IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

Intelligent Hybrid Task Management System Using Dynamic Priority Scoring and Generative AI Task Decomposition

Haripriya D[1], Komal[2], L Rakshitha[3], M R Pavithra[4], Mr. Isanka Sai Praneet[5]

Don Bosco Institute of Technology Visvesvaraya Technological University, Bengaluru, India[1,2,3,4]

Assistant Professor, Computer Science and Engineering, Don Bosco Institute of Technology, Bengaluru, India[5]

Abstract: Traditional task managers rely on fixed Low/Medium/High priority labels, offer no step-by-step plan for complex tasks, and allow repeatedly postponed tasks to be ignored or forgotten. This paper presents the Intelligent Hybrid Task Management System, a web-based productivity application that combines a rule-based heuristic engine with Generative AI. The heuristic engine dynamically calculates a priority score from deadline proximity, urgency, effort and postponement history, and detects overlapping schedules. Google Gemini decomposes complex tasks into smaller subtasks with estimated durations. The application is built with React, TypeScript, Vite and Tailwind CSS, uses Supabase (PostgreSQL) for authentication, cloud storage and real-time synchronization, and falls back to browser LocalStorage so the application keeps working offline. A Recharts-based analytics module provides productivity insights. The resulting prototype gives dynamic rather than static prioritization, AI-assisted planning and offline resilience. Future work targets rigorous user evaluation, adaptive weight learning and calendar integration.

Keywords: Task Management, Dynamic Priority Scoring, Generative AI, Google Gemini, Task Decomposition, Schedule Conflict Detection, Supabase, Offline-First, React, TypeScript.

I. INTRODUCTION

People now juggle academic, professional and personal commitments across many deadlines, and productivity applications have become a core part of daily planning. Most of them, however, treat a task as a title, a due date and a manually chosen priority label. Such static labels do not change as deadlines approach, as effort accumulates, or as a task is pushed back again and again.

Generative AI offers a way to go beyond static lists. Large language models can read a task description written in natural language and propose a structured plan, but on their own they lack the predictable, explainable behaviour that people expect from a scheduling tool. Deterministic heuristics, on the other hand, are transparent and fast but cannot understand the content of a task.

In this work, an intelligent task management system is developed that combines both approaches. A rule-based priority engine evaluates every task dynamically using deadline, urgency, effort and postponement history. Google Gemini is used only where language understanding is needed: breaking a complex task into subtasks, estimating durations and suggesting continuation steps. A conflict detection module identifies overlapping schedules, and a hybrid data layer synchronizes tasks to the cloud while keeping a local copy for offline use.

The main contributions of this work are threefold. First, a transparent priority score that is recalculated from deadline proximity, urgency, effort and postponement history, so that rankings change as circumstances change. Second, an AI-assisted planning module in which Google Gemini turns a broad task into reviewable subtasks with estimated durations, while the user keeps the final decision. Third, a hybrid cloud and local data layer that keeps the application usable offline and synchronizes when a connection returns.

The proposed system aims to help users decide what to do first, make large tasks approachable, and keep postponed work visible. The remainder of this paper reviews related work, summarizes the findings of the survey, describes the implementation and methodology, presents the prototype results, and concludes with future scope.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

II. LITERATURE SURVEY

Xu et al. (2026) published a survey of personalized LLM-powered agents and organized the field around four capabilities: profile modeling, memory, planning and action execution. For planning, they separate one-shot planning, which is fast but can be brittle when user signals are sparse or change, from feedback-driven planning, which fits the user better through clarification and user edits but costs more latency and effort. They treat user edits as a strong personalization signal and list privacy, user control and efficient deployment as open problems. This supports the design used here, where Gemini proposes subtasks that the user reviews and edits, and where data stays under the user's control in local storage [1].

Jain and Wetter (2025) introduced R-ConstraintBench, which tests large language models on Resource-Constrained Project Scheduling Problems, a class of scheduling problems that is NP-complete. The benchmark starts with precedence constraints between tasks and then adds downtime, time windows and mutual-exclusion constraints. Strong models were close to perfect on precedence-only graphs, but their feasibility dropped sharply when downtime, time windows and exclusion constraints interacted. The authors conclude that interaction between constraints, not the depth of the task graph, is the main bottleneck, and that good results on synthetic cases do not guarantee transfer to realistic ones. This is a reason to keep schedule conflict detection in deterministic code rather than leave it to the language model [2].

Wei et al. (2025) presented PlanGenLLMs, a survey of the planning capabilities of large language models. The survey builds on work in AI planning and evaluates existing methods against five criteria: completeness, executability, optimality, representation and generalization. It reviews how language models translate natural-language tasks into formal representations such as PDDL, how they work with classical planners and search algorithms, and how task decomposition and fine-tuning improve planning. The review shows that hybrid systems, in which a language model supports deterministic planning logic, are an established direction, which matches the choice made here to pair Gemini with a rule-based engine. The survey itself is theoretical and does not describe a personal task manager [3].

