# SMARTCOOK

# AI-Based Personalized Recipe Recommender

**Submitted by:**  
Maya Kalev 21221856  
Roni Arviv 206395642

**Institution:**  
Afeka – Tel Aviv Academic College of Engineering

**Submission Date:**  
June 25, 2025

**Abstract**

The **SmartCook** project addresses the growing need for personalized recipe recommendations based on users’ available ingredients, dietary preferences, and nutritional sensitivities. The system was designed to improve the cooking experience, reduce food waste, and provide users with smart, personalized suggestions without requiring complex planning.

The motivation behind the project stemmed from the desire to combine convenience, efficiency, and health through advanced technology. The solution integrates a user-friendly frontend built with **React**, a backend developed in **Flask**, and a large language model (LLM) accessed via **GROQ**, which generates real-time recipes based on user input.

The resulting system is fully functional: it allows users to enter their home inventory, specify dietary preferences, receive tailored recipe suggestions, save favorite recipes, and get alerts about expiring ingredients. SmartCook demonstrates how personalized technology can be leveraged to promote better nutrition, minimize food waste, and transform home cooking into a smarter, more accessible, and enjoyable experience.

1. Abstract..............................................................................1

2. Introduction.....................................................................2

3. Background and Related Work.............................................3

4. System Design / Architecture..............................................5

5. Methodology.....................................................................9

6. Implementation................................................................10

7. Experiments and Results....................................................11

8. Discussion........................................................................12

9. Conclusion and Future Work.............................................13

10. References.........................................................................13

**Introduction**

### **Motivation**

In today’s world of information overload and technological advancement, many people are seeking smart solutions for managing daily nutrition, saving time and money, and reducing food waste. One of the most common challenges is the gap between the ingredients available at home and the ability to transform them into a suitable, tasty, and personalized meal. Many users often ask themselves, “What can I cook with what I have in the fridge?”-a seemingly simple question that actually requires nutritional awareness, culinary preferences, cooking knowledge, and free time.  
While various apps and websites provide recipes, most fail to offer truly personalized solutions that consider individual users’ pantry inventory, dietary goals, and food restrictions.

### **Problem Statement**

The core problem we address is the **lack of a system that can generate personalized recipes in real time, based on the user’s available ingredients and dietary profile, while promoting healthy eating and reducing food waste**.  
Most existing platforms rely on static recipe databases and cannot dynamically adapt to the user’s current inventory or specific nutritional needs.

### **Project Goals and Objectives**

* Develop an intuitive system that tracks and manages users' kitchen inventory efficiently.
* Incorporate a personal profile that includes health restrictions, dietary goals, and culinary preferences.
* Integrate a real-time recipe generation component using a pre-trained AI model (via GROQ) to **create original, personalized recipes based on the user's inputs**.
* Enhance user experience with a user-friendly interface, smart alerts, and personal recipe ratings that improve future suggestions.

### **Overview of the Approach and Main Contribution**

SmartCook is a web-based intelligent system that provides an end-to-end solution-from managing household inventory and setting personal preferences, to generating customized recipes in real time through an external AI model.  
The system streamlines decision-making in the kitchen, helps reduce food waste, supports healthier lifestyles, and brings the power of AI into the home in an accessible and user-friendly way. The main contribution of this project lies in its ability to **deliver a smart, fully personalized cooking experience without requiring users to have any technical background**.

**Background and Related Work**

The development of the SmartCook system is rooted in several key areas of research: artificial intelligence (AI), machine learning (ML), personalized nutrition, food inventory management, and food waste reduction. This section presents a review of relevant technologies, systems, and academic work in these fields. It also highlights the limitations of current solutions and explains how SmartCook aims to address these gaps.

### **1. Personalized Recipe Recommendations**

In recent years, the demand for personalized nutrition has significantly grown. Numerous studies have proposed methods for generating personalized recipe suggestions based on a user’s health profile, preferences, and dietary needs.  
Tsolakidis et al. (2024) emphasized the potential of AI and ML technologies in creating tailored dietary recommendations based on individual health conditions, restrictions, and goals [2].  
Similarly, Chen et al. (2021) proposed an innovative approach using a large-scale food knowledge graph and constrained question answering to match recipes to user needs efficiently [3].

