# AI Based Resume Shortlisting and Job Recommendation systems

**Umesh Kailas Shingare[1], Om Balasaheb Taskar[2], Kalpesh Subhash Wagh[3], Saurav Anil Sultane[4], Om vishnu Autade[5], Prof D.S.Shingate[6]**

*Under the guidance Of Information Technology Met's Institute Ofengineering,Nashik*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract**- *In today's world, companies receive hundreds of resumes for every job opening, making it difficult and time-consuming for HR teams to find the right candidates. To solve this problem, this project introduces an AI-Based Resume Shortlisting and Job Recommendation System that automates the recruitment process using artificial intelligence and machine learning. This system helps both candidates and HR (Admin) users. Candidates can register, upload their resumes, and instantly receive a resume score, skill improvement suggestions, and job recommendations based on their profile. They can also apply for jobs directly through the system. On the other hand, HR (Admin) can create job posts, view uploaded resumes, and use the machine learning model to automatically find the most suitable candidates for each job. The admin can also provide feedback to candidates who were not selected, helping them understand their skill gaps and improve. Developed using Python and Django, this project makes the recruitment process faster, more accurate, and smarter by combining automation with artificial intelligence to match the right people with the right jobs*

***Key Words:*** **Artificial Intelligence, Machine Learning, Resume Screening, Job Recommendation, Recruitment Automation, Candidate Evaluation.**

## I. INTRODUCTION

In today's competitive job market, companies receive hundreds of resumes for every job opening. It becomes a difficult and time-consuming task for HR teams to go through each resume manually and find the right candidate. At the same time, job seekers often do not know which jobs best match their skills or how to improve their resumes to stand out. To address these problems, this project introduces an AI- Based Resume Shortlisting and Job Recommendation System developed using Python and Django. The system uses artificial intelligence and ma chine Learning to automatically analyze resumes and match them with the most suitable job roles. There are two types of users in this system: Admin (HR) and Candidate. Candidates can register, upload their resumes, and receive a resume score, skill improvement suggestions, and job recommendations based on their profile. The Admin (HR) can create job posts, view uploaded resumes, and use the AI model to shortlist the best candidates for a particular job. Admins can also provide feedback to candidates who are not selected, helping them identify skill gaps and improve. By using AI to automate resume analysis and job matching, this system makes the recruitment process faster, fairer, and more efficient for both employers and job seekers.

## II. BACKGROUND

### *A.AI-Based Resume Shortlisting and Job Recommendation*

Artificial Intelligence (AI) has reshaped recruitment workflows by automating the processes of resume evaluation, skill extraction, and candidate-job matching. Conventional manual screening is often slow, inconsistent, and susceptible to human bias, especially when organizations receive thousands of resumes for a single opening. AI-based systems address these inefficiencies by applying Natural Language Processing (NLP) and Machine Learning (ML) algorithms to analyze resume content, extract key competencies, and evaluate compatibility with available job descriptions.

These intelligent systems transform unstructured text (PDF or DOCX resumes) into structured, comparable data formats. Using textual and semantic similarity measures, they can rank candidates, recommend suitable positions, and even identify missing skill areas. Such systems benefit both candidates, who gain insights into job fit and skill improvement, and HR administrators, who can shortlist top applicants rapidly and objectively. By integrating automated scoring, recommendation engines, and feedback loops, AI-driven shortlisting enhances efficiency, fairness, and data-driven hiring decisions.

### *B. Setbacks in Traditional and AI-Based Systems*

Despite notable advancements, both conventional and modern AI recruitment approaches present several limitations. Traditional systems rely heavily on keyword matching or manual scanning, which leads to inconsistent outcomes, redundancy, and bias. They often fail to detect context, synonyms, or implied competencies within resumes. For example, a candidate listing “statistical modelling” may not be recognized for a role requiring “data analytics” because of lexical mismatch.

Early AI solutions improved automation but lacked contextual awareness and adaptability across industries. Their accuracy depended on dataset diversity and resume formatting consistency, which vary greatly among applicants. Furthermore, ethical concerns—such as algorithmic bias, transparency, and data privacy—remain critical challenges. A model trained on biased data may unintentionally Favor specific demographics or institutions, undermining fairness in candidate selection.

Hence, a robust resume-shortlisting system must balance accuracy, interpretability, and ethical integrity by integrating advanced NLP architectures, transparent feedback mechanisms, and secure data- handling protocols.

