© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715447

AI-Driven Intelligent Automated Task Allocation System Using Context Aware Decision Modeling and Dynamic Workload Management

A ELUMALAI[1], DR K PONMOZHI[2]
[1] *PG Student, Department of Computer Application, SRM Valliammai Engineering College, Kattankulathur, Chennai, Tamil Nadu, India*
[2] *Associate Professor, Department of Computer Applications, SRM Valliammai Engineering college, Kattankulathur. Chennai, Tamil Nadu, India*

Abstract ***Manual task assignment in organizations is time-consuming, inconsistent, and leads to unbalanced workloads as team sizes grow. This paper presents an AI-driven intelligent task allocation system that automates the entire assignment process by integrating WhatsApp as the task submission interface with a Discord bot and Groq LLM. When a task message is received, the LLM evaluates the content, assigns a priority level, selects the most suitable worker from the Experts table in Supabase, and writes the structured record — containing task, priority, worker name, and worker email — directly into the Task table without any manual intervention. When all workers are occupied, the system identifies the worker nearest to completing their current task and assigns accordingly. A React frontend delivers role-specific dashboards, task tracking, pie chart reports, and inter-role chat for Super Admin, Admin, and Worker roles, while completed tasks are automatically archived into the CompletedTask table.***

Index Terms- ***Automated Task Allocation, Groq LLM, Discord Bot Integration, WhatsApp Task Pipeline, Dynamic Workload Balancing, Role-Based Access, Priority Classification, Expert Matching System.***

I. INTRODUCTION

In any organization, deciding who should handle which task becomes increasingly difficult as team sizes and domain complexity grow. Managers overseeing workers across IT, HR, and Accounting cannot realistically track individual workloads, skill profiles, and deadlines simultaneously, leading to assignments made on habit rather than informed judgment. This cognitive burden results in unbalanced workloads, missed deadlines, and reduced organizational productivity over time.

Most organizations rely on WhatsApp to communicate task requests informally, but this approach disconnects the request from any structured tracking or processing system. A message sent through a chat carries no metadata, triggers no database entry, and requires someone to manually read, interpret, decide on an assignee, and record the outcome — a chain of manual steps that introduces delay and error at every stage.

This paper presents a platform that eliminates that manual chain entirely by routing WhatsApp task messages through a Discord bot to a Groq LLM, which selects the right worker from the Experts table, assigns a priority, and saves a structured record to the Supabase Task table in seconds. Beyond allocation, the platform delivers a complete operational environment — role-specific dashboards for Super Admin, Admin, and Worker, real-time task status tracking, pie chart analytics, and an inter-role chat system — making it a comprehensive solution for managing organizational work from receipt to completion.

II. LITERATURE REVIEW

Recent advancements in Artificial Intelligence have significantly influenced the development of intelligent task management and automated workforce allocation systems. The literature highlights the growing adoption of AI-driven decision engines and scheduling algorithms to automate task distribution in organizational environments. Rule-based and optimization-driven

approaches have demonstrated measurable improvements in reducing manual coordination effort and improving workload balance across teams. However, these systems typically require pre-structured inputs and lack the flexibility to handle task requests arriving through informal communication channels such as WhatsApp, limiting their practical applicability in real-world organizational settings. While individual components such as scheduling algorithms or notification systems are well explored, comprehensive AI-driven platforms that unify task ingestion, intelligent allocation, and real-time workforce visibility into a single automated workflow remain limited.

Large Language Models have demonstrated strong capabilities in understanding conversational text and producing structured allocation decisions with minimal latency, making them well suited for real-time organizational workflows. Transformer-based architectures effectively classify priority levels and identify suitable workers based on skill profiles stored in structured databases, replacing the need for hand-crafted rule sets. The integration of messaging platform bots with LLM inference engines has emerged as a practical approach for connecting informal communication channels with backend automation systems, significantly reducing adoption friction for non-technical users. Role-based access control frameworks built on cloud database platforms provide secure and domain-scoped data visibility essential for managing multi-role organizational systems. Dynamic workload management strategies that evaluate real-time worker availability and predicted task completion timelines further strengthen allocation accuracy, ensuring uninterrupted task flow without requiring supervisor involvement at any stage.