Zheng et al. (2024) introduced NATURAL PLAN, a planning benchmark written in natural language with three tasks: trip planning, meeting planning and calendar scheduling. Tool outputs from Google Flights, Google Maps and Google Calendar are supplied as context, so only the planning ability is tested. The benchmark proved hard for the models tested: GPT-4 and Gemini 1.5 Pro solved 31.1% and 34.8% of the trip planning problems, and accuracy fell as the number of people, cities or days grew. Self-correction did not help, and with long context Gemini 1.5 Pro reached about 50% on calendar scheduling. The results show that language models alone are not yet reliable for scheduling, which supports the hybrid design of the present system [4].

Huang et al. (2024) surveyed the planning ability of large language model agents. They note that earlier planning work relied on symbolic methods such as PDDL or on reinforcement learning, which often lack error tolerance, and that language models marked a paradigm shift. The survey proposes a taxonomy of five directions: task decomposition, multi-plan selection, external module-aided planning, reflection and refinement, and memory-augmented planning. Task decomposition follows a divide-and-conquer idea and can be done decomposition-first, where all subgoals are produced before planning, or interleaved with execution. The survey lists the added overhead and the context length limit of the model as challenges. The present system uses the decomposition-first style, as Gemini returns a full set of subtasks that the user reviews [5].

The Gemini Team at Google (2023) introduced Gemini, a family of models that are multimodal from the start and can understand text, images, audio and video. The first version, Gemini 1.0, comes in three sizes: Ultra for highly complex tasks, Pro for performance and deployability at scale, and Nano for on-device use. The report states that Gemini Ultra advanced the state of the art in 30 of the 32 benchmarks examined. Developers can reach the models through the Gemini API in Google AI Studio and Cloud Vertex AI, which is how an application like this one can call Gemini. Because Gemini is a general-purpose model that was not tuned for personal scheduling, the suggestions it makes for subtasks and durations are shown to the user for review before they are added [6].

Zhang et al. (2022), from Brandeis University and Microsoft Research, formulated task prioritization as a ranking problem over a user's pending tasks, given the history of previous interactions with a task management application. They analysed large-scale anonymized and de-identified logs of a popular task management application and derived a dataset of grounded, real-world tasks. The analysis also showed that the way people mark tasks as complete varies consistently with the nature of the task. The authors then proposed a contextual deep learning model for personalized prioritization, which outperformed several operational baselines and other sequential ranking models in their tests. The

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

study shows that users benefit from automatic prioritization, but its model depends on historical interaction data and does not break tasks into subtasks, which the present system addresses with Gemini [7].

Kleppmann et al. (2019) proposed local-first software in an essay presented at the ACM Onward! symposium. They point out that cloud applications give easy multi-device access and collaboration, but all data access goes through the server, so the user cannot use the software or reach their data if the service is unavailable. In local-first applications the copy of the data on the user's own device is the primary copy, and the cloud is used for synchronization. The authors set out seven ideals, among them fast response without waiting for the network, access on multiple devices, and work that continues offline. The essay is conceptual and does not describe a task manager. The present system applies the same idea in a simpler form, with LocalStorage as the offline copy and Supabase as the synchronized store [8].

Myers et al. (2007) described an intelligent personal assistant built to help a busy knowledge worker manage time commitments and perform tasks. Its design had two aims: to relieve the user of routine tasks, and to step in when cognitive overload leads to oversights or mistakes. The system combines several AI technologies inside a Belief-Desire-Intention agent framework and is user-centric, responsive to user input and adaptive to the user's working style. It covers time management, such as appointment scheduling, reminder generation and workload balancing, and task management, meaning the planning, execution and oversight of tasks. These goals are close to those of the present work, but the system predates large language models and offers no natural-language task understanding or decomposition [9].

Bellotti et al. (2004), at the Palo Alto Research Center, studied how people record, organize and complete their to-dos in order to design a personal task list manager. Their work included a study in which participants counted the to-dos they carried and the resources they used to track them, and a longer study that followed seven professionals for a month. They found that people are not poor at prioritizing, as is often assumed, but have well-developed strategies for particular task management challenges. They also observed that predictions of how long a task will take are often inaccurate. The studies support giving users a flexible task list with time and importance support, and they are a reason why the duration estimates in this project are only suggestions that the user can change [10].

