IJIRSET
International Journal of Innovative Research in Science | Engineering | Technology

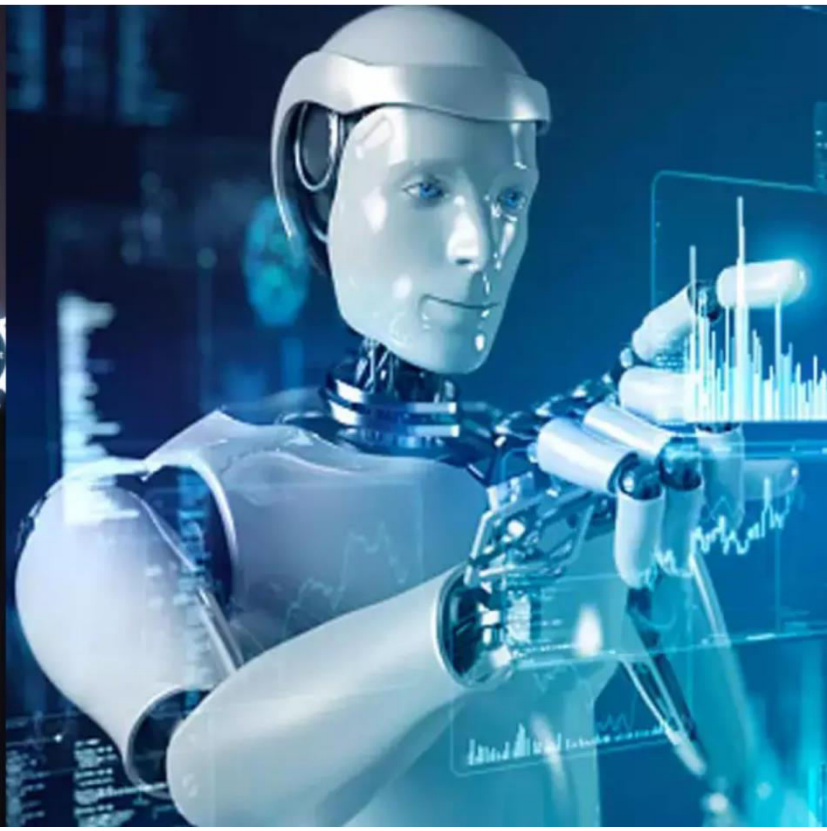
ISSN(O): 2319-8753
ISSN(P): 2347-6710



International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)





Impact Factor: 9.935

Volume 15, Issue 11, November 2026

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511044|

AI-Based Multi-Agent Business Automation System using Agentic AI: A Review

Swapnil Bhapkar, Dahale Samarth, Patil Samarth,Satpute Rohit, Anawade.A.V.

U.G. Student, Department of AI&DS Engineering, VACO Engineering College, A.Nagar, Maharashtra, India

U.G. Student, Department of AI&DS Engineering, VACO Engineering College, A.Nagar, Maharashtra, India

U.G. Student, Department of AI&DS Engineering, VACO Engineering College, A.Nagar, Maharashtra, India

U.G. Student, Department of AI&DS Engineering, VACO Engineering College, A.Nagar, Maharashtra, India

Associate Professor, Department of AI&DS Engineering, VACO Engineering College, A.Nagar, Maharashtra, India

ABSTRACT: Routine business tasks such as resume screening, expense analysis and lead management are still handled manually and by separate departments. Rule-based automation can execute fixed steps but cannot understand a natural-language request or divide a complex request among different functions. This paper reviews Agentic AI and multi-agent systems and presents the design of an AI-based multi-agent business automation system. In the proposed system, a Manager Agent interprets a request submitted through a web dashboard, divides it into subtasks and delegates them to specialized HR, Finance and Sales Agents. The agents use a Large Language Model (LLM) through LangChain, supported by Retrieval-Augmented Generation (RAG), memory and tool calling. The Manager Agent combines the outputs into one result, which is finalized only after human approval. The system is designed with React.js, FastAPI, PostgreSQL and JWT authentication, and generates a PDF report for every approved task. Eight papers are surveyed, a research gap is identified, and the architecture, algorithm and expected outcomes are discussed. The system is expected to reduce manual coordination effort and to give small and medium enterprises (SMEs) controlled and auditable assistance.

KEYWORDS: Agentic AI; Multi-Agent System; Large Language Model; Retrieval-Augmented Generation; Business Automation; Human-in-the-Loop

I. INTRODUCTION

Every organization depends on a steady flow of routine work, such as screening resumes, checking expenses against a budget and following up on sales leads. These tasks are repetitive and are usually handled by different departments, each with its own tools. When one business goal involves several departments, for example planning a new hire within the budget and the expected revenue, a person has to split the goal, ask each department and combine the answers manually.

Recent progress in Large Language Models (LLMs) allows software to understand natural-language instructions and to reason about them. Agentic AI goes one step further: an agent can plan, use tools, remember earlier information and work towards a goal with limited human intervention, and several specialized agents can cooperate by dividing a larger task into smaller subtasks [1]. This paper reviews the literature on these ideas and applies them to everyday business work through an AI-based multi-agent business automation system, in which sensitive actions always remain under human control.

II. LITERATURE SURVEY

Eight papers on Agentic AI, multi-agent systems and LLM-based agents are reviewed. The first two are the base papers of the work; a summary of all eight is given in Table 1.

A. Review of Base Papers

The concept of Agentic AI, in which software achieves complex goals with minimal human intervention, is presented in Ref. [1]. Multi-agent systems are explained, where specialized agents collaborate by dividing a larger task into

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511044|

subtasks, and key concepts such as planning, reasoning, tool use, memory and human-in-the-loop control are discussed. This gives the conceptual foundation for the Manager Agent and the specialized agents. An LLM-based multi-agent workflow that automates a complex security task, vulnerability detection and remediation in Infrastructure-as-Code, is proposed in Ref. [2]. Specialized agents handle retrieval, detection and report generation, and an overall workflow coordinates them. The work shows that modular agents improve coordination, scalability and maintainability.

B. Review of Other Papers

Agent-based programming for multi-agent systems is reviewed in Ref. [3], which guides the design of the agents and of the communication between them. Agent Q, which combines reasoning and learning to improve autonomous agents on multi-step tasks, is proposed in Ref. [4]. Goal-oriented semantic communication, in which information is exchanged according to the goal of the task, is surveyed in Ref. [5]; this matches the passing of only the relevant subtask to each agent. AgentTuning, which enables generalized agent abilities in LLMs, is presented in Ref. [6] and helps in choosing and prompting an LLM for each agent role. Deep learning, reinforcement learning and world models are discussed in Ref. [7] as background on adaptive behaviour. It is argued in Ref. [8] that agents must be judged on accuracy, cost and reliability together, and this view is followed for evaluation..

The work in this paper is divided in two stages. 1) Text- Detection 2) Inpainting. Text detection is done by applying morphological open-close and close-open filters and combines the images. Thereafter, gradient is applied to detect the edges followed by thresholding and morphological dilation, erosion operation. Then, connected component labelling is performed to label each object separately. Finally, the set of selection criteria is applied to filter out non text regions. After text detection, text inpainting is accomplished by using exemplar based Inpainting algorithm.

Table 1 Summary of literature survey

Ref.	Focus / contribution	Relevance to the proposed system
[1]	Agentic AI for complex goals: planning, reasoning, tool use, memory, human-in-the-loop (IEEE Access, 2025)	Foundation for the Manager Agent and specialized agents
[2]	Modular LLM agents for retrieval, detection and report generation in Infrastructure-as-Code (IEEE Access, 2025)	Workflow-based coordination of agents
[3]	Programming approaches for multi-agent systems (Computers, 2021)	Agent design and inter-agent communication
[4]	Reasoning and learning for autonomous agents (arXiv, 2024)	Improving agent reasoning on multi-step tasks
[5]	Goal-oriented exchange of information (IEEE Access, 2024)	Goal-driven task understanding and exchange
[6]	Generalized agent abilities in LLMs (arXiv, 2023)	Selecting and prompting LLMs for agent roles
[7]	Learning and world models for decision-making (Neural Networks, 2022)	Background on adaptive, learning-based behaviour
[8]	Evaluating agents on accuracy, cost and reliability (arXiv, 2024)	Guide for evaluating the proposed system

C. Research Gap

The surveyed work establishes the concepts of Agentic AI, multi-agent coordination and agent evaluation, but most of it is conceptual or addresses a technical domain such as security. Few works put these ideas into one practical business application for SMEs that combines a Manager Agent, HR, Finance and Sales Agents, RAG, memory, tool calling, human approval and report generation behind a simple web dashboard. The proposed system aims to fill this gap.

