# PROJECT AND TEAM INFORMATION

## Project Title

**Ai Based Resume Scanner**

Student / Team Information

|  |  |
| --- | --- |
| *Team Name: Team #* |  |
| **Team member 1 (Team Lead)**  *(Last Name, name: student ID: email, picture):* | *Khatana, Shanu – 240221096*  *Shanukhatana062@gmail.com* |
| **Team member 2**  *(Last Name, name: student ID: email, picture):* | *Sharma, Lavanya –240222224*  *Lavanyasharma614@gmail.com* |
| **Team member 2**  *(Last Name, name: student ID: email, picture):* | *Sethi, Samridhi – 240222067*  [*Ridhi.sethi04@gmail.com*](mailto:Ridhi.sethi04@gmail.com) |

# PROPOSAL DESCRIPTION

## Motivation

The modern hiring process faces major challenges as recruiters struggle with massive volumes of resumes, manual screening inefficiencies, and unconscious human biases that hinder diversity. Traditional Applicant Tracking Systems (ATS) often reduce resumes to keyword filters, overlooking context, synonyms, and true candidate potential—leading to missed opportunities and unfair exclusions. This project proposes an AI-driven resume screening platform that leverages semantic understanding, contextual matching, and bias mitigation. The aim is to augment recruiters’ decision-making, making hiring faster, fairer, and more effective while ensuring diverse, high-potential talent is recognized and not lost to competitors.

## State of the Art / Current solution

The recruitment industry has attempted to solve the resume screening challenge through manual review and early automation, but both approaches fall short.

Traditional manual screening gives recruiters direct control but is unbearably slow, inconsistent, and heavily influenced by unconscious biases. In practice, recruiters spend only seconds per resume, making fairness and accuracy impossible at scale. To address this, organizations adopted Applicant Tracking Systems (ATS). These systems were designed to handle volume by filtering applications through keyword matching. While effective at reducing workload, ATS have become “keyword

prisons.” They cannot recognize synonyms, context, or intent. A candidate with the right skills can be excluded simply because their resume uses different terminology than the job description. This rigidness produces high false negatives and systematically discards talent. Attempts to mitigate bias through structured shortlisting or blind reviews have been limited, since existing systems lack advanced natural language understanding. Moreover, today’s solutions provide no strategic insight. Recruiters end up with yes/no piles of resumes, but without analytics to identify skill gaps, source quality, or hiring trends. In short, current solutions either drown recruiters in inefficiency or rely on brittle automation. Neither delivers the fairness, intelligence, or adaptability that modern hiring urgently demands.

## 

## Project Goals and Milestones

The overarching goal of this project is to create a next-generation, AI-powered resume screening platform that directly addresses the failures of manual review and traditional Applicant Tracking Systems. Instead of rigid keyword filters or biased human judgments, the system will harness semantic intelligence, contextual scoring, and data-driven insights to revolutionize how talent is discovered and evaluated.

## 

GENERAL GOALS:

* Engineer a data pipeline capable of ingesting resumes in multiple formats and converting them into high-quality, machine-readable data.
* Deploy a state-of-the-art semantic matching engine using Transformer architectures (Sentence-BERT) to understand meaning rather than just words.
* Design a multi-dimensional scoring algorithm that evaluates candidates on skills, experience, education, and resume quality—allowing recruiters to configure weights according to organizational priorities
* Build a “Blind Screening Mode” that redacts personal identifiers with Named Entity Recognition, systematically reducing unconscious bias.
* Deliver an interactive recruiter dashboard that not only ranks candidates but also surfaces strategic insights about applicant pools, skills gaps, and sourcing quality.

MILESTONES :

**Weeks 1:** Research, finalize technologies, and prepare dataset.

**Weeks 2–3:** Implement text extraction, preprocessing, and semantic matching modules.

**Weeks 4–6:** Add advanced features—Skill Gap Analysis, Resume Quality Scoring, and Bias Reduction.

**Weeks 7–9:** Develop backend with Flask, design REST APIs, and integrate database.

**Weeks 9–11:** Build frontend recruiter interface and analytics dashboard.

**Weeks 12–13:** Integrate all modules, conduct full system testing, refine algorithms, and prepare final deliverables.

By executing this roadmap, the project will deliver not just a prototype, but a proof-of-concept that demonstrates how intelligent automation can unlock fairness, speed, and strategic power in modern hiring.

