FINAL REPORT

FLAVOUR FUSION: AI-Driven Recipe Blogging

1. INTRODUCTION

1.1 Project Overview

Flavor Fusion is a web-based Generative AI application designed to automate the creation of structured recipe blog posts. The system leverages Google's Gemini (gemini-flash-latest) model to generate complete blog-style recipes based on user input.

The application allows users to:

- Enter a recipe topic
- Select desired word count (100–2000 words)
- Generate a structured recipe blog
- Download the output as a Markdown (.md) file

The solution is implemented using Streamlit for the frontend interface and Python for application logic, with Gemini API integration for AI-driven content generation.

1.2 Purpose

The purpose of this project is to:

- Reduce the time required to write structured recipe blogs
- Provide AI-assisted content generation for food enthusiasts and bloggers
- Demonstrate practical integration of Generative AI into a real-world web application
- Deliver a lightweight, scalable, and deployable AI-based system

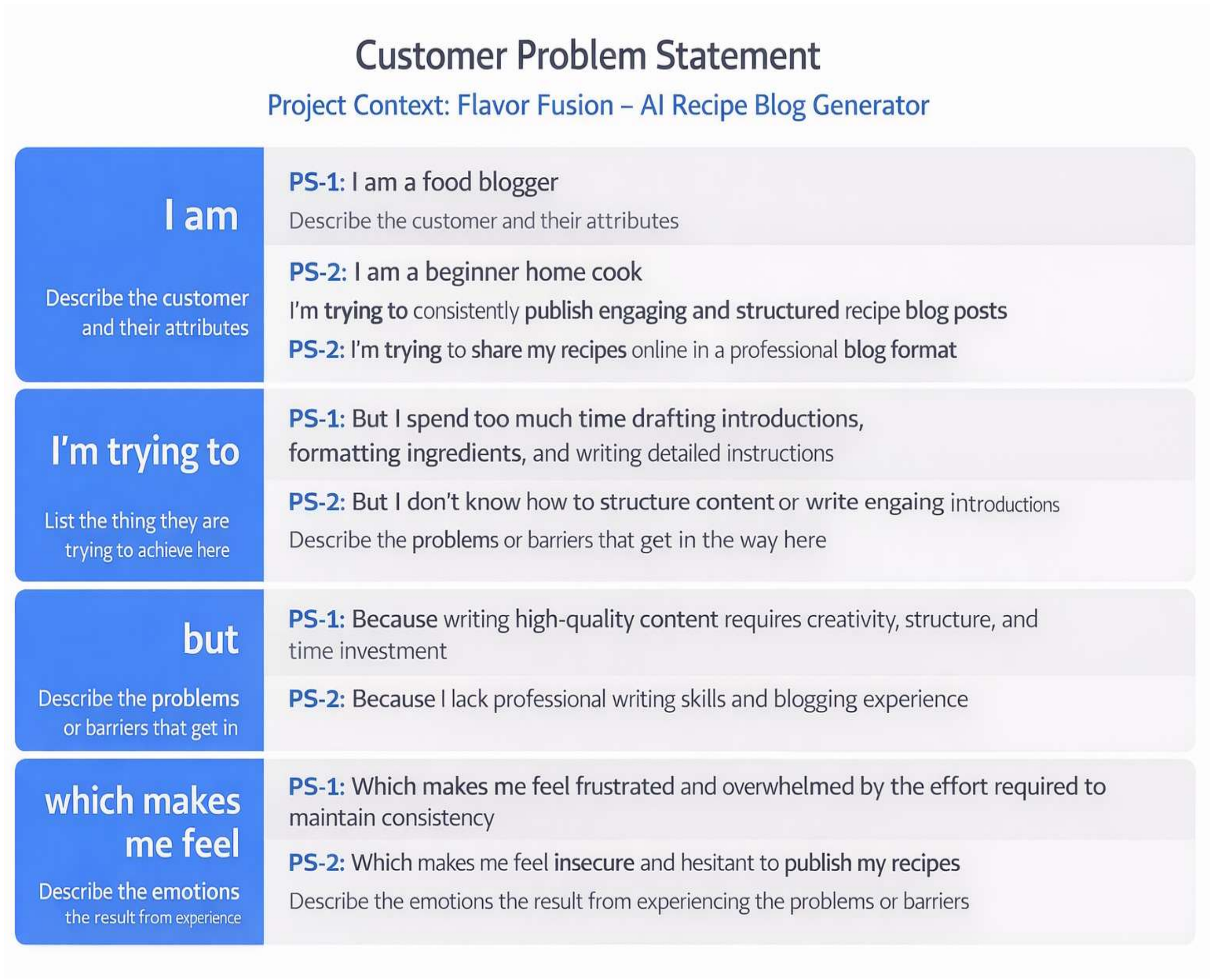
2. IDEATION PHASE

2.1 Problem Statement

Food bloggers and home cooks spend significant time writing structured recipe blog posts. Drafting engaging introductions, organizing ingredients, and formatting instructions requires writing skills and effort.

Traditional recipe websites provide static content and do not assist users in generating customized, blog-ready content instantly.

Flavor Fusion addresses this gap by providing AI-powered automated recipe blog generation.



Customer Problem Statement

Project Context: Flavor Fusion – AI Recipe Blog Generator

I am Describe the customer and their attributes	**PS-1:** I am a food blogger Describe the customer and their attributes **PS-2:** I am a beginner home cook I'm **trying to** consistently **publish engaging and structured** recipe **blog posts** **PS-2:** I'm **trying** to **share my recipes** online in a professional **blog format**
I'm trying to List the thing they are trying to achieve here	**PS-1:** But I spend too much time drafting introductions, formatting ingredients, and writing detailed instructions **PS-2:** But I don't know how to **structure content** or **write engaging** introductions Describe the **problems** or barriers that get in the way here
but Describe the problems or barriers that get in	**PS-1:** Because writing **high-quality content** requires creativity, structure, and time investment **PS-2:** Because I lack professional writing skills and blogging experience
which makes me feel Describe the emotions the result from experience	**PS-1:** Which makes me feel frustrated and overwhelmed by the effort required to maintain consistency **PS-2:** Which makes me feel **insecure** and hesitant to **publish my recipes** Describe the emotions the result from experiencing the problems or barriers

2.2 Empathy Map Canvas

Target User

Aspiring food blogger or home cook with limited writing time.

