IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.02

AI-Powered Task Management

Prof.P.Gopala Krishna
Assistant Professor, Dept. of CSE,
Vemana Institute of Technology, Bengaluru, India
gopala.krishna@vemanait.edu.in
https://orcid.org/0009-0000-8645-1109
Charangoopalli M, G Revanth, Hemanth KR, Jai Krishna S
Students, Dept. of CSE
Vemana Institute of Technology, Bengaluru, India
charangoopalli20@gmail.com, revanthreddy2050@gmail.com
reddyh68hemanth@gmail.com, jaikrishnas2603@gmail.com

Publication History
Manuscript Reference: IRJCS/RS/Vol.13/Issue01/CSJA26.JACS10081
Research Article | Open Access | Double-Blind Peer Reviewed Article ID: IRJCS/RS/Vol.13/Issue01/CSJA26.JACS10081
Received:12,December 2025,Revised:24,December 2025,Accepted:02 January 2026 Published Online:20 January 2026
https://www.irjcs.com/volumes/Vol13/iss-01/02.CSJA26.JACS10081.pdf
Article Citation: Gopala,Charangoopalli,Revanth,Hemanth,Jai(2026),AI-Powered Task Management, IRJCS: International Research Journal of Computer Science, Volume 13, Issue 01 of 2026 pages 05-09
Doi:>https://doi.org/10.26562/irjcs.2026.v1301.02
BibTeX Key Gopala@2026AI-Powered
IRJCS papers should be cited as IRJCS (International Research Journal of Computer Science, AM Publications, India 2026, ISSN 2393-9842, https://doi.org/10.26562/ijrcs.2025.v1301.02 The journal's official abbreviation is IRJCS.
Orcid: https://orcid.org/0009-0004-9398-7488

Abstract: Our innovative system is a comprehensive task management system designed to streamline and enhance the efficiency of daily activities. Rooted in the realm of artificial intelligence, Ai-Powered Task Management seamlessly integrates advanced functionalities to redefine how users organize their tasks. This system envisions a user-centric approach to task management, empowering individuals with a personalized and intelligent solution. Ai-Powered Task Management boasts an array of features, including automatic task categorization, reminding tasks with deadline, user performance, insights about user and task suggestions. It is capable of accepting multiple tasks as input along with tasks that requires reminding on time. It is capable of suggesting users their frequent tasks and provide them with valuable insights. It's main strength surrounds the fact that it can accept input from a user in natural language with spelling mistakes and then corrects them into proper for- mat. Ai-Powered Task Management stands at the intersection of simplicity and sophistication, offering users a powerful tool that effortlessly balances functionality and accessibility in the realm of task management.
Index Terms: Text Prioritization, Text Processing, Neural Networks, Workflow Optimization and Task Scheduling.

I. INTRODUCTION

As we delve into the development and implementation of Ai-Powered Task Management, we anticipate a transformative impact on the way users engage with their daily responsibilities. Task management has never been a question in attaining productivity and efficiency, both at the individual and organizational levels. Early on, individuals used manual systems paper planners, notebooks, wall calendars, and sticky notes to manage their tasks. As organizations became more sophisticated, there was a need for digital software. Programs like Trello, Asana, Microsoft Planner, and other project management software were well liked for their capability to centralize task information and allow for collaboration. Although these technologies were a big improvement, they were still highly labor-intensive. People had to enter, arrange, and sequence tasks them, usually creating information overload or mismanagement in rapidly changing environments. As the world of work accelerated, the need emerged for systems that could provide increased flexibility, responsiveness, and support in managing day-to-day work something wiser and more attentive than dead static electronic to-do lists. Intelligent task management software has stepped in to address this need. This software's surpass mere listing of tasks. They provide capabilities that can learn from patterns, comprehend user commands more naturally, and adapt flows in real time, thus revolutionizing how tasks are processed. Smart task management is the application of contemporary technologies and automating approaches to optimize task management for being more efficient, personalized, and proactive. Rather than just being a repository of to-dos, smart systems actively assist in keeping the user on track, assisting the user in making better decisions, and alleviating mental load.

