Merck

AI Scenario Planning

Sponsor: Ryan Blanton
Project Manager: Japjot Bedi
Percy Flores
Ricardo Cruz Hernandez
Tanvir Longia
Nicholas Mustrat

New Jersey Institute of Technology
CS 491 Senior Capstone Project
Dr. Osama Eljabiri
7 May 2025

Table of Contents

Chapter 1: Introduction

I. Project Background

The AI Planning scenario Chatbot project was developed as part of the YWCC Capstone program under the sponsorship of Ryan Blanton from Merck. The primary objective of the project was to create a conversational interface that allows users to upload and interact with Microsoft Project plan files, enabling natural language exploration of project tasks, schedules, and dependencies. By replacing the traditional rigid and manual methods of interpreting project plans, the chatbot aimed to bring automation and accessibility into the planning workflow.

Our sponsor provided the team with full creative freedom regarding technology selection and architectural direction. After evaluating different options, we decided to build our system on top of the open-source AI chatbot template by Vercel. Which provided a production ready foundation and seamless integration with modern full stack technologies. The base template includes:

- Next.js App router for efficient routing and server side rendering using React server components.
- AI SDK with support for tool use and output parsing from large language models (LLM)
- Shadcn/ui built on Tailwind CSS and Radix UI for clean, accessible interfaces.
- Vercel Blob for secure file upload and Vecel Postgres (via Neon) for optional date persistence.
- NextAuth.js for secure, built-in user authentication.

With this framework in place, we focused on adding project specific functionality. Our backend micro services, implemented using Fast API (Python) and Spring Boot (Java), handled parsing

.mpp files (Microsoft Project Plans) using the MPXJ library and converting them into structured JSON. This data was then stored in Blob storage and made accessible to the chatbot for intelligent querying and reasoning.

To power the AI, the team conducted a comparative study of various LLM providers, including Open AI (ChatGPT), Hugging Face, Ollama, and Google Gemini. While each had strengths, Gemini stood out by offering advanced reasoning, structures output formatting, and high-quality model performance. This was all accessible through its free tier developer API. This made it the most suitable option for our project scope, especially given our focus on query reliability and cost effectiveness.

Although the team planned to implement a Gantt chart visualization layer, a two-week reduction in available development time forced a pivot. Instead, the team concentrated on solidifying the core experience, file ingestion, project plan parsing, and intelligent conversational querying. By the end of the project, the team successfully delivered a chatbot that could:

- Accept Microsoft Project plan uploads.
- Parse and structure the data for task-level access.
- Answer user questions about task scheduling and dependencies.
- Simulate simple scenario-based questions through LLM integration.

This AI Planning Scenario chatbot serves as a practical demonstration of how emerging technologies can be combined to solve real-world planning challenges in scalable, accessible ways.

II. Problem Definition

In large organizations like Merck, managing complex project timelines is critical but also difficult and complex. Project managers must oversee hundreds, sometimes thousands, of tasks each linked though dependency chains. A change to one task, whether a delay, an acceleration, or modification, can distort the project in unexpected ways. Traditionally, these impacts rely on rigid tools such as Gantt charts or manual dependency analysis with software like Microsoft Projects.

This manual process poses several key challenges:

- **Time Consuming Simulations**: Analyzing schedule impacts manually requires significant time and effort.
- **Error Prone updates**: The complexity of dependencies increases the chances of missing critical adjustments that lead to project risks.
- **Rigid User Interface**: Existing project management tools often force its user into fixed workflows without flexible “what if” conversations.
- **Lack of Intelligent Scenario Planning**: Users cannot easily ask “What happens if” questions and receive immediate feedback from their project plans.

To address these shortcomings, the team implemented a hybrid approach that combined logic and AI based interactions. Our backend Python services handled programmatic dependency updates, such as shifting task’s start date and adjusting all affected successors. Meanwhile, Gemini handled follow up conversation, summaries, and picking which function was best suited for the user’s question. This allowed the team to deliver both precision and flexibility, supporting real time decision making while avoiding the rigidity of traditional systems.

In real world applications, there was a need for a system that could parse a project's structure and allow dynamic, conversational exploration of scenarios. Our chatbot addresses this by offering a natural language interface that allows users to ask detailed questions about their projects without needing to manually manipulate tasks or review dependency charts.

In real world applications, efficiency in updating and reviewing project plans can have a direct impact on resource allocation, budgeting, and organizational strategy. Automating portions of this process through an AI enhanced chatbot allows significant potential benefits in both speed and decision-making accuracy.

While the project initially aimed to incorporate a direct Gantt chart visualization into the conversational system, a two-week delay meant that prioritizing a working conversational planning tool over visualization components was the main goal when completing this project. Nonetheless, the delivered system effectively mitigates several tedious points in current project management workflows by automating parsing, dependency identification, and interaction though a streamlined AI based interface.

III. Glossary of Terms

- **AI Scenario Planning Chatbot** - A web-based AI chatbot that allows users to upload and analyze Microsoft Project files through functions.
- **Next.js** – A react based framework that supports server-side rendering, routing, and performance optimization.
- **React Server Components (RSCs)** – Components rendered on the server to improve performance and reduce client-side JavaScript.

- **Vercel AI SDK** – A software development kit that provides hooks and APIs for building generative AI applications using models like Gemini, OpenAI, and Cohere.
- **shadcn/ui** – A component library built on top of Radix UI, styled with Tailwind CSS for accessible and customizable user interface.
- **Tailwind CSS** – A utility first CSS framework used for consistent, responsive styling.
- **Radix UI** – A set of low-level UI primitives for building accessible components with custom styling.
- **Vercel Blob** – A file storage system provided by Vercel for uploading and managing binary files.
- **Vercel Postgres (Neon)** – A serverless Postgres database service integrated into the Vercel platform for storing structured data (e.g. chat history).
- **NextAuth.js –** A simple and secure authentication framework for Next.js applications.
- **FastAPI –** A high performance web framework for building APIs with Python, used for auxiliary microservices.
- **Spring Boot –** A java-based framework used to develop microservices such as the MPP file parser using the MPXJ library.
- **MPXJ –** A java library that supports reading Microsoft Project .mpp files.
- **.mpp file** – A proprietary file format used by Microsoft Project to store project schedule information.
- **Blob Storage** – Cloud storage for handling and retrieving binary data (like file uploads).
- **Gemini (Google)** – The LLM selected for the chatbot, chosen for its context handling and available free tier.

- **ChatGPT/ Ollama/ Hugging Face** – Other AI model providers that were researched during the project's early evaluation phase.
- **Scenario Planning** – A method of exploring "what if" changes to project timelines and assessing impacts.
- **Dependency Chain** – A network of relationships where one task's schedule is contingent on another's.
- **Microservice** – An independently deployable service that performs a specific task within a broader system.

IV. Iteration and Revision:

Through development, the team evaluated two possible approaches: using prompt engineering alone to manipulate project schedules or building functions to handle changes precisely. After prototyping both, the team chose a hybrid method.

This method involved writing custom python functions for rescheduling tasks, detecting holidays/weekends conflicts, and cascading updates through dependency chains. These functions gave us precise control over date manipulation and business logic. On top of this, we integrated the Gemini LLM to interpret user input, understand context, pick the best function for the prompt, and generate a response.

This hybrid approach became a staple of the project. It enabled the system to apply rule-based logic behind the scenes while offering a conversational, human friendly interface for users. The

LLM didn’t make schedule changes directly, it collaborated with backend services that executed actual logic, ensuring accuracy and traceability.

As development progresses, the team also introduced:

- JSON formatting to enhance task metadata retrieval.
- Blob storage linking for persistent file access.

Although the project scope had to be trimmed due to timeline constraints, the core system delivers a robust, flexible foundation for future expansion. Including potential features like Gantt chart rendering, 7-day work week and deeper scenario simulations.

Chapter 2: Project Management

I. Task Analysis

The goal of this project was to develop a tool that can simulate alternate scenarios of a Microsoft Project timeline and instantly reveal the changes made. Our solution integrates a chatbot UI with backend Python functions, allowing the user to shift project schedules forward or backward while considering weekends, holidays, and predecessor/successor tasks.

During the early phases of development, the team tested two different approaches. One was fully AI based and the other was a hybrid approach combining AI with hard-coded logic. After testing, the hybrid approach proved to be more efficient and accurate. It also aligned better with Merck's needs.

II. Team Roles & Responsibilities

The team conducted daily stand-up meetings to share progress with each other and to further collaborate. We also had weekly meetings with our Merck sponsor, Ryan Blanton. These meetings allowed us to share updates, clarify requirements and address any issues the team might have run into during the week. Ryan played a key role by providing us with feedback and the necessary information to successfully complete the project.

All team members contributed across multiple areas, the primary responsibilities were as follows:

- Japjot Bedi - Project Manager, AI Integration

- Percy Flores - AI Integration, Backend Development
- Nick Mustrat - Backend Development
- Tanvir Longia - Backend Development
- Ricardo Cruz Hernandez - Frontend Development

As the Project Manager, Japjot assumed multiple responsibilities to keep the team on track while ensuring the project was headed in the direction wanted by the sponsor. He scheduled team meetings to ensure everyone was on the same page, submitted weekly progress reports, and was in communication with the sponsor. Japjot also integrated Artificial Intelligence, testing various Large Language Models (LLMs) to select the best one.

Percy worked on Artificial Intelligence integration as well as backend development to build the foundation of the chatbot and implement multiple functions that fulfilled the requirements set by the sponsor. He tested numerous LLMs to find the most suitable for the predetermined use cases.

Nicholas and Tanvir worked on the frontend development assisting Ricardo. They also developed functions that would handle the holiday logic on the backend.

Ricardo worked mainly on frontend development. He designed a user-friendly UI that would clearly display the chatbot's output. Ricardo also contributed to the backend development by implementing a microservice to convert MPP files into JSON format for more efficient data processing.

III. Work Breakdown Structure (WBS)

The team followed the Manage, Define, Design, Develop and Evaluate (M.D.D.D.E) structure for the project, as outlined in our WBS (Figure 2.1). These five phases helped prioritize tasks from stakeholder communication and success criteria to backend integration and efficiency testing.

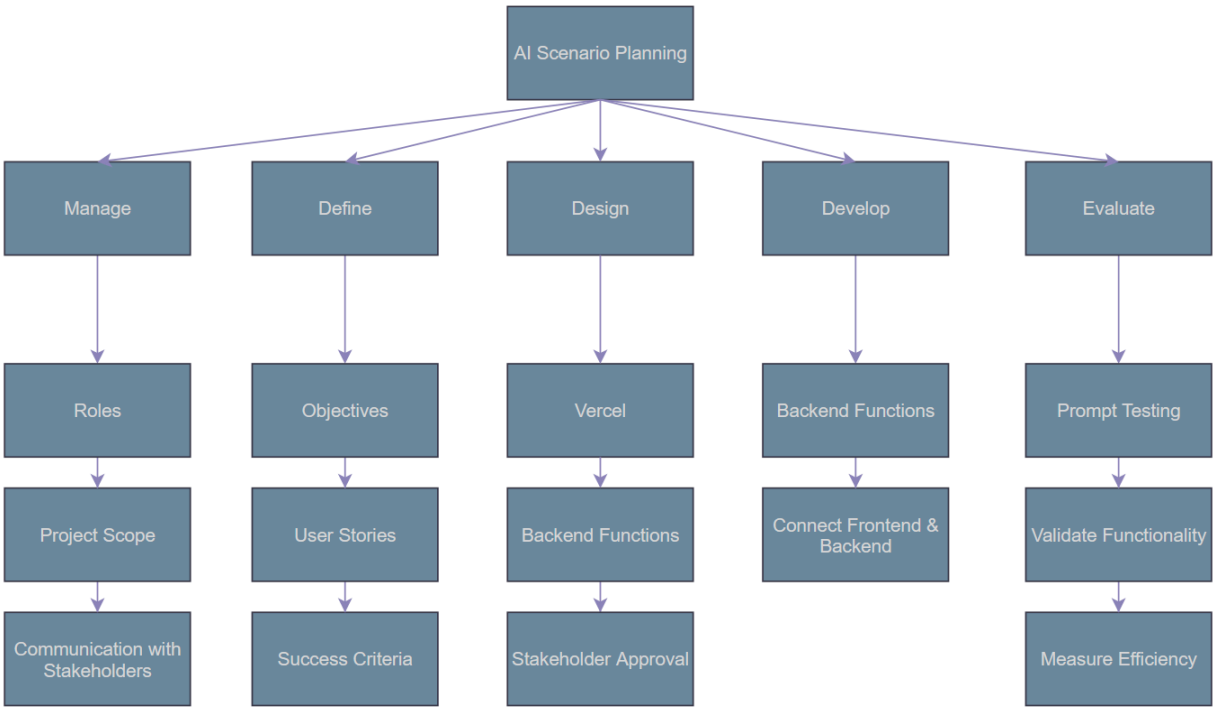


Figure 2.1: WBS diagram

IV. Gantt Chart

The team utilized a Jira board to manage and assign tasks, track progress throughout the duration of the project. A Gantt chart was also created to visualize the project timeline along with the sprints. These tools ensured the team remained organized and on track to meet all the critical

deadlines and milestones. Below is the Jira board and the Gantt chart that were utilized during the project (Figure 2.2).

