Project Plan

GPT API Challenge

CS467 Online Capstone Project - Spring 2025

Alan Chan

Tavner Murphy

Masseeh Safi

Introduction

Imagine a typical day– you come back home from a long, busy day at work and your stomach begins to rumble. You ponder the idea of whether you should cook that night. Taking a glance at your fridge and your pantry full of ingredients you aren't sure what to cook up. It all seems overwhelming. The grocery store is at least 10 minutes away in traffic, and the last thing you want to do is leave the house again. Should you just order from DoorDash or UberEats? Those options tend to be unhealthy, and take money right out of the paycheck you spent all day earning. All of us are pretty familiar with this situation, right?

What if there was an app that recommends recipes to make based off of the ingredients that are already in your pantry and fridge? With the power of HomeSpice it's now possible to do just that. What makes our web app different from regular recipe recommendations is due to its simplicity. Rather than having to scroll through endless recipe blogs to find a good recipe for what you have on hand, our app helps you keep track of the ingredients you have in your pantry and generates a yummy recipe for you to cook at the press of a button. That way, rather than having to pay for expensive delivery food, you can cook a delicious meal in the comfort of your home with ingredients you already have. You can also save recipes you like so you can access them again in the future. Easy, simple, and delicious.

Problem Statement

One of the challenges faced by every adult human being daily is what to eat, and how to acquire it. In the modern world, we often have the ingredients we need in our pantry or refrigerator, but lack the time, preparation, and information to properly prepare and serve them. For people with busy schedules, such as budding software engineers, it's often difficult and time consuming to plan recipes and costly grocery store trips for ingredients we might never use again. As a result money is wasted, food spoils, and we often find ourselves returning to the ever present temptation of food delivery apps to get our caloric fix, trading away our hard earned money and health for convenience. We all have faced this problem in one way or another, and many have turned to recipe apps or meal planning to help provide a solution.

However, most meal planning tools offer a fixed road map to achieve a recipe, and do not allow the user to modify or generate a recipe based on what they already have available at hand. Pantry tracking tools usually require users to manually enter each ingredient in their pantry or fridge. Food blogs often require special trips to the grocery store for special spices or unique ingredients. As a result of all this friction, all too often the tools just over-complicate the problem, and perfectly good, healthy leftover ingredients in the pantry and refrigerator are "orphaned" and left to gather dust, and the decision of what to cook and how to make it isn't solved in a sustainable, meaningful way.

This problem affects anyone who cooks at home, from students to working professionals to families; from couples trying to stick to a budget, to new parents who lack the time to make frequent grocery store trips. We often aren't aware of how much time, efficiency, and money we

are losing out on, but we do feel it in our pocketbooks and in our gut. What if there were a way to harness large language models and their cutting edge image recognition features to provide a solution to this problem? What if instead of manually entering each ingredient in the pantry, you could just take one photo of what you have to cook with? That's where our app comes in.

With the GPT API we can have the LLM quickly generate recipes based on which ingredients the user has already available in their pantry. In this way, the painful process of searching for a recipe that matches what a user has, or making last minute trips to the grocery is taken out entirely. This reduces food waste, saves time, reduces the friction of meal planning, and makes cooking easy, simple, and fun! Users simply keep a running log of their pantry items, and the once laborious process of generating a tailored recipe that utilizes the ingredients they have available is now as simple as clicking a button. Additionally, the user will be able to save recipes they like to their account so they can access them again in the future.

Requirements

HomeSpice is a web-based smart recipe generator designed to streamline daily meal planning by automatically generating recipes based on ingredients users have on-hand, whether input via text or image. The application will integrate OpenAI's GPT API to formulate recipes, giving users the ability to auto-generate multiple options, and allowing users to save their favorite recipes to their secure user profiles. This application aims to reduce food waste by making use of partially-used pantry items and ingredients, save time by only showing recipes the user can create with ingredients on-hand, and gives users the ability to store and recall recipes for future use.

To meet these requirements, the software must be robust, scalable, and secure, with a user-friendly interface with minimal friction and simplistic navigation. The following outlines the high-level requirements/features/tasks necessary for the application and serves as the basis for development.