Says

- "Writing takes more time than cooking."
- "I want professional-looking blogs."

Thinks

- "My content isn't engaging enough."

- “I need to post consistently.”

Does

- Searches online for blog format examples
- Spends hours formatting content

Feels

- Frustrated by writing effort
- Insecure about content quality

Gains Expected

- Faster content creation
- Structured blog output
- Increased publishing confidence

2.3 Brainstorming

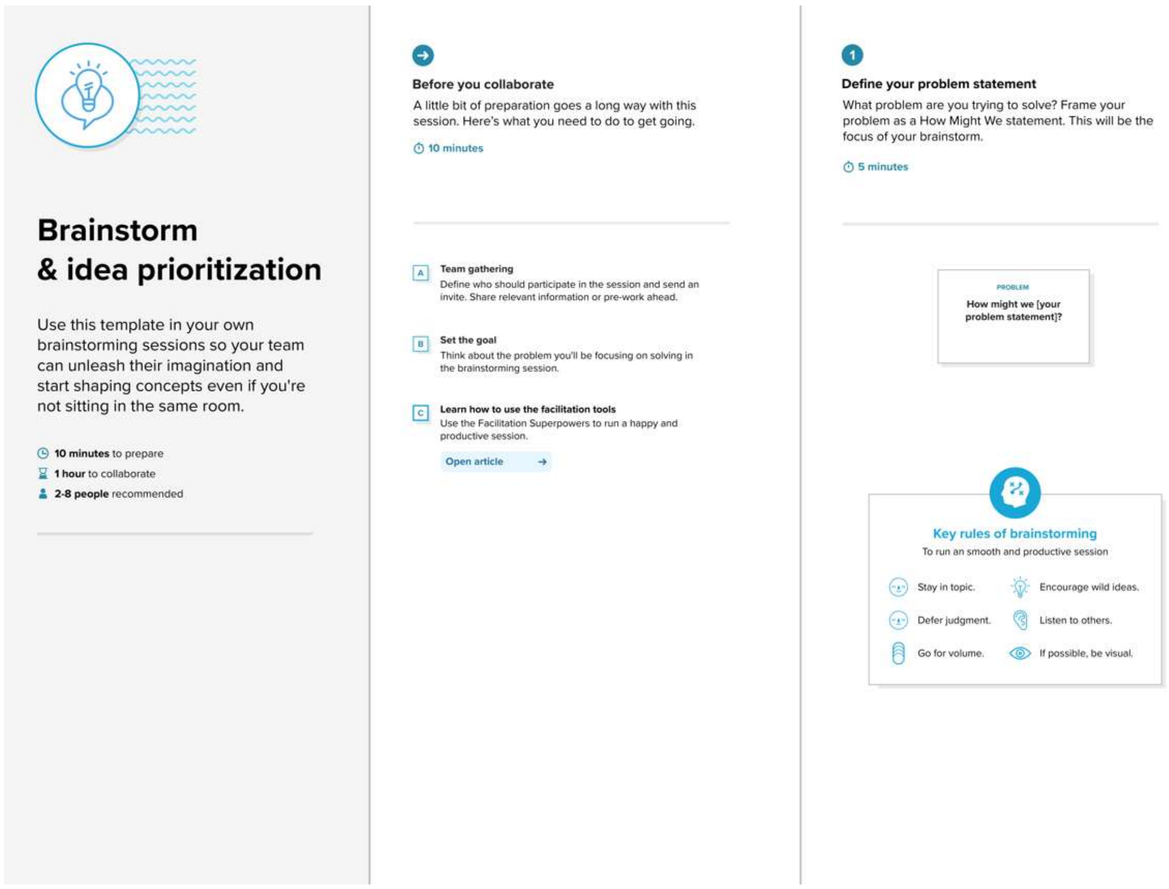
During ideation, the team evaluated multiple AI-based content generation ideas including:

- Travel blog generator
- Fitness plan generator
- Academic summary tool
- AI recipe generator

The AI recipe blog generator was selected due to:

- Clear user problem
- Feasible implementation
- Strong demonstration of Generative AI capability
- Practical real-world relevance

Features were prioritized based on impact and development effort.


Brainstorm
& idea prioritization

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.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

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

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 **Define your problem statement**
What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.
5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.	Encourage wild ideas.
Defer judgment.	Listen to others.
Go for volume.	If possible, be visual.

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map

1. User opens web application
2. Enters recipe topic
3. Selects word count
4. Clicks "Generate Recipe"
5. Views AI-generated blog
6. Downloads Markdown file

Pain Point Addressed:
Manual content drafting replaced with instant AI generation.

3.2 Solution Requirement

Functional Requirements

- Accept recipe topic input
- Accept word count selection
- Generate structured recipe blog via Gemini API
- Display generated content
- Provide Markdown download
- Show loading feedback
- Handle API errors gracefully

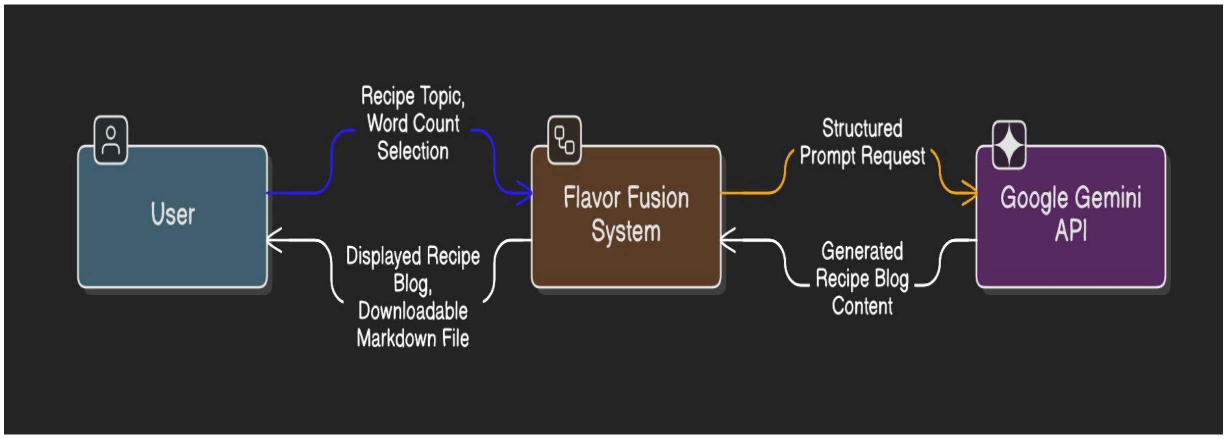
Non-Functional Requirements

- Usability: Simple and intuitive UI
- Performance: Generation within acceptable API latency
- Reliability: No system crash during API failure
- Security: API key stored securely
- Scalability: Cloud deployable architecture