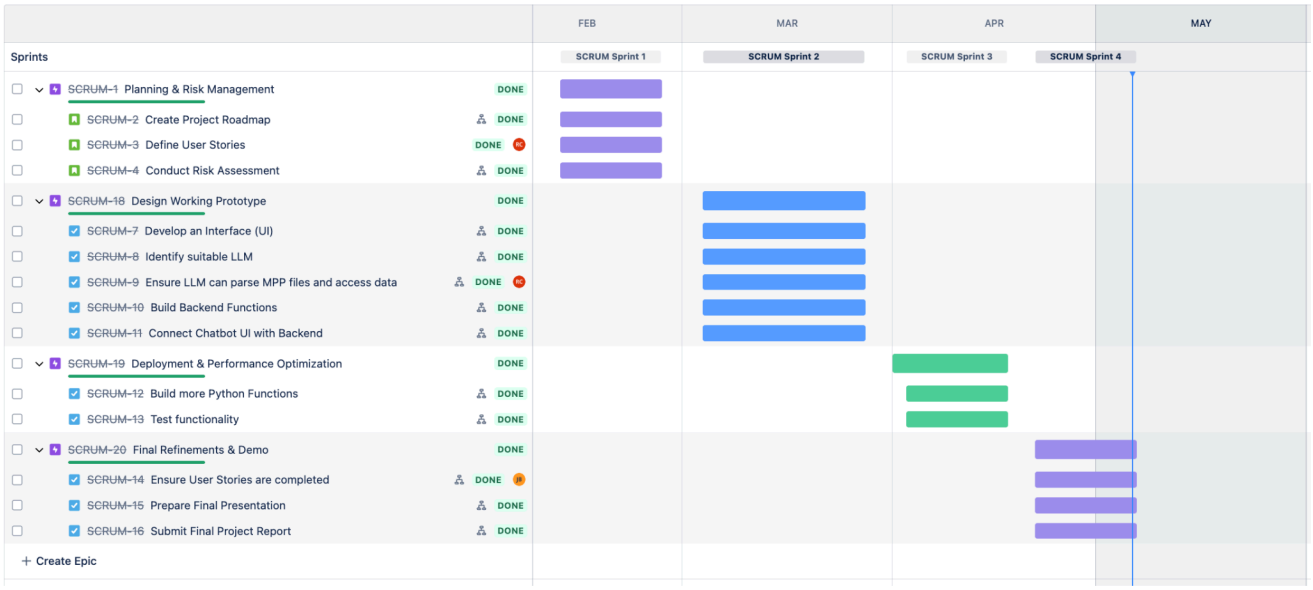


Figure 2.2: Jira Board & Gantt Chart

V. Risk Identification and Management

Throughout the project, the team identified and proactively addressed key risks that could have impacted the quality and timeliness of our deliverables.

1. *Time Restraints*

   We had a delayed project start due to time conflicts with the sponsor, which potentially posed a risk of us not completing in time. To stay on track, the team held daily stand-up meetings, maintained consistent communication with the sponsors, and conducted additional meetings during spring break to recover lost time.

2. *AI Model Accuracy*

   In the early stages, we found that the LLMs were occasionally returning inaccurate results or hallucinations. To mitigate this, we used LangChain to fine-tune the model,

improving its accuracy. Along with this, we reduced our reliance on AI by using hard coded backend logic, to reduce AI exhaustion and essentially achieve better efficiency.

3. *Data Ingestion Issues*

   One of Merck's key requirements was for the chatbot to ingest Microsoft Project (MPP) files. Handling Microsoft Project (MPP) files was a challenge due to its complexity. To resolve this, we used MPXJ libraries and microservices to convert the MPP files into JSON format. This allowed for better data-handling within our backend system.

By identifying and mitigating these risks early on, the team was able to ensure that our final solution was accurate, efficient, and aligned with Merck's technical requirements and standards.

Chapter 3: Define

I. Stakeholders

Merck

Members: Ryan Blanton

Responsibilities: Help establish project scope and deliverables. Discuss with team members certain features which ought to be added to the project given the time allotted to work on said project. Suggest improvements to implemented features and design as the team progresses to a more finalized project. Describes requirements and suggested methods for testing the project for quality assurance. Meet with the Capstone Team once per week as well as respond to any emails throughout the week discussing any concerns or questions regarding the project.

Capstone

Member: Dr. Osama Eljabiri

Responsibilities:

Creates the opportunity for students and sponsors alike to collaborate and communicate in teams with one another with the goal of fostering an industry-like experience throughout the semester. Oversees all teams such as industry, start-ups, RWC, operations, marketing, data, etc. Establishes events such as the Merck open house, industry open house, the Capstone showcase, as well as workshops held throughout the semester. Responsible for student's final grades and which sponsors will be accommodated for each semester. Ensures all students' questions are answered and makes known to students the requirements for their respective tracks.

Merck AI Scenario Planning Capstone Team

Members: Japjot Bedi, Percy Flores, Nicholas Mustrat, Ricardo Hernandez, & Tanvir Longia

Responsibilities:

Create a well working and finalized product for Merck's internal team of Submission Planners by the project deadline. Take part in communication channels with consistency between each other, the project sponsor, the project mentor, and the project manager. Deliver on both individual tasks and team responsibilities in accordance with each sprint. Work with the sponsor to realize project goals and establish any changes to said goals accordingly in order to deliver the end goal of a final product which is to be demonstrated in both midterm assessment and the final Capstone showcase. Meet with the sponsor once per week to discuss progress, gain clarity on certain questions, concerns, and adjustments to the project plan.

End Users

Members: Merck Employees (Submission Planners, Project Managers, IT Team)

Responsibilities:

Utilize the AI Scenario Planning Chatbot to evaluate a Microsoft Project Plan file's tasks as well as each task's predecessor and successor tasks. With the chatbot's help determine the impact of a filing date on various scenarios. Help alter timelines for a certain project plan via pushing tasks forward and backward throughout the project plan, determine which tasks fall on weekends and holidays, determine which tasks for a given timeframe go through holidays, and adjust said project plan accordingly. Help establish a 5-day work week while avoiding weekends and holidays throughout the duration of each individual task within said project plan.

II. Requirements Gathering

Assign Roles

- The Merck AI Scenario Planning Capstone team as well as our project manager were established at the start of the week following the Merck Open House.
- Japjot Bedi was assigned as our project manager and was thus responsible for establishing communication channels with our sponsor, required to file all progress reports, and expected to attend all project manager meetings.
- Percy Flores and Tanvir Longia were designated the responsibility of ensuring the project had a robust and working back end.
- Ricardo Cruz was responsible for establishing an interactive and user-friendly front end which displayed information in a digestible manner to the user.
- Nicholas Mustrat and PM Japjot Bedi were responsible for the integration of AI within a chatbot fashion for the user to communicate with as well as several functions with regard to the tool's back-end functionality.
- Ryan Blanton was assigned as the Merck AI Scenario Planning Capstone Team's sponsor as a seasoned Merck Employee.

Interview Stakeholders

- Following our acceptance into the AI-Scenario Planning Team, we met with our sponsor Ryan Blanton virtually in order to establish our project scope and deliverables given limited resources and time constraints. This would ensure we would have a working and agreed upon product by the time our deadline arrived in May of 2025.

Gather & Document

- Further adjustments, understanding, and solutions came forth via subsequent meetings with our sponsor Ryan Blanton. These meetings, held weekly, would bring further clarification to our project as we discussed solutions to any and all roadblocks, the preferred method to visualize certain data, and other features in which the chatbot should have. During these meetings we would update our sponsor Ryan Blanton on the overall progress of the project and proceed to do so via several “demos” of the product throughout many phases of its development.

Assumptions & Requirements

- Utilized Jira & Gantt Chart in order to enforce scrum methodology. This included sprints, daily standup meetings, weekly sponsor meetings and code reviews. Applying these tools helped delegate tasks with ease among team members and helped foster consistent progress throughout the span of the project. The team also utilized a Discord channel in order to document progress, tools, resources, and possible solutions to any roadblocks we faced. This Discord channel also proved to be an effective route of communication amongst our team members as well as our Operations mentor Nadia Manoppo.

III. Project Scope

Our team, Merck AI-Scenario Planning will develop a Generative AI Tool in the form of a chatbot which ingests Microsoft Project Plan Files. This tool will simulate various scenarios through understanding task dependencies. This includes identifying tasks which start or end on a

weekend or recognized US holiday as well as recognizing tasks whose duration passes through a US holiday. The user will then be able to adjust the project plan according to their specification, for instance reworking the project plan into a 5-day work week format. This adjustment will automatically identify weekends and holidays and proceed to adjust the project plan accordingly. Further functionality includes the ability for the user to move tasks forward and backwards as many days as they see fit, all while recognizing the duration of these tasks and any successor or predecessor tasks linked to currently identified tasks in question.