***C. Domain and Context Dependency Challenges***

Resume interpretation is inherently domain-sensitive. A technical resume emphasizing programming languages and frameworks differs vastly from one for finance or healthcare. Keywords such as “Python,” “budgeting,” or “patient care” hold high value in their respective domains but may be irrelevant elsewhere. Additionally, the contextual weight of certain phrases changes with job type—for instance, “client communication” is critical in marketing but Peripheral in software engineering.

Models trained on generic corpora struggle to maintain accuracy across domains due to context dependency and semantic drift. Moreover, resumes often contain implicit indicators—such as project descriptions or certifications—that require contextual linking rather than surface-level matching. Therefore, advanced systems must integrate context-aware embeddings (e.g., BERT or Sentence Transformers) and domain-adaptation techniques to generalize effectively across multiple job sectors.

The proposed AI-Based Resume Shortlisting and Job Recommendation System addresses these challenges through semantic feature extraction, domain-tuned ML models, and contextual job mapping, ensuring precise, unbiased, and scalable recruitment outcomes

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have significantly enhanced recruitment automation by enabling data- driven decision-making in resume screening and job matching. Traditional manual screening methods, which rely on subjective human judgment, are being replaced by intelligent systems capable of parsing, evaluating, and ranking candidate resumes with greater accuracy and efficiency. This section reviews several prominent studies relevant to AI-based resume shortlisting and job recommendation systems, analyzing their methodologies, strengths, and limitation

AI-Based Resume Screening and Ranking System (A. Kumar, P. Sharma, R. Gupta, 2023): This study proposed an AI-powered framework using Natural Language Processing (NLP) and Machine Learning (ML) techniques to automatically analyze and rank resumes. The system extracted key features such as skills, education, and experience to determine candidate suitability. Although the approach demonstrated efficient ranking performance, it relied on a limited dataset, leading to potential bias and underperformance in identifying soft skills. The authors suggested expanding training data diversity to enhance model generalization and accuracy.

Job Recommendation System Using Machine Learning (M. Patel, S. Verma, and N. Mehta, 2022) This work introduced a personalized job recommendation platform leveraging ML models to predict the best job matches for candidates based on profile data. The system utilized content-based filtering and similarity metrics to recommend job roles aligned with user skills and past applications. However, it lacked mechanisms to handle real-time job updates, which restricted adaptability in dynamic employment markets. The authors proposed integrating live data feeds and adaptive learning algorithms to address this limitation

Automated Recruitment System Using Natural Language Processing (R. Das, K. Sen, and T. Roy, 2021) This paper presented an automated recruitment system utilizing NLP for extracting essential details from resumes and matching them against job descriptions. The model achieved notable efficiency improvements in candidate selection through automated information extraction. However, system performance was highly dependent on resume formatting, resulting in inconsistent parsing outcomes when handling unstructured or visually varied documents. The study emphasized the need for robust preprocessing and data standardization methods.

AI for Human Resource Management: A Review (S. Singh, J. Thomas, and V. Nair, 2020) This comprehensive review explored various AI applications in human resource management, including recruitment, screening, and employee analytics. The authors highlighted the potential of AI to enhance transparency and reduce human intervention in early-stage hiring. Nevertheless, the paper provided only a conceptual overview and lacked practical implementation insights. It also underscored the importance of ensuring ethical AI practices, focusing on fairness, explainability, and bias mitigation in automated decision systems.

**A. Summary of Reviewed Studies**
**B. Summary of Reviewed Studies**

**1) Integration of NLP and ML in Recruitment Automation:**
Most systems employ Natural Language Processing (NLP) for feature extraction and Machine Learning (ML) for resume ranking and job prediction. These models effectively identify relevant keywords and skills but often lack deeper semantic understanding, limiting their ability to evaluate contextual or implicit competencies.

**2) Challenges in Data Diversity and Real-Time Adaptability:**
Several studies emphasize limitations due to restricted and domain-specific datasets, which hinder generalization across industries. Additionally, a lack of real-time data integration restricts responsiveness to dynamic job markets and emerging skill demands, highlighting the need for continuous data updates and adaptive model retraining.

