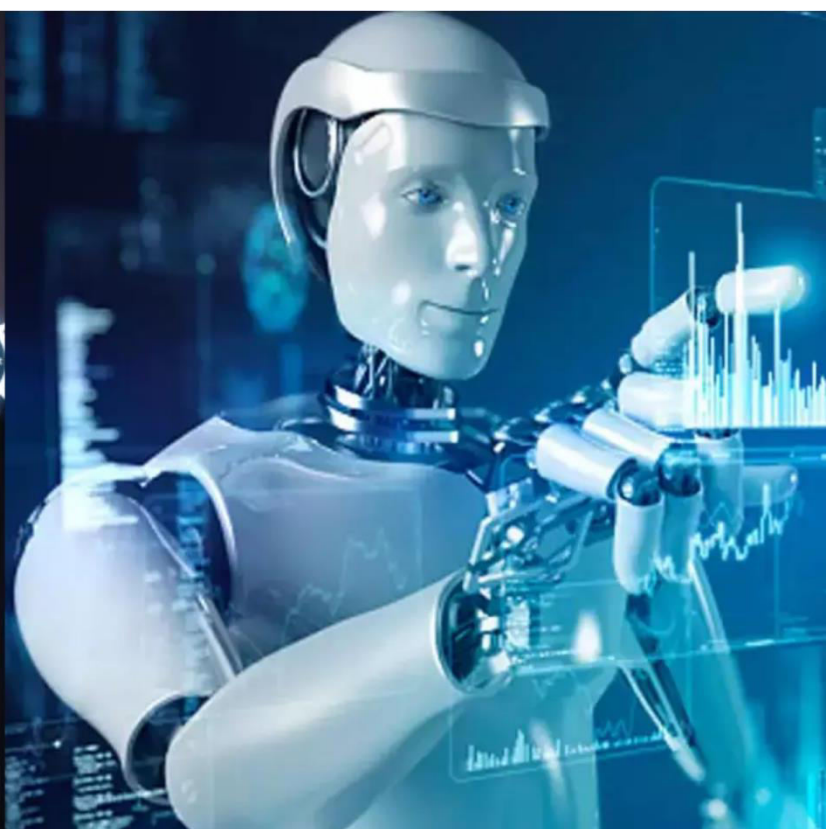




International Journal of Innovative Research in Science Engineering and Technology (IJIRSET)

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



Impact Factor: 9.935

Volume 15, Issue 10, October 2026

AI-Powered Smart Task Management System

Dharanika K, Janani N, Vigneshwaran R

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai,
Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai,
Tamil Nadu, India

Assistant Professor, Department of Computer Science and Engineering, K.L.N. College of Engineering, Sivagangai,
Tamil Nadu, India

ABSTRACT: Effective project planning and task coordination remain significant challenges in collaborative software development, particularly during project initiation, when identifying and organizing development tasks is a time-consuming manual process. This paper presents the AI-Powered Smart Task Management System, a web-based application that integrates a Large Language Model (LLM) with a collaborative project workspace. From a project description supplied by the Team Leader, an AI module generates structured tasks with descriptions and priorities, which the Team Leader reviews and assigns. A real-time Kanban board, synchronized across connected clients using Socket.IO, supports task tracking across the To Do, In Progress and Done stages. In addition, the system provides an AI Project Summary, which analyzes task statistics to report progress and recommend items that need attention, and an AI Risk Analysis, which examines the project's actual task data to identify potential delivery risks and suggest mitigation actions. The system is implemented on the MERN stack, with AI functionality provided by the Groq API using the openai/gpt-oss-120b model. The implemented modules are evaluated qualitatively through functional observation. An AI Project Assistant is identified as a planned enhancement and is not part of the current implementation.

KEYWORDS: Task management, Large Language Model, AI risk analysis, Kanban board, Socket.IO, MERN stack.

I. INTRODUCTION

Modern software engineering teams work in fast-paced environments characterized by agile practices, distributed team structures and continuously changing deliverables. The early phases of a project, such as scoping requirements, breaking them down into a development backlog and estimating task priorities, rely heavily on manual input, and project managers and technical leads spend considerable time translating high-level requirements into technical tasks. Keeping the team aligned further requires frequent status reporting, manual risk assessment and board updates. The integration of Large Language Models (LLMs) with reactive web technologies offers an opportunity to reduce this administrative overhead so that teams can concentrate on implementation.

