



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

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 through thick fog for many new grads. Scrolling endlessly on 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.

One part finds jobs, another tweaks resumes, while others prepare applications or follow up - each handles its own piece. Connected by a shared hub, they pass information cleanly without hiccups. Job posts get broken down using smart tools that spot key skills and duties. Matching resume sections then 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 turns out that mix actually works when tackling today's messy 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 together by 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 what amounts to digital paperwork shuffle. Mistakes creep 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 slips when energy runs thin near month's end.

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, people face scattered efforts - sorting roles by fit feels separate from touching up a resume, which feels nothing like watching where each submission lands. Without pieces linked together, extra work piles up fast. Job hunting grows 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. Yet most current platform stacks are just one piece at a time - one handles matching jobs, another scans qualifications - not everything together. While pieces exist separately, few bring them into one smooth sequence from start to finish.

Instead of tackling everything at once, this study introduces 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 constant oversight. Efficiency rises when steps don't wait on one another unnecessarily.

What stands out next is how users stay in charge during key moments. Even though the software handles many parts on its own, things like sending off a form still need a person's go-ahead. That small pause keeps mistakes low and stops moves that weren't meant to happen.

One big aim here? Cutting down busywork when applying for jobs while lifting the overall quality of each submission.

A clearer structure shows up too - one that keeps job hunts less messy. What makes it tick? Smarts from AI, working alongside multiple agents that handle separate tasks. This mix tackles real issues students hit when chasing roles. The result lands solidly in daily usefulness, not just theory. Problems like scattered details or repeated forms get eased through smart automation instead of brute force.

II. RELATED WORK

A. Multi-Agent Systems

Splitting tough jobs into tiny pieces using separate smart 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, bouncing signals between units helps adapt faster than rigid setups. Working side by side, these fragments adjust without central control crowding their moves. For careers and hiring tools, chunks pull info, reshape it, sort what matters, then weigh choices - each step breathing on its own. Earlier research points to agent-driven setups boosting performance through simultaneous task handling while cutting down on 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. Job Recommendation Systems

Finding work fits often leans on tools built to connect people with openings matching what they know, like, or have done before. Most of these setups pull from methods - content-focused picks, group pattern spotting, or closeness scoring - to narrow down listings [8]. When profiles meet posting details head-on, matches tend to line up better with a person's history. Even though hunting time drops thanks 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. Resume Screening (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. Efficiency jumps when sorting large numbers of applicants. Yet many setups care mostly about helping hiring managers say yes or no quickly. Little thought goes 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-based Systems

Now imagine a machine that reads like us. It writes too, almost naturally. Thanks to tweaks under the hood, today's big text models get meaning better than before. 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 work history to a job post line by line. Yet there's something - 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. Limitations of Existing Systems

Even so, gains in resume parsing or role matching haven't fixed deeper flaws. One piece works here, another there - rarely do they fit together. When parts refuse to talk, 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 together under one roof using interacting agents. Efficiency creeps in when tasks line up without overlap or confusion. 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 by design, not effort. A smoother path through applications begins with connected parts working as one.

III. PROPOSED SYSTEM

One goal drives Job Spyde AI: making job hunting less messy. Instead of one big program doing everything, 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 chores piling up. Clutter fades when tasks split across smart pieces working together. Order grows quietly beneath what you see. This setup aims at one thing - simpler days for those seeking work. No noise, no drama, just steady progress in the background.

A. System Overview

Out of what someone enters - skills, work history, desired jobs, place they want to be - the whole process sparks. Because of those details, pieces start moving inside separate modules handling specific tasks. From websites across the internet, openings get pulled automatically soon after. Which ones stay depends on how well each lines up with the person's background; mismatches drop away quietly. Matching results shape into clear summaries ready 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. Tracking every role you apply for becomes automatic. Progress checks happen smoothly since everything stays in one place. No more jugglings, spreadsheets, or remembering 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-Agent Architecture

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 behind the scenes, a main coordinator guides how they talk and when jobs happen. Swapping in fresh modules or changing current ones doesn't break anything else around them. Growth becomes smoother, adjustments feel lighter - all because the structure allows room to shift.

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 means smoother results overall. Organization jumps ahead when compared to old single-block designs.

C. Agent Description

- **Agent 1: Job Discovery Agent** This agent is responsible for collecting job listings from online platforms based on user preferences such as role, location, and skills. It filters and ranks jobs according to relevance.
- **Agent 2: Resume Optimization Agent** This agent analyses job descriptions and generates a customized version of the user's resume by highlighting relevant skills and experiences aligned with the job requirements.
- **Agent 3: Application Assistant Agent** This agent prepares the job application by generating responses to common questions and organizing required documents. The final submission is controlled by the user to ensure reliability.
- **Agent 4: Application Tracking Agent** 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.
- **Agent 5: Notification Agent** 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 from the user at the start. Job data gets gathered next, sorted through carefully by an automated helper. Matching happens after that, linking what's found to the person's background, pulling out fits. A tailored resume comes last, shaped by another tool focused on alignment. Once the form gets ready, help arrives through a digital helper that guides each step. Following this stage, oversight begins by 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

Starting off, the approach breaks down job applications into clear stages, using data methods along with linked agents working together. Resumes and job posts - messy at first - are turned into useful details through this setup, allowing better matches between roles and applicants. One step 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. Data Acquisition and Preprocessing

One way to start is pulling info from what users share plus detail off public careers sites. What people enter - 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, employer facts. That material tends to come 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. Feature Extraction and Representation

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. Technical know-how stands out when framed this 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. Semantic Job Matching

Looking at both user profiles and job details, the system checks how closely they fit together. Rather than just hunting for identical words, it reads between the lines to see 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-Aware Resume Adaptation

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 - placed where they make more sense for the hiring team. Even when things move, nothing changes in meaning - 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-Agent Coordination and Task Orchestration

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. After one step finishes, its result feeds directly into what comes after it. Job finder grabs openings first, whereas matcher steps in next to sort useful ones out. Custom writer adjusts your background based on those picks before a submission helper arranges how everything gets sent off. Each stage waits for the prior to finish, though timing shifts slightly depending on load. After that step, the tracking agent adjusts how the app appears. Because everything runs together like this, things stay aligned while extra work fades away - handling tough sequences becomes smoother too [6].

F. Application Tracking and State Management

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 things unfold becomes easier when everything shows up in one place. Nothing slips through cracks because alerts 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-Loop Interaction Mechanism

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 an application need a person's go-ahead. That way, 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. Workflow Optimization and System Efficiency

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, so adding new functions later feels natural. Quality rises when details matter more than speed alone ever could.

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 task tracking, it tackles major hiring hurdles without losing 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 front part deals with how users engage while the 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. Technology Stack

Built on Next.js paired with Tailwind CSS, the front end delivers a clean, adaptive layout that works across devices. Users enter personal information, 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 posts or resumes, using remote AI systems. Because of this 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].

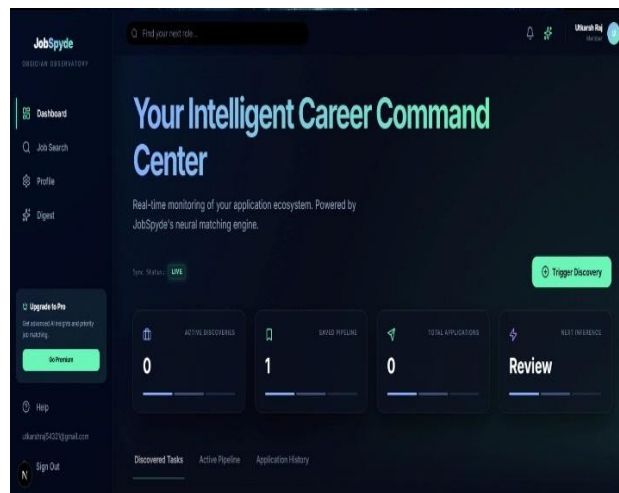


Fig:01

B. Database Design and Storage

Information gets held in the databases so the system can 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, while a match number tags along 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, or finished. Because details are stored in a 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.

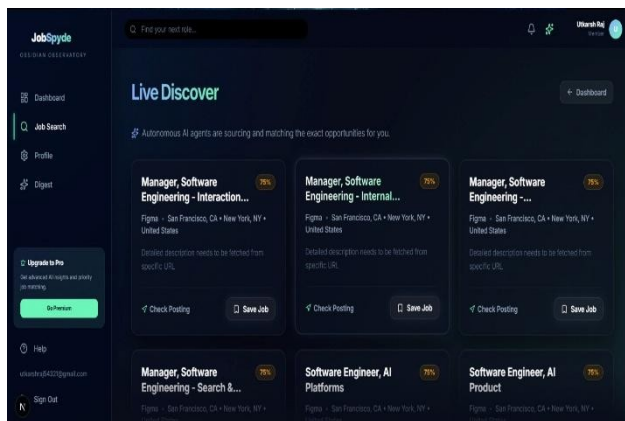


Fig:03

D. External Services and APIs

From outside websites, the system pulls job details using API connections. Updates stay current because live sources feed fresh listings regularly. To understand posting 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.

With these links in place, typing out info by hand drops off while result precision goes up. Real-time details flow through smoothly because the setup adapts on the fly during actual use.

E. Security and Privacy Considerations

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 locked doors. Only those allowed see what belongs to them.

VI. RESULTS AND DISCUSSION

Midway through testing, Job Spyde AI began showing how it handles key parts of applying for jobs. 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 increased across tasks. In practice, the tool made chaos feel 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 hunt for matching words, here meaning matters more than exact phrases - so fits feel less forced. Other smart tools do similar tasks also get better results when they dig into word meanings instead of counting keywords [4].

Shifting how a resume gets adjusted helps boost application quality, quietly. Relevant abilities pop out because the layout follows what each role actually asks for. Matching structure to position needs keeps facts intact while fitting better. Getting picked might go up since one-size-fits-all versions often blend into the background. Language-based methods already proved useful in shaping documents well elsewhere [5].

When looking at how well the system runs, it becomes clear the setup with multiple agents manages varied jobs without slowing down. One after another, these agents work together, making sure steps like pulling assignments, working through them, then marking progress flow without hiccups. Real-time information moves through smoothly, showing up on the display almost right away. That kind of 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 arrive without delay, keeping everything visible all at once. When something shifts, alerts pop up - whether it's a fresh 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. Even though many jobs run on their own, things like 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 jobs matched well hinges on how clear and steady the job details are. If formats differ or bits are missing, outcomes might shift. What's more, outside tools handle much of the data flow - delays or glitches could pop up now and then.

Looking at everything, it's clear the new setup boosts 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 makes keeping track less messy over time. Efficiency climbs because actions flow without constant oversight. What stands out is how little effort users need once it starts. The whole thing runs on teamwork between parts that know 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 jobs easier for students and applicants. Handling one piece at a time, each component tackles a different step, 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.

One feature finds jobs while another matches them to your background. Because profiles shape results, resumes change automatically depending on openings. When the system adjusts details, it writes materials suited to each role. That means fewer mistakes happen during submissions. With everything in one place, tracking progress becomes simpler over time. Since tasks run smoother, people spend less energy applying. Smart steps inside the software help spot chances users might miss otherwise. Better fit emerges without extra work from the applicant.

A key part of the setup lies in how it's built with separate agents that fit together like puzzle pieces. Because each one handles just one job, they link up smoothly using shared processes. Thanks to this layout, things stay neat while opening doors for later growth. New tools or helpers 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 make key choices because control stays in their hands. 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 incoming information is; differences in describing jobs can 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-adjusted resumes. Instead of basic picks, suggestions could grow sharper through deeper learning tricks. Practice help for interviews may slip in naturally down the line. Real-time speed gets a nudge upward when updates 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.Sharma and R.Mehta, "A Multi-Agent System for Job Recommendation," International Journal of Computer Applications, vol. 176, no. 38, pp. 25–30, 2020.
- [2] A.Patel, M.Shah, and P.Gupta, "AI-Powered Resume 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, "FastAPI Framework Documentation," Available: <https://fastapi.tiangolo.com> (accessed 2026).
- [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)