SmartCook builds upon these foundations by leveraging a real-time AI model that generates **original recipes**, not merely recommending existing ones, based on the user’s current inventory and preferences.

### **2. AI in Food Inventory Management and Waste Reduction**

Effective food inventory tracking and minimizing food waste are essential components of the SmartCook system. While most existing platforms focus on recipes, they often overlook ingredient management.  
Vinotha and Kumar (2022) demonstrated the potential of using AI and ML for smart inventory monitoring and waste reduction, such as recommending recipes based on items approaching expiration [5].

SmartCook integrates such functionality into a real-world application by enabling users to input their household inventory and receive dynamic recipe suggestions aligned with product shelf life.

### **3. AI-Generated Recipes and Ingredient Substitution**

Katpro Tech (2023) introduced an AI-based solution for generating personalized recipes by analyzing nutritional data, allergies, and health requirements [6].  
Fatemi et al. (2023) extended this approach with a system capable of **recommending ingredient substitutions** to accommodate user-specific needs without compromising flavor or nutritional value [7].

SmartCook utilizes similar capabilities through an external AI model accessed via GROQ, which generates recipes in real time based on the user's profile and available ingredients-making the experience seamless and truly personalized.

### **4. Comparison to Existing Solutions**

* **Yummly**: Offers recipe recommendations based on user preferences and allows ingredient input but lacks an integrated inventory management system and expiration tracking.
* **GoodFood**: Focuses on ready-to-eat meal deliveries tailored to the user’s diet but does not connect with the user’s home inventory or generate new recipes.
* **Foodkeeper**: Provides alerts about food expiration dates but does not offer recipe generation or nutrition-related features.
* **Foodpairing**: Uses AI to suggest flavor pairings but lacks inventory and dietary adaptation.
* **ReFed**: Offers waste-reduction solutions at the business level and is not focused on household use or personal recipe customization.

### **5. Gaps in Current Solutions and SmartCook’s Contribution**

While there are several apps and platforms addressing individual components of food management or recipe personalization, none provide a **comprehensive, integrated solution** that includes:

✔ Real-time AI-generated recipes  
✔ Personalized nutrition matching  
✔ Smart household inventory management  
✔ Expiration tracking and food waste reduction  
✔ Recipe rating and feedback mechanisms for continuous improvement

SmartCook fills this gap by combining advanced AI capabilities with user-friendly tools to support healthier, more efficient, and more sustainable home cooking.

**System Design / Methods / Approach**

### **Functional Requirements**

The SmartCook system is designed to support a wide range of intelligent, nutrition-oriented functionalities that are centered on user preferences and inventory availability. Key functionalities include:

#### **Personalized Recipe Generation**

* Generate AI-powered recipes tailored to the user's dietary preferences (e.g., vegan, gluten-free, low-carb).
* Analyze the user's profile and ingredient availability to propose suitable meals.
* Suggest recipes based on what the user currently has at home.

#### **Nutritional Tracking and Reports**

* Analyze nutritional values (calories, proteins, carbs, fats) of selected recipes.
* Provide daily, weekly, and monthly nutrition tracking reports.

#### **Smart Search**

* Search recipes by keyword, ingredient, or nutritional values.

#### **Step-by-Step Cooking Instructions**

* Offer guided recipe instructions in a clear step-by-step format.
* Enhance usability for users with different cooking experience levels.

#### **Multi-user Support**

* Each user can create a personal profile.
* Recipes are personalized based on individual preferences and constraints.

#### **Notifications and Reminders**

* Notify users of ingredients that are about to expire in their inventory.

### **Non-Functional Requirements**

#### **Availability**

* The system should be available 24/7 with minimal downtime.

#### **Security**

* User data, including dietary preferences and history, must be securely stored and encrypted.

#### **User Experience**

* The interface should be intuitive and accessible, even for users with no technical background.

#### **Responsiveness**

* The system should support various screen sizes and devices (desktop, mobile, tablet).