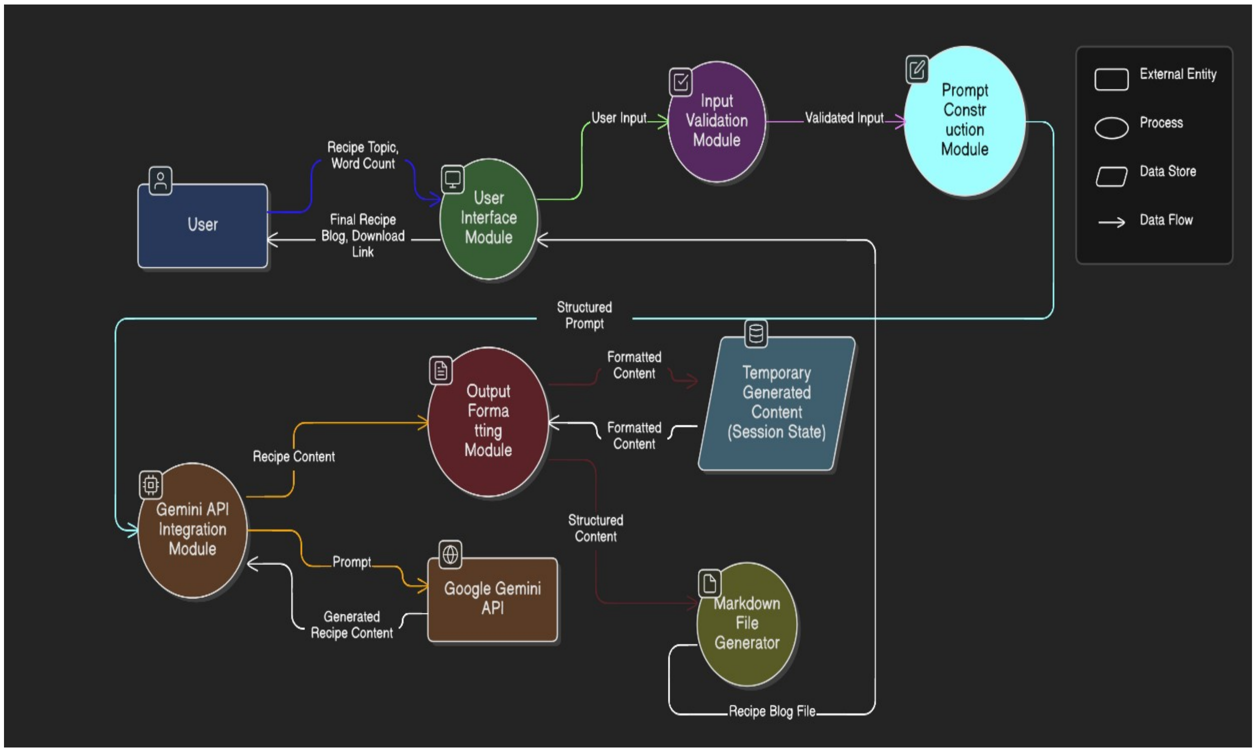
3.3 Data Flow Diagram (Description)

User → Streamlit UI → Application Logic → Gemini API
Gemini API → Application Logic → Content Formatter → Markdown Generator → User

DFD Level 0 (Industry Standard)


Recipe Topic, Word Count Selection
User
Flavor Fusion System
Structured Prompt Request
Google Gemini API
Displayed Recipe Blog, Downloadable Markdown File
Generated Recipe Blog Content

DFD Level 1 (Industry Standard)



The system follows a stateless architecture with no persistent database.

3.4 Technology Stack

Frontend: Streamlit
Backend: Python
AI Model: Google Gemini (gemini-flash-latest)
File Format: Markdown (.md)
Deployment: Local / Streamlit Cloud