Year	Author / Paper	Method / Technique Used	Dataset / Environment	Results / Findings	Limitations
2026	Kaustubh Gulwade et al. – AI-Powered Personal Assistant for Smart Task Scheduling, Email Deadline	Large language model combined with productivity APIs	Email and calendar data via productivity APIs	Summarises emails, extracts action items and deadlines, creates calendar events, and answers questions about email content	Enron emails may not reflect current writing styles, results depend on the Gemini API (model drift), and third-party API calls add cost and latency. Errors came from vague time phrases, multi-task emails and implied intent.
2025	Ioannis Refanidis, Anastasios Alexiadis – Deployment and Evaluation of SelfPlanner, an Automated Individual Task Management System	Automated task scheduling in a web-based intelligent calendar app (SelfPlanner), which considers task constraints instead of leaving scheduling to the user	Web-based calendar application	Schedules individual tasks in time and space automatically	From an earlier SelfPlanner paper by the same authors (not this one): the algorithm is incomplete, so it can't guarantee a solution even if one exists, and tight schedules may force changes to existing plans. Meeting scheduling is out of scope.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

2025	Karen Myers, Pauline Berry et al.. – An Intelligent Personal Assistant for Task and Time Management	Intelligent personal assistant for managing tasks and time commitments (the rest of the text is cut off)	PExA, built as part of the CALO project, an adaptive cognitive assistant for an office environment	PExA already offered task performance, time management, execution monitoring and team coordination, though the project was only midway through its five years	I couldn't open the full text (the publisher site blocked access), so I can't confirm the authors' own limitations section. It is a system-description paper with no controlled experiment in the parts I could see.
2024	Thakur et al. – Task Management System Using AI Prioritisation	Random Forest predicts task priority, Linear Regression predicts deadlines and workload, and KNN does collaborative filtering for personalised recommendations	Random Forest predicts task priority, Linear Regression predicts deadlines and workload, and KNN does collaborative filtering for personalised recommendations	The authors report promising prioritisation accuracy and response time, and compare the F1 scores of the three algorithms, but the numbers appear only in figures	The authors don't state limitations. My own observations: no dataset size or numeric results in the text, and the time-series deadline forecasting is proposed but not evaluated.
2024	X. Huang et al. – Understanding the Planning of LLM Agents: A Survey	Literature survey with a taxonomy: task decomposition, multi-plan selection, external modules, reflection, memory	Published LLM agent planning methods; no dataset of its own	Decomposition can be decomposition-first or interleaved; each has trade-offs	Decomposition adds overhead and is limited by context length
2023	Gemini Team, Google – Gemini: A Family of Highly Capable Multimodal Models	Transformer-based multimodal models trained for text, image, audio and video in three sizes: Ultra, Pro, Nano	Evaluated on language, code, reasoning and multimodal benchmarks; available through Google AI Studio and Vertex AI	Gemini Ultra advances the state of the art in 30 of 32 benchmarks examined	General-purpose model; suggestions need user review
2022	C. Zhang et al. – Grounded Task Prioritization with Context-Aware Sequential Ranking	Contextual deep learning model that ranks a user's pending tasks from their interaction history	Anonymized, de-identified logs of a popular task management application	Outperforms operational baselines and earlier sequential ranking models	Needs historical interaction data; no subtask decomposition
2019	M. Kleppmann et al. – Local-First Software: You Own Your Data, in Spite of the Cloud	Local-first design principles (seven ideals): the local copy is the primary copy, with sync and CRDT-based collaboration	Conceptual essay comparing cloud apps with local-first software; no dataset	Keeps multi-device cloud access and adds offline use and data ownership	Conceptual essay; no task-management implementation

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

2007	V. Bellotti et al. – What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manag	Belief-Desire-Intention agent combining several AI technologies for time and task management	Office environment, as a component of the CALO assistant project	Agent-based assistant that supports scheduling and tasks and adapts to the user	Predates LLMs; no natural-language task understanding or decomposition
2004	V. Bellotti et al. – What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager	Ethnographic studies: counting and classifying to-dos, and a month of weekly interviews and shadowing	Workplace studies at PARC with office professionals, including seven tracked for a month	People prioritize with their own strategies; task time predictions are often inaccurate	Small participant groups; predates mobile and AI assistance

III. FINDINGS

The surveyed studies show that people prioritize with their own informal strategies and often misjudge how long a task takes [10], and that task assistants and learned ranking models were later built to take over part of this work [7], [9]. The learned model outperformed its baselines but depends on a history of user interactions and does not break tasks into steps [7].

Recent surveys show that language models can decompose goals into subtasks and plan, and that user feedback and personalization improve the result [1], [3], [5]. Benchmarks, however, show that model accuracy falls on calendar and scheduling problems as constraints grow, especially when constraints interact [2], [4]. This supports a division of labour in which Gemini handles language-heavy decomposition and rule-based code handles scoring and conflict detection.

The survey also shows a gap: the reviewed work either learns a ranking from interaction logs, benchmarks language-model planning, or describes an assistant, but does not combine a transparent dynamic priority score, AI task decomposition and offline-first synchronization in one lightweight web application. Fixed priority labels and tasks that are repeatedly postponed are also not addressed by the reviewed systems.