- UI/UX
    - Intuitive and responsive UI
    - Simple design for easy navigation and clear recipe presentation
- User Ingredient Input
    - Build input form that allows users to manually enter list of available ingredients
    - Implement input validation and preprocessing to ensure ingredients are correctly formatted for API processing.
- Recipe Generation with GPT API
    - Construct dynamic and context-aware prompts based on user-entered ingredients.
    - Integrate with OpenAI's GPT API to generate recipes
    - Allow user to auto-generate multiple recipe options
    - Allow user option to save recipes
- User Accounts & Data Management

    - Allow user authentication and secure profile management
    - Store users' pantry logs and saved recipes in an persistent database
- Performance & Scalability
    - Build backend framework with FastAPI for optimized asynchronous API calls and processing
- Stretch Goals: Image Integration
    - Develop optional module for user input of ingredients via image upload
    - Integrate with a cloud-based image recognition service for image recognition and ingredient extraction
    - Optional image generation to compliment AI-generated recipes

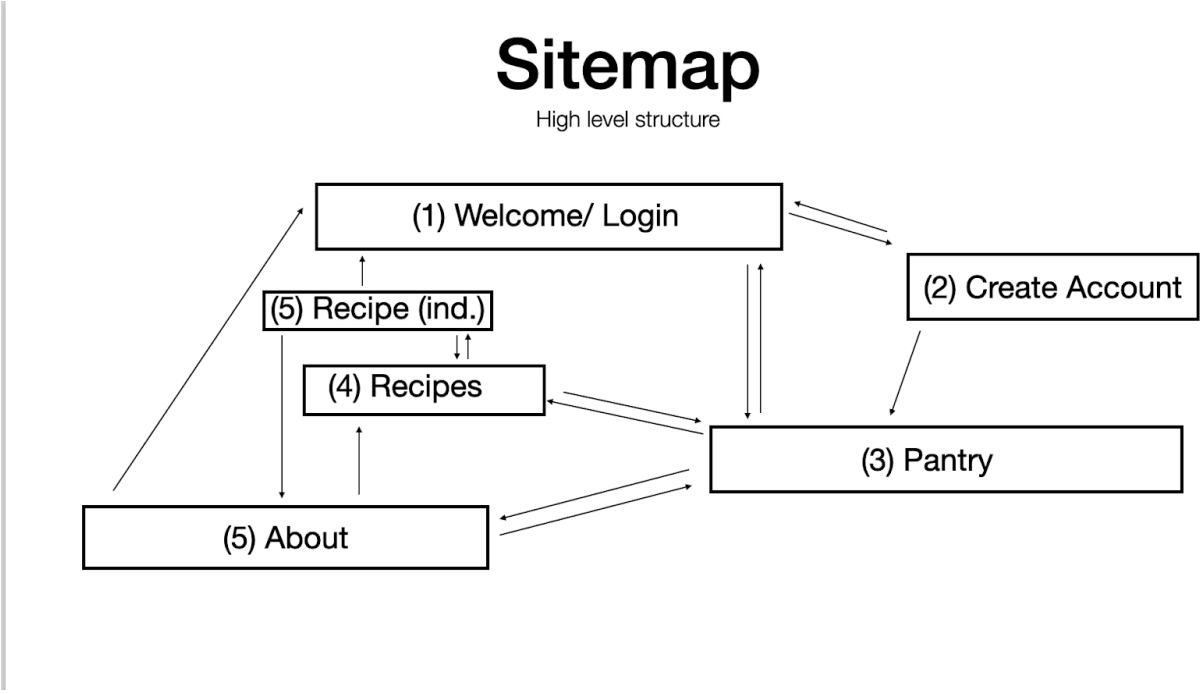
Design / Architecture

At a high level, our application will be structured as a UI connected to a backend database, which will store user data as well as the saved recipes generated by the GPT API. We plan to embrace a microservices architecture, emphasizing scalability and simplicity for ease of implementation and testing. Our Front-End UI will include a Welcome/Login page, a Create Account page, a Pantry Page, a Recipes page, a Recipe(individual) page), and an About Page.

Our microservices include a Recipe Generation Service, which takes the user's pantry table as an input and prompts the GPT API to return a recipe based on those ingredients, a User Authentication Microservice, which handles user authentication and secure session management via token-based authentication, and a Pantry Management Microservice, which handles CRUD operations and retrieval of pantry data from the database.