II. RELATED WORK

The development and validation of the Intelligent Task Management System (ITMS) builds upon established research across three critical areas: intelligent project management specialized testing for AI/ML components, and robust methodologies for modern full-stack web applications. Existing literature confirms a trend toward augmenting traditional task managers with Artificial Intelligence, specifically for functionalities like Smart Task Prioritization, Automated Work Allocation, and Predictive Deadlines, often leveraging Natural Language Processing (NLP) to categorize and analyze task inputs.

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.02

Our work aligns with this by focusing on NLP-based classification, necessitating a specialized testing approach. For the AI components, research indicates that traditional functional testing is insufficient; instead, validation must address the model's accuracy, stability, and inference latency, which is a key focus of our performance testing strategy. Furthermore, the use of a React / Flask architecture requires adherence to established full- stack testing best practices. This body of work underscores the necessity of a layered testing approach covering Unit testing for isolated modules (like task parsing and authentication handlers), Integration testing for RESTful data exchange, and End-to-End testing to validate complete user journeys including secure login via JWT, task classification, and reporting. Finally, prior research consistently mandates rigorous Security Testing, encompassing JWT validation and SQL injection prevention, alongside Performance and Load Testing to ensure system scalability and low latency under concurrent user activity. Collectively, this related work establishes the foundational principles upon which our comprehensive testing framework for the ITMS is constructed, validating its robustness, usability, and reliability. Task prioritization and intelligent to-do systems. Several applied studies and prototypes apply supervised and ensemble ML models to predict task importance, or to automatically rank items in a user's to-do list based on features like deadlines, estimated effort, historical completion patterns, and contextual signals such as calendar and email.

III. PROBLEM STATEMENT, SCOPE AND OBJECTIVES

A. Problem Statement

Effective task management remains a significant challenge for individuals and teams in both professional and personal environments. Despite the availability of various task management tools, many still struggle with inefficiencies, missed deadlines, and lack of motivation, all of which impact overall productivity. Task Genius, an AI-powered task monitor, addresses these issues by leveraging advanced AI technologies to optimize task organization, prioritization, and execution. Many users face the problem of managing numerous tasks simultaneously, often leading to stress and burnout. Prioritize tasks based on urgency, importance, and user behavior.

B. Scope

The scope of the proposed system encompasses natural language-based task entry, where users can create tasks without adhering to strict input formats. Using AI and NLP, the system interprets task details, categorizes tasks automatically, and predicts their urgency or status. Users can view, update, filter, and monitor all tasks through a centralized dashboard, while the system generates performance analytics to provide insights into productivity trends. Deadline notifications and reminders help users manage time effectively and reduce the risk of incomplete tasks. Additionally, the system supports report generation for past tasks, enabling detailed documentation of completed, failed, and pending tasks. Designed for cross platform accessibility, the system ensures convenient usage on both web and mobile devices. However, the system does not include integration with third-party project management tools, advanced AI-based task delegation, or offline task processing, as NLP features require an active internet connection.

C. Objective

The primary objective of the proposed AI-Powered Task Management System is to provide an intelligent and efficient Platform for task management, allowing users to input tasks in natural language and enabling the system to understand, categorize, and manage them automatically using AI and NLP techniques. The system aims to enhance user productivity by providing personalized insights and performance analytics, helping users monitor their task completion efficiency and identify areas for improvement. It also seeks to automate routine operations, such as task categorization, status updates, and deadline notifications, reducing manual intervention and minimizing the risk of missed deadlines. Furthermore, the system facilitates comprehensive task tracking, including completed, failed, and pending tasks, and allows users to generate detailed reports for record-keeping and performance evaluation. Finally, the platform emphasizes a user-friendly interface, ensuring seamless interaction across multiple platforms while supporting efficient task management.

IV. METHODOLOGY / SYSTEM DESIGN

AI-powered task management system where users can in- put tasks with or without reminders. Using natural language processing (NLP), tasks are categorized into general, work, or important categories, and the system corrects spelling mistakes and resolves ambiguities. Users can mark tasks as completed or failed, receiving notifications for reminders set with specific times (e.g., "Buy milk 10 am"). The system utilizes historical data of completed and failed tasks to offer insights and suggest new frequently done tasks, improving with more user data over time.

