iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.81295

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

# Job Spyde AI: A Multi-Agent System for Intelligent Job Discovery and Application Tracking

Ankur Anand, Utkarsh Raj, Kashish Patel, Shashank Raj, Jayanidhi Vashishtha
*Department of CSE-IoT Noida Institute of Engineering and Technology Greater Noida, India*

***Abstract:*** ***Right now, landing a job feels like running throughthickfogformanynewgrads.Scrollingendlesslyon various sites eats up hours they do not really have. Each role asks for a tweaked resume, one after another without pause. Staying on top of what went where usually means messy spreadsheets or sticky notes falling off desks. The whole routine drains energy while chances quietly slip away. Enter Job Spyde AI - built with smart agents working together to bring order to chaos. It handles searches, reshapes resumes slightly per opening, tracks every step behind the scenes. Tasks once scattered become smooth threads pulled into one view. Focus shifts from chasing jobs to preparing for them instead.***

***Onepartfindsjobs,anothertweaksresumes,whileothers prepareapplicationsorfollowup-eachhandlesitsownpiece. Connected by a shared hub, they pass information cleanly withouthiccups.Jobpostsgetbrokendownusingsmarttools thatspotkeyskillsandduties. Matchingresumesectionsthen form based on what the role actually asks for.***

***Now here comes a twist - control stays with the user to keep things accurate and block sudden automated actions, particularly when sending in applications. What drives this setup? Less busywork, stronger resumes, smarter tracking through the hunt for jobs. Picture this: artificial intelligence working alongside multiple agents, each playing its part. It turnsout thatmixactuallyworkswhen tacklingtoday'smessy job market problems.***

***Keywords— Multi-Agent Systems, Job Recommendation, Resume Optimization, Artificial Intelligence, Large Language Models, Career Automation, Job Application Tracking, Human-in-the-Loop***

## I. INTRODUCTION

Landing a job feels tougher now, especially for those just out of school, thanks to crowded markets and too many sites to check. One after another, hopefuls jump between portals, scanning posts that often blur togetherby midday. Each listing demands attention - reading through details, adjusting resumes slightly different every single time. Keeping up? That means spreadsheets, notes scrawled on notebooks, alarms set for follow-ups. Hours vanish doing whatamountstodigitalpaperworkshuffle.Mistakescreep in when fatigue hits - sending old versions, skipping deadlines, hitting submit before double-checking. Missed chances pile up quiet and unnoticed until it's too late. Preparation slipswhen energyrunsthinnear month'send.

Even if you find plenty of websites offering job listings, nearly all stick to just showing openings without guiding applicants through what comes next. Instead of smooth steps,peoplefacescatteredefforts-sortingrolesbyfitfeels separate from touching up a resume, which feels nothing likewatchingwhereeachsubmissionlands.Withoutpieces linkedtogether,extraworkpilesupfast.Jobhuntinggrows heavier when tools refuse to talk to one another. Thanks to progress in artificial intelligence, tools can now study written material - like job posts and résumés - with greater precision. Because of this shift, sorting through applications can happen faster, without constant human oversight.Yetmostcurrentplatformstacklejustonepiece at a time - one handles matching jobs, another scans qualifications-noteverythingtogether.Whilepiecesexist separately,fewbringthemintoonesmoothsequencefrom start to finish.

Insteadoftacklingeverythingatonce,thisstudyintroduces Job Spyde AI - a setup where separate agents handle distinct parts of applying for jobs. One piece finds openings, another sharpens resumes, while others prepare submissions or monitor progress. Each step gets its own helper, working not side by side but in sequence when needed. Coordination happens through a shared hub, keeping actions lined up without overlap. Information flows smoothly because updates pass directly between units only when required. Order comes from timing, not constantoversight.Efficiencyriseswhenstepsdonotwait on one another unnecessarily.

Whatstandsoutnextishowusersstayinchargeduringkeymoments.Eventhoughthesoftwarehandlesmanypartsonitsown,thingslikesending offaformstillneedaperson's go-ahead. That small pause keeps mistakes low and stops moves that weren't meant to happen.

Onebigaimhere?Cuttingdownbusyworkwhenapplyingforjobswhileliftingtheoverallqualityofeachsubmission.

