

1. INTRODUCTION

1.1. Project Overview

Flavour Fusion is an AI-driven recipe blogging web application designed to generate customized and well-structured recipe blogs based on user input. The application allows users to enter a recipe topic and specify the desired word count. Using a pre-trained generative AI model, the system produces detailed and engaging recipe content in a short time.

The project aims to simplify the process of recipe content creation for food bloggers, cooking enthusiasts, and busy individuals. Instead of manually writing lengthy recipes, users can rely on the application to generate high-quality content instantly. The application is built using Streamlit for the user interface and integrates the Gemini Flash Lite (models/gemini-flash-lite-latest) model for efficient and fast text generation.

Overall, Flavour Fusion demonstrates the practical use of generative AI in content creation by providing a user-friendly, time-saving, and effective solution for automated recipe blog generation.

1.2. Objectives

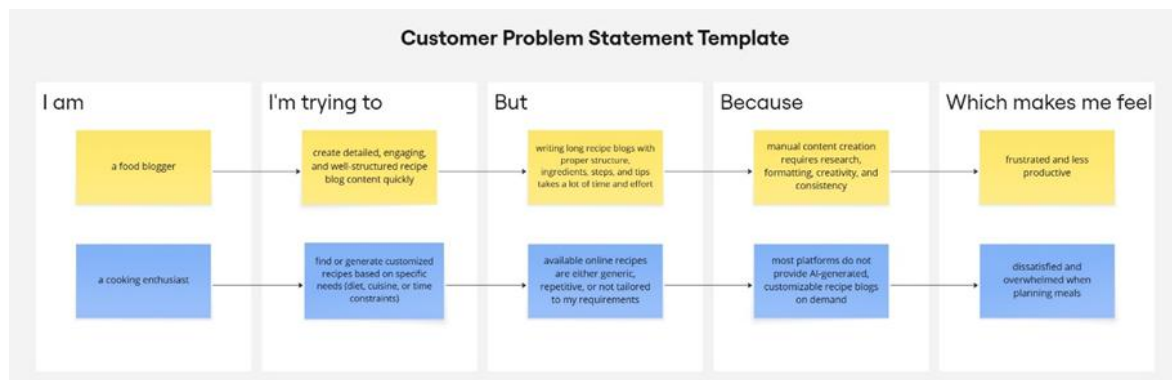
The main objectives of the Flavour Fusion: AI-Driven Recipe Blogging project are:

- To develop a web-based application that generates recipe blogs using artificial intelligence
- To allow users to create customized recipe content by providing a topic and word count
- To reduce the time and effort required for manual recipe writing
- To integrate a pre-trained generative AI model for fast and accurate text generation
- To provide a simple and user-friendly interface using Streamlit
- To enhance user experience through efficient and engaging content generation

2. Ideation Phase

2.1. Problem Statement

Therefore, food bloggers and busy individuals are in dire need of an easier way to pen or write a detailed recipe blog in the shortest length of time. Writing recipes manually requires a lot of efforts, creativities, and proper formatting that are really very time-consuming. Also, most of the recipes available online tend to be general and do not meet specific user needs. The result is usually frustration and reduced productivity. The Flavour Fusion application solves this problem by using an AI model to quickly generate customized and well-structured recipe blogs based on user input.



Problem Statement (PS)	I am	I'm trying to	But	Because	Which makes me feel
PS-1	A food blogger	Create detailed, and high-quality recipe blog content in less time	Writing long and recipe blogs manually is time-consuming	It requires research, creativity, formatting, and consistency	Frustrated and less productive
PS-2	A cooking enthusia	Generate customized	Existing recipes available	Most platforms do not offer AI-driven recipe blog	Dissatisfied and overwhelme

	st	recipes	online are generic and not customizable	generation	d while planning meals
--	----	---------	---	------------	------------------------