Four problems motivate this work:

- (i) deconstructing a project scope into granular tasks demands considerable manual effort;
- (ii) bottlenecks and potential delivery hazards are usually identified reactively rather than proactively;
- (iii) conventional tracking systems often show stale data unless users refresh their dashboards, causing duplicated or misaligned effort; and
- (iv) collecting metrics and writing progress summaries for stakeholders consumes engineering time.

The objectives of the project are:

- (a) to automate the breakdown of a high-level project description into structured, prioritized tasks using an integrated LLM;
 - (b) to provide an interactive Kanban board with drag-and-drop workflow tracking;
 - (c) to propagate updates to all active users in real time using WebSocket connections (Socket.IO); and
 - (d) to provide an AI-generated project summary and an AI-based risk analysis, built on the project's actual task data, that highlight progress, potential delivery risks and tasks requiring attention, together with suggested mitigation actions.
- The contribution of this work is an implemented system that combines these functions in a single web application.

The rest of the paper is organized as follows. Section II reviews related work. Section III describes the system model, architecture and database design. Section IV presents the methodology and implementation of each module, including the AI Risk Analysis. Section V discusses the results, and Section VI concludes the paper and outlines future work.

II. LITERATURE SURVEY

Three earlier studies are relevant to this work. A task management system built on the MERN stack was reported in [1]; it provides user authentication, task creation, project management and dashboard-based monitoring. Its merits are a fast user interface, a scalable architecture, REST API integration and easy task tracking, but it offers no AI-based task generation and has limited real-time collaboration. A collaborative project management system using the MERN stack with Socket.IO was described in [2], in which changes to tasks are communicated instantly among team members. It supports real-time updates, improved collaboration and efficient communication, but it provides no intelligent task recommendations or AI-based project planning. The use of LLMs to convert project descriptions into structured task lists and to assist project planning was explored in [3]. The approach reduces manual planning effort and supports AI-generated task breakdown; its limitations are dependence on API availability and the need to review the generated tasks manually before use.

Based on these works, three gaps can be identified. First, existing platforms typically provide either conventional board-based tracking or isolated AI-assisted text generation, so teams must still define backlogs and prepare status reports manually. Second, systems that explore LLMs for project planning mostly treat the model as a stand-alone text generator, without coupling it to a multi-user Kanban workflow with live synchronization. Third, current solutions lack an integrated mechanism in which the live task data of a project is analyzed by generative modules to produce project summaries and risk analyses with suggested mitigation actions. The system presented in this paper addresses these three aspects within one application.

III. MATERIALS AND METHODS: SYSTEM MODEL

A. Existing Practice and Its Limitations

Conventional project management tools behave largely as passive repositories in which the project manager must enter every requirement, estimate complexity, delegate work and aggregate metrics manually. Task creation and allocation are manual and may lead to uneven workloads and planning delays at project kickoff. Tracking depends on members updating their own task states, and platforms based on plain HTTP request-response cycles show stale boards until a page is reloaded. Status reports are compiled by hand and are prone to subjective bias, and risks and overdue tasks are identified largely through human judgment during periodic meetings, so they are handled reactively.

B. System Architecture

The system follows a four-tier architecture (Fig. 1). The client tier is a React.js single-page application that provides the dashboard, the real-time Kanban board and the AI screens. The application tier is a Node.js and Express.js backend that exposes REST APIs, performs JWT authentication and hosts a Socket.IO gateway for real-time synchronization. The database tier is MongoDB Atlas, which stores users, teams, projects, tasks and AI request logs. The external AI tier is the Groq Cloud API running the openai/gpt-oss-120b model, which is used for task generation, project summaries and risk analysis. Transactional operations such as login and project creation use HTTP/REST requests, whereas Socket.IO WebSockets provide live synchronization during Kanban operations. Files attached to tasks are uploaded to Cloudinary, and the resulting file URL is stored with the task record in MongoDB.

