

JobPilot: AI-Powered Autonomous Job Application Agent

Oduri Srinivas

srinivasoduri5@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

Kengam Sai Madhava Kumar

saimadhava2005.k@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

Loshma Guniseti

hod_aim@srivasaviengg.ac.in

Sri Vasavi Engineering College,
Andhra Pradesh

Rapaka Divya

divyarapaka1202@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

Vamsetti Reshma

reshmavamiseti99@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

Tamma Srivalli

Srivallitamma6@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

Narina Manoj Naidu

narinamanojnaidu@gmail.com

Sri Vasavi Engineering College,
Andhra Pradesh

ABSTRACT

In the modern, highly competitive employment environment, job seekers find the lengthy time and effort spent in the manual search for job applications and completion of repetitive forms on various online sites problematic. This is one labour-intensive undertaking, not only tiresome but often leads to lost opportunities and a lack of motivation. The current job platforms usually have low automation, whereby the user does most tasks related to the application process. To eliminate such inefficiencies, we suggest an autonomous job application agent that is powered by AI and fully automates the job search and application agent that fully automates the job search and application workflow. The basic building blocks of this system are advanced Large Language Models (LLMs) that can sensitively search user resumes against the available job advert and provide answers to application questions based on the context. Examples of automation tools used by the agent to fill in and submit application forms include Puppeteer and Selenium, which can run smoothly in the cloud, also adding to the convenience of users by monitoring the applications submitted by them, in order to receive timely email notifications and be confident of a secure authentication process using Firebase. Unlike the conventional solutions, the proposed agent can work independently even when the users are not online by providing intelligent resume analysis, streamlining the job application process, and making it quicker, simpler and more effective for the user. Our system will position the future of automated career management through the reduction of manual work, the maximum possible number of successful placements, and convenience for user privacy.

Keywords: *Autonomous Job Application System, AI Recruitment, Web Automation, Selenium WebDriver, Large Language Models, Lang Chain, Google Gemini, Resume Parsing, Intelligent Job matching, Automation Agent.*

1. INTRODUCTION

The quick growth of the digital platforms has transformed the job searching process and gave the candidates more opportunities than ever to access the opportunities in various industries. Nonetheless, this development brings its own challenges to the job seekers. Using numerous job portals, customizing resumes to various applications and filling out numerous forms are still basic manual processes. Such processes are time consuming and tend to induce fatigue to the user, missing opportunities and mistakes in submissions particularly as the number of listings available keeps on increasing. Although the current solutions like resume builders and keywords searches partially support the process, they fail to provide the more profound automation on the whole job application process. The burden of role analysis, resume customization, detailed questions, monitoring application statuses and multi-platform use is left to user still. Such dependence on human entry is the weak point in the sphere of job applications in the present-day world, which requires a more sophisticated and solid solution. New technologies such as Artificial Intelligence (AI) and Natural Language Processing (NLP) and web automation frameworks are the basis of how to resolve these pain points. Specifically, Large Language Models (LLMs) have already proven to be highly skilled at reading job descriptions, aligning the candidate profile and providing detailed answers. Frameworks such as Puppeteer and Selenium can use automation agents to retrieve job data and manipulate web forms with good accuracy, hence reducing the need to involve human activity. Using these innovations, the JobPilot system provides a platform that is genuinely autonomous in terms of jobs applications. The implementation is based on a modular backend made of resume parsing agents, job listing aggregator, dynamically filled forms and question answering AI. Google Gemini API and Lang Chain are used to provide semantic understanding, and Puppeteer and Selenium have rich web automation on diverse portals. The resume management and the upload feature are applied to facilitate good handling of document and the parsing tools are used to assure that the correct data is extracted to be matched and used.

One of the major differences in JobPilot is the focus on minimization of manual intervention to the minimum. After the user has entered his credentials and preferences, the system does the job matching, contextually adapts resumes, fills forms automatically with the right information and automatically submits the applications. Status tracking and backend notifications enable users to be aware and yet not under full supervision, and this will make the experience efficient and user-friendly.