**3) Ethical and Practical Constraints in AI Recruitment:**
While automation increases efficiency, ethical concerns such as algorithmic bias, data privacy, and explainability remain underexplored. Few frameworks address fairness or provide interpretability mechanisms for HR professionals to understand AI-driven decisions, posing risks of unintentional discrimination.

**Summary**: The reviewed literature reveals a steady transition from rule-based systems to intelligent, learning-driven models that incorporate natural language understanding and predictive analytics. However, persistent challenges—including dataset bias, contextual ambiguity, and the absence of domain adaptability—limit system robustness.
The proposed AI-Based Resume Shortlisting and Job Recommendation System builds upon these foundations by integrating context-aware NLP techniques, domain-tuned machine learning models, and transparent feedback mechanisms. This design ensures both operational accuracy and ethical fairness, positioning the system as an advanced, scalable solution for intelligent recruitment automation.

## III. THEORETICAL BACKGROUND

### *A. Resume Screening and Recommendation Framework*

The theoretical foundation of the proposed system is rooted in the concept of automated information retrieval combined with semantic text analysis. A typical resume-screening framework comprises multiple stages—resume parsing, skill extraction, feature encoding, and candidate–job alignment. Each resume is first transformed from an unstructured textual document into a structured feature vector, representing professional attributes such as skills, education, experience, and certifications.

The system then computes a resume score and job-fit index by comparing extracted candidate features with job-description vectors. This approach enables fine- grained ranking rather than binary selection, allowing recruiters to evaluate the suitability of multiple candidates simultaneously. By integrating contextual embedding models and recommendation algorithms, the framework bridges the gap between candidate qualifications and employer expectations, ensuring a data-driven and unbiased hiring pipeline.

### *B. Transformer Architecture and Pre-Trained Language Models*

The proposed model leverages advances in Transformer-based Natural Language Processing (NLP) architectures for feature extraction and semantic representation. Transformers, introduced by Vaswani et al., employ a self-attention mechanism that enables parallel processing and long-range dependency modeling within textual data. Unlike sequential models such as LSTM or GRU, Transformers capture bidirectional contextual relationships, making them ideal for analyzing resume text where context determines meaning.

Pre-trained Language Models (PLMs) such as BERT, RoBERTa, and Sentence Transformers form the semantic core of the system. They are fine-tuned on domain-specific corpora (e.g., technical, managerial, or healthcare resumes) to improve adaptability. These models generate dense vector embeddings that encode both syntactic and semantic nuances of candidate resumes and job descriptions, enabling accurate skill-to-role mapping and personalized recommendations.

### C. Syntactic and Semantic Encoding Mechanisms

While PLMs effectively capture semantic context, explicit syntactic structure is equally vital for understanding resume text. Job profiles often contain hierarchical relationships—such as roles, achievements, and technologies—that require syntactic parsing to maintain logical associations. To address this, the system employs a hybrid encoding strategy that combines semantic embeddings from Transformers with syntactic dependency features derived from parsing tools such as spaCy or NLTK.

This dual-encoding approach enhances the model's interpretability by linking job-related entities (e.g., "Developed ML model using Python") through dependency relations while preserving contextual meaning. The integration of both encoding forms reduces ambiguity and improves matching precision, particularly in multi-domain or complex resumes containing diverse skill clusters.

### D. Summary

The theoretical design unites transformer-based contextual understanding with structured syntactic representation, forming a robust analytical backbone for AI-driven recruitment. Through layered encoding, the system not only learns lexical and semantic dependencies but also recognizes hierarchical resume organization, ensuring accurate, explainable, and scalable candidate evaluation. This theoretical integration establishes the foundation for the subsequent System Overview and Methodology sections, where the practical implementation of these concepts is detailed

## IV. SYSTEM OVERVIEW

The proposed AI-Based Resume Shortlisting and Job Recommendation System is designed to automate and enhance the recruitment process using artificial intelligence and machine learning. It follows a modular architecture integrating NLP-based resume parsing, contextual feature extraction, candidate-job compatibility evaluation, and intelligent feedback generation. The system consists of six primary components—Input Resume, Data Preprocessing and Tokenization, Feature Extraction Module, Resume Matching and Recommendation Engine, Refining Module, and Technology Stack.

Each component performs a distinct role while collectively ensuring a seamless flow of data from resume submission to job recommendation. The overall framework is implemented using Python and Django, with AI components developed through Transformer-based models and machine learning pipelines.

