AI-Powered Personalized Recipe Recommendation System

Ms. Deeksha Keshav Naik[1], Dr. Sandeep Bhat[2], Ms. Dhanya S[3], Ms. Disha Mahesh Shetty[4], Ms. Manisha Mahesh Naik[5]

[1,3,4,5]*Student, Srinivas Institute of Technology*

[2]*Professor, Srinivas Institute of Technology*

doi.org/10.64643/IJIRTV12I12-206723-459

***Abstract*—The development of an AI-powered personalized recipe recommendation system provides an intelligent and user-centric approach to simplifying everyday meal planning. This web-based platform enables users to generate customized recipes based on available ingredients, emotional mood, dietary preferences, allergies, and health goals through an interactive interface. Users can input ingredients using text or voice, while the system processes this data using artificial intelligence to generate practical, mood-aligned recipes. Secure authentication ensures personalized experiences, and user profiles maintain preferences and history for improved recommendations. The platform incorporates key modules such as ingredient processing, mood-based personalization, recipe history, favorites management, and analytics for tracking usage patterns. Implemented using React for the frontend, FastAPI for the backend, MongoDB for data storage, and Gemini AI for intelligent recipe generation, the system ensures scalability, usability, and real-time interaction. Experimental results demonstrate improved efficiency, reduced manual recipe search effort, and enhanced user satisfaction. This project contributes to smart lifestyle applications by offering a reliable, adaptive, and personalized solution for modern cooking needs.**

***Index Terms*—AI-powered recipe recommendation, personalized cooking, mood-based food suggestions, ingredient-based recipes, artificial intelligence, web-based application, dietary preferences, smart meal planning.**

I. INTRODUCTION

In recent years, changing lifestyles and increased professional commitments have significantly influenced how individuals plan and prepare their daily meals. Limited time, busy schedules, and varying dietary needs often make it difficult for users to consistently follow healthy and balanced eating habits [1]. Traditional approaches to recipe selection, such as printed cookbooks or general-purpose recipe websites, usually require extensive searching and manual filtering, while offering limited personalization [2]. These limitations frequently lead to repetitive meal choices, inefficient ingredient usage, and increased food wastage [3].

Advancements in artificial intelligence and web technologies have enabled the development of intelligent systems capable of adapting to individual user preferences and constraints [4]. Personalized recommendation systems have gained widespread adoption across multiple domains, as they can analyze user data and deliver customized content efficiently [5]. When applied to food and nutrition, such systems can significantly improve meal planning by recommending recipes that align with user needs, available ingredients, and lifestyle preferences [2].

Web-based platforms provide an effective medium for deploying intelligent recommendation systems, allowing users to access services from any internet-enabled device [7]. Through structured and interactive interfaces, users can input information such as available ingredients, dietary restrictions, and health objectives with minimal effort [8]. Maintaining this data within a centralized storage system supports consistent personalization, enables preference retention, and improves recommendation accuracy over time [9].

Ensuring data security and privacy is a critical requirement, as users often share sensitive information related to allergies, nutrition, and personal eating habits [10]. Secure authentication mechanisms and validation techniques help protect user data while ensuring system reliability and integrity [11].

Additionally, organizing recipes based on cuisine type, nutritional value, preparation complexity, and cooking time improves usability and allows users to identify suitable meals more efficiently [5].

Incorporating emotional context into recipe recommendation introduces an innovative layer of personalization [3]. Food choices are strongly influenced by mood and emotional state, and recognizing this relationship enables the system to provide suggestions that better align with user feelings and energy levels [14]. Voice-based input further enhances accessibility by allowing users to interact naturally with the platform, reducing effort during ingredient entry [8, 15].

Real-time processing and automated recipe generation improve system responsiveness and user engagement by delivering instant, relevant recommendations [12]. By analyzing historical interaction data, the system can identify behavioral patterns and continuously refine its outputs, leading to improved accuracy and user satisfaction [1]. These data-driven insights also support healthier decision-making and encourage sustainable food consumption practices [7].

AI-powered personalized recipe recommendation systems align closely with the broader goals of smart living and digital wellness initiatives [5]. The integration of modern technologies such as React for frontend development, backend frameworks, NoSQL databases, and advanced AI models enables the creation of scalable, responsive, and user-friendly applications [9, 13]. Such systems offer a practical and intelligent solution for modern meal planning by combining personalization, emotional awareness, and automation to enhance the overall cooking experience.

II. SYSTEM DESIGN AND WORKFLOW

A. Architecture Design

The AI-powered recipe recommendation system uses a multi-layered architecture connecting the frontend, backend, and database [1, 2, 5]. Users enter ingredients and moods via a web interface, with inputs sent to the backend through RESTful APIs [9, 10]. The backend validates requests and coordinates the Recommendation Engine, which filters and ranks recipes using internal and external data [2, 3, 6, 12]. The database stores user profiles, mood history, and saved recipes, enabling personalized, up-to-date recommendations [1, 5, 7, 13], ensuring an efficient and user-friendly experience.

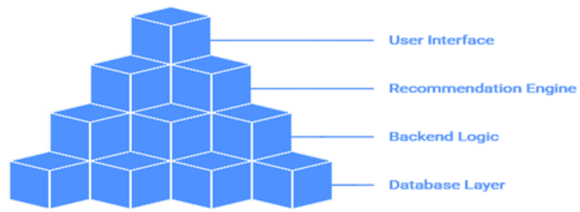

Fig.1: Architecture Design

B. Dataflow Diagram

The Data Flow Diagram (DFD) of the AI-powered personalized recipe recommendation system illustrates how user inputs—such as ingredients, mood, cuisine preference, and dietary restrictions—move through the system to generate tailored recipes [1, 2, 5]. In the Level 0 context diagram, the user interacts directly with Mood Munch, which sends input data to the AI model for processing [2, 3]. The Level 1 DFD details internal flow: ingredient, mood, and cuisine inputs are captured by their respective modules, validated, and sent request before passing it to the AI model for generating personalized recipe suggestions [6, 12, 13]

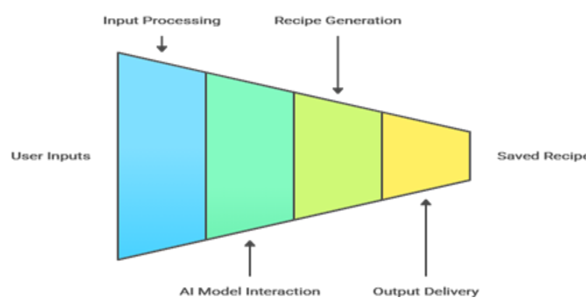

Fig.2: Database Flow and Issue Categorization

C. Database Management and Recipe Categorization

The backend stores user profiles, recipe histories, preferences, and recommendations securely using a NoSQL database [13]. Recipes are categorized by cuisine, nutritional value, preparation time, difficulty, and mood, enabling efficient filtering and retrieval [2, 3]. Each record includes ingredients, generation timestamps, and user feedback to support tracking and continuous improvement [7, 12]. Optimized indexing ensures fast access even with large volumes of user interactions and recipe data [6, 11]. The database supports scalable storage of multimodal data, including text-based inputs and AI-generated outputs, ensuring flexibility for future system expansion [13].

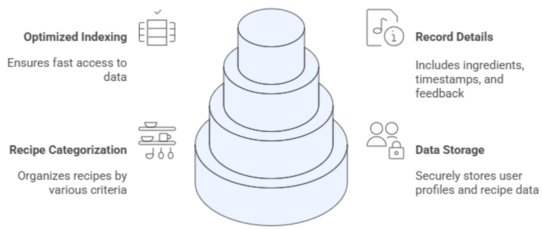

Fig 3: Database Management and Recipe Categorization

D. Personalized Recipe Generation Process

After accessing the platform, users authenticate securely through login and logout mechanisms before using system features [10]. Ingredients can be entered using text or voice input, along with mood selection and preferred cuisine, which guide the personalized recipe recommendation process [2, 3, 8]. Upon initiating recipe generation, the system processes these inputs to produce AI-generated recipes tailored to the user's preferences and emotional state [1, 4]. Users can view and save generated recipes for future use and update profile details such as dietary restrictions to maintain accurate personalization in recipe generation for each user [5, 7, 9].

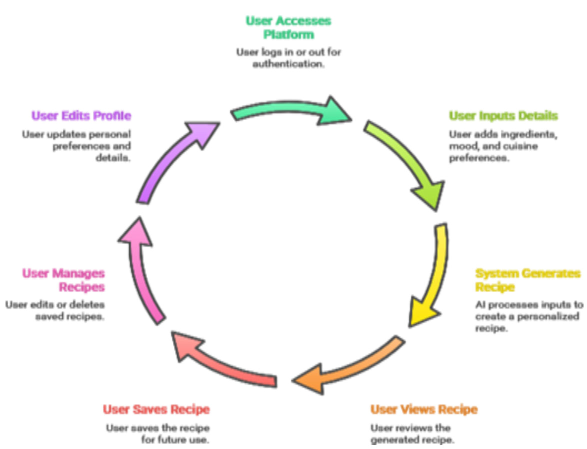

Fig 4: Personalized Recipe Generation Process

III. METHODOLOGY

This section walks through how we designed and built MoodMunch, the AI-Powered Personalized Recipe Recommendation System. We cover the key workflow steps and explain the role of each technology component in generating recipes based on mood and available ingredients. User inputs go through a clean frontend interface and are handled by a backend that coordinates the AI recipe generation. The system is divided into three main parts: the User Interface, Backend Services, and the AI Recipe Generation Module, each handling a distinct stage of the recommendation process.

A. User Registration and Authentication

When a user first opens the platform, they land on the login or sign-up page. New users fill in basic details—name, email, password, dietary preferences, any allergies, and health goals—while returning users can simply log in with their saved credentials. As part of sign-up, users can also specify food preferences like vegetarian, vegan, keto, or allergy-based restrictions so the system knows from the start what kind of recipes are suitable for them. We used JWT tokens to handle authentication, which keeps user data secure and makes sure only authorised users can access personalised features.

B. Mood and Preference Selection

Once logged in, users move into the Generate Recipe workflow and start by answering a few mood-based questions. The system asks about their energy level and what kind of meal vibe they are going for—options include Comforting, Light & Fresh, Hearty & Filling, or Quick & Simple. These answers give the AI a sense of what the user is in the mood for at that moment. Users also pick a cuisine type, such as Indian, Chinese, or Italian, which helps narrow things down further and makes sure the recipes feel personally relevant.

C. Ingredient Processing

The Ingredient Processing module takes whatever the user types or speaks and turns it into a clean, structured list that the AI can work with. Users can either type their ingredients or use voice input, which makes the process quick and easy regardless of how comfortable someone is with technology. Once the inputs come in, Gemini AI handles the cleaning—fixing typos, removing duplicate entries, and standardising how each ingredient is written. This ensures the ingredient list going into the AI prompt is accurate. By creating a precise ingredient list, the system ensures that recipes generated later are feasible and ingredient list, the system ensures that only practical and feasible recipes are generated. This cleaned data is then merged with the user's mood and cuisine preferences to form a complete dataset for recipe generation.

D. AI Recipe Generation Workflow

The AI Recipe Generation Workflow is the core of AI-Powered Personalized Recipe Recommendation System, which drives the entire recipe creation process in MoodMunch. Once the user's mood, available ingredients, dietary restrictions, health goals, and cuisine preference are collected, all of this gets packaged into a structured prompt and sent to the Gemini AI model. The model then produces a full recipe—covering step-by-step cooking instructions, ingredient quantities, estimated cooking time, difficulty level, and nutrition highlights—with the tone adjusted to suit the user's mood. Before the recipe is shown to the user, the system checks it against the user's restrictions to confirm it is safe and practical. This keeps the output quality consistent and trustworthy across different use cases.

E. Automation and Workflow

The automation layer keeps everything running without requiring manual oversight. Whenever a recipe is generated, the system automatically applies the user's dietary and allergy filters, so there is no chance of an unsuitable recipe slipping through. If the AI returns an incomplete or inconsistent output, a retry mechanism kicks in to get a better result. Everything the user does—recipes they viewed, ratings they gave, items they saved as favourites—is logged so the system can learn their preferences over time. The frontend and backend are tightly integrated so the whole pipeline, from input to validated recipe output, runs without any manual steps in between.

IV. RESULTS AND DISCUSSIONS

A. Results:

When we tested the MoodMunch platform we found that all the main features worked as we expected them to Users could type in the ingredients, they had at home choose a mood and a type of food and get a recipe in seconds. The suggestions made sense. They used the ingredients that were available and matched the mood that was chosen.

The platform did not give results. Navigation was smooth. Nothing went wrong on different screen sizes or devices. Login, saving data and checking inputs all worked consistently throughout the testing period. The platform did what it was made for. It took the users situation and turned it into a recipe. What stood out was that the suggestions did not seem to come from a fixed list. Because the system considered the ingredients, how the user was feeling and any restrictions they had set each recipe seemed suited to that person at that time. This is what makes it different from searching for "quick dinner ideas" online. The interface was clean and easy to use. It did not sacrifice the smart technology behind it. This version of the platform lays the groundwork for improvements, such as more detailed nutrition information, more mood options and more personalization. more detailed nutrition breakdowns, a wider range of mood options, and more personalisation across the board.

B. Classification Report

We looked at how the system classified recipes by checking precision, recall, F1-score and overall accuracy across different recipe categories. The inputs we tested were different ways of describing moods different ingredient combinations and different types of food. And the model classified them correctly most of the time. The system was not giving users results, which meant it was saving them time. The model also picked up on the intent even when it was phrased differently which showed it was working well.

The F1-scores were balanced across all categories, which means the system was not relying heavily on one or two main classes. The overall accuracy was good in realistic conditions not just with clean test data. When we looked at these figures together it was clear that the classification layer was doing its job and contributing to the quality of recommendations users received.

C. Discussions

Based on the testing the MoodMunch system really delivers on what it promises. The biggest difference we noticed compared to recipe tools was the mood input. Users were getting meals that suited how they were feeling, not just what ingredients they had. This extra layer made a difference in how relevant the results felt. The system was also easy to use. Pages loaded quickly the steps were clear. Users did not get confused.

The model handled the variation in how people describe things so two users who expressed the same mood in different words both got suitable suggestions. For people who often stand in the kitchen unsure what to cook this system provides a solution.

V. CONCLUSION

We built MoodMunch to solve a problem. Existing recipe tools give you a list based on what you have but they do not consider who you are or how you are feeling. By combining the users' ingredients their current mood, their usual preferences and any dietary needs the system gives one well-suited suggestion rather than overwhelming the user with options. This focus is what makes it more useful in life, than a standard search.

From a standpoint the system worked well. The React frontend, FastAPI backend and MongoDB database all worked together without issues. The authentication prompt-building logic and recipe validation all ran reliably. The meaningful outcome was that users discovered recipes they would never have thought to look for themselves. This tells us that the mood-based layer is really adding something. In the future we plan to work on nutritional information more mood categories and features that explain to users why a particular recipe was recommended to them. The foundation is solid. There is clear room to build on it.

REFERENCES

[1] W. Min, S. Jiang, L. Liu, Y. Rui, and R. Jain, "A survey on food computing," *ACM Computing Surveys*, vol. 52, no. 5, pp. 1–36, 2019, doi: 10.1145/3329168.

[2] J. Freyne and S. Berkovsky, "Intelligent food planning: Personalized recipe recommendation," in Proc. 15th Int. Conf. Intelligent User Interfaces (IUI), 2010, pp. 321–324, doi: 10.1145/1719970.1720021.

[3] D. Elsweiler, C. Ludwig, A. Said, and C. Trattner, "Food and Mood: Just-in-time support for emotional eating," in Proc. CHI Conf. Extended Abstracts on Human Factors in Computing Systems, 2016, pp. 232–245.

[4] Google Gemini Team, Gemini: A Family of Highly Capable Multimodal Models, Google DeepMind Technical Report, 2023.

[5] C. Trattner and D. Elsweiler, "Food recommender systems: Important contributions, challenges and future research directions," arXiv preprint arXiv:1711.02760, 2017.

[6] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Inverse cooking: Recipe generation from food images," in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 10453–10462.

[7] C. Celis-Morales et al., "Effect of personalized nutrition on health-related behaviour change: Evidence from the Food4Me European randomized controlled trial," International Journal of Epidemiology, vol. 46, no. 2, pp. 571–581, 2017, doi: 10.1093/ije/dyw186.

[8] M. Porcheron, J. E. Fischer, S. Reeves, and S. Sharples, "Voice interfaces in everyday life," in Proc. CHI Conf. Human Factors in Computing Systems, 2018, pp. 1–12, doi: 10.1145/3173574.3174214.

[9] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[10] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," Internet Engineering Task Force (IETF), RFC 7519, May 2015.

[11] J. Chen and C.-W. Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in Proc. 24th ACM Int. Conf. Multimedia (MM), 2016, pp. 3241–3249, doi: 10.1145/2964284.2964315.

[12] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901, 2020.

[13] M. Fowler and P. J. Sadalage, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston, MA, USA: Addison-Wesley/Pearson Education, 2013.

[14] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1–2, pp. 1–135, 2008, doi: 10.1561/1500000011.

[15] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012, doi: 10.1109/MSP.2012.2205597.