

Final Project Proposal Report

1. Student info

Fullname	Student ID
Hoàng Tiến Huy	22120134
Phan Thanh Tiến	22120368
Nguyễn Bùi Vương Tiến	22120370

2. Proposal Overview

- **Project Name:** PantryChef AI
- **Purpose & Problem:** This project addresses the common real-world problem of food waste and "kitchen paralysis." Many people have a collection of random ingredients in their pantry and fridge but lack the inspiration or knowledge to combine them into a meal. This leads to decision fatigue, ordering takeout, and ultimately, throwing away expired food.
- **Main Goal:** The main goal of PantryChef AI is to empower users to instantly generate creative and practical recipes using only the ingredients they already have. By simply listing their available items, users will receive 3-5 unique recipe ideas, complete with step-by-step instructions. This will help users reduce food waste, save money, and discover new meal possibilities from their own kitchen.

3. Competitor Review

1. SuperCook

- **Features:** Users check off ingredients they have from a massive, pre-defined list. The site then filters a database of existing recipes to show what they can *almost* make.
- **Strengths:** Very large database of ingredients and recipes. Good filtering.
- **Weaknesses:** The user interface is tedious (checking off hundreds of items). It's a database filter, not an AI. It often shows recipes where you are *missing* 1-2 key items, which defeats the purpose.

2. Tesco Real Food

- **Features:** Offers thousands of recipes with step-by-step guides, filters by ingredient or dietary need, and links directly to Tesco's online grocery store.
- **Strengths:** Trusted brand, large and well-organized recipe database, smooth shopping list integration, and professional-quality content.
- **Weaknesses:** Similar to SuperCook, it's a search/filter, not a generator. The "match" is often imperfect and still requires a trip to the store.

3. MyFridgeFood.com

- **Features:** A basic search function where users can include some specific ingredients.
- **Strengths:** Massive, user-rated recipe database.
- **Weaknesses:** Very low-tech. It's a simple boolean search that requires the user to do all the work. It won't create a recipe; it just finds existing ones that happen to contain your search terms. Also the list of ingredients is limited.

Comparison Table of 3 web applications

Criteria	SuperCook	Tesco Real Food	MyFridgeFood.com
Core Concept	Ingredient checklist that filters recipes from a static database.	Recipe database linked to Tesco's online grocery system.	Simple ingredient search for existing recipes.
Database	Very large, pre-defined list of ingredients and recipes.	Large, curated recipe database with professional content.	Large user-submitted recipe base.
User Experience (UX)	Tedious — users must check hundreds of boxes.	Smooth and polished, but still search-based.	Simple but outdated interface.
Output Quality	Often shows recipes missing 1–2 key ingredients.	High-quality recipes but may need extra ingredients.	Basic results that may not match pantry contents well.
Flexibility & Constraints	Limited filtering options.	Filter by diet or ingredient.	Very limited.
Integration	None.	Strong Tesco shopping integration.	None.
Target Audience	Home cooks with patience to manage long ingredient lists.	Tesco customers and home cooks seeking polished recipes.	Casual users looking for quick ideas.

PantryChef AI overcomes the weaknesses of them by integrating advanced AI capabilities to dynamically generate personalized recipes that perfectly match the user's available ingredients, offering a faster, smarter, and more intuitive cooking experience than traditional database filters.

4. Core Features

- **Dynamic Ingredient Input:** A simple UI component where users can type and add ingredients to a list (e.g., as tags). Users can easily add or remove ingredients before generating.
- **Recipe Customization Options:** A set of simple inputs (e.g., dropdowns or checkboxes) for the user to refine their request. Options will include:
 - Dietary Needs: (e.g., "Vegetarian," "Gluten-Free," "Dairy-Free")
 - Cuisine Type: (e.g., "Any," "Italian," "Mexican")
 - Cooking Time: (e.g., "Under 30 mins")
- **Backend API Integration:** A secure, server-side function (e.g., a Node.js serverless function) that:
 - Receives the ingredient list and customizations from the frontend.
 - Formats this data into a specific, high-quality "prompt."
 - Securely sends the prompt to the Generative AI API (e.g., Gemini or OpenAI).
 - Receives the text-based recipe response from the AI.
- **Recipe Storage & Personalization:**
 - "Save Recipe" Functionality: A simple button on the generated recipe that allows the user to save it to a personal account/local storage.
 - User Recipe History: A dedicated section or tab where users can view all their past generated and saved recipes.
 - User Rating/Notes: Allow users to add a simple 1-5 star rating and personal cooking notes to saved recipes (e.g., "Use less chili powder next time," or "Perfect with a side of rice").
- **AI Nutrition Analyzer:** Automatically estimates nutritional information (calories, macros, vitamins) for each generated recipe:
- **Meal Planning & Scheduling:** AI suggests multi-day meal plans using available ingredients.
- **Mood-Based Meal Suggestions:** Users can input or select their current mood ("lazy," "energetic," "comfort food," etc.).

5. Technical Challenges and Learning Goals

1. Technical Challenges:

- **Prompt Engineering:** The single biggest challenge. The prompt sent to the AI must be meticulously crafted. It needs to be strict enough to *only* use the provided ingredients but flexible enough to be creative.
 - **Plan:** We will iterate on a prompt template that explicitly forbids the AI from "hallucinating" new ingredients, while encouraging it to use common pantry staples (oil, salt, pepper) if needed.
- **Parsing AI Response:** The AI will return a single block of text. I need to reliably parse this text to separate the recipe's title, ingredients, and instructions for clean UI formatting.

- **Plan:** I will either instruct the AI in the prompt to return the response in a specific format (like Markdown or JSON) or use regular expressions (regex) on the frontend to split the string by keywords like "Instructions:".
- **Secure API Key Management:** The AI API key is sensitive and cannot be exposed on the frontend.
 - **Plan:** I will create a simple backend API endpoint (e.g., using Vercel Serverless Functions) that the frontend calls. This endpoint will securely store the API key as an environment variable and be the only part of the application that communicates with the AI service.

2. Learning Goals:

- **Third-Party AI API Integration:** To gain hands-on experience integrating a modern AI service. This includes managing API keys, handling asynchronous data (fetch, async/await), and managing loading and error states for a (potentially) slow API response.
- **Complex State Management:** To manage a multi-part user state effectively. This includes the list of ingredients, the selected customization options, the loading status, and the final recipe data. This will be a great opportunity to practice using React hooks like `useState` and `useReducer` for a non-trivial application.
- **Learn how to identify and implement techniques to optimize the performance of a React application,** including reducing unnecessary re-renders, improving component efficiency, and applying best practices for state management and code splitting.