keras
- Data: pandas, NumPy
- Web: Streamlit or Flask
- Document Processing: PyPDF2 or python-docx

**Datasets:** Kaggle resume datasets, public job posting datasets

## 8. Success Criteria

- Over 85% of relevant abilities are successfully extracted by the resume parser.
- On the test dataset, the ML model achieves  $\geq 75\%$  accuracy.
- Suggestions produced in less than three seconds
- A useful user interface that displays matches and accepts resumes
- Full documentation and a functional example

## 9. Challenges & Mitigation

Challenge	Mitigation
Limited time for model development	Use pre-trained NLP models; focus on simpler algorithms first
Data availability	Utilize existing public datasets from Kaggle
Handling skill variations	Implement basic skill normalization and synonym matching
Model tuning	Use grid search with limited parameter ranges
Complex UI	Use Streamlit for quick, simple interface development

## 10. What's Due When

Phase	Report Due	Deliverable
1	31st Oct 2025	Data Collection & Resume Parser
2	15th Nov2025	AI Model Development & Analysis
3	30th Nov 2025	System Integration & UI
4	15th Dec 2025	Testing & Documentation
<b>Final</b>	<b>12th Jan 2026</b>	<b>Complete Project Submission</b>

### Final Deliverables:

- Working prototype with functional UI
- Source code (documented and organized)
- Project report with methodology and results
- Model performance analysis
- Live demonstration/presentation

## 11. Resources

- Development environment.
- Local machine or university computing resources.
- Open-source libraries.
- Publicly available datasets

## 12. Conclusion

This project successfully applies machine learning and natural language processing to produce a useful, functional job recommendation system. The timeline's focus on essential features enables quick development and a clear illustration of the system's usefulness in resolving actual job-matching issues.