The overall conclusion is that a hybrid system, combining transparent heuristic algorithms with Generative AI [6] and a local-first data layer [8], is a practical way to deliver dynamic prioritization, AI-assisted planning and offline resilience.

IV. IMPLEMENTATION

The application is implemented as a web application in TypeScript, with a cloud backend provided by Supabase and an AI service provided by Google Gemini.

- **Frontend:** The user interface is developed with React and TypeScript, built with Vite and styled with Tailwind CSS. Users create and manage tasks in natural language or through a form and view insights on a dashboard.
- **Task Service:** Handles task creation, editing, completion and state management.
- Heuristic (Priority) Engine: Calculates a dynamic priority score for each task and performs schedule conflict detection and reminder scheduling.
- **AI Integration:** Complex tasks are sent to Google Gemini as a request. The response contains subtasks, estimated durations, suggestions and continuation recommendations, which the user can review and add as tasks.
- **Analytics Engine:** Produces productivity insights and visualizations with Recharts, such as task volume by category and priority level distribution.
- **Hybrid Data Layer:** Supabase provides user authentication, a PostgreSQL database and real-time synchronization. LocalStorage keeps data on the device for offline use and persistence, and tasks are synchronized when the user is online.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

The core logic is expressed in the following simplified form.

```
let score = 10;                                    // baseline
if (diffHours &lt;= 12)
      score += 35;
else if (diffHours &lt;= 24)
      score += 30;
if (postponeCount &gt; 0)
      score += Math.min(15, postponeCount * 4);
score = Math.min(100, Math.max(5, score));

const overlapStart = Math.max(aStart, bStart);
const overlapEnd = Math.min(aEnd, bEnd);

if (overlapStart &lt; overlapEnd) {
conflicts.push({ taskA, taskB, overlapMinutes: overlapEnd - overlapStart });
}
```

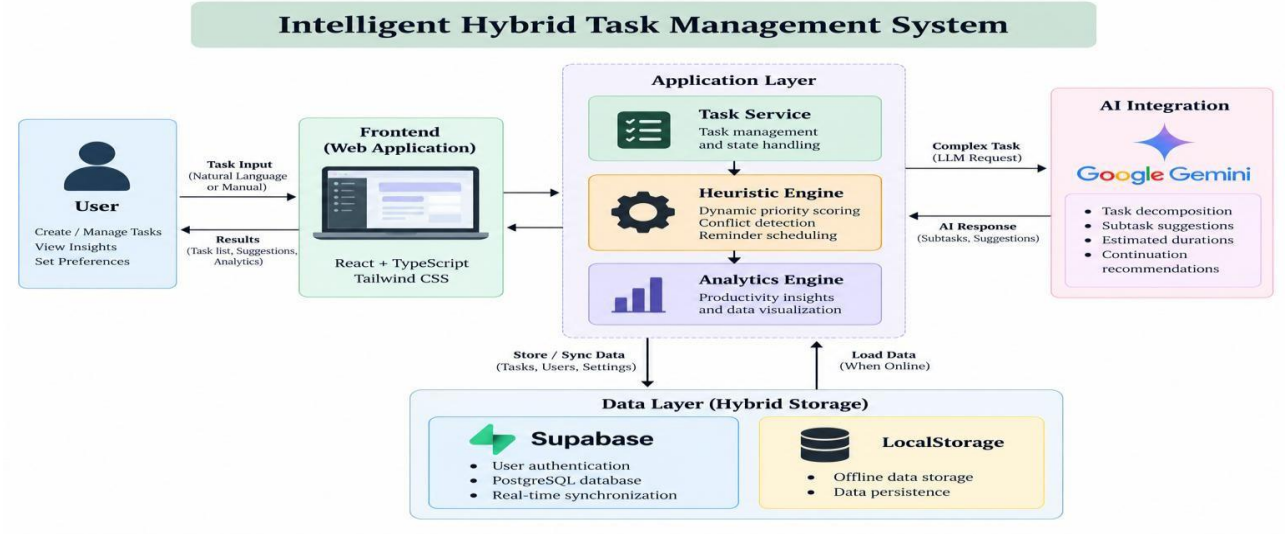


Fig. 1 Implementation architecture

V. METHODOLOGY

The proposed methodology delivers dynamic, explainable prioritization on the client and reserves the AI model for tasks that benefit from language understanding.