III. PROBLEM STATEMENT

The gap between how organizations receive task requests and how they track and manage those requests is a source of persistent inefficiency. Many teams still depend on supervisors manually reading WhatsApp messages, deciding on assignees based on personal familiarity, and recording outcomes informally. A supervisor occupied with other responsibilities may not see a task message for hours, causing delays that propagate through the entire delivery chain — the worker starts late, deadline pressure builds, and output quality suffers. Beyond delays, assignment decisions driven by personal judgment introduce inconsistency, where frequently visible workers receive a disproportionate share of urgent tasks while equally capable but less prominent team members are regularly overlooked. These patterns reinforce themselves over time, creating workload imbalances that are difficult to identify or correct without structured data.

When task assignments exist only within chat threads, there is no centralized view showing what every worker is handling, what remains pending, and what has already been completed. Supervisors who need performance insights must manually scroll through message histories and compile information by hand, making routine reporting a time-consuming exercise rather than an automated operation. The absence of priority flagging, skill-based matching, and real-time workload visibility means that high-value tasks can sit unattended while low-priority work gets handled first. The system presented in this paper addresses these failure points by replacing the informal manual loop with an automated pipeline built on LLM reasoning, structured Supabase storage, and a role-differentiated React frontend that makes the state of every task visible to every authorized user at all times.

IV. OBJECTIVE

The primary objective of this work is to design and develop an intelligent automated task allocation system that captures task messages submitted through WhatsApp via a Discord bot and processes them using Groq LLM without any manual intervention. The LLM evaluates the incoming task, assigns an appropriate priority level from Low, Medium, and High, and selects the most suitable worker from the Experts table based on skill match and current workload. The processed result is stored as a structured record in the Supabase Task table, populating the task description, priority, worker name, and worker email columns automatically. When all qualified workers are occupied, the system identifies the worker with the nearest task completion date and assigns the incoming task accordingly,

ensuring uninterrupted allocation flow at all times. The system also enforces role-based access control through Supabase Authentication, where Super Admin views domain-level cards for IT, HR, and Accounting, Admins access only their domain workers and tasks, and Workers see exclusively their own assigned tasks, with every task progressing through a defined lifecycle of Todo, In Progress, and Completed before being archived into the CompletedTask table.

To further strengthen the platform's operational value, the system delivers real-time bell notifications to workers when a new task is assigned and when any active task deadline falls within three days. A Report module provides organization-wide and domain-specific pie charts with date-range filtering accessible at the level appropriate to each role, giving leaders actionable insights into team performance and task completion trends. A Chat module enables real-time inter-role messaging between Super Admin, Admins, and Workers with full support for message reply, forward, edit, delete, and copy operations. An Organize Task section allows Super Admins and Admins to manually create task records directly in the Task table whenever the WhatsApp pipeline is temporarily unavailable, ensuring complete operational continuity under all conditions.

IV. SYSTEM ANALYSIS

System analysis is a crucial phase in the development of the proposed AI-driven task allocation platform, as it defines the functional and non-functional requirements of the system and evaluates existing limitations. The proposed system is designed to address the challenges associated with manual task assignment and informal workforce coordination by integrating LLM-based decision making, a WhatsApp-Discord bot pipeline, and a role-differentiated React frontend within a scalable cloud-based architecture.

A. Existing System

Existing task management practices in most organizations rely entirely on manual coordination, where supervisors receive task requests through WhatsApp messages and assign them based on personal judgment and informal awareness of team workloads. These systems lack any automated mechanism for skill-based worker matching, priority classification, or real-time task tracking. Even where basic digital tools are used, they require manual data entry, remain disconnected from communication platforms, and provide no centralized visibility into worker availability or task status across multiple organizational domains, making reporting and performance monitoring a time-consuming manual effort.