## Project Approach

This project will be realized through a carefully engineered pipeline that combines cutting-edge Artificial Intelligence, robust software design, and human-centric interfaces. The approach emphasizes modularity, scalability, and fairness, ensuring that the system not only functions as a prototype but also demonstrates real-world viability.

**At the core lies Natural Language Processing (NLP). Using advanced**

**Transformer-based models (Sentence-BERT), the system will move beyond**

**keyword searches and into true semantic understanding. Job descriptions and resumes will be transformed into high-dimensional embeddings, enabling the engine to calculate contextual relevance with unprecedented accuracy. This ensures that candidates are assessed on meaning and intent rather than superficial word matches.To mitigate bias, a Blind Screening Mode will redact sensitive information using spaCy’s Named Entity Recognition (NER),**

**augmented with regex-based detection of emails, phone numbers, and other identifiers. This ensures that evaluations are based on merit, not unconscious bias.**

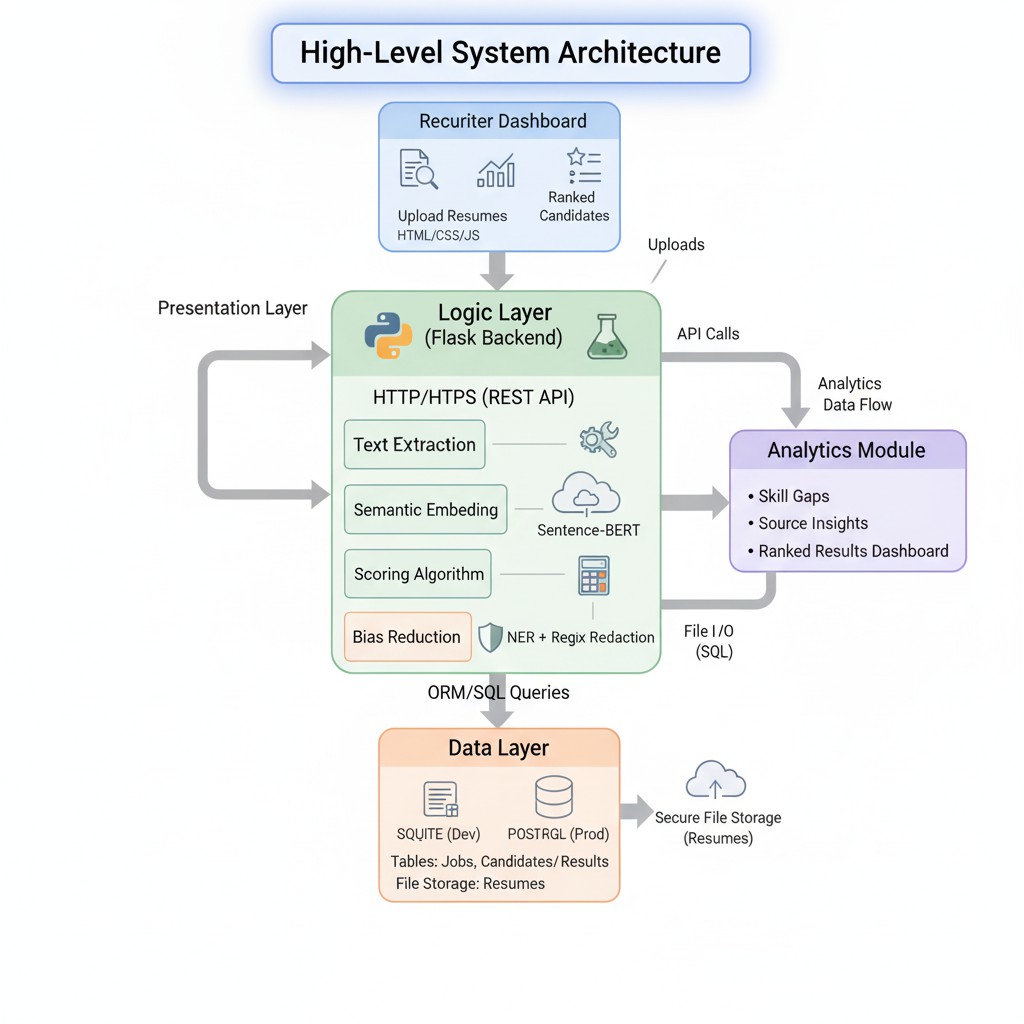
The architecture will follow a **three-tier model**:

1. **Frontend:** A web-based recruiter dashboard (HTML, CSS, JavaScript) for uploading resumes, managing jobs, and visualizing ranked results.
2. **Backend:** A Flask-powered engine in Python that orchestrates data ingestion, semantic analysis, scoring, and API delivery.
3. **Database Layer:** SQLite for development and PostgreSQL for production, storing jobs, candidates, and screening results, with JSON support for flexible scoring weights and analytics.

The scoring algorithm will combine semantic similarity, experience, education, and resume quality into a recruiter-configurable final score. The system will also feature an analytics dashboard, empowering HR to identify skill gaps, sourcing trends, and recruitment bottlenecks.

By integrating state-of-the-art AI with practical system design, this approach transforms resume screening from a manual bottleneck into an intelligent, fair, and strategic decision-support process.

## System Architecture (High Level Diagram)



## 

## Project Outcome / Deliverables

The outcome of this project will be a fully functional **AI-powered resume screening prototype** that demonstrates how intelligent automation can transform recruitment. Unlike traditional ATS systems, the deliverable will showcase **semantic understanding, bias reduction, and actionable analytics**—three capabilities largely absent in current solutions.

CORE DELIVERABLES

1.A **modular resume ingestion and parsing pipeline**, capable of processing resumes in PDF and DOCX formats and extracting structured, machine-readable text.

2.A **semantic matching engine**, powered by Transformer-based models, that aligns candidate resumes with job descriptions at a contextual level rather than mere keyword matching.

3.A **multi-dimensional scoring algorithm** that evaluates skills, experience, education, and resume quality, producing a recruiter-configurable ranking of candidates.

4.A **Blind Screening Mode** that redacts sensitive identifiers using NER and regex, minimizing unconscious bias and promoting fairer evaluations.

1. A **recruiter-facing dashboard** built with Flask and web technologies, featuring ranked candidate lists, visual analytics, and insights into applicant trends and skill gaps.
2. **Comprehensiv**e **documentation and testing**, ensuring the system’s reliability, clarity, and future extensibility.

In essence, the project will deliver a decision-support tool that empowers HR professionals to make faster, smarter, and more equitable hiring decisions—laying the groundwork for future enhancements like predictive analytics and seamless HRIS integration.

## 

## Assumptions

**This project assumes that all resumes are submitted in supported digital formats (PDF or DOCX) and that applicants provide truthful, clear information in their documents. The prototype will focus exclusively on the screening and ranking phase of recruitment, not on interviews or later stages of hiring. It is also assumed that English will be the primary language for initial testing, with limited support for Hindi. Finally, the system will not verify credentials or past employment; instead, it will treat provided data at face value to evaluate candidates fairly and consistently.**

## References

1. **Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K.** (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT. <https://arxiv.org/abs/1810.048>
2. **Reimers, N., & Gurevych, I.** (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP. <https://arxiv.org/abs/1908.10084>
3. **spaCy Documentation.** (2023). Industrial-strength Natural Language Processing in Python. [https://spacy.io](https://spacy.io/)
4. **Sentence-Transformers Library.** (2023). Pre-trained Models for Semantic Textual Similarity. [https://www.sbert.net](https://www.sbert.net/)
5. **Industry Reports on Recruitment & HR Tech.** (2022–2023). Deloitte Insights,

Gartner HR Analytics, and LinkedIn Talent Solutions research reports on AI in hiring and unconscious bias.

1. **Python Flask Framework.** (2023). Lightweight WSGI web application framework documentation. [https://flask.palletsprojects.com](https://flask.palletsprojects.com/)
2. **SQLite & PostgreSQL Documentation.** (2023). Official database references for structured data management.