- **Task Input and Profiling:** The user enters a task with its deadline, effort and category. The system also tracks how many times the task has been postponed.
- **Dynamic Priority Scoring:** Each task starts from a baseline score of 10. Points are added as the deadline approaches: 35 points when the remaining time is 12 hours or less, and 30 points when it is 24 hours or less. Postponement adds 4 points per postponement up to a maximum of 15, so that repeatedly delayed tasks rise in the ranking instead of being forgotten. The final score is bounded between 5 and 100. Urgency and effort are also taken into account, and the score is recalculated whenever task data changes.
- **AI-Based Task Decomposition:** For complex tasks, the title and description are sent to Gemini, which returns a set of subtasks with estimated durations. The user reviews the proposal and adds the selected subtasks, so the AI assists but does not act without confirmation.
- **Schedule Conflict Detection:** For every pair of scheduled tasks, the overlap window is computed as the later of the two start times to the earlier of the two end times. If the start of this window is before its end, the pair is flagged together with the number of overlapping minutes.
- **Hybrid Data Management:** Tasks are written to local storage first and synchronized with Supabase when a connection is available, so the user can continue working offline.
- **Productivity Analytics:** Stored task data is aggregated into charts and cognitive-pattern summaries, such as which tasks are postponed most often.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

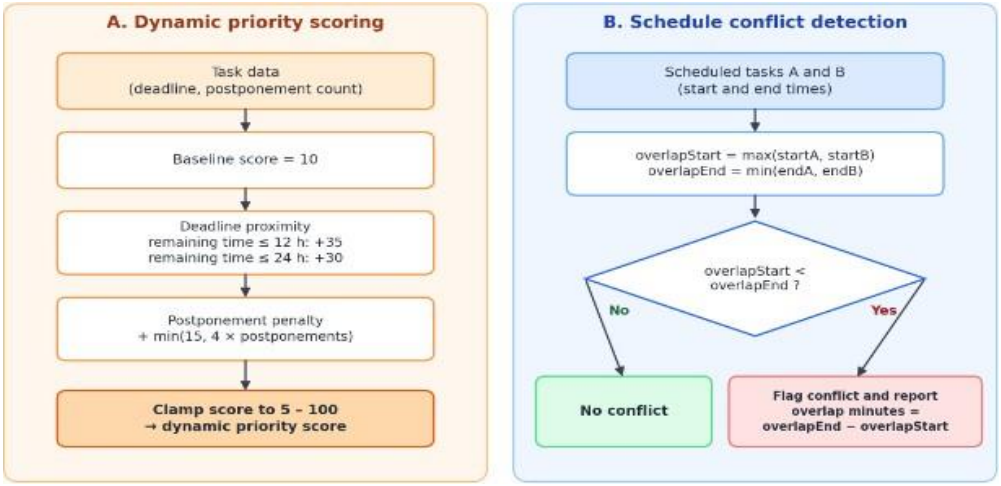
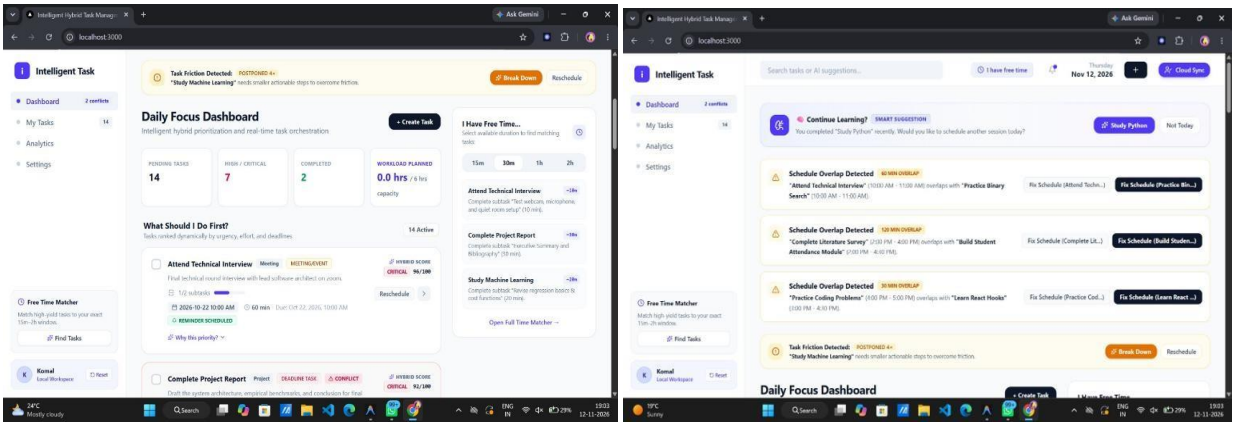


Fig. 2 Dynamic priority scoring and schedule conflict detection

VI. RESULT

The prototype was developed as a working web application and tested locally. Its main screens are described below.

- **Authentication and Cloud Sync:** A sign-in and sign-up screen connects the user to cloud synchronization through Supabase.
- **Daily Focus Dashboard:** The dashboard answers the question of what the user should do first. It shows summary counts of tasks, ranks tasks by dynamic priority with labels such as critical, flags schedule conflicts, and offers a focus timer and an option to reschedule.
- **AI Task Breakdown Assistant:** The user enters a goal, for example learning a topic, and the assistant generates subtasks with estimated durations that can be accepted or edited individually.
- **Analytics:** Charts show task volume by category and priority level distribution, together with AI-synthesized observations about the user's task patterns.