B. Proposed System

The proposed system introduces an intelligent automated task allocation platform that integrates a WhatsApp-Discord bot pipeline with Groq LLM to eliminate manual assignment entirely. When a task message is received, the LLM analyzes the content, assigns a priority level, selects the most suitable worker from the Experts table in Supabase, and writes a structured record containing the task, priority, worker name, and worker email into the Task table automatically. The platform enforces role-based access control for Super Admin, Admin, and Worker roles, each served by a tailored React dashboard with real-time task tracking, notifications, pie chart analytics, and inter-role chat, while completed tasks are automatically archived into the CompletedTask table.

C. System Architecture

The system architecture is organized across three interconnected layers. The Frontend Layer built using React and Vite provides role-specific dashboards, task view, and chat with notification functionalities for all user roles. The Application Layer handles core operations including AI-based task allocation using Groq LLM, role-based access control, and notification handling, processing incoming task requests and storing results in real time. The Database Layer powered by Supabase manages four key tables — Users, Tasks, Completed Tasks, and Messages — while WhatsApp integration serves as the external communication interface, forwarding incoming task messages into the application pipeline for automated processing and storage. The processed task record is then written directly into the Supabase Task table and instantly reflected across all role-based dashboards through real-time data synchronization, ensuring a scalable and

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715447

maintainable platform capable of handling continuous task allocation without any manual intervention.

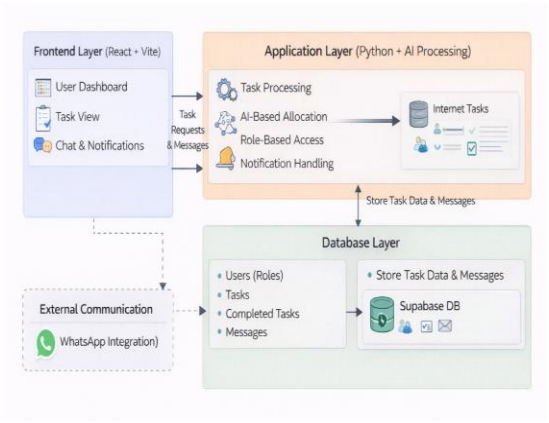

Fig. 1. System Architecture Diagram

D. Data Flow Diagram

The data flow begins when a user submits a task either through WhatsApp commands or manual entry via the Admin interface, which forwards the input to the Task Processing and AI Decision Engine. The engine performs priority classification and skill-based worker matching using Groq LLM, generating a structured task record that is stored into the Task Database containing Users, Tasks, Completed Tasks, and Messages tables in Supabase. The processed task data is then routed through the Role-Based Access Control layer, ensuring Super Admin, Admin, and Worker roles receive only the data relevant to their responsibilities. Validated task records are retrieved from the database and delivered as output in the form of real-time dashboard updates and bell notifications, keeping all authenticated users informed of task assignments and status changes instantly.

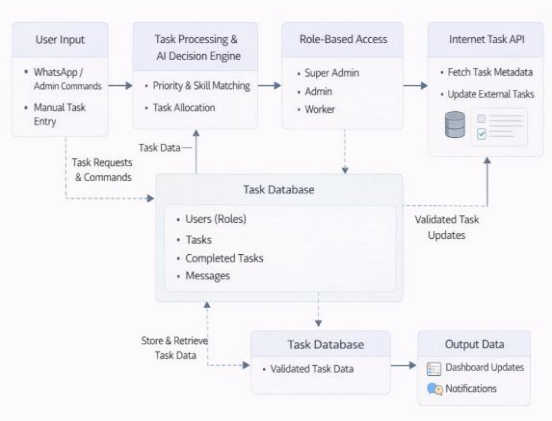

Fig. 2. Data Flow Diagram

VI. METHODOLOGY

The proposed AI-driven intelligent task allocation system follows a structured and modular methodology to transform incoming task requests into accurately assigned, tracked, and completed organizational tasks. The methodology is designed to ensure efficiency, scalability, and reliability by integrating a WhatsApp-Discord bot pipeline, Groq LLM inference, role-based access control, and a real-time React frontend within a unified and cohesive workflow.