III. METHODOLOGY

The proposed system is a local multi-agent business automation platform with a layered architecture, shown in Figure 1. The presentation layer is a React.js and Tailwind CSS dashboard, where the user logs in, submits tasks, approves or

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511044|

rejects results and views logs, history and reports. The application layer is a FastAPI backend that exposes REST APIs, checks JWT authentication and contains a simple Python orchestration layer for communication between agents. The agent layer contains the Manager Agent and the HR, Finance and Sales Agents, which analyze subtasks using an LLM (OpenAI or Gemini) through LangChain, with RAG for business context, memory for continuity and tools for specific operations. The data layer is PostgreSQL, which stores users, tasks, results, approvals, logs and history, together with a vector store for the documents used in RAG.

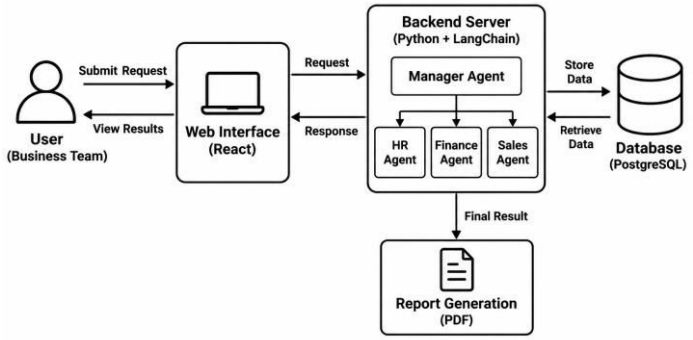


Figure 1 Proposed system architecture

The flow of a request is described by the following algorithm, which is executed for every business request.
Step 1: The user registers or logs in, and the backend verifies the credentials and issues a JWT token.
Step 2: The user submits a request in natural language, and it is stored as a task with the status "Pending".
Step 3: The Manager Agent interprets the request with the LLM and creates a plan, that is, a list of subtasks with the responsible agent for each.
Step 4: Each subtask is sent to the assigned agent, which retrieves context with RAG, reads its memory, calls the required tools and asks the LLM to analyze the subtask.
Step 5: The outputs are returned to the Manager Agent and logged, and are aggregated into one structured result with the status "Awaiting approval".
Step 6: The result is shown on the dashboard. If it is approved, it is stored in PostgreSQL and a PDF report is generated. If it is rejected, the feedback is passed to the Manager Agent, which re-plans from Step 3.

The design follows an object-oriented approach. An abstract BaseAgent class defines the properties and the analyze(subtask) operation shared by all agents, and the HRAgent, FinanceAgent and SalesAgent classes extend it. The ManagerAgent class handles planning, delegation, aggregation and approval requests, and the User, Task and Report classes model the business entities. The Waterfall model of the software development life cycle is followed, because the requirements are clearly defined at the start.

IV. EXPERIMENTAL RESULTS

Businesses handle repetitive tasks such as resume screening, expense analysis and lead management manually. Traditional automation has limited reasoning and flexibility, and a single request may need inputs from HR, Finance and Sales without an efficient way to divide, coordinate and combine them. A system that can understand, divide and coordinate complex tasks among specialized agents, and generate a single meaningful result with human approval, is therefore needed.

Existing business automation is mostly rule-based and built on predefined workflows. The main disadvantages are listed below.

- Fixed workflows cannot adapt to new or differently worded requests, and natural-language understanding is limited.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511044|

- Separate departmental systems keep information isolated, and there is no intelligent delegation of work between them.
- Considerable human effort is needed for coordination and for combining results, which makes the process slow, error-prone and hard to scale.
- Results are rarely stored together with logs and history, which makes auditing difficult.