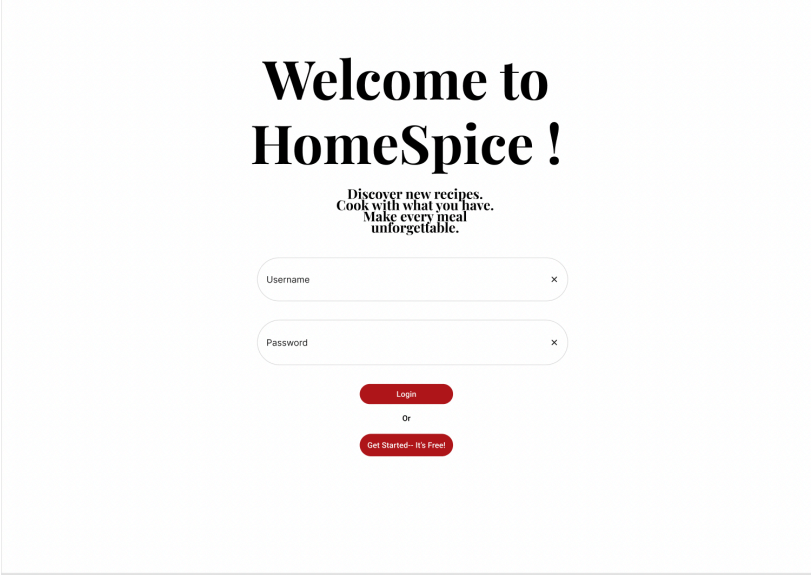
- **Front End UI (React):** Handles user interactions, displays recipes, and facilitates intuitive navigation across pages including Welcome/Login, Create Account, Pantry, Recipes, Individual Recipe, and About.
- **User Authentication Microservice**: Manages secure user authentication, session handling, and profile security using token-based authentication (JWT).
- **Pantry Management Microservice**: Handles CRUD operations for user pantry data, storing and retrieving ingredients effectively from the persistent database.
- **Recipe Generation Microservice:** Constructs ingredient-based prompts and interacts with OpenAI's GPT API to dynamically generate recipes tailored specifically to the user's pantry contents.

Sitemap:

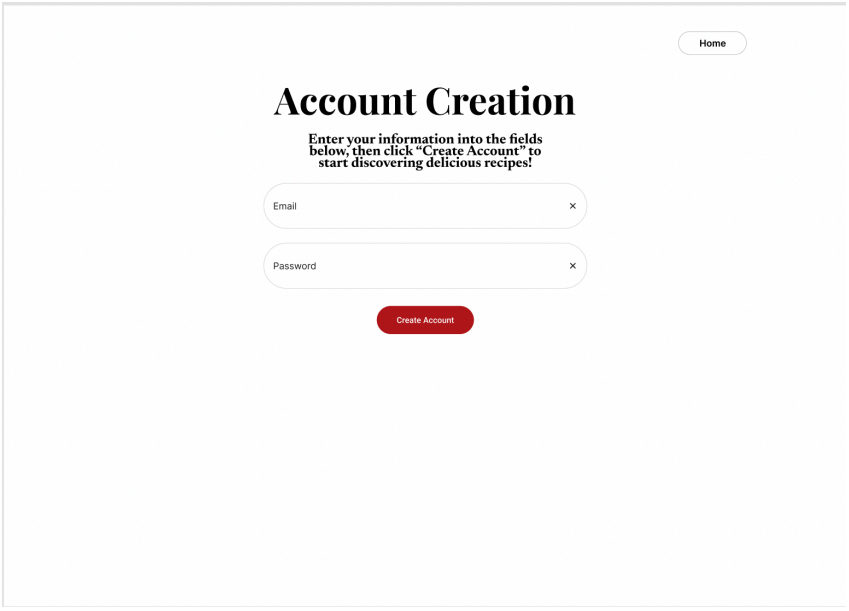


Page Mockups:

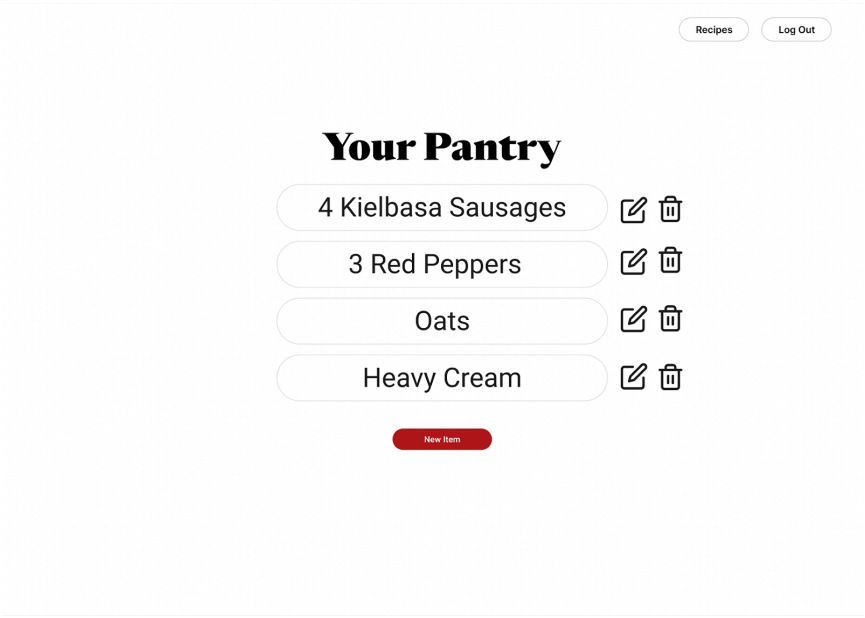
Landing/Login Page:



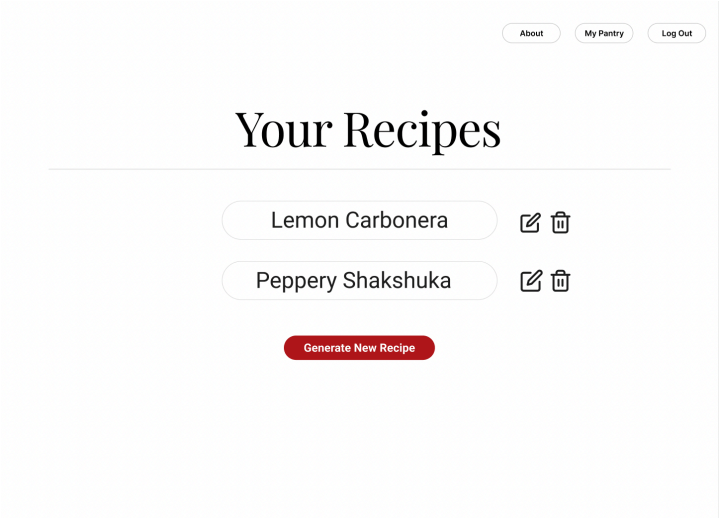
Create Account Page:



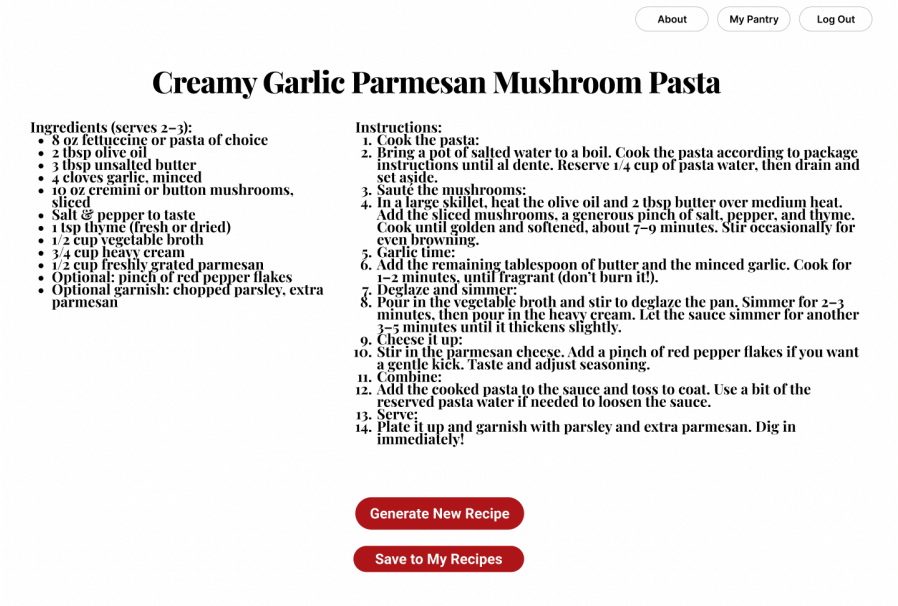
Pantry Page:



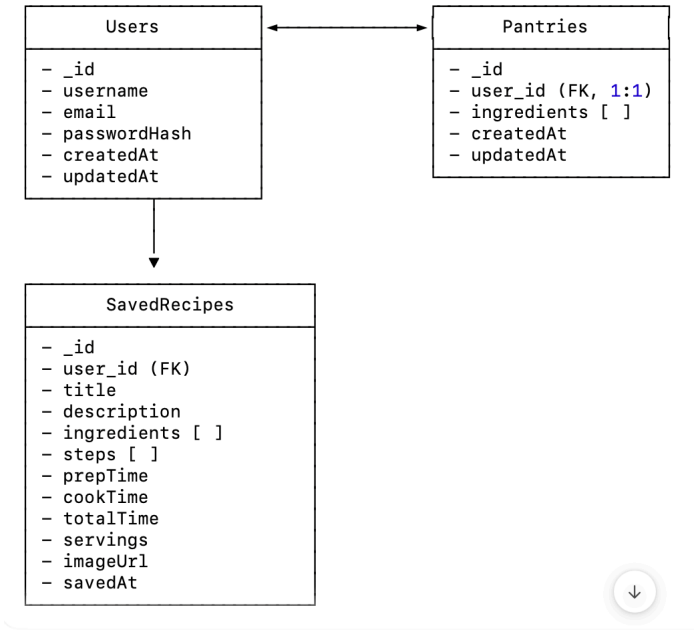
Recipes Page



Recipe (Individual) Page:



Entity Relationship Diagram:



Technologies

Development Area	Technology
Frontend	React
Backend	-FastAPI, Async HTTP Client Library (httpx)
Image Recognition (stretch goal)	-Cloud-based API (TBD): tentatively considering either AWS Rekognition, Google Cloud Vision, or Azure Computer Vision.
Database	-MongoDB: schema-less approach that can integrate with Flask through PyMongo library.
Hosting & Deployment	-Uvicorn (or Hypercorn): ASGI servers that natively support FastAPI's async operations [1] -Docker: containerizing the application for streamlined deployment -Cloud Platform (TBD): tentatively considering deploying on Google Cloud (per Tips and Tools recommendation on Canvas). May also consider OSU servers
Additional Considerations	Security - FastAPI's security utilities, such as OAuth2 with JSON Web Tokens (via packages like fastapi-users or fastapi-jwt-auth), to secure user accounts and data [1]

Discussion

A significant decision point for our team was whether to develop the backend with Flask or FastAPI. In initial discussions we were leaning towards Flask as most of us have experience with the framework. However, upon further review it was found that our choice for the backend framework would impact several other aspects of development and application performance.

Flask has a mature ecosystem, with many third-party extensions and fairly straightforward framework for developing projects that don't demand high concurrency. There are lots of tutorials and community support that would benefit a team such as ours in a learning environment. However there are some technical drawbacks for a project such as ours. Flask uses a synchronous model by default. If we want to develop our app for scalability, where it could handle several users and concurrent calls to APIs without significantly hindering performance, the framework would need to include additional components like Celery to handle asynchronous tasks and other libraries for data validation and auto-generation of API documentation. Our other option is the **FastAPI** framework. This framework has a few technical advantages over Flask for our purposes. FastAPI was built for asynchronous operations, which would benefit an application making many concurrent API calls, with lower latency.. It contains built-in data validation and auto-generates API documentation. However there are some drawbacks to consider. It's a newer framework and does not have the same extensive ecosystem of extensions or as large a community of users and documentation. There will also be a slightly steeper learning curve for our developers unfamiliar with asynchronous programming - e.g. async/await constructs.

Important Resources

- GPT openAI text/prompting document
- GPT openAI Vision/Image generation document
- Flask Application Set Up
- Flask Resource to connect backend to React Front end
- MongoDB documents on setting up
- Official Python openAI api library
- Flask database connection resource
- Image generator guide Dalle GPT
- React components Guide (if needed)
- FastAPI User Guide Tutorial

Development Plan

Alan Chan

- **Progress Report #1**
    - High-level Requirement/Story/Feature/Task - Setting up environment, Set up Git branch for story, get familiar with tech stack, setting up database (help)
        - Time estimate in hours: 10 - 30 hours
- **Progress Report #2**
    - High-level Requirement/Story/Feature/Task - As a user, I want to securely create an account and log in, so that I can access my personalized pantry and save recipes across devices while also ensuring authentication.
        - Time estimate in hours: 20 - 30 hours
- **Progress Report #3**
    - High-level Requirement/Story/Feature/Task - Work and finish landing page, login page, create accounts page for users, styling for react pages, possibly assist on stretch goal image output
        - Time estimate in hours: 20 - 30 hours
- **Project Archive - Final**
    - High-level Requirement/Story/Feature/Task - Work on final report, tie up project, take on any remaining feature/tasks
        - Time estimate in hours: 20 hours
- **Total Hours: Sum all hours for the term for THIS team member : at least 80 hours**

Tavner Murphy

- **Progress Report #1**
    - High-level Requirement/Story/Feature/Task - Setting up Environment, Git Branch, Database
        - 5 hours
    - High-level Requirement/Story/Feature/Task - Tech Stack familiarization, Documentation familiarization
        - 5 hours
- **Progress Report #2**
    - High-level Requirement/Story/Feature/Task- Set up the React front end page UI for each page
        - 3 hours
    - High-level Requirement/Story/Feature/Task- “As a user, I want to view a digital list of my current pantry inventory, so that I can add, edit, or remove items manually as needed.”
        - 5 hours
    - High-level Requirement/Story/Feature/Task- Stretch Goals
        - 2 Hours
- **Progress Report #3**

    - High-level Requirement/Story/Feature/Task - As a user, I want to view a digital list of my current pantry inventory, so that I can add, edit, or remove items manually as needed.
        - 5 Hours
    - “As a user, I want to receive recipe recommendations based on my pantry list at the click of a button, so that I can cook a meal without needing to buy more groceries.”
        - 5 hours

- **Project Archive - Final**
    - High-level Requirement/Story/Feature/Task - ” As a user, I want to receive recipe recommendations based on my pantry list at the click of a button, so that I can cook a meal without needing to buy more groceries.”
        - 5
    - High-level Requirement/Story/Feature/Task - “As a user, I want to save the generated recipes I like, so that I can easily revisit them later when I want to cook them again.”
        - 5
- **Total Hours: At least 80 Hours**

Masseeh Safi

- **Progress Report #1**
    - Setting up Environment, Git Branch, Project Structure, Tech Familiarization, Fleshing Out User Stories/Features
        - 15 hours
    - Start Database Set-up in MongoDB
        - 5 hours
- **Progress Report #2**
    - Complete database Schema Set-up in MongoDB
        - 10 hours
    - As a user, I want to view a digital list of my current pantry inventory, so that I can add, edit, or remove items manually as needed.
        - 10 Hours
- **Progress Report #3**
    - As a user, I want to view a digital list of my current pantry inventory, so that I can add, edit, or remove items manually as needed.
        - 10 Hours

    - As a user, I want to receive recipe recommendations based on my pantry list at the click of a button, so that I can cook a meal without needing to buy more groceries.
        - 10 hours
    - (Stretch Goal) As a user, I want to see an AI-generated image of the AI-generated recipes, so that I can better visualize what the dish will look like before I make it.
        - Consider additional development time needed to incorporate - 3-5 hours
- **Project Archive - Final**
    - As a user, I want to receive recipe recommendations based on my pantry list at the click of a button, so that I can cook a meal without needing to buy more groceries.
        - 5 hours
    - As a user, I want to save the generated recipes I like, so that I can easily revisit them later when I want to cook them again.
        - 5 hours
    - Final Report
        - 10 hours
- **Total Hours: at least 80 hours**

Conclusion

In the modern world we all have busy schedules and our pantries end up piling up with leftover ingredients. HomeSpice empowers users to make use of those orphaned pantry items and generate delicious recipes on the fly, reduce food waste, save money, and optimize their meal prep experience. Our solution to this everyday problem will have our users looking like the cat in the meme above, satisfied with their meal, and ready to take on anything the world throws at them.

References

[1] OpenAI. (2023). *ChatGPT (Mar. 14 version) [ChatGPT output]*. Retrieved April 10, 2025, from https://chat.openai.com/