A. WhatsApp / Organize Tasks Module

The system supports two task entry points to ensure continuous operation. In the primary route, a supervisor submits a task through WhatsApp, which forwards it into the processing pipeline. When the WhatsApp route is unavailable, the Super Admin or Admin can manually enter the task through the Organize Task section on the React frontend, ensuring both routes feed into the same downstream processing without interruption.

B. Discord Bot and Groq LLM Processing Module

Once the Discord bot receives the task message from WhatsApp, it forwards the content to the Groq LLM along with a structured prompt containing the organization's worker profiles and priority criteria. The LLM analyzes the task, assigns a priority level — Low, Medium, or High — and identifies the most suitable worker from the Experts table based on skill match. The resulting structured output containing the task description, priority, worker name, and worker email is then written directly into the Task table in Supabase by the Discord bot.

C. Worker Availability and Dynamic Assignment Module

Once the LLM selects the best-matching worker, the system checks their current availability. If the worker is free, the task is assigned immediately and a dashboard notification is triggered. If all qualified workers are occupied, the system examines the completion_date column across active tasks and assigns the incoming task to the worker whose deadline falls within the nearest three days, ensuring the task reaches the person most likely to become available soonest without any manual intervention.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715447

D. Role-Based Dashboard Module

Authentication via Supabase determines which dashboard a user sees upon login. The Super Admin dashboard displays three domain cards — IT, HR, and Accounting — each expandable to reveal individual worker cards showing their name and current task state. The Admin dashboard presents only the workers and task statuses belonging to their own domain, while the Worker dashboard lists all personally assigned tasks with description, priority, and deadline. All three dashboards refresh in real time as the underlying Supabase data changes.

E. Report Module

The Report module converts accumulated Task and CompletedTask data into visual pie chart summaries. Super Admins view an organization-wide chart along with separate domain-specific charts for IT, HR, and Accounting, all filterable by a date-range selector. Admins access only their domain's chart with the same filtering capability, while Workers see a personal chart reflecting their own task history and completion activity over time.

F. Chat Module

The Chat module stores all messages in a dedicated Supabase message table and delivers them through real-time subscriptions. Super Admins can message any domain Admin, while Admins can converse with the Super Admin, their own domain workers, and other domain Admins. Every conversation supports message reply, forward, edit, delete, and copy operations, all handled in-app without page reloads.

G. Settings Module

The Settings module allows any authenticated user to update three aspects of their account profile: their profile image, their preferred display name shown throughout the application, and their account password. Changes are saved to the relevant Supabase user record and take effect immediately across all interface elements that display the user's identity.

VII. RESULTS AND DISCUSSION

The completed platform was tested across all three role types to verify that each module behaved as designed. Task messages submitted through WhatsApp were consistently picked up by the Discord bot, processed by the Groq LLM, and inserted into the Task table with correct priority assignments and accurate worker selections within a few seconds of the original message being sent. Worker dashboards reflected new assignments immediately, and notification alerts appeared without requiring a page refresh. Task status updates made by Workers progressed correctly through the Todo, In Progress, and Completed stages, with completed task records automatically archived into the CompletedTask table and instantly reflected across Admin and Super Admin dashboards in real time.