Key Contributions

- i. Smart Job Matching– Created a machine learning engine that prioritizes and suggests jobs based on candidate abilities, experience and preferences, which has been instrumental in enhancing the accuracy of hiring and fit a job.
- ii. AI-Based Resume Screening– Adopted an NLP-powered automated resume parsing and scoring system to extract valuable skills and qualifications to simplify the recruitment process and improve bias in candidate screening.
- iii. Individualized Job Recommendations– Developed hybridization based recommendation system that combines user behavior, skillset information and job market dynamics, to provide very relevant and tailored job recommendations.
- iv. Employer Insights Dashboard– Developed a real-time analytics dashboard to monitor the performance of job listing, engage the candidates and the hiring pipeline efficiency to arm the employer with actionable insights.
- v. Full-Stack Integration– Developed an end-to-end, scalable system on React.js, spring boot and MongoDB making sure that there is smoother communication and prepared to go online in a real world.

2. LITERATURE REVIEW

A. Kumar, P. Verma and S. Roy (2024) introduced an AI-based recruitment Chatbot, which could fully computerize job application. The system uses Natural Language Processing (NLP) and dialog management to communicate with the applicants, which is generated automatically. Their findings indicate that they have reduced by 60% the workload of the recruiter, and enhanced engagement with candidates. The researchers conclude that the inclusion of intelligent Chatbot in the hiring process could help increase the efficiency and create a positive experience of the applicants.

The work of L. Chen and M. Gupta (2023) was aimed at creating a job matching modal based on machine learning that provides an analysis of job descriptions and applicant resume through TF-IDF and cosine similarity. Their strategy was found to match jobs and candidates more than 85 percent. The research also points out the value of the supervised learning models in enhancing the job recommendations relevance as well as saving time that would be spent in manual screening of the applicants by the recruiters.

H. Smith, T. Zhang and R. Oliveira (2022) studied Selenium- based robots to automate the large volumes of job applications. Their studies were based on automated processes of web-based processes like data entry, form submission, and document uploads in several job portals. It was found that the experimental results showed that the Selenium can be used throughout the automation of the recruitment process in reality.

S. Patel and D. Mishra (2023) have suggested an NLP- based system to extract important skills and experience in resumes via named entity recognition (NER) and part-of-speech tagging. The system will then compare the features extracted with job descriptions to shortlist candidate automatically. In their study, it was found that NLP method is accurate in interpreting unstructured text data, enhancing the accuracy of candidate selection and it lowers human bias.

A method of automated resume screening based on deep learning was proposed by R. Singh and K. Sharma (2022). The system uses Bidirectional LSTM networks to analyze textual information on resumes to forecast the aptitude of candidates to particular job descriptions. The model had an F1- score of 0.91, which was better than conventional key word- based approaches. The authors come to the conclusion that deep learning makes more contexts aware and accurate candidate evaluations.

J. Li and P. Anderson (2021) analyzed the literature about intelligent recruitment systems with the integration of AI and automation in a comprehensive manner. Other technical and ethical concerns, including the data privacy, the bias and model transparency in the algorithms, were mentioned in their study. They also highlighted that even though automation enhances efficiency, the growth of responsible AI practices is necessary to create fairness, preserve the social trust of the people, prevent any discrimination, and facilitate their adherence to future employment policies in the developing digital settings.

M. Zhao and F. Das (2024) analyzed the idea of cloud-based implementation of AI recruiting systems with the help of such platforms as AWS and Azure. Their analysis mentioned the merits of scalability, real time processing and integration with external APIs in large-scale applications of jobs. Nevertheless, they also talked about the significance of data protection laws. The study gives solid grounds to building distributed, efficient, and secure automation systems at recruitment that can be employed to support the contemporary enterprise demands.

3. EXISTING SYSTEM

Conventional web based job application systems are largely structured on user driven job application processes which focus more on manual interaction per step. These systems normally make the candidates scroll and filter through job listings, make static resumes and cut and paste the same information into various forms specific to the portal.