2.2. Empathy Map Canvas

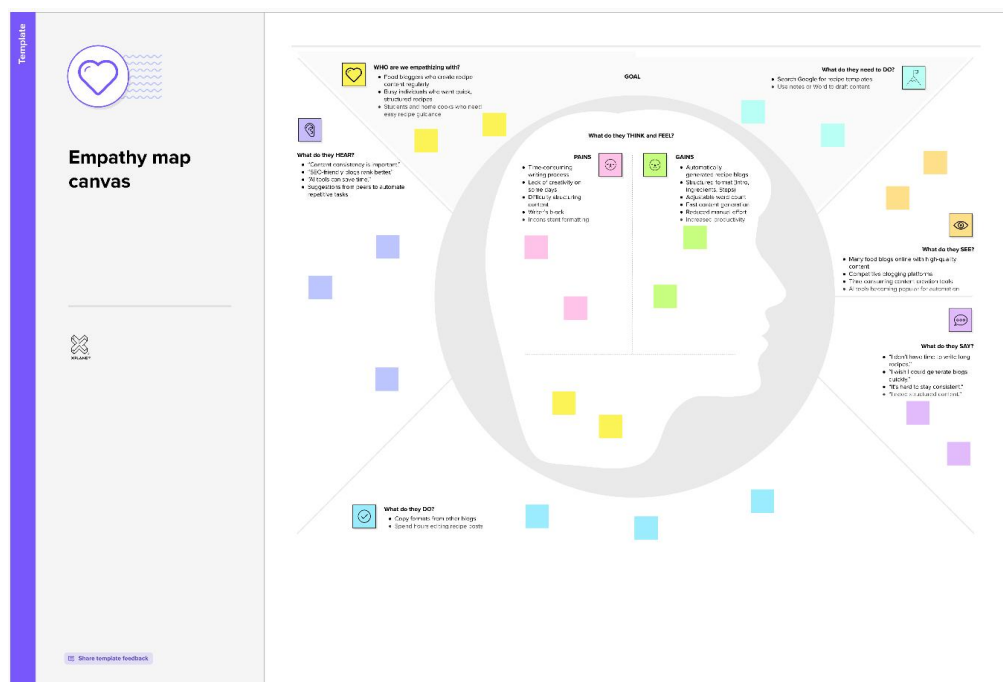
Empathy Map Canvas:

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

It is a useful tool to help teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

Example:



2.3 Brainstorming


Brainstorm & Idea Prioritization:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Step-1: Team Gathering, Collaboration and Select the Problem Statement

Template



Brainstorm & idea prioritization for Flavour Fusion

Use this brainstorming session to identify challenges faced in recipe blogging and explore how Generative AI can automate recipe content creation. The goal is to define a clear problem statement and select the most impactful solution idea for development.

15 minutes to prepare
1 hour team discussion
4 hours execution

Before Team Discussion

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- Team gathering**
All team members discuss current issues in recipe blogging and content creation.
- Set the goal**
"Manual recipe blog writing is time-consuming and inefficient."
- Final discussion**
 - Scenario for AI
 - Scenario that fits model for AI generation

1 Define your problem statement

What problem are we solving?
"How can we use Generative AI to automatically generate structured and customizable recipe blogs for users?"

5 minutes

Outcome

Manual recipe blogging requires significant time and effort.

25 Key Discussion Guidelines

- Focus on automation
- Prioritize user-friendly interface
- Ensure fast AI response
- Maintain content quality
- Keep implementation simple

Step-2: Brainstorm, Idea Listing and Grouping

2 Brainstorm

Write down possible features and improvements that can help automate recipe blogging using Generative AI.

10 minutes

Person 1

Generate recipe blog from topic

User selects top 10 content

Structured format (Intro, Ingredients, Steps)

Instant content generation

Person 2

Use Gemini Flash Lite model

Adjust creativity using temperature

Adjust creativity using temperature

Add engaging title

Person 3

Streamlined web interface

Simple input box

Generate buttons

Display output nicely

Person 4

Progress saving plan after recipe

Copy to clipboard action

Save recipe option (AI-generated)

Multi-language support (Future)

3 Group Ideas

Take time sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a headline-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

Core Application

Recipe topic input

Word count selection

Structured recipe output

AI Integration

Gemini Flash Lite API

Prompt design

Parameter tuning

User Experience

Clean UI

Fast response time

Easy readability

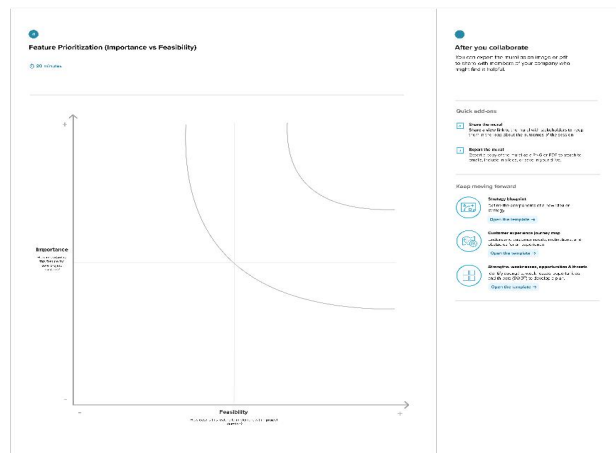
Future Enhancements

Multi-language support

Image generation

User accounts

Step-3: Idea Prioritization



3. Requirement Analysis

3.1. Solution Requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Recipe Input Module	Enter recipe topic & Select word count
FR-2	Input Validation	Validate empty input & word count range
FR-3	AI Integration	Connect to Gemini Flash Lite API Send structured prompt to model
FR-4	Recipe Generation	Generate structured recipe blog Include introduction, ingredients, steps
FR-5	Output Display	Display generated recipe in UI Allow user to copy content
FR-6	Additional Feature	Generate programming joke (optional feature)

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

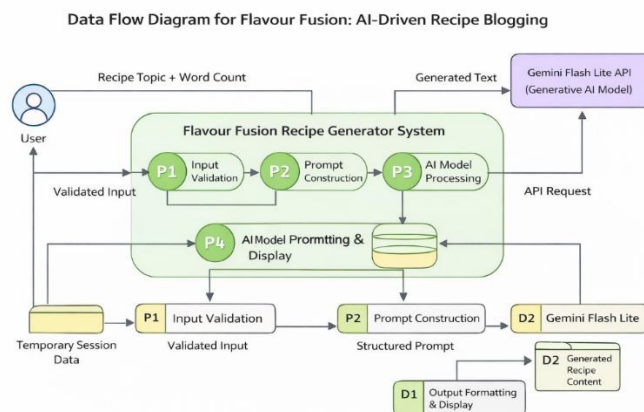
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The application must have a simple and intuitive Streamlit interface that is easy to use.

NFR-2	Security	API keys must be securely stored and not exposed in the frontend.
NFR-3	Reliability	The system should generate consistent and structured outputs for valid inputs.
NFR-4	Performance	Recipe generation should complete within a few seconds.
NFR-5	Availability	The application should be accessible online whenever deployed.
NFR-6	Scalability	The system should handle multiple users without significant performance degradation.

3.2. Data Flow Diagram

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	User Interface Setup	USN-1	As a user, I can access a Streamlit-based interface to enter a recipe topic and word count.	I can access my account / dashboard	High	Sprint-1
Administrator	Input Validation	USN-2	As a user, I want the application to validate my inputs before generating the recipe.	I can validate the inputs	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer Care Executive	AI Model Integration	USN-3	As a user, I want the system to generate a recipe blog using the Gemini Flash Lite model.	I can integrate model	High	Sprint-2
Administrator	Joke Generation	USN-4	As a user, I want to see a programming joke while the recipe is being generated.	I can generate a joke for user interaction while recipe generation delayed	Medium	Sprint-2
Customer (Mobile user)	Output Display	USN-5	As a user, I want to view the generated recipe blog clearly on the screen.	I can see the output of recipe generated	High	Sprint-3
Customer Care Executive	Deployment	USN-6	As a user, I want the application to be deployed and accessible through the internet.	I can deploy in the system	Medium	Sprint-3

3.3. Technology Stack

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web-based interface where users enter recipe topic and word count.	Streamlit (Python Web Framework)
2.	Application Logic-1	Input validation and prompt construction logic	Python
3.	Application Logic-2	AI request handling and response processing	Google Generative AI API
4.	Application Logic-3	Recipe formatting and output structuring	Python
5.	File Storage	Local environment for source code and logs	Local File System
6.	External API-1	Generative AI service for recipe blog generation	Gemini Flash Lite (models/gemini-flash-lite-latest)
7.	Machine Learning Model	Pre-trained generative AI model for text generation	Gemini Flash Lite Model

8.	Infrastructure (Server / Cloud)	Deployment of application	Streamlit Cloud / Local Deployment
----	------------------------------------	---------------------------	---------------------------------------

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Web framework and development tools used	Streamlit, Python
2.	Security Implementations	Secure storage of API keys and environment variables	Environment Variables (.env), Streamlit Secrets
3.	Scalable Architecture	Web-based architecture supporting multiple users	Cloud-based deployment (Streamlit Cloud)
4.	Availability	Application accessible online after deployment	Streamlit Cloud Hosting
5.	Performance	Fast response generation using lightweight AI model	Gemini Flash Lite (optimized for low latency)

4. PROJECT DESIGN

4.1. Problem Solution Fit

Problem – Solution Fit Template:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why

Purpose:

- ☐ Solve complex problems in a way that fits the state of your customers.
- ☐ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- ☐ Sharpen your communication and marketing strategy with the right triggers and messaging.
- ☐ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
- ☐ **Understand the existing situation in order to improve it for your target group.**

Template:

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Who is your customer? I.e. working parents of 0-5 y.o. kids	CS	6. CUSTOMER CONSTRAINTS What constraints prevent your customers from taking action or limit their choices of solutions? I.e. spending power, budget, no cash, network connection, available devices.	CC	5. AVAILABLE SOLUTIONS Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? I.e. pen and paper is an alternative to digital notetaking	AS	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.	J&P	9. PROBLEM ROOT CAUSE What is the real reason that this problem exists? What is the back story behind the need to do this job? I.e. customers have to do it because of the change in regulations.	RC	7. BEHAVIOUR What does your customer do to address the problem and get the job done? I.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)	BE	
Identify strong TR & EM	3. TRIGGERS What triggers customers to act? I.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.	TR	10. YOUR SOLUTION If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.	SL	8. CHANNELS of BEHAVIOUR 8.1 ONLINE What kind of actions do customers take online? Extract online channels from #7	CH	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE / AFTER How do customers feel when they face a problem or a job and afterwards? I.e. lost, insecure > confident, in control - use it in your communication strategy & design.	EM	8.2 OFFLINE What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.				

4.2. Proposed Solution

Proposed Solution Template:

Project team shall fill the following information in the proposed solution template.

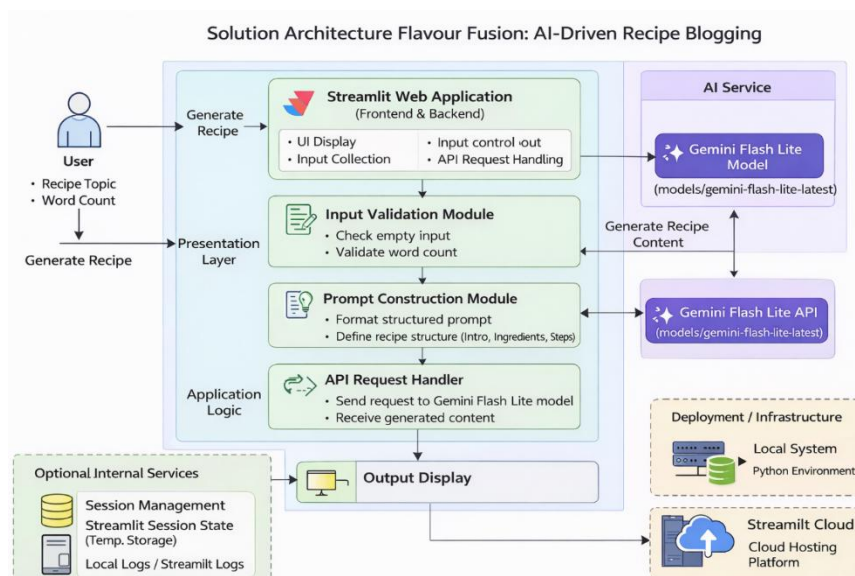
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Many food bloggers, students, and busy individuals struggle to create well-structured and detailed recipe blogs due to time constraints and lack of consistent creativity. Manual recipe writing is repetitive, time-consuming, and requires formatting effort. There is a need for an automated system that can generate structured recipe blogs quickly and efficiently.
2.	Idea / Solution description	Flavour Fusion is an AI-powered web application that generates customized recipe blogs based on user input. Users enter a recipe topic and desired word count, and the system uses a pre-trained generative AI model (Gemini Flash Lite) to create a complete, structured recipe blog including introduction, ingredients, and preparation steps. The application provides instant results through a simple Streamlit-based interface.
3.	Novelty / Uniqueness	Unlike traditional recipe platforms that only display pre-written recipes, Flavour Fusion dynamically generates unique, customized recipe blogs in real-time. The integration of Generative AI allows users to control word count and receive structured content instantly, making it more flexible and

		interactive than static recipe websites.
4.	Social Impact / Customer Satisfaction	The solution saves time for food bloggers, home cooks, and students by automating content creation. It improves productivity, reduces manual effort, and provides accessible recipe content generation. Users benefit from structured and ready-to-use blogs without requiring advanced writing skills.
5.	Business Model (Revenue Model)	The application can adopt a freemium model where basic recipe generation is free, and advanced features (multi-language support, image generation, premium templates) are offered through subscription. Revenue can also be generated through advertisements, affiliate marketing, or API-based enterprise usage.
6.	Scalability of the Solution	The application is cloud-deployable and uses a scalable generative AI API. It can handle multiple users simultaneously when deployed on cloud infrastructure. Additional features such as multi-language support, user accounts, and content storage can be added to scale the system further.

4.3 Solution Architecture

Solution Architecture:

The solution architecture of Flavour Fusion consists of a Streamlit-based web application that collects user inputs and validates them. The application constructs a structured prompt and sends it to the Gemini Flash Lite generative AI model through an API call. The AI model generates a structured recipe blog, which is then formatted and displayed to the user. The system can be deployed locally or on a cloud platform such as Streamlit Cloud.



5. PROJECT PLANNING & SCHEDULING

5.1. Project Planning

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
<i>Sprint-1</i>	<i>User Interface Setup</i>	<i>USN-1</i>	<i>As a user, I can access a Streamlit-</i>	<i>2</i>	<i>High</i>	<i>All Team Members</i>

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			<i>based interface to enter a recipe topic and word count.</i>			
<i>Sprint-1</i>	<i>Input Validation</i>	<i>USN-2</i>	<i>As a user, I want the application to validate my inputs before generating the recipe.</i>	<i>1</i>	<i>High</i>	<i>All Team Members</i>
<i>Sprint-2</i>	<i>AI Model Integration</i>	<i>USN-3</i>	<i>As a user, I want the system to generate a recipe blog using the Gemini Flash Lite model.</i>	<i>3</i>	<i>High</i>	<i>All Team Members</i>
<i>Sprint-2</i>	<i>Joke Generation</i>	<i>USN-4</i>	<i>As a user, I want to see a programming joke while the recipe is being generated.</i>	<i>1</i>	<i>Medium</i>	<i>All Team Members</i>
<i>Sprint-3</i>	<i>Output Display</i>	<i>USN-5</i>	<i>As a user, I want to view the generated recipe blog clearly on the</i>	<i>2</i>	<i>High</i>	<i>All Team Members</i>

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
			screen.			
Sprint-3	Deployment	USN-6	As a user, I want the application to be deployed and accessible through the internet.	2	Medium	All Team Members

Project Tracker, Velocity & Burndown Chart: (4 Marks)

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint 1	20	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 1	20	4 Days	28 January 2026	31 January 2026	20	31 January 2026
Sprint 2	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 2	20	8 Days	02 February 2026	09 February 2026	20	09 February 2026
Sprint 3	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026
Sprint 3	20	7 Days	12 February 2026	18 February 2026	20	18 February 2026

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Test Scenarios & Results

Test Case ID	Scenario (What to test)	Test Steps (How to test)	Expected Result	Actual Result	Pass/Fail
FT-01	Recipe Topic Input Validation	Enter valid recipe topic and leave field empty	Valid input accepted, error shown for empty input	As Expected	Pass
FT-02	Ingredient Input Validation	Enter valid ingredients and try empty/invalid inputs	Accepts valid input, shows warning for invalid input	As Expected	Pass
FT-03	AI Recipe Generation	Enter recipe topic and click "Generate Recipe"	Structured recipe with title,	As Expected	Pass

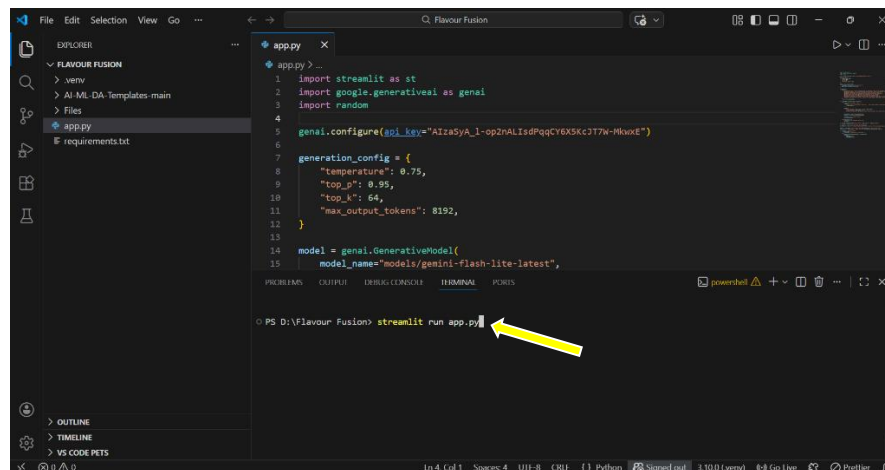
			<i>ingredients, and steps generated</i>		
FT-04	<i>Gemini API Connection Check</i>	<i>Trigger recipe generation with valid API key</i>	<i>API responds successfully and generates recipe</i>	<i>As Expected</i>	<i>Pass</i>
FT-05	<i>Programming Joke Feature</i>	<i>Generate recipe and check joke display</i>	<i>Programming joke appears along with recipe</i>	<i>As Expected</i>	<i>Pass</i>
PT-01	<i>Response Time Test</i>	<i>Measure time after clicking generate</i>	<i>Recipe generated within 3–5 seconds</i>	<i>Within Limit</i>	<i>Pass</i>
PT-02	<i>Multiple Request Handling</i>	<i>Generate multiple recipes sequentially</i>	<i>Application handles requests without crash</i>	<i>Stable</i>	<i>Pass</i>
PT-03	<i>Deployment Test</i>	<i>Access deployed app via browser</i>	<i>Application loads and works correctly online</i>	<i>Working</i>	<i>Pass</i>

7. RESULTS

7.1. Output Screenshots

The complete execution of Flavour Fusion: AI-Driven Recipe Blogging application is represented step by step in the following screenshots.

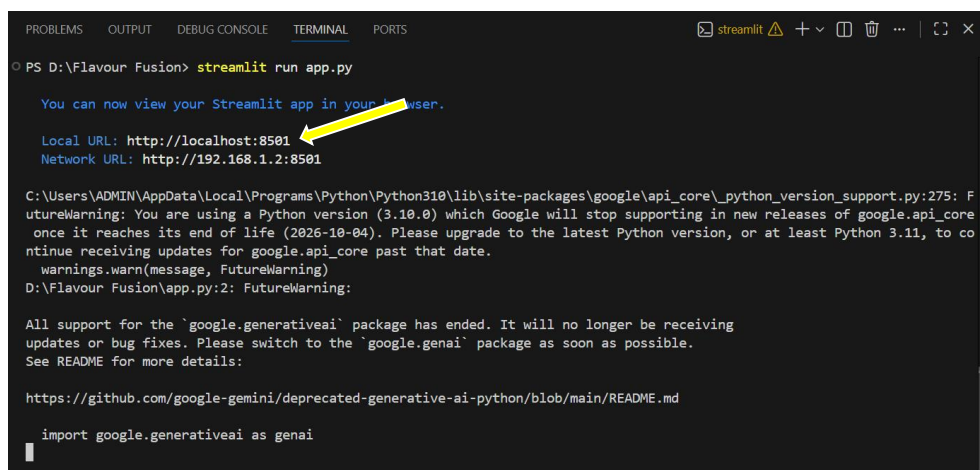
Step 1: To run the Streamlit Application we have to use the command `streamlit run app.py` in the terminal in path where the `app.py` file is located.



The screenshot shows the Visual Studio Code editor with a file named `app.py` open. The file contains Python code for a Streamlit application that uses the Google Gemini API. The terminal at the bottom shows the command `streamlit run app.py` being executed, with a yellow arrow pointing to it.

```
1 import streamlit as st
2 import google.generativeai as genai
3 import random
4
5 genai.configure(api_key="AIzaSYA_1-op2nALIsDpQqCY6X5Kc377N-MkuxE")
6
7 generation_config = {
8     "temperature": 0.75,
9     "top_p": 0.95,
10    "top_k": 64,
11    "max_output_tokens": 8192,
12 }
13
14 model = genai.GenerativeModel(
15     model_name="models/gemini-flash-lite-latest",
```

Step 2: After running the command in terminal, the code will get executed and the webpage will open directly. Another way to open webpage is that a localhost link will get generated in the terminal, we can access the webpage using that link.



The screenshot shows the terminal output after running `streamlit run app.py`. It displays the message "You can now view your Streamlit app in your browser." and provides the local URL `http://localhost:8501` and the network URL `http://192.168.1.2:8501`. A yellow arrow points to the local URL. Below this, there is a FutureWarning about the Python version (3.10.0) and a message about the deprecation of the `google.generativeai` package.

```
PS D:\Flavour Fusion> streamlit run app.py

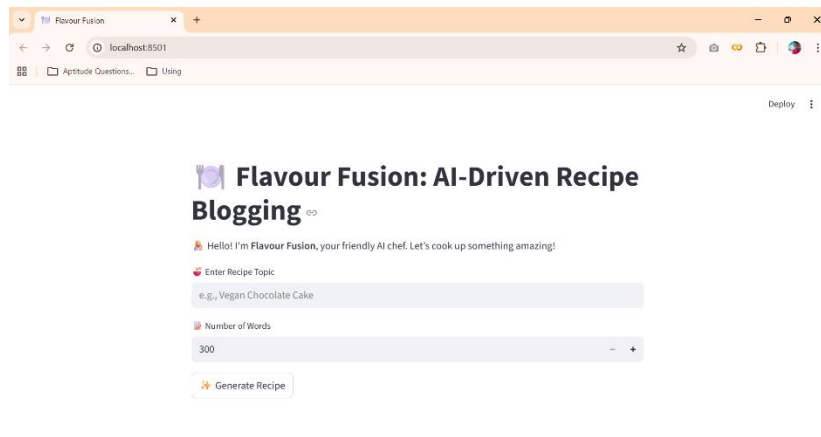
You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.1.2:8501

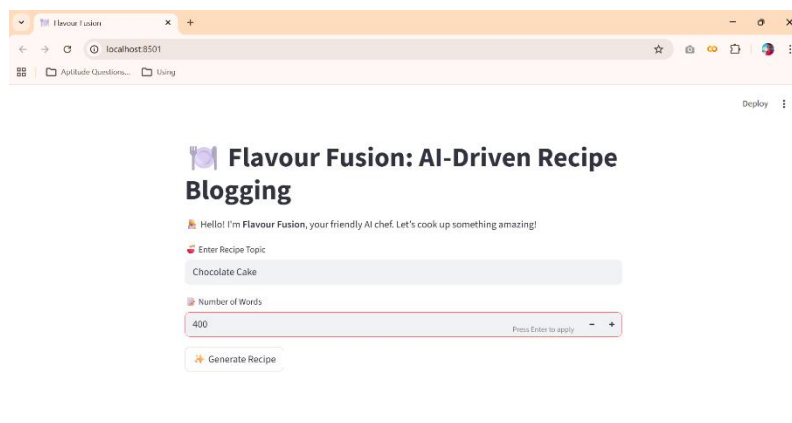
C:\Users\ADMIN\AppData\Local\Programs\Python\Python310\lib\site-packages\google\api_core\_python_version_support.py:275: FutureWarning: You are using a Python version (3.10.0) which Google will stop supporting in new releases of google.api_core once it reaches its end of life (2026-10-04). Please upgrade to the latest Python version, or at least Python 3.11, to continue receiving updates for google.api_core past that date.
  warnings.warn(message, FutureWarning)
D:\Flavour Fusion\app.py:2: FutureWarning:
All support for the 'google.generativeai' package has ended. It will no longer be receiving updates or bug fixes. Please switch to the 'google.genai' package as soon as possible.
See README for more details:
https://github.com/google-gemini/deprecated-generative-ai-python/blob/main/README.md

import google.generativeai as genai
```

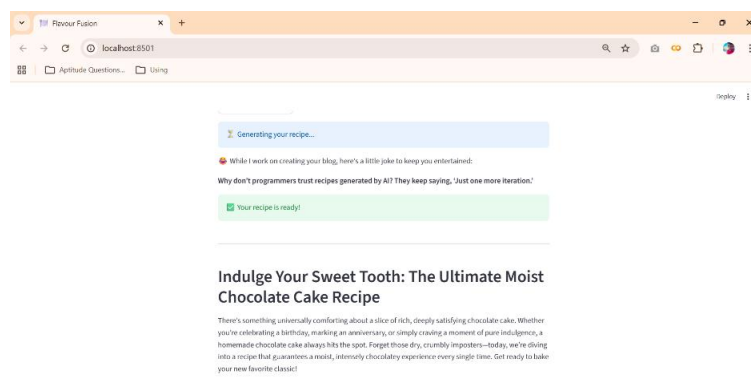
Step 3: The Streamlit webpage opens as shown in the figure given below. This is an automated webpage. No secondary HTML codes required to build this webpage. Python code itself consists the webpage building code.

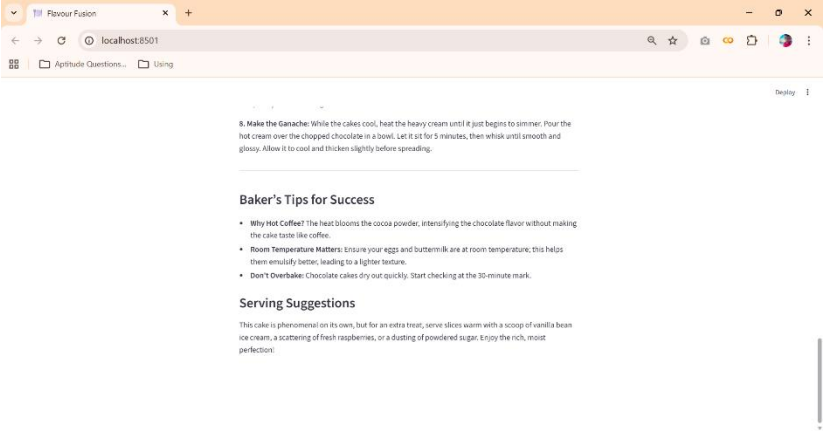
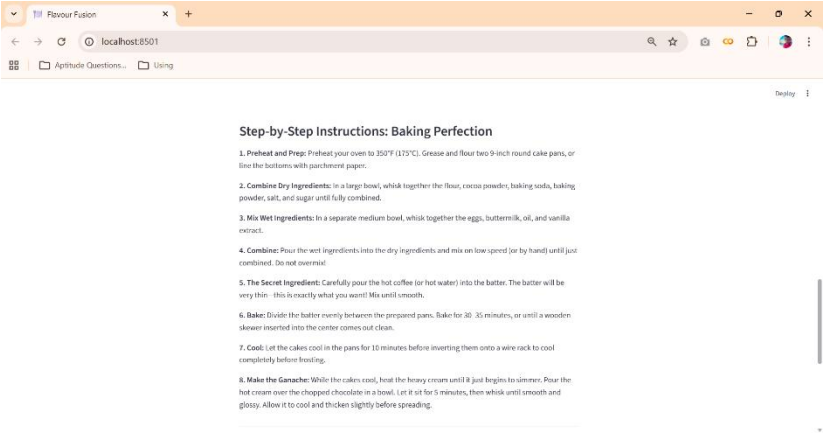
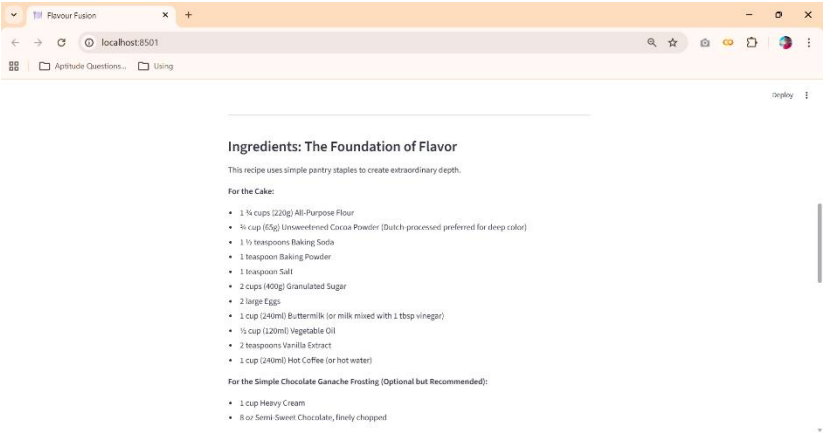


Step 3: The user has to give inputs in the website such as a Recipe Name and Number of Words. The Number of words means with how many words the recipe should be generated. After entering the required details, the user should click on Generate Recipe button to generate the recipe. Here I chose the Chocolate Cake Recipe with 400 words count.



Step 4: After clicking the Generate Recipe button, in fraction of seconds a simple joke will be generated as shown in the below figure to engage the users if recipe generation get delayed. The Food Recipe will be generated based on the user inputs as shown in the following four images.





8. ADVANTAGES AND DISADVANTAGES

Advantages

- Saves time by automatically generating detailed recipe blogs
- Reduces manual effort for food bloggers and users
- Generates customized content based on user input
- User-friendly interface built using Streamlit
- Fast and efficient content generation using a pre-trained AI model
- No requirement for dataset collection or model training

Disadvantages

- Requires an active internet connection to access the AI model
- Depends on third-party AI APIs for content generation
- Limited to text-based recipe content only
- Output quality depends on the clarity of user input

9. CONCLUSION

The Flavour Fusion: AI-Driven Recipe Blogging project successfully demonstrates how generative AI can be used to automate recipe blog creation. The application allows users to generate customized and well-structured recipe content by providing a topic and word count, reducing the time and effort required for manual writing. By integrating a pre-trained AI model with a user-friendly Streamlit interface, the project delivers fast, efficient, and high-quality results, making it a useful tool for food bloggers and cooking enthusiasts.

10. FUTURE SCOPE

The Flavour Fusion project can be enhanced further by adding support for multiple languages to reach a wider audience. Future improvements may include generating recipe images along with text, adding user accounts to save favorite recipes, and providing personalized recipe recommendations. The application can also be extended to support voice-based input and mobile platform deployment, making it more accessible and user-friendly.

11. APPENDIX

11.1. Source Code

The source code for the Flavour Fusion: AI-Driven Recipe Blogging project includes the implementation of the Streamlit user interface, integration of the Gemini Flash Lite model using the Google Generative AI API, recipe blog generation logic, and the programmer joke feature. The code is written in Python and follows a modular and readable structure.

11.2. Github & Project Demo Link

Github Repository Link: <https://github.com/ushameruva/Flavour-Fusion-AI-Driven>

Demo Link:

<https://drive.google.com/file/d/1UNs310OFiQYlrSMNxBOeY25v4pP6wgDN/view?usp=sharing>