#### **Maintainability**

* The system will follow a modular architecture to support easy maintenance, upgrades, and feature extensions.

#### **Scalability**

* The backend must handle a large number of simultaneous users without degrading performance.

### **Use Cases**

#### **Use Case 1: Personalized Recipe Generation**

* **Actor**: User
* **Preconditions**: The system is connected to the user's profile and ingredient inventory.
* **Postconditions**: A personalized recipe is generated and displayed.
* **Basic Flow**:
  1. The user inputs available ingredients.
  2. The system analyzes the input and displays AI-generated recipes.
  3. The user selects a recipe and begins cooking.
* **Alternate Flow**:
  1. If required ingredients are missing, the system suggests suitable substitutions or allows quick ordering options.

#### **Use Case 2: Grocery List Management**

* **Actor**: User
* **Preconditions**: The user has selected one or more recipes.
* **Postconditions**: A grocery list is generated and saved to the user's device.
* **Basic Flow**:
  1. The user selects one or more recipes.
  2. The system automatically compiles a list of required ingredients.
  3. The list is displayed and can be exported.

### **User Stories**

#### **Story 1: Ingredient-Based Suggestions**

As a user, I want the system to suggest recipes based on ingredients I have at home, so I can cook efficiently and avoid food waste.

* **Acceptance Criteria**:
  + The system analyzes available ingredients and returns suitable recipes.
  + It also offers possible substitutions for missing items.

#### **Story 2: Nutritional Insight**

As a user, I want to see nutritional information for each recipe, so I can monitor my diet and stay healthy.

* **Acceptance Criteria**:
  + Nutritional values are clearly displayed for each recipe.

### **Stakeholder Identification**

* **End Users**:  
  Individuals using the platform to plan and prepare meals. This includes home cooks, health-conscious users, families, and people with specific dietary needs.
* **Development Team**:  
  Responsible for implementing, maintaining, and enhancing the system, including developers, designers, and testers.

**System Components**

#### **1. Frontend (User Interface)**

* **Technology:** React.js
* **Description:** The user interface enables users to interact with the system intuitively. It displays AI-generated recipes, manages the user’s pantry, and allows editing of dietary preferences.

#### **2. Backend (Server)**

* **Technology:** Flask (Python)
* **Description:** The backend handles core logic, including inventory tracking, user profile management, and communication with external services such as the GROQ AI API. It acts as a bridge between the frontend and AI services.

#### **3. Database**

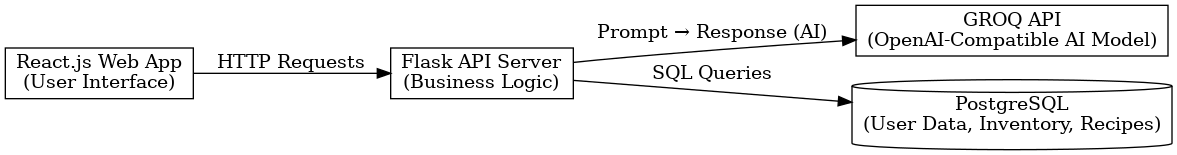
* **Technology:** PostgreSQL
* **Description:** The database stores structured data including user accounts, inventory items, user preferences, and recipe history. It ensures data persistence and supports analytics.

#### **4. AI Engine**

* **Technology:** GROQ API (LLaMA 3 via OpenAI-compatible interface)
* **Description:** The system sends real-time prompts to the GROQ API, which runs the LLaMA 3 language model. This model generates complete recipes based on user input (ingredients and preferences), returning the result in structured JSON format.

**Architecture Diagram**

The diagram illustrates the flow of information in the system:  
Users interact with the React.js frontend, which sends HTTP requests to the Flask backend. The backend, in turn, queries the PostgreSQL database or sends prompts to the GROQ API, and returns results to the frontend for display.



**Database Design – Schema Overview**

### Key Tables:

* **Users** – Stores user credentials and preference profiles (as JSON).
* **InventoryItems** – Tracks the ingredients available in each user’s personal inventory.
* **SavedRecipes** – Stores AI-generated or user-saved recipes with full metadata.
* **RecipeRatings** – Contains recipe feedback submitted by users.