A. System Architecture

The proposed system architecture is illustrated in Figure 1. Users input multiple or single tasks with or without reminders into AI-Powered Task Management, which categorizes the tasks using NLP into general, work, or important categories, and corrects spelling mistakes or ambiguities. Users can mark tasks as completed or failed by pressing the done or failed button, respectively. If a reminder is set by adding a time next to the task (e.g.,"Buy milk 10 am"), users receive notifications. AI-Powered Task Management has access to both failed and completed tasks, providing valuable insights and suggesting new frequently done tasks to the user, improving with more historical data

B. Functional and Non-Functional Requirements

Key functional requirements of the Medicure system include:

User Authentication: The system shall provide secure user registration and login mechanisms. It must validate user credentials before granting access to the application

Access to AI-Powered Task Dashboard: Upon successful authentication, the system shall open the AI-enabled task management interface where users can interact with all available features.

IRJCS: International Research Journal of Computer Science
Volume 13, Issue 01, January 2026
ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.02

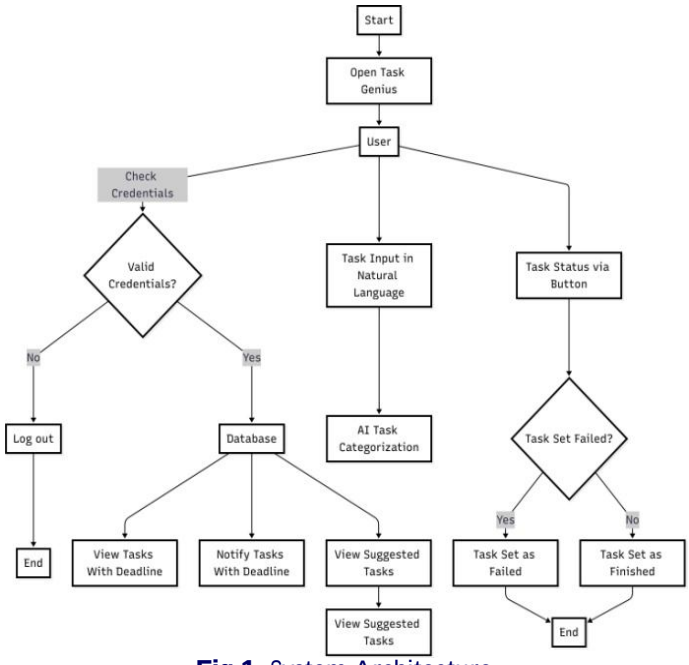

Fig.1. System Architecture

Natural Language Task Input: The system shall allow users to input tasks in natural language. This includes short descriptions, deadlines, and contextual details.

Natural Language Processing (NLP): The system shall process user-entered natural language using NLP techniques to extract relevant task parameters such as title, category, priority, and deadlines.

Automated Task Categorization: The system shall automatically categorize tasks based on their nature, urgency, and semantic meaning derived from NLP analysis

Automated Task Status Identification: Based on user statements (e.g.,“ completed the assignment”),the system shall automatically set the task status as “Finished” or “Failed.”

Task Viewing and Management: The system shall allow users to view all tasks, including ongoing, completed, and failed tasks. Users may also filter tasks based on status or category.

Performance Insights and Analytics: The system shall generate user-specific insights, including task completion rate, productivity patterns, and monthly/weekly performance analytics.

Deadline Notifications: The system shall notify users regarding approaching deadlines for pending tasks and provide reminder alerts.

Report Generation: The system shall generate detailed reports of past tasks and allow users to export the reports for future reference.

Logout: The system shall allow users to securely log out of the application at any time.

Non-functional requirements ensure the system’s reliability and effectiveness:

Performance: The system should provide fast response times. NLP-based task processing should not exceed 2–3 seconds, and the dashboard should load all tasks within 2 seconds.

Usability: The user interface should be simple, intuitive, and easy to navigate. The system must minimize the learning curve and support seamless task entry through natural language.

Reliability: The system should maintain high availability, with at least uptime. All notifications and reminders must be delivered accurately and without delay.

Security: The system must ensure secure handling of user data using encrypted storage methods. Only authenticated users shall access personal tasks and reports. Data protection measures must prevent any unauthorized modification of stored records.

Scalability: The system should scale efficiently as the number of users and tasks grows. AI/NLP models and cloud resources should adapt according to workload variations.

