**Tribhuvan University**

**Institute of Science and Technology**

**Bhaktapur Multiple Campus**

****

# BIT

**4th Year / 7th Semester**

PROJECT PROPOSAL ON

**Matchfolio - AI-Based Resume Analyzer**

**Submitted to**

Department of IT

**Bhaktapur Multiple Campus**

**Submitted by**

Raju Mainali

Ujjwol Mainali

Jenish Adhikari

**1. Introduction**

In today’s fast-paced hiring environment, companies receive an overwhelming number of resumes for every job opening. Manual screening is not only time-consuming but also prone to human biases and inconsistencies. Matchfolio is an AI-driven resume analyser that streamlines this process by intelligently matching resumes with job descriptions. It leverages natural language processing (NLP) and machine learning (ML) to assess candidate profiles against job requirements, thus enabling recruiters to make data-driven hiring decisions efficiently.

This project addresses the challenge of high-volume recruitment through automation and precision. Inspired by analytical modeling techniques used in structural engineering for complex simulations, we seek to create a computational model capable of producing high-confidence predictions about candidate-job fit, improving the efficiency and objectivity of recruitment pipelines.

**2. Problem Statement**

In today’s fast-paced hiring environment, recruiters and HR professionals are overwhelmed by the sheer volume of job applications they receive for every open position. A single job posting can attract hundreds—if not thousands—of resumes, and each of these documents needs to be reviewed, parsed, and evaluated. This creates a bottleneck in the recruitment process.

**Manually reviewing resumes is not only time-consuming but also prone to human error and bias.** Recruiters often spend countless hours scanning through resumes looking for keywords, skills, and relevant experiences that match a job description. In reality, they only have a few minutes—or even seconds—to decide whether a candidate should move to the next stage. As a result, qualified candidates can be easily overlooked, while less suitable ones might get through due to formatting tricks or keyword stuffing.

Furthermore, **traditional Applicant Tracking Systems (ATS)** primarily rely on keyword matching and rigid filters. These systems often fail to understand context or variations in how skills and experiences are expressed. For instance, a candidate with excellent data analysis experience might be missed if the job description uses the term “data interpretation” and the ATS fails to associate the two.

From the candidate’s perspective, it’s equally frustrating. Many applicants never hear back or get rejected by automated systems without understanding why. They’re often left in the dark, not knowing whether their resume was even seen by a human being.

This leads to three key challenges in modern recruitment:

1. **Inefficiency** – Too much time is spent on repetitive, low-value screening tasks.
2. **Inaccuracy** – Good candidates are missed, and mismatches occur frequently.
3. **Lack of Transparency** – Neither recruiters nor applicants get clear insights into why decisions are made.

**This is the problem we aim to solve.** By building an AI-powered resume analyzer, we want to **bring intelligence and fairness into the recruitment process**. Our solution will help recruiters automatically parse, evaluate, and rank resumes based on contextual matching with job descriptions using machine learning and natural language processing. It will save time, reduce bias, and ensure that the best candidates get noticed—not just the ones who know how to “game” the system.

In a world where the right job can change someone’s life—and the right hire can elevate a company—our goal is to make those matches smarter, faster, and fairer.

**3. Objectives**

* To develop a web-based application that allows uploading of resumes and job descriptions.
* To use AI techniques for extracting and comparing key resume information with job requirements.
* To provide a match score and analysis report for each candidate.
* To enhance hiring accuracy, speed, and fairness.
* To test and validate the system using synthetic and real-world resume data.
* To document the comparative performance of various similarity algorithms and recommend best-fit models.

1. **Methodology**

The creation of the AI-based Resume Analyzer will be done using the Agile SDLC approach, which enables iterative and incremental software delivery in the form of functional modules and permits ongoing feedback, testing, and refinement.  
**Agile SDLC Phases Used in This Project**

1. **Concept/Initiation Phase:**  
    • Set project vision and objectives  
    • Meetings with stakeholders (developers, HR professionals) for high-level requirements gathering
2. **Requirement Gathering and Analysis:**  
    • In-depth requirement gathering  
    • Creation of user stories and product backlog  
    • Prioritizing based upon importance and dependencies
3. **Design Phase:**  
    • System architecture planning  
    • High-level module design for frontend, backend, and AI engine  
    • Database schema definition and API contract documentation
4. **Iteration Planning (Sprints):**  
    • Identify and define sprint goals and deliverables (2-week sprints)  
    • Task assignments for developers and data scientists
5. **Development Phase:**  
    • Incrementally built backend APIs and frontend elements  
    • ML/NLP models built and integrated into back-end services  
    • Continuous integration and deployment practices employed
6. **Testing Phase:**  
    • Unit Testing, Integration Testing, and System Testing  
    • User acceptance testing (UAT) using real resumes and job descriptions
7. **Deploy Phase:**  
    • Hosting on cloud platforms such as AWS and Heroku  
    • Final deployment into production environment
8. **Maintenance and Feedback Loop:**  
    • Bug fixes and performance tuning

• Gathering user feedback and scheduling future versions for new functionality or enhancements

**Rationale for Agile Model**  
• Ability to adjust according to shifting needs  
• Ongoing supply of working pieces  
• Timely identification of mistakes and conformity to user requirements

The Agile SDLC approach allows for the project development in an adaptive and user-focused way, suited well for emerging, complex AI-based applications.

**a. Requirement Identification**

i. **Study of Existing System / Literature Review**

To initiate the development of the AI-based Resume Analyzer, we conducted a comprehensive review of existing systems and literature. Several commercial and open-source resume parsing and analysis tools such as HireAbility, Sovren, and RChilli were studied to identify the current standards in resume processing. Key limitations in these systems include inadequate semantic understanding, lack of customization to specific job roles, and poor integration with modern tech stacks. Academic papers focusing on Natural Language Processing (NLP), Machine Learning (ML) in recruitment, and semantic text similarity were also reviewed to guide the architecture and algorithmic choices of the proposed system.

ii. **Requirement Analysis**

Functional Requirements:

* Upload resume in various formats (PDF, DOCX, TXT).
* Extract and parse relevant fields (name, contact, education, experience, skills).
* Match candidate resume with job descriptions.
* Score and rank resumes based on AI algorithms.
* Display results on a dashboard.

Non-Functional Requirements:

* High performance and scalability.
* Secure data handling and privacy.
* User-friendly UI and UX.
* RESTful API integration for extensibility.

**b. Feasibility Study**

i. **Technical Feasibility**

The system will use Python for AI and ML models, Node.js with Express for the backend API, and React.js for the frontend interface. Python’s robust libraries (NLTK, spaCy, scikit-learn, transformers) offer powerful tools for NLP, while Node.js provides scalable server-side functionality. React ensures a dynamic and responsive user interface.

ii. **Operational Feasibility**

The system is designed to be user-friendly for both HR professionals and developers. Its modular structure supports easy maintenance and future upgrades. Integration with existing HRMS systems through APIs will allow smooth operational deployment in real-world scenarios.

iii. **Economic Feasibility**

The initial cost involves development time, cloud resources for hosting and ML training, and minimal licensing for development tools. In comparison to commercial solutions, this custom system offers better return on investment through tailored functionality, open-source technology, and extensibility.

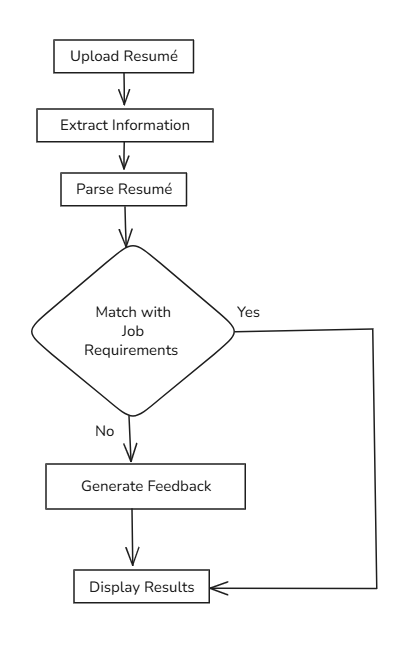
iv. **Schedule (Gantt Chart Showing the Project Timeline)**

**c. High-Level Design of System**

**Methodology of the Proposed System:**

1. **Data Input Layer**: Users upload resumes and job descriptions via the frontend interface.
2. **Preprocessing Module**: Text is extracted and cleaned using Python libraries.
3. **Parsing & Extraction Module**: Information is parsed using NLP techniques into structured data.
4. **Matching & Scoring Module**: Resumes are matched to job descriptions using semantic similarity and ML algorithms.
5. **Results Layer**: Ranked candidates and detailed insights are shown on the dashboard.

**Flow Chart:**



**Working Mechanism of Proposed System:**

* Utilizes transformer models (e.g., BERT) to understand context in resumes.
* Cosine similarity or classification algorithms (e.g., Random Forest, Logistic Regression) used for matching.
* REST APIs built with Express handle data flow between frontend and backend.
* React dashboard presents visual insights like match percentage, key strengths, and suggested improvements.

**Description of Algorithms:**

* **Text Preprocessing**: Tokenization, stop-word removal, stemming/lemmatization.
* **Entity Recognition**: spaCy models for extracting entities like skills, education, etc.
* **Similarity Matching**: Sentence-BERT embeddings + Cosine Similarity.
* **Scoring**: Weighted algorithm considering skill match, experience, and keyword density.

This high-level design ensures the system is scalable, intelligent, and effective in aiding the recruitment process using modern AI capabilities

.**5. Expected Outcome**

* A working prototype of Matchfolio capable of analysing and scoring resumes
* Enhanced recruiter productivity and decision-making
* Reduced bias and improved fairness in hiring
* Scalable system ready for integration with enterprise platforms
* An empirical report comparing parsing precision and matching accuracy of different models (e.g., TF-IDF, BERT, USE)

**6. References**

* Jurafsky, D., & Martin, J. H. (2021). *Speech and Language Processing*
* Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*
* GitHub repositories on resume parsers and NLP models
* Research articles from IEEE Xplore and Springer on AI in recruitment
* Official documentation of BERT, spaCy, and scikit-learn
* Abbas, A., et al. (2019). Behavior of reinforced concrete sandwiched panels under blast load — for methodological inspiration
* Cervenka, V., et al. (2002). ATENA — A tool for engineering analysis of fracture in concrete — relevant to simulation fidelity