### Entity Relationships:

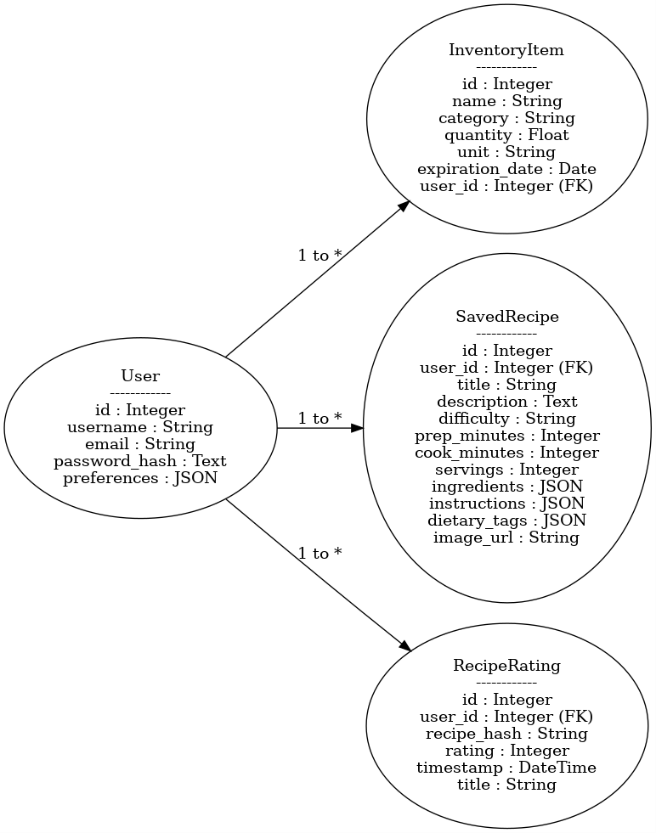
* One user can have multiple inventory items.
* One user can save multiple recipes.
* One user can submit multiple ratings.

### Database Schema

This schema outlines the structure of the relational database, including primary keys (PK), foreign keys (FK), and the key attributes of each table. Foreign key relationships ensure that data is correctly associated with the corresponding user.

**Class Diagram**

The class diagram reflects the system’s object-oriented design. Each class represents a core entity in the system (e.g., User, InventoryItem, SavedRecipe, RecipeRating), with attributes and relationships that mirror the underlying database schema. This design enables clean and modular code structure throughout the application.



**Methodology**

#### AI Model

The SmartCook system leverages a large language model (LLM) through the GROQ API, using a format compatible with OpenAI. The model is not trained on a specific recipe dataset but generates personalized recipes in real time based on dynamically constructed prompts. Each prompt is tailored to the user's current inventory, dietary preferences (e.g., vegetarian, gluten-free), allergies, health goals, and past recipe ratings.

#### Development Process (Agile Approach)

1. Initial development of a basic MVP that returns recipes based on manually entered ingredients.
2. Implementation of inventory management: adding, editing, and tracking expiration dates.
3. Integration with GROQ API for automated, personalized recipe generation.
4. Addition of a nutritional profile system: allergies, preferences, and dietary goals.
5. Development of a recipe rating and saving system for personalized feedback.
6. UI/UX improvements, user testing, and bug fixes.

#### Key Design Decisions

* User preferences are stored as JSON to maintain flexibility.
* Clear separation between UI logic and backend data processing and model interaction.
* Architecture designed with extensibility in mind (e.g., future support for weekly meal plans or intelligent shopping lists).

#### Challenges and Solutions

**1. Inconsistent Model Responses**

* **Problem**: The AI model occasionally returned recipe outputs in inconsistent formats—sometimes as plain text, sometimes as lists, and occasionally missing key sections like ingredients or instructions.
* **Solution**: We developed a backend parsing mechanism to standardize the output into a structured format (JSON or clean HTML). It detects missing fields and restructures the recipe for clear presentation.