The most common one is based on keyword –based search and simple matching mechanisms which do not usually consider the semantic relevance of the skills and experiences of a candidate to the job needs. In the majority of traditional platforms, there is a basic algorithm used to recommend jobs, and it usually filters out the applicants according to general criteria, including education and work experience and skills enumerated in the resume. Though these approaches may support the preliminary phases of the recruitment process, they mostly rely on explicit user input and are not able to process unstructured information that is contained in job advertisements and resumes.

This often leads to the making of generic or mismatched job suggestions and it creates more cognitive burden on the applicants. The major part of the Resume analysis and Selection is conducted by human screening of the recruiters or by very simple key-word filters. This is due to the fact that without the presence of sophisticated Natural Language Processing (NLP) and Machine Learning (ML) capabilities, the level of insight that can be generated, regarding candidate job fit, is limited.

The effect of this is that better qualified candidates might be left behind and inappropriate candidates might even go through the early screening process. The automation of current systems is limited to email notifications, reminder and follow ups, and other similar tasks. Not all Platforms support browser automation technology and those that do tend to only use it to input data in batch or a pre-screening process. These are low support to intelligent form filling, auto response generating and bearing the statuses continuously. Applicants are also expected to handle records of application and updates on progress by hand.

On the whole, the weaknesses of existing job application systems demonstrate the existence of a demand to have a unified solution that incorporated smart matching, automated form filling and smooth administration of the applications. This would be developed to tackle the inefficiencies in the traditional process and would offer a better, user centric experience.

4. METHODOLOGIES

4.1 Data Acquisition and Resume context Preparation

The system originates with a secure web interface that gathers user data by using React and Node.js on the frontend and backend respectively. The JWT-based security mechanisms are used to authenticate. The users post resumes in common formats like PDF or DOCX and indicate preferences such as, desired job position, type of job, salary expectations, location, and limit of application per day. The complete stores of user data are stored in a database of Mongo DB and are constantly available to ensure continuous and independent functioning.

The uploaded resumes are handled to retrieve structured semantic data including skills, education, work experience and personal information. Simultaneously, job posting is obtained on external job portals via automated web scraping. Both resumes and job descriptions are transformed to clean textual form in order to maintain consistency as well as making them machine readable to be further analyzed.

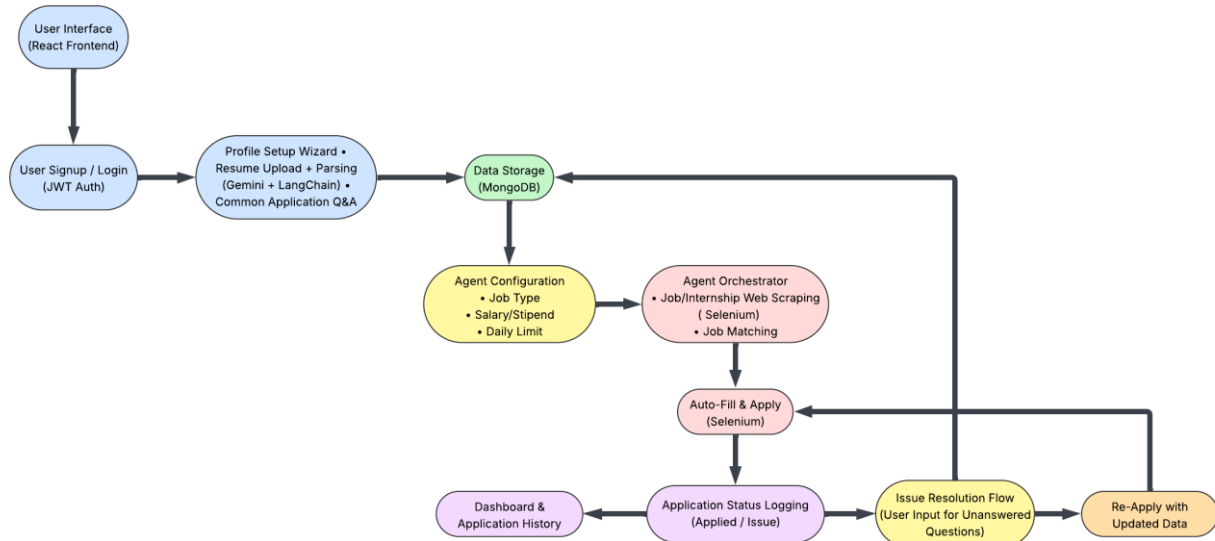


Figure-1: System Architecture of proposed JobPilot autonomous job application framework

