”Developing a Kitchen Recipe Chatbot Using Google Generative AI and Streamlit”

Ashok Sai Doredla, Sai Ganesh Donka, and Varun Aditya Madala

***Abstract*—The following work discusses the design and development of a kitchen recipe chatbot using Google Generative AI, which is represented by the Gemini model, and Streamlit. This chatbot will help users look for recipes depending on what ingredients they have in their cuisine or simply state what dish they want. It will be implemented through API calls to Google's generative model, so this bot will make sure to give very long recipes with ingredient lists, preparation instructions, and culinary tips. This paper focuses on the methodology and shows results that prove the effectiveness of the chatbot in creating user-specific recipes, thus serving as an accessible cooking assistant for a great number of people.**

***Index Terms*—Conversational AI, recipe chatbot, generative AI, Streamlit, Hugging Face, data-driven applications.**

I. Introduction

With the rise of artificial intelligence, conversational agents have become more versatile, extending their usefulness across numerous domains, from customer support to healthcare assistance[1]. In the culinary field, where creativity and specificity are essential, a virtual assistant can provide significant value by delivering tailored recipe recommendations, meal preparation instructions, and cooking tips. The need for such an assistant arises from the limitations users often face: the challenge of identifying suitable recipes based on the ingredients they already have, or simply needing quick, reliable culinary guidance.

This study explores the development of a kitchen recipe chatbot aimed at addressing these specific challenges. By leveraging generative AI, the chatbot can provide users with recipes that are contextualized to their unique requests. The primary aim is to create a culinary assistant that responds effectively to user queries—whether it's a list of available ingredients or a request for a specific dish[2]. This approach allows for an interactive, informative, and user-friendly experience that can support home cooks, culinary enthusiasts, and professionals alike.

”The research question that would be guiding this study is: How can generative AI be exploited to provide personalized and accurate recommendations in culinary arts via a chatbot interface? In addressing this, we used Google's Generative AI Gemini-Pro model well known for its coherence in natural language processing along with Streamlit, a Python framework which allows for fast, interactive development. The study investigates the development of a chatbot that suggests recipes for user needs, best practices for AI-driven culinary assistance, and the greater ramifications of AI in culinary settings.

II. Literature Review

This has been the focus of many studies that have investigated the use of AI and conversational agents in food applications. So far, conversational agents or chatbots have been useful in carrying out a lot of activities automatically that require consistent user interaction and personalized responses. Previous research has identified several ways in which AI can support gastronomic experiences, such as personalized meal planning, and suggesting ingredient substitutions that could very well result in a high user experience and satisfaction in such food-related interactions[4].

Generative AI advancements, exemplified by models such as GPT and Gemini have significantly enhanced chatbot capabilities in recent years[2]. In fact, Google's Gemini models are products of generative AI that comes out strong in understanding human-like text and hence generating human-like text with accurate responses to user queries with a great deal of relevance and coherence. This often has to be elaborated in culinary contexts to receive detailed, well-reliable instructions from users. Generative models, such as Gemini, pick up much more context and can therefore provide very long, step-by-step recipes, which are then consistent with the ingredients specified by the user or even dish preferences.

Streamlit has, therefore, as a framework, been used widely in the deployment of machine learning applications because of its ease and flexibility in constructing user-friendly web applications. Similarly, combining studies conducted on generative models with Streamlit increases accessibility and ease of use of such AI tools; hence, it is ideal for applications where user interaction is paramount. This would include several research works, that have focused on the ways in which recipe recommendation systems could be enhanced through AI-driven personalization[3]. In each case, these works point toward the fact that AI can extend the capabilities of culinary tools to make them adaptive to diets, preferences, and ingredient availability. Such findings, in the context of this project, provide grounds for the design of the chatbot, where flexibility, user-specific suggestions, and ease of use are underlined as important aspects in a culinary assistant. Building on this significant research, our study aims to advance the understanding of AI's role in developing recipe chatbots that are both practical and efficient. By utilizing advanced generative AI models, we demonstrate the potential of these tools to simplify cooking experiences, offering personalized solutions that cater to individual needs. This exploration emphasizes

the real-world applications of AI in everyday life, paving the way for future advancements that could make cooking more accessible and enjoyable for users of all backgrounds.

III. Methodology

A. Environment Setup and API Integration

For secure management of environment variables, particularly sensitive ones like API keys, we initially used the `dotenv` library to load these variables from a `.env` file. This method ensures that sensitive information, such as the API key for Google's Generative AI, is not exposed in the source code, thus enhancing security. However, during deployment on Hugging Face Spaces, the `.env` file was not uploaded due to platform restrictions. Instead, we utilized the secrets and variables feature available in the Hugging Face Spaces settings to securely store the Google API key. This approach allowed us to manage API credentials directly within the deployment environment, ensuring secure and efficient access to Google's Generative AI model without compromising sensitive information. The chatbot accesses the language model using this securely stored API key, configured with the command `genai.configure(api_key = api_key)`. This setup facilitates a secure, seamless connection to Google's Generative AI, enabling effective data flow and functionality for the chatbot.

B. Generative AI Model Configuration

The particular model, Gemini-Pro, from Google Generative AI was chosen for this project due to its exceptional capabilities in natural language generation. It produces highly coherent responses, which are essential for delivering accurate and contextually appropriate replies. We initialize a conversational interface with the model by calling it with `start_chat(history = [])`. This setup allows us to keep track of the chat history, maintaining context across multiple exchanges. Preserving chat history enhances the user experience, as it enables the chatbot to consider what the user has communicated in previous interactions, resulting in more relevant and personalized responses.

C. User Interface and Framework

We used Streamlit because it is a Python-based framework, making the design of an interactive user interface for the chatbot more straightforward. Streamlit is simple and flexible for building and deployment, with minimal overhead. The interface includes chat history, user input, and real-time bot responses. The layout was optimized for user-friendliness using `st.set_page_config()`. Additionally, the chat history is stored in `st.session_state` to ensure persistence throughout the conversation, allowing users to maintain a continuous interaction with the chatbot.

D. Prompt Engineering and Retry Mechanism

An important aspect of the chatbot's performance is the structure of the initial prompt, which frames the model as a culinary expert. The prompt requests recipes with ingredient amounts, step-by-step instructions, and cooking tips, ensuring that the responses are both relevant and comprehensive for culinary queries. This approach to prompt engineering enables the chatbot to deliver detailed and contextually appropriate responses tailored to user cooking needs.

E. Workflow of the Kitchen Recipe Chatbot System

The following *Fig 1* illustrates the complete workflow of the Kitchen Recipe Chatbot, showcasing the steps involved in generating recipe responses based on user input. This system begins with user interaction, where a query is processed, tokenized, and then sent to Google's Gemini LLM (Large Language Model) for response generation. The model returns a structured response, which is formatted for clarity before being displayed to the user and provides the capability to save the generated response in a `.txt` format.

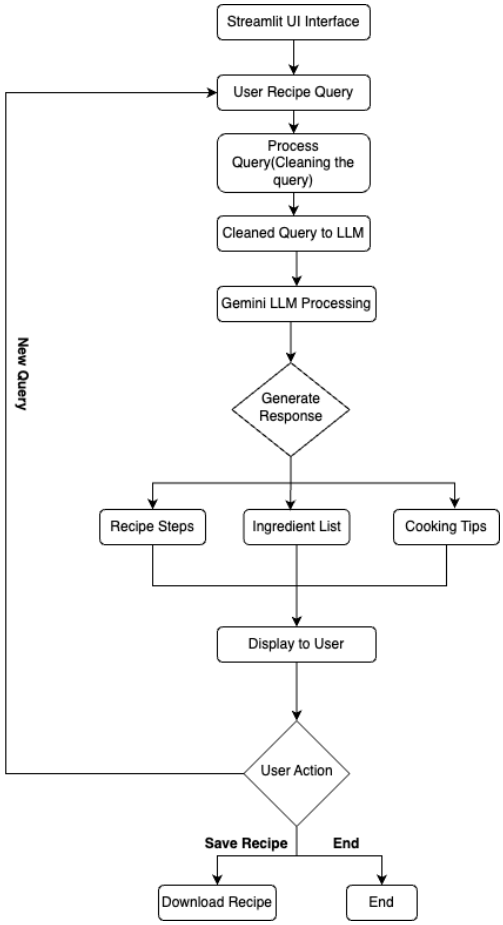

Fig. 1. Flow diagram of the Recipe Chatbot.

F. Retry Mechanism for Response Reliability

Since APIs can occasionally fail, we implemented a retry mechanism within the function `get_bot_response_with_retry()`. This mechanism attempts to retrieve a response up to five times, with a five-second delay between attempts, enhancing the system's reliability. In case of an error, it waits briefly before retrying.

If warnings or error messages occur, they are displayed to the user, ensuring transparency. This approach allows the function to work effectively even in cases of temporary network issues, thereby providing a smoother user experience.

IV. Outcomes

The kitchen recipe chatbot demonstrated a strong ability to interpret user queries and generate appropriate responses, delivering detailed recipes, instructions, and culinary tips based on the user's input. Key outcomes from the implementation include:

A. User Query and Recipe Generation

It is designed to understand user queries in an intelligent manner related to cooking and creating recipes. The user may give their input based on the list of available ingredients or, for example, on a specific name of any dish they want to prepare. The system processes these smoothly and will generate a customized recipe that fits all user needs. For instance, when the user inputs "pasta, tomatoes, garlic,", the chatbot understands these as ingredients on which to anchor a recipe and provides a cooking description in formal terms. This encompasses preparation steps, exact measurements, and time for preparation and cooking.

The flexibility of the chatbot ensures that it will respond with flexibility to the diverse types of needs from different users, be they in need of using leftovers or in need of preparing a fresh meal. It also depicts the adaptability in its usage-from people who started to cook to the even more professional ones. Secondly, it responds in a more formatted or user-friendly manner, which makes following through in its preparation easy. This capability increases not only the usability for users but also shows that the chatbot can be used as a trusted companion during cooking.

B. Enhanced User Engagement

While the recipes are great, this chatbot goes one step beyond the basics and delivers some really valuable culinary tips that enhance the user experience. Included are suggestions for ingredient substitutions to enable users to adapt recipes based on what they may have on hand or their particular dietary needs. For example, if a recipe calls for butter, it would suggest olive oil as a healthier alternative or a non-dairy substitute for those with dietary restrictions.

It also informs about the cooking methodology and therefore helps users perfect their art of cooking. These may range from suggestions to improve the texture of a meal to tips on how to get the best flavor out of it. In this respect, the chatbot becomes not only a simple recipe provider but a full cooking assistant by providing this level of guidance. It allows users to make variations and customize recipes to their taste, making cooking even more intuitive and pleasant. This extra personalization aspect shows how well this chatbot can adapt to different cooking styles and needs, further strengthening its position as a multivaried tool in the kitchen.

C. Error Handling and Consistency

This had a retry mechanism in place that really helped bypass most of these interruptions since responses would be delivered with a minimal number of tries by the user. This error-handling feature is paramount for the smooth flow of the experience a user would get from such a system, hence minimal frustration from users as far as connectivity issues or an error in the API is concerned.

V. Results

This section presents the outcome of implementing the kitchen recipe chatbot using Google's Generative AI (Gemini-Pro Model) and Streamlit framework. The chatbot effectively provides customized recipe suggestions based on user input, which can include a specific dish or a list of available ingredients. The results demonstrated the chatbot's efficiency in generating accurate, user-friendly recipes with detailed instructions and additional tips.

A. User Interaction and Response Generation

As shown in *Fig 2*, upon entering "cappuccino" as the recipe request, the chatbot returns a complete recipe, including ingredients, step-by-step instructions, and culinary tips. The user-friendly interface ensures ease of interaction, while the generated response is clear, structured, and adheres to the request context.

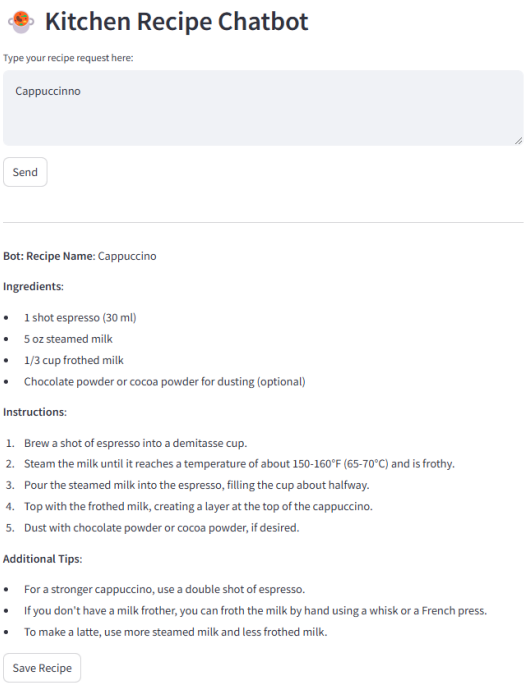

Fig. 2. Chatbot user interface displaying the response for a recipe.

- **Recipe Details**: For each request, the chatbot provides a list of ingredients and clear instructions for preparation. For instance, when the user requested a "cappuccino" recipe, the bot generated a detailed response, including

the ingredients and step-by-step instructions for preparing the drink.

- **Additional Tips**: The chatbot also provides additional cooking tips to enhance the user's cooking experience, such as the recommendation to use double-shot espresso for a richer flavor and alternative methods for making a latte using steamed milk. This added information showcases the chatbot's capacity to deliver comprehensive guidance, similar to that of a culinary expert.

B. File Download Feature

Additionally, as shown in *Fig 3*, the chatbot allows users to download the generated recipe in a `.txt` format, providing an accessible way for users to save and refer back to recipes. This functionality enhances user experience by offering convenient access to recipe information beyond the chat interface.

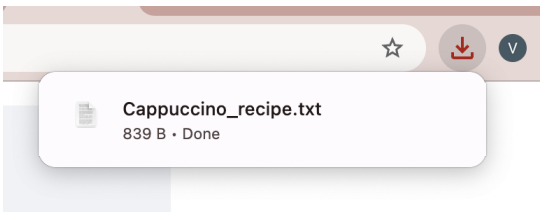

Fig. 3. The Downloaded Recipe Text File.

C. Ensuring Relevant Interactions and Reliability

A query irrelevant to recipes during testing was handled very robustly by the chatbot: It handled irrelevant queries successfully. Look at the *Fig 4* for instance, every time users ask questions unrelated to recipes or cooking, the chatbot's response is a polite message indicating its purpose as an assistant designed for recipes. This feature helps the user redirect his questions toward the appropriate ones and continues his smooth and reliable experience.

Fig. 4. Chatbot responding to an irrelevant query.

VI. Conclusion and Future Work

The project is a very successful application of generative AI in the culinary world through Google's Gemini-Pro model, Streamlit, and Hugging Face spaces for deployment which developed into a responsive user-friendly kitchen recipe chatbot. This chatbot is helpful for both home cooks and gastronomes alike with its customized recipe recommendations and cooking tips based on what the user inputs. Also, the incorporation of the retry mechanism increased its reliability by ensuring it always works, even when temporary API errors pop up. This could be a chatbot that is likely to be significantly enhanced in the future to enhance its functionality and user experience. One such important feature would be to extend its knowledge of cooking by training on more and larger datasets that would eventually make it capable of being useful with a wide range of cuisines, dietary varieties, and regional styles. Adding on to this would be the inclusion of voice recognition, which would further enhance usability by allowing smooth operation without necessarily having hands free for typing. This would be convenient and helpful for a busy kitchen environment where the user may have no free hands to type anything. Another important feature would involve saving interaction history for different login users. By including a logging-in mechanism, the chatbot could store interaction histories personalized to individual users so that users can revisit previous queries, retrieve recipes generated earlier, and track their culinary journey over time. This would introduce a high degree of personalization, whereby the chatbot is in a position to make recommendations or respond in certain ways in line with individual preferences and usage patterns. That would be conducive to an environment that elicits better user experience and efficiency. Moreover, the extension of language support would make the chatbot more usable by people who do not speak English, hence multiplying its popularity and inclusiveness. Multiple languages supported would mean that people from different linguistically diverse backgrounds can use its functionalities, hence versatile and more appropriate for many uses. Taken together, these improvements gesture toward a bright future where AI-powered solutions, such as this chatbot, are concerned with the daily routines of adding value to user experiences and delivering very accurate answers, adaptive to particular needs.

References

[1] M. Tanaka, K. Yamaguchi, S. Inoue, and K. Kojima, "Smart Cuisine: Generative recipe & ChatGPT powered nutrition assistance for sustainable cooking," *Procedia Computer Science*, vol. 225, pp. 2028–2036, 2023. https://www.sciencedirect.com/science/article/pii/S1877050923013510

[2] Asha, P. N., Ramachandra, H. V., Sowmya, S., Chaithra, Subramanya, S. G., & Satheesha, T. Y. (2024). NLPANTRY: AI-powered recipe creation. *Acta Scientiae, 7*(1), 486-495. https://doi.org/10.17648/acta.scientiae.6383

[3] Brendan Hanon, Yulia Kumar, J. Jenny Li, and Patricia Morreale, "Recent advancements in AI-driven personalized health recommendations," *Preprints*, 2024. https://www.preprints.org/manuscript/202404.0334

[4] J. Chu, "Recipe Bot: The Application of Conversational AI in Home Cooking Assistant," in *Proceedings of the 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, 2021, IEEE. https://doi.org/10.1109/ICBASE53849.2021.00136.

[5] S. Safitri, M. A. C. Bhakti, T. Mantoro, and W. Wandy, "Cooking and food information chatbot system using GPT-3," in *9th International Conference on Computing, Engineering, and Design (ICCED)*, 2023, IEEE. https://doi.org/10.1109/ICCED60214.2023.10425553.