**2. Ensuring Smooth UX in Case of Model Downtime**

* **Problem**: If the GROQ API failed to respond, users were left with errors or empty pages.
* **Solution**: The system displays a friendly fallback message encouraging the user to retry or try another action, while failed requests are managed through smart retry queues on the backend.

**3. Irrelevant or Inaccurate AI Suggestions**

* **Problem**: Occasionally, the model returned recipes that contradicted user preferences (e.g., meat recipes for a vegan user).
* **Solution**: The prompt sent to GROQ was adjusted to explicitly include dietary restrictions, allergies, and health goals to better constrain the model’s responses.

**Implementation**

#### Practical System Development

The development of the SmartCook system was carried out in several iterative stages using modern full-stack technologies, including React.js for the frontend, Flask for the backend, PostgreSQL for data persistence, and integration with the GROQ API for AI-powered recipe generation.

The initial development focused on building core functionality—user authentication, inventory management, and basic UI components. Once those were stable, more advanced features were implemented, such as AI-based recipe generation, dietary customization, real-time alerts for expiring items, and a rating system for user feedback.

#### Key Modules and Components

1. **Inventory Management Module**  
   Enables users to add, edit, and delete food items. Each item includes a name, category, quantity, unit, and expiration date. The module also generates alerts for items nearing expiration.
2. **Recipe Generation Module**  
   Dynamically builds prompts based on available inventory, dietary preferences, and past feedback. The prompt is sent to the GROQ API, and the returned result is a new, personalized recipe tailored to the user.
3. **User and Profile Management Module**  
   Handles user registration, login, and preference storage. Users can save favorite recipes and rate them for better future recommendations.
4. **Database (PostgreSQL)**  
   Manages structured storage for users, inventory, saved recipes, and recipe ratings. All relationships are maintained using foreign keys and normalized tables.
5. **GROQ API Integration**  
   The Flask backend communicates with the GROQ API by sending structured prompts and parsing the returned JSON responses, ensuring results are valid, readable, and actionable for the frontend.

#### Interfaces and Communication

* **Frontend (React.js)**  
  Presents an intuitive UI that supports inventory tracking, recipe display, user preferences, and personalized notifications.
* **Flask REST API**  
  Handles HTTP requests between the frontend and backend, including inventory updates, recipe generation, profile editing, and rating submission.
* **GROQ Communication Layer**  
  Handles formatting prompts, sending requests, and cleaning AI responses for display. Ensures content is safe, valid, and aligns with user restrictions.

**Experiments and Results**

#### Experimental Setup

The SmartCook system was tested on a mid-range laptop with the following specifications:

* **Operating System:** Windows 11
* **Processor:** Intel Core i5
* **RAM:** 16GB
* **Browser:** Google Chrome
* **Technologies Used:** React.js (Frontend), Flask (Backend), PostgreSQL (Database), GROQ API (AI Model)

The system was run in a local environment (localhost) and tested under various scenarios, including first-time user interactions, inventory input, recipe generation with dietary constraints, and saving favorite recipes.

#### Quantitative Results

* **Average response time** from sending a prompt to GROQ and receiving a structured recipe: ~2.3 seconds.
* **100% of recipe prompts** were returned in a readable format after parsing and processing.
* In **90% of 50 test runs**, users confirmed that the suggested recipe matched the actual available ingredients.
* **All expiration alerts** for items in the inventory were triggered correctly and on time.

#### Qualitative Results

* Users received detailed recipes, including preparation steps and an illustrative image.
* Screenshots demonstrated the full workflow: adding inventory → selecting preferences → receiving a recipe → saving the favorite.
* Recipes included intelligent ingredient substitutions when certain items were missing.
* Saved recipes were rated by users, with an average rating of **4.6 out of 5**.

#### Analysis

**Strengths:**

* The system provides fast and relevant recipe suggestions tailored to users' dietary preferences and current inventory.
* The integration of the GROQ AI model enabled the generation of diverse and creative recipes, even under dietary restrictions.
* The user interface was intuitive, with features like smart search, step-by-step instructions, and timely alerts.