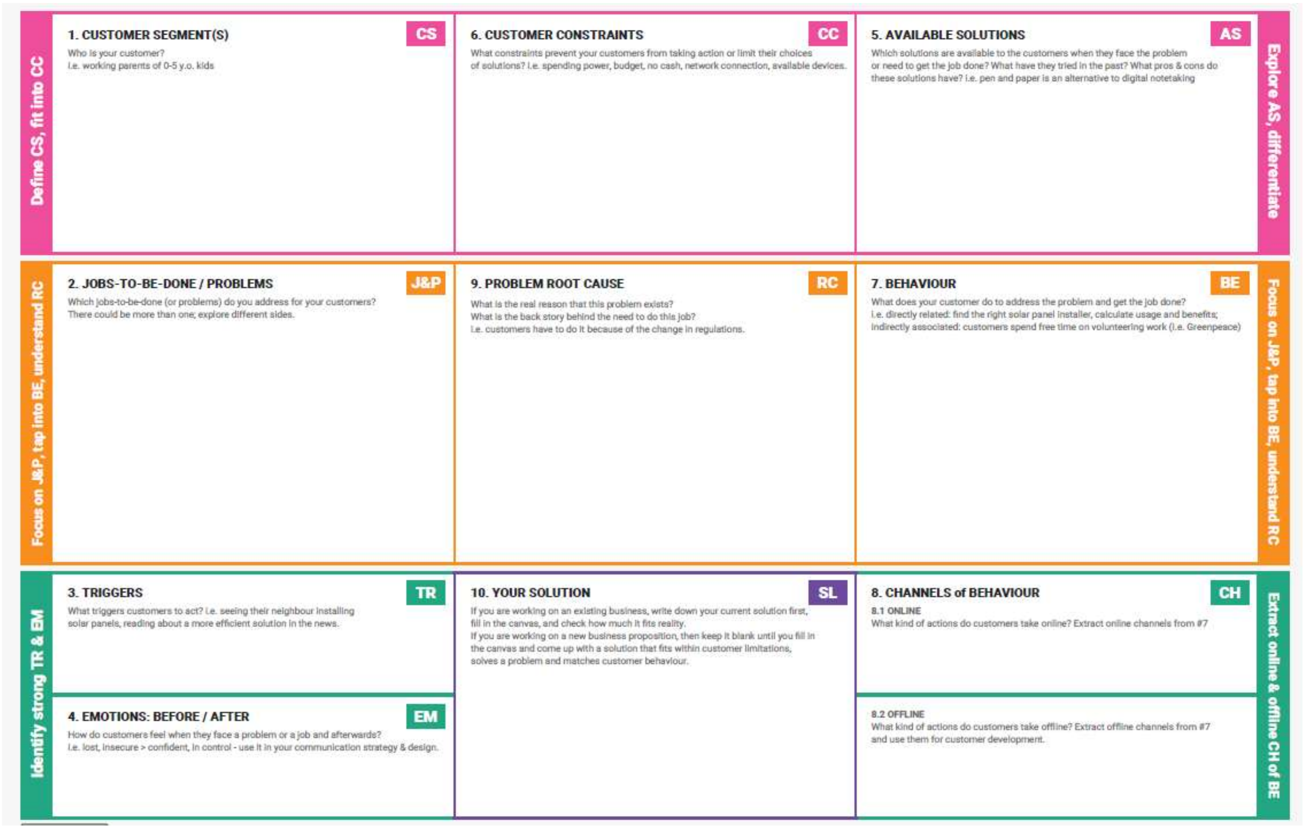
4. PROJECT DESIGN

4.1 Problem–Solution Fit

Problem:
Time-consuming manual recipe blog writing.

Solution:
AI-powered automated blog generation with structured formatting.

Fit Justification:
The solution directly reduces writing effort while maintaining professional output quality.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS
Who is your customer?
i.e. working parents of 0-5 y.o. kids

6. CUSTOMER CONSTRAINTS CC
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.

5. AVAILABLE SOLUTIONS AS
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

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

9. PROBLEM ROOT CAUSE RC
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.

7. BEHAVIOUR BE
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)

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR
What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

4. EMOTIONS: BEFORE / AFTER EM
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.

10. YOUR SOLUTION SL
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.

8. CHANNELS of BEHAVIOUR CH
8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Extract online & offline CH of BE

4.2 Proposed Solution

Flavor Fusion is a lightweight AI-powered web application that:

- Dynamically generates structured recipe blogs
- Provides customizable word length
- Offers export-ready Markdown files
- Enhances user engagement via interactive UI

The system bridges business need (content automation) with AI technology.

4.3 Solution Architecture

The architecture consists of four layers:

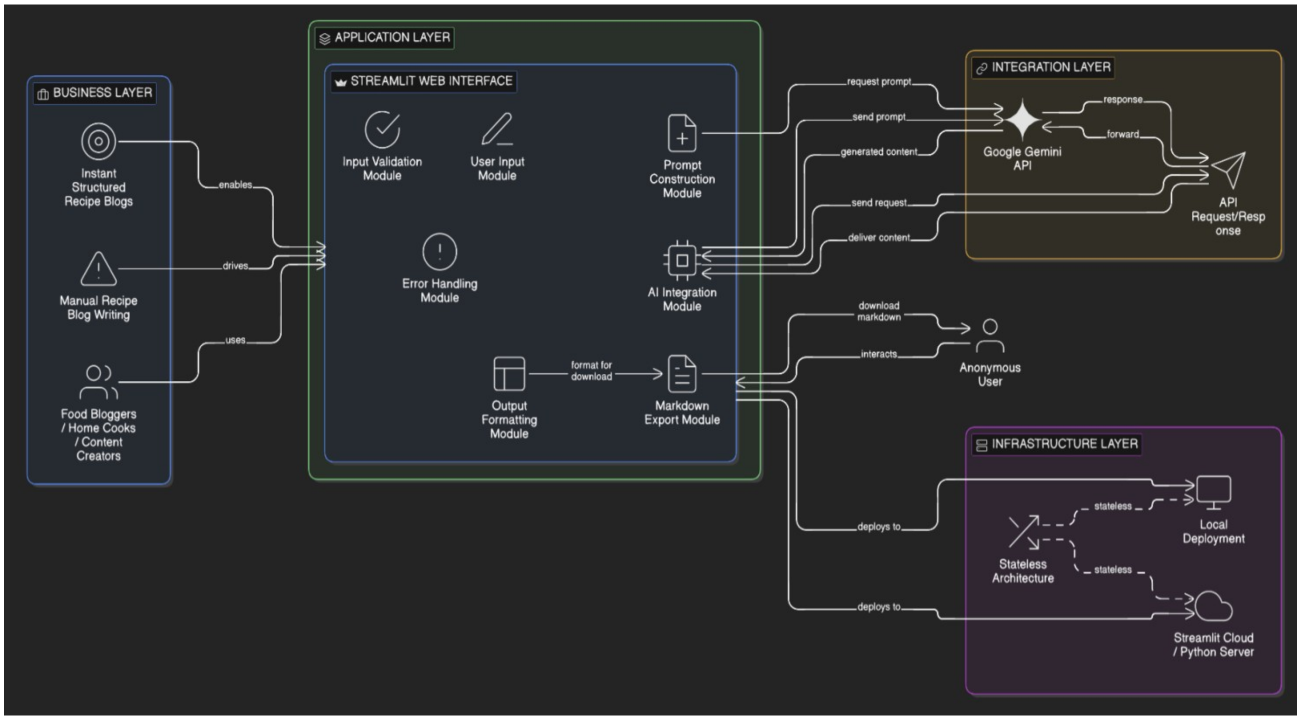
Business Layer:
User need for faster blog creation.

Application Layer:
Streamlit UI + Python logic modules.

Integration Layer:
Google Gemini API.

Infrastructure Layer:
Local or cloud deployment environment.

The application is stateless and scalable through horizontal cloud deployment.