A. Dashboard

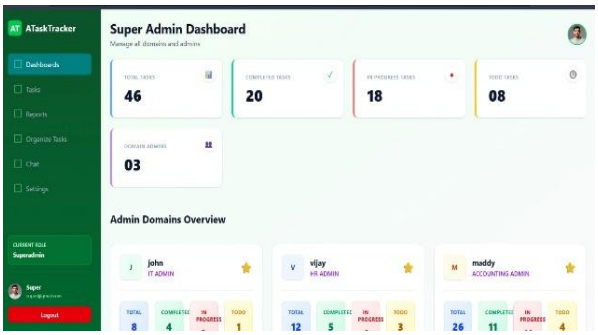

Fig. 3. System Dashboard

Super Admin dashboard presents domain-based admin cards for IT, HR, and Accounting, each expandable to reveal worker details and their current task states. Clicking any domain card displays the workers registered under that domain along with their active task information in real time.

B. Tasks

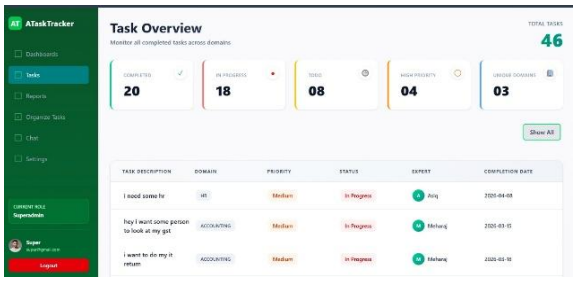

Fig 7.2 Tasks

The Task section displays all assigned tasks with their priority level and current status. Workers and

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715447

Admins can update the task status directly from Todo to In Progress and finally to Completed.

C. Reports

Fig. 4. Reports

The Report section displays organization-wide and domain-specific pie charts summarizing task distribution and completion rates. Super Admins, Admins, and Workers can filter the charts by date range to analyse task activity within any selected period.

D. Chat Interface

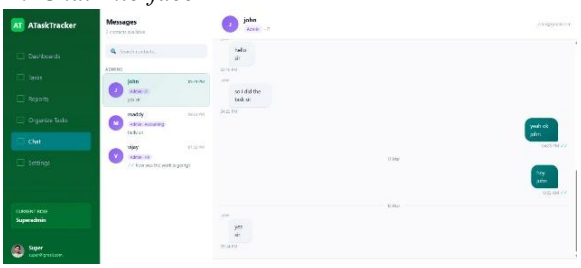

Fig. 5. Chat

The Chat module enables real-time messaging between all roles stored in a dedicated Supabase message table. Super Admins can communicate with all domain Admins, while Admins can converse with their domain workers and other Admins. Every conversation supports message reply, forward, edit, delete, and copy operations within the application

VIII. CONCLUSION

The proposed AI-driven intelligent task allocation system successfully demonstrates that combining a familiar WhatsApp messaging interface with a Discord bot pipeline and Groq LLM reasoning produces an allocation engine that operates faster, more consistently, and more fairly than any manual assignment process. The system takes over completely from the moment a task is submitted — analyzing the content, selecting the right worker, assigning priority, and recording the structured result in Supabase — without requiring any further human action from the supervisor.

The role-differentiated React frontend ensures that every user receives information in the form most relevant to their responsibilities. Workers are notified instantly upon assignment and reminded before deadlines approach, Admins maintain clear domain-level visibility, and the Super Admin oversees the entire organization from a single dashboard. The reporting, chat, and task archival modules collectively extend the platform into a complete operational environment for managing organizational work from initial receipt through final completion.

IX. LIMITATIONS

The proposed AI-driven intelligent task allocation system, while effective in demonstrating automated task assignment through WhatsApp-Discord bot integration and Groq LLM inference, is subject to certain limitations that should be acknowledged. The accuracy of task allocation is highly dependent on the clarity and completeness of the task message submitted through WhatsApp. Vague or ambiguous task descriptions may cause the LLM to make incorrect priority assignments or select a less suitable worker, directly impacting the reliability of the allocation decision and the overall quality of task distribution across the organization.

