International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 2, 2026

AI Powered ATS Resume And Skill Gap Analyzer

G Sudhakar Raju[1], B.Sounidhi[2], K. Varshitha[3], B.Navyanjali[4]

[1]Associate Professor; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3,4]B.Tech Student's; Department Of Information Technology Bhoj Reddy Engineering College For Women Hyderabad India.

Abstract

The exponential rise in digital job applications has made manual resume evaluation increasingly impractical, introducing inefficiencies, high operational costs, and susceptibility to subjective bias. Conventional Applicant Tracking Systems (ATS) attempt to address scalability through rule-based keyword filtering; however, such systems depend heavily on exact lexical matches, often discarding competent candidates due to variations in terminology or document structure. This study proposes an intelligent recruitment framework developed using the Django web framework and enhanced with advanced Natural Language Processing (NLP) techniques. The system leverages Sentence-BERT (SBERT) embeddings to represent resumes and job descriptions in a semantic vector space, enabling contextual comparison beyond surface-level keyword matching. A cosine similarity mechanism is applied to quantify alignment between candidate profiles and dynamically generated job requirements. To address the limitations of static job descriptions, the framework integrates a real-time data acquisition module that constructs role-specific descriptions using DuckDuckGo-based metadata extraction. Additionally, a skill gap analysis component identifies discrepancies between candidate competencies and job expectations, providing actionable insights for both recruiters and applicants. An administrative dashboard further supports decision-making by offering interpretable matching scores and career guidance suggestions. The proposed system enhances recruitment efficiency, reduces bias, and improves candidate-job alignment by combining semantic understanding with automated data generation, thereby presenting a scalable alternative to traditional ATS models.

Keywords

Applicant Tracking System, Natural Language Processing, Sentence-BERT, Semantic Similarity, Recruitment Automation, Resume Screening, Skill Gap Analysis, Django, Cosine Similarity

Introduction

The rapid growth of digital recruitment platforms has significantly increased the volume of job applications, creating substantial challenges for organizations in managing candidate evaluation efficiently. Manual screening methods are no longer practical at scale, leading to widespread reliance on Applicant Tracking Systems (ATS). However, most conventional ATS solutions operate on rule-based keyword matching, which lacks the ability to interpret contextual meaning within resumes and job descriptions. This limitation often results in the exclusion of capable candidates who describe their qualifications using varied terminology or non-standard formats, thereby reducing the effectiveness and fairness of the hiring process.To overcome these shortcomings, this work introduces an AI-driven Applicant Tracking System integrated with a Resume and Skill Gap Analysis framework. The proposed system employs advanced Natural Language Processing (NLP) techniques to capture semantic relationships between candidate profiles and job requirements. By extracting critical attributes such as skills, educational background, and professional experience, the system performs context-aware comparisons and generates relevance-based rankings. In addition, it incorporates a skill gap identification module that highlights missing competencies and provides actionable recommendations for candidate improvement. This approach not only enhances the accuracy of candidate selection but also contributes to a more transparent, efficient, and intelligent recruitment workflow.

Related Work

Survey

Recent advancements in the integration of Natural Language Processing (NLP) within recruitment technologies have significantly transformed automated candidate evaluation. Early approaches primarily relied on statistical techniques such as Term Frequency–Inverse Document Frequency (TF-IDF) and Bag-of-Words models, which focused on frequency-based text representation. While these methods were effective for basic document indexing, they lacked the ability to capture semantic relationships, often failing to recognize contextual similarities between related terms. This limitation reduced their effectiveness in resume screening tasks, where variations in wording are common. The introduction of distributed word representations, including Word2Vec and GloVe, marked an improvement by enabling models to learn semantic associations between words. However, these

embeddings remained context-independent, meaning that the same word representation was applied regardless of its usage in different contexts. A major breakthrough occurred with the development of Bidirectional Encoder Representations from Transformers (BERT), which introduced deep contextual understanding by processing text bidirectionally. Despite its effectiveness, the original BERT architecture relies on cross-encoder mechanisms that are computationally expensive for large-scale similarity comparisons. To address these efficiency challenges, Sentence-BERT (SBERT) was proposed as an optimized variant designed specifically for sentence-level similarity tasks. By employing Siamese and triplet network structures, SBERT generates fixed-size embeddings that can be compared efficiently using cosine similarity. This approach significantly reduces computational overhead while maintaining high semantic accuracy. The present work builds upon this advancement by applying SBERT-based embeddings to resume analysis, enabling context-aware candidate evaluation that surpasses traditional keyword-matching techniques.