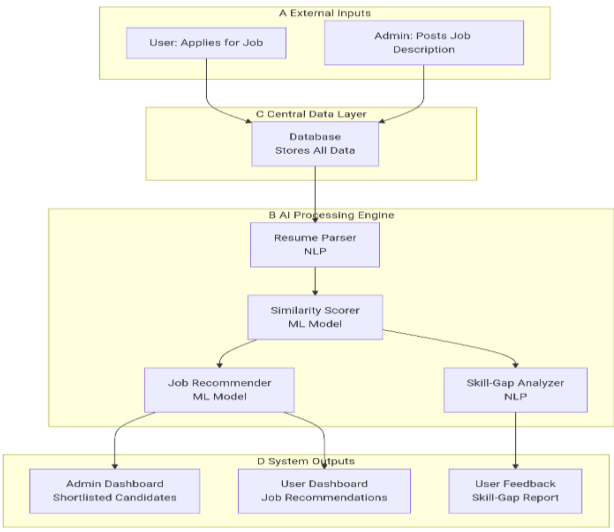



**Fig:- 1** System Architecture

### A. Input Resume

This is the initial stage of the system where the candidate uploads their resume through a secure web interface. The input can be in formats such as PDF or DOCX, which are automatically converted into text using parsing libraries like PyMuPDF or docx2txt. The uploaded data is then stored in a centralized database for further processing.

The system ensures file validation, checking for missing or unreadable content. This raw input acts as the foundation for downstream modules, initiating the workflow of information extraction and analysis.

**B. Data Preprocessing and Tokenization**

Once the resume text is extracted, it undergoes data preprocessing to remove unnecessary symbols, punctuation, and formatting inconsistencies. Common techniques such as lowercasing, stop-word removal, and lemmatization are applied to normalize the data.

After cleaning, the text is tokenized into smaller linguistic units (tokens) using libraries like NLTK or spaCy. This process enables the system to interpret resumes at a granular level, breaking down sentences into identifiable entities such as skills, degrees, and project experiences.

Tokenization and normalization ensure uniform input quality, improving the accuracy of subsequent NLP and ML operations.

***C. Feature Extraction Module***

The Feature Extraction Module transforms the preprocessed text into numerical vectors using embedding models such as TF-IDF, Word2Vec, or BERT-based Sentence Transformers. These embeddings capture semantic relationships between words, ensuring that similar concepts (e.g., "Python" and "Programming") are recognized as related.

This module extracts key features including technical skills, experience years, educational qualifications, and project summaries, which are crucial for determining candidate suitability. Each feature vector represents the candidate profile in a multi- dimensional space, enabling precise comparison with job descriptions.

**D. Resume Matching and Recommendation Engine**

This is the core AI component of the system. It evaluates the similarity between the extracted candidate vectors and job description vectors using methods such as cosine similarity or supervised ML classification.

The engine generates two key outputs:

1. A Resume Score, reflecting the overall alignment between the candidate's skills and job requirements.
2. A Job Recommendation List, ranking the most suitable openings for the applicant.

The system continuously improves through feedback loops from HR administrators, allowing model retraining and refinement based on real-world recruitment outcomes

**E. Refining Module**

The Refining Module performs post-processing to ensure precision and clarity in the generated results. It filters redundant data, resolves ambiguities, and optimizes the ranking order based on priority parameters such as required experience, skill importance, and role relevance.

This component also supports feedback integration— where HR users can mark candidate outcomes (selected or rejected)—which the model uses to fine- tune its prediction logic. The refining layer ensures the system remains adaptive, interpretable, and responsive to evolving job market patterns.

**Technology Stack**

The system integrates a modern and scalable technology stack for both backend intelligence and frontend usability:

- Backend Framework: Django (Python) for web-based implementation and API integration.
- AI Components: NLP models (BERT, TF- IDF) and ML algorithms (Logistic Regression, Random Forest).
- Database: SQLite or MySQL for structured storage of resumes, job listings, and feedback.
- Frontend: Bootstrap-based responsive web interface ensuring intuitive user interaction.

- Security: Data encryption using HTTPS and hashed credentials to protect user information.

The chosen stack ensures modularity, scalability, and easy integration with external platforms such as LinkedIn or company HR portals for data import and synchronization

## V. METHODOLOGY

The first step is **Resume submission**. The user uploads their resume through a secure web form integrated into the system's frontend interface. This allows the system to capture the candidate's information in a standardized and accessible format for further processing.