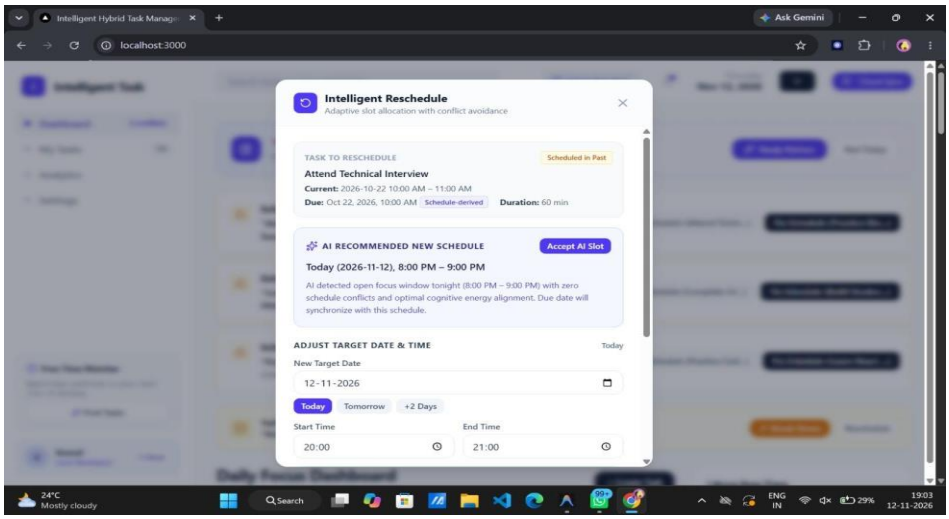


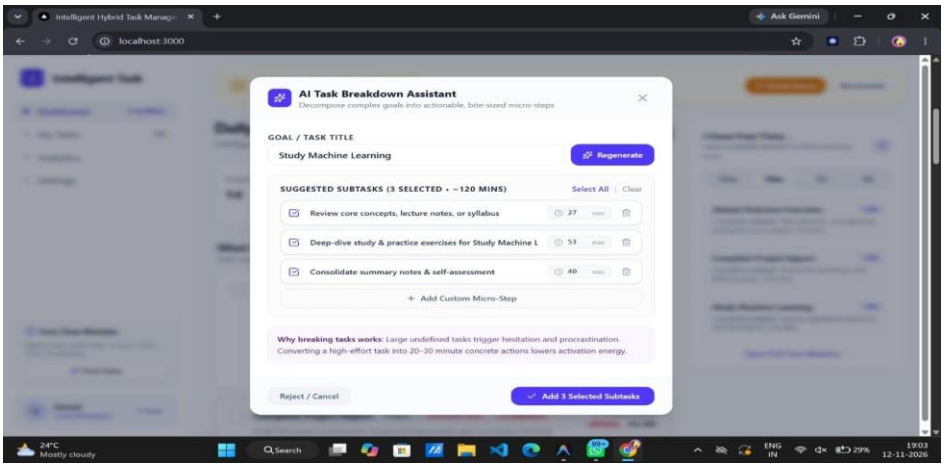
IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

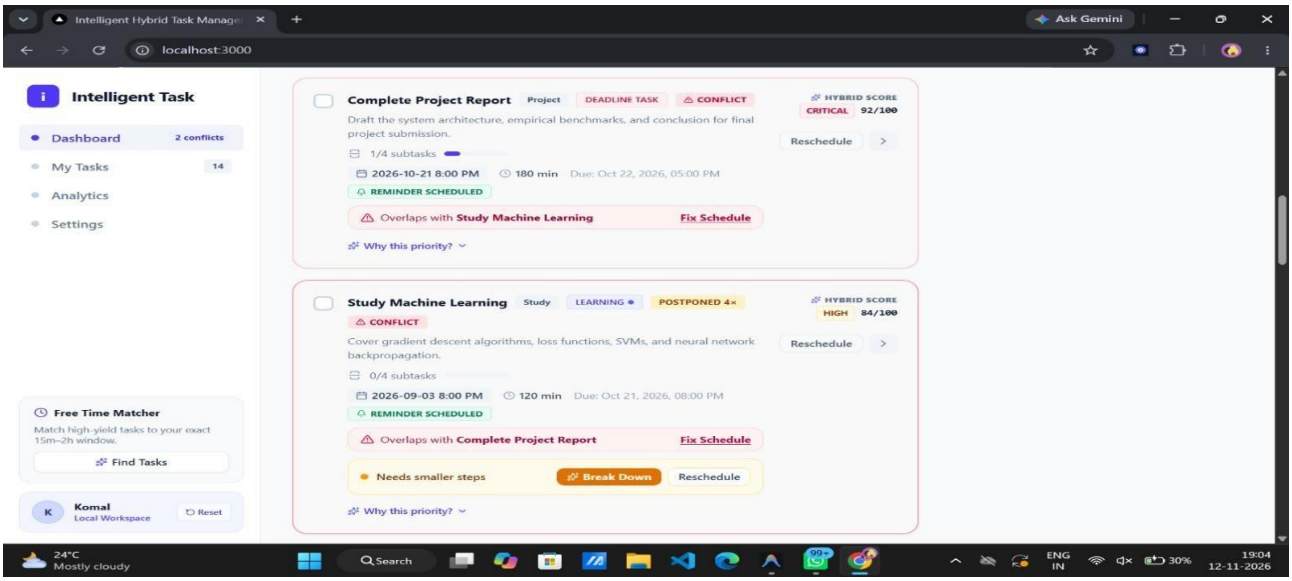
International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185







© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

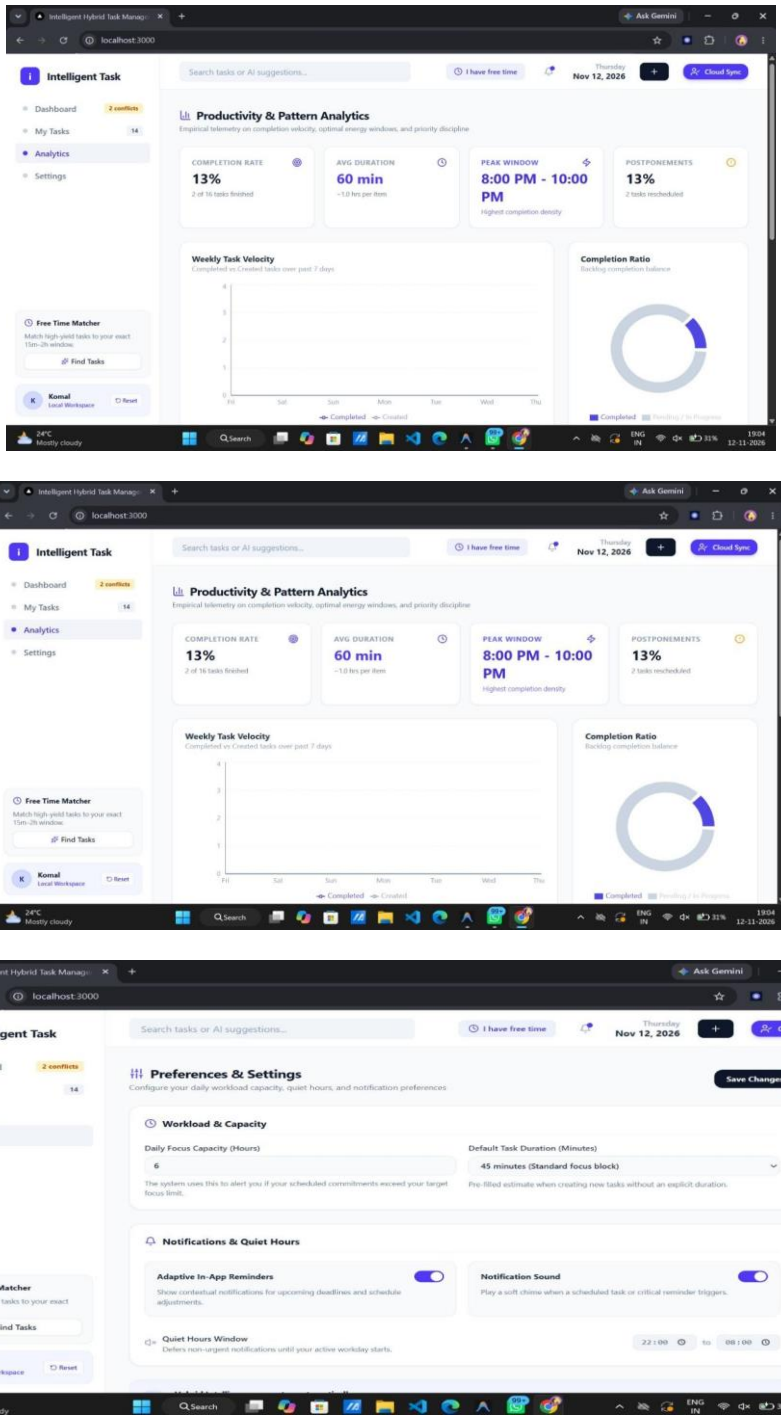


Fig. 3 Prototype screens of the Intelligent Hybrid Task Management System

VII. RESULT ANALYSIS

The prototype demonstrates the following behaviour.

