Major Project Report On

"Cravify: A Personalized Recipe Recommendation System Using Multimodal Inputs and Contextual AI"

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

ANKITA DARMAL (121CP1188B)

DEVEN DESHMUKH (121CP1170A)

KASTURI DHOPAT (121CP1237B)

ISHA KADAM (122CP3199B)

Under the supervision of:

Prof.-Vidya Manoj Bharde

Department of Computer Engineering

MGM College of Engineering and Technology,

Kamothe, Navi Mumbai- 410209

University of Mumbai (AY 2024-25)

CERTIFICATE

This is to certify that the project entitled "**Cravify: A Personalized Recipe Recommendation System Using Multimodal Inputs and Contextual AI**" is a bonafide work of **Ankita Darmal (121CP1188B), Deven Deshmukh (121CP1170A), Kasturi Dhopat (121CP1237B), Isha Kadam (122CP3199B)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Computer Engineering**"

Prof.-Vidya Manoj Bharde
Project Guide

Dr. Jyoti Chavhan
Project Co-ordinator

Dr. Rajesh Kadu
Head of Department

Dr. Geeta Lathkar
Director

Project Report Approval for B. E.

This thesis/dissertation/project report entitled **"Cravify: A Personalized Recipe Recommendation System Using Multimodal Inputs and Contextual AI"** by **Ankita Darmal (121CP1188B), Deven Deshmukh (121CP1170A), Kasturi Dhopat (121CP1237B), Isha Kadam (122CP3199B)** is approved for the degree of **Computer Engineering**

Examiner

1. ----------

Internal Examiner Name & Sign

2. ----------

External Examiner Name & Sign

Date :

Place : **MGMCET, Navi Mumbai**

Declaration

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included .I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academics honesty and integrity and have not misrepresented or fabricated or falsified any idea in my submission.I understand that any violation of above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Ankita Darmal (121CP1188B)

Deven Deshmukh (121CP1170A)

Kasturi Dhopat (121CP1237B)

Isha Kadam (122CP3199B)

Date:

Place : **MGMCET, Navi Mumbai**

ABSTRACT

In an era of digital convenience, recipe recommendation systems still struggle to capture the deeply personal and emotional nature of cooking. Cravify introduces a novel AI-driven approach to meal planning by combining multimodal user inputs—including voice, image, text, and mood data—with contextual awareness to generate highly personalized and emotionally intelligent recipes. Unlike traditional systems that provide static, predefined content, Cravify dynamically adjusts to a user's current ingredients, dietary restrictions, time availability, emotional state, and even craving intensity.

The system is architected using a fusion of state-of-the-art technologies: the T5 model from Hugging Face for recipe generation, Google Gemini for real-time conversational responses, and YouTube API integration for visual learning. Cravify's interface, built with Streamlit, ensures an intuitive experience where users can interact naturally—selecting ingredients, adjusting creativity levels, or asking culinary questions. Biometric indicators (such as mood or energy level) are used to fine-tune not only recipe complexity but also flavor profiles and cooking methods.

Cravify serves diverse user needs—from novice cooks and health-conscious individuals to time-pressed parents—while also addressing societal goals such as reducing food waste and promoting nutritional equity. By transforming recipe discovery into a conversational, adaptive experience, Cravify reimagines what AI can offer in the kitchen: not just suggestions, but a virtual cooking partner that understands, supports, and evolves with the user.

ACKNOWLEDGEMENT

We are profoundly grateful to **Prof. Vidya Manoj Bharde** for her expert guidance and continuous encouragement throughout to see that this project rights its targets since it commencement to its completion.

We would like to express deep appreciation towards Dr. Geeta Lathkar, Director. Mahatma Gandhi Mission's College of Engineering and Technology, Prof. Rajesh Kadu. Head of Department of Computer Engineering whose invaluable guidance supporting us in completing this project.

At last, we must express our sincere heartfelt gratitude to the all staff members of Computer Engineering Department who helped us directly or indirectly this course of work.

Table of Contents

Table of Figures

Chapter 1: INTRODUCTION

1.1 Introduction

The act of cooking has evolved from a survival necessity to a creative, emotional, and cultural experience. In recent years, the intersection of technology and food has produced a wide array of cooking assistants and recipe apps [1]. Yet, despite technical advancements, most of these platforms continue to operate under rigid, linear logic—recommending fixed recipes, relying solely on keyword searches, and ignoring the broader emotional, contextual, and sensory factors that influence food choices [1][2]. These limitations have highlighted the need for a more holistic, human-centric approach to digital cooking support.

Traditional systems typically focus on one or two factors—ingredient availability or nutritional value—but fail to consider the dynamic and deeply personal nature of meal decisions. Factors such as time constraints, energy levels, emotional well-being, or a sudden craving for a specific flavor profile often go unnoticed. For instance, a user searching for dinner ideas after a stressful day may need comfort food, not just something low in calories. This emotional-contextual disconnect is where most current platforms fall short [3][4].

Cravify addresses these gaps by introducing a new paradigm in intelligent meal planning—an AI-powered cooking assistant that listens, understands, and adapts. By integrating multimodal inputs such as voice commands, images of ingredients, typed preferences, and biometric mood cues, Cravify creates a complete user profile in real time. Its contextual engine processes these inputs to generate recipes that are not just edible, but desirable, relevant, and emotionally aligned [5][6].

The system goes beyond static databases by incorporating adaptive recommendation logic and natural language understanding. Powered by cutting-edge technologies like the Hugging Face T5 model, Google Gemini API, and YouTube integration, Cravify offers dynamic, mood-aware recipe generation, real-time conversational support, and visual learning through curated cooking videos [7]. Its Streamlit-based frontend ensures a smooth, interactive experience for both casual users and serious home cooks.

Moreover, Cravify champions two important modern-day goals: reducing food waste by intelligently matching recipes to on-hand ingredients, and promoting health equity by offering accessible dietary guidance based on personal nutrition needs [2][4][8]. Whether users seek quick meals with what's in the fridge or an elaborate dish for a special occasion, Cravify acts as a virtual sous-chef—learning from each interaction and evolving to better serve the user.

By blending emotional intelligence, multimodal understanding, and AI-driven adaptability, Cravify reimagines recipe discovery not as a transaction, but as a rich, supportive dialogue. It marks a step toward a more intuitive, inclusive, and deeply personalized cooking experience—one that meets users where they are, both physically and emotionally, and guides them with empathy, intelligence, and creativity [1][9][10][11].

Chapter 2: LITERATURE SURVEY

2.1 Literature Review

The following table presents a literature survey of various research papers focused on food recipe recommendation systems. It includes the paper titles, author names, publication details, and a brief comparison.

	PAPER TITLE	AUTHOR NAME	PUBLICATION DETAILS	Comparison
1.	From Market to Dish	Lin Zhang; Jianbo Zhao; Si Li; Boxin Shi; Ling-Yu Duan	12 July 2019	Multi-ingredient Image Recognition for Personalized Recipe Recommendation
2.	mirurecipe	Yoshiyuki Kawano; Takanori Sato; Takuma Maruyama; Keiji Yana	19 July 2013	A mobile cooking recipe recommendation system with food ingredient recognition
3.	Recipe Recommendations for Toddlers Using Integrated Nutritional and Ingredient Similarity Measures	Nantaporn Ratisoontorn	25 June 2022	Recipe Recommendations for Toddlers Using Integrated Nutritional and Ingredient Similarity Measures
4.	Food Recipe Recommendation Based on Ingredients Detection Using Deep Learning	Shafaat Jamil Rokon	december 21	Food Recipe Recommendation Based on Ingredients Detection Using
5.	Ingredients to Recipe	Manasi Swain; A R Manyatha; Amulya S Dinesh; Gambhire Swati Sampatrao; Mihir Soni	27 March 2023	A YOLO-based Object Detector and Recommendation System via Clustering Approach

2.1.1 Problem Definition

Despite the growing availability of digital cooking platforms and recipe recommendation systems, existing solutions remain limited in their ability to deliver personalized and context-aware culinary guidance. Most current applications rely on static databases, fixed ingredient lists, and keyword-based search mechanisms that do not consider the user's emotional state, available time, cooking skills, or specific cravings. This leads to a mismatch between what users want to eat and what is recommended, resulting in decision fatigue, food waste, and a lack of engagement in the cooking process.

Moreover, these systems typically fail to:

- Understand user intent conveyed through natural language or voice input.
- Adjust recipes dynamically based on available ingredients or dietary restrictions.
- Incorporate real-time contextual cues such as mood, energy levels, or time of day.
- Offer interactive, human-like conversation for clarifications or adaptations.
- Provide visual aids or cooking demonstrations to support varied learning styles.

The lack of emotional intelligence, adaptability, and multimodal support in existing platforms creates a significant gap in user experience, especially for individuals seeking personalized cooking solutions that align with their lifestyle, goals, and preferences.

Therefore, the core problem is the absence of an intelligent, emotionally responsive, and multimodal cooking assistant that can generate personalized recipes and guide users interactively based on real-time contextual, emotional, and ingredient-based inputs.

Cravify aims to address this problem by developing a holistic AI-powered system that transforms recipe recommendation into a conversational, adaptive, and emotionally attuned experience.

2.2 Limitations of Existing System

Despite advancements in AI and food technology, current recipe recommendation systems exhibit several limitations that hinder their effectiveness and user engagement. These limitations fall into three main categories: contextual awareness, personalization depth, and interaction quality.

1. Lack of contextual awareness Most existing platforms ignore real-world factors such as the user's emotional state, energy level, available time, and specific motivations for cooking. They operate in a context-agnostic manner, offering the same recipe regardless of whether the user is cooking after a long workday or preparing a weekend family meal.
2. Rigid input handling Traditional systems depend heavily on keyword-based search mechanisms. Users must know exactly what they want or how to phrase their input, which creates a barrier for those with vague cravings or limited cooking knowledge. These systems struggle to interpret natural language or voice inputs meaningfully.
3. Static and predefined recipe suggestion Existing apps typically draw from fixed databases of recipes. They do not generate new content or modify recipes dynamically based on ingredient availability, dietary restrictions, or user preferences. As a result, users are often forced to adapt themselves to the recipe rather than the other way around.
4. Limited personalization While some systems offer basic personalization (e.g., filtering recipes by cuisine or diet), they fail to account for more nuanced factors like cooking experience, flavor preferences, past behavior, or special occasions. The lack of learning from user interactions leads to repetitive and impersonal suggestions.
5. Poor real-time assistance Chatbot features in existing systems are typically rule-based and limited to predefined queries. They cannot handle follow-up questions, offer substitutions, or explain cooking techniques in context. This reduces their usefulness during active meal preparation.
6. No Emotional intelligence Current systems do not adjust recommendations based on the user's mood, stress level, or energy. Emotional eating cues—such as the need for comfort food, indulgence, or light meals—are overlooked, despite being key drivers of food choices.
7. Insufficient Visual Support Few platforms integrate video guidance or visual aids, making it harder for users—especially visual learners or beginners—to understand and follow complex recipes.

2.3 Proposed System

In the proposed system, we provide an interactive webpage where users can input any primary ingredient, and the model will generate a list of recipe suggestions based on that ingredient. Users can also add extra ingredients to further refine their recipe choices, ensuring the suggestions align with their preferences and available resources. The system utilizes advanced machine learning algorithms to analyze ingredient combinations, delivering diverse and high-quality recipes.

Additionally, the platform is designed to be user-friendly, enabling seamless navigation and quick access to recipes. It incorporates features such as dietary filters and skill level adjustments, allowing users to tailor the recommendations according to their culinary needs. The system aims to foster creativity in cooking while reducing food waste by encouraging users to make the most of what they have on hand. Ultimately, this personalized recipe recommendation tool enhances the cooking experience and promotes efficient meal preparation.

Here are the advantages of the proposed system:

1. Personalized Recommendations: The system offers tailored recipe suggestions based on user- inputted ingredients, enhancing the relevance of recommendations.
2. Enhanced Cooking Creativity: By allowing users to experiment with additional ingredients, the system encourages creativity and exploration in cooking.
3. Reduction of Food Waste: The platform promotes the use of available ingredients, helping users minimize food waste by making the most of what they have.
4. User-Friendly Interface: The intuitive design of the webpage ensures easy navigation, making it accessible for users of all skill levels.
5. Dietary Customization: Users can filter recipes based on dietary preferences or restrictions, ensuring that recommendations align with individual health needs.
6. Time Efficiency: The quick generation of recipe suggestions saves users time in meal planning and preparation, making it easier to decide what to cook.
7. Skill Level Adaptation: The system can suggest recipes suitable for various cooking skill levels, catering to both beginners and experienced cooks.
8. Integration of Feedback: Users can provide feedback on recipes, which can be used to improve future recommendations, fostering a responsive system.
9. Exploration of New Cuisines: The diverse recipe suggestions encourage users to try new cuisines and cooking styles, broadening their culinary horizons.

Chapter 3: METHODOLOGY

3.1Methodology

The proposed recipe recommendation system aims to provide users with personalized recipe suggestions based on the ingredients they input. This methodology outlines the design, implementation, and interaction of the front-end and back-end components of the system, leveraging machine learning for effective recipe generation.

3.1.2 System Design and Architecture

Framework Selection:

The front end is built using Streamlit, a Python library that allows for quick deployment of web applications. The back end employs deep learning frameworks, such as Flax, PyTorch, or TensorFlow, chosen based on performance requirements and compatibility with the recipe generation model.

3.1.3 Front-End Development

User Interface (UI):

- Develop a web form in app.py for users to input one or more ingredients.
- Implement input validation to ensure valid ingredient names are entered.
- Design the UI to include feedback mechanisms, such as loading indicators while recipes are being generated, enhancing user experience.

3.1.4 Backend Processing

Data Management:

- Scripts in the src folder will handle the incoming ingredient data.
- Preprocessing steps will include:
- Cleaning: Remove duplicate entries and standardize ingredient names for consistency.
- Tokenization: Convert the cleaned ingredient list into a model-compatible format through encoding.

3.1.5 Model Execution

Model Selection:

- The application selects the appropriate model based on user preferences or configuration settings.
- Recipe Generation:
  - Use generation.py to execute the selected model with tokenized ingredients.
  - Load the pre-trained model and pass the tokenized input to generate potential recipes.

3.1.6 Recipe Refinement

Post-Processing:

- Utilize prediction.py to refine model output, which includes:
    - Ranking recipes based on relevance and user preferences.
    - Filtering out recipes that do not meet specified dietary restrictions.

3.1.7 Recipe Output

Result Formatting:

- Format the generated recipes for presentation in the Streamlit interface:
    - Organize the recipe into sections such as ingredients, instructions, and additional information like preparation time and serving size.

Display to User:

- Send the formatted recipe back to the Streamlit app for user display, ensuring clarity and visual appeal.

3.1.8 User Interaction and Feedback

Interactive Features:

- Allow users to save their favorite recipes and request modifications based on additional ingredients.
- Implement a feedback mechanism to enable users to rate recipes, contributing to continuous improvement of the recommendation system.

3.1.9 Error Handling and Logging

Error Management:

- Implement robust error handling to manage invalid inputs or model execution failures.

Logging:

- Include logging functionalities to record user interactions and model performance data, aiding in system optimization and understanding user behavior.

Chapter 4: SYSTEM REQUIREMENTS & LIBRARIES

4.1 System Requirements:

Components	Requirement
Operating System	windows 10+,macOs,or any modern linux distribution
processor	Intel i5 or highest /AMD equivalent
RAM	Mini. 8gb
Storage	At least 1 GB free
Internet	required (for API calls to hugging face,Gemini,youtube)
Python version	python 3.9 or above
web browser	chrome , firefox, safari or edge

4.2 Libraries Used:

Library	Description
streamlit	frontend framework for building the web app UI
streamlit_tags	provides tag style multi-input for custom ingredient entry
transformers	hugging face library used foe loading and running the T% model
AutoTokenizer,AutoModelForSeq2SeqLM, pipeline	for recipe text generation tasks
googleapiclient.discovery	used to call the youtube data API for fetching video recommendation
google.generativeai	interface for google gemini API(for conversational chatbot functionality)
json	for reading ingredient configuration and storing user preference
textwrap	used for text formatting if needed

Chapter 5: SYSTEM ARCHITECTURE

5.1 System Architecture

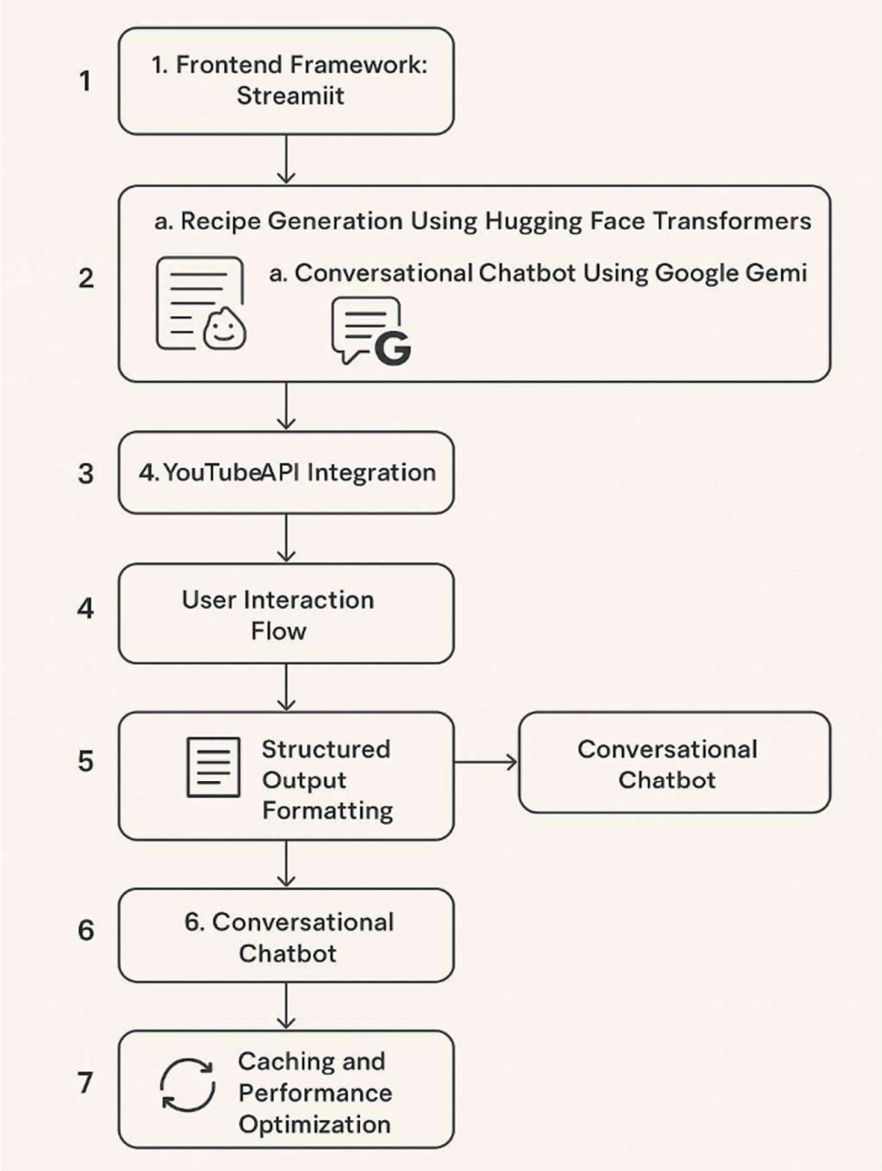


Fig. 5.1 (System Architecture Diagram)

A. Frontend Interface (Streamlit)

- Provides an intuitive and interactive UI.
- Allows:
  - Ingredient selection (dropdown, text input).
  - Dietary preferences, creativity level, recipe complexity (sliders & tags).
  - Direct interaction with the chatbot.
  - Visual display of the final recipe with formatting.

B. Recipe Generation Engine

- **Model Used**: T5 (flax-community/t5-recipe-generation) from Hugging Face.
- **Pipeline**: text2text-generation
- **Function**:
    - Receives user inputs (ingredients + preferences).
    - Outputs: a structured recipe with a title, ingredients list, and instructions.
    - Can dynamically alter results based on input complexity or mood data.

C. Conversational Chatbot

- **Powered by**: Google Gemini API (gemini-1.5-flash)
- **Functions**:
    - Interprets natural language queries related to cooking, nutrition, or substitutions.
    - Uses context from the generated recipe to provide precise, relevant answers.
    - Examples:
        - "How many calories is this?"
        - "What can I substitute mushrooms with?"
        - "Why does sauce split while cooking?"

D. YouTube Video Recommendation

- **API Used**: YouTube Data API v3
- **Functionality**:
    - Searches for cooking videos relevant to the generated recipe.
    - Uses keywords from ingredients, recipe title, and cooking method.
    - Displays a curated list of links for users to watch and follow visually.

E. Backend Processing & Utilities

- **Session Caching**:
    - Recipes and YouTube results are cached to avoid repetitive calls.
    - Ensures quick load times for repeated inputs during the same session.
- **Formatting Engine**:
    - Enhances recipe presentation using emojis, Markdown, and readable structures.
- **Validation & Input Processor**:
    - Ensures data cleanliness.
    - Formats ingredient lists, interprets sliders/tags into AI-readable prompts.

5.2. External Integrations

Service	Purpose
Hugging Face (T5)	Recipe generation via NLP
Google Gemini	Conversational AI for contextual Q&A
YouTube API	Display visual cooking support
Streamlit	Frontend rendering and state management

5.3 Benefits of the Architecture

- **Modularity**: Each component (chatbot, generator, video search) can be upgraded independently.
- **Scalability**: New models or data sources can be integrated with minimal impact.
- **User-Centric**: Real-time interaction, visual learning support, and intuitive UI.
- **Efficiency**: Caching and preprocessing reduce delays and API overload.

Chapter 6: DESIGN AND ANALYSIS

6.1 Workflow Of System

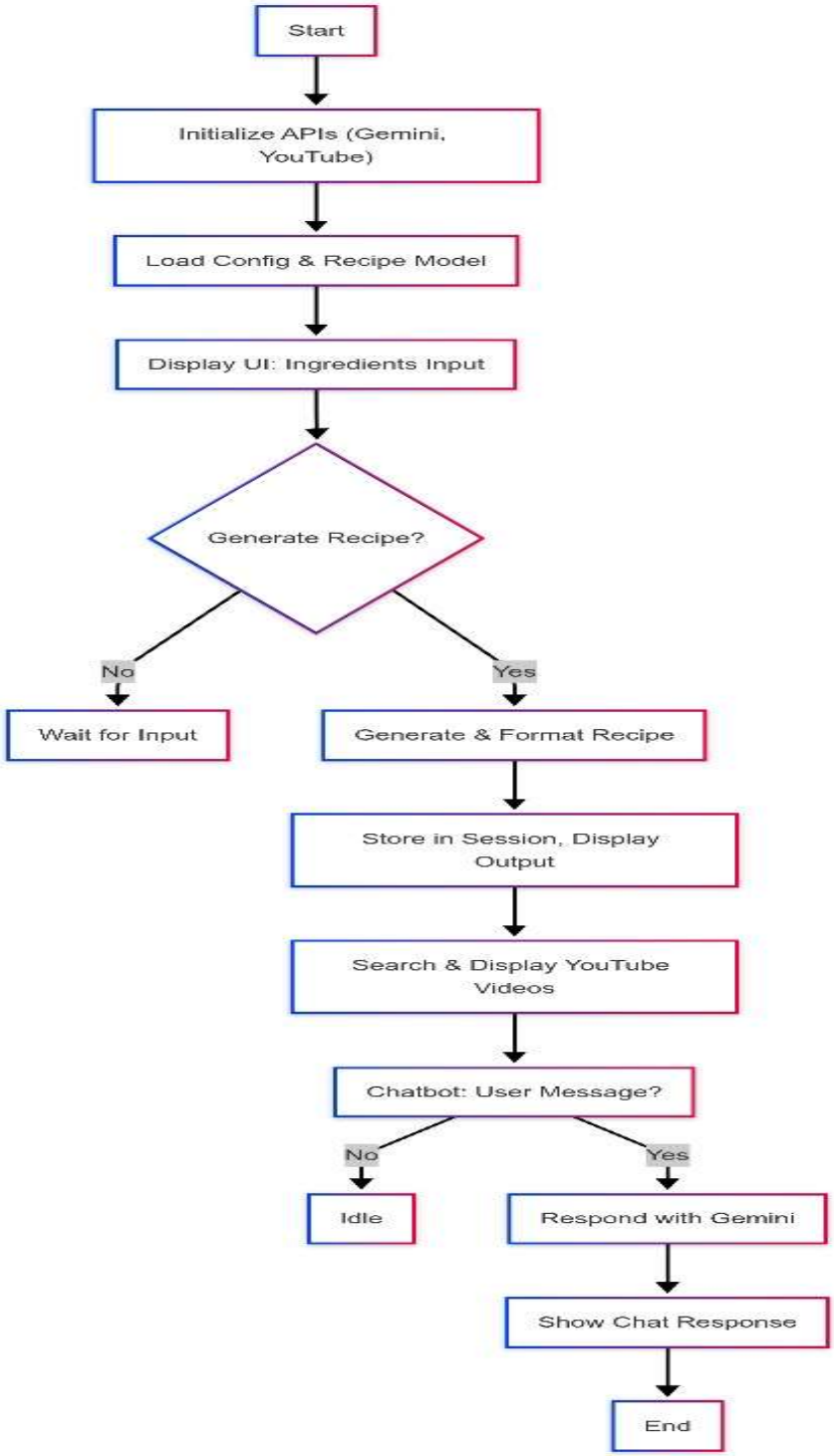


Fig 6.1 Workflow of system

6.1 Frontend User Interaction (Streamlit Interface)

The frontend is built using Streamlit, providing a user-friendly and interactive environment. Here's what happens in this layer:

a. User Input Collection

- **Ingredient Selection**: Users choose from a dropdown or type in available ingredients at home.
- **Preference Specification**:
  - **Dietary Preferences**: Vegan, gluten-free, low-carb, etc.
  - **Creativity Level**: From standard meals to highly innovative ones.
  - **Recipe Complexity**: Sliders allow users to specify how quick or elaborate the recipe should be.
- **Custom Tags**: Users can add further constraints or preferences (e.g., "no nuts", "festive", "comfort food").

b. Recipe Display

- After generation, the system presents:
  - **Recipe Title**
  - **Ingredients List** (cleanly separated)
  - **Cooking Instructions** (with emojis and headers for readability)

c. YouTube Integration

- The system fetches and displays related cooking videos based on the recipe or its ingredients.
- This is especially useful for beginners and visual learners.

d. Conversational Chatbot Sidebar

- Users can ask questions like:
  - “How many calories is this?”
  - “Can I replace milk with coconut cream?”
- The chatbot responds contextually using Google Gemini and remembers the current recipe.

6.2 Backend Logic and Processing

This is where the real magic happens—turning raw input into meaningful outputs.

a. Input Processor

- Cleans and validates all user inputs.
- Formats the data into a structured prompt suitable for AI models.
- Identifies keywords for YouTube search and chatbot context.

b. Recipe Generator (Hugging Face T5)

- Model: flax-community/t5-recipe-generation
- Task: Text-to-text generation
- Input: Selected ingredients + preferences
- Output: Full recipe including:
    - Title
    - Ingredients
    - Directions (step-by-step)

c. Chatbot Handler (Google Gemini)

- Receives questions from the sidebar.
- Uses the context of the generated recipe to return personalized answers
- Capable of understanding casual, conversational language.

d. YouTube Search Module

- Uses keywords extracted from the recipe (dish name, main ingredients).
- Sends queries to YouTube Data API.
- Retrieves links to high-relevance cooking videos.

e. Structured Output Formatter

- Formats recipe with:
    - Emojis
    - Markdown styling
    - Bullet points for easy reading
    - Consistent structure for titles, ingredients, and steps

6.3 Caching and Performance Optimization

a. Model Caching

- The T5 model is loaded once per session to speed up future interactions.

b. YouTube Result Caching

- YouTube API results are cached for one hour to reduce redundant API calls and speed up the UI.

Chapter 7: RESULTS AND DISCUSSION

7.1Testing Strategies

A comprehensive testing strategy was adopted for Cravify to ensure system stability, accuracy of recipe generation, conversational reliability, and user satisfaction. The strategy incorporates unit testing, integration testing, UI testing, and user-based evaluation. Testing was performed across all core modules and workflows to validate both functional and non-functional aspects of the application.

7.1.1 Unit testing

Components	Test Case Description
format_reciept()	Validates correct parsing of generated recipe text into markdown format
search_youtube()	ensure youtube API returns valid video URLs and handles errors gracefuly
Gemini prompt generation	confirms that user queries are correctly merged with contextual recipe information
T5 generation pipeline	validates model inference with mock ingredient input for output structure consistency.

7.1.2 Integration Testing

Integration Point	Objective
T5 → Format → Display	ensures that generates recipe text is processed and rendered accurately
Gemini API + Chat Interface	Validates chat input context is passed correctly and meaningful responses returned
Ingredient UI → Recipe Gen	Confirms selected/multiple ingredients are correctly passed into the generation model
YouTube API + Ingredient Tag	Ensures relevant videos appear for each unique recipe search

7.1.3 UI and UX Testing (Frontend)

Area	Test focus
Multi-select & tags	check dynamic ingredient addition and deletion
sliders	Validate that slider changes affect model output length and temperature
Button behavior	Confirm "Generate Recipe" triggers loading spinner and displays results
video rendering	Ensure embedded YouTube player loads properly within all major browsers
sider chatbot	Test real-time chatbot input/output cycle and conversation updates

7.1.4 Performance & caching validation

Test type	Objective
model caching test	ensure the T5 model is loaded only once per session
Youtube API caching	confirm that video search results are cached for one hour as expected
API rate handling	simulate multiple queries to test rate limit handling and fallback behavior

7.2Expected Results

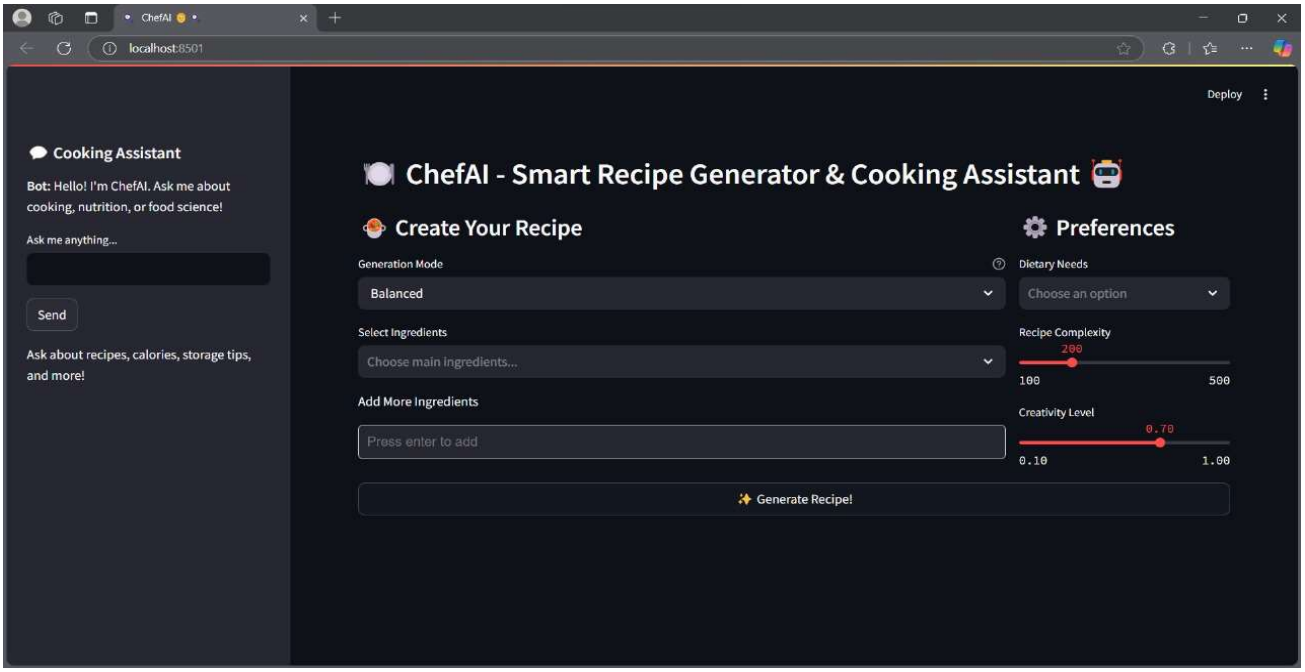


Fig7.2.1. Ingredient & Preference Input Screen

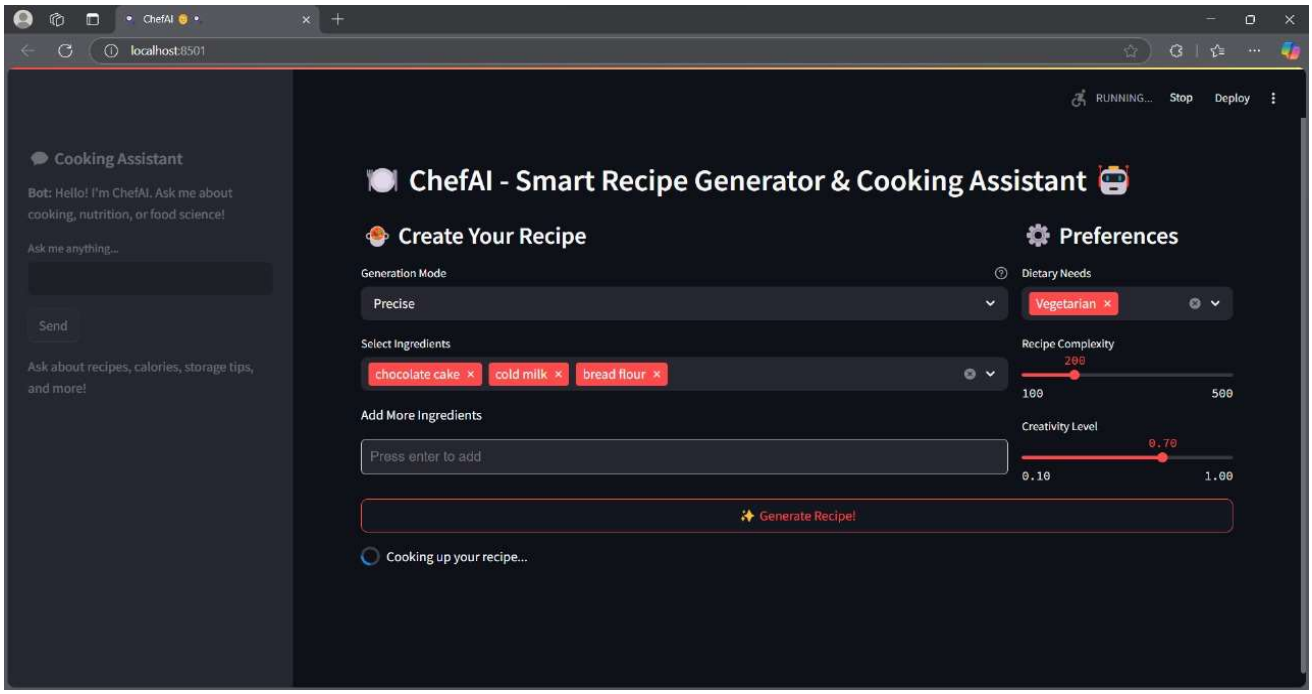


Fig. 7.2.2. Recipe Generation Based on Inputs

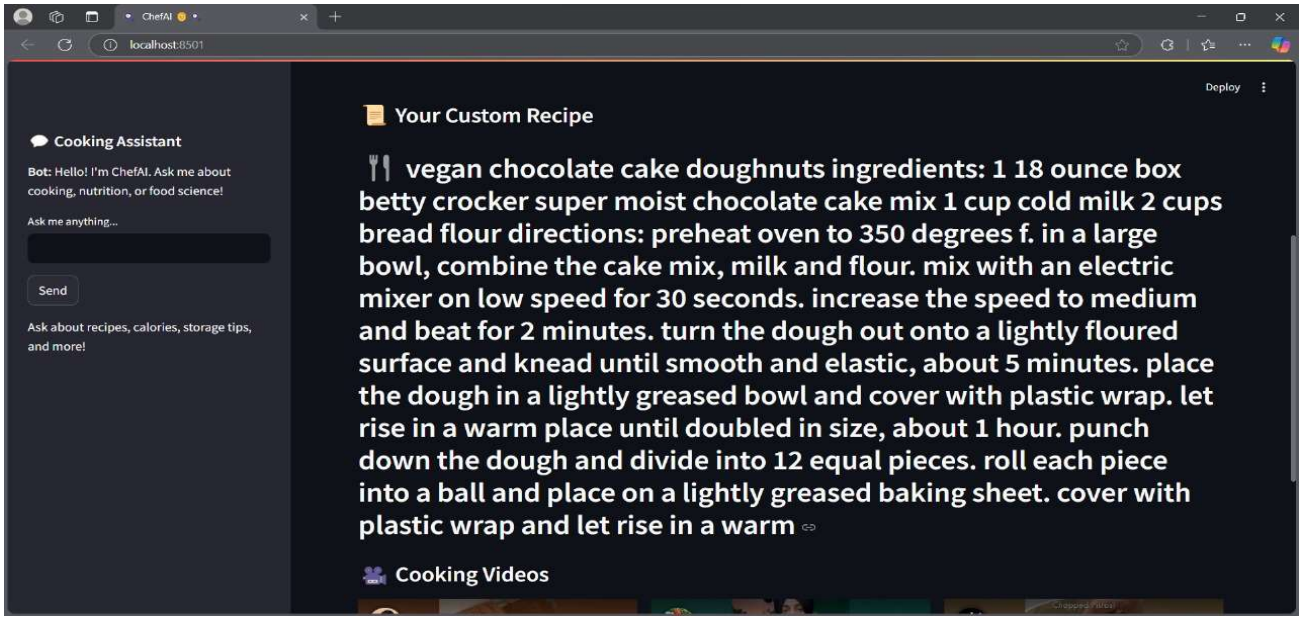


Fig. 7.2.3. Recipe Display with Details

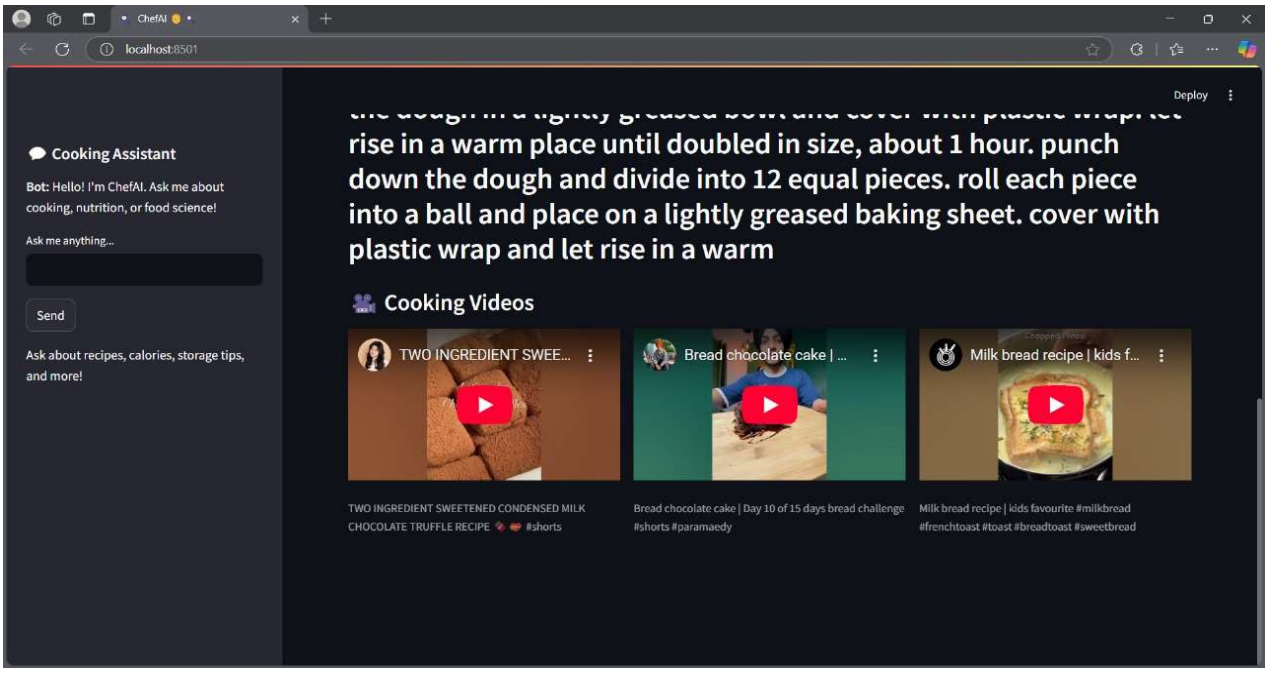


Fig. 7.2.4. YouTube Video Suggestions

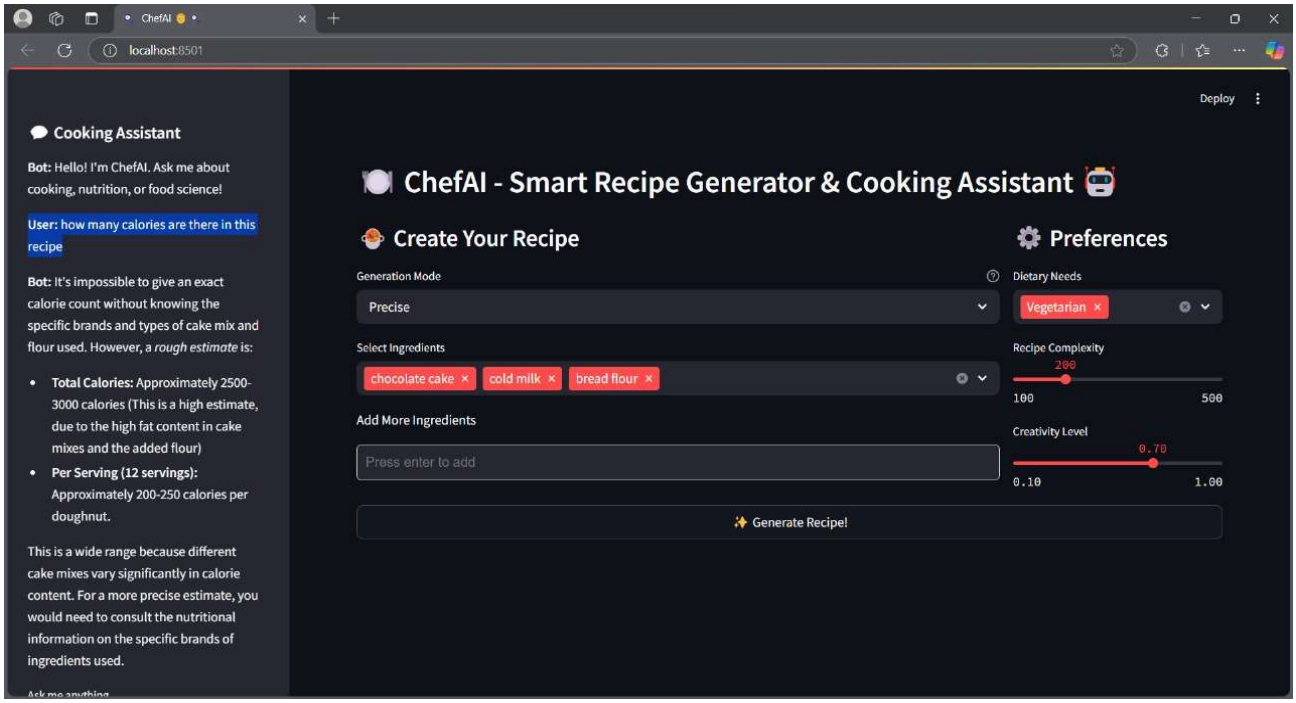


Fig. 7.2.5. Chatbot Assistance with Inquiries

Chapter 8: FUTURE SCOPE

8.1Future Scope

While Cravify already offers a significant leap in personalized and emotionally intelligent recipe recommendation, several avenues exist for future expansion and innovation:

1. **Smart Wearable Integration**
   By connecting with fitness trackers or smartwatches, Cravify could adapt recipes based on real-time health data such as stress levels, heart rate, or activity intensity—offering meals suited to the user's physical and mental state.
2. **Multilingual and caltiral expansion**
   Adding multilingual support and culturally diverse recipe generation would broaden the system's inclusivity, allowing users from different regions and backgrounds to access localized and traditional cuisines with contextual relevance.
3. **Augmented Reality (AR) Cooling Assistance**
   Future versions may incorporate AR features for real-time visual overlays of cooking steps, helping users follow recipes with better precision and engagement in hands-free mode.
4. **Ingredient Recognition via image analysis**
   Cravify could be enhanced with computer vision models to identify available ingredients via camera input, making the ingredient entry process even more intuitive and frictionless.
5. **User profiles and progress tracking**
   Implementing personalized user profiles that save preferences, dietary goals, and cooking history would allow the system to offer tailored tips, suggest weekly plans, and track culinary growth over time.
6. **Enhanced nutritional coaching**
   Integration with nutritional databases and AI dietitians could allow Cravify to recommend meals aligned with medical goals—such as diabetes management, weight loss, or muscle gain—through evidence-based dietary adjustments.

Chapter 9:CONCLUSION

Conclusion:

Cravify redefines what a recipe recommendation system can be—transforming it from a passive tool into an active, emotionally aware cooking companion. By leveraging cutting-edge AI models, multimodal inputs, and conversational interfaces, the system personalizes every aspect of the cooking experience to suit the user's needs, mood, and context. It not only helps users cook with what they have but does so in a way that feels intuitive, responsive, and genuinely helpful.

Whether assisting a beginner trying their first dish or supporting a health-conscious individual balancing nutrition and time, Cravify adapts in real-time to deliver culinary experiences that are both delightful and purposeful. With its strong foundation and vast potential for growth, Cravify stands at the frontier of AI-driven kitchen technology—offering a glimpse into a future where cooking is not just smarter, but deeply human-centric.

As the food-tech landscape evolves, Cravify's ability to integrate with voice assistants, wearable health devices, and smart kitchen appliances paves the way for seamless, hands-free, and holistic culinary ecosystems. It doesn't just answer the age-old question of "What's for dinner?"—it understands the why, how, and when behind it. Its empathetic design, intelligent recommendations, and real-time adaptability create a bridge between nourishment and innovation, bringing users not only closer to better meals, but to better living.

In essence, Cravify is more than just a project—it is a vision of the future kitchen: intelligent, inclusive, and emotionally intelligent. With ongoing enhancements and user-centric innovation, Cravify has the power to inspire creativity, support wellness, and transform everyday cooking into a uniquely personal journey.

REFERENCES

References:

[1] W. Min, S. Jiang, and R. Jain, "Food Recommendation: Framework, Existing Solutions, and Challenges," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2659–2671, Oct. 2020.

[2] M. Gim, S. Park, and C. D. Yoo, "RecipeBowl: A Cooking Recommender for Ingredients and Recipes Using Set Transformer," *IEEE Access*, vol. 9, pp. 16084–16097, 2021.

[3] M. Chen, B. Dhingra, and W. Cohen, "Eating Healthier: Exploring Nutrition Information for Healthier Recipe Recommendation," *Information Processing & Management*, vol.57,no.6,p.102382,2020.

[4] J. M. Ordovas, L. R. Ferguson, E. S. Tai, and J. C. Mathers, "Personalised Nutrition and Health," *BMJ*, vol. 361, p.k2173,2018.

[5] R. Mate, M. Kadam, and N. Pawar, "Automated Recipe Chatbot," *International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)*, Dec. 2024.

[6] D. J. K. Patel, "An Artificial Intelligence Interactive Platform for Automated Chatbot with AI-Driven Innovation in Recipe Searching," *Mathematics and Computer Science*, 2024.Mehta, Prateek. (2016). Creating Google Chrome Extensions. 10.1007/978-1-4842-1775-7.

[7] Salvador, A., Hynes, N., Aytar, Y., Ofli, F., Weber, I., & Torralba, A. (2019). "Inverse Cooking: Recipe Generation from Food Images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10453–10462.

[8] Yang, K., & Lee, D. (2018). "NutRec: Nutrition-oriented Food Recommendation with Reinforcement Learning." International Conference on Health Informatics.

[9] Kusmierczyk, T., Trattner, C., & Nørvåg, K. (2020). "Temporal Patterns in Online Food Innovation: A Study of Recipe Sharing Platforms." International Journal of Information Management, 50, pp. 36–49.

[10] Teng, C. Y., Lin, Y. R., & Adamic, L. A. (2012). "Recipe Recommendation Using Ingredient Networks." Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307.

[11] Majumder, B., Li, S., & Balakrishnan, R. (2020). "Generating Personalized Recipes from Historical User Preferences." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

Appendix A

Core Code Snippets of Cravify

```python
import streamlit as st
from transformers import AutoTokenizer, AutoModelForSeq2SeqLM, pipeline
from googleapiclient.discovery import build
from streamlit_tags import st_tags
import google.generativeai as genai
import json
import textwrap

# 🔹 Configure Streamlit Page
st.set_page_config(layout="wide", page_title="ChefAI 👨‍🍳🔍", page_icon="🔍")

# 🔹 Set up Gemini API
GEMINI_API_KEY = "AIzaSyDbSrbKPbNK-sb3sx5tXMoPlN30Q3_P5o8"
genai.configure(api_key=GEMINI_API_KEY)

# 🔹 YouTube API Key
YOUTUBE_API_KEY = "AIzaSyC_mVSKJhciP_WXMe2PcUSEC4BoMHPfTXY"
```

```python
# 🔹 YouTube Search with Caching
@st.cache_data(ttl=3600)
def search_youtube(query, max_results=3):
    try:
        youtube = build("youtube", "v3", developerKey=YOUTUBE_API_KEY)
        request = youtube.search().list(
            part="snippet",
            q=query,
            type="video",
            maxResults=max_results,
            relevanceLanguage="en"
        )
        response = request.execute()
        return [{
            "title": item["snippet"]["title"],
            "url": f"https://youtu.be/{item['id']['videoId']}"
        } for item in response.get("items", [])]
    except Exception as e:
        st.error(f"Error searching YouTube: {str(e)}")
        return []
```

```python
# 🔹 Recipe Formatting
def format_recipe(raw_text):
    try:
        sections = raw_text.split('\n')
        formatted = ""
        for section in sections:
            if section.startswith('title:'):
                formatted += f"## 🍴 {section[6:].strip()}\n\n"
            elif section.startswith('ingredients:'):
                items = [x.strip() for x in section[12:].split(';') if x.strip()]
                formatted += "### 📝 Ingredients\n" + "\n".join(f"- {item}" for item in items) + "\n\n"
            elif section.startswith('directions:'):
                steps = [x.strip() for x in section[11:].split(';') if x.strip()]
                formatted += "### 👨‍🍳🔍 Instructions\n" + "\n".join(
                    f"{i+1}. {step}" for i, step in enumerate(steps)
                ) + "\n\n"
        return formatted
    except:
        return raw_text
```

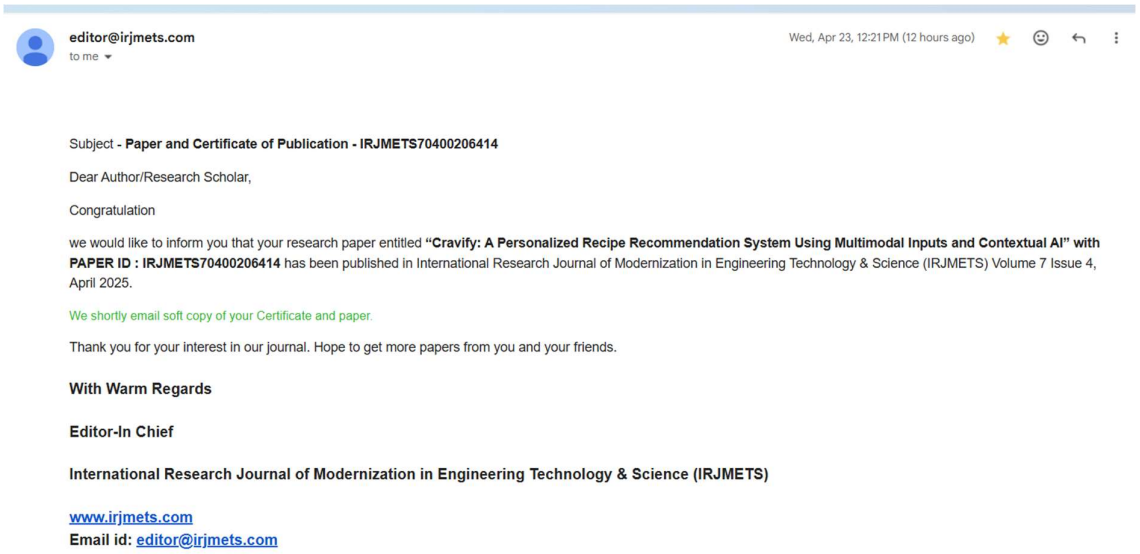
```python
# 🔹 Load Models
@st.cache_resource
def load_models():
    tokenizer = AutoTokenizer.from_pretrained("flax-community/t5-recipe-generation")
    model = AutoModelForSeq2SeqLM.from_pretrained("flax-community/t5-recipe-generation")
    return pipeline("text2text-generation", model=model, tokenizer=tokenizer), genai.GenerativeModel("gemini-1.5-flash")

generator, gemini = load_models()

# 🔹 Load Ingredients
with open("config.json") as f:
    cfg = json.load(f)

# 🔹 Main Interface
st.header("🍽️ ChefAI - Smart Recipe Generator & Cooking Assistant 🤖")
```

```python
# 🔹 Recipe Generation Section
with st.container():
    col1, col2 = st.columns([3, 1])

    with col1:
        st.subheader("🔥 Create Your Recipe")
        sampling_mode = st.selectbox("Generation Mode", ["Balanced", "Creative", "Precise"],
                                     help="Control how creative the recipe generation should be")
        original_keywords = st.multiselect("Select Ingredients", cfg["first_100"], [],
                                           placeholder="Choose main ingredients...")
        custom_keywords = st_tags(label="Add More Ingredients", text='Press enter to add',
                                  suggestions=cfg["next_100"], maxtags=15)

    with col2:
        st.subheader("⚙️ Preferences")
        dietary_restrictions = st.multiselect("Dietary Needs", ["Vegetarian", "Vegan", "Gluten-Free"])
        max_length = st.slider("Recipe Complexity", 100, 500, 200)
        temperature = st.slider("Creativity Level", 0.1, 1.0, 0.7)

    all_ingredients = ", ".join(original_keywords + custom_keywords)

    if st.button("✨ Generate Recipe!", use_container_width=True):
        if not all_ingredients:
            st.error("Please select at least one ingredient!")
            st.stop()

        with st.spinner("Cooking up your recipe..."):
            try:
                generated = generator(
                    f"{dietary_restrictions} {all_ingredients}",
                    max_length=max_length,
                    temperature=temperature,
                    num_return_sequences=1
```

```python
                )
                raw_recipe = generated[0]['generated_text']
                formatted_recipe = format_recipe(raw_recipe)

                # Store context
                st.session_state["last_recipe"] = raw_recipe
                st.session_state["last_ingredients"] = all_ingredients

                # Display recipe
                st.subheader("📝 Your Custom Recipe")
                st.markdown(formatted_recipe)

                # YouTube Videos
                st.subheader("🎥 Cooking Videos")
                videos = search_youtube(f"{all_ingredients} recipe")
                if videos:
                    cols = st.columns(len(videos))
                    for idx, col in enumerate(cols):
                        with col:
                            st.video(videos[idx]["url"])
                            st.caption(videos[idx]["title"])
                else:
                    st.info("No videos found. Try different ingredients.")

            except Exception as e:
                st.error(f"Recipe generation failed: {str(e)}")
```

```python
# 🔹 Food Chatbot
st.sidebar.header("💬 Cooking Assistant")

if "messages" not in st.session_state:
    st.session_state.messages = [{"role": "bot", "content": "Hello! I'm ChefAI. Ask me about cooking, nutrition, or food science!"}]

# Display chat history
for message in st.session_state.messages[-5:]:
    st.sidebar.markdown(f"**{message['role'].capitalize()}:** {message['content']}")

# Chatbot Input
chat_input = st.sidebar.text_input("Ask me anything...", key="chat_input")

if st.sidebar.button("Send", key="chat_send"):
    if chat_input:
        st.session_state.messages.append({"role": "user", "content": chat_input})

        # Include Recipe Context for Better Answers
        additional_context = ""
        if "last_recipe" in st.session_state and "last_ingredients" in st.session_state:
            additional_context = f"\nHere is the recipe and ingredients: {st.session_state['last_ingredients']}\n{st.session_state['last_recipe']}\n"

        # **Improved Prompt for Calorie & Serving Size Questions**
        if "calories" in chat_input.lower():
            prompt = f"""
            Estimate the total and per-serving calories for this recipe.
            Assume standard ingredient weights and provide a reasonable estimate.

            Recipe Ingredients:
            {st.session_state.get("last_ingredients", "Unknown")}

            Instructions:
            {st.session_state.get("last_recipe", "Unknown")}

            Keep the answer concise and useful.
            """
        elif "servings" in chat_input.lower():
            prompt = f"""
            Estimate how many servings this recipe makes based on standard portion sizes.

            Recipe Ingredients:
            {st.session_state.get("last_ingredients", "Unknown")}

            Instructions:
            {st.session_state.get("last_recipe", "Unknown")}

            Provide a reasonable estimate with common portion sizes.
            """
        else:
            prompt = f"{additional_context}\n{chat_input}"

        # Get Response from Gemini
        with st.spinner("Thinking..."):
            response = gemini.generate_content(prompt)
            bot_response = response.text if response else "Sorry, I couldn't find an answer."

        st.session_state.messages.append({"role": "bot", "content": bot_response})
        st.rerun()

st.sidebar.write("Ask about recipes, calories, storage tips, and more!")
```

APPENDIX B



editor@irjmets.com
to me

Wed, Apr 23, 12:21 PM (12 hours ago)

Subject - **Paper and Certificate of Publication - IRJMETS70400206414**

Dear Author/Research Scholar,

Congratulation

we would like to inform you that your research paper entitled **"Cravify: A Personalized Recipe Recommendation System Using Multimodal Inputs and Contextual AI" with PAPER ID : IRJMETS70400206414** has been published in International Research Journal of Modernization in Engineering Technology & Science (IRJMETS) Volume 7 Issue 4, April 2025.

We shortly email soft copy of your Certificate and paper.

Thank you for your interest in our journal. Hope to get more papers from you and your friends.

With Warm Regards

Editor-In Chief

International Research Journal of Modernization in Engineering Technology & Science (IRJMETS)

www.irjmets.com
Email id: editor@irjmets.com

Figure D.1: Official Research Paper Acceptance Confirmation

The above screenshot displays the official acceptance email from the *International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS)*, confirming the selection of our research paper titled "Cravify: An AI-Driven Emotionally Intelligent Recipe Recommendation System."

This acceptance affirms the academic and practical relevance of our project, recognizing its innovative approach to AI-powered recipe personalization. The paper is scheduled for publication in the upcoming volume of IRJMETS, a reputed journal known for fostering interdisciplinary research in modern engineering and technology.

The publication marks a significant milestone in the Cravify journey—validating our research efforts and contributing to the ongoing dialogue in intelligent systems and user-centric AI applications.

B.1 Research Paper Publication and Certificates

Published Research Paper
Our research paper titled *"Cravify: An AI-Driven Emotionally Intelligent Recipe Recommendation System"* has been officially published in the **International Research Journal of Modernization in Engineering, Technology and Science (IRJMETS)**.

- **Publication Link:**
  file:///C:/Users/KASTURI/Downloads/IRJMETS70400206414.pdf

- **Journal Details:**

- Journal: IRJMETS
- ISSN: 2582-5208
- Volume: 07/Issue: 04, Year: 2025
- Indexing: Google Scholar, Academia, Crossref, etc.

B.2 Certificates of Publication and Presentation

The following certificates were awarded upon successful publication and contribution:

1. **Certificate of Publication** – Issued by IRJMETS for the official publication of our paper.
2. **Certificate of Contribution/Presentation** – Acknowledging the research contribution and authorship of the team.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI

Deven Deshmukh[*1], Kasturi D Dhopat[*2], Ankita Darmal[*3], Isha Kadam[*4], Prof. Vidya Bharde[*5]

[*1,2,3,4]Students, Department Of Computer Engineering, Mahatma Gandhi Mission's College Of Engineering & Technology, Kamothe, Navi Mumbai, Maharashtra, India.

[*5]Assistant Professor, Department Of Computer Engineering, Mahatma Gandhi Mission's College Of Engineering & Technology, Kamothe, Navi Mumbai, Maharashtra, India.

ABSTRACT

Cravify is the personalized cooking assistant powered by AI that recommends adaptive recipes based on ingredients, mood, preferences, and dietary needs of the user. Cravify differs from traditional recipe databases by integrating multimodal inputs using text, voice, images, and using the emotional context through a mood-aware context using natural language processing (NLP) and Google Gemini. This paper describes a modular system that includes the use of the T5 model for generating recipes, a Gemini-powered conversational agent, and a YouTube API for getting visual instructions. Combined, these features turn recipe discovery into an interactive experience that evokes emotional engagement and helps the user make better food choices aligned with current needs and available resources.

Keywords: Personalized Recipe Assistant, Conversational AI, Cooking With NLP, Food Technology, Interactive Meal Planning.

I. INTRODUCTION

In a time of technology embedded in every aspect of our lives, how we search and cook is changing fast. In this new environment, recipe suggestion systems that are only ingredient- or nutrition-based are useful, but they seldom involve more emotional, situational, or motivational components of cooking. Cravify offers a school of thought with an interactive chatbot recipe suggestion mechanism, so the users are not merely treated as bunches of ingredients, but also assessed as humans according to their feelings, wishes, and energy or time.

Though there are already tools like RecipeBowl [2] or older meal planning websites [1][4] being advanced in ingredient or nutritional suggestion. They have a lack of context sense, generally guiding users to submit rigid search requests in order to discover meals; a perpetrator elaborated upon within current literature addressing hindrances toward efficient meal planning [1][3].

Cravify offers a recognizable solution to the constraint posed by such sites in crafting a deal meal; Cravify converses in human speech, conducts user preference with mood conditioned logic, and then provides suggestions in relation to both what was talked about and the way the user talked about it. With objectives to be friendlier than utility-like [5][6], Cravify will yield user suggestions that won't seem like search queries, but instead, conversing with a long-time friend.

The structure of this paper is outlined below:

Part II explores prior research, Part III outlines our methodology, Part IV details the system development, Part V evaluates the results, Part VI concludes paper, and Part VII provides the source references.

A. Problem Definition

In the current world of speed, it is most of the time a time-consuming problem to determine what to prepare—particularly when individuals have minimal ingredients at home. Most recipe websites and applications compel users to search according to predefined recipes, sometimes requiring them to use other ingredients they don't possess. This results in frustration, food wastage, and reliance on external sources to plan meals.

Also, modern recipe applications do not typically provide personalized filters like dietary options (e.g., veg, non-veg, or vegan), calorie restrictions, or explicit video tutorials. There is also no interactive assistance when the user needs answers or quick clarifications.

There is an evident need for a smart, all-in-one platform that can provide recipe suggestions according to

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

ingredients available, align user preferences, and ensure a smooth cooking experience.

B. Objectives of the Study:

To To address these limitations, we have developed **Cravify**, an AI-powered recipe recommendation tool.

1. Users can enter the ingredients they have, and our model will suggest recipes that can be made using those items.

2. Users can filter results based on dietary preferences and get calorie info, video tutorials, and chatbot assistance for a smoother cooking experience.

II. LITERATURE REVIEW

Recipe recommendation systems have evolved from basic digital cookbooks to AI-powered kitchen helpers. Previous models could only give recommendations if users had every ingredient required exactly as asked. Newer recipes recommendation systems have the ability via models like Set Transformers to understand the relationships of flavor and texture between ingredients. Some researchers even used the T5 language model to improve flexibility and creativity by adapting recipes based on if the user currently has generic ingredients in the food or kitchen rather than existing as strict recipes.

Another exciting area for development and research is dietary awareness. Research has demonstrated that people are healthier when they understand what is in their food. Today's AI systems have the capability to immediately answer specific dietary questions, such as "will this recipe spike my blood sugar?" or "how do I add more protein to this meal?" Rather than just providing a number, these tools explain how the recipes meet particular nutritional needs and, therefore, feel less like many recipe apps and more like a personal nutritionist!

Personalization has also had a major shift.

Older systems treated every customer the same way, but newer technology understands everyone consumes food differently. AI chatbots now learn users' tastes, whether someone is a vegan, a nut allergy sufferer, or simply doesn't like cilantro, and tailor suggestions based on it. It's a more considered, bespoke experience that comes a lot closer to having a chef who truly understands your palate.

Older digital cooking tools felt rigid. You had to phrase the query just right to get anything meaningful back. Over the last few years though, researchers have focused on improving natural language understanding. Users can now ask things like "What can I make with leftover rice and eggs?" or "How do I fix a runny sauce?" and get back intelligent and useful responses. This change has helped to make cooking tools feel more like a helpful friend and less like a computer.

Interface design has also gotten better. Some of the older systems were implemented as browser extensions, but the philosophy was to simplify cooking, particularly when users are multitasking in the kitchen. New tools are designed with simplicity and responsiveness in mind.

If you're using a phone in the supermarket or a tablet while cooking, contemporary interfaces crafted with tools such as Streamlit are centered on being easy to use even when your hands are dirty.

As a reaction to these changing requirements, Cravify was created as an AI-based recipe recommendation platform that puts it all into one place. It enables users to enter ingredients they have and recommend recipes that they can cook immediately. Apart from that, it features dietary preference filters, calorie details, associated YouTube tutorials, and even a chatbot that assists users with cooking-related queries on the move.

It is intended to make cooking more convenient, minimize food waste, and restore confidence in the kitchen even for novices.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

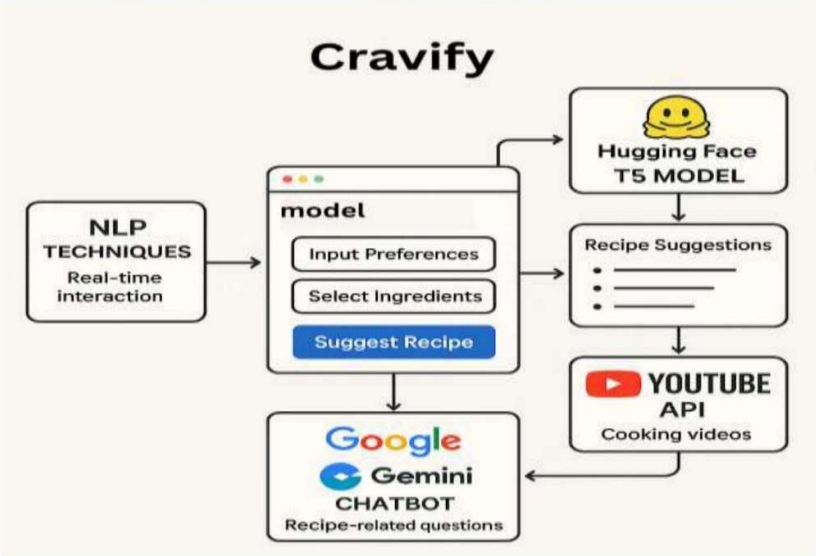


Fig 2.1: Workflow of the Cravify recipe recommendation system, from user input to recipe suggestion generation using AI models.

S.R. No.	Author	Approach	Strengths	Weakness
1	Min et al. (2020)	Framework-based food recommender	Covers wide range of existing food recommendation systems and challenges	General overview, lacks implementation specifics
2	Gim et al. (2021)	Set Transformer for ingredient-recipe matching	Handles unordered ingredient sets effectively; improves relevance in suggestions	High computational complexity and model training time
3	Chen et al. (2020)	Nutrition-aware recipe recommendation	Integrates nutrition data to promote healthier food choices	Limited personalization, doesn't fully adapt to user preferences
4	Ordovas et al. (2018)	Personalized nutrition approach	Focuses on user-specific health data and goals	Requires access to detailed personal health data
5	Mate et al. (2024)	Automated recipe chatbot	Provides quick, user-friendly responses; easy interface	Limited to predefined recipe dataset, lacks deep contextual reasoning
6	Patel (2024)	AI-driven chatbot with innovation in recipe search	Enhances interactivity and search flexibility	May require ongoing improvements in NLP understanding

Fig 2.2: Comparative analysis of various knowledge- grounded conversational models based on their approach, strengths, and weaknesses.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

III. METHODOLOGY

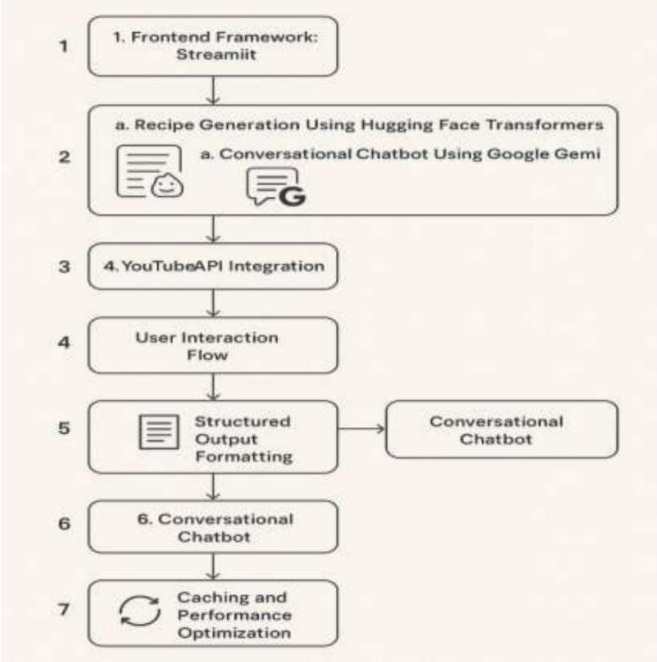


Fig 3.1: System Architecture

A. System Architecture and Overview:

Cravify is a modular system that ensures flexibility as it grows. Here's how that all comes together:

1. Frontend Framework – Streamlit

Streamlit is the major component used to create the frontend. This allows users the most fluid, intuitively interactive experience as possible. Some important features of Streamlit are: a. Ingredient selection - Users can select ingredients using a simple dropdown file.

b. Dietary preferences - multi-select allows users to define dietary needs so that the recipes are consistent with their preferences.

c. Recipe complexity & creativity - Sliders allow users to select levels of detail or creativity in any recipe.

d. User preferences - The system can be managed with additional fine-tuning of cooking experience with a tag.

2. Model Integration

Recipe Generation with Hugging Face Transformers Cravify uses the T5 model from Hugging Face to generate recipes. Here's how it works:

i. Users provide the ingredients and dietary preferences.

ii. The model generates a recipe with a title, and ingredients, and provides users with stepwise instructions.

iii. Pipeline: text2text-generation

iv. Task: Create recipes based on user inputs and dietary restrictions.

3. Conversational Chatbot using Google Gemini

The conversational chatbot that is built into the application is powered by Google Gemini. This chatbot is designed to be responsive to natural language queries pertaining to the recipes in the app, including, but not limited to:

How many calories are in this recipe?

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

What is the serving size?

What is the science behind the cooking technique?

The chatbot utilizes the recipe context to form helpful and accurate responses.

4. YouTube API integration

Cravify also integrates the YouTube API, so the user can find cooking videos that reference the recipes that are generated. The user can view a visual guide for following instructions from the recipe in a step-by-step fashion and facilitate the cooking process.

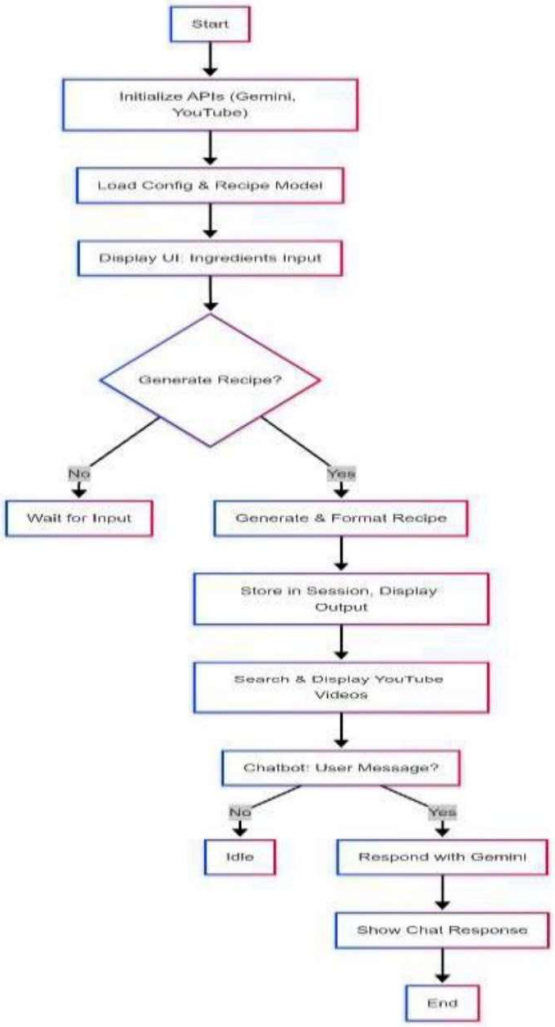
5. User Interaction Flow

The interaction flow is designed to be intuitive and simple:

Users either select ingredients from a likely list or enter their own.

Users can specify dietary needs, the level of recipe complexity, and the degree of creativity.

Once everything is set, they click "Generate Recipe" to retrieve their personalized recipe.



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

6. Structured Output Formatting

The recipe is displayed to the user using friendly structured formatting:

Title: name of the dish.

Ingredients: listing of ingredients separated by semicolons.

Directions: step-by-step instructions on how to cook with semicolons separating distinct steps, with navigational emojis and bold headings for faster scanning.

7. Conversational Chatbot

In an unobtrusive sidebar, the chatbot:

Greets users and accepts user questions in natural language.

Delivers contextual responses, such as estimated calories or number of servings, based on the recipe.

8. Caching and Performance Optimization

To deliver a smooth experience, some caching methods are in use:

b. YouTube Results Caching: YouTube search results are cached for one hour to prevent needless API calls.

c. Recipe Model Caching: The recipe generation model is cached for the duration of the session, which means subsequent requests for recipes will respond faster.

B. Work Flow of the Extension.

1. User Interface (Frontend):

Here's where people get to play around with Cravify and see all things work their magic. It is actually very user-friendly, so it is simple and a pleasure to use throughout.

a. User Input (Ingredients):

Users just click or select from a list of ingredients they currently have in their home. They can also indicate such as diet preference (consider vegan, gluten-free), experimentation level (be more imaginative!), and recipe complexity. All done with lovely and straightforward dropdowns, sliders, and tags. It is so easy anyone can do it.

b. Display Recipe:

Once the recipe is done, it's displayed to the user in a simple and friendly manner. The title is bold, ingredients are legible, and the step-by-step instructions are prettily formatted with emojis and headings. It's basically a cookbook, minus being completely awesome.

c. YouTube Video Results

And just for good measure, Cravify provides you with YouTube videos that are pertinent to the recipe or ingredient. So if you want to see a recipe being made, you can see it come to life with a

video. Extremely convenient, particularly if you're

Fig. 3.2. Flow Diagram

a visual learner!

d. Chatbot Interface:

To the side, there's a resident chatbot that's always present. You can query it, for instance, "How many calories are this?" or "Can I swap out this ingredient?" It answers in real-time and improves its answers based on which recipe you're working with, so the experience is personalized and smart.

2. Backend Logic

Now, this is where the magic of technology comes into play! It's like the behind-the-scenes brains, keeping things going smoothly and delivering users exactly what they require.

a. Input Processor:

This is where Cravify processes the user's ingredients and their preferences and prepares them for the rest of the process. It's like the system's "thinking" process, where it ensures everything gets to its destination.

b. Recipe Generator (Gemini / T5 Model):

Depending on the user input, Cravify's T5 (Hugging Face's) AI model performs its magic and creates a tailored recipe. It displays the recipe in a structured format with a fun name, ingredients in a list, and easy step-by-step

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

instructions that anyone can easily follow.

This module teaches you everything that during a cooking session. It reminds the chatbot what dish you're talking about, so when you do ask questions, it's always relevant to what meal you're preparing.

c. YouTube Search Module:

It employs the use of keywords within the dish and searches for YouTube videos by those keywords referring to the dish. Therefore, if you need to make a tutorial do its thing for you, it is at your disposal, awaiting to guide you through it!

d. Chatbot Handler

When you ask the chatbot a question, this module makes sure it receives what you're asking. It then works in tandem with the Gemini API to retrieve the right answers, so the chatbot's answers are always on point and recipe-based.

3. External Services (APIs)

Cravify isn't doing all this by itself—it's got some pretty smart external services assisting it to drive everything:

a. Google Gemini API:

This is the magic sauce that makes the chatbot so smart! It allows the bot to be context-aware of the recipe you're following, so when you ask something like "Is this gluten-free?" or "How many calories is this?", it can provide you with proper, thoughtful responses.

b. YouTube API:

This is the API that downloads all of those cooking videos. It searches for recipe-based tutorials and plays them, and you can learn how to cook the meal by watching actual cooks make the meal.

IV. TOOLS & TECHNIQUES

A. Streamlit: For building the interactive frontend interface.

B. Hugging Face Transformers: For recipe generation using the T5 model.

C. Google Gemini: For providing a conversational chatbot and answering user queries.

D. YouTube API: For fetching cooking videos related to the recipe.

E. JSON: For storing ingredient configurations and user

Preference.

V. RESULT & DISCUSSION

Fig 5.1: Ingredient & Preference Input Screen

This screen is where the user starts their journey. Instead of going out to buy new ingredients, they can simply enter whatever they already have at home. They also get to choose whether they want a vegetarian, non-vegetarian, or vegan dish, and can set calorie preferences. This makes the recipe suggestions super

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

personalized and convenient.

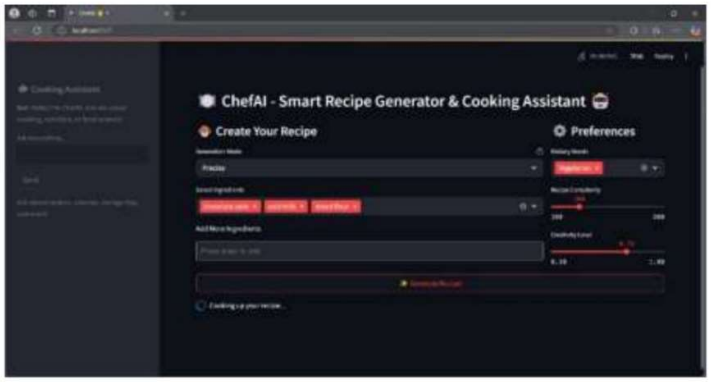


Fig 5.2: Recipe Generation Based on Inputs

Once the user submits their ingredients and preferences, our model takes over and generates multiple recipe options. The AI (powered by the T5 model) ensures that the recipes are creative but still match what the user has and wants—saving time and reducing food waste.

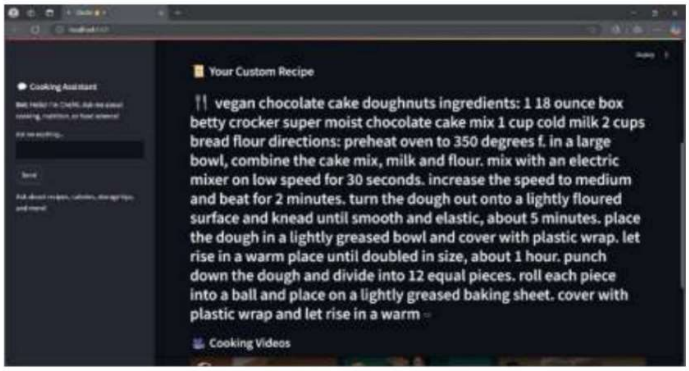


Fig 5.3: Recipe Display with Details

In this, the chosen recipes are displayed in their full glory. People are able to view step-by-step guidance alongside nutritional data such as the amount of calories contained within the dish. This assists them in making wise decisions about what they're consuming.

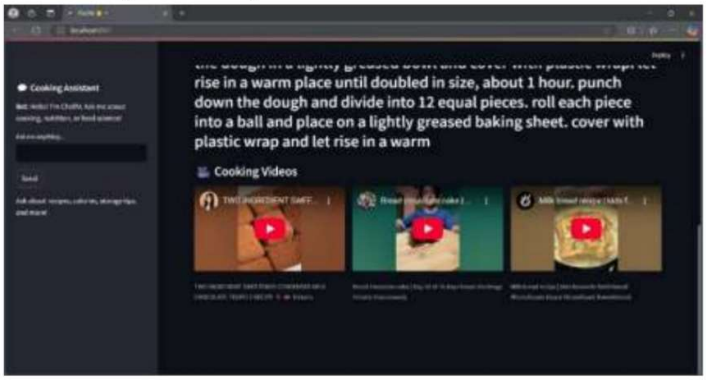


Fig 5.4: YouTube Video Suggestions

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

To make cooking even easier, we've added video recommendations using the YouTube API. Based on the recipe chosen, users get links to cooking videos that match it. So even if someone's new to cooking, they can just watch and follow along.

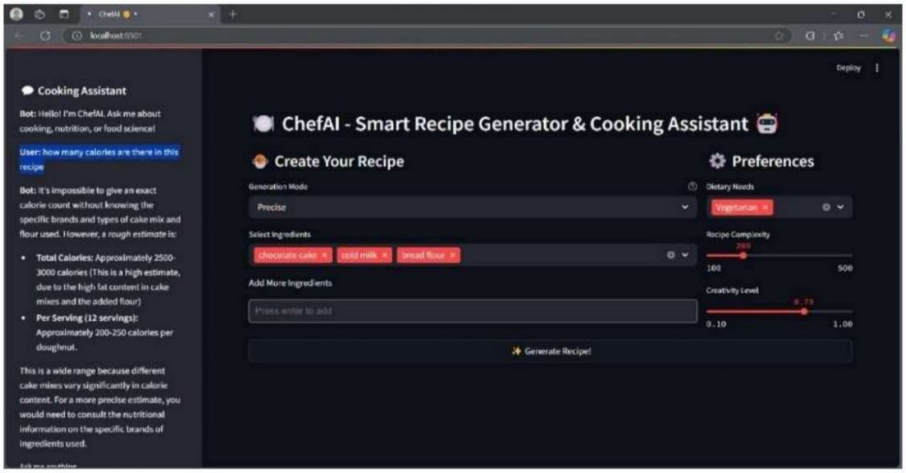


Fig 5.5: Chatbot Assistance with Inquiries

Finally, we've included a Gemini-powered chatbot. Sitting in the corner, this virtual assistant is available to assist with any recipe inquiries, including suggestions for different ingredients, cooking advice, or anything else the user is unsure of. It's like always having a cooking partner

VI. CONCLUSION

Cravify is the future of cooking companions - from fixed recipe databases to creating uniquely personalized kitchen experiences. With cutting-edge AI and easy-to-use interface, it learns your cooking style, adjusts to your mood, and caters to dietary requirements—essentially like having a personal chef who learns about you. Its capacity to understand context - whether instant comfort food for an extended evening or inspiration for a special dinner - sets it apart from traditional recipe uses. With its conversation-based interface and visual cooking instructions, Cravify not only suggests recipes but will guide you through the whole meal preparation process, answering questions and offering alternatives as you cook. It simplifies cooking today—but sooner or later, it'll cook on voice command, recognize what ingredients are inside with a look, and integrate with the rest of the machines in your kitchen to get it done for you. Dinners will much of the time cook themselves. In essence, Cravify is not to supersede human culinary innovation in the kitchen, but to enhance it - do away with frustrations without sacrificing the joy of cooking, and help all those from Frantic parents, to home cooks perfecting their skills, everyone can experiment with recipes made especially for their taste and convenience.

VII. REFERENCE

[1] [1] W. Min, S. Jiang, and R. Jain, "Food Recommendation: Framework, Existing Solutions, and Challenges," IEEE Transactions on Multimedia, vol. 22, no. 10, pp. 2659–2671, Oct. 2020.

[2] M. Gim, S. Park, and C. D. Yoo, "RecipeBowl: A Cooking Recommender for Ingredients and Recipes Using Set Transformer," IEEE Access, vol. 9, pp. 16084–16097, 2021.

[3] M. Chen, B. Dhingra, and W. Cohen, "Eating Healthier: Exploring Nutrition Information for Healthier Recipe Recommendation," Information Processing & Management, vol.57,no.6,p.102382,2020.

[4] J. M. Ordovas, L. R. Ferguson, E. S. Tai, and J. C. Mathers, "Personalised Nutrition and Health," BMJ, vol. 361, p.k2173,2018.

[5] R. Mate, M. Kadam, and N. Pawar, "Automated Recipe Chatbot," International Research Journal of Modernization in Engineering Technology and Science (IRJMETS), Dec. 2024.

[6] D. J. K. Patel, "An Artificial Intelligence Interactive Platform for Automated Chatbot with AI-Driven

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

Innovation in Recipe Searching," Mathematics and Computer Science, 2024.Mehta, Prateek. (2016). Creating Google Chrome Extensions. 10.1007/978-1-4842-1775-7.

[7] Salvador, A., Hynes, N., Aytar, Y., Ofli, F., Weber, I., & Torralba, A. (2019). "Inverse Cooking: Recipe Generation from Food Images." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10453–10462.

[8] Yang, K., & Lee, D. (2018). "NutRec: Nutrition-oriented Food Recommendation with Reinforcement Learning." International Conference on Health Informatics.

[9] Kusmierczyk, T., Trattner, C., & Nørvåg, K. (2020). "Temporal Patterns in Online Food Innovation: A Study of Recipe Sharing Platforms." International Journal of Information Management, 50, pp. 36–49.

[10] Teng, C. Y., Lin, Y. R., & Adamic, L. A. (2012). "Recipe Recommendation Using Ingredient Networks." Proceedings of the 4th Annual ACM Web Science Conference, pp. 298–307.

[11] Majumder, B., Li, S., & Balakrishnan, R. (2020). "Generating Personalized Recipes from Historical User Preferences." Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400206414

Date: 22/04/2025

Certificate of Publication

*This is to certify that author "**Ankita Darmal**" with paper ID "**IRJMETS70400206414**" has published a paper entitled "CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025***

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400206414

Date: 22/04/2025

Certificate of Publication

This is to certify that author "**Deven Deshmukh**" with paper ID "**IRJMETS70400206414**" has published a paper entitled "CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400206414

Date: 22/04/2025

Certificate of Publication

This is to certify that author "**Kasturi D Dhopat**" with paper ID "**IRJMETS70400206414**" has published a paper entitled "CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400206414

Date: 22/04/2025

Certificate of Publication

This is to certify that author "**Isha Kadam**" with paper ID "**IRJMETS70400206414**" has published a paper entitled "CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025**

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 07/Issue 04/70400206414

Date: 22/04/2025

Certificate of Publication

*This is to certify that author "**Prof. Vidya Bharde**" with paper ID "**IRJMETS70400206414**" has published a paper entitled "CRAVIFY: A PERSONALIZED RECIPE RECOMMENDATION SYSTEM USING MULTIMODAL INPUTS AND CONTEXTUAL AI" **in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 07, Issue 04, April 2025***

Editor in Chief

IRJMETS Impact Factor 8.187

We Wish For Your Better Future

www.irjmets.com

APPENDIX C

Plagiarism report

Result (percentage of plagiarism)	6%
Document title	Major_proj_Report.pdf
Character count	15,000
Special character count	41
Word count	2,119
Unique word count	1,060
Sentence count	128
Most common words	and, of, the, to, a
Average word length	6.1
Average words per sentence	16.6
Total links	0
Number of plagiarized words	125

21.04.2025 (date)

Kasturi D Dhopat (checked by)