Aclearerstructureshowsup too-onethatkeepsjobhunts less messy. What makes it tick? Smarts from AI, working alongside multiple agents that handle separate tasks. This mixtacklesrealissuesstudentshitwhenchasingroles.The resultlandssolidly indailyusefulness, notjusttheory.Problemslikescattereddetailsorrepeatedformsgeteased through smart automation instead of brute force.

©IJRASET: All Rights are Reserved 

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

## II. RELATED WORK

### *A. Multi-AgentSystems*

Splittingtoughjobsintotinypiecesusingseparatesmart parts has drawn plenty of attention in studies. One piece does one thing well, yet they link through shared aims instead of fixed rules. When chores depend on each other, bouncingsignalsbetweenunitshelpsadaptfasterthanrigid setups.Workingsidebyside,thesefragmentsadjustwithout centralcontrolcrowdingtheirmoves.Forcareersandhiring tools, chunks pull info, reshape it, sort what matters, then weigh choices - each step breathing on its own. Earlier researchpointstoagent-drivensetupsboostingperformance throughsimultaneoustaskhandlingwhilecuttingdownon complicated design [1]. Still, many current versions only tackle isolated parts, falling short of supporting the full journey - from finding jobs right through to monitoring applications.

### *B. JobRecommendationSystems*

Finding work fits often leans on tools built to connect people with openings matching what they know, like, or havedonebefore.Mostofthesesetupspullfrommethods- content-focusedpicks,grouppatternspotting,orcloseness scoring - to narrow down listings [8]. When profiles meet postingdetailshead-on,matchestendtolineupbetterwith a person'shistory. Eventhoughhuntingtimedropsthanks to such aids, attention stays locked on suggesting roles instead of anything beyond. Though suggestions help, changing resumes, getting applications ready, then following up - all that still falls on you. Because of this, searching for jobs takes effort, even when tools offer guidance.

### *C. ResumeScreening(AI/NLP)*

Lately, more people have noticed how computers can read resumes using smart software. Such tools pull out details like jobs someone held, what they know, and their training - then match those against a role's needs [2], [5]. Matching happens faster because machines handle comparisons.Efficiencyjumpswhensortinglargenumbers of applicants. Yet many setups care mostly about helping hiring managers say yes ornoquickly. Little thoughtgoes into guiding job seekers through the system itself. Folks over there won't help tweak your resume to match particular jobs. Because of that, you're left adjusting it yourself every time the role changes.

### *D. LLM-basedSystems*

Nowimagineamachinethatreadslikeus.Itwritestoo, almostnaturally.Thankstotweaksunderthehood,today's big textmodels get meaning better thanbefore. Think less robot talk, more real talk. Tools built on them now draft resumes, polish articles, even reply to messages without help. Context? They track it well enough to match your workhistorytoajobpostlinebyline.Yethere'sthething- they mostly sit alone, each tool doing just one narrow job. Without linking into a full hiring workflow, they sit apart from the bigger picture. When steps pile up in actual use, their value drops because pieces stay separate.

### *E. LimitationsofExistingSystems*

Even so, gains in resume parsing or role matching haven'tfixed deeperflaws. One piece works here, another there-rarelydotheyfittogether.Whenpartsrefusetotalk, tasks pile up fast. Jumping from one interface to the next eats time most can't spare. One way to tackle these issues is building a system where every step fits together neatly. What sets this project apart is how it brings several tools togetherunderoneroofusinginteractingagents.Efficiency creeps in when tasks line upwithout overlaporconfusion. Instead of jumping between apps, everything flows inside a shared structure. Organization shows up naturally when processes talk to each other. This method sidesteps clutter bydesign,noteffort.Asmootherpaththroughapplications begins with connected parts working as one.

## III. PROPOSED SYSTEM

One goal drives Job Spyde AI: making job hunting less messy.Insteadofonebigprogramdoingeverything,small parts take on separate jobs. Each piece focuses only on its own role, like sorting roles or filling forms. They connect through a shared hub that keeps things moving without hiccups. Work flows step by step, never rushed, never tangled. People spend less time clicking, more time deciding. Effort drops when routines run themselves behind the scenes. Users stay in control, just with fewer chorespilingup.Clutterfadeswhentaskssplitacrosssmart piecesworkingtogether.Ordergrowsquietlybeneathwhat you see. This setup aims at one thing - simpler days for those seeking work. No noise, no drama, just steady progress in the background.