IV. FDD Requirement Grouping & Use Case Diagrams

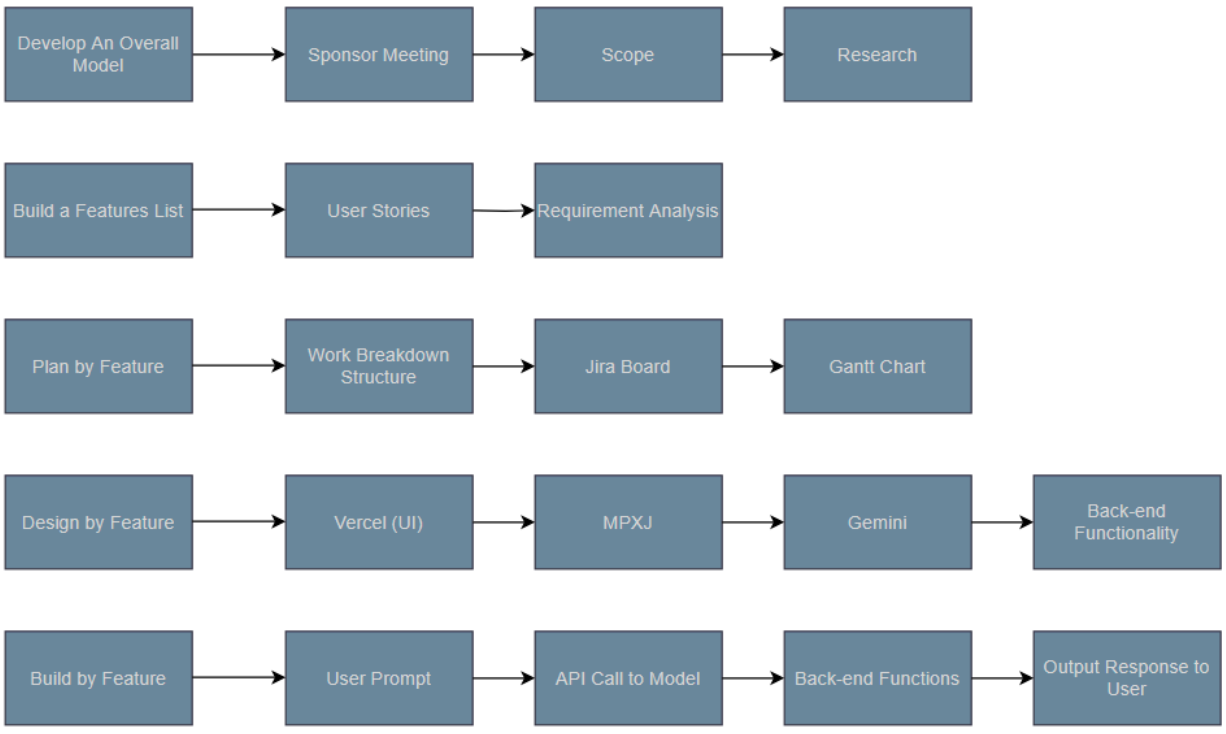


Figure 3.1: Feature Driven Design Diagram

Chapter 4: Design

I. ER Diagram

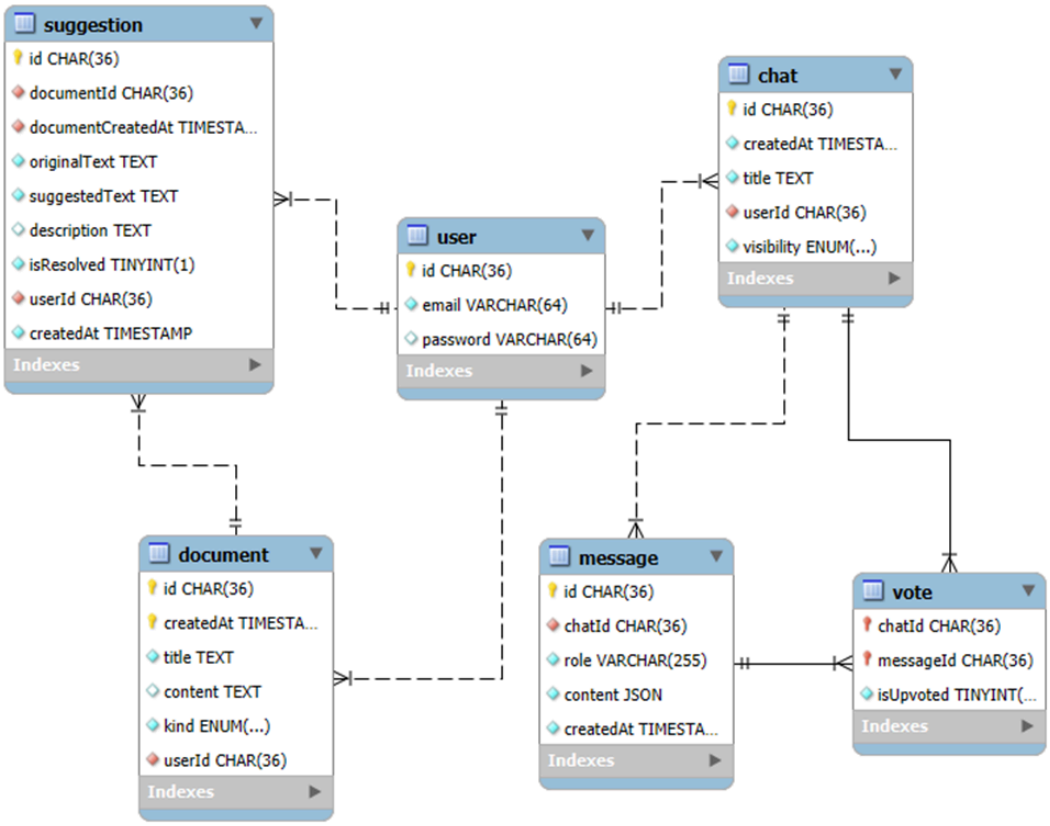


Figure 4.1: ER Diagram

The ER diagram (Figure 4.1) shows the implemented database structure. The database is implemented in a way that supports interactions between a user and the chatbot. The database comprises tables, user, document, suggestion, chat, message, and vote.

Tables

- User: The User table holds the username and password of the user.
- Document: The document table holds any documents uploaded to the chatbot.

- Suggestion: The suggestion table stores any suggestions made by the user for a document
- Chat: The chat table holds all the chats that the user has with the chatbot
- Message: The message table holds each message in the chat sent by the user
- Vote: The vote table holds the up or down vote that a user can use for a message

Workings

- User-Chat: Users have the ability to create multiple chats that are stored in the chat table.
- User-Document: Users have the ability to upload multiple documents.
- Document-Suggestion: Users have the ability to suggest changes to the uploaded document.
- Chat-Message: A single chat can have multiple messages.
- Message-Vote: Users have the option to vote on a message.

II. Blob Storage

A blob storage is utilized for bulk storage. The uploaded MS project files and the JSON outputs of those files are stored in the blob.

III. Implementation

Frontend

Next.js is the primary framework used to implement the frontend for the chatbot. It is mainly utilized for building the user interface. React server components are used to reduce the loading times and render the JavaScript components effectively. Vercel AI

SDK is utilized to integrate the LMM, Google Gemini, into the chatbot. Shadcn/ui, Tailwind CSS, and Radix UI are used to customize the UI.

Backend

Vercel Postgres is utilized to implement the database that stores user information and chat details. Vercel Blob is used to store larger files, and in this case, the MS Project files. User authentication is done by using NextAuth.js. FastAPI is utilized to build and implement the backend APIs. MPXJ is a Java library that is used to process the MS Project files into a JSON format. MPXJ is implemented using Spring Boot.

AI

An api to Google Gemini 2.0 Flash is used as the LLM.

The information from the MS project file is retrieved, processed, and edited using a combination of AI and Python functions. The LLM processes the prompts and decides which Python function to utilize to get the desired output. The Python functions are implemented to get a 100% accurate output every time.

IV. Alternative Solutions

1. Using in-built Copilot MS-project

   MS Project has built-in Copilot functionality that can be an alternative to the chatbot. It provides a seamless integration. There is no need to save and upload the MPP file.

   The Copilot used is in its early stages of development and can only provide correct outputs for simple prompts. For even slightly complex tasks, it starts giving errors or outright fails to acknowledge the tasks.

2. Fully LLM-based

   Using an LLM trained on a huge dataset of MS project files. As the LLM receives more and more data, it will become more accurate in giving the desired output. Eventually, it will achieve a very high accuracy and will work for all the prompts, including the edge cases.

   This approach requires a huge amount of resources and will be highly costly. Training an LLM will need a lot of time, and making sure that it achieves high accuracy will require even more time and money.

3. Fully Python-based

   A properly implemented Python-based approach will result in highly accurate outputs. It can be fully customizable and incorporate any custom algorithms needed for the project analysis. The cost required to implement will be relatively less than the LLM-based solution.

   Properly implementing a Python-based approach can be highly complex. Being highly complex, there might be scalability issues that might arise. Furthermore, implementing a function for every scenario, every edge case, is not feasible.

Chapter 5: Development

Our solution involves recognition of the problem and creation of tailored functions meant to be interpreted and used as tools by the Agentic chatbot, as requested by the sponsor, as well as a clear UI output that showcases each response. The elaborated solution contains nine functions which are: (1) moving a task forward by a number of days, (2) moving a task backwards by a number of days, (3) getting the tasks that start or end on weekends given a month and a year, (4) getting tasks that start or end on weekends given only the year, (5) seeing if a task lands or not on a weekend, (6) seeing if a task lands or not on a US holiday, (7) getting tasks that contain a US holiday given a specific year, (8) getting tasks that contain a US holiday given both a year and a month, and finally (9) seeing if a task goes through a US holiday.

By implementing these functions, we satisfy Merck's requirements for the project. The functionality and outputs of the chatbot will now be demonstrated.

User Manual:

I. Deployment

- By the time of usage, three servers must be running: the NextJS server, the Maven server, and the Python server.

II. User Interface Display

- The side navigation bar:
  - This navbar will allow the user to navigate through the current and previous conversations with the chatbot. If any conversation is clicked, the entire conversation history will be displayed on the right part of the screen.

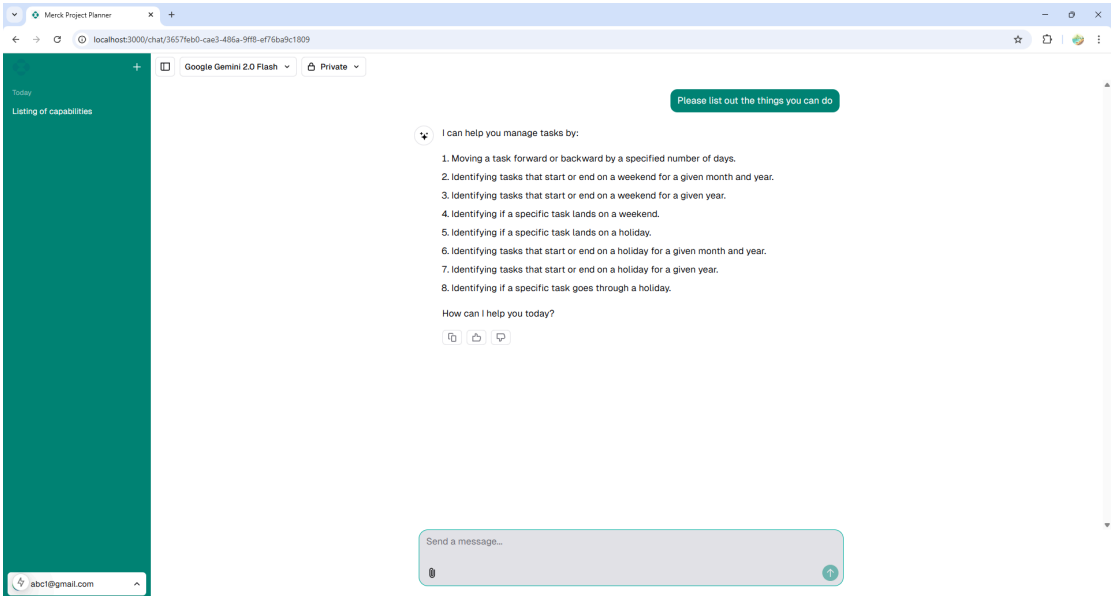
- The input field:
  - This box allows the user to input their query. It also contains an arrow-like button that, if clicked, sends and saves the current message to the chatbot on the current conversation.
- Message history:
  - The message history is displayed each time the user opens up a saved chat. The user can explore past messages between the LLM and themselves, being able to edit any message and prompt the LLM at any point.
- Profile menu:
  - Located at the bottom left of the screen, the simple profile menu allows the user to log out and switch between light mode and dark mode for the chatbot's appearance.
- Message Bubbles
  - These bubbles represent either the user’s input or the chatbot’s messages. In the case of the user’s input, it features an edit button where, if necessary, the user can change their input and prompt for another response.
- The Login Page:
  - If the user is not logged in, the chatbot will prompt for a username and password on a single page before accessing the chatbot’s functionalities.

III. Chatbot Functionalities

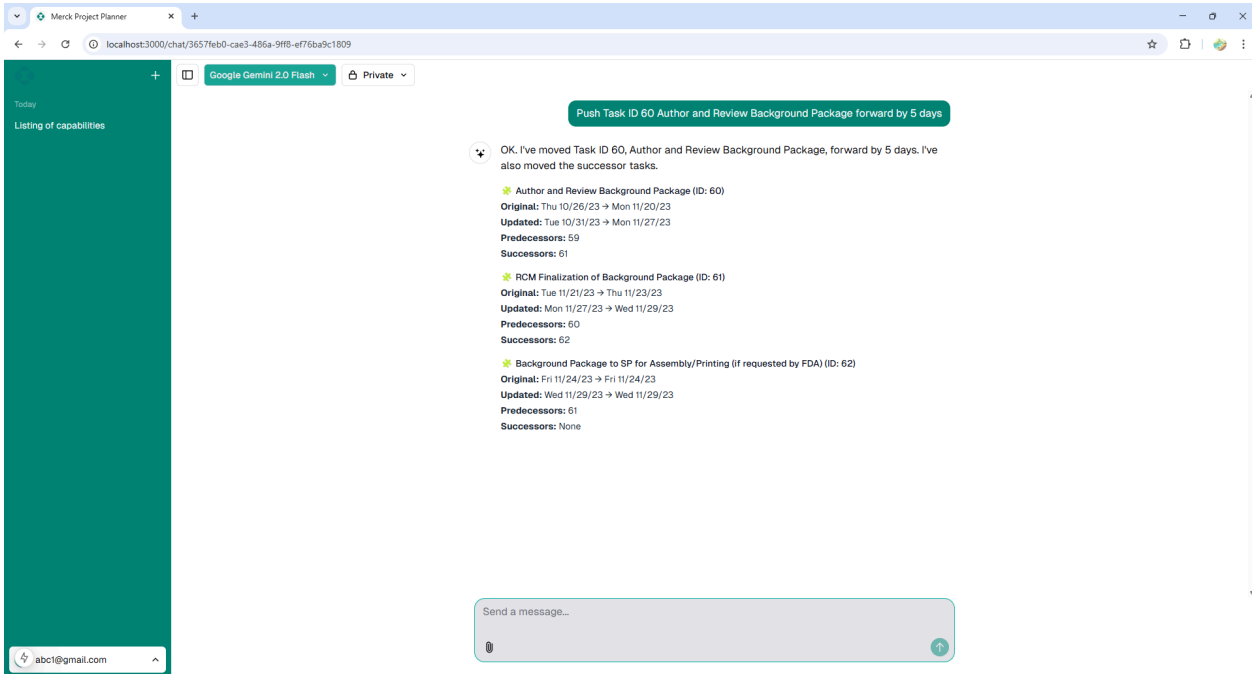
- To better understand each functionality, let’s provide examples for each function.