Maintainability: The system shall follow modular code architecture, allowing easy updates to NLP models, UI components, and database schemas. Proper documentation must be maintained for future development.

Portability: The system should be deployable on multiple platforms, including Android, web, or hybrid cross-platform environments. Minimal configuration should be required for system migration or upgrades.

V. IMPLEMENTING AND TESTING

A. Technology stack

This project uses a full stack architecture, including a responsive React frontend and a robust Python Flask backend. On the front end, it will use React.js with Vite for superfast bundling, React Router to handle navigation, and Tailwind CSS for styling; there are chart libraries for analytics visualization. The backend will use Flask, with SQL Alchemy as its ORM for all database operations, hence supporting SQL databases like SQLite or MySQL. Machine learning using Scikit-Learn,

ISSN: 2393-9842
https://www.irjcs.com/archives
https://doi.org/10.26562/irjcs.2025.v1301.02

Tensor Flow, NumPy, and P and as will be used to enable intelligent task prioritization and predictive analytics. Other backend services include JWT-based authentication, SMTP-based email notifications, and export functionalities enabled through the use of OpenPyXL, XLSX Writer, and Report Lab for Excel and PDF report generation. The described tech stack empowers a powerful, scalable, and intelligent task management system.

B. Hardware and Deployment

The Task Management System is designed using a full- stack architecture that integrates a React-based frontend with a robust Python Flask backend, and its hardware requirements remain modest to ensure accessibility and efficiency. For development, the system can operate smoothly on a dual-core processor with at least 8 GB of RAM and 10 GB of available storage, while an optional CUDA-enabled GPU may be utilized to accelerate machine learning model training. In production environments, a quad-core processor with 8–16GB of RAM and SSD-based storage is recommended to support backend operations and concurrent user activity, with GPU acceleration required only for intensive real-time analytics. Deployment of the frontend is accomplished through React and the Vitebundler, enabling hosting on services such as Ver- cel, Netlify, GitHub Pages, or traditional servers using Nginx or Apache. The Flask backend, developed with SQLAlchemy, JWT authentication, Scikit-Learn, TensorFlow, Pandas, and NumPy, can be deployed on cloud platforms including AWS EC2, Google Cloud, and Microsoft Azure, or through containerized environments using Docker, Docker Compose, or Kubernetes for scalable orchestration. SQL databases such as SQLite, MySQL, or Post greSQL are supported for data persistence depending on deployment requirements, while backend services such as SMTP-based email notifications and export generation using OpenPyXL, XlsxWriter, and ReportLab further enhance system capabilities. Overall, the deployment architecture and hardware configuration collectively support a scalable, efficient, and intelligent task management platform capable of delivering advanced analytics, predictive insights, and seamless user interaction

C. Functional Testing

Functional testing was employed across three hierarchical levels to validate that all specified application behaviors were executed correctly, ranging from isolated functions to complete workflows.

Unit Testing: Unit testing was employed extensively to validate individual software modules in isolation. On the backend, this included rigorous testing of task parsing functions, database models, core authentication handlers, and the machine learning algorithms responsible for task prioritization and predictive analytics. On the frontend, component-level tests verified UI responsiveness, routing behavior, and the proper rendering of all graphical elements, such as task lists, input forms, and analytical charts.

Integration Testing: Integration testing evaluated the interaction and accurate communication between various components. A primary focus was the exchange of data between the backend APIs and frontend interfaces, ensuring data was accurately and efficiently exchanged through the defined RESTful endpoints. Additionally, database integration tests specifically validated the correctness of SQL Alchemy ORM operations and confirmed the consistency of CRUD (Create, Read, Update, Delete) functionalities across the chosen database environments (SQLite or MySQL).

End-to-End (E2E) Testing: End-to-End testing replicated comprehensive real-user workflows to confirm that the sys- tem performed as expected under real operating conditions. Representative test cases included the entire user journey: account creation, secure login utilizing JWT, standard task creation, NLP-based task classification, generation of email notifications, and exporting reports, there by validating the full system flow.

D. Non-Functional Testing Categories

Non-functional testing evaluated critical operational qualities of the system, including its efficiency, stability under stress, and protection against threats.