The second step is **Data transmission and Backend handling**. Once the resume is uploaded, the Django framework receives the file and sends it to the backend logic for processing. This ensures efficient communication between the frontend and backend components of the system.

The third step is **Information extraction**. The backend extracts important sections from the resume, such as skills, education, and work experience, using text-processing techniques. This structured data helps the AI model understand the candidate's qualifications and strengths.

The fourth step is **AI-based evaluation**. The machine learning model analyses the extracted data and predicts a compatibility score or job fit percentage. This score indicates how well the candidate matches the job requirements and helps HR professional's shortlist candidates more effectively.

The final step is **Result presentation**. The analyzed results, including the resume score and job recommendations, are displayed on the frontend in a clean and user-friendly interface. This allows both candidates and recruiters to easily interpret the outcomes, making the recruitment process faster, smarter, and more transparent.

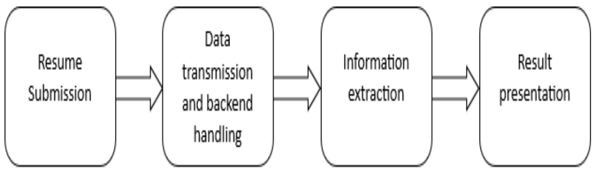



Fig. 2. System Methodology

## VI. MATHEMATICAL MODEL OF THE SYSTEM

### A. Input and Process Definition

Let the complete system S be represented as a triplet: S={I,F,O}

where,

- $I$= Set of inputs provided to the system.
- $F$= Set of functions or processes applied to the input.
- $O$= Set of outputs generated by the system.

The **input set (I)** represents the data provided by the candidate:

$$I=\{R,J\}$$

where,

- $R$= Candidate Resume Data (text extracted from uploaded files).
- $J$= Job Description Data (requirements provided by HR or admin).

These two inputs form the foundation for the AI evaluation and job matching processes.

**B. Functional Mapping**

The function set $F$ defines all the transformations applied to the input to generate meaningful output:

$$F = \{f_1, f_2, f_3, f_4, f_5\}$$

- $f_2$: Feature Extraction and Embedding Generation
- $f_3$: Resume–Job Similarity Computation
- $f_4$: Resume Scoring and Ranking
- $f_5$: Job Recommendation and Feedback Refinement

Thus, the overall functional transformation can be mathematically expressed as:

$$F: I \rightarrow O$$

And more specifically,

$$F(R, J) = O$$

**C. Output Representation**

The output set $O$represents the final results generated by the system, defined as:

$$O = \{S_R, L_J\}$$

Where,

- $S_R$ = Resume Score indicating candidate suitability for specific job roles.
- $L_J$ = List of Recommended Jobs ranked according to match percentage.

Each candidate $c_i$receives a score $s_i$calculated using similarity between candidate feature vector $V_R$ and job vector $V_J$:

$$s_i = \frac{V_R \cdot V_J}{\|V_R\| \; \|V_J\|}$$

Here, the numerator represents the dot product of the two vectors, while the denominator normalizes the similarity measure using vector magnitudes — this is the cosine similarity metric commonly used in information retrieval and recommendation systems.

The system finally outputs a ranked list of job recommendations for each candidate, where:

where each function represents a processing stage in the system workflow:

- $f_1$: Resume Preprocessing and Tokenization

$O$ = Rank({($J_k$

, $s_i$) | $k$ = 1,2, … , $n$})