- **Dynamic versus static prioritization:** Because the score is recomputed from the time remaining and the postponement count, a task rises in the ranking as its deadline nears or as it is postponed, without the user changing any label. A fixed Low/Medium/High scheme cannot do this.
- **Task decomposition:** A single broad task is converted into several concrete subtasks with time estimates, giving the user a clear starting point.
- **Conflict handling:** Overlapping tasks are reported with the number of overlapping minutes, so the user can reschedule before a clash occurs.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 11, November 2026

DOI: 10.17148/IARJSET.2026.131185

- **Offline resilience:** Since data is held in LocalStorage and synchronized when online, the application remains usable without a connection.

VIII. CONCLUSION

This paper presented an Intelligent Hybrid Task Management System that provides dynamic rather than static task prioritization, combines heuristic algorithms with Generative AI, simplifies complex tasks through AI-powered decomposition and offers cloud synchronization with offline resilience. Keeping prioritization and conflict detection rule-based makes the system transparent and fast, while Gemini is used only for language-heavy planning. The reported behaviour has been observed on a prototype and should be validated through user studies and on larger task sets.

IX. FUTURE SCOPE

Future work includes learning the priority weights from user behaviour, integrating calendar services so that conflicts are detected against external events, adding reminders and notifications on mobile devices, supporting team collaboration and shared tasks, and conducting a formal usability evaluation with real users.

REFERENCES

[1]. K. Gulwade et al., “AI-Powered Personal Assistant for Smart Task Scheduling, Email Deadline,” International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS), vol. 15, no. 1, pp. 265–274, 2026, doi: 10.51583/IJLTEMAS.2026.150100020.

[2]. I. Refanidis and A. Alexiadis, “Deployment and Evaluation of SelfPlanner, an Automated Individual Task Management System,” Computational Intelligence, vol. 27, no. 1, pp. 41–59, 2025, doi: 10.1111/j.1467-8640.2010.00371.x.

[3]. K. Myers et al., “An Intelligent Personal Assistant for Task and Time Management,” AI Magazine, vol. 28, no. 2, pp. 47–61, 2025, doi: 10.1609/aimag.v28i2.2039

[4]. R. Thakur, N. Manikandan, G. Kumar S, and R. Jayaraj, “Task Management System using AI Prioritizations,” International Journal of Novel Research and Development (IJNRD), vol. 9, no. 3, pp. e1–e6, 2024. [5] X. Huang et al., "Understanding the Planning of LLM Agents: A Survey," arXiv:2402.02716, 2024.

[5]. Gemini Team, Google, "Gemini: A Family of Highly Capable Multimodal Models," arXiv:2312.11805, 2023.

[6]. C. Zhang, J. Kiseleva, S. K. Jauhar, and R. W. White, "Grounded Task Prioritization with Context-Aware Sequential Ranking," ACM Transactions on Information Systems, vol. 40, no. 4, pp. 1–28, 2022, doi: 10.1145/3486861.

[7]. M. Kleppmann, A. Wiggins, P. van Hardenberg, and M. McGranaghan, "Local-First Software: You Own Your Data, in Spite of the Cloud," in Proc. ACM SIGPLAN Int. Symp. on New Ideas, New Paradigms, and Reflections on Programming and Software (Onward!), 2019, pp. 154–178, doi: 10.1145/3359591.3359737.

[8]. K. Myers et al., "An Intelligent Personal Assistant for Task and Time Management," AI Magazine, vol. 28, no. 2, pp. 47–61, 2007, doi: 10.1609/aimag.v28i2.2039.

[9]. V. Bellotti, B. Dalal, N. Good, P. Flynn, D. G. Bobrow, and N. Ducheneaut, "What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager," in Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI), 2004, pp. 735–742, doi: 10.1145/985692.985785.

[10]. K. Valmeekam, M. Marquez, S. Sreedharan, and S. Kambhampati, "On the Planning Abilities of Large Language Models – A Critical Investigation," in Advances in Neural Information Processing Systems (NeurIPS), vol. 36, 2023.

[11]. D. Zhou et al., "Least-to-Most Prompting Enables Complex Reasoning in Large Language Models," in Proc. Int. Conf. on Learning Representations (ICLR), 2023.

[12]. J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in Advances in Neural Information Processing Systems (NeurIPS), vol. 35, 2022, pp. 24824–24837.

[13]. P. Steel, "The Nature of Procrastination: A Meta-Analytic and Theoretical Review of Quintessential Self-Regulatory Failure," Psychological Bulletin, vol. 133, no. 1, pp. 65–94, 2007, doi: 10.1037/0033-2909.133.1.65.

[14]. C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," Journal of the ACM, vol. 20, no. 1, pp. 46–61, 1973, doi: 10.1145/321738.321743.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License