

Flavour Fusion: AI-Driven Recipe Blogging

Team ID: LTVIP2026TMIDS42501

Project Overview:

Flavour Fusion is an AI-powered web platform designed to automatically create detailed recipe blogs based on user input. Users can provide a recipe topic along with the desired word count, and the system generates a complete blog post that includes ingredients, preparation steps, cooking tips, and an engaging introduction.

The application utilizes Google Generative AI (Gemini API) for content creation and Streamlit to build an interactive and user-friendly web interface. This project highlights how Generative AI can be integrated into real-world applications to automate content generation efficiently.

Problem Statement:

Manual recipe blog writing is time-consuming and requires consistent formatting and creativity.

There is a growing need for:

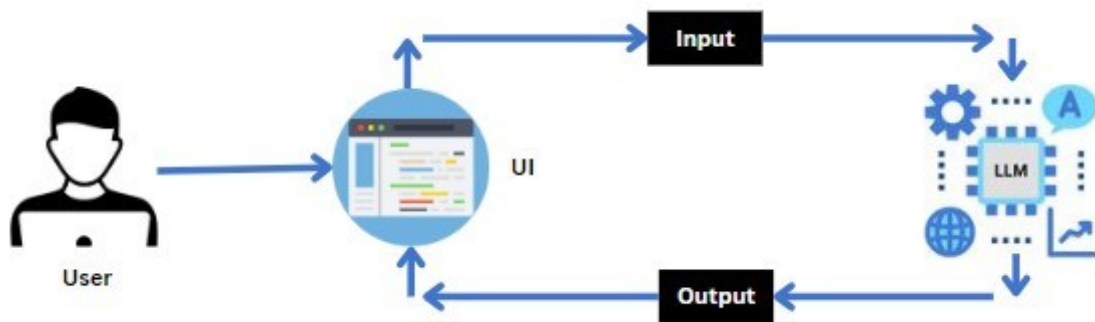
- Automated content generation
- Structured and SEO-friendly blog creation
- Faster blog production
- User-friendly interface
- AI-based intelligent writing support

Objectives:

The primary objectives of this project are:

- To develop an AI-powered recipe blog generator.
- To integrate Google Gemini API for intelligent content generation.
- To design a clean and responsive UI using Streamlit.
- To allow users to specify recipe topic and word count.
- To generate structured and engaging blog content automatically.

System Architecture:



Architecture Flow:

User Input → Streamlit Interface → Gemini API → AI Content Generation → Output Display

Components:

- **Frontend** – Streamlit
- **Backend** – Python
- **AI Model** – Google Gemini 1.5 Flash
- **Environment Variables** – Dotenv

This modular architecture ensures smooth communication between user input and AI-generated output.

Technologies Used:

Technology	Purpose
Python	Core programming
Streamlit	Web interface
Google Gen AI	Content generation
Dotenv	API key management

Folder Structure:

```
Flavour-Fusion/  
├── app.py  
├── .env  
├── requirements.txt  
├── README.md  
└── assets/  
    └── chatbot_image.png
```

Methodology:

Step 1: User Input User

enters:

- Recipe Topic
- Word Count

Step 2: Prompt Engineering

A structured prompt is created such as:

“Write a detailed, engaging, SEO-friendly blog post about Malai Kofta including ingredients, preparation steps, cooking tips, and conclusion.”

Step 3: API Request

The prompt is sent securely to the Gemini 1.5 Flash model.

Step 4: Content Generation

The AI model generates:

- Introduction
- Ingredients
- Preparation Steps
- Pro Tips

- Conclusion

Step 5: Output Display

The generated blog is displayed in the Streamlit web interface.

Implementation:

1) Installing Required Libraries:

```
pip install google-generativeai pip  
install python-dotenv    pip install  
streamlit
```

2) Main Code Structure:

```
import streamlit as st  
import  
google.generativeai as genai  
import os  
  
import random  
from dotenv  
import load_dotenv
```

3) Load API Key:

```
load_dotenv() api_key =  
os.getenv("GOOGLE_API_KEY")  
genai.configure(api_key=api_key)
```

4) Model Configuration:

```
generation_config = {  
    "temperature": 0.75,  
    "top_p": 0.95,  
    "top_k": 64,  
    "max_output_tokens": 2048,  
}
```

5) Creating Model:

```
model = genai.GenerativeModel( model_name="gemini-1.5-flash-  
latest", generation_config=generation_config  
)
```

6) Generate Recipe Function:

```
def generate_recipe(topic, word_count):  
    prompt = f"""  
    Write a detailed recipe blog about {topic}.  
    The blog should be approximately {word_count} words.  
    Include:  
    - Introduction  
    - Ingredients  
    - Step-by-step preparation  
    - Tips  
    - Conclusion  
    """  
    response = model.generate_content(prompt)  
    return response.text
```

7) User Interface Design:

Features:

- Clean heading
- Chatbot-style introduction
- Input fields: ◦ Recipe topic ◦ Word count
- Generate button
- Joke section while generating

- Styled background

8) Output:

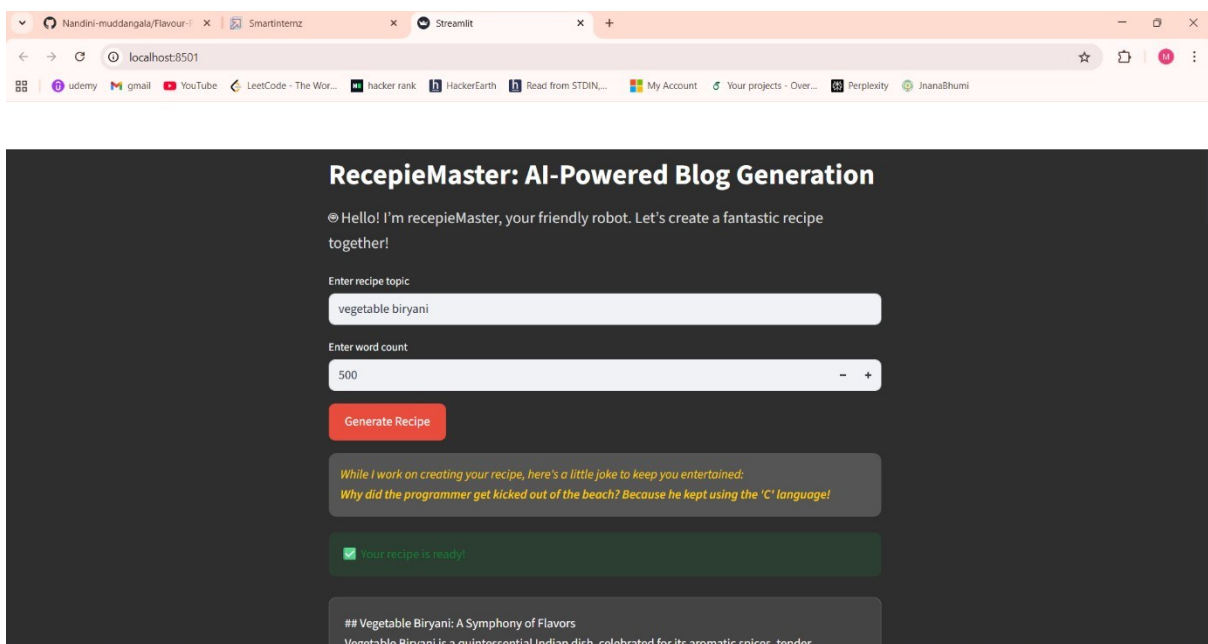
When user enters:

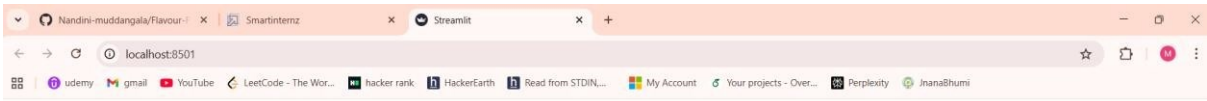
Topic: Malai Kofta , **Word Count:** 500

The system generates:

- Introduction
- Ingredients list
- Cooking steps
- Pro tips
- Conclusion

Output Screenshots:





Vegetable Biryani: A Symphony of Flavors

Vegetable Biryani is a quintessential Indian dish, celebrated for its aromatic spices, tender vegetables, and perfectly cooked, fragrant basmati rice. Originating from the royal kitchens of the Mughal Empire, biryani has evolved into countless regional variations, with the vegetarian rendition becoming a beloved staple across India and beyond. It's not just a meal; it's an experience – a harmonious blend of textures and tastes, making it perfect for festive occasions, family gatherings, or a comforting weeknight dinner.

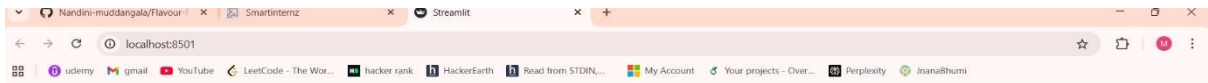
This recipe guides you through creating a layered biryani, where fluffy basmati rice and a rich, spicy vegetable curry are cooked together in a sealed pot (dum method) to infuse every grain with incredible flavor.

Yields: 4-6 servings **Prep time:** 30 minutes (plus 30 minutes rice soaking) **Cook time:** 45-60 minutes

Ingredients:

For the Basmati Rice:

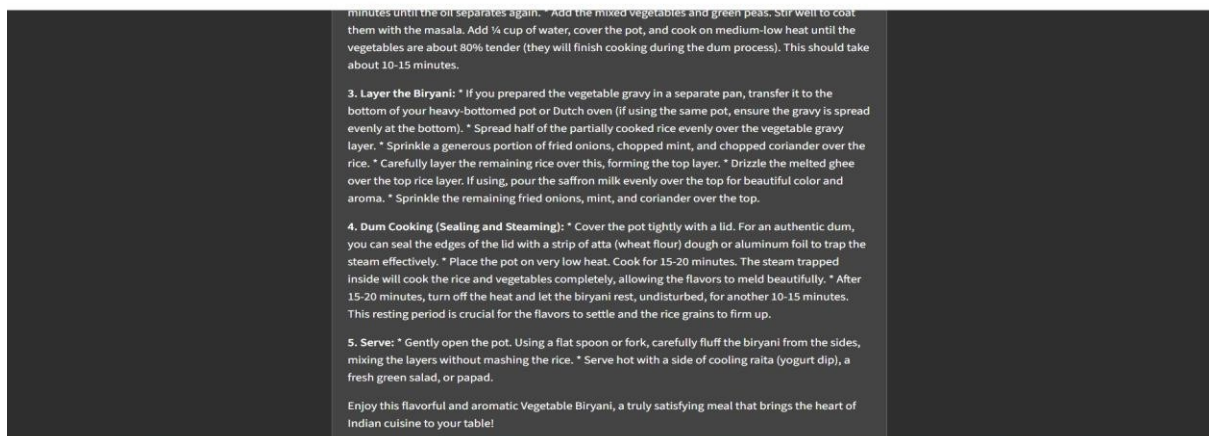
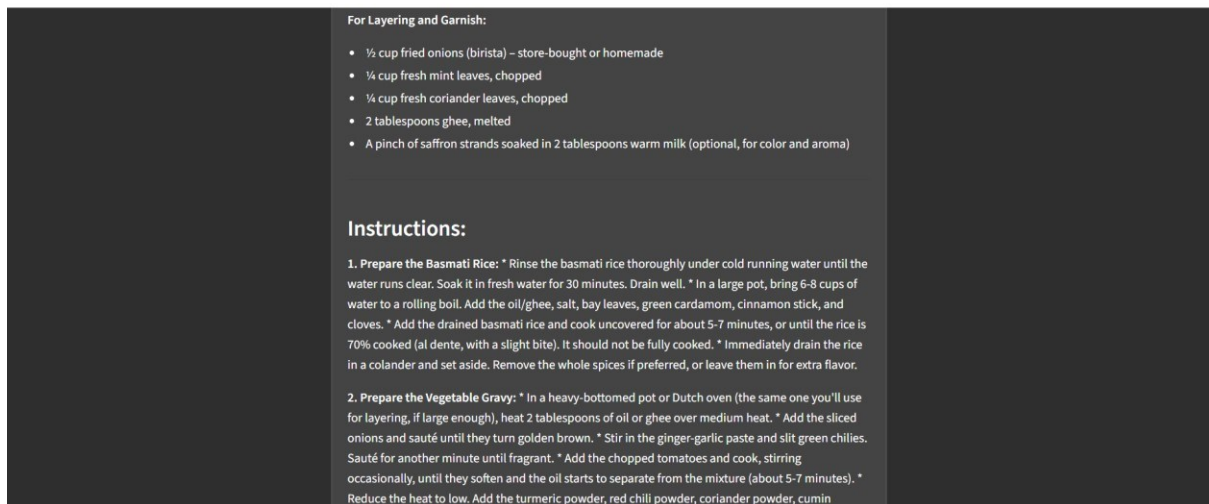
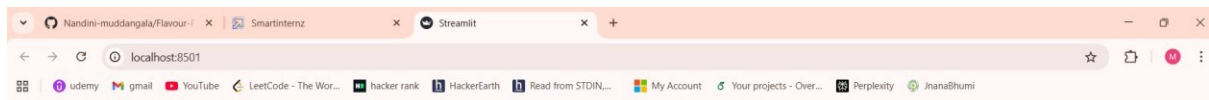
- 2 cups (approximately 400g) good quality Basmati rice
- 6-8 cups water (for boiling)
- 1 tablespoon oil or ghee
- 1 teaspoon salt



- 2 bay leaves
- 4 green cardamom pods
- 2-inch cinnamon stick
- 4 cloves

For the Vegetable Gravy (Curry):

- 2 tablespoons oil or ghee
- 1 large onion, thinly sliced
- 1 tablespoon ginger-garlic paste
- 2-3 green chilies, slit lengthwise (adjust to taste)
- 1 large tomato, finely chopped
- 1 cup mixed vegetables (carrots, green beans, potatoes, cauliflower florets – all diced into 1-inch pieces)
- ½ cup fresh green peas
- ½ cup plain yogurt, whisked smooth
- 1 teaspoon turmeric powder
- 1 teaspoon red chili powder (or Kashmiri chili for color)
- 1 tablespoon coriander powder
- 1 teaspoon cumin powder
- 2 tablespoons Biryani Masala (store-bought or homemade)
- Salt to taste
- ¼ cup water (if needed)



Advantages:

- Saves time
- AI-powered automation
- Easy to use
- No writing skills required
- Customizable word count

Future Enhancements

- Multi-language blog generation
- SEO keyword optimization support
- Direct publishing to blogging platforms

- Blog export in PDF/Word format
- Personalized tone selection
- AI-generated recipe images

Run the Application:

Inside the project folder, run:“ streamlit run app.py ”

Conclusion:

Flavour Fusion demonstrates the effective integration of Generative AI into a real-world content automation system. By combining Streamlit with the Gemini 1.5 Flash model, the application generates customized and structured recipe blogs instantly. The project highlights the growing potential of AI-powered solutions in digital content creation. With future enhancements, the system can evolve into a comprehensive blogging assistant platform.