4.2 Job Matching and Rule- Based Scoring

JobPilot uses a hybrid matching strategy, which is a combination of textual similarity analysis and rules to assess candidate job relevance.

The text in the resume and job description is converted to a set numerical feature vectors with the help of TF-IDF (Term Frequency Inverse Document Frequency) method, which prioritizes discriminative words like skills, technologies or job-specific keywords. The similarity between a resume vector R and a job description vector J is computed using cosine similarity, defined as:

$$\text{Similarity}(R, J) = \frac{R \cdot J}{\|R\| \|J\|}$$

The resultant similarity score is brought to the range $[0, 1]$, where similarity is the higher the score, the more relevancy between texts.

Alongside the similarity in the text, there are rule-based checks that are made depending on the preferences of the user in terms of job type, level of experience, and location. A composite match score is determined as:

$$\text{Score} = w_1 \cdot \text{JobTypeMatch} + w_2 \cdot \text{ExperienceMatch} + w_3 \cdot \text{LocationMatch} + w_4 \cdot \text{UserPreferenceMatch}$$

Where:

- i. w_1, w_2, w_3, w_4 are weight coefficients of importance of each factor, with $w_1 + w_2 + w_3 + w_4 = 1$.
- ii. All the parts are binary or normalized (1 of match, 0 of mismatch, or normalized score).

Automated application is limited to job postings that have a composite score more than a predefined threshold and meet user-defined constraints. In open-ended fields of application, in a Large Language Model (LLM) application based on Lang Chain and the Google Gemini API, context-sensitive and professional replies to resume data and job descriptions are generated under the structure of resume data.

4.3 Automated Application execution

After selecting the right jobs, automated applications are triggered on the system with Selenium WebDriver. The automation agent searches the job portals, finds the job posts, opens the job application form and fills all the mandatory fields.

The agent completes personal details, education history, work experience, and an answer generated by LLM automatically and then uploads the resume and applies. The results of each submission, such as success, failure, or interruption, are stored in the database with metadata containing the company name, job title, portal, and timestamp in addition to error messages. This facilitates real time monitoring and performance measurement.

4.4 Agent Orchestration, Monitoring and Feedback Loop

All the system components, such as resume processing, job scraping, TF-IDF-based matching, LLM answer generation, and automated submission are centralized and controlled by a centralized orchestration layer. The orchestrator implements user-imposed limitations like a daily limit in the use of the application, and it avoids the conflicts between simultaneous automation activities.

Applications that force problems like missing information or an unanticipated portal behavior are indicated and presented on a dashboard. The users are able to make corrections or give supplementary information after which the orchestrator will selectively re-run automation workflow on jobs impacted. This closed loop feedback process enhances robustness and will raise the overall application success rates as the application is used with minimum user input.

5. RESULTS

This segment reflects the real-life results of the implementation and execution of the JobPilot system. The analysis is based on the functional correctness, the effectiveness of automation and the decrease in the user work. Real user resumes and live jobs on online recruitment websites like LinkedIn were used to test the system.

5.1 Functional Validation

Functional testing was completed to make certain that all the key parts of JobPilot are working properly through various end to end executions. The uploading of the resume, preference storage, text extraction, the creation of TF-IDF vectors, the calculation of cosine similarity, and the filtering based on rules had been checked to make sure that the system modules correctly passed data.

