JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AI-POWERED RECIPE GENERATION BASED ON USER INGREDIENTS, MOOD, AND DIETARY PREFERENCES USING GENERATIVE AI

[1]Deeksha Keshav Naik, [1]Dhanya, [1]Disha Shetty, [1]Manisha Naik [2]Prof.Deepthi Dsouza
[1]Student, [2]Professor [1]CSE,
[1]Srinivas Institute of Technology, Mangaluru, India.

Abstract: In today's fast-paced world, the demand for personalized, efficient, and health-conscious meal planning solutions is steadily increasing. The proposed project, AI-Based Recipe Finder, presents an intelligent web application that generates customized cooking recipes based on user-provided ingredients, emotional state (mood), and dietary preferences. Unlike static recipe databases or tag-based search systems, this application uses Generative AI to create entirely new recipes in real-time, offering a unique experience tailored to each user's needs.The system is developed using a React.js frontend and a FastAPI backend, integrated with Google's Gemini API—a large language model capable of understanding natural language prompts and producing coherent, step-by-step recipe instructions. This project not only simplifies daily meal decisions but also encourages mindful cooking by minimizing food waste and aligning suggestions with the user's lifestyle choices. The generated recipes are displayed on a dedicated page, enhancing clarity and user experience. Additionally, the platform responds swiftly to inputs and provides a "Generating recipe..." indicator during processing, maintaining user engagement. Overall, this project highlights a practical and innovative application of generative AI in everyday life, transforming meal preparation into a more intelligent and personalized process.

Index Terms **– Artificial Intelligence, Generative AI, Recipe Generation, FastAPI, React.js, Gemini API, Personalized Cooking, Natural Language Processing, Mood-Based Recommendation, Dietary Preferences, Web Application, Google AI Studio.**

I. INTRODUCTION

The intersection of artificial intelligence (AI) and daily life continues to expand, with applications emerging in areas ranging from healthcare and finance to entertainment and education. One particularly practical yet underexplored domain is the use of AI in culinary assistance—especially for personalized meal planning. Traditional recipe recommendation systems typically rely on keyword searches, filters, or manually curated datasets, offering limited flexibility and often failing to address user-specific contexts such as available ingredients, mood, or dietary needs [1]. With recent advances in Natural Language Processing (NLP) and Generative AI, it has become possible to generate human-like text outputs that are not only grammatically correct but also semantically rich and context-aware. Large Language Models (LLMs) like OpenAI's GPT and Google's Gemini API have demonstrated capabilities in content generation, task automation, and contextual understanding [2][3]. This shift opens the door to applications that go beyond static information retrieval toward dynamic, AI-generated content tailored to real-time user inputs.

This project, proposes a web-based system titled AI-Based Recipe Finder, which generates personalized recipes using user-provided inputs such as ingredients, emotional state, and dietary restrictions. The system leverages a modern frontend built with React.js, a lightweight and asynchronous backend built using FastAPI, and connects to Google's Gemini 1.5 Flash API for AI-generated content.

The motivation behind this work is twofold:

1. To help users efficiently discover cooking ideas tailored to what they already have, reducing time and food waste.
2. To enable a more emotionally intelligent and health-conscious approach to cooking by factoring in mood and dietary preferences.

This system aims to go beyond the capabilities of traditional food recommendation apps by offering recipes that are dynamically created, context-aware, and truly personalized—demonstrating the practical use of generative AI in everyday life. As eating habits increasingly intersect with digital tools and wellness trends, users are also seeking experiences that feel intuitive and adaptable. A user who is feeling tired may prefer a quick and simple meal, while someone feeling creative might enjoy a

more elaborate cooking experience. These subtle preferences, while often overlooked by traditional systems, can be effectively interpreted and addressed using generative models that understand the semantics of human prompts [4]. Moreover, the global rise in special dietary needs—such as vegetarianism, veganism, gluten intolerance, and ketogenic diets—demands that recipe platforms become more inclusive and adaptable. Static databases often fall short in covering these specialized requirements unless explicitly filtered. Our system addresses this by embedding dietary preferences into the prompt sent to the AI model, enabling it to generate recipes that are more aligned with the user's health and nutritional goals.

The Recipe Finder System is also designed with simplicity and scalability in mind. By offloading the heavy lifting of recipe generation to a powerful cloud-based AI model (Gemini), the application avoids the need for maintaining a local dataset or training a custom model. This not only reduces development complexity but also ensures that the system remains up-to-date with the latest advancements in generative AI. This project bridges the gap between artificial intelligence and personal wellness through a highly practical use case—helping people cook better, smarter, and more creatively. It showcases how modern AI capabilities can enhance user experience by offering context-aware, efficient, and enjoyable meal suggestions based on real-life variables.

II. METHODOLOGY

The proposed system follows a modular and interactive architecture designed to provide real-time, AI-powered recipe generation. The methodology consists of four main components: user interface, backend processing, integration with Google's Gemini API, and output rendering. Each module plays a distinct role in ensuring an effective and user-friendly experience.

A. Frontend Interface

The frontend of the application is built using React.js, providing a responsive and dynamic form for the user to enter:

- Ingredients (comma-separated)
- Current mood (e.g., happy, tired, stressed)
- Dietary preferences (e.g., vegan, gluten-free, low-carb)

Upon form submission, the application displays a loading indicator ("Generating recipe...") and then redirects the user to a separate page that displays the generated recipe.

B. Backend with FastAPI

The backend server, developed using FastAPI, serves as the bridge between the frontend and the Gemini API. It receives the user inputs as a POST request, constructs a natural language prompt, and forwards it to the Gemini model via the official Google Generative AI SDK**.**

```
# Configure your Google Generative AI API
genai.configure(api_key=os.getenv("API_KEY"))

class RecipeRequest(BaseModel):
    ingredients: str
    mood: str
    dietary: str

# Endpoint for generating a recipe
@app.post("/generate-recipe")
async def generate_recipe(request: RecipeRequest):
    prompt = f"Create a detailed recipe using the following ingredients: {request.ingredients}. The user is feeling {request.
    mood} and prefers {request.dietary} recipes. Provide step-by-step instructions."

    # Generate the recipe using the generative AI model
    model = genai.GenerativeModel("gemini-1.5-flash")
    response = model.generate_content(prompt)

    return {"recipe": response.text}
```

Fig. 1: FastAPI endpoint handling recipe generation and prompt construction for the Gemini AI model.

C. Integration with Gemini API

Google's Gemini 1.5 Flash API is used to process the structured prompt. The model generates a complete recipe, including the dish name, ingredients, and cooking instructions. Since the model is cloud-hosted, this reduces the computational load on the user's device.

D. Integration with Gemini API

Google's Gemini 1.5 Flash API is used to process the structured prompt. The model generates a complete recipe, including the dish name, ingredients, and cooking instructions. Since the model is cloud-hosted, this reduces the computational load on

the user's device.

E. Output Rendering

After the recipe is generated, the user is redirected to the recipe display page where the full content is shown in a neatly formatted card. The interface automatically adjusts to accommodate longer recipes and maintains readability across devices.

F. Error Handling and User Feedback

To ensure reliability and usability, the system includes basic error handling mechanisms on both the frontend and backend. If the Gemini API fails to respond or if the user input is incomplete, appropriate error messages are displayed to inform the user without disrupting the interface. On the frontend, the application uses conditional rendering to show a "Generating recipe..." message while waiting for a response from the server. This improves the perceived performance and keeps the user informed about ongoing activity. Errors such as network timeouts, invalid API keys, or malformed inputs are logged on the backend and gracefully communicated to the user via status messages. This makes the application resilient to typical real-world failures and enhances the user experience**.**

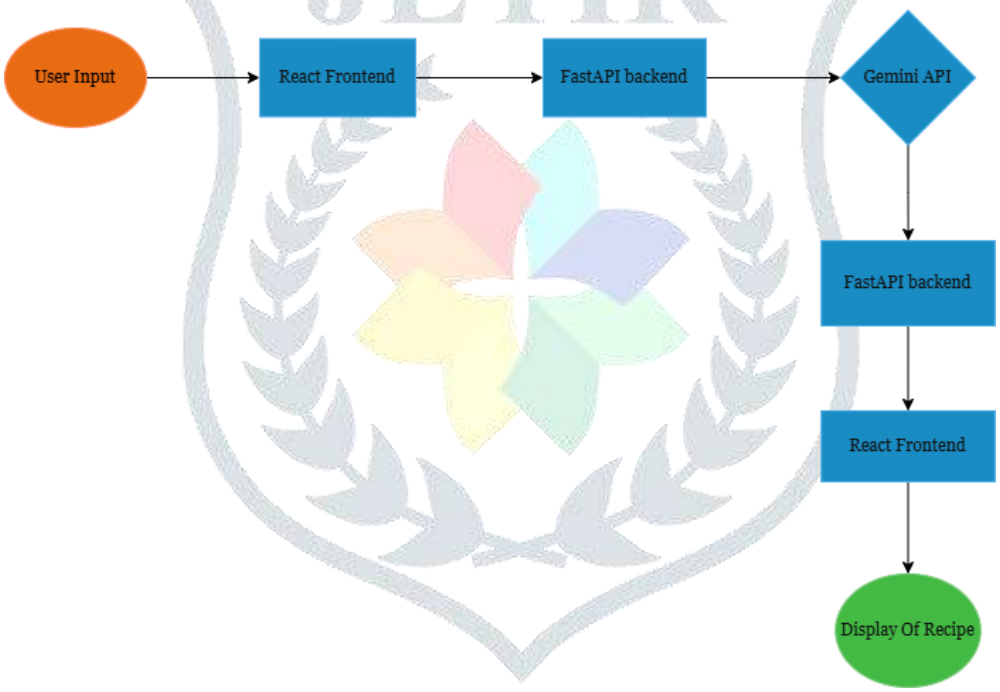

Figure. 2: AI Recipe Finder – Functional Flow Diagram

III. RESULTS AND DISCUSSION

The AI-Based Recipe Finder system was successfully implemented and tested across multiple scenarios to evaluate its ability to generate personalized and contextually relevant recipes. The results indicate that the application performs well in terms of response quality, user engagement, and adaptability.

```
{
  "ingredients": "tomatoes, onions, garlic",
  "mood": "happy",
  "dietary": "vegan"
}
```

Figure. 3: Ingredients, Mood and Dietary Inputs

A. Functional Testing

The system was tested with various input combinations involving different ingredients, moods, and dietary preferences. In each case, it successfully generated relevant and coherent recipes that matched the user's context. For example, when the mood was set to "tired," the AI returned simpler recipes with fewer steps, while for "happy" or "excited," it generated more elaborate and creative dishes. The system also respected dietary restrictions, ensuring that vegan or gluten-free recipes did not include incompatible ingredients.

The consistency of results across different inputs indicates that the system handles variations in user data effectively. Whether the user entered a single item or a detailed list of ingredients, the AI adapted well and returned recipes that felt natural and practical. This reinforces the reliability of both the prompt construction and the underlying AI model in understanding user intent.

Additionally, the application's behavior remained stable even with incomplete or slightly misspelled inputs, showing tolerance for real-world usage. For instance, an input like "tomto" still resulted in a tomato-based recipe due to the AI's language understanding capabilities. The frontend also validated that all required fields were filled before sending a request to the

backend. This ensured consistent interaction flow and prevented errors. Overall, the functional testing confirmed that the system performs reliably and intelligently under various user scenarios.

Figure. 4: Recipe Finder Page

Figure. 5: Generating Recipe

Figure. 6: Generated Recipe Page with Instructions to Cook

A. User Feedback

Informal feedback from test users indicated:

- Satisfaction with the creativity and clarity of the generated recipes.
- Appreciation for the simplicity of the interface.
- Surprise at how the system "understood" their mood and cooking constraints.
- A majority of users agreed that the AI-generated recipes were useful and would consider using the app regularly for inspiration.

B. Performance

The performance of the AI Recipe Finder system was evaluated based on response time, frontend behavior, and system stability during multiple test cycles. On average, the application was able to generate a recipe and display it to the user within

2.5 to 4 seconds, depending on the complexity of the input and internet latency.

The frontend, built using React.js, remained responsive and smooth even when the backend was processing longer prompts. During testing, the "Generating recipe" message provided clear feedback to users, which helped reduce perceived wait time and improved the overall user experience.

The backend, implemented using FastAPI, efficiently handled API requests and constructed prompts without introducing delays. Even under repeated requests, the system maintained consistent output quality and minimal memory usage. The use of Google's Gemini API further enhanced reliability, as it allowed for fast and scalable content generation without local model hosting. In edge-case tests, such as sending large ingredient lists or intentionally vague moods — the system handled responses gracefully and returned usable recipes without crashing or timing out. This demonstrated the robustness and scalability of the architecture for light-to-moderate real-world use.

D. Discussion

The project demonstrates that generative AI can effectively be applied in niche real-life tasks like recipe creation. It also highlights the importance of designing user prompts and interfaces thoughtfully, as even minor input changes can influence the recipe tone and content. The loading feedback and seamless navigation to the output page were key in maintaining user engagement.

IV. CONCLUSION

The development of the AI-Based Recipe Finder marks a meaningful step forward in the integration of generative AI into daily life. By leveraging user-specific inputs such as available ingredients, emotional state, and dietary preferences, the system is capable of generating unique, coherent, and personalized recipes on demand. The successful fusion of a modern frontend (React.js), a high-performance backend (FastAPI), and the power of Google's Gemini API demonstrates how emerging technologies can be orchestrated to provide intelligent, context-aware solutions.

The system not only fulfils its core objective of recipe generation but also provides an intuitive and responsive user experience. Real-time loading indicators, prompt feedback, and clear recipe formatting contribute to a smooth and engaging workflow. Additionally, the project highlights how AI models can adapt to mood-driven and health-oriented content requests—transforming cooking from a routine task into a more enjoyable and personalized activity. In testing scenarios, the application proved to be robust, adaptable, and efficient, even when handling edge cases or varying levels of input detail. Its quick response time and the contextual accuracy of the generated outputs underscore the effectiveness of prompt engineering and proper API utilization. Overall, the project demonstrates that generative AI is no longer limited to experimental or entertainment uses—it can now be applied in meaningful, practical ways that benefit everyday users.

Beyond the technical implementation, the project also emphasizes the growing relevance of human-centered AI applications. The ability to tailor outputs based on emotional cues (such as mood) adds a layer of empathy and adaptability to the system—an aspect often missing in traditional recommendation engines. This unique feature allows the application to stand out by offering suggestions that not only meet nutritional needs but also resonate with the user's state of mind, potentially making mealtime more enjoyable and satisfying. Moreover, the system promotes sustainable cooking practices by encouraging the use of existing ingredients. This can lead to reduced food waste in households and smarter grocery habits. By suggesting recipes aligned with what users already have at home, the application contributes to responsible consumption—an increasingly important goal in the age of environmental consciousness.

From a development perspective, the modular architecture allows for scalability and flexibility. Since the system relies on cloud-based AI (Gemini API), it avoids the need for large on-device models or storage of datasets, making it lightweight and adaptable for future deployment as a mobile or desktop app. This design also makes it feasible for integration with other smart kitchen technologies or digital assistants. This project not only serves as a demonstration of what generative AI can achieve in the culinary space but also reflects the broader potential of such models to enhance everyday life. It bridges the gap between artificial intelligence and personal wellness in a creative, user-friendly way—setting the stage for further exploration and innovation in AI-driven lifestyle tools.

V. FUTURE WORK

While the current system achieves its primary goal of generating recipes dynamically, there are several avenues through which the application can be significantly enhanced in future versions. One important area of improvement is the addition of image recognition functionality. By integrating computer vision models, users could upload photos of their available ingredients instead of typing them manually. This would further streamline the input process and make the system more user-friendly, especially for non-technical audiences. Another enhancement could involve voice input support, allowing users to speak their mood and ingredients, which would make the system more accessible to users with disabilities or in hands-busy environments like kitchens. In addition to usability improvements, the project can be extended by incorporating user authentication and profile management. This would allow individuals to save their dietary preferences, view previous recipes, and track their interactions with the system over time. For users with health goals, a meal planning module can be added where the AI suggests recipes across a full day or week, based on nutritional goals like calorie control, protein intake, or weight loss. Integration with third-party APIs such as nutrition analysis platforms would also allow the app to provide nutritional breakdowns (e.g., calories, fat, protein) of each generated recipe, making it useful for health-conscious users or fitness tracking.

Lastly, multilingual support can expand the application's reach by enabling recipe generation in multiple languages, making it valuable for users across different regions. By addressing these possibilities, the AI Recipe Finder has the potential to evolve from a recipe generator into a comprehensive AI-driven smart cooking assistant.

REFERENCES

[1]. Google AI Studio. (2024). *Makersuite and Gemini API documentation.* Available: https://makersuite.google.com

[2]. FastAPI Documentation. (2024). *Fast, modern web framework for APIs with Python.* Available: https://fastapi.tiangolo.com

[3]. React.js Official Documentation. (2024). *A JavaScript library for building user interfaces.* Available: https://react.dev

[4]. Google Cloud. (2024). *Gemini Model Overview.* Available: https://cloud.google.com/vertex-ai/docs/generative-ai/overview

[5]. OpenAI. (2023). *GPT-4 Technical Report.* Available: https://openai.com/research/gpt-4

[6]. Arora, V., Sharma, R., & Kaur, P. (2023). *AI-Powered Content Generation: A Comparative Study*, International Journal of Artificial Intelligence Research.

[7]. Khan, R., & Shaikh, M. (2022). *Web-Based Recipe Recommendation Using NLP Techniques*, International Journal of Computer and Information Technology (IJCIT).

[8]. Mozilla Developer Network. (2024). *Axios – Promise-based HTTP client.* Available: https://axios-http.com

[9]. Draw.io. (2024). *Diagrams.net – Online tool for system diagrams.* Available: https://app.diagrams.net

[10]. GitHub. (2024). *Google Generative AI Python SDK.* Available: https://github.com/google/generative-ai-python