5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Development was completed in two sprints:

Sprint 1:

- UI development
- Gemini API integration
- Prompt engineering

Sprint 2:

- Markdown export
- Loading indicator and joke display
- Error handling
- Testing and deployment

Average team velocity: 19 Story Points per sprint.

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

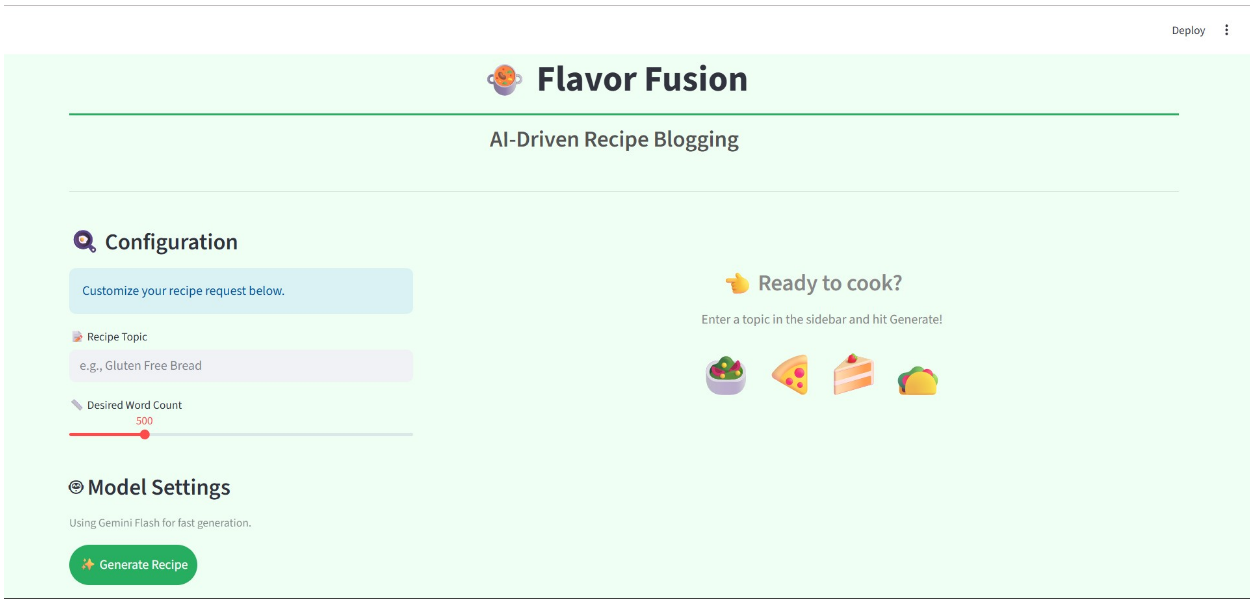
- AI generation time tested under multiple word count conditions.
- Short output (~200 words): 3–5 seconds.
- Long output (~1500 words): 5–8 seconds depending on API latency.
- No application crash during API timeout simulation.

System confirmed stable for UAT.

7. RESULTS

7.1 Output Screenshots

- Main interface screen



Customize your recipe request below.

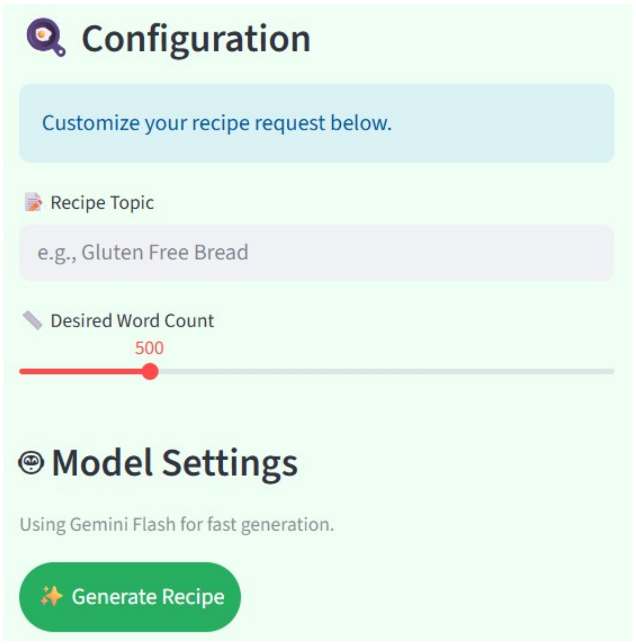
Recipe Topic: e.g., Gluten Free Bread

Desired Word Count: 500

Using Gemini Flash for fast generation.

Enter a topic in the sidebar and hit Generate!

- Input section



Configuration

Customize your recipe request below.

Recipe Topic

e.g., Gluten Free Bread

Desired Word Count

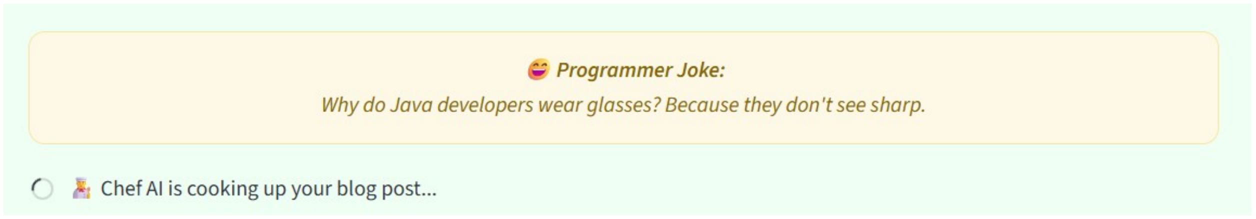
500

Model Settings

Using Gemini Flash for fast generation.