The system currently relies on the Groq LLM API for intelligent task processing and worker selection, which introduces dependency on external service availability and stable internet connectivity. In scenarios involving high volumes of simultaneous task submissions or network instability, API latency may increase response times and affect the real-time performance of the allocation pipeline. Furthermore, the accuracy of worker selection is inherently constrained by the quality and completeness of skill profile data maintained in the Experts table within Supabase.

The fallback allocation strategy, while ensuring continuous task flow when all workers are occupied, assigns tasks based solely on the nearest completion date without considering task complexity or worker capacity beyond current assignments. Finally, the system has been evaluated under moderate organizational conditions and has not been tested at large-scale enterprise levels involving hundreds of concurrent users and simultaneous task submissions, indicating that further infrastructure optimization would be necessary before deployment in high-demand production environments.

IX. FUTURE ENHANCEMENTS

The proposed AI-driven intelligent task allocation system can be further enhanced by incorporating advanced techniques and additional functionalities to improve its performance, scalability, and usability across diverse organizational environments. Future improvements may include training the Groq LLM on historical task allocation records, enabling the model to learn from past assignment outcomes and progressively improve worker matching accuracy and priority classification over time. The system can also be extended to support additional task ingestion channels such as Telegram, email, and internal web forms, enabling wider accessibility without requiring any changes to the existing downstream processing pipeline.

Developing a mobile-native version of the React frontend would significantly improve accessibility for workers who primarily operate from smartphones, allowing them to view assignments, update task statuses, and receive notifications on the go. Integration with calendar applications would enable automatic synchronization of task deadlines with workers' personal schedules, reinforcing the existing in-app notification system with a channel users already monitor regularly. The Report module can be enhanced with individual worker performance trend lines, domain-to-domain productivity comparisons, and predictive workload forecasting capabilities to support forward planning and strategic decision-making beyond retrospective reporting.

The AI allocation engine can be further strengthened by incorporating workload complexity scoring, allowing the LLM to consider not only current task count but also the estimated effort required for each active assignment when selecting workers. Implementing anomaly detection mechanisms would help identify unusual workload patterns, consistently overloaded workers, or recurring missed deadlines before they escalate into larger operational issues. Deployment on scalable cloud infrastructure with advanced analytics dashboards and enhanced security policies would further improve system performance, governance, and user experience across large-scale enterprise environments with high volumes of concurrent users and simultaneous task submissions.

REFERENCES

[1] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 5998–6008, 2017.

[2] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.

[3] A W. X. Zhao et al., "A Survey of Large Language Models," IEEE Transactions on Neural Networks and Learning Systems, vol. 14, no. 6, pp. 1–28, 2023.

[4] D. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed Standard for Role-Based Access Control," ACM Transactions on Information and System Security, vol. 4, no. 3, pp. 224–274, 2001.

[5] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed., McGraw-Hill, New York, 2020.

[6] P. Brucker, Scheduling Algorithms, 5th ed., Springer, Berlin, 2007.

[7] M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed., Wiley, Chichester, 2009.

[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, Cambridge, MA, 2016.

[9] G. Adomavicius and A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible

Extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, 2005.

[10] J. Smith and T. Brown, "Automated Task Management Systems Using Machine Learning Techniques," International Journal of Computer Applications, vol. 178, no. 10, pp. 1–7, 2020.

[11] A. Radford et al., "Language Models are Unsupervised Multitask Learners," OpenAI Technical Report, vol. 1, no. 8, pp. 1–24, 2019.

[12] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Pearson, Upper Saddle River, NJ, 2021.

[13] C. C. Aggarwal, Artificial Intelligence: A Textbook, Springer, Cham, 2021.

[14] K. Schwaber and J. Sutherland, "The Scrum Guide: The Definitive Guide to Scrum — The Rules of the Game," Scrum.org, pp. 1–19, 2020.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in Proceedings of the International Conference on Learning Representations (ICLR), pp. 1–12, 2013.