### D. Use Case Diagram Explanation

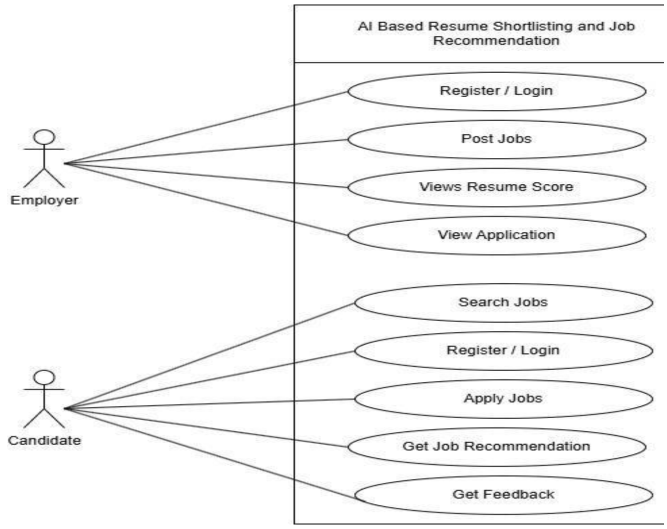



**Fig. 3.** Use Case Diagram

The Use Case Diagram illustrates the interaction between different users and the recruitment system. The primary actors include the Candidate, Recruiter, Web-to-Candidate, and Salesperson. Each actor performs specific actions within the system to facilitate the recruitment process.

The Candidate submits applications, views placement results, and engages in contract-related actions. The Recruiter is responsible for approving and reviewing candidate profiles, while the Salesperson manages client and contract information. The Web-to-Candidate actor enables online applications through the web interface.

Together, these interactions define the core functionalities of the system—application submission, candidate approval, review, placement, and contract management—ensuring seamless coordination between users and the recruitment platform

### E. Activity Diagram Explanation

The flowchart represents the operational workflow of the AI-Based Resume Shortlisting System. The process starts with reading resumes and corresponding job descriptions. The system performs keyword extraction, followed by stemming and lemmatization to normalize text data. The ranked keywords from both the resume and job description are then processed through BERT, generating feature vectors for semantic representation.

Using cosine similarity, the system evaluates the match between candidate resumes and job requirements. Once all resumes are analyzed, they are arranged according to their similarity scores, and the employer is notified of the most suitable candidates. This ensures accurate, automated, and unbiased shortlisting in the recruitment process

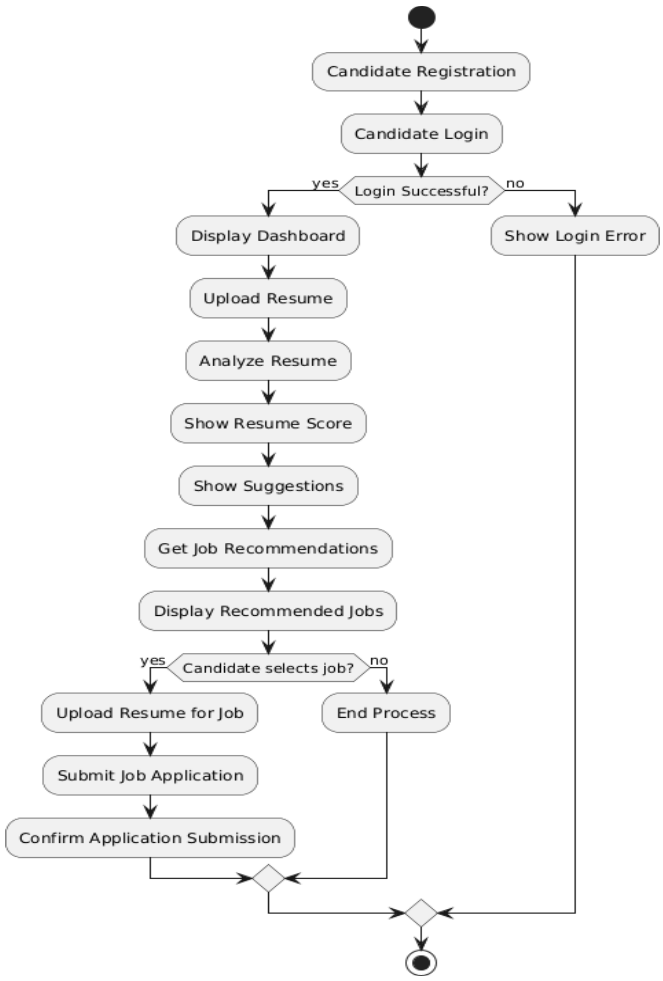
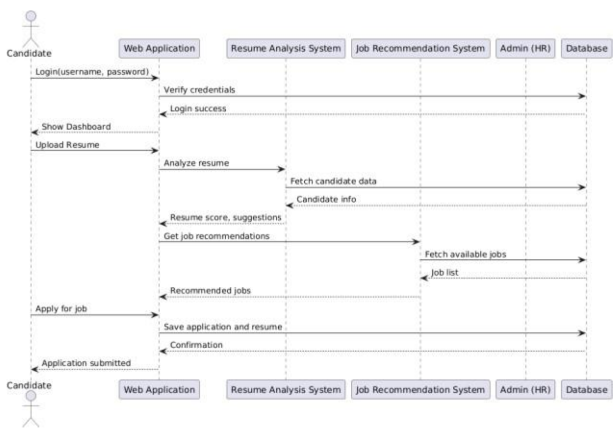