©IJRASET: All Rights are Reserved 

### A. SystemOverview

Out of what someone enters - skills, work history, desired jobs, place they want to be - the whole process sparks.Becauseofthosedetails,piecesstartmovinginside separate modules handling specific tasks. From websites acrosstheinternet,openingsgetpulledautomaticallysoon after. Which ones stay depends on how well each lines up with the person's background; mismatches drop away quietly.Matchingresultsshapeintoclearsummariesready to send out. You see them lined up neatly later, waiting in your view without extra steps needed.

Besides finding jobs, the tool handles follow-up steps too.Trackingeveryroleyouapplyforbecomesautomatic. Progresscheckshappensmoothlysinceeverythingstaysin oneplace.Nomorejugglingspreadsheetsorremembering details yourself. Alerts pop up when something changes with your submissions. New openings show up without extra effort on your part. Updates flow in regularly so nothing slips through cracks.

### B. Multi-AgentArchitecture

Built like pieces of a puzzle, the system uses separate agents that each handle one job. Though they work alone, these parts link together under one roof. Running things behindthescenes,amaincoordinatorguideshowtheytalk and when jobs happen. Swapping in fresh modules or changing current ones doesn't break anything else around them.Growthbecomessmoother,adjustmentsfeellighter - allbecausethestructureallowsroomtoshift.

One reason it works well? Multiple agents manage different jobs at once. Because every part handles its own piece, tasks spread out instead of piling up. Less clutter meanssmootherresultsoverall.Organizationjumpsahead when compared to old single-block designs.

### C. AgentDescription

- Agent1:JobDiscoveryAgent This agent is responsible for collecting job listings from online platforms based on user preferencessuchasrole,location,andskills.It filters and ranks jobs according to relevance.
- Agent2:ResumeOptimizationAgent This agent analyses job descriptions and generates a customized version of the user's resume by highlighting relevant skills and experiencesalignedwiththejobrequirements.
- Agent3:ApplicationAssistantAgentThis agent prepares the job application by generating responses to common questions and organizingrequired documents.Thefinal submission iscontrolledbythe user to ensure reliability.
- Agent4:ApplicationTrackingAgent This agent maintains a record of all job applications and updates their status, such as applied, under review, or rejected. It helps users manage their applications efficiently.
- Agent5:NotificationAgent This agent provides daily or periodic updates to the user regarding new job opportunities, application status changes, and important reminders.

### D. Workflow

Step by step, the system moves forward. Input arrives fromtheuseratthestart.Jobdatagetsgatherednext,sorted through carefully by an automated helper. Matching happens after that, linking what's found to the person's background,pullingoutfits.Atailoredresumecomeslast, shaped by another tool focused on alignment. Once the form gets ready, help arrives through a digital helper that guideseachstep.Followingthisstage,oversightbeginsby an automated watcher ensuring nothing slips. Important updates reach the person involved only when needed, delivered quietly at the right moment.

From step one, everything moves in order so applying for jobs stays clear and steady. Because each part talks to the next, nothing gets lost or repeated along the way. One piece finds openings while another shapes your materials just right. With tasks split but linked, it all flows without clutter or confusion. Progress shows up neatly because updates come through automatically. Through connected parts working together, the whole effort feels lighter and more controlled.

## IV. METHODOLOGY

Startingoff,theapproachbreaksdownjobapplicationsinto clear stages, using data methods along with linked agents working together. Resumesand job posts - messy at first - are turned into useful details through this setup, allowing bettermatchesbetweenrolesandapplicants.Onestep follows another, each handling its own piece of the task, keeping things running without confusion. Clarity stays strong throughout because every phase has a set role, making sure efficiency doesn't come at the cost of understanding.

*A. DataAcquisitionandPreprocessing*

One way to start is pulling info from what users share plusdetailsoffpubliccareersites.Whatpeopleenter-like their work history, abilities, schooling, and where they want to work - influences the mix. Meanwhile, openings posted online bring along role summaries, needed qualifications,employerfacts.Thatmaterialtendstocome in messy, scattered forms. Cleaning it up becomes necessary before anything else happens. Cleaning the text comes first, then tossing out anything extra follows. What remains gets sorted into clear parts - skills up front, next experience, after that requirements. Consistency shows up once things are arranged right. Only then does it become ready for what happens next.