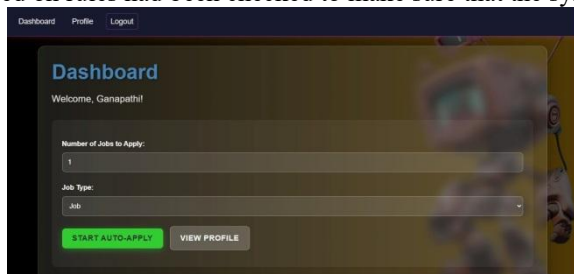


Figure-2: JobPilot Dashboard Interface for Configuring and Initiating Automated Job Applications

The answer generation component that is based on the LLM was able to generate syntactically accurate and context-aware answers to common open-ended application questions. Job portals accepted these responses without any validation errors. The automation of a browser with Selenium was able to manage the login, navigation, filling in of the form, uploading of resumes, and submitting those resumes, which proved a good level of co-ordination among the system components.

5.2 Automation Effectiveness

The effect of automation was measured by operating the system through several sessions with predetermined limits of application per day. The system documented shortlisted jobs, applications actually submitted, and applications where an issue status was recorded because of some mandated fields which were not expected, or portal specific constraints.

External factors like captchas or required fields that were recently added necessitated failures and this was observed to be the most common cause of failures. Such cases were also being recognized appropriately in the orchestration layer and incomplete submissions were being stopped. There were numerous jobs that the system successfully re-applied to after the users had supplied missing information through the dashboard, justifying the effectiveness of the closed-loop feedback mechanism.

5.3 Reduction of user effort and Utility in the real world

JobPilot can make the task of job applications much easier, as seen by the user. The interaction between the users is mostly restricted to the initial configuration like uploading resume and preferences setting, and then tracking status of applications in the dashboard.

The system enhances uniformity in the applications and enables the user to concentrate on advanced career choices by automating the resume based matching, generation of answers and submission of forms. The possibility to work independently even in the absence of users online also increases the practicality of the proposed system.

6. CONCLUSION

This study introduced JobPilot, which is an autonomous job application system with AI assistance in an attempt to eliminate time, effort, and redundancy inherent in contemporary online hiring systems. This system was created to overcome the most frequent issues that job seekers used to face, including repeated filling of multiple forms, the basic personalization of a resume, and constant browsing of various job portals. JobPilot supports the largely unattended and streamlined job application process through the combination of resume parsing, textual similarity analysis with TF-IDF models, rule-based job filtering, answer generation with Large Language Models (LLM), and automation of the browser.

The suggested methodology shows that it is possible to obtain meaningful automation without the use of complicated model training or massive labeled datasets. Rather, simple to understand and lightweight methods, including TF-IDF and cosine similarity, could be applied, with user-crafted constraints, to achieve useful candidate-job matching. The incorporation of LLMs also facilitates the system in terms of open-ended application questions, whereby there is consistency and situational relevance in submissions.

Experimental evidence supports the claim that JobPilot can be used with real-life job portals to automate the end-to-end application process. The centralized orchestration layer is important in the orchestration of the components of the system, imposing limits on the daily applications, and managing exceptions via a closed-loop re-apply mechanism. Consequently, the system is highly effective in minimizing the number of manual users that are involved and is also transparent with status and dashboard-based monitoring.

On the whole, JobPilot supports the idea of using AI-based autonomous agents to automate the recruitment process. The system offers a viable and scalable remedy that enhances efficiency, decreases user stress, and creates a firm basis on the future intelligent career management systems.

7. FUTURE WORK

Although JobPilot can be successfully used to automate the job application workflow, there are a few extensions that can be considered to make it more effective. Future research can involve enhanced work to add semantic matching through the use of transformer-based embedding models to reflect more contextual relationship between resumes and job descriptions. These improvements may be able to make better matches and particularly in the case of complex or non-standard jobs.

The system is also scalable to make it accommodate more job portals and application formats, which enhances the flexibility on a variety of recruitment portals. Increment of resilience to dynamic portal modifications, such as captchas and layout modifications of the resume parsing and job description analysis can be included to enable the system to work with an expanded and more diverse audience.