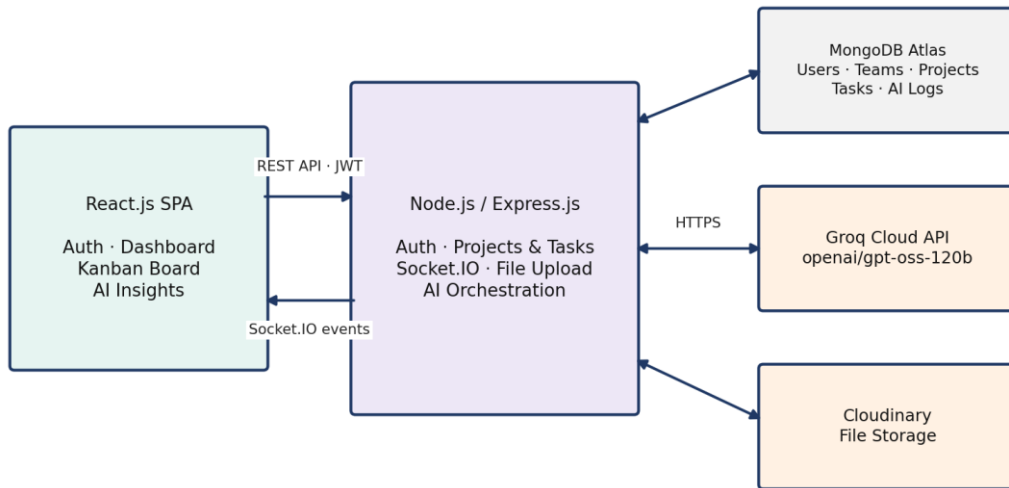


Fig.1. System architecture of the AI-Powered Smart Task Management System

C. Technology Stack

The technologies used are summarized in Table I. React.js [4] builds the single-page client; Node.js [5] and Express.js [6] implement the backend; MongoDB [7], accessed through the Mongoose ODM, stores the data; Socket.IO [8] provides bidirectional real-time communication with an HTTP long-polling fallback; stateless JSON Web Tokens [9] authenticate REST requests and Socket.IO handshakes, with passwords hashed using bcrypt; the Groq platform [10] serves the openai/gpt-oss-120b model; and Cloudinary [11] stores uploaded files.

TABLE I .TECHNOLOGY STACK

Layer	Technologies
Frontend	React.js, HTML5, CSS3, Tailwind CSS, React Router, Axios, DnD Kit, Socket.IO Client, Context API
Backend	Node.js, Express.js, Multer (multipart file upload)
Database	MongoDB Atlas, Mongoose (ODM)
AI service	Groq Cloud API, openai/gpt-oss-120b
Real-time communication	Socket.IO (WebSocket with HTTP long-polling fallback)
Security	JSON Web Token (JWT), bcrypt.js
File storage	Cloudinary
Development tools	Visual Studio Code, npm or yarn, Postman or Thunder Client

D. System Workflow

The operational sequence is shown in Fig. 2. Users authenticate with JWT-based credentials. The Team Leader creates a project and submits its requirements, and the AI module parses them into structured tasks with priorities. The Team Leader reviews, edits and assigns the tasks, after which team members update task stages on the Kanban board with real-time synchronization. On request, the system analyzes the project's task data to produce an AI Project Summary and an AI Risk Analysis with suggested mitigation actions.

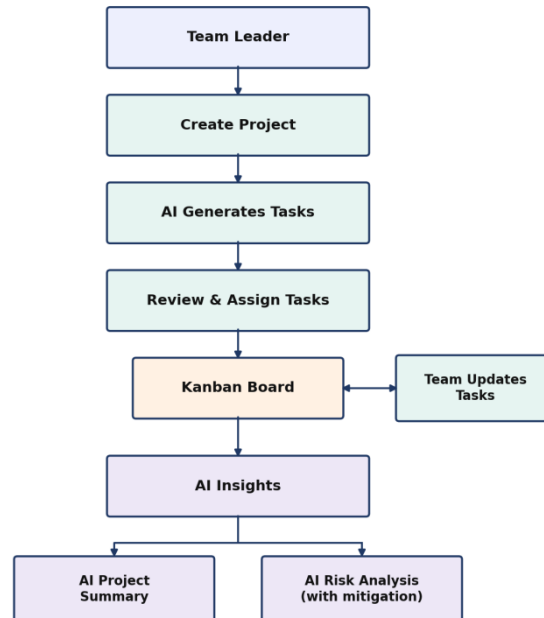


Fig.2. Project workflow from project creation to AI insights

E. Data Flow and Database Design

At the context level, Team Leaders submit project requirements and receive analytical reports, Team Members update task states and track progress, and the system communicates with the external Groq LLM inference service. Level 1 of the data flow diagram (Fig. 3) decomposes the system into six processes: Authentication (1.0), Team and Project Management (2.0), AI Task Generation (3.0), Kanban Task Management and File Attachment (4.0), Socket.IO Real-Time Synchronization (5.0), and AI Insights covering the Project Summary and Risk Analysis (6.0). Each process interacts with the MongoDB collections.