✨ Generate Recipe

- Waiting Time

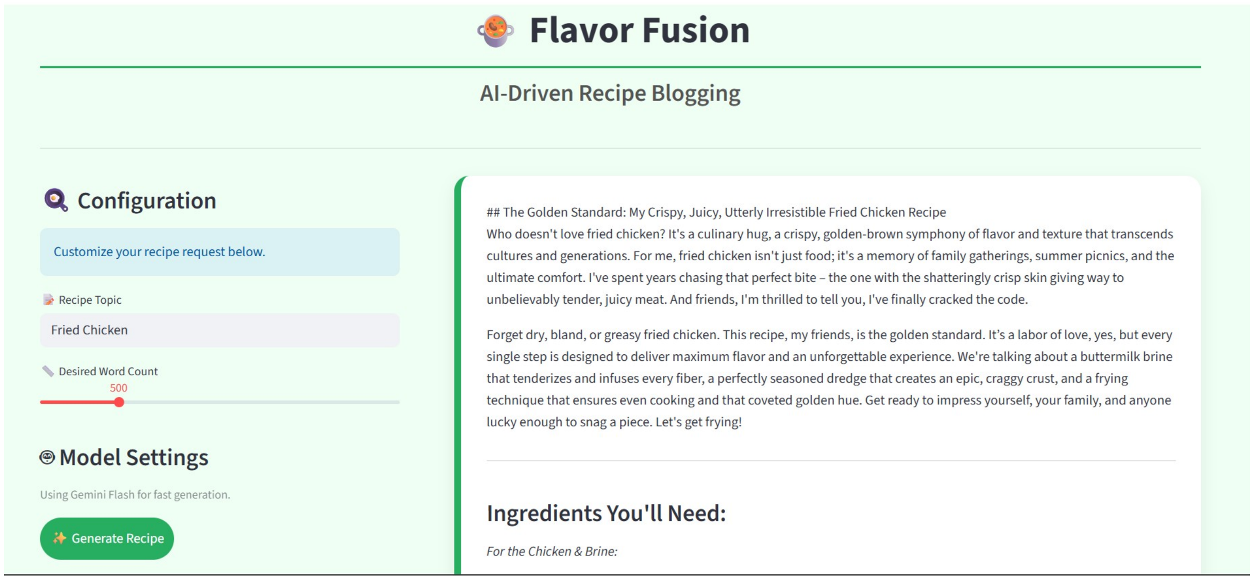


Programmer Joke:

Why do Java developers wear glasses? Because they don't see sharp.

Chef AI is cooking up your blog post...

- Generated recipe output



Flavor Fusion

AI-Driven Recipe Blogging

Configuration

Customize your recipe request below.

Recipe Topic

Fried Chicken

Desired Word Count

500

Model Settings

Using Gemini Flash for fast generation.

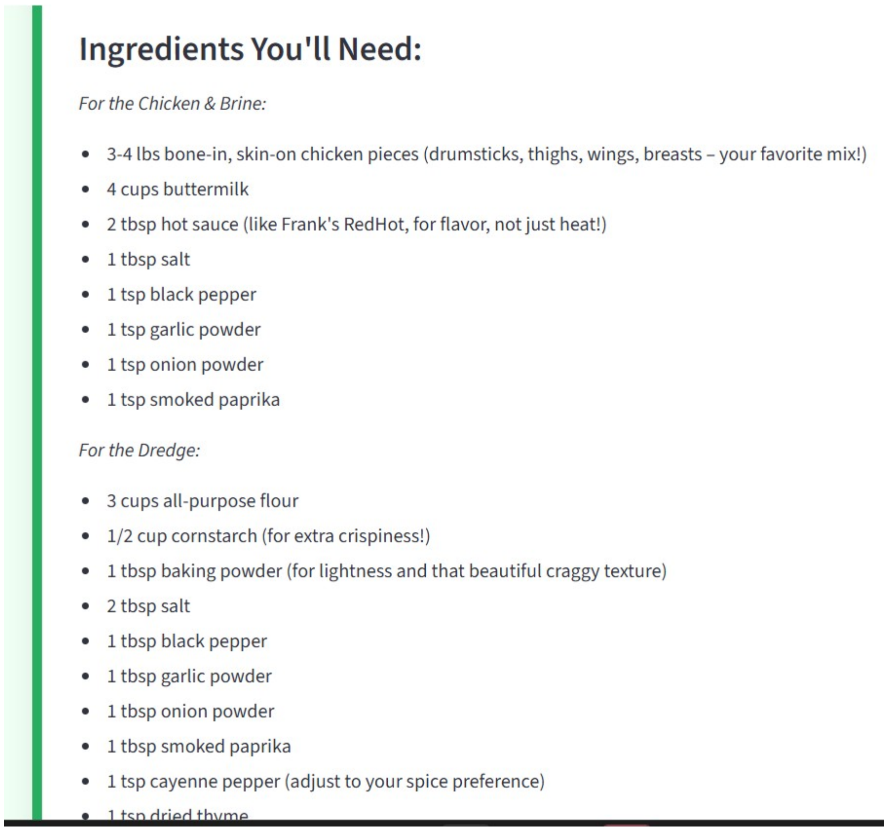
✨ Generate Recipe

The Golden Standard: My Crispy, Juicy, Utterly Irresistible Fried Chicken Recipe
Who doesn't love fried chicken? It's a culinary hug, a crispy, golden-brown symphony of flavor and texture that transcends cultures and generations. For me, fried chicken isn't just food; it's a memory of family gatherings, summer picnics, and the ultimate comfort. I've spent years chasing that perfect bite – the one with the shatteringly crisp skin giving way to unbelievably tender, juicy meat. And friends, I'm thrilled to tell you, I've finally cracked the code.

Forget dry, bland, or greasy fried chicken. This recipe, my friends, is the golden standard. It's a labor of love, yes, but every single step is designed to deliver maximum flavor and an unforgettable experience. We're talking about a buttermilk brine that tenderizes and infuses every fiber, a perfectly seasoned dredge that creates an epic, craggy crust, and a frying technique that ensures even cooking and that coveted golden hue. Get ready to impress yourself, your family, and anyone lucky enough to snag a piece. Let's get frying!