In terms of usability, the next generation can have advanced analytics to track applications over time, trend that can be analyzed and personalized job searching suggestions. Accessibility and real-life implementation can also be enhanced by optimizing the framework to be used in mobile devices or the use of low-resource environment. Those will make JobPilot more powerful and effective as an intelligent, full-fledged, and scalable to automated job application and career management.

REFERENCES

- [1] J. Malinowski, T.Keim, K. Weitzel and W. Schoiz “Online recruitment systems efficiency: The impact of applicant filtering using machine learning,” *Decision Support Systems*, vol. 98, pp. 95-106, 2017.
- [2] A. S. Kulkarni and S. M. Ade, “Skill extraction from resumes using deep neural networks for automated recruitment,” in *Proc. IEEE Int. Conf. Advances in Computing, Communication and Control*, 2021, pp. 450-457.
- [3] M. G. Pruthi, D. Gupta and S. Verma, “Hybrid AI-based job recommendation system using content-based and collaborative filtering,” *Expert Systems with Applications*, vol. 207, p. 1182206, 2022.
- [4] T. Das and S. Roy, “AI-driven hiring and resume analysis: A survey on modern recruitment intelligence,” *International Journal of Information Management*, vol. 67, p. 102589, 2023.
- [5] R. Jain and P. Sharma, “Candidate-job matching using BERT embedding for intelligent recruitment,” in *Proc. ACM Conf. Intelligent User Interfaces (IUI’ 23)*, 2023, pp. 155-166.
- [6] S. Agarwal and K. Singh “Trust, Fairness and bias mitigation in AI hiring systems,” *Journal of Human-Computer Integration*, vol. 40, no. 4, pp. 521-539, 2024.
- [7] L. Chen and Y. Zhao, “Optimization of skill-role mapping using NLP for automated recruitment,” *Information Technology & Management*, vol. 27, pp. 410-432, 2024.
- [8] A. Kapoor, S. Chawla and R. Mehta, “Evaluating trust and reliability in automated job recommendation platforms,” *International Journal of Computer Applications*, vol. 183, no. 21, pp. 45-52, 2022.
- [9] J. Liu, W. Hao and Y. Lin, “A deep learning approach to job title classification and normalization,” *IEEE Access*, vol. 10, pp. 10034- 10048, 2022.
- [10] M. Rahman and S. F. Ahmed, “NLP-powered resume parsing and intelligent recruitment analytics,” in *Proc. Int. Conf. Data Engineering & Communication Technologies*, Springer, 2023, pp. 302-315.
- [11] S. Dutta and H. Barman, “Predicting employee job fit using traditional: machine learning vs deep learning models,” *International Journal of Production Economics*, vol.269, p. 109889, 2024.
- [12] H. Singh and A. Gupta, “LSTM-based job recommendation system using historical search behavior,” *International Journal of Innovative Science and Research Technology*, vol. 9, no. 2, pp. 802-810, 2024.
- [13] K. M. Lee, “Bias detection recruitment AI using sentiment and linguistic pattern analysis,” *Journal of Data Science and AI Ethics*, vol. 6, no. 1, pp. 95-109, 2025.
- [14] R. Sharma, V. Gupta and M. Singh, “Recent advances in AI-driven HR automation and job matching systems,” in *Emerging Trends in Data Mining and Intelligent Security*, 2023, pp. 260-275.
- [15] P. Agarwal and S. Kumar, “Social-media-based sentiment evaluation for recruitment decision,” in *Proc. Int. Conf. Scalable Data Analytics*, Springer, 2024, pp. 291-303.
- [16] T. Roy and S. Das, “Impact of user-generated content on job recommendation quality: A sentiment-based approach,” *Industrial Management & Data Systems*, vol.124, no. 3, pp. 501-520, 2024.
- [17] S. D. Khan and P. Verma, “Performance comparison of hybrid recommender models in professional networking platforms,” *IEEE Transactions on Computational Systems*, vol. 11, no. 2, pp. 365-378, 2024.