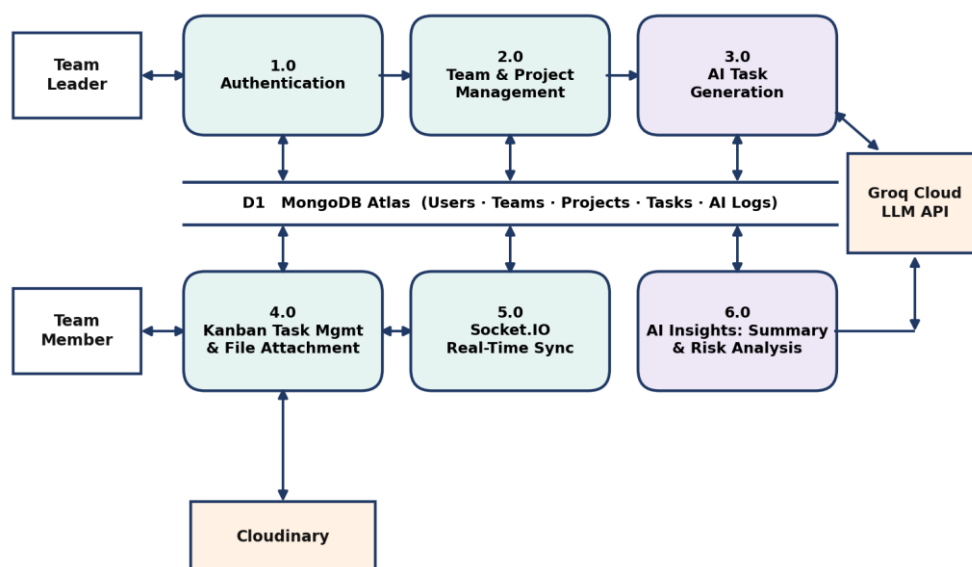


Fig.3. Data flow diagram, Level 1

The database (Fig. 4) contains five collections. The Users collection stores the name, unique email, bcrypt-hashed password, role (leader or member) and team reference. The Teams collection stores the team name, the leader reference and the member identifiers. The Projects collection stores the title, description, team reference, creator and timestamps. The Tasks collection stores the parent project, title, description, priority (Low, Medium, High), status (To Do, In Progress, Done), assignee, due date, attached file URL and name, creator and timestamps. The AI Logs collection records, for backend use only, each AI request with its project, requesting user, prompt, response, status (success or failed), AI feature and timestamps.

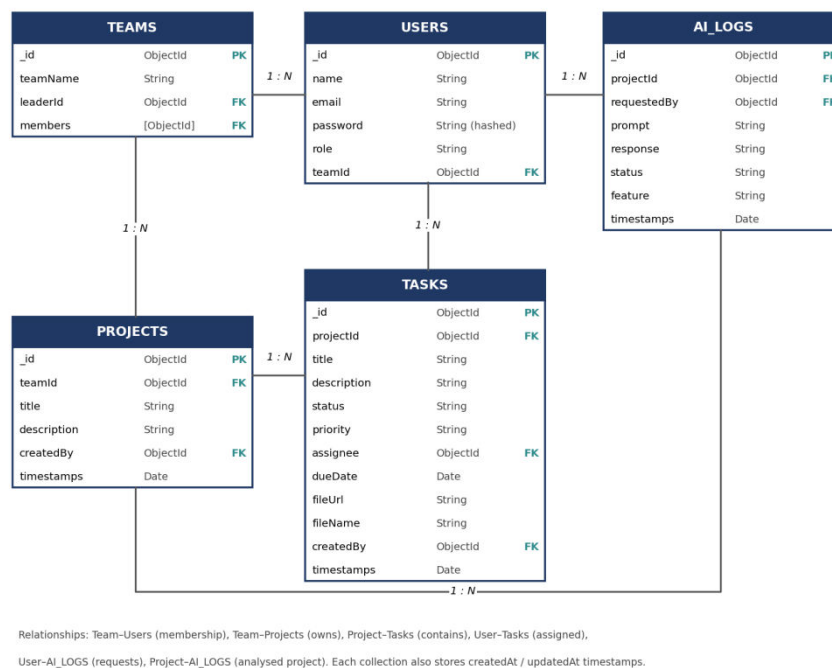


Fig.4. Entity relationship diagram

IV. PROPOSED METHODOLOGY AND IMPLEMENTATION

The system is implemented as a set of modules that share the same backend and database. The modules that are implemented are described in sub-sections A to E, and Table II summarizes the implementation status of the AI features.