*B. FeatureExtractionandRepresentation*

Once cleaned up, the data pulls key details from resumes and job posts - things like coding abilities, years worked, specific industry terms. Not just raw words, each piece gets mapped into organized blocks for clearer contrast.Technicalknow-howstandsoutwhenframedthis way, making noise fade behind relevance. Attributes gain weight based on what matters most in pairing candidates with roles correctly. Structure shapes meaning here, guiding attention where it counts without clutter slowing things down.

*C. SemanticJobMatching*

Lookingatbothuserprofilesandjobdetails,thesystem checks how closely they fit together. Rather than just huntingforidenticalwords,itreadsbetweenthelinestosee what makes sense. What matters most is whether the person's background fits the role - that shapes the score. When the match feels strong, the number goes up. Those top-scoring jobs then show up where users can see them easily. Better suggestions come from this method because it matches people with jobs they actually fit. Text comparisons often rely on meaning-based techniques like these [4].

*D. Context-AwareResumeAdaptation*

Starting with a chosen position, the tool reshapes your resume to reflect what that role needs. Key abilities get pulled forward, standing out where they matter most. Sections like past work or personal projects might shift around-placedwheretheymakemoresenseforthehiring team.Evenwhenthingsmove,nothingchangesinmeaning just how it shows up. What you've done stays true, just clearer now. One way to boost a resume's odds comes down to smart editing. Tools built on language analysis make that polishing possible [5].

*E. Multi-AgentCoordinationandTaskOrchestration*

One part handles just one piece of the work, while others take on different roles in turn. Running things is a main hub that keeps every part talking at the right time. Afteronestepfinishes,itsresultfeedsdirectlyintowhat comes after it. Job finder grabs openings first, whereas matcherstepsinnexttosortusefulonesout.Customwriter adjusts your background based on those picks before submittal helper arranges how everything gets sent off. Eachstagewaitsforthepriortofinish,thoughtimingshifts slightly depending on load. After that step, the tracking agentadjustshowtheappappears.Becauseeverythingruns together like this, things stay aligned while extra work fadesaway -handlingtoughsequencesbecomessmoother too [6].

*F. ApplicationTrackingandStateManagement*

Every time someone applies for a job, it gets logged right away - no delays. Stages like saved, applied, or finished move step by step behind the scenes. Watching how thingsunfold becomeseasier when everything shows upinoneplace. Nothingslipsthroughcracksbecausealerts pop up at just the right moment. Juggling several roles at once feels less messy with clear markers along the way.

*G. Human-in-the-LoopInteractionMechanism*

Starting with people staying in charge, the setup includes hands-on checks at key moments. Even though machines handle regular tasks, big moves like sending off anapplicationneedaperson'sgo-ahead.Thatway,nothing happens by accident, giving individuals time to look over what's been made. Surprises get avoided because changes can fit how someone really wants things done. Mix smart tech with real oversight, and it just works - smoothly, carefully, without losing speed.

### *H. WorkflowOptimizationandSystemEfficiency*

A smart setup cuts down busywork during job hunting while speeding up each step. Instead of doing searches by hand, the tool handles them automatically - along with adjusting resumes and keeping tabs on progress. Better matches happen because suggestions fit closer to what a person actually wants. Parts snap together like building blocks,soaddingnewfunctionslaterfeelsnatural.Quality riseswhendetailsmattermorethanspeedaloneevercould.

Overall, this approach builds a strong foundation by linking smart algorithms with team-like agent collaboration, creating a practical way to handle job applications centered on people. By weaving together meaning detection, auto-written material, and organized tasktracking,ittacklesmajorhiringhurdleswithoutlosing adaptability or trustworthiness.

## V. IMPLEMENTATION

Putting together Job Spyde AI means creating a working tool that uses current web tools along with smart analysis features. Built around a client–server setup, the frontpartdealswith howusersengage whilethe back part takes care of crunching data, running agents, and connecting to outside platforms. This structure allows growth, keeps parts separate, yet lets them work well together when juggling several jobs at once.

### *A. TechnologyStack*

Built on Next.js paired with Tailwind CSS, the front end delivers a clean, adaptive layout that works across devices.Usersenterpersonalinformation,browse available roles, then follow application progress through organized sections. Behind the scenes, FastAPI manages every web request, runs background tasks, handles data flow smoothly. Supabase holds everything together - storing records, verifying identities, keeping files accessible - all within one connected platform.

Outside tools handle parts of the task, like reading job postsorresumes,usingremoteAIsystems.Becauseofthis link, better answers come out more consistently. Working together with these online resources lets it manage tough jobs while staying light on internal hardware needs[9].