V. REQUIREMENTS AND TECHNOLOGIES

The system is a web application that runs locally on Windows 10/11 or Ubuntu Linux, and an internet connection is needed to call the cloud-based LLM API. The technologies and requirements are summarized below.

- **Software:** React.js and Tailwind CSS (frontend); Python 3.x and FastAPI (backend); LangChain with the OpenAI / Gemini API (agents); PostgreSQL with a vector store (database); JWT (authentication); a Python library for PDF reports.
- **Hardware:** Intel Core i5/i7 processor (2.4 GHz or above), 8 GB RAM or more and 128 GB storage, with an SSD recommended. No sensors are required, and a GPU is optional.
- **Security:** Hashed passwords, JWT-based access control, server-side storage of API keys, logging of all task activity and explicit human approval of sensitive actions.

VI. EXPECTED OUTCOMES, ADVANTAGES AND APPLICATIONS

The work reported here covers Stage I, that is, the study, requirement analysis and design, while the implementation and testing are planned for the next stage. Experimental results are therefore not yet available, and the expected outcomes are given instead. A functional local multi-agent system is expected to analyze and delegate tasks automatically, assist with HR, Finance and Sales tasks and integrate an LLM, RAG, memory and tools, with human-controlled execution, logs and task history.

The main advantages are the intelligent understanding of natural-language tasks, collaboration between agents through delegation and aggregation, controlled execution with auditable logs and PDF reports, and a modular design in which new agents can be added without changing the other components. The system is applicable to resume screening and shortlisting (HR), expense analysis and budget checks (Finance), lead management and revenue forecasting (Sales), combined planning tasks such as a hiring and budget plan, and intelligent assistance for SMEs and startups.

VII. CONCLUSION AND FUTURE SCOPE

This paper reviewed eight papers on Agentic AI and multi-agent systems and presented an AI-based multi-agent business automation system in which a Manager Agent plans and delegates, and specialized HR, Finance and Sales Agents use an LLM, RAG, memory and tool calling to analyze subtasks. Human approval keeps sensitive operations under control, while logs, history and PDF reports make the system auditable. The agents, the dashboard and the backend will be implemented and tested in the next stage.

As the system is modular, more agents (for example Marketing, Legal or Customer Support) and connectors to real business tools such as email, accounting software, CRM and HRMS can be added. Multilingual and voice-based task submission, role-based approval workflows, local or open-source LLMs for data privacy and learning from the approval history are other possible extensions.

ACKNOWLEDGEMENT

The authors sincerely thank the guide for constant support and guidance, the researchers and publishers whose work was reviewed, and the college authorities for providing the necessary infrastructure and support.

REFERENCES

[1] "Agentic AI: Autonomous Intelligence for Complex Goals," IEEE Access, 2025.
[2] "LLM Agentic Workflow for Automated Vulnerability Detection and Remediation in Infrastructure-as-Code," IEEE Access, 2025.

International Journal of Innovative Research of Science, Engineering and Technology (IJIRSET)

Impact Factor 9.935

|www.ijirset.com |A Monthly, Peer Reviewed & Refereed Journal| e-ISSN: 2319-8753| p-ISSN: 2347-6710|

Volume 15, Issue 11, November 2026

|DOI: 10.15680/IJIRSET.2026.1511044|

[3] R. C. Cardoso and A. Ferrando, "A Review of Agent-Based Programming for Multi-Agent Systems," Computers, vol. 10, no. 2, p. 16, 2021.
[4] P. Putta et al., "Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents," arXiv:2408.07199, 2024.
[5] T. M. Getu, G. Kaddoum and M. Bennis, "A Survey on Goal-Oriented Semantic Communication: Techniques, Challenges, and Future Directions," IEEE Access, vol. 12, pp. 51223-51274, 2024.
[6] A. Zeng et al., "AgentTuning: Enabling Generalized Agent Abilities for LLMs," arXiv:2310.12823, 2023.
[7] Y. Matsuo et al., "Deep Learning, Reinforcement Learning, and World Models," Neural Networks, vol. 152, pp. 267-275, 2022.
[8] S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir and A. Narayanan, "AI Agents That Matter," arXiv:2407.01502, 2024.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details