A. Authentication, Team and Project Management

Users register with a name, unique email and password; passwords are salted and hashed with bcrypt before storage. After login the server issues a signed JWT, which the client uses for subsequent REST requests and for the Socket.IO handshake. Team Leaders create teams, manage member lists and create projects by defining a title and a description of the project's objectives and requirements. Project metadata, team records and task collections are kept consistent in MongoDB.

B. AI Task Generation

The Team Leader enters an unstructured description of the project's goals, scope and technical requirements. The backend constructs a prompt from this description and queries the Groq API running openai/gpt-oss-120b to break the requirements down into discrete components. For each component the model returns a concise description and a priority tier (Low, Medium or High). The raw output is validated and parsed into structured JSON, which can be ingested directly into the database for the Team Leader's review. Task assignment remains the decision of the Team Leader; the AI reduces only the effort of task creation.

C. Real-Time Kanban Task Management and File Attachment

Reviewed tasks are assigned to team members with due dates and are displayed as cards on a board with three stages: To Do, In Progress and Done. Members drag and drop cards or open a detail view to change the assignee, due date, priority or status. Each change is emitted to the server through Socket.IO and broadcast to all connected clients, so that the board is updated without a page refresh. Files related to a task are received by the backend through Multer and uploaded to Cloudinary; the file URL and file name are stored with the task record.

D. AI Project Summary

On request, the system aggregates task statistics from MongoDB: the total number of tasks, tasks by status and priority, overdue tasks and unassigned tasks. These statistics are sent to the Groq API, which returns a natural-language progress summary together with recommendations on the tasks that need attention.

E. AI Risk Analysis

The AI Risk Analysis module extends the Project Summary by focusing on potential risks to project delivery rather than on current progress alone. It works on the actual project and task data stored in MongoDB, not on sample data, and is integrated as an additional, independent AI feature: the task generation, Kanban and real-time synchronization modules are not altered by it. The module proceeds in four steps.

1) Data collection. When the Team Leader requests a risk analysis, the request passes through the same JWT-based authentication as the other project APIs, and the backend retrieves the project's task data, such as task status, priority, assignee and due date.

2) Risk identification. The collected data is organized into a prompt and sent to the Groq API running openai/gpt-oss-120b, which analyzes it to identify potential risks to project delivery, including possible delays.

3) Mitigation suggestions. For the identified risks the model also suggests mitigation actions. The result is returned to the client and presented to the Team Leader, who may act on it by updating tasks, priorities or assignments on the Kanban board.

4) Logging. AI requests made through the backend are recorded in the AI log together with the AI feature that produced them.

The output is advisory: final decisions remain with the Team Leader. Its usefulness depends on the quality of the underlying task data; for example, tasks without due dates or assignees provide less information to the analysis.

F. Implementation Status of the AI Features

An AI Project Assistant, a conversational interface that would answer project-specific questions such as which tasks the team should focus on next, is proposed as a future enhancement. It has not been implemented in the current version of the system. Table II distinguishes the implemented features from the planned one.

TABLE II .AI FEATURES AND IMPLEMENTATION STATUS

Feature	Input	Output	Status
AI Task Generation	Project description entered by the Team Leader	Structured tasks with title, description and priority (Low, Medium, High) for review	Implemented
AI Project Summary	Task statistics from MongoDB (counts by status and priority, overdue, unassigned)	Progress summary with recommendations	Implemented
AI Risk Analysis	Project task data (status, priority, assignee, due date)	Potential delivery risks and suggested mitigation actions (advisory)	Implemented
AI Project Assistant	Project and task data as context	Answers to project-specific questions	Planned, not implemented

V. EXPERIMENTAL RESULTS AND DISCUSSION

The implemented modules were assessed by observing their functional behavior. The results are therefore qualitative; no quantitative performance or accuracy measurements are reported in this paper.

A. Authentication and Project Management

User onboarding and login were verified with bcrypt password hashing and stateless JWT generation. Workspace configuration and project initialization were restricted to designated Team Leaders, and project metadata, workspace boundaries and member allocations remained consistent across team rosters.

B. AI Task Generation

Natural-language project descriptions submitted through the dashboard were parsed by the openai/gpt-oss-120b model via the Groq API. The model decomposed high-level goals into discrete, practical tasks and returned structured JSON arrays containing objective titles, execution scopes and recommended priorities.