Ingredients You'll Need:

For the Chicken & Brine:



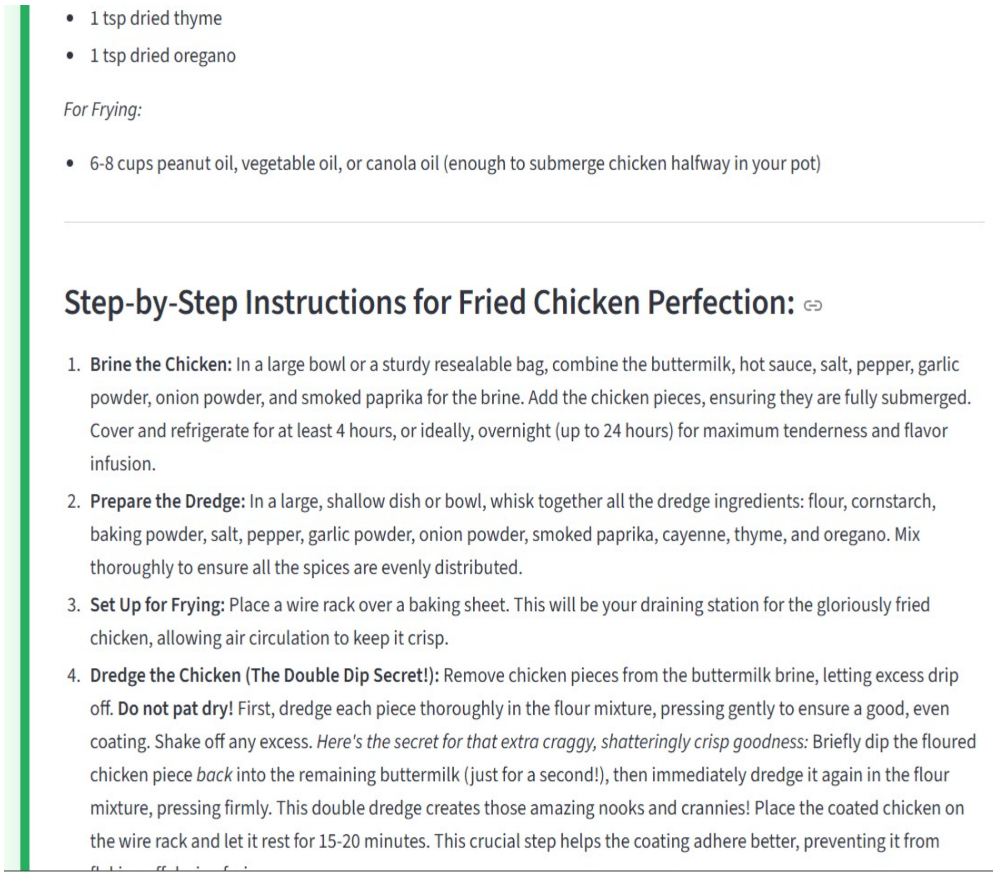
Ingredients You'll Need:

For the Chicken & Brine:

- 3-4 lbs bone-in, skin-on chicken pieces (drumsticks, thighs, wings, breasts – your favorite mix!)
- 4 cups buttermilk
- 2 tbsp hot sauce (like Frank's RedHot, for flavor, not just heat!)
- 1 tbsp salt
- 1 tsp black pepper
- 1 tsp garlic powder
- 1 tsp onion powder
- 1 tsp smoked paprika

For the Dredge:

- 3 cups all-purpose flour
- 1/2 cup cornstarch (for extra crispiness!)
- 1 tbsp baking powder (for lightness and that beautiful craggy texture)
- 2 tbsp salt
- 1 tbsp black pepper
- 1 tbsp garlic powder
- 1 tbsp onion powder
- 1 tbsp smoked paprika
- 1 tsp cayenne pepper (adjust to your spice preference)
- 1 tsp dried thyme



- 1 tsp dried thyme

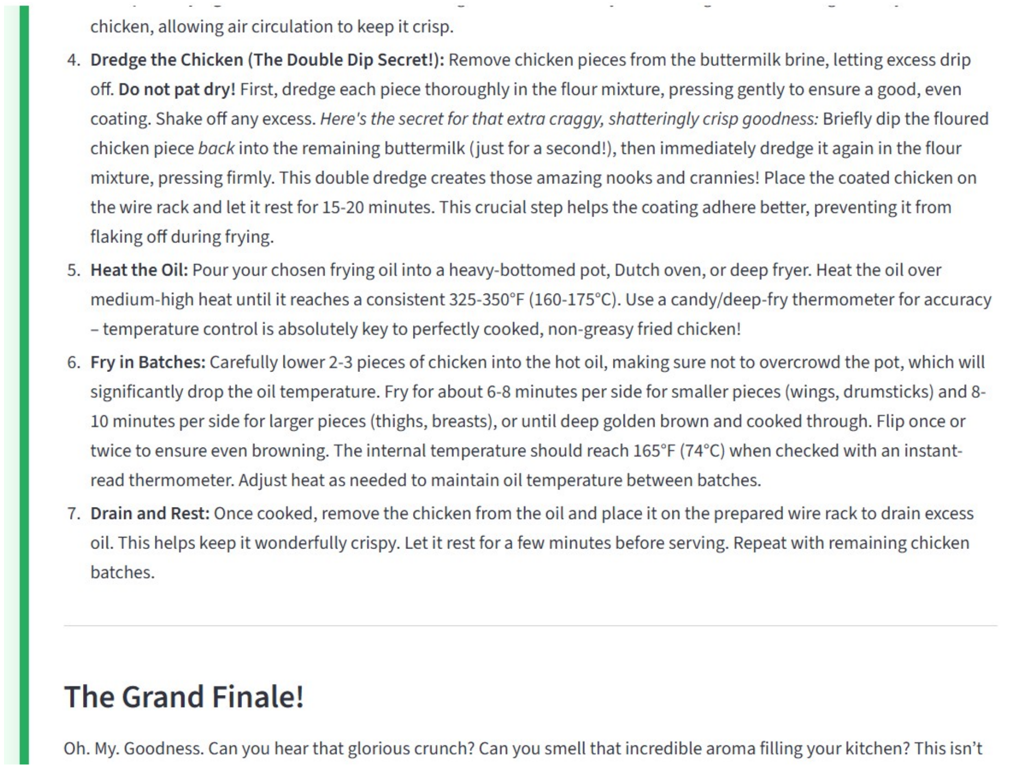
- 1 tsp dried oregano

For Frying:

- 6-8 cups peanut oil, vegetable oil, or canola oil (enough to submerge chicken halfway in your pot)

Step-by-Step Instructions for Fried Chicken Perfection:

1. **Brine the Chicken:** In a large bowl or a sturdy resealable bag, combine the buttermilk, hot sauce, salt, pepper, garlic powder, onion powder, and smoked paprika for the brine. Add the chicken pieces, ensuring they are fully submerged. Cover and refrigerate for at least 4 hours, or ideally, overnight (up to 24 hours) for maximum tenderness and flavor infusion.
2. **Prepare the Dredge:** In a large, shallow dish or bowl, whisk together all the dredge ingredients: flour, cornstarch, baking powder, salt, pepper, garlic powder, onion powder, smoked paprika, cayenne, thyme, and oregano. Mix thoroughly to ensure all the spices are evenly distributed.
3. **Set Up for Frying:** Place a wire rack over a baking sheet. This will be your draining station for the gloriously fried chicken, allowing air circulation to keep it crisp.
4. **Dredge the Chicken (The Double Dip Secret!):** Remove chicken pieces from the buttermilk brine, letting excess drip off. **Do not pat dry!** First, dredge each piece thoroughly in the flour mixture, pressing gently to ensure a good, even coating. Shake off any excess. *Here's the secret for that extra craggy, shatteringly crisp goodness:* Briefly dip the floured chicken piece *back* into the remaining buttermilk (just for a second!), then immediately dredge it again in the flour mixture, pressing firmly. This double dredge creates those amazing nooks and crannies! Place the coated chicken on the wire rack and let it rest for 15-20 minutes. This crucial step helps the coating adhere better, preventing it from

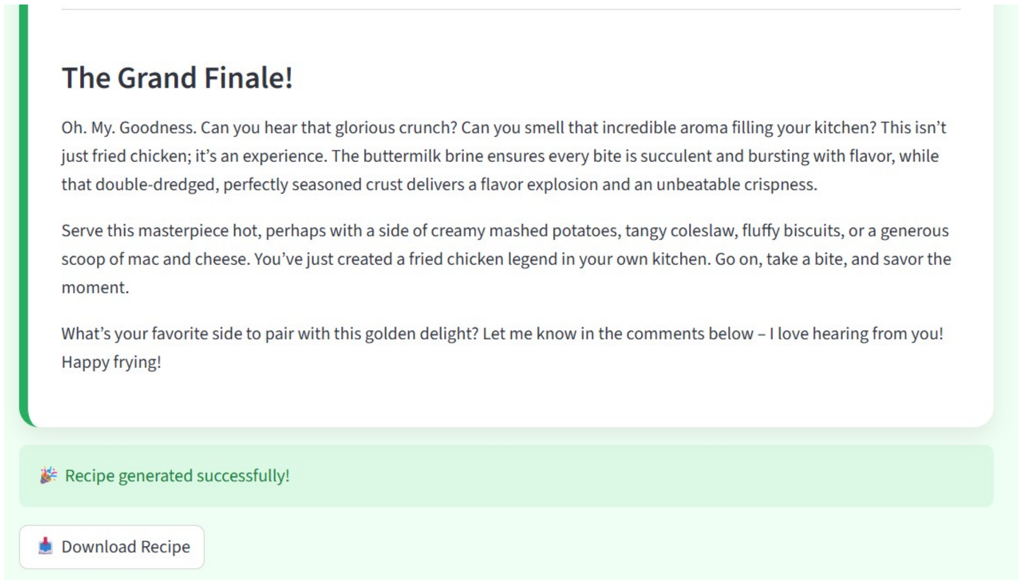


chicken, allowing air circulation to keep it crisp.

4. **Dredge the Chicken (The Double Dip Secret!):** Remove chicken pieces from the buttermilk brine, letting excess drip off. **Do not pat dry!** First, dredge each piece thoroughly in the flour mixture, pressing gently to ensure a good, even coating. Shake off any excess. *Here's the secret for that extra craggy, shatteringly crisp goodness:* Briefly dip the floured chicken piece *back* into the remaining buttermilk (just for a second!), then immediately dredge it again in the flour mixture, pressing firmly. This double dredge creates those amazing nooks and crannies! Place the coated chicken on the wire rack and let it rest for 15-20 minutes. This crucial step helps the coating adhere better, preventing it from flaking off during frying.
5. **Heat the Oil:** Pour your chosen frying oil into a heavy-bottomed pot, Dutch oven, or deep fryer. Heat the oil over medium-high heat until it reaches a consistent 325-350°F (160-175°C). Use a candy/deep-fry thermometer for accuracy – temperature control is absolutely key to perfectly cooked, non-greasy fried chicken!
6. **Fry in Batches:** Carefully lower 2-3 pieces of chicken into the hot oil, making sure not to overcrowd the pot, which will significantly drop the oil temperature. Fry for about 6-8 minutes per side for smaller pieces (wings, drumsticks) and 8-10 minutes per side for larger pieces (thighs, breasts), or until deep golden brown and cooked through. Flip once or twice to ensure even browning. The internal temperature should reach 165°F (74°C) when checked with an instant-read thermometer. Adjust heat as needed to maintain oil temperature between batches.
7. **Drain and Rest:** Once cooked, remove the chicken from the oil and place it on the prepared wire rack to drain excess oil. This helps keep it wonderfully crispy. Let it rest for a few minutes before serving. Repeat with remaining chicken batches.

The Grand Finale!

Oh. My. Goodness. Can you hear that glorious crunch? Can you smell that incredible aroma filling your kitchen? This isn't



The Grand Finale!

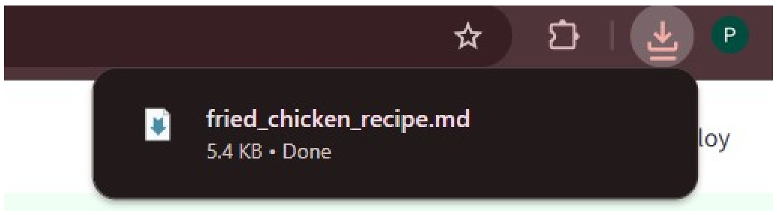
Oh. My. Goodness. Can you hear that glorious crunch? Can you smell that incredible aroma filling your kitchen? This isn't just fried chicken; it's an experience. The buttermilk brine ensures every bite is succulent and bursting with flavor, while that double-dredged, perfectly seasoned crust delivers a flavor explosion and an unbeatable crispness.

Serve this masterpiece hot, perhaps with a side of creