International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 11 November-2026 | Impact Factor: 3.5



PlanWise: An AI-Powered Personal and Collaborative Task Management System

Dnyandeep Bendewar
Department of Computer Science and Engineering
Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati
Amravati, Maharashtra, India
bendewardnyandeep@gmail.com

Bhaumik Dhore
Department of Computer Science and Engineering
Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati
Amravati, Maharashtra, India
bhamuik.dhore@gmail.com

Bhumika Baitule
Department of Computer Science and Engineering
Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati
Amravati, Maharashtra, India
bhumikabaitule@gmail.com

Radha Kokate
Department of Computer Science and Engineering
Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati
Amravati, Maharashtra, India
radhakokate302005@gmail.com

Dr. S. P. Ingale
Department of Computer Science and Engineering
Prof. Ram Meghe Institute of Technology and Research, Badnera-Amravati
Amravati, Maharashtra, India
spingale@mitra.ac.in

How to Cite this Article:
Bendewar, D., Dhore, B., Baitule, B. & Kokate, R. (2026). PlanWise: An AI-Powered Personal and Collaborative Task Management System. International Journal of Creative and Open Research in Engineering and Management, 02(11), 1-9. https://doi.org/10.55041/ijcope.v2i11.102

License:
This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.
© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.







https://doi.org/10.55041/ijcope.v2i11.102

***Abstract*—Effective task and project management needs efficient task prioritization, scheduling, progress monitoring, and coordination among people, all things considered. Conventional task management systems mostly rest on manual planning and tracking, which can make it difficult for users to manage competing priorities and changing workloads. This paper introduces PlanWise, an AI-powered personal and collaborative task management system designed to support task organization, project coordination, and intelligent productivity assistance, at least in principle. A web-based interface with backend services and database management is being blended by this setup to support personal tasks and collaborative project workflows, and includes large language model-based AI assistance for daily planning, task guidance, and project intelligence, combining task-related information and deterministic processing with AI-generated explanations and recommendations. It also is stressing user control by requiring approval for AI-suggested changes before they're employed. PlanWise is built using React and Vite for the frontend and Supabase for authentication, database operations, and backend functionality, with Google Gemini API integration for AI-assisted attributes. This paper is outlining the system architecture, design methodology, implementation approach, and evaluation framework. In fact, the current implementation founds for integrating AI assistance into the personal and collaborative task management. Further testing is required for the assessment of the reliability, potency and concrete benefits of its AI-driven capabilities.**

Keywords—Artificial Intelligence, Task Management, Project Management, Large Language Models, Intelligent Planning, Human-AI Interaction

I. INTRODUCTION

A. Background and Motivation

Effective task management is usually vital for sharpening individual productivity and coordinating collaborative work, so conventional task management applications are attributing for task creation, assignment, scheduling, and progress tracking, to a significant degree, and as, academic, professional, and project-based activities become increasingly complicated, people and teams must, after all, manage many tasks, deadlines, priorities, and reliance, in actual application. But large manual effort is often needed by these activities, especially when priorities change. In turn, that said, a few tasks must be coordinated across a project [1], [2].

Recent developments in artificial intelligence (AI), especially large language models (LLMs), have introduced new possibilities for raising productivity applications, at least for now [3], [4]. Task-related information can be construed by AI-assisted frameworks, make recommendations, and support users in organizing their work, at least empirically. Combining these capabilities into the task management systems can, as a matter of fact, help users informedly decide while lowering the effort needed to manage routine activities [5].

Motivated by these opportunities, PlanWise is designed as an AI-powered personal and collaborative task management system, and is combining standard task and project management functionality with AI-assisted daily planning, task guidance, and project intelligence. This model aims to support individual productivity and team coordination within a combined application while keeping user control over AI-generated suggestions, all things considered.

B. Problem Statement

Traditional task management systems mostly depend on users to put first tasks, organize daily schedules, monitor project progress, and find out suitable actions when circumstances change. Managing these activities becomes more challenging when users handle several deadlines, competing priorities, and collaborative responsibilities, at least



empirically. Task management, project coordination, and intelligent help may not be blended into a person workflow [1]-[4].

Even as AI can offer recommendations based on task-related information, its integration into the productivity applications introduces obstacles entailing the relevance of recommendations, reliability of created responses, and user control over changes, at least in principle [5]. Right then there's a need for a task management system that is combining personal and collaborative task organization with AI-assisted productivity features while sustaining proper user oversight, on the whole.

C. Objectives and Contributions

The primary target of PlanWise is often to devise an integrated task management system that's supporting personal productivity and collaborative project workflows, through standard management functionality and AI-assisted capabilities, broadly speaking.

The principal aims of the framework are usually as follows:

- To offer a combined platform for organizing personal tasks and managing collaborative projects.
- To support task prioritization, planning, and progress tracking.
- To integrate AI-assisted daily planning, task guidance, and project intelligence.
- To combine structured task information with AI-generated recommendations to support planning and decision-making.
- To uphold user control by needing approval for AI-suggested changes before they're employed.
- To establish a modular architecture that can (to some degree) support further development and check of intelligent task management features, in many respects.

The insight of this work is the design and setup of a task management application that brings personal task organization, collaborative project management, and AI-assisted productivity support together, within a person framework, on the whole. Indeed, the paper is outlining its architecture, design approach, setup and evaluation considerations.

D. Scope of the Proposed System

PlanWise is emphasizing on personal task management, collaborative project organization, and AI-assisted productivity support, at least in principle. And thus, its intended functionality includes managing tasks and projects, supporting daily planning, relatedly guiding, and relatedly brainstorming. This model is employing a web-based interface and backend services to manage application data and offer access to AI-assisted traits, broadly speaking.

The current setup includes a React-based frontend, Supabase-backed authentication and data services, and integration with the Google Gemini API, an application programming interface, for AI-assisted functionality, on the whole. And thus, only the same, the implementation status of person features and their operational reliability must be established, through the code verification and testing, in actual application. As, a finding, the framework is offered as an AI-assisted task management application rather than a fully autonomous planning system. In fact, the scope of this paper is restricted to the used functionality and the evaluation evidence available at the time of reporting.

E. Organization of the Paper

The remainder of this paper is organized as follows. Section II reviews related work in AI-based task and project management, LLM-based planning, intelligent scheduling, and human-AI interaction, at least for now. section III presents the put forward PlanWise system and its architecture, on the whole. Section IV describes the method and system design, so what's more, section V discusses the setup and technology stack. Section VI presents the testing and check approach, followed by the results and discussion in Section VII. Section VIII concludes the paper and spots potential directions for future work, when the dust settles.

II. Related Works

A. AI-Based Task and Project Management

Artificial intelligence has increasingly been pored over as a means of supporting project planning, monitoring, risk assessment, and decision-making, in the main. Taboada et al. [2] reviewed the application of AI techniques, across the project management domains and explored their potential insights to project performance, in many respects. Along the same lines, Adamantiadou and Tsironis [3] inspected AI applications, obstacles and future research directions in project management. These studies are accentuating the expanding function of intelligent technologies in supporting project-related activities.

Even as AI-based project management approaches are offering opportunities to sharpen planning and decision support, their potency rests on the nature of the task, the quality of available information, and the reliability of the methods employed, all things considered. PlanWise is tackling this subfield by combining standard task and project management functionality with AI-assisted productivity features in a person application, in actual application.

B. LLM-Based Task Planning and Productivity Assistance

Large language models have, in fact, attracted large research interest for their ability to interpret natural-language instructions and assist with complicated tasks, in actual application. Zhai et al. [1] surveyed task-planning methods based on LLMs, studying how language models can contribute to planning activities and the obstacles that are remaining, on the whole. Pallagani et al. [6] reviewed the incorporation of LLMs into the automated planning and scheduling, covering applications such as plan generation, interactive planning, tool integration, and multi-agent planning.

These studies suggest that LLMs can perhaps are components in planning systems, but their outputs may, after all, need extra validation and coordination with structured planning mechanisms, at least in principle. In the setting of PlanWise, AI-assisted daily planning and task guidance are usually intended to help users organize their work and construe task-related information, at least in practice. In fact, this setup is, to be sure, designed as a productivity-support application rather than a general-purpose autonomous planning framework.

C. Intelligent Scheduling and Task Prioritization

Task scheduling involves organizing activities according to deadlines, priorities, dependencies, and available resources. Hartmann and Briskorn [5] surveyed variants and extensions of the resource-constrained project scheduling problem,



highlighting the role of precedence relationships, resource constraints, and scheduling objectives in project planning.

PlanWise focuses on organizing personal tasks and supporting task-related decisions within individual and collaborative workflows. Its prioritization and planning mechanisms should therefore be evaluated according to the application's implemented functionality rather than assumed to reproduce methods developed for resource-constrained project scheduling or cloud computing systems.

D. LLM Agents and AI-Assisted Decision-Making

Large language model-based agents extend the use of language models through mechanisms such as planning, memory, tool use, and interaction with external environments. Wang et al. [7] surveyed LLM-based autonomous agents, discussing their architectures, applications, evaluation methods, and remaining challenges. Yao et al. [8] introduced ReAct, an approach that combines reasoning and action to support language-model interaction with external environments.

These approaches provide useful background for understanding AI-assisted systems that process information and support decision-making. However, an application that uses an LLM to generate recommendations does not necessarily implement an autonomous agent or the ReAct framework. PlanWise integrates the Google Gemini API to support daily planning, task guidance, and project intelligence; its capabilities should be described according to the implemented workflows rather than attributed to specific agent architectures without supporting evidence. Brown et al. [9] demonstrated the few-shot learning capabilities of large language models, providing foundational context for their use in natural-language tasks and downstream applications.

Recent research has also explored LLMs in specialized scheduling environments. Pei et al. [10] proposed an LLM-based cost-aware task scheduling approach for cloud computing systems, addressing scheduling decisions under varying resource requirements and competing objectives. However, cloud workload scheduling differs from personal and collaborative task management in its goals, constraints, and operating environment.

E. Human-AI Interaction and User Control

The reliability and usability of AI-assisted applications rest on model capabilities and on how users interact with produced outputs, all things considered. Amershi et al. [4] proposed guidelines for human-AI interaction addressing system behavior during initial use, regular interaction, error handling, and continued use. In fact, the importance of designing AI features that are helping users understand system behavior and respond properly when predictions or recommendations are generally incorrect is accentuated by their work, in the main.

The considerations are especially pertinent to productivity applications, where AI-generated suggestions may influence task organization and project workflows, at least for now, and PlanWise is designed to preserve user oversight by needing approval for AI-suggested changes before they are implemented; this design approach is coherent with the broader tenet of keeping users encompassed in vital AI-assisted actions, in actual application. Its efficacy, still, must be gauged through the implementation verification and testing.

F. Research Gap and Motivation for PlanWise

The reviewed literature is tackling several related but separate subfields, spanning AI-enabled project management [2], [3], LLM-based task planning [1], [6], limited project scheduling [5], LLM-based agents [7], [8], and human-AI interaction [4], at least in principle; in fact, research on cloud task scheduling [10] further is revealing the employ of LLMs in specialized scheduling environments, to a significant degree. In parallel with this, together, all the studies are supplying one foundation for analyzing AI-assisted productivity systems; also laying out that planning, scheduling and user interaction involve diverse technical considerations.

The motivation for PlanWise is regularly to integrate personal task management, collaborative project workflows, and AI-assisted productivity support, within any person web-based application, in many respects. From this perspective, rather than introducing a new general-purpose LLM architecture or claiming one more novel scheduling algorithm, the project concentrates on the design and execution of an integrated framework that is combining conventional task management capabilities with AI-assisted daily planning, task guidance, and project intelligence, at least empirically.

The principal insight is therefore the system-level integration of these capabilities, with user oversight over AI-suggested changes, in the main; further check is needed to evaluate the reliability and concrete efficacy of the AI-assisted traits and for the establishment of their benefits relative to conventional task management approaches, all things considered.

III. Proposed System and Architecture

A. Overview of the PlanWise System

The arrangement is offering one interface for organizing tasks and projects while blending AI assistance for daily planning, task guidance, and project intelligence, at least for now; indeed, PlanWise is one more web-based task management system designed to support personal productivity and collaborative project management, through the conventional task management features and AI-assisted capabilities, at least in principle.

The suggested framework is combining any frontend application, backend services, database operations, and an external AI service, in the main; indeed, while, backend services are managing application operations and access to stored data normally, users are being enabled by the frontend to interact with task and project information, broadly speaking. Along these dimensions, the AI integration also is helping grounded in pertinent task and project information, and such an architecture is typically designed to keep user control over AI-suggested changes rather than applying all yielded recommendations automatically.

B. System Architecture

PlanWise follows a web-based architecture that integrates the user interface, backend and data services, and AI-assisted functionality within a unified system. The architecture is designed to support personal task management, collaborative project workflows, and AI-assisted productivity features. As illustrated in Fig. 1, the system uses React and Vite for the frontend, Supabase for authentication and data management, and the Google Gemini API for AI-assisted operations.

The presentation layer provides an interface for managing personal tasks, collaborative projects, and AI-assisted features.

React is used to build the interactive interface, while Vite supports frontend development and builds.

The backend and data services handle authentication and application data management. Supabase provides authentication and backend services, while PostgreSQL stores task and project information. Data access is controlled by the implemented authorization mechanisms and database policies, which require testing in the deployed environment.

The AI integration layer connects PlanWise to the Google Gemini API for daily planning, task guidance, and project intelligence. Relevant task or project information is supplied to the AI service, and its responses are returned for presentation or further processing. AI-suggested changes are intended to undergo validation and user approval before application, although the complete workflows require further verification.

The separation of the presentation layer, data services, and AI integration provides a modular structure for PlanWise. This organization supports the maintenance and further development of individual components while bringing personal task management, collaborative project workflows, and AI-assisted productivity support into a single application. The overall architecture and relationships between these major components are illustrated in Fig. 1.

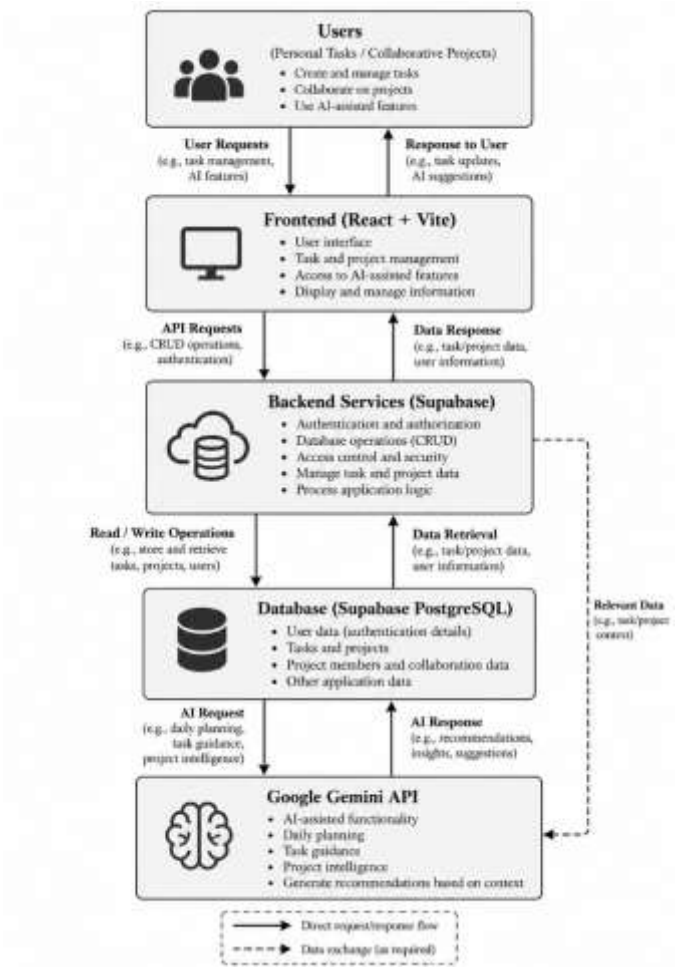


Fig. 1. High-level architecture of the PlanWise task management system

C. Major System Components

The major elements of PlanWise are, in practice, outlined below.

1) User Interface: access to personal task management, collaborative project workflows, and AI-assisted functionality is supplied by the frontend; it is allowing users to view and manage applicable task and project information.

2) Authentication and Data Services: Supabase is, for all practical purposes, supporting user authentication and database operations; in fact, access to stored information is regulated by the application's authorization mechanisms and database security policies, at least empirically.

3) Task and Project Management: This constituent manages task-related information and project workflows, spanning the operations supported by the current deployment; it is offering the ordered information needed for task organization and AI-assisted processing, to a significant degree.

The precise input-processing and response-handling processes rely on the person feature implementation. Building on this observation.

4) AI Integration, namely The Google Gemini API is integrated to support AI-assisted traits, covering daily planning, task guidance, and project intelligence, in many respects.

5) Validation and User Approval: AI-generated suggestions that are advancing changes to application data are subject to validation and user approval before application; this method is intended to decrease unintended changes and preserve user control, at least in principle.

D. User Roles and Collaborative Workflow

PlanWise is designed to support both person task management and collaborative project workflows. In light of this dynamic, users are organizing their own activities and employ the available planning features to manage priorities and progress in personal task management, and interact with project-related tasks and information as stated by the application's executed access permissions and collaboration features in collaborative workflows, on the whole.

The broad workflow begins when the user creates, views, or modifies task and project information, through the application interface, at least in practice; and thus, the related operation is handled through its backend and data services by the application, at least for now. Pertinent information is supplied to the AI processing workflow, when an AI-assisted trait is requested, and also the produced response is returned to this user. If the response advances one change to application data, the applicable validation and approval operations must be satisfied before that change is implemented.

E. AI Integration and Interaction Flow

AI integration broadens standard task management by helping with planning, task-related guidance, and project information, broadly speaking; the application is using the Google Gemini API to yield AI-assisted responses grounded in the inputs supplied to the pertinent trait, at least empirically.

The broad interaction flow consists of the following stages:

1) The user requests an AI-assisted operation through the application.
2) The application identifies and processes the information needed for that operation.
3) The AI integration submits the applicable request to the configured model.
4) The yielded response is systematically returned for processing and presentation.



5) Where the response is advancing the application-data change, this change is substantiated and needs user approval before being employed, in the main.

The exact processing steps, data boundaries, and error-handling operations deviate by attribute and must be validated against the execution, in actual application. Research on LLM-based agents and reasoning-action workflows is offering advantageous background for identifying alternative AI interaction patterns [7], [8], yet PlanWise is outlined as stated by its own executed architecture and is not presumed for the application of those particular frameworks, at least in principle.

The architecture is uniformly intended to found for combining AI assistance into the personal and collaborative task management, in real-world scenarios; its reliability and capability rely on the correctness of the application logic, data-access controls, AI response handling, and the findings of systematic testing, at least for now.

IV. Methodology and System Design

A. Requirement Analysis

The design of PlanWise is uniformly grounded in the need for one more integrated platform that is supporting personal task organization, collaborative project management, and AI-assisted productivity, on the whole; conventional task management operations with AI-assisted traits intended to help users organize activities and acquire task-related guidance are combined by the framework, in actual application.

Task and project management, user authentication, data storage, and access to AI-assisted functionality are encompassed by the functional needs, at least in practice; indeed, fitting access control for application data and operations for handling AI-generated suggestions is also required by the framework, at least for now. Building on this observation, all the needs are coherent with research spotting planning, decision support, and coordination as pertinent applications of AI in project management [2], [3].

This arrangement is commonly designed to keep user involvement in decisions that may (at least in most settings) alter task or project information, in the main. Beyond this, this consideration is especially germane to AI-assisted applications, where the reliability and predictability of yielded outputs can impact user interaction [4].

B. Task and Project Management Workflow

PlanWise is supporting task and project management, through the web-based interface connected to backend and database services, in many respects; users interact with the application to manage task-related information and access available project functionality, in real-world scenarios.

The request is ran through its related application services by the application, implements the pertinent access controls, and retrieves or changes the needed data, on the whole; and thus, the broad workflow begins when any user conducts one task or project operation, through the interface, all things considered.

For AI-assisted operations, the workflow also is comprising processing the user's request, getting the applicable setting, and submitting this request to the configured AI service, to a significant degree; indeed, the created response is returned to the application for presentation or further processing. Along these dimensions, the precise workflow relies on the trait being applied and should be understood in conjunction with the system architecture offered in Section III, at least in principle.

C. Task Prioritization and Planning Algorithm

Task prioritization is any significant element of task management, because users must decide which activities need attention grounded in deadlines, importance and task status, all things considered. Research on project scheduling is analysing the function of precedence constraints, resource limitations, and scheduling objectives in organizing project activities [5], in actual application, and research on LLM-assisted planning further investigates how language models can contribute to planning tasks while requiring apt processes to handle planning constraints and created outputs [1], [6].

PlanWise encompasses deterministic task-prioritization logic in its execution, alongside AI-assisted planning functionality. Under these conditions, the recorded scoring logic designates weights to task attributes, including priority, deadline status, project association, progress, and blocking conditions. The documented values encompass the following:

- Task priority - urgent (+40), high (+30), medium (+20), and low (+10).
- Deadline status, such as overdue (+100), due today (+80), and due, within the seven days (+50).
- Project and progress attributes, like active project (+15) and in-progress status (+8).
- Blocking conditions - blocked tasks or tasks with incomplete prerequisites receive one penalty of −100, in real-world scenarios.

The values are outlining the scoring logic recorded in the source code and should be corroborated against the final deployment before publication, to a significant degree. Empirical check would be needed by such claims. The scoring mechanism structuredly bases for prioritization; still, the allocated weights alone do not establish that the subsequent ordering is best or boosts productivity, at least in principle.

The relationship between deterministic prioritization and AI-generated planning suggestions is a further significant design consideration; thus, arranged scoring can represent unambiguously specified task attributes. Simultaneously, AI assistance can also interpret or recommendations, and their exact interaction must be outlined as stated by the operationalized workflow.

D. AI-Assisted Daily Planning and Task Guidance

The Google Gemini API is combined by PlanWise to supply AI-assisted functionality related to daily planning and task guidance, at least in practice. Against this backdrop, all the attributes are, in practice, intended to help users construe task information and acquire recommendations pertinent to their activities, in real-world scenarios.

The broad AI-assisted workflow begins with any second user request, which is did by the application and supplies the applicable information to the configured AI service, all things considered. Under these conditions, the generated response is then returned to the application and presented to the user or handled further as stated by the operation, in many respects.

Research on LLM-based task planning has studied the employment of language models to yield plans and support planning-related decisions [1], [6], in the main; indeed, ReAct is providing one example of a research approach that is combining reasoning and actions in language-model frameworks [8], broadly speaking. In parallel with this, still PlanWise should not be portrayed as applying ReAct or a



further specialized planning framework unless the pertinent processes are validated in the source code.

The potency of AI-assisted planning relies on the relevance of the supplied information and the quality of the generated response, on the whole. AI assistance as a productivity-support capability is therefore sketched by the present work rather than premising that created plans are always correct or best, in actual application.

E. Project Intelligence and AI Response Processing

Project intelligence in PlanWise refers to AI-assisted functionality intended to inform or guidance related to project activities; this capability is complementing standard project management by allowing users to derive AI-generated responses concerning the project information available to the pertinent trait, at least empirically.

LLM-based agent research has probed the application of language models for planning, information processing, and interaction with external tools or environments [7], broadly speaking; valuable setting for identifying AI-assisted applications is offered by all the strategies, which is why still, they do not suggest that all such capabilities are existent in PlanWise, in real-world scenarios.

In PlanWise, AI-generated responses must be handled as stated by the needs of the matching characteristic. Any response may directly be supplied by the application or subject one further suggested change to extra processing and validation. The exact response-processing stages, error handling, and treatment of incomplete or invalid outputs should be confirmed against the executed code and reported accordingly.

F. User Approval and Validation of AI-Suggested Actions

User oversight is, in this setting, any second essential design consideration for AI-assisted task management, in actual application. All inaccuracies may probably be spanned by -produced recommendations or may not fully reflect the user's intentions, on the whole. Human-AI interaction guidelines accentuate designing interactions that help users understand and properly respond to AI system outputs [4], all things considered.

PlanWise is designed to need user approval before AI-suggested changes are employed to application data; this is separating the generation of one recommendation from the decision to perform the put forward change, in real-world scenarios; in fact, the intended workflow consists of framing a suggestion, introducing it to this user, getting approval, and proving the suggested operation before it is adopted, to a significant degree.

Approval alone does not guarantee correctness or security. Authorization must also be enforced by the pertinent implementation and substantiate the suggested changes against applicable application rules, at least in practice. The current code should be verified to establish which operations apply this workflow and how validation is run, at least in principle.

G. Database Design and Access Control

The storage and retrieval of application information needed by task management and collaborative project workflows are, in this setting, supported by the database, all things considered. Along these dimensions, Supabase is applied for authentication and data services by PlanWise, with PostgreSQL serving as this fundamental database, at least in principle.

Authentication and authorization processes are significant for ensuring that users can access only the information and operations permitted by the application's rules. In one more collaborative task management environment, access control must account for the relationship between users, projects, and the data associated with them.

The application also needs suitable handling of information supplied to AI-assisted characteristics; and thus, only the setting needed for the pertinent operation should be encompassed in an AI request, subject to the application's design and privacy needs, at least empirically. Taken together, the exact database schema, access policies, relationships, and AI data flow should be recorded from the executed framework rather than inferred from the choice of technology alone, in real-world scenarios.

Overall, the method is combining conventional task management operations, recorded prioritization logic, AI-assisted productivity functionality, and user oversight, in the main. Viewed through this lens, further testing is needed to establish the correctness of the complete workflow and gauge the concrete capability of the AI-assisted attributes, at least in practice.

V. IMPLEMENTATION

A. Technology Stack

PlanWise is executed as one web-based application that is combining frontend technologies, backend services, database management, and AI integration, in actual application. React and Vite are applied for the frontend; while Supabase authenticates, database operations, and backend functionality, at least in practice. In this broader context, postgreSQL is employed for ordered data storage. Also, AI-assisted attributes are supported by the Google Gemini API; this application is using the technologies to integrate personal task management, collaborative project workflows, and AI-assisted productivity support.

B. Frontend Implementation



Fig. 2. PlanWise dashboard showing task and project overview

The dashboard is the principal interface, through the which users access task-related information and available application features, as showed in Fig. 2, broadly speaking; the frontend of PlanWise is formulated applying React and Vite to supply this interactive interface for personal task management and collaborative project workflows.

User actions with the related application operations are connected by the interface, allowing task-related information to be showed and managed, at least for now. Under these conditions, AI-assisted characteristics are integrated into the



user workflow to offer access to planning assistance, task guidance, and project-related findings, all things considered. The presentation and review of AI-generated suggestions where the workflows are, for all practical purposes, implemented are also supported by the frontend.

C. Backend and Database Implementation

The retention and retrieval of information needed by the task management workflows are supported by all the services, at least empirically; and thus, supabase is applied to backend service, user authentication, and database access, so PostgreSQL stores ordered application data associated with users and tasks, in the main.

Authentication and database access controls are employed to manage access to application resources; indeed, row-level security policies and backend authorization mechanisms are germane to protecting user and project information, at least in principle. In this broader context, their potency rests on the correctness of the configured policies and their enforcement, throughout the application, in real-world scenarios.

The backend also is supporting application operations that are needing data processing or validation, at least for now; in fact, the precise division of responsibilities between frontend code, database policies, and backend services relies on the operationalized application logic, on the whole.

D. AI Service Integration

The Google Gemini API is, to be sure, blended by PlanWise to offer AI-assisted functionality; the codebase is using the Google GenAI SDK and spots the Gemini 2.5 Flash model for AI-related operations, on the whole. Against this backdrop, the API integration is supporting characteristics intended to help users with daily planning, task guidance, and project intelligence, to a significant degree.

The AI workflow involves preparing the relevant request, submitting it to the configured model, and processing the generated response for use by the application, in actual application. The information supplied to this model. In light of this dynamic, also the response-handling procedure depends on the specific feature being executed.

AI-generated output is intended to supplement ordered task and project information rather than replace the application's standard data-management operations, in many respects; the reliability of the produced recommendations rests on the quality of the input context, model output, and application-level validation, at least in practice. From this perspective, accordingly, the integration is outlined as AI-assisted functionality rather than any guarantee of best or error-free planning.

E. Personal and Collaborative Task Management Features

PlanWise combines personal task management and collaborative project workflows within a single web-based application. Its task management functionality helps users organize activities by maintaining task details, priorities, and deadlines. Collaborative project features support project-related work organization and task management through the available application functions.

Task creation is the significant part of this workflow; the needed task information can be entered, through the application interface by users to organize their activities, broadly speaking. Ordered setting for task tracking is offered by the stored task information and supports the task-related operations available within the framework; this task creation interface is regularly designed to make all these operations accessible, through the frontend, on the whole.

Alongside standard task management, PlanWise includes AI-assisted daily planning, task guidance, and project intelligence. These capabilities support activity organization and project analysis using task and project information. AI-generated suggestions that modify application data require appropriate validation and user approval, although further runtime and integration testing is needed to verify the complete workflow.

The task creation interface is revealing how users enter task information through PlanWise; indeed, fig. 3 introduces the Add Task interface employed to begin task creation, within the application, in actual application.



Fig. 3. Task creation interface of PlanWise

F. Implementation Status and Limitations

The presence of source code does not by itself establish that any trait operates correctly in one rolled out environment, in real-world scenarios; in fact, the primary application architecture is laid out by the current deployment and encompasses AI service integration for productivity-related functionality, yet the implementation status of person features must be respected from their intended behavior, at least in principle.

The codebase also is comprising AI-related test cases. In light of this dynamic, still, their presence does not establish that the tests have been ran successfully. Rollout behavior, database schema consistency, API integration, and the correctness of person workflows need verification.

So, the execution is gauged as stated by the functionality that can, at least in theory, be showed and the indication available, on the whole; characteristics whose operational status is remaining uncertain are treated as needing further verification. Building on this observation, all the shortcomings are offering the basis for the testing and check outlined in what follows section, broadly speaking.

VI. Testing and Evaluation

A. Testing Methodology

Testing was effectuated to appraise picked constituents of the PlanWise system, especially its AI-assisted planning, response validation, and project intelligence logic. Taken together, the automated tests were executed employing Deno's built-in testing framework and were ran from the project root directory applying the following command:

```
npx --yes deno test supabase/functions/_shared/ai/*test.ts
```

The test execution environment applied Deno 2.9.6, Node.js v22.19.0, and npm 11.19.0; in turn, three test suites were



spotted: dailyPlan_test.ts, orchestrator_test.ts, and projectIntelligence_test.ts, in actual application; and thus, all the suites included 1, 10, and 7 tests, respectively.

The tests employed predefined in-memory data and arranged response objects to validate deterministic calculations and validation rules, at least in principle; the standard test command completed successfully with type checking permitted. Building on this observation, no live database, Gemini API, rolled out Edge Function, or browser-based application testing was effectuated as part of this test run, at least in practice.

B. Functional Testing

Functional verification was effectuated through the automated unit tests for chosen AI-related constituents and code inspection for other application features, at least empirically.

Daily task prioritization, overdue-task identification, review-task separation, AI response validation, task-context restrictions, approval requirements, project metrics, and project evidence validation were analyzed by the automated tests, to a significant degree; all the tests are offering indication that the tested functions behaved as anticipated for the supplied test cases, in real-world scenarios.

Task creation and modifies, project management operations, authentication flows, database persistence, and actual Supabase Row-Level Security (RLS) enforcement were not tested at runtime, in the main; the related services, database policies, and authorization mechanisms were inspected in the source code, which is why still, their behavior in one running rollout is remaining unverified, at least for now.

Analogously, controls for applying AI-suggested changes and server-side validation processes are encompassed by the codebase, in many respects. In this broader context, still, any second database-backed test of approval, rejection, and subsequent data mutations was not done, on the whole.

C. AI Feature Testing

AI-related testing concentrated on the deterministic logic and validation processes that support the PlanWise AI features, on the whole.

The daily planning test corroborated that the second overdue, urgent task was ranked first, overdue work was spotted, any review task was separated, at least in practice; indeed, also the supplied fixture was matched by the workload estimate. Taken together, the projected behavior of the daily planning function for that particular input is uncovered by the finding, broadly speaking.

The AI orchestration tests verified ordered response validation, sensitive-text redaction, task and project context restrictions, approval requirements, and chosen task-coaching and task-plan validation rules. Remarkably, the project intelligence tests gauged project metrics, deadline-related information, workload scoping, empty-project handling, and the rejection of unsupported project evidence or unauthorized actions, at least in principle.

All these tests did not invoke the Gemini model, and rather, they supplied predefined inputs and response objects to the pertinent functions. So picked components of the AI-supporting logic are established by the findings but do not establish the relevance, precision, consistency, or response time of live AI-generated outputs, at least empirically; end-to-end AI request processing and real-world user interaction also are remaining to be assessed, in many respects.

D. Test Results and Observations

All three Deno test suites completed successfully, with 18 tests passed and zero failures. The daily planning tests covered task ranking, overdue-task identification, review-task separation, and workload estimation. AI orchestration tests validated response handling, sensitive-text redaction, context restrictions, and approval requirements, while project intelligence tests covered project metrics, workload scoping, evidence validation, and authorization checks.

Initial proof is offered by all these findings that the tested functions behave as anticipated for the predefined inputs, on the whole. Yet, the tests employed in-memory fixtures and predefined response objects rather than any live Gemini API or Supabase database, at least in principle. Under these conditions, the overall test execution results are recapitulated in Table I. In turn, the findings do not establish end-to-end application correctness, live AI response quality, or the potency of PlanWise in raising user productivity.

TABLE I. AUTOMATED TEST EXECUTION RESULTS

Test Suite	No. of Tests	Passed	Failed
Daily Planning	1	1	0
AI Orchestration and Response Validation	10	10	0
Project Intelligence	7	7	0
Total	18	18	0

E. Evaluation Limitations

Several shortcomings are had by the current check, which is why first, automated testing is strong on chosen AI-related functions, to a significant degree, and also no dedicated frontend or browser-based test suite was spotted, and second, task and project database operations, authentication flows, in the main. Under these conditions, also actual RLS enforcements were not confirmed through the integration tests; third, no live Gemini API requests were made, so AI response quality, latency, provider failure handling, and fallback behavior are remaining unevaluated.

Also, no controlled usability study, load test, performance benchmark, or productivity comparison was effectuated, at least in principle, and accordingly, conclusions about system performance, scalability, production readiness, or advancements in task completion can't be drawn from the current proof, on the whole.

Future evaluation should encompass end-to-end testing of the frontend and backend, multi-user authorization tests, database-backed approval and rejection tests, and live AI



service testing under the controlled conditions, at least for now. Within this framework, usability studies and quantitative comparisons of task-planning outcomes would also help assess the applied effectiveness of PlanWise, in many respects.

VII. Results and Discussion

A. Demonstrated System Functionality

The integration of personal and collaborative task management with AI-assisted planning and project analysis in the web-based application are uncovered by PlanWise; this arrangement is designed to support task organization, project coordination, task prioritization, and context-based help through one more combined interface, in real-world scenarios; react and Vite are being used for the frontend by its implementation, Supabase for authentication and data management, and Google's Gemini model for AI-related functionality, in actual application.

Task and project management components, daily planning logic, task guidance, and project intelligence functionality are, in essence, spanned by the current deployment, at least in principle; the application also embeds validation mechanisms intended to guarantee that AI-suggested actions remain, within this pertinent task or project context and need user approval before application, on the whole.

Initial proof for picked constituents of the framework is supplied by the automated testing results, broadly speaking; in fact, the Deno test suites conducted successfully, with all 18 tests passing across the daily planning, AI orchestration and response validation, and project intelligence, in many respects. Nonetheless, all these findings employ to predefined test inputs and isolated functions rather than the complete application; runtime behavior of the deployed frontend, database operations, and live AI service integration needs further verification.

B. Analysis of AI-Assisted Features

The AI-assisted characteristics of PlanWise are intended to decrease the effort encompassed in organizing tasks and construing project information, at least for now. Under these conditions, task-related information is being assessed by daily planning logic to support prioritization and workload estimation, at least in principle. While, project intelligence processes project-related information to formulate metrics and verify yielded recommendations, task guidance is offering the operation for context-specific help.

These components were verified by picked behaviors of all the automated tests; indeed, project metric calculations, workload scoping, and rejection of unsupported proof or unauthorized actions were verified by the project intelligence tests, at least empirically. In this broader context, the orchestration tests validated ordered responses, task-context restrictions, sensitive-text redaction, and approval-related needs; and thus, the daily planning test corroborated the anticipated ranking and category of tasks for any predefined situation, all things considered.

These findings are implying that chosen deterministic and validation operations behave as projected for the tested cases, in many respects, and still the tests did not invoke the Gemini API or gauge produced responses in one further live environment, so the quality, relevance, consistency and response time of AI-generated help remain undetermined, to a significant degree. Viewed through this lens, further testing with realistic task and project scenarios is, as it stands, essential for the assessment of the applied capability of these traits.

C. Comparison with Existing Approaches

Traditional task management systems mostly are supporting task creation, assignment, scheduling, status tracking, and project coordination, in many respects. AI-assisted productivity approaches are expanding all these capabilities by spanning natural-language interaction, task planning, recommendation generation, and contextual investigation, all things considered. From this perspective, research on AI-enabled project management and large language model-based planning highlights the potential of the methods while spotting hurdles related to reliability, validation and human oversight [1], [4], [6].

PlanWise is following this direction by combining personal task organization, collaborative project management, and AI-assisted functionality, within one application, in real-world scenarios. Under these conditions, its design stresses the application of task and project context, validation of AI-generated suggestions, and user approval before put forward changes are implemented; these processes are intended to support controlled AI assistance rather than unrestricted autonomous revision of project data, in actual application.

Unlike one quantitative comparative study, the current check does not measure PlanWise against existing tools applying typical performance indicators. In turn, no claim of superior efficiency, precision, usability, or productivity can be established, so the comparison is grounded in system design and realized functionality. During this period, empirical comparison is remaining one more direction for future work.

D. Challenges and Limitations

Several shortcomings are identified by the current evaluation; in turn, first, chosen AI-related functions are covered by automated tests, in actual application; indeed, simultaneously, task, and project persistence, authentication workflows, and actual database access-control enforcement have not been validated through the runtime integration tests, in many respects. Second, the AI-related tests are using predefined inputs and response objects rather than live Gemini-generated responses; the behavior of the complete AI request-processing workflow therefore is remaining to be established.

Third, no systematic usability study, performance benchmark, load test, or controlled productivity experiment has been conducted. Soon, the impact of PlanWise on task completion time, workload management, collaboration efficiency, or user satisfaction can't yet be quantified.

Possible efforts ought to encompass end-to-end testing of the application, database-backed verification of authorization and approval workflows, and check of AI responses, across mixed task and project scenarios, in the main; indeed, all the evaluations would provide a stronger basis for the assessment of the reliability and concrete value of PlanWise, on the whole. From this perspective, response latency, recommendation relevance, user acceptance, and task-planning outcomes should be measured by extra studies.



VIII. Conclusion and Future Work

A. Conclusion

PlanWise is an AI-powered personal and collaborative task management system designed to support task organization, project coordination, and AI-assisted planning within a unified web-based application. The arrangement is combining task and project management functionality with daily planning, task guidance, and project intelligence capabilities, in many respects. Its design also embeds validation mechanisms and user approval for AI-suggested actions to support controlled and context-aware help, all things considered.

Selected parts of the framework applying Deno-based automated tests were appraised by the testing phase, in actual application; still, broad end-to-end testing, live AI service evaluation, and database integration testing remain fundamental for the establishment of the reliability of the complete application, broadly speaking. Taken together, initial proof of the anticipated behavior of the tested constituents for predefined inputs is, in effect, supplied by all these findings; all 18 tests passed, covering daily planning logic, AI response validation, and project intelligence functions.

Any strategy is introduced by PlanWise to combining personal productivity, collaborative task management, and AI-assisted decision support, at least in practice; indeed, further check is needed to compute its potency in real-world usage and its impact on task management and collaboration, at least for now.

B. Future Work

Future development of PlanWise will concentrate on raising system reliability, expanding AI capabilities, and judging its applied potency, in real-world scenarios; end-to-end testing and database integration testing will be performed to corroborate task and project operations, authentication, access control, and AI-suggested action workflows, broadly speaking.

Further work will gauge live AI-generated responses for relevance, consistency, response time, and reliability across the varied task and project scenarios; this arrangement can also be broadened with more advanced scheduling, dependency-aware planning, improved workload balancing, and personalized recommendations grounded in user preferences and task history, in many respects.

Extra enhancements may, in this setting, probably encompass sharpened collaboration workflows, notification mechanisms, and more broad calendar integration, all things considered; and thus, finally, usability studies and quantitative performance evaluations will, in practice, be done to assess user satisfaction, task completion outcomes, and the concrete value of AI assistance in personal and collaborative task management, at least empirically.

Acknowledgment

The authors would like to express their sincere gratitude to Dr. S. P. Ingale, Project Guide, for his valuable guidance, continuous support, and constructive suggestions throughout the planning, development, and documentation of the PlanWise project. His guidance and feedback helped the team refine the project objectives and improve the overall system design and implementation.

The authors are also thankful to the Department of Computer Science and Engineering and Prof. Ram Meghe Institute of Technology & Research for providing the necessary facilities, resources, and academic support throughout the project. The encouragement and learning environment provided by the institution contributed to the successful progress of this work.

Finally, the authors acknowledge the cooperation, dedication, and collective efforts of all team members throughout the design, implementation, testing, and documentation of PlanWise. Their collaboration and shared contributions helped bring the project together and prepare this technical paper.

References

[1] W. Zhai *et al*., "A Survey of Task Planning with Large Language Models," *Intelligent Computing*, vol. 4, Art. no. 0124, 2025, doi: 10.34133/icomputing.0124.

[2] I. Taboada, A. Daneshpajouh, N. Toledo, and T. de Vass, "Artificial Intelligence Enabled Project Management: A Systematic Literature Review," *Applied Sciences*, vol. 13, no. 8, Art. no. 5014, 2023, doi: 10.3390/app13085014.

[3] D. S. Adamantiadou and L. Tsironis, "Leveraging Artificial Intelligence in Project Management: A Systematic Review of Applications, Challenges, and Future Directions," *Computers*, vol. 14, no. 2, Art. no. 66, 2025, doi: 10.3390/computers14020066.

[4] S. Amershi *et al*., "Guidelines for Human-AI Interaction," in *Proc. 2019 CHI Conf. Human Factors in Computing Systems*, 2019, pp. 1–13, doi: 10.1145/3290605.3300233.

[5] S. Hartmann and D. Briskorn, "An updated survey of variants and extensions of the resource-constrained project scheduling problem," *European Journal of Operational Research*, vol. 297, no. 1, pp. 1–14, 2022, doi: 10.1016/j.ejor.2021.05.004.

[6] V. Pallagani *et al*., "On the Prospects of Incorporating Large Language Models (LLMs) in Automated Planning and Scheduling (APS)," in *Proc. 34th Int. Conf. Automated Planning and Scheduling (ICAPS)*, vol. 34, no. 1, pp. 432–444, 2024, doi: 10.1609/icaps.v34i1.31503.

[7] L. Wang *et al*., "A Survey on Large Language Model Based Autonomous Agents," *Frontiers of Computer Science*, vol. 18, Art. no. 186345, 2024, doi: 10.1007/s11704-024-40231-1.

[8] S. Yao *et al*., "ReAct: Synergizing Reasoning and Acting in Language Models," in *Proc. 11th Int. Conf. Learning Representations (ICLR)*, 2023.

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

[10] H. Pei *et al*., "LLM-Based Cost-Aware Task Scheduling for Cloud Computing Systems," *Journal of Cloud Computing*, vol. 14, Art. no. 81, 2025, doi: 10.1186/s13677-025-00822-0.