Performance Testing: Performance testing assessed the system's efficiency and responsiveness. This involved evaluating key metrics such as API response times, the inference latency of machine learning models, and the overall efficiency of frontend rendering, ensuring the system operates quickly and smoothly.

Load Testing: Load testing focused on assessing scalability and system stability by simulating conditions beyond typical usage. This involved simulating multiple concurrent users interacting with the system simultaneously to identify potential bottlenecks and confirm that the system remains stable and reliable under significant user demand.

Security Testing: Security testing ensured protection against common vulnerabilities and maintained the integrity of user data. This included rigorous checks for JWT validation, thorough input sanitization across all forms, and verification of SQL injection prevention mechanisms, establishing a secure operational environment

VI. CONCLUSION

Our AI-Powered task management application is an AI- powered task management system where users can input tasks with or without reminders on Android platform. It changes the way individuals manage their daily tasks and commitments in modern world. AI-Powered task management consists of many features, including automatic task categorization, reminding tasks with deadline, user performance, insights about user and task suggestions. AI-Powered task management gets lists of completed and failed tasks and gives advice to the user. The main strength of AI-Powered task management surrounds the fact that it can accept input from a user in natural language with spelling mistakes and then corrects them into proper format. The exploration of AI-powered task reminder systems rep resents an exciting frontier in technological advancement, and while the research is currently in its nascent stages, the potential benefits are indeed profound. The integration of artificial intelligence has the capacity to revolutionize the way we approach task management by introducing a level of personalization and relevance that traditional systems lack. Through the utilization of sophisticated algorithms and Natural Language Processing (NLP), AI can tailor reminders to individual preferences and habits, ensuring that they are not only timely but also specifically tailored to the user's unique needs.

One of the key promises of AI-powered task reminders lies in their ability to mitigate distractions and interruptions. By leveraging machine learning to understand user behavior and context, these systems can provide reminders at opportune moments, minimizing disruptions and enhancing overall efficiency. This capability is particularly crucial in today's fast-paced and interconnected world, where constant digital notifications can contribute to stress and overwhelm. As AI and NLP technologies continue to advance, we can anticipate the emergence of even more sophisticated and effective task reminder systems in the coming years. These advancements may include a deeper understanding of user intent, improved contextual awareness, and seamless integration with other productivity tools. The potential outcome is a paradigm shift in personal organization and productivity, offering individuals the tools they need to navigate the demands of modern life with greater ease. In essence, the evolution of AI-powered task reminder systems holds the promise of not only making individuals more productive but also contributing to a reduction in stress and anxiety associated with managing multiple responsibilities.

ACKNOWLEDGMENT

We express our sincere gratitude to our project guide and faculty members for their continuous support, guidance, and encouragement throughout the development of this project. We also thank the institution for providing the necessary resources, laboratory facilities, and a conducive environment for experimentation. Finally, we appreciate our classmates and friends for their cooperation, feedback, and assistance, which greatly contributed to the successful completion of this project.

REFERENCES

1. C.L.Paul,A.Komlodi and W.G.Lutters, "Interruptive notifications in support of task management", International Journal of Human-Computer Studies, vol. 79 , pp. 20–34, July 2015. https://doi.org/10.1016/j.ijhcs.2015.02.001
2. D.Khurana, A.Koli, K.Khatter and S. Singh, "Natural language processing: state of the art, current trends and challenges", vol. 82, no.3, pp. 3713– 3744, July 2022. https://doi.org/10.1007/s11042-022-13428-4
3. D.Graus,P.N.Bennett,R.W.White,E.Horvitz,"Analyzing and Predicting Task Reminders",Proceedings of the Conference on User Modeling Adaptation and Personalization, pp. 7–15, 2016. https://doi.org/10.1145/2930238.2930239
4. A.N.Liyanage, A.G.A.D.Madhushanka, U.M. L.Uduwara, W. M.S.Jayarathne, S. Siriwardana, S. Reyal, et al., "Schedule ME - Smart-Digital Personal Assistant for Automatic Priority Based Task Scheduling and Time Management", 2nd Global Conference for Advancement in Technology (GCAT), 2021.https://doi.org/10.1109/GCAT52182.2021.9587876
5. Integrating AI into Task Management by Smith et al. (Journal of Artificial Intelligence, 2020)