C. Kanban Board and Real-Time Updates

The Kanban board rendered tasks in the To Do, In Progress and Done columns. Leaders assigned tasks to specific members, and ownership was shown on the corresponding task cards. Drag-and-drop operations updated the task state and the status stored in the database. Socket.IO maintained continuous connections between the clients and the Node.js backend; stage transitions, column moves and assignments were reflected on all active member dashboards without page reloads, including during concurrent multi-user sessions.

D. AI Project Summary and AI Risk Analysis

The project summary consolidated task statistics across statuses and priorities, overdue tasks and unassigned tasks, and converted them into a progress summary with recommendations, giving team leaders a high-level view of progress and remaining work. The AI Risk Analysis retrieved the actual project and task data from MongoDB and used it to generate its analysis, rather than relying on sample data. It highlighted potential risks to project delivery on the basis of task status, priority, assignment and due dates, and presented mitigation suggestions to the Team Leader, who can act on them through the Kanban board. As a separate AI function, it left the existing task generation, Kanban and real-time update workflows unchanged.

E. Discussion

Taken together, the observations show that task generation, board tracking and AI-driven project insights can be integrated in a single web application, and that automated task decomposition reduced the manual effort needed to create project tasks. This reduction was observed qualitatively and was not measured. The AI Risk Analysis adds a proactive view of potential risks that complements the progress summary. Several limitations should be noted. The AI outputs are advisory and depend on the quality of the task data; the AI features rely on an external LLM inference service and therefore on reliable internet connectivity; and the AI Project Assistant has not been implemented or evaluated.

VI. CONCLUSION AND FUTURE WORK

This paper presented an AI-Powered Smart Task Management System that combines a MERN-based collaborative workspace with generative AI to reduce the manual effort of project planning and monitoring. Users can form teams, create projects and manage tasks on a Kanban board synchronized across clients using Socket.IO. The AI Task Generation feature converts a project description into structured, prioritized tasks that the Team Leader reviews and assigns, while the AI Project Summary and AI Risk Analysis use the project's actual task data to report progress, identify potential delivery risks and suggest mitigation actions. The AI output is advisory, and task assignment and project decisions remain with the team.

The following enhancements are planned and are not part of the current implementation: integration with Git repositories such as GitHub to link repository activity to tasks; notification and calendar integration for task assignments and due dates; predictive project forecasting, extending the current risk analysis, which examines the present state of a project, to forecast likely delays and completion dates using historical project data; a native mobile

application; AI-assisted task distribution, since task assignment is currently manual and would require reliable data on member skills and workload; and the AI Project Assistant described in Section IV-F.

REFERENCES

1. T. Schick et al., “Toolformer: Language Models Can Teach Themselves to Use Tools,” Advances in Neural Information Processing Systems (NeurIPS), vol. 36, 2023. Paper: Official NeurIPS paper
2. J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” Advances in Neural Information Processing Systems (NeurIPS), vol. 35, pp. 24824–24837, 2022. Paper: Official NeurIPS paper
3. T. Yao et al., “ReAct: Synergizing Reasoning and Acting in Language Models,” International Conference on Learning Representations (ICLR), 2023. Paper: Official ICLR paper
4. Q. Yuan et al., “TaskLAMA: Probing the Complex Task Understanding of Language Models,” arXiv preprint arXiv:2308.15299, 2023. Paper: TaskLAMA research paper
5. A. Prasad et al., “ADaPT: As-Needed Decomposition and Planning with Language Models,” arXiv preprint arXiv:2311.05772, 2023.
6. Z. Zhao, W. S. Lee, and D. Hsu, “Large Language Models as Commonsense Knowledge for Large-Scale Task Planning,” Advances in Neural Information Processing Systems (NeurIPS), vol. 36, 2023.
7. Paper: Official NeurIPS paper
8. [7] Harnessing the Power of Large Language Model for Natural Language-Driven Task Scheduling Optimization,” 2025 30th International Conference on Automation and Computing (ICAC), 2025.
9. [8] S. Liu et al., “Select-Then-Decompose: From Empirical Analysis to Adaptive Selection Strategy for Task Decomposition in Large Language Models,” Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5454–5477, 2025.
10. [9] S. Hao, Y. Gu, H. Ma, J. Hong, Z. Wang, D. Z. Wang, and Z. Hu, “Reasoning with Language Model is Planning with World Model,” Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 8154–8173, 2023.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



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