Requirement Analysis

Functional Requirements

The proposed system is designed to support a comprehensive set of functionalities that enable efficient recruitment analysis. It incorporates a secure user management module that facilitates registration, authentication, and session handling while maintaining clear separation between candidate and administrative roles. The system is capable of processing resumes in multiple formats, including PDF and DOCX, ensuring accurate extraction of textual content without structural loss.

In addition, the platform integrates a dynamic job description generation component that retrieves relevant information from online sources based on user input. This ensures that job requirements remain current and aligned with industry standards. The core processing unit utilizes NLP models to transform textual data into structured representations, enabling semantic comparison between resumes and job descriptions. The system computes similarity scores and generates detailed evaluation metrics, which are presented through an interactive dashboard that provides insights into candidate performance and historical analysis.

Non-Functional Requirements

The system is designed with a strong emphasis on performance, security, scalability, and usability to ensure a reliable user experience. It is expected to process user inputs efficiently and deliver analysis results with minimal latency, even under concurrent usage conditions. Security measures such as authentication, data protection, and secure handling of user inputs are implemented to safeguard sensitive information.

Scalability is a key consideration, allowing the system to accommodate increasing data volumes and user activity without degradation in performance. The interface is designed to be intuitive and accessible, ensuring ease of use for both technical and non-technical users. Reliability and availability are also prioritized, with mechanisms in place to handle errors gracefully and maintain consistent system operation. The platform aims to provide continuous access with minimal downtime, ensuring uninterrupted service delivery.

Computational Resources

The implementation of the system requires a combination of software and hardware resources to support its functionality. On the software side, the application is developed using Python (version 3.9 or higher) and the Django framework for backend processing and web integration. The frontend is built using HTML5, JavaScript, and Tailwind CSS to ensure responsive design. SQLite is used as the default database for development purposes, while NLP processing is supported through libraries such as scikit-learn, sentence-transformers, NLTK, and spaCy. Additional tools, including PyPDF2 and python-docx, are utilized for document parsing, and web data retrieval is handled through DuckDuckGo-based search integration.From a hardware perspective, the system requires a machine with at least an Intel i5 processor or equivalent, a minimum of 8 GB RAM, and sufficient storage capacity (512 GB or higher) to handle data processing and storage needs efficiently.

System Design

Architecture

System Architecture

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026

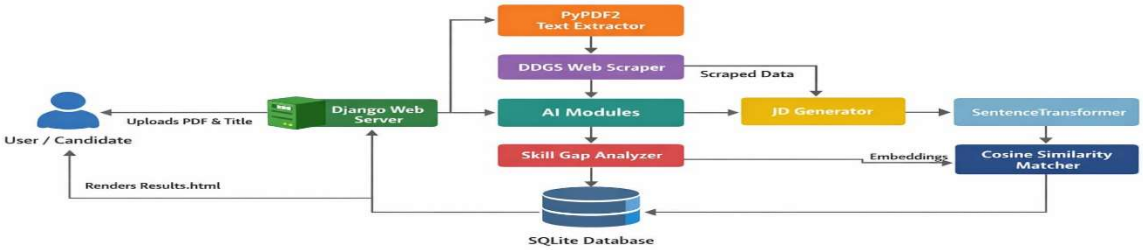


Fig. 1 System Architecture

System architecture defines the overall structure, organization, and operational flow of the proposed application, ensuring that all components function cohesively to meet the specified requirements. In this work, the architecture is designed as a modular and scalable framework that integrates web technologies with advanced Natural Language Processing (NLP) models. The system follows a layered approach, consisting of presentation, application, and data processing layers, each responsible for specific functionalities within the recruitment pipeline.At a high level, the system begins with user interaction through a web-based interface, where candidates upload resumes and administrators input job roles. The backend processes these inputs by extracting textual data, which is then transformed into semantic representations using embedding models. A similarity computation module evaluates the alignment between candidate profiles and job descriptions, generating quantitative scores. These results are further analyzed to identify skill gaps and are presented through an interactive dashboard.The architecture emphasizes abstraction and modularity, allowing individual components—such as resume parsing, job description generation, and similarity analysis—to operate independently while maintaining interoperability. This design supports future enhancements, including integration with external data sources and deployment in distributed environments. Additionally, the system aligns with established architectural principles by addressing stakeholder requirements, ensuring scalability, and maintaining consistency across the application lifecycle.