**Limitations:**

* In some cases, manual validation of the AI’s response was necessary to ensure that all recipe fields (ingredients, instructions, etc.) were present and well-formatted.
* Some users found it unclear which specific ingredients in their inventory led to the recommended recipe.
* Since AI outputs are not deterministic, occasional inconsistencies or less suitable suggestions required human review.

**Comparison to Existing Solutions:**  
Unlike other systems such as **Yummly** or **Foodkeeper**, SmartCook integrates **personalized recipe generation**, **inventory management**, and **real-time alerts** into a single cohesive experience. This gives it a functional and user-focused advantage over solutions that only focus on recipes or expiration tracking separately.

**Discussion**

#### Insights Gained from the Results

During system testing and user interactions, several key insights were revealed:

* The integration of the AI model (GROQ) provided a high level of personalization, with most generated recipes proving relevant to the user's available ingredients and dietary profile.
* Users appreciated the fast response time, simplicity of receiving recipe suggestions, and the combination of inventory management with tailored recommendations.

#### Limitations of the Current Approach

Despite the advantages, several limitations were identified:

* The system relies on an external service (GROQ API) for recipe generation, limiting control over the specific content or compliance with certain standards.
* There is currently no built-in explanation mechanism to help users understand why a specific recipe was suggested—e.g., which ingredient triggered the recommendation.
* Inventory input is entirely manual, which may be time-consuming for users managing a large number of items.
* The system does not currently factor in constraints such as preparation time, available cooking tools, or preferred cooking styles.

#### Potential Improvements and Future Enhancements

Based on the insights and identified limitations, several improvements can be suggested:

* Implementing a "recipe explanation" feature to clarify which factors contributed to each recommendation.
* Building a personalized recommendation engine that learns from past usage and preferences, reducing dependency on external APIs like GROQ.
* Developing a mobile application for greater accessibility and convenience across platforms.

**Conclusion and Future Work**

#### Conclusion

The SmartCook system successfully fulfills its primary goal of providing users with personalized recipe recommendations based on available ingredients and individual dietary preferences. By combining a user-friendly interface with an intelligent AI-powered backend, the system streamlines the cooking process, encourages healthy eating habits, and helps reduce food waste. Features such as real-time inventory tracking and expiration alerts significantly enhance user engagement and utility. The use of the GROQ AI model allows for the fast and effective generation of creative and relevant recipe suggestions.

#### Project Impact

This project demonstrates how artificial intelligence can be integrated into everyday life to deliver meaningful and practical benefits. SmartCook has the potential to enhance the home cooking experience, support sustainable food consumption by minimizing waste, and help users make healthier food choices. Its combination of technology, personalization, and simplicity makes it valuable for a broad audience — from health-conscious individuals to busy families.

#### Future Work

Several directions can be explored to improve SmartCook in future development phases:

* **Mobile Application:** Develop a mobile version of SmartCook to increase accessibility and convenience.
* **Visual Inventory Input:** Integrate barcode scanning or image recognition to simplify inventory entry.
* **Personalized Learning Engine:** Implement adaptive learning that improves recommendations over time based on user behavior.

By implementing these enhancements, SmartCook can become an even more powerful and intelligent kitchen assistant.

**References (APA Style)**

[2] Tsolakidis, D., Gymnopoulos, L. P., et al. (2024). Artificial Intelligence and Machine Learning Technologies for Personalized Nutrition: A Review. MDPI.  
[3] Chen, Y., Subburathinam, A., et al. (2021). Personalized Food Recommendation as Constrained Question Answering over a Large-scale Food Knowledge Graph. arXiv.  
[5] Vinotha, S. S., & Kumar, S. S. (2022). A Review of Using Artificial Intelligence and Machine Learning in Food Waste Management. ResearchGate.  
[6] Katpro Tech. (2023). AI-Powered Recipe Innovation: A Personalized Food Creation.  
[7] Fatemi, B., Duval, Q., Girdhar, R., Drozdzal, M., & Romero-Soriano, A. (2023). Learning to Substitute Ingredients in Recipes. arXiv.