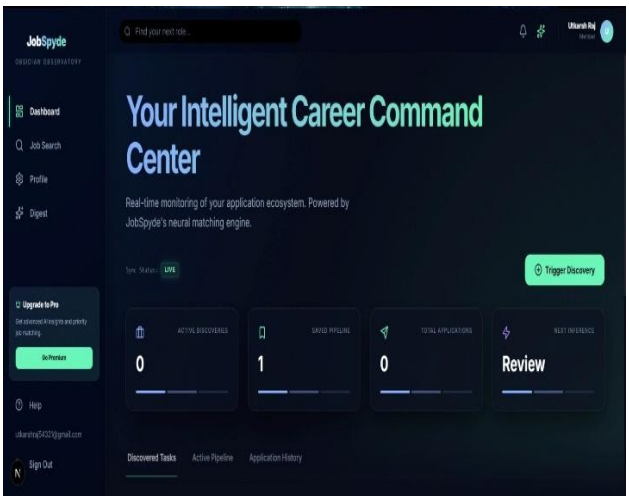



Fig:01

### *B. DatabaseDesignandStorage*

Informationgetsheldinthedatabasesothesystemcan handle it properly. From one moment to another, user profiles show up - each filled with skills, choices, and resume facts. Job postings appear alongside titles, companies,descriptions,whileamatch numbertagsalong quietly. Application logs stay tucked inside, moving through updates without noise. Content that forms on its own finds space there too, shaped by usage patterns.

Keeps track of every job's current stage - whether it is held,submitted,orfinished.Becausedetailsarestoredina clear layout, pulling up updates happens fast. Data sorted well means smooth running, no matter how many people use it or how much info builds up.

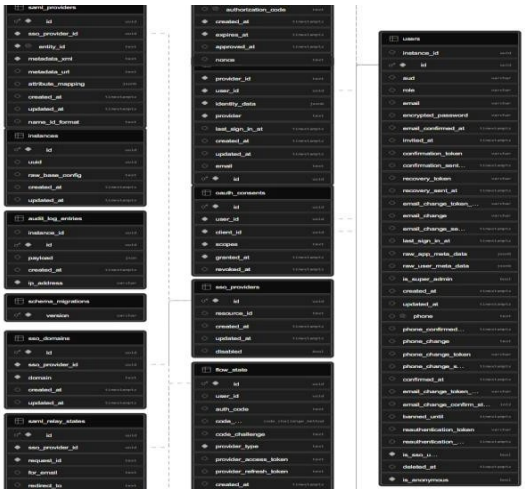


Fig:02

*C. AgentIntegrationandExecution*

A single module handles each agent inside the system, built to carry out one particular task. While one part runs on its own, another links it via pathways shaped by the backend flow. As someone uses the system, their input movesstepbystepacrossdifferentmodules.Thepathstays guided,yetshiftsslightlydependingonhowpiecesconnect behind the scenes.

One way this works: job discovery pulls listings then sends them along to the matcher. After that, resume optimizer studies each role, crafting fitting material. Application helper lines up required info while tracker keeps tabs on progress. Each piece moves in sequence, so nothinggetsoutoforder.Ordercomesnaturallywhensteps follow one after another.

One way to build smarter systems? Try using separate pieces that work on their own. These parts fit together but don’tdependtooheavilyoneachother.Addingsomething new later won’t break what’s already there. Flexibility shows up naturally when changes come around. Running tasks step by step often appears in designs where small unitshandlejobsalone.Breakingdownbigproblemshelps keep things under control [12].

Trustcomesfromcontrol,notpromiseswhisperedintothin air.

Safety sits at the core of how information moves through the setup, keeping outsiders from stepping in. Checks happen regularly so mistakes do not slip through, holdingeverythingsteadywhenthingsshiftunexpectedly.