Technical Architecture

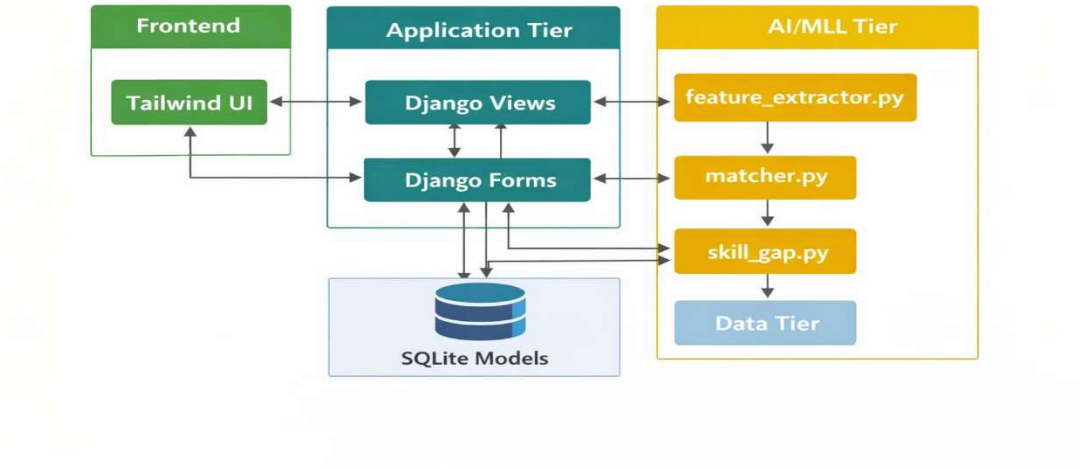


Fig. 2 Technical Architecture

The technical architecture outlines the implementation-level structure of the system, focusing on the interaction between software components and the supporting infrastructure. The proposed solution is developed using the Django framework, which manages request handling, business logic, and

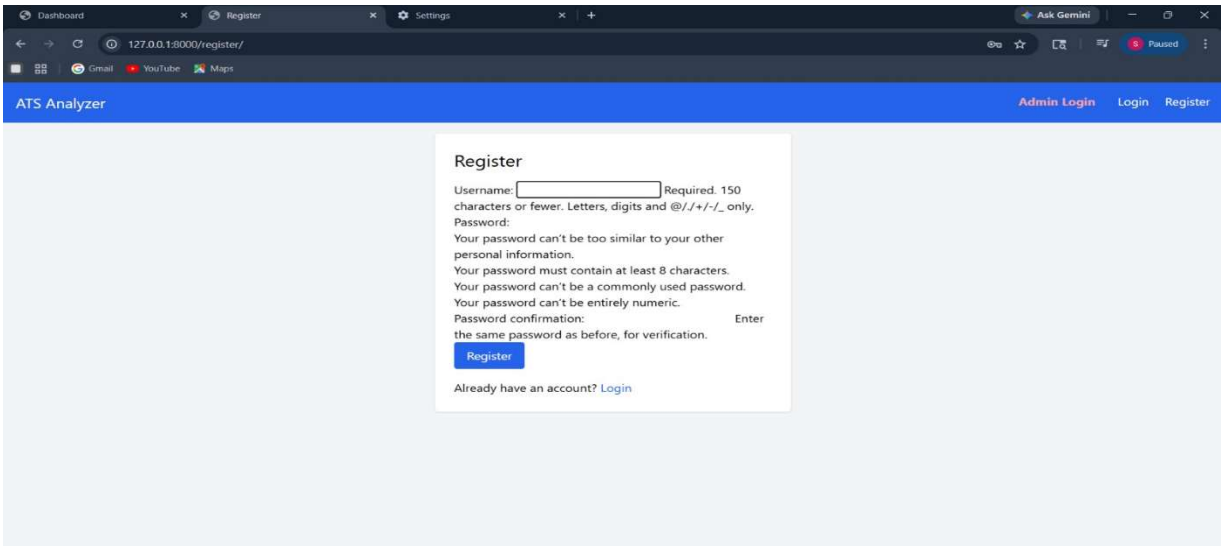
database interactions. The frontend layer is implemented using HTML5, JavaScript, and Tailwind CSS, providing a responsive and user-friendly interface.The core processing engine integrates multiple NLP libraries, including sentence-transformers for semantic embedding generation, along with supporting tools for text preprocessing and feature extraction. Resume documents in PDF and DOCX formats are processed using specialized parsing libraries, ensuring accurate data extraction. The system also incorporates an external data retrieval module that dynamically generates job descriptions based on real-time web information. From an infrastructure perspective, the application is designed to operate on standard computing environments with moderate hardware requirements. The architecture supports efficient communication between components through well-defined interfaces, enabling smooth data flow across the system. Furthermore, the design ensures compatibility with modern deployment practices, allowing scalability through cloud-based or distributed setups if required.

Implementation

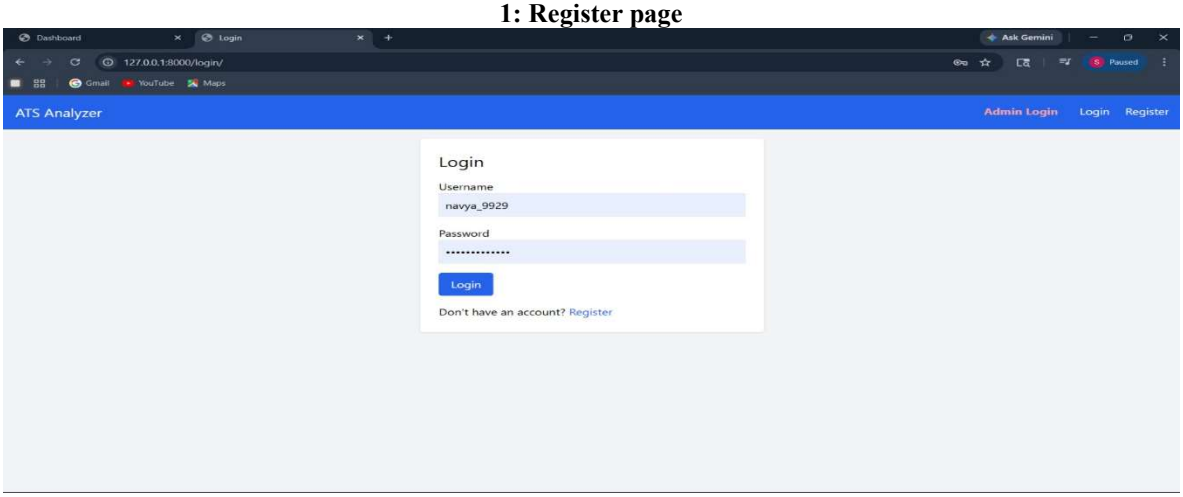
The proposed system is implemented as a web-based application using the Django framework, integrating backend processing, database management, and frontend visualization into a unified platform. The architecture follows a modular design in which authentication, resume processing, analysis, and visualization are handled as independent yet interconnected components. The system supports two primary user roles: candidates and administrators. Authentication mechanisms ensure secure access, with role-based routing that directs users to personalized dashboards or administrative control panels. The resume ingestion module allows users to upload documents in PDF format or directly input textual content. Uploaded files are processed using document parsing libraries to extract raw text, which is subsequently pre-processed through tokenization and stop-word removal. The cleaned text is then passed to a feature extraction pipeline that combines traditional techniques such as TF-IDF with advanced embedding models. Sentence-level embeddings are generated using a pre-trained Sentence-BERT model, enabling the transformation of unstructured text into semantically meaningful vector representations.To evaluate candidate suitability, the system employs a similarity computation module that compares resume embeddings with job description embeddings. Job descriptions are dynamically generated through an integrated web retrieval mechanism, which gathers relevant role-specific information from online sources. In cases where external data is unavailable, a fallback template-based generator produces structured job requirements based on predefined skill mappings. The similarity between resume and job description vectors is calculated using cosine similarity, resulting in a normalized match score that reflects contextual alignment.In addition to similarity scoring, a skill gap analysis component is implemented to identify discrepancies between candidate competencies and job requirements. This module uses a curated repository of technical skills to extract known skills from both resume and job description texts. Skills present in the job description but absent in the resume are flagged as gaps, providing actionable insights for improvement. The extracted and missing skills are stored in a structured format within the database for further analysis.The system also includes a dynamic career path generation feature, which leverages the latest analysis results to construct a predictive progression roadmap. This module estimates future roles, required skills, and development milestones based on the candidate's current profile and target position. Additionally, it generates recommended job opportunities with approximate match scores, offering users a broader perspective on their employability.From a data management perspective, the application uses a relational database to store user information, resumes, job descriptions, and analysis results. Each resume is associated with a user profile, and analysis outputs are persistently recorded to enable historical tracking and dashboard visualization. The administrative interface provides aggregated insights, including total users, uploaded resumes, and system activity metrics.The frontend is developed using HTML, JavaScript, and a utility-first CSS framework, ensuring a responsive and user-friendly interface. Dashboards display match scores, extracted skills, missing competencies, and detailed analysis results in an intuitive format. The system is designed to handle multiple user requests efficiently, with scalable components that can be extended for cloud deployment or large-scale enterprise use.

Screenshots

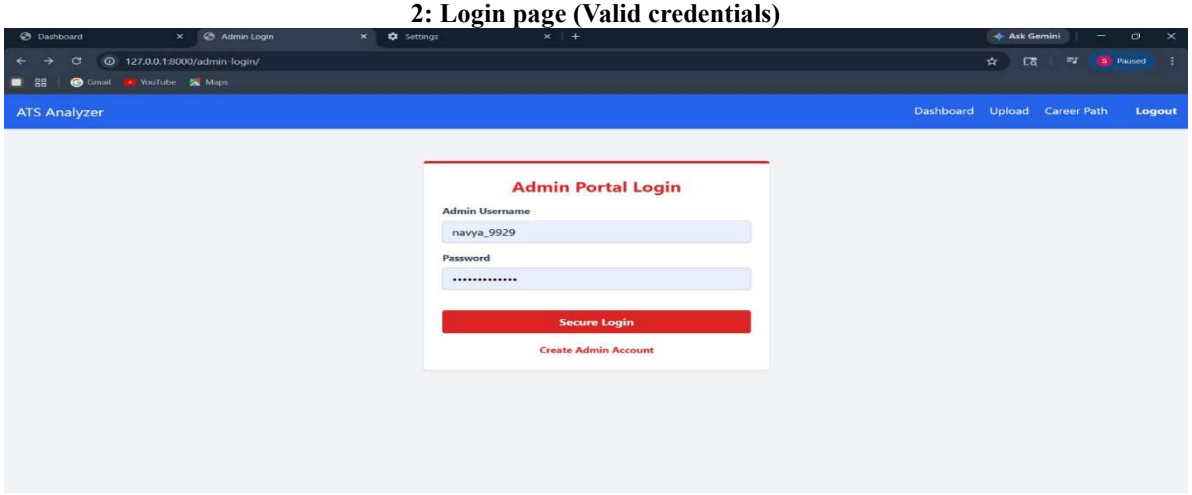
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



1: Register page

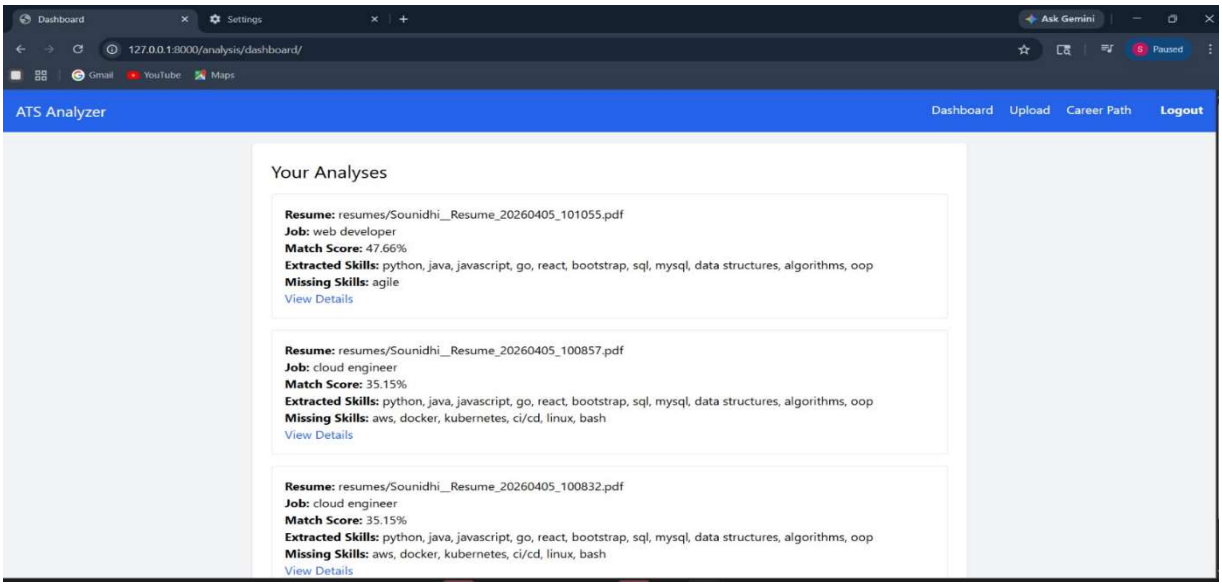


2: Login page (Valid credentials)

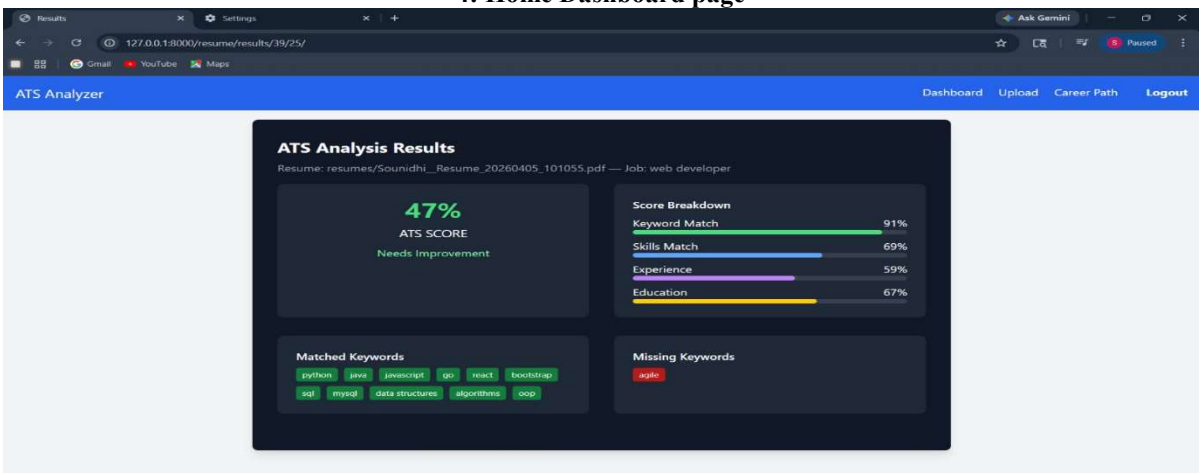


3: Admin Login page

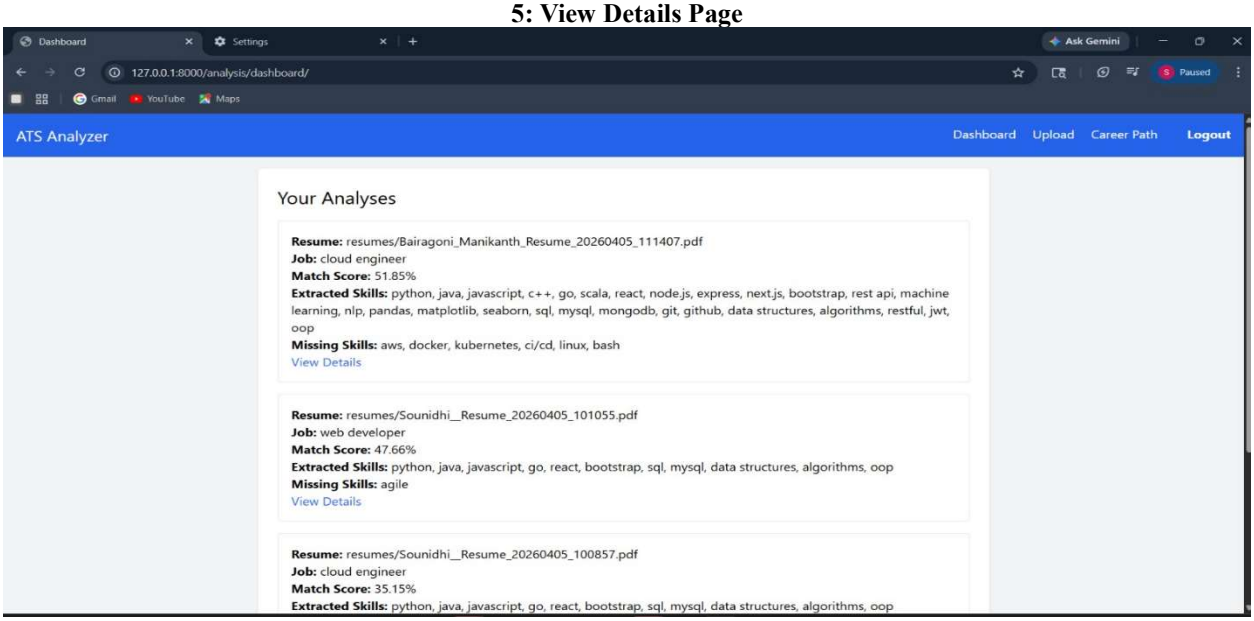
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



4: Home Dashboard page

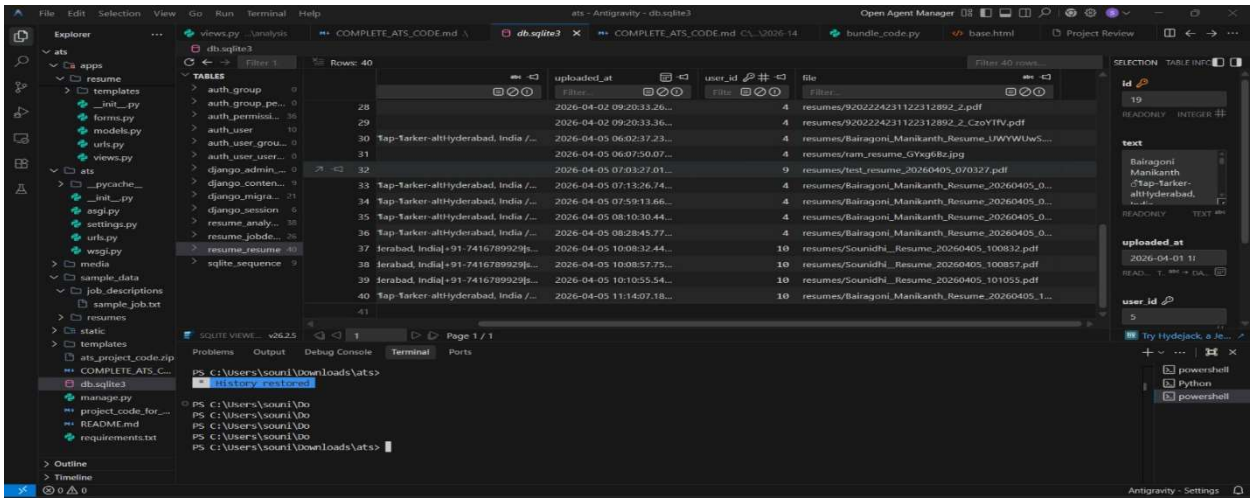


5: View Details Page



6: analysis Dashboard page

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 2, 2026



7: Backend Database age

Test Cases

The developed system was evaluated through a combination of unit testing, integration testing, and system-level validation to ensure functional correctness and robustness. Unit tests were conducted on individual modules such as user authentication, resume parsing, job description generation, semantic similarity computation, and skill gap analysis. The authentication module successfully validated new user registrations by securely storing hashed credentials in the database. The document processing component demonstrated reliable extraction of continuous textual content from PDF files without significant loss of structure.The job description generation module was tested using various role inputs, consistently producing relevant and structured descriptions enriched with domain-specific keywords. The semantic similarity module was evaluated using contrasting and identical input pairs. For unrelated roles such as healthcare and software development, the model produced similarity scores close to zero, indicating correct contextual differentiation. Conversely, identical role comparisons yielded high similarity values, validating the accuracy of embedding-based matching. Minor inconsistencies observed in early testing were resolved through preprocessing improvements and embedding normalization.The skill gap analysis component was verified by comparing predefined resume and job description inputs, accurately identifying missing competencies. Integration testing ensured smooth interaction between modules, including successful file uploads, triggering of job description generation, embedding conversion, and storage of computed results in the database. End-to-end system testing confirmed that the complete workflow—from user login to resume analysis—operates efficiently, typically producing results within a few seconds.Additional system-level tests validated robustness under edge conditions. Invalid file uploads were correctly rejected with appropriate error messages, and simulated network failures during job description retrieval triggered fallback mechanisms without interrupting system execution. User interface responsiveness was also verified across different screen sizes, confirming adaptability for mobile and desktop environments. Overall, the testing process demonstrates that the system is stable, reliable, and capable of handling real-world usage scenarios.

Conclusion

This research presents the design and implementation of an AI-driven Applicant Tracking System that enhances traditional recruitment processes through the integration of Natural Language Processing and machine learning techniques. Unlike conventional systems that rely on rigid keyword matching, the proposed approach leverages semantic embeddings to evaluate the contextual alignment between resumes and job descriptions. This results in a more accurate and fair assessment of candidate suitability. The system effectively automates key stages of recruitment, including resume parsing, job description generation, similarity evaluation, and skill gap identification. By reducing manual effort and minimizing bias introduced by superficial keyword filtering, it improves both efficiency and decision-making quality. The inclusion of analytical dashboards and career guidance features further extends its utility beyond candidate screening, providing actionable insights for users. Extensive testing confirms that the system performs reliably across different scenarios, including invalid inputs and external data failures. Its modular design ensures scalability and flexibility,

allowing future enhancements without significant architectural changes. Overall, the project demonstrates the practical applicability of intelligent systems in modern recruitment and highlights the potential of AI to transform hiring practices into more data-driven and inclusive processes.

Future Scope

While the current system delivers strong performance, several opportunities exist for further enhancement. One potential direction involves integrating advanced large-scale language models to generate more personalized and context-aware feedback for candidates. Such models could provide detailed recommendations for skill development and career progression, moving beyond static analysis toward interactive guidance. Another area of improvement lies in document processing capabilities. Incorporating Optical Character Recognition (OCR) technologies would enable the system to handle scanned or image-based resumes more effectively. Additionally, expanding the system to support multilingual processing would increase accessibility for a broader range of users. Future developments may also include real-time labor market analysis, enabling dynamic alignment of job descriptions with evolving industry trends. Integration with cloud-based infrastructure could further improve scalability and performance, making the system suitable for enterprise-level deployment. These enhancements would strengthen the system's ability to function as a comprehensive and intelligent recruitment platform.

References

[1] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," *Proceedings of EMNLP*, 2019.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018.

[3] Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2024.

[4] S. Bird, E. Klein, and E. Loper, "Natural Language Toolkit (NLTK)," 2009.

[5] Tailwind Labs, "Tailwind CSS Documentation," 2024.

[6] Hugging Face Inc., "Sentence Transformers Model Documentation," 2024.

[7] Python Software Foundation, "SQLite3 Documentation," 2024.

[8] PyPDF2 Developers, "PyPDF2 Documentation," 2024.

[9] DuckDuckGo, "Search Engine and API Documentation," 2024.