**Fig 4:-** Activity Diagram

### F. Sequence Diagram Explanation

The sequence diagram illustrates the interaction between the User, Resume Builder, Template Service, and Export Service during the resume creation process. The sequence begins when the User selects the "Create Resume" option, prompting the Resume Builder to request available templates from the Template Service. Once the list of templates is returned, the user selects one and inputs personal details.



The Template Service then processes these details using the selected template and returns a resume preview for the user to review. After confirmation, the Export Service generates the final resume file, which is made available for the user to download. This sequence ensures a structured, automated, and user-friendly workflow for creating and exporting professional resumes efficiently

## VII. RESULTS AND DISCUSSION

In conclusion, the AI-Based Resume Shortlisting and Job Recommendation System provides an intelligent and efficient solution for the recruitment process. By using Python, Django, and SQLite, the system automates resume evaluation and job matching based on candidates' skills, experience, and job requirements. The platform helps employers save time by automatically shortlisting suitable candidates and providing job recommendations to applicants. It also ensures fair and accurate screening using AI- driven analysis, reducing human bias in hiring. The user-friendly interface built with Bootstrap allows both candidates and HR users to interact with the system easily. Overall, this system improves the recruitment process by combining artificial intelligence and web technology to make hiring faster, smarter, and more effective.

## ACKNOWLEDGMENT

The authors would like to express their heartfelt gratitude to Prof. D.S.Shingate, Department of Information Technology, MET's Institute of Engineering, Nashik, for his invaluable guidance, constant encouragement, and insightful suggestions throughout the course of this project. His expertise and support played a crucial role in the successful completion of this research work. The authors also extend their appreciation to the faculty members and peers for their constructive feedback and cooperation during the development of this study.

## REFERENCES

1. Rohit Jain, Neha Sharma, and Priya Verma. (2024). AI-Powered Recruitment: Enhancing Resume Screening Using Natural Language Processing. Focuses on automating resume shortlisting using NLP to extract candidate skills and match them with job descriptions.

2. Karan Patel, Deepika Singh, and Ravi Nair. (2023). Automated Job Recommendation System Using Machine Learning and User Profiling. Develops a recommendation engine that suggests jobs based on resume data, interests, and candidate skill sets.

3. Ahmed Mohamed, Fatima Khan, and John Lewis. (2022). Intelligent Hiring System Using Django Framework and Machine Learning Algorithms. Integrates Django web framework with AI to streamline the recruitment and candidate evaluation process.

4. Vikas Kumar, Sneha Joshi, and Arjun Rao. (2024). A Study on Resume Classification Using Deep Learning Approaches. Compares deep learning models like CNN and BERT for skill extraction, job fit prediction, and automated shortlisting.

5. Aditi Singh and Meena Iyer. (2023). AI in Human Resource Management: An Overview of Recruitment Automation. Explores how artificial intelligence transforms HR operations by improving decision- making in hiring.

6. Rajesh Gupta and Harshita Malhotra. (2022). TextMining Techniques for Resume Screen ing: Challenges and Opportunities. Analyzes NLP-based resume parsing techniques and their role in automating candidate evaluation.

7. Li Chen, Maria Santos, and David Park. (2024). Design and Implementation of a Job Matching Platform Using Python and SQLite. Describes the backend development and database structure for AI- driven job recommendation systems.

8. George Thomas and Aishwarya Das. (2023). Skill Gap Analysis Using Machine Learning for Career Recommendation. Proposes an AI model that identifies missing skills in candidate profiles and provides personalized improvement advice.

9. S. R. Raoand K .Ramesh.(2025). Django-Based Web Application for Automated Recruitment Process. Presents a full-stack implementation for resume upload, job posting, and candidate ranking using AI models.

10. A. Krizhevsky, I. Sutskever, and G. Hinton. (2017). ImageNet Classification with Deep Convolutional Neural Networks. Introduced foundational deep learning concepts later adapted for NLP-based resume classification systems.