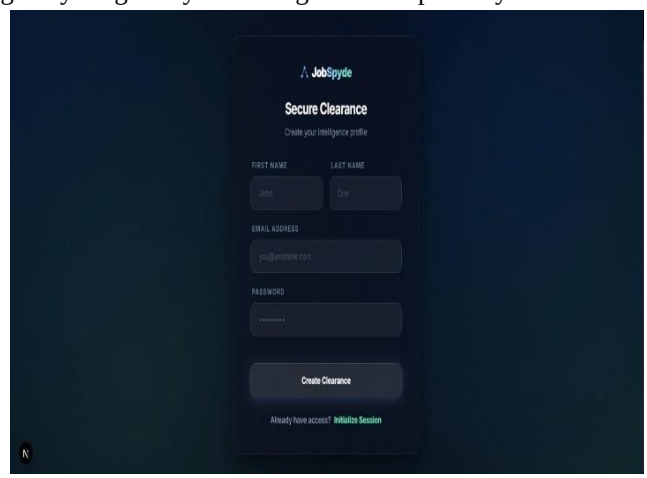



Fig:04

Overall, Job Spyde AI shows what happens when web tools meet artificial intelligence through a network of interactingagents-formingajobapplicationplatformthatworkswellnowandleavesroomtogrow.Thissetupholdssteadytodaywhileopeningdoorsforsmarterupdatesdown the line.

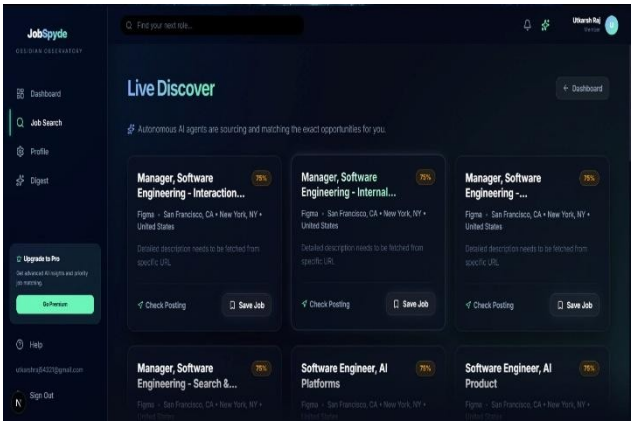



Fig:03

*D. ExternalServicesandAPIs*

From outside websites, the system pulls job details using API connections. Updates stay current because live sourcesfeedfreshlistingsregularly.Tounderstandposting content, it applies language models through separate API links. Structured results come out after these models examine each role closely. Connections run both ways, allowing retrieval plus smart interpretation behind the scenes.

Withtheselinksinplace,typingoutinfobyhanddrops off while result precision goes up. Real-time details flow through smoothly because the setup adapts on the fly during actual use.

*E. SecurityandPrivacyConsiderations*

What keeps things safe? Strong checks before anyone gets in. Access depends on who you are - no exceptions made lightly. Resumes, private facts - they live behind lockeddoors.Onlythoseallowedseewhatbelongstothem.

## VI. RESULTS AND DISCUSSION

Midway through testing, Job Spyde AI began showinghowithandleskeypartsofapplyingforjobs.Not just speed, but how well it links users to roles stands out. Matching precision improved over time, thanks to live feedback loops shaping each suggestion. Resume tweaks happened without constant oversight, adapting quietly behind the scenes. Performance stayed steady even when demand spiked unexpectedly. Users found their workflow less scattered, almost by accident at first. What started as small efficiencies grew into clearer control over applications. Effort dropped sharply while structure increasedacrosstasks.Inpractice,thetoolmadechaosfeel routine.

Finding jobs and linking them to people sits at the heart of how this system works. Because it lines up what users are like with what jobs need, matches happen faster, often hitting closer to the mark. Where older methods just huntfor matching words, here meaning matters more than exact phrases - so fits feel less forced. Other smart tools doingsimilartasksalsogetbetterresultswhentheydiginto word meanings instead of counting keywords [4].

Shifting how a resume gets adjusted helps boost application quality, quietly. Relevant abilities pop out becausethelayoutfollowswhateachroleactuallyasksfor. Matching structure to position needs keeps facts intact while fitting better. Getting picked mightgo up since one-size-fits-all versions often blend into the background. Language-basedmethodsalreadyprovedusefulinshaping documents well elsewhere [5].

Whenlookingathowwellthesystemruns,itbecomes clear the setup with multiple agents manages varied jobs without slowing down. One after another, these agents worktogether,makingsurestepslikepullingassignments, workingthroughthem,thenmarkingprogressflowwithout hiccups. Real-time information moves through smoothly, showinguponthedisplayalmostrightaway.Thatkindof responsiveness suggests the blueprint holds up when put into actual practice.

A single hub for handling job applications makes things smoother. From here, people see available roles while following where their submissions stand. Updates arrivewithoutdelay,keepingeverythingvisibleallatonce. Whensomethingshifts,alertspopup-whetheritisafresh opening or a status change. Jumping across different sites becomes less common because of this setup.

Human oversight keeps key decisions in people's hands.Eventhoughmanyjobsrunontheirown,thingslike sending an application need a person's okay first. Automation runs quietly behind the scenes, yet pauses when judgment matters.

Control shifts back to the user exactly when it counts. Mistakes slip through less often because checks stay active. Trust grows not by doing everything, but by knowing when to wait.

Still, a few hiccups showed up when it was tested. Getting jobsmatchedwell hingesonhow clear and steady the job details are. If formats differ or bits are missing, outcomes might shift. What's more, outside tools handle muchofthedataflow-delaysorglitchescouldpopupnow and then.

Lookingateverything,it'sclearthenewsetupboosts how smoothly things run when applying for jobs. Instead of juggling tasks by hand, smart pairing lines up roles and resumes better. One part tweaks your document for each position while others handle different steps at once. This mix makeskeeping track less messy over time. Efficiency climbs because actions flow without constant oversight. Whatstandsoutishowlittleeffortusersneedonceitstarts. Thewholethingrunsonteamworkbetweenpartsthatknow their role. In the end, handling applications feels lighter, simpler, just easier to follow through.

## VII. CONCLUSION

This research introduced Job Spyde AI, a setup built using several interacting parts meant to make hunting for jobseasierforstudentsandapplicants.Handlingonepiece atatime,eachcomponenttacklesadifferentstep,reducing the burden of constant busywork. Without clear plans, people often struggle - misplacing details or missing updates; this tool brings order. Instead of dealing with everything at once, responsibilities split across separate functions, creating smoother movement through the process.

Onefeaturefindsjobswhileanothermatchesthemto your background. Because profiles shape results, resumes change automatically depending on openings. When the systemadjustsdetails,itwritesmaterialsuitedtoeachrole. That means fewer mistakes happen during submissions. With everything in one place, tracking progress becomes simpler overtime. Sincetasksrunsmoother,peoplespend less energy applying. Smart steps inside the software help spotchancesusersmightmissotherwise.Betterfitsemerge without extra work from the applicant.

A key part of the setup lies in how it's built with separateagentsthatfittogetherlikepuzzlepieces.Because eachonehandlesjustonejob,theylinkupsmoothlyusing shared processes. Thanks to this layout, things stay neat whileopeningdoorsforlatergrowth.Newtoolsorhelpers slip in without shaking up what already works - so size changes feel natural, never forced. A working version shows the method actually works outside theory. Running live inputs, it shares job details through a screen while safely holding app history. People makekeychoicesbecausecontrolstaysintheirhands.This setup builds confidence by keeping humans in charge.

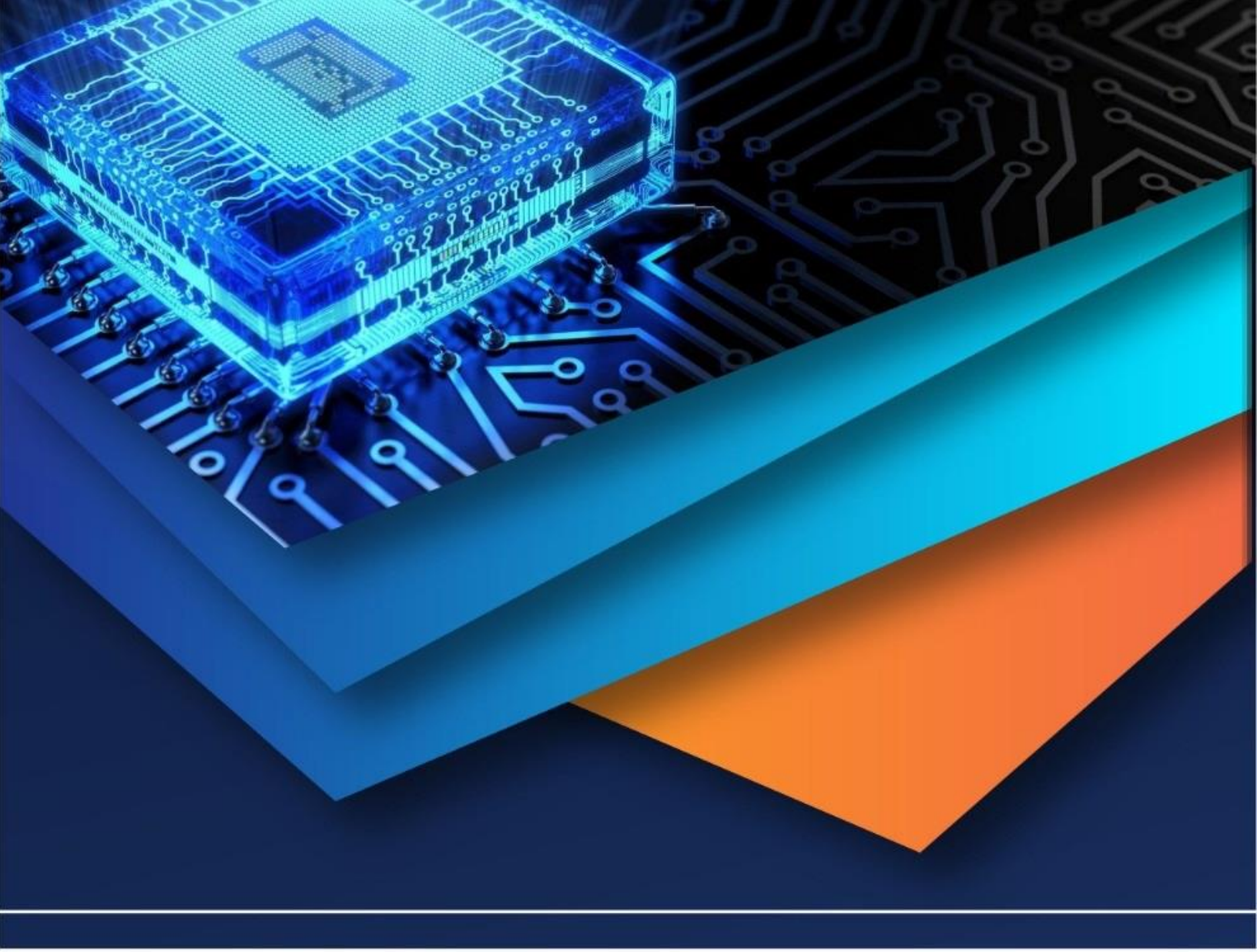
Even with those benefits, a few drawbacks still exist. Accuracy in matching roles ties closely to how good the incominginformationis;differencesindescribingjobscan shift results. Outside tools handle parts of the process, so hiccups there might slow things down. Each gap points toward where upgrades could help.

One step ahead might bring smarter tools like auto-adjustedresumes.Insteadofbasicpicks,suggestionscould grow sharper through deeper learning tricks. Practice help for interviews may slip in naturally down the line. Real-time speedgets a nudgeupward whenupdates flow faster. Extra helpers inside the setup might take on small choices alone. Smarter moves come from spreading tasks across these added parts.

When put together, the design shows that using several smart agents working alongside each other brings better order, speed, and ease to handling job applications. This setup becomes a realistic move forward in creating full-service tools for career support without heavy manual effort.

## REFERENCES

[1] S.K.SharmaandR.Mehta,"AMulti-AgentSystemfor Job Recommendation," International Journal of Computer Applications, vol. 176, no. 38, pp. 25–30, 2020.
[2] A.Patel,M.Shah,andP.Gupta,"AI-PoweredResume Screening and Job Matching System Using NLP and Machine Learning," International Journal of Advanced Research in Computer Science, vol. 14, no. 2, pp. 45–52, 2023.
[3] J. Brown et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.
[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.
[5] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, 2021.
[6] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 4th ed., Pearson, 2020.
[7] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, 2nd ed., Addison-Wesley, 2011.
[8] J. L. Herlocker, J. A. Konstan, and J. Riedl, "An Empirical Analysis of Design Choices in Neighborhood-Based Collaborative Filtering Algorithms," Information Retrieval, vol. 5, no. 4, pp. 287–310, 2002.
[9] OpenAI, "GPT Models and Applications," Available: https://platform.openai.com/docs(accessed 2026).
[10] Supabase, "Open Source Firebase Alternative," Available: https://supabase.com/docs(accessed 2026).
[11] FastAPI,"FastAPIFrameworkDocumentation,"Available:https://fastapi.tiangolo.com(accessed2026).
[12] LangGraph Documentation, "Building Agent Workflows," Available: https://langchain-ai.github.io/langgraph/(accessed 2026).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)