- Upload Microsoft Project File: First, we need to upload the Microsoft Project file we intend to use.
  - Click on the file icon
  - Send the file using the arrow-like button
  - If the loading animation disappears, it means that the file was uploaded successfully. Otherwise, a pop-up in the top part of the screen will appear, and in such a case, opening a new chat or checking the file format to be a Microsoft Project file (.mpp) would solve the issue.

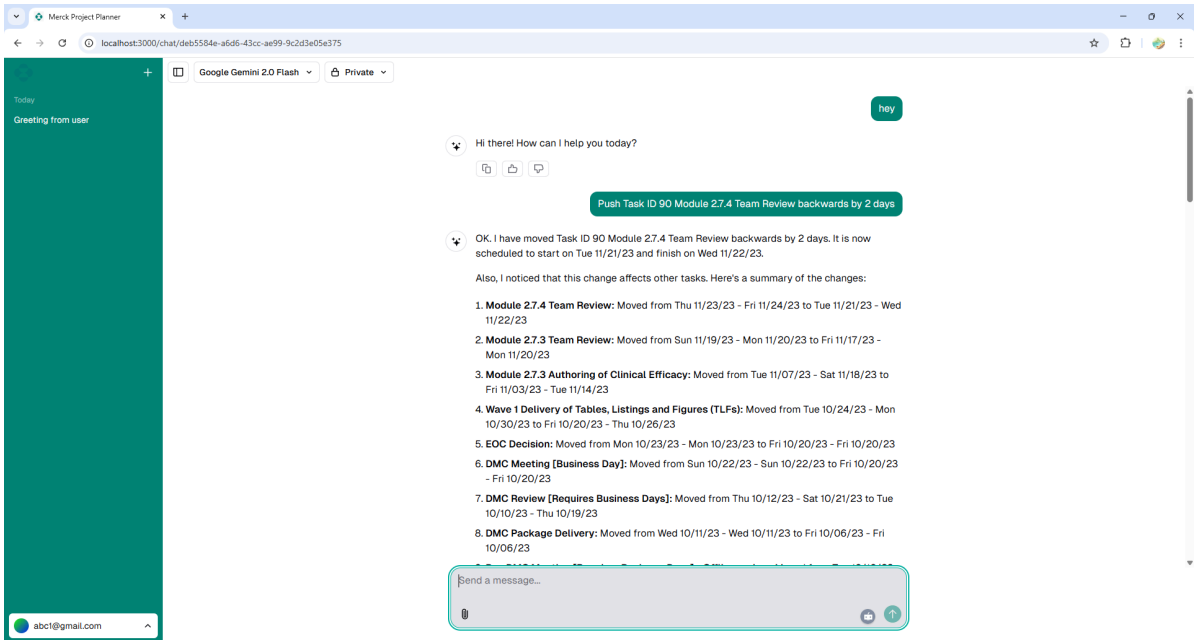
- Get to see all the chatbot's functionalities. To see an enumerated list of all the functions at the user's disposal, they can prompt: "Please list out the things you can do" to which the chatbot will answer:

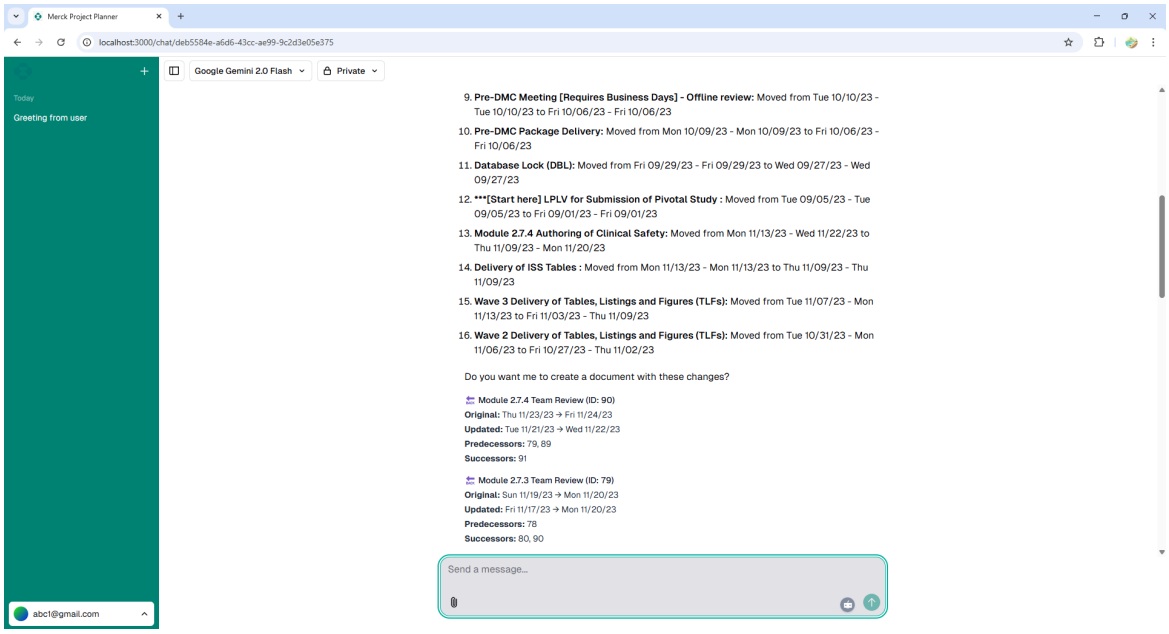


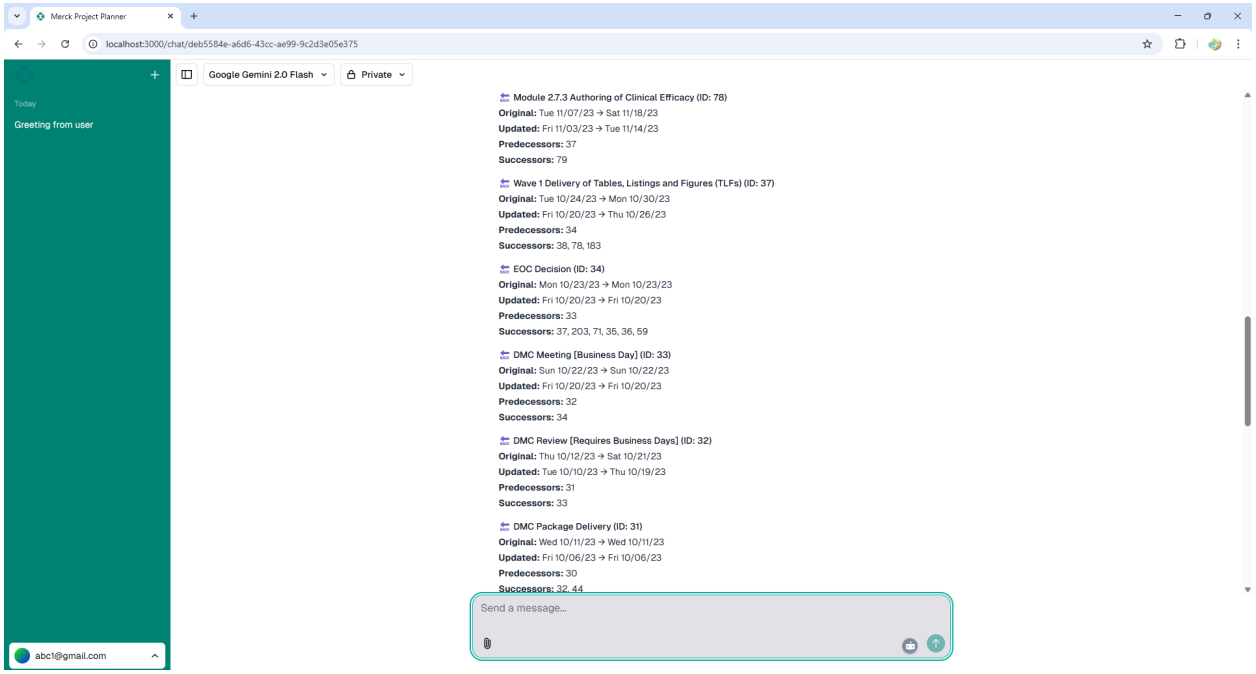
- Moving a task forward by a number of days: Let's say we want to move the task with ID 60 forward by 5 days, so we would prompt: "Push Task ID 60 Author and Review Background Package forward by 5 days" to which the chatbot will answer:

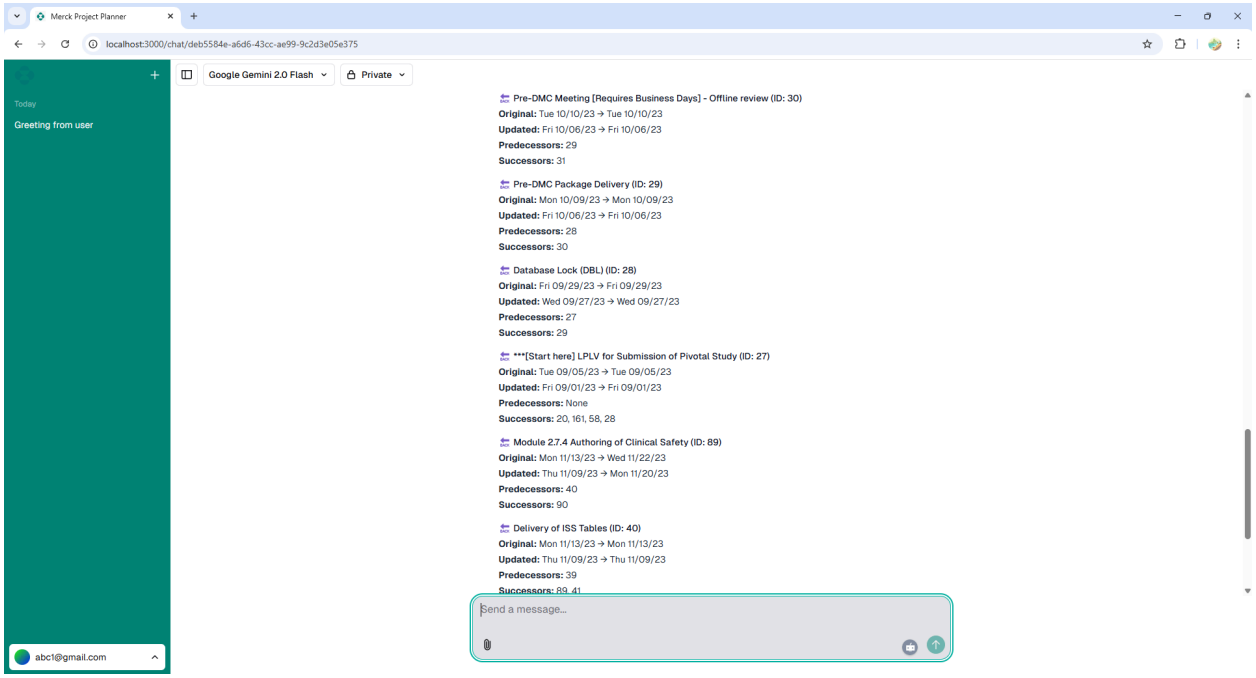


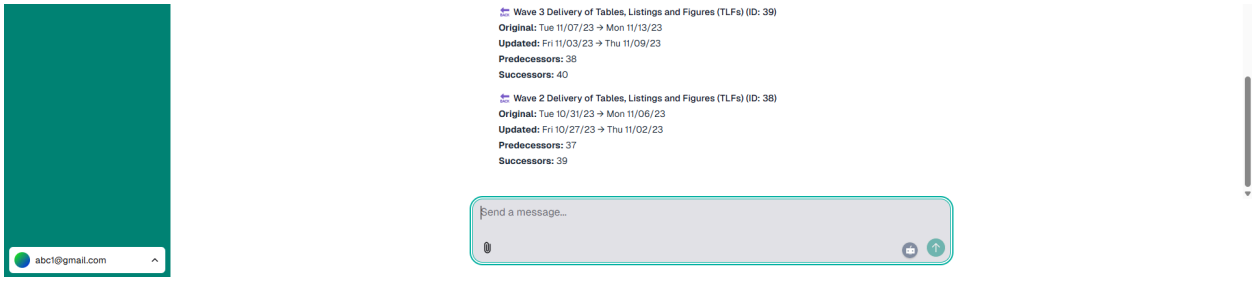
- Moving a task backwards by a number of days: Let's say we want to move the task with ID 90 backwards by 2 days, so we would prompt: "Push Task ID 90 Module 2.7.4 Team Review backwards by 2 days" to which the chatbot will answer:


Merck Project Planner
localhost:3000/chat/deb5584e-a6d6-43cc-ae99-9c2d3e05e375
Google Gemini 2.0 Flash
Private
Today
Greeting from user
abc1@gmail.com
hey
Hi there! How can I help you today?
Push Task ID 90 Module 2.7.4 Team Review backwards by 2 days
OK. I have moved Task ID 90 Module 2.7.4 Team Review backwards by 2 days. It is now scheduled to start on Tue 11/21/23 and finish on Wed 11/22/23.
Also, I noticed that this change affects other tasks. Here's a summary of the changes:
1. Module 2.7.4 Team Review: Moved from Thu 11/23/23 - Fri 11/24/23 to Tue 11/21/23 - Wed 11/22/23
2. Module 2.7.3 Team Review: Moved from Sun 11/19/23 - Mon 11/20/23 to Fri 11/17/23 - Mon 11/20/23
3. Module 2.7.3 Authoring of Clinical Efficacy: Moved from Tue 11/07/23 - Sat 11/18/23 to Fri 11/03/23 - Tue 11/14/23
4. Wave 1 Delivery of Tables, Listings and Figures (TLFs): Moved from Tue 10/24/23 - Mon 10/30/23 to Fri 10/20/23 - Thu 10/26/23
5. EOC Decision: Moved from Mon 10/23/23 - Mon 10/23/23 to Fri 10/20/23 - Fri 10/20/23
6. DMC Meeting [Business Day]: Moved from Sun 10/22/23 - Sun 10/22/23 to Fri 10/20/23 - Fri 10/20/23
7. DMC Review [Requires Business Days]: Moved from Thu 10/12/23 - Sat 10/21/23 to Tue 10/10/23 - Thu 10/19/23
8. DMC Package Delivery: Moved from Wed 10/11/23 - Wed 10/11/23 to Fri 10/06/23 - Fri 10/06/23
Send a message...

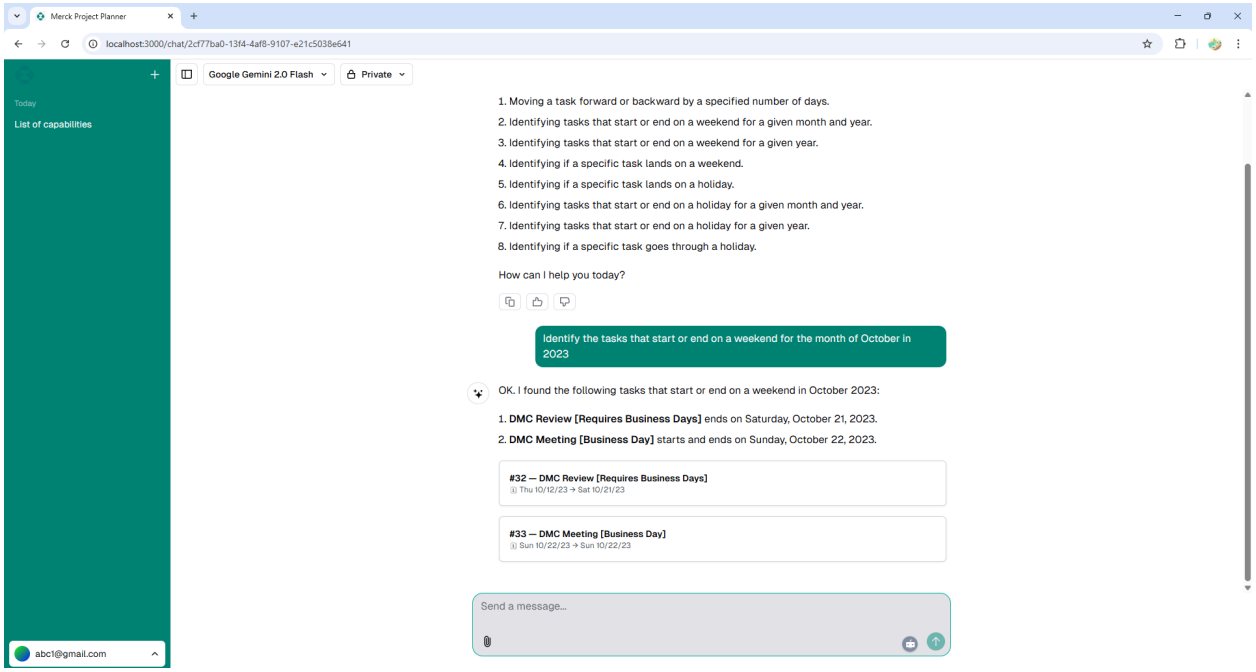

Merck Project Planner
localhost:3000/chat/deb5584e-a6d6-43cc-ae99-9c2d3e05e375
Google Gemini 2.0 Flash
Private
Today
Greeting from user
abc1@gmail.com
9. Pre-DMC Meeting [Requires Business Days] - Offline review: Moved from Tue 10/10/23 - Tue 10/10/23 to Fri 10/06/23 - Fri 10/06/23
10. Pre-DMC Package Delivery: Moved from Mon 10/09/23 - Mon 10/09/23 to Fri 10/06/23 - Fri 10/06/23
11. Database Lock (DBL): Moved from Fri 09/29/23 - Fri 09/29/23 to Wed 09/27/23 - Wed 09/27/23
12. ***[Start here] LPLV for Submission of Pivotal Study : Moved from Tue 09/05/23 - Tue 09/05/23 to Fri 09/01/23 - Fri 09/01/23
13. Module 2.7.4 Authoring of Clinical Safety: Moved from Mon 11/13/23 - Wed 11/22/23 to Thu 11/09/23 - Mon 11/20/23
14. Delivery of ISS Tables : Moved from Mon 11/13/23 - Mon 11/13/23 to Thu 11/09/23 - Thu 11/09/23
15. Wave 3 Delivery of Tables, Listings and Figures (TLFs): Moved from Tue 11/07/23 - Mon 11/13/23 to Fri 11/03/23 - Thu 11/09/23
16. Wave 2 Delivery of Tables, Listings and Figures (TLFs): Moved from Tue 10/31/23 - Mon 11/06/23 to Fri 10/27/23 - Thu 11/02/23
Do you want me to create a document with these changes?
Module 2.7.4 Team Review (ID: 90)
Original: Thu 11/23/23 → Fri 11/24/23
Updated: Tue 11/21/23 → Wed 11/22/23
Predecessors: 79, 89
Successors: 91
Module 2.7.3 Team Review (ID: 79)
Original: Sun 11/19/23 → Mon 11/20/23
Updated: Fri 11/17/23 → Mon 11/20/23
Predecessors: 78
Successors: 80, 90
Send a message...


Merck Project Planner
localhost:3000/chat/deb5584e-a6d6-43cc-ae99-9c2d3e05e375
Google Gemini 2.0 Flash
Private
Today
Greeting from user
Module 2.7.3 Authoring of Clinical Efficacy (ID: 78)
Original: Tue 11/07/23 → Sat 11/18/23
Updated: Fri 11/03/23 → Tue 11/14/23
Predecessors: 37
Successors: 79
Wave 1 Delivery of Tables, Listings and Figures (TLFs) (ID: 37)
Original: Tue 10/24/23 → Mon 10/30/23
Updated: Fri 10/20/23 → Thu 10/26/23
Predecessors: 34
Successors: 38, 78, 183
EOC Decision (ID: 34)
Original: Mon 10/23/23 → Mon 10/23/23
Updated: Fri 10/20/23 → Fri 10/20/23
Predecessors: 33
Successors: 37, 203, 71, 35, 36, 59
DMC Meeting [Business Day] (ID: 33)
Original: Sun 10/22/23 → Sun 10/22/23
Updated: Fri 10/20/23 → Fri 10/20/23
Predecessors: 32
Successors: 34
DMC Review [Requires Business Days] (ID: 32)
Original: Thu 10/12/23 → Sat 10/21/23
Updated: Tue 10/10/23 → Thu 10/19/23
Predecessors: 31
Successors: 33
DMC Package Delivery (ID: 31)
Original: Wed 10/11/23 → Wed 10/11/23
Updated: Fri 10/06/23 → Fri 10/06/23
Predecessors: 30
Successors: 32, 44
Send a message...
abc1@gmail.com

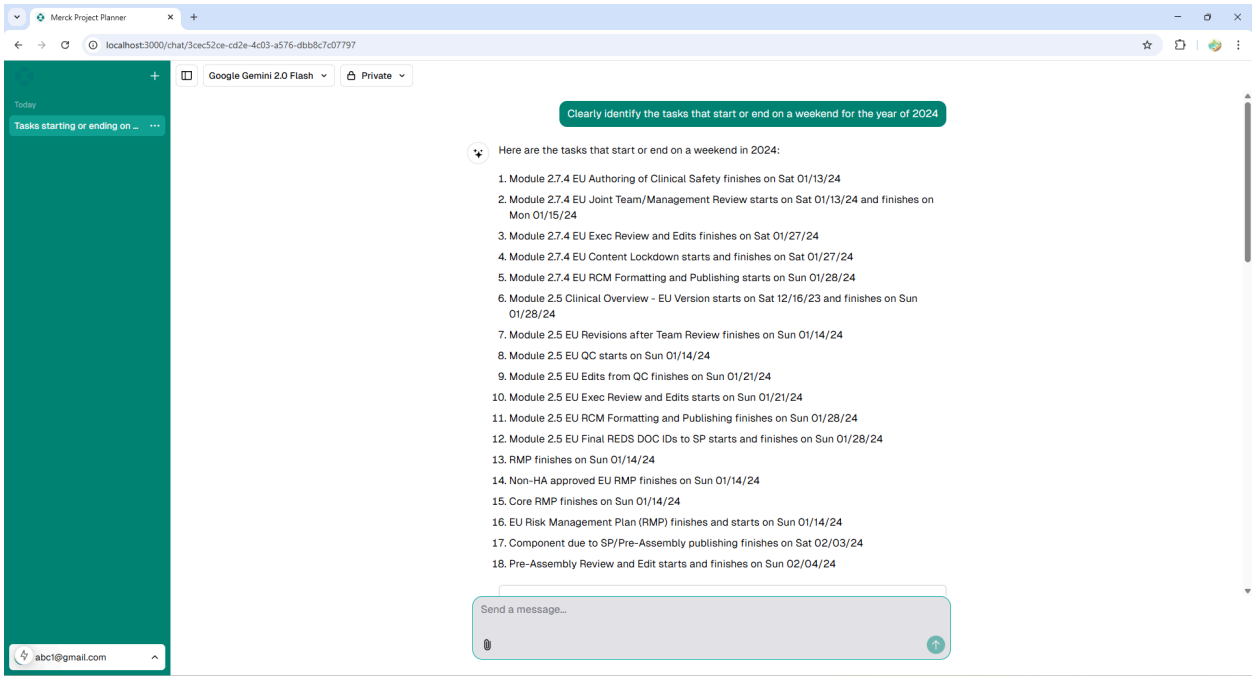

Merck Project Planner
localhost:3000/chat/deb5584e-a6d6-43cc-ae99-9c2d3e05e375
Google Gemini 2.0 Flash
Private
Today
Greeting from user
Pre-DMC Meeting [Requires Business Days] - Offline review (ID: 30)
Original: Tue 10/10/23 → Tue 10/10/23
Updated: Fri 10/06/23 → Fri 10/06/23
Predecessors: 29
Successors: 31
Pre-DMC Package Delivery (ID: 29)
Original: Mon 10/09/23 → Mon 10/09/23
Updated: Fri 10/06/23 → Fri 10/06/23
Predecessors: 28
Successors: 30
Database Lock (DBL) (ID: 28)
Original: Fri 09/29/23 → Fri 09/29/23
Updated: Wed 09/27/23 → Wed 09/27/23
Predecessors: 27
Successors: 29
***[Start here] LPLV for Submission of Pivotal Study (ID: 27)
Original: Tue 09/05/23 → Tue 09/05/23
Updated: Fri 09/01/23 → Fri 09/01/23
Predecessors: None
Successors: 20, 161, 58, 28
Module 2.7.4 Authoring of Clinical Safety (ID: 89)
Original: Mon 11/13/23 → Wed 11/22/23
Updated: Thu 11/09/23 → Mon 11/20/23
Predecessors: 40
Successors: 90
Delivery of ISS Tables (ID: 40)
Original: Mon 11/13/23 → Mon 11/13/23
Updated: Thu 11/09/23 → Thu 11/09/23
Predecessors: 39
Successors: 89, 41
Send a message...
abc1@gmail.com

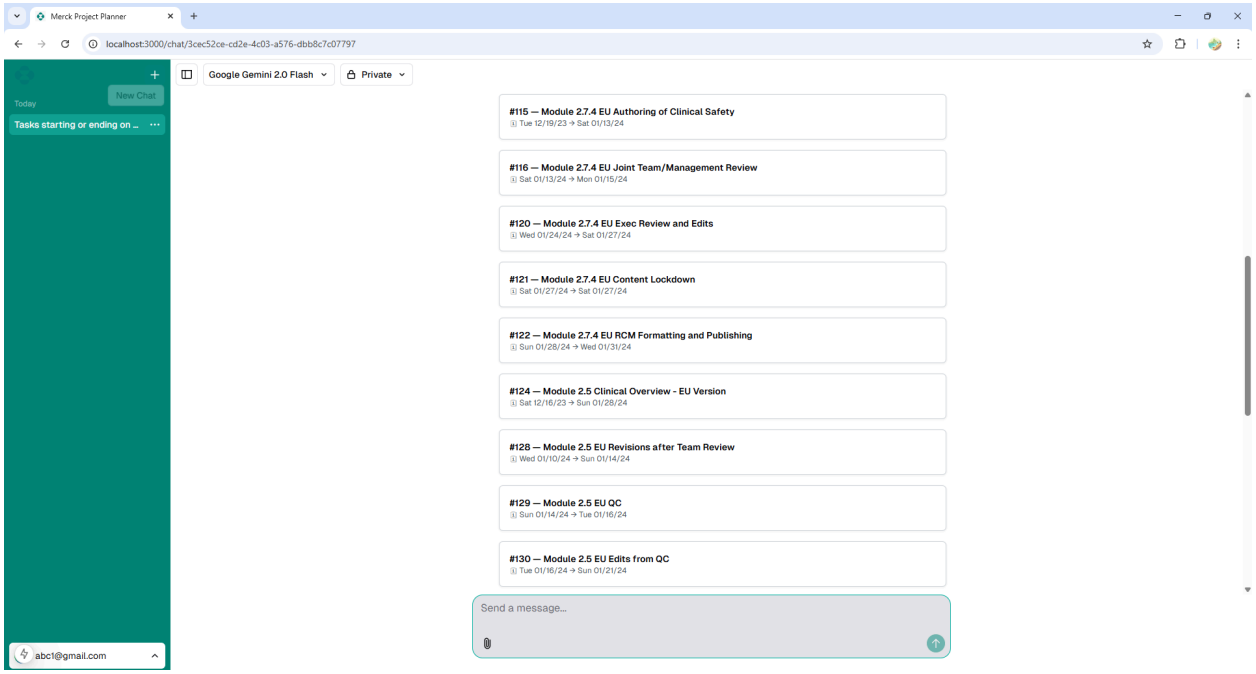

Wave 3 Delivery of Tables, Listings and Figures (TLFs) (ID: 39)
Original: Tue 11/07/23 → Mon 11/13/23
Updated: Fri 11/03/23 → Thu 11/09/23
Predecessors: 38
Successors: 40
Wave 2 Delivery of Tables, Listings and Figures (TLFs) (ID: 38)
Original: Tue 10/31/23 → Mon 11/06/23
Updated: Fri 10/27/23 → Thu 11/02/23
Predecessors: 37
Successors: 39
Send a message...
abc1@gmail.com

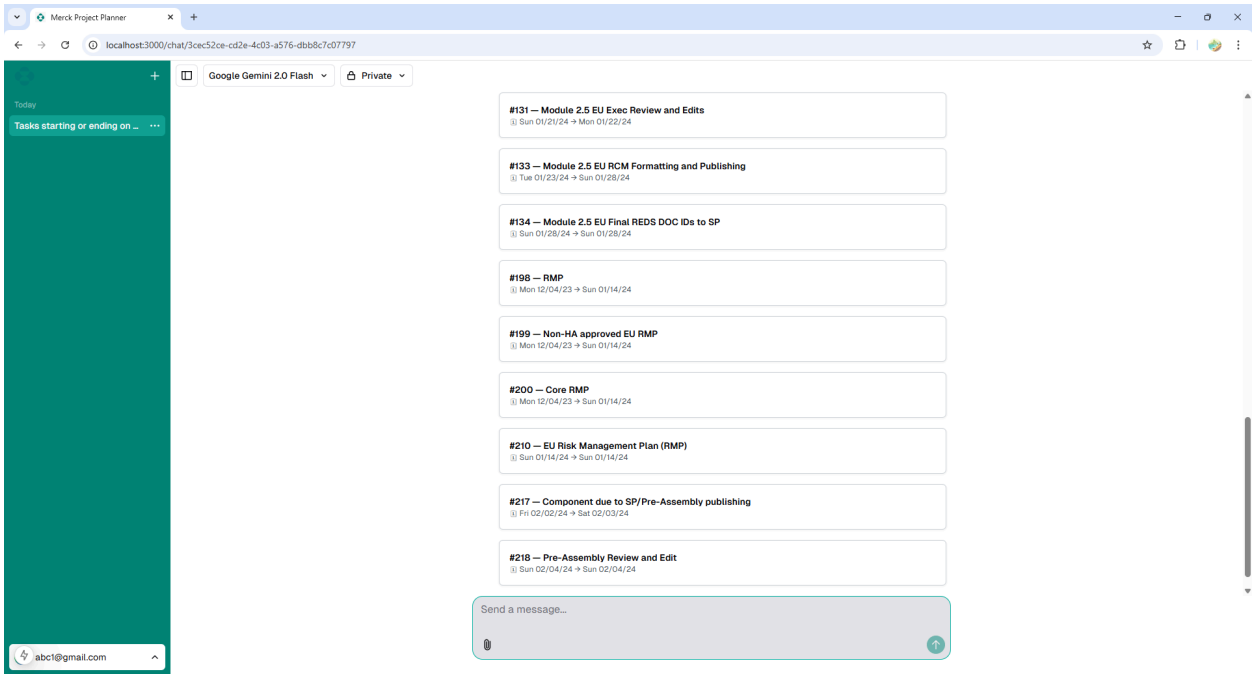
- Getting the tasks that start or end on a weekend given a month and a year: Let's say we want to choose October of 2023, so we would prompt: "Identify the tasks that start or end on a weekend for the month of October in 2023" to which the chatbot will answer:



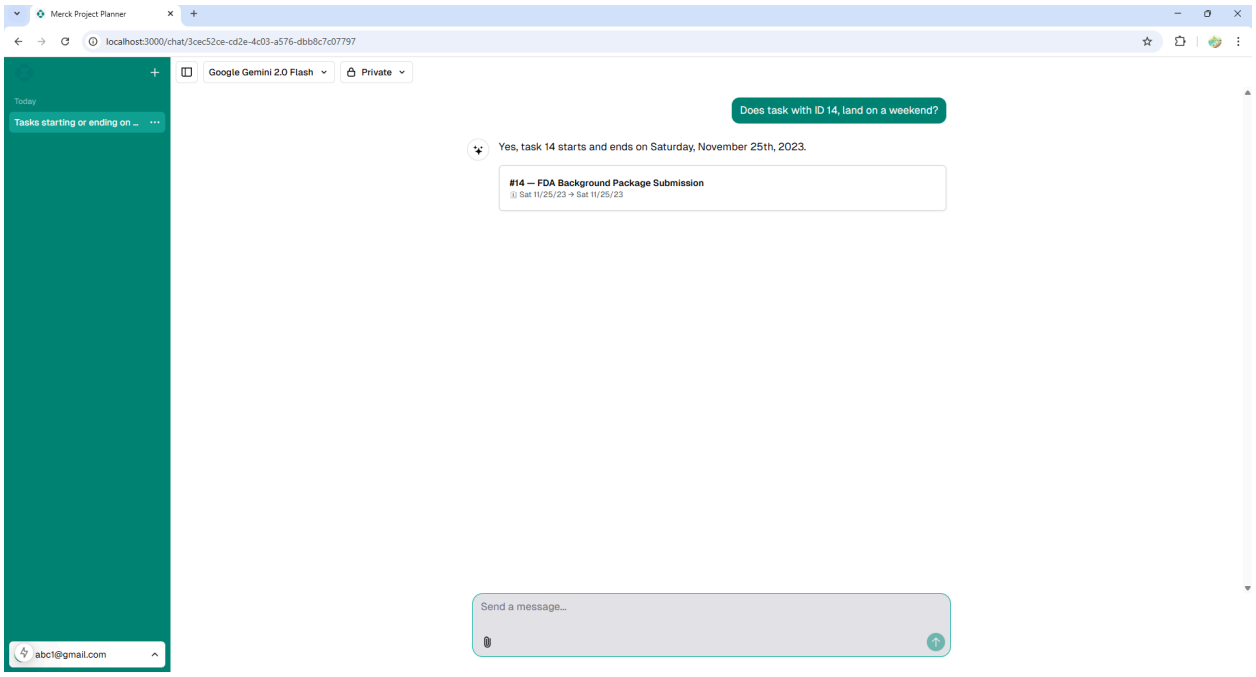
- Getting the tasks that start or end on a weekend given only the year: Let's say we want to choose the year 2024, so we would prompt: "Clearly identify the tasks that start or end on a weekend for the year of 2024" to which the chatbot will answer:


Merck Project Planner
localhost:3000/chat/3cec52ce-cd2e-4c03-a576-dbb8c7c07797
Google Gemini 2.0 Flash
Private
Today
Tasks starting or ending on ...
abc1@gmail.com
Clearly identify the tasks that start or end on a weekend for the year of 2024
Here are the tasks that start or end on a weekend in 2024:
1. Module 2.7.4 EU Authoring of Clinical Safety finishes on Sat 01/13/24
2. Module 2.7.4 EU Joint Team/Management Review starts on Sat 01/13/24 and finishes on Mon 01/15/24
3. Module 2.7.4 EU Exec Review and Edits finishes on Sat 01/27/24
4. Module 2.7.4 EU Content Lockdown starts and finishes on Sat 01/27/24
5. Module 2.7.4 EU RCM Formatting and Publishing starts on Sun 01/28/24
6. Module 2.5 Clinical Overview - EU Version starts on Sat 12/16/23 and finishes on Sun 01/28/24
7. Module 2.5 EU Revisions after Team Review finishes on Sun 01/14/24
8. Module 2.5 EU QC starts on Sun 01/14/24
9. Module 2.5 EU Edits from QC finishes on Sun 01/21/24
10. Module 2.5 EU Exec Review and Edits starts on Sun 01/21/24
11. Module 2.5 EU RCM Formatting and Publishing finishes on Sun 01/28/24
12. Module 2.5 EU Final REDS DOC IDs to SP starts and finishes on Sun 01/28/24
13. RMP finishes on Sun 01/14/24
14. Non-HA approved EU RMP finishes on Sun 01/14/24
15. Core RMP finishes on Sun 01/14/24
16. EU Risk Management Plan (RMP) finishes and starts on Sun 01/14/24
17. Component due to SP/Pre-Assembly publishing finishes on Sat 02/03/24
18. Pre-Assembly Review and Edit starts and finishes on Sun 02/04/24
Send a message...

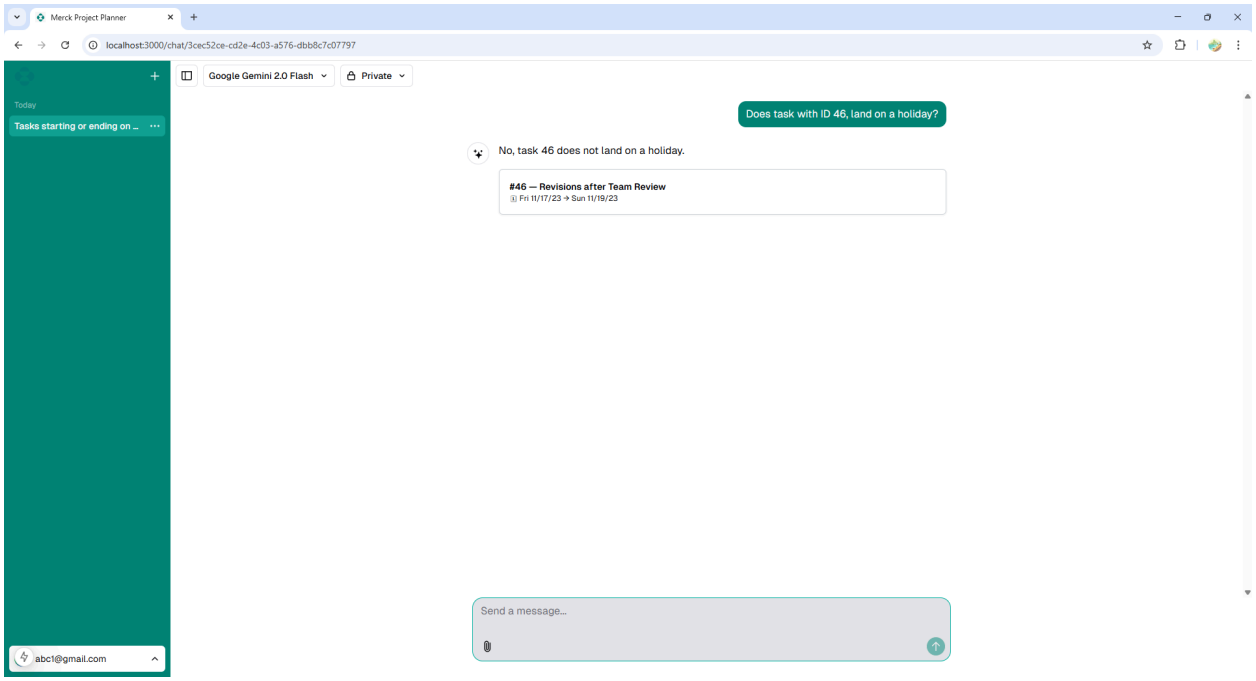

Merck Project Planner
localhost:3000/chat/3cec52ce-cd2e-4c03-a576-dbb8c7c07797
Google Gemini 2.0 Flash
Private
New Chat
Today
Tasks starting or ending on ...
abc1@gmail.com
#115 — Module 2.7.4 EU Authoring of Clinical Safety
Tue 12/19/23 → Sat 01/13/24
#116 — Module 2.7.4 EU Joint Team/Management Review
Sat 01/13/24 → Mon 01/15/24
#120 — Module 2.7.4 EU Exec Review and Edits
Wed 01/24/24 → Sat 01/27/24
#121 — Module 2.7.4 EU Content Lockdown
Sat 01/27/24 → Sat 01/27/24
#122 — Module 2.7.4 EU RCM Formatting and Publishing
Sun 01/28/24 → Wed 01/31/24
#124 — Module 2.5 Clinical Overview - EU Version
Sat 12/16/23 → Sun 01/28/24
#128 — Module 2.5 EU Revisions after Team Review
Wed 01/10/24 → Sun 01/14/24
#129 — Module 2.5 EU QC
Sun 01/14/24 → Tue 01/16/24
#130 — Module 2.5 EU Edits from QC
Tue 01/16/24 → Sun 01/21/24
Send a message...



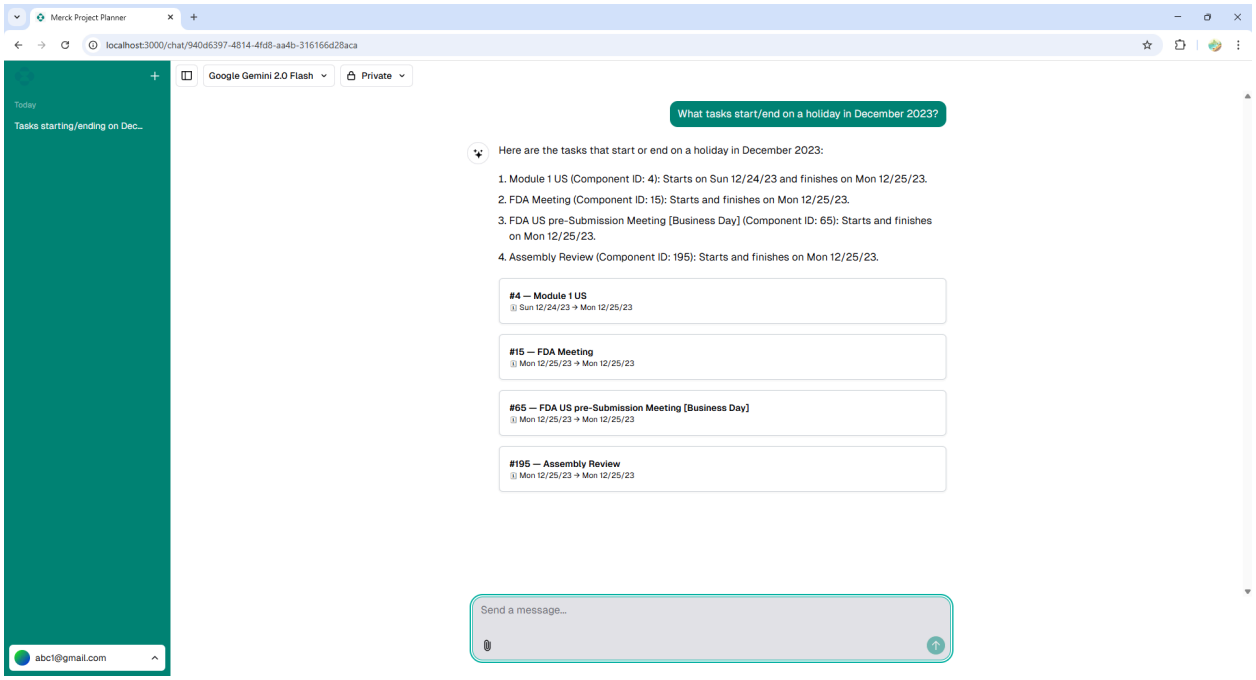
- Seeing if a task lands or not on a weekend: Let's say we want to check if a specific task has a start or end date falling on a weekend, so we would prompt: "Does task with ID 14 land on a weekend?" to which the chatbot will answer:



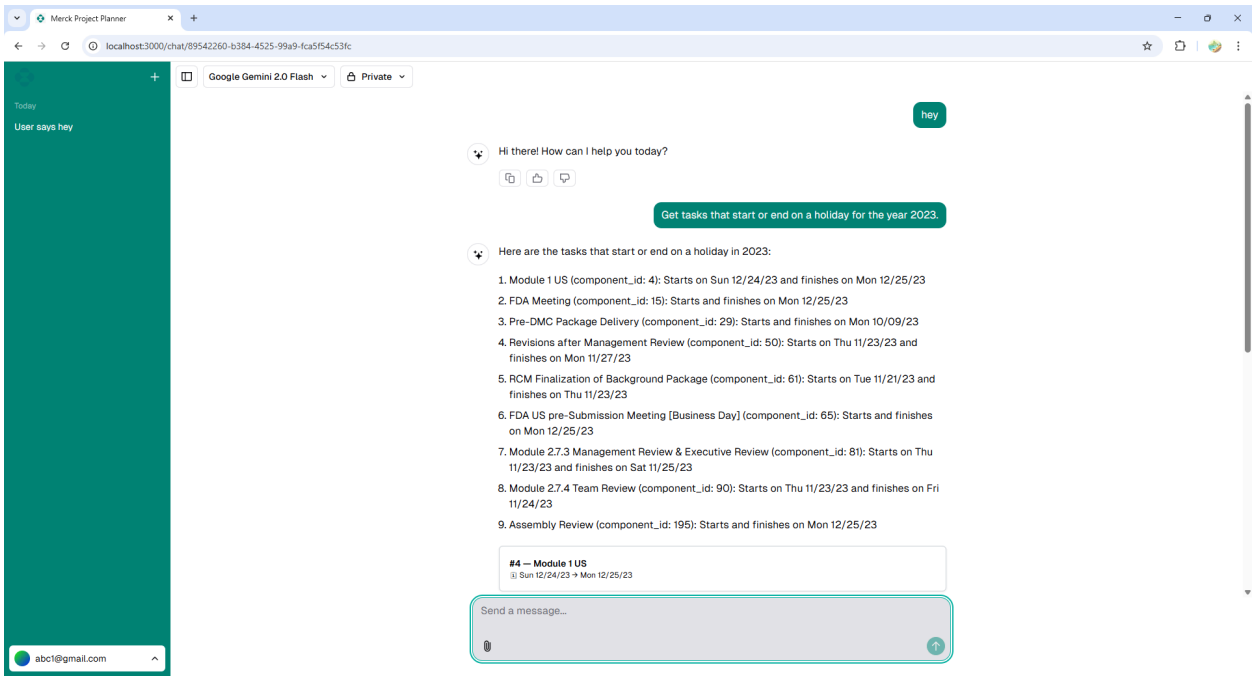
- Seeing if a task lands or not on a holiday: Let's say we want to check if a specific task has a start or end date falling on a holiday, so we would prompt: "Does task with ID 46, land on a holiday?" to which the chatbot will answer:

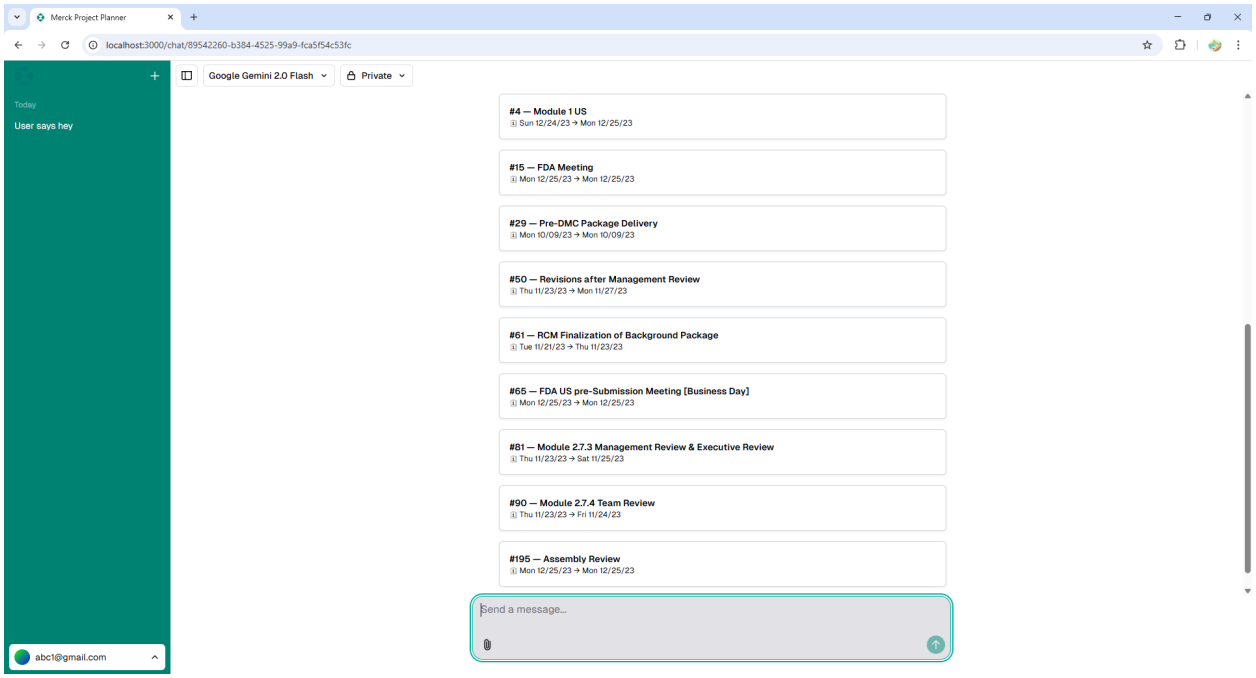


- Getting tasks that start or end on a holiday for a given month and year: Let's say we want to choose December of 2023, so we would prompt: "What tasks start/end on a holiday in December 2023?" to which the chatbot will answer:

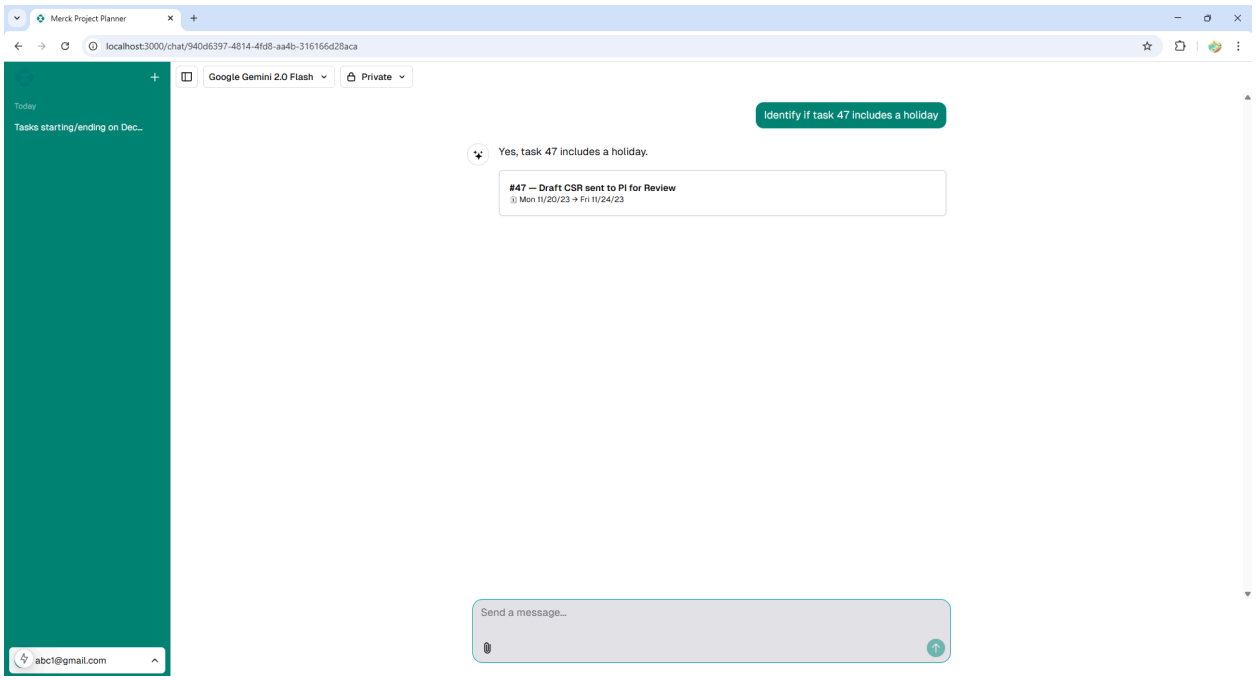


- Getting tasks that start or end on a holiday given only the year: Let's say we want to choose the year 2023, so we would prompt: "Get tasks that start or end on a holiday for the year 2023" to which the chatbot will answer:





- Seeing if a task goes through a holiday: Let's say we want to make sure that a specific task includes a holiday, so we would prompt: "Identify if task 47 includes a holiday", to which the chatbot will answer:



Chapter 6: Evaluation & Conclusions

I. Solution Testing

To verify that our solution worked accurately, we manually tested the chatbot with a variety of scenarios. Ryan Blanton, our sponsor from Merck, specifically requested that we manually validate and verify the results using a calendar and ChatGPT. He had requested this because of Merck having critical and strict FDA deadlines that could not be missed. Therefore, Ryan emphasized that automation alone could not be trusted, and the best way to accurately accomplish testing was to do it manually.

Each team member conducted multiple tests for all of the functions implemented within our chatbot. This includes shifting tasks forward and backwards an X amount of days, skipping weekends and holidays. The team also verified that successor and predecessor tasks were also being updated accordingly. Every function was tested repeatedly using different tasks and shift intervals in order to evaluate our solution's accuracy and efficiency.

Figure 6.1 below shows the chatbot's response to the following prompt: "Push Task with ID 60 'Author and Review Background Package' forward by 2 days." As you can see the original task 60 starts on Thursday, October 26, 2023 and ends on Monday, November 20, 2023. When shifted forward by two days, the new start date becomes Monday, October 30, 2023 and the new end date is Friday, November 24, 2023. Along with that, both successor tasks (IDs 61 and 62) were also automatically updated to reflect this change.

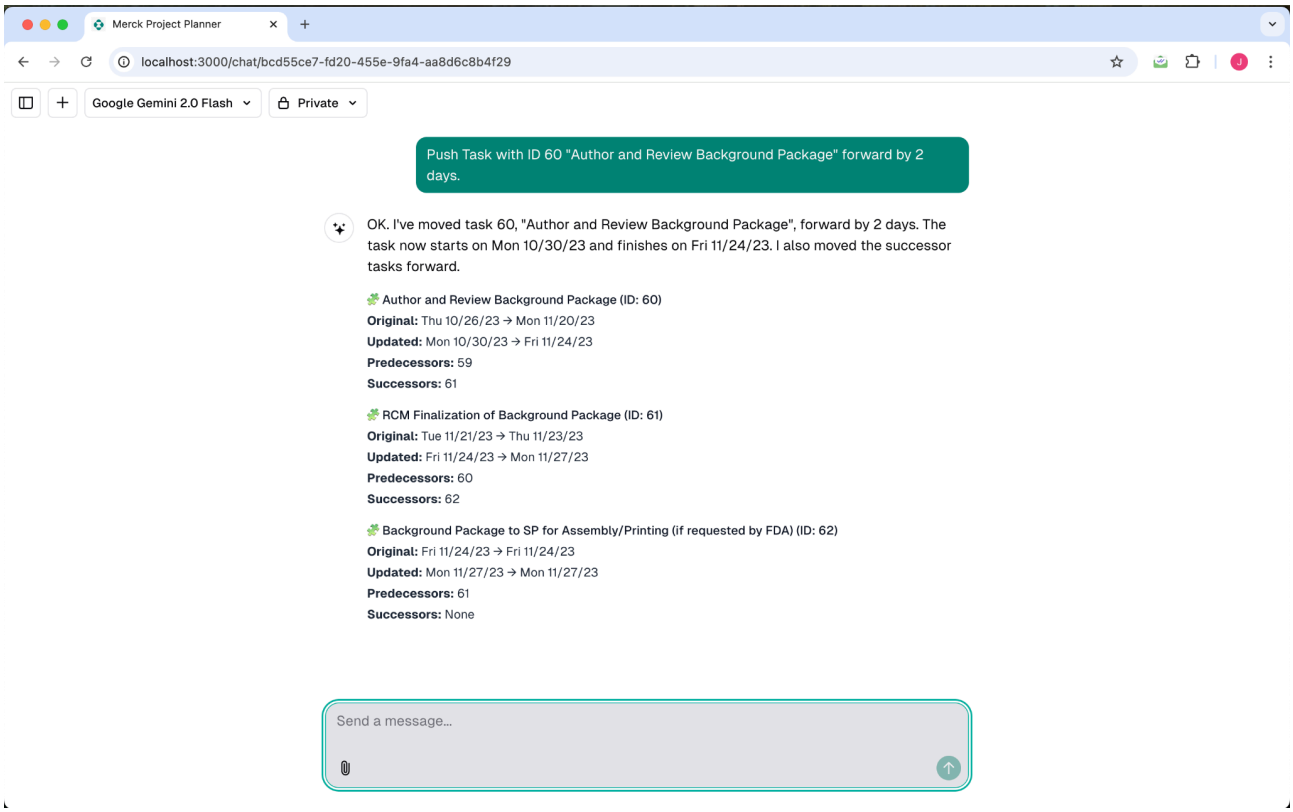


Figure 6.1: Output when a task is shifted forward by two days.

To verify this output, the calendar below (Figure 6.2) was used to manually verify the output of Figure 6.1 above. Shifting task 60 forward by two days would land on Saturday, October 28, 2023, but our solution skips weekends by default. Therefore, the new start date is Monday, October 30, 2023. This aligns with the chatbot's response and therefore the response given in Figure 6.1 is correct.

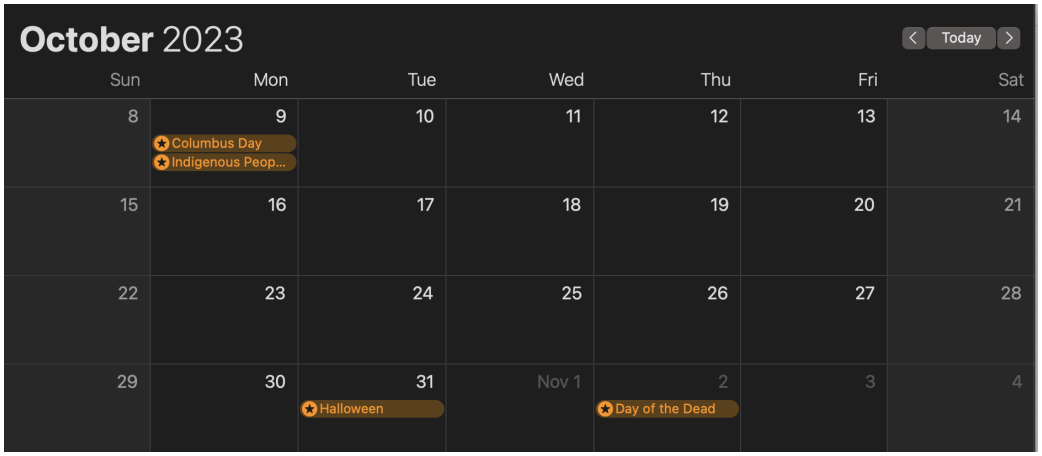


Figure 6.2: Calendar of October 2023 verifying Figure 6.1 output

II. Verification

During the verification process, the team ensured that each individual function/feature of our solution worked as intended. This included reviewing the backend logic to verify that the scheduling rules were accurately defined and implemented correctly. We also confirmed that when prompting the AI, it understood what was being asked and it triggered the appropriate function.

We verified that when shifting tasks forward/backward it would exclude weekends and holidays by default, we checked that the duration between tasks remained the same and finally we ensured that all successor/predecessor tasks were updated accordingly.

Additionally, as a team we verified that Gemini was accurately able to parse and interpret different prompt phrasings. For instance, we ensured that both of the following prompts would trigger the same function: “shift task by 3 days” and “move task forward by 3 days.”

The goal of this process was to check if the backend logic was correct and that when the AI is prompted it is accurately able to trigger the correct function.

III. Validation

The validation portion focused on confirming that our solution met Merck's requirements and expectations. While verification was the process of ensuring each and every function worked as intended, validation on the other hand was making sure that the solution was reliable, efficient and beneficial for the sponsor and what they had expected.

The main goal for this project was to let users prompt various scenarios in which the timeline would be altered and instantly reveal the changes made. The chatbot needed to shift tasks properly, handle holidays and weekends, and update dependent tasks accordingly so that Merck submission planners could make informed and timely decisions.

Throughout the duration of the project, we met with our sponsor on a weekly basis to demo features and walk through the different test scenarios. Ryan provided feedback on how the solution performed, allowing the team to make changes based on his input.

Additionally, we validated our solution by using multiple datasets. We created different MPP files with different number of tasks, dependencies, etc. to ensure our solution could handle alternate forms of data and still provide accurate results efficiently.

The chatbot UI was also validated for readability, making sure all users of all technical levels could easily understand and interact with the chatbot.

IV. Team Conclusions

Percy Flores

“An amazing experience working with great teammates for a company like Merck that taught me the importance of prototyping and carrying a concept into a minimum viable product meant to inspire industry-level developers at Merck; further empowering the company and saving precious time for project managers. On a personal level, this project has pushed me to come up with a viable architecture within the first week, taking into consideration many variables and consolidating on a single approach, having foreseen other solutions’ pros and cons. The development of this project has opened my mind to think on a large scale and execute on a project with industry-level impact. I’m grateful to Merck for letting me take part in this opportunity to develop my programming abilities and strengthen my software and peer management skills.”

Japjot Bedi

“This project gave me the opportunity to lead a team through a real-world software development project from start to finish. As Project Manager, I learned how to keep everyone on the same page, stay in contact with the sponsor, and make sure the project was headed in the right direction. As a developer, I got the opportunity to work more closely with AI. It showed me how unreliable AI could be if not fine-tuned properly. Overall, I learned a lot from this project, whether it was refining my technical skills, leading a team, or just collaborating with my team to

build a product with a real purpose. I'm grateful for the opportunity to work on this project with Merck and my teammates, and for everything I learned throughout the semester."

Tanvir Longia

"This project helped me significantly in understanding and learning about various topics. Working in a team, communicating, and brainstorming helped me grow as a team player. On an individual level, this project allowed me to refine my skills required to be a full-stack developer. This project provided us with an opportunity to familiarize ourselves with and work with the latest technologies that are currently used in the industry."

Ricardo Cruz

"This project taught me how to integrate AI into real world applications. One of the biggest takeaways for me was understanding the power and limitations of large language models. I initially assumed that AI could handle most logic through prompting but quickly realized that's not always the case. We ended up taking a hybrid approach, combining AI with backend logic to ensure reliability. From a collaborative standpoint, I also learned how to manage time constraints and work effectively within a team with diverse technical skills."

Nicholas Mustrat

"This project helped reveal the many difficulties that industry professionals face today in regard to scheduling. Not only were we tasked with developing a tool in which its purpose is for scheduling a professional project, the process of creating such a tool equipped my team with real world experience in the realm of software engineering. Furthermore, this project exposed me to a

professional's approach when it comes to developing practical solutions. Making on the fly decisions to deliver a working project given the limitations of our resources while simultaneously collaborating with others as part of a team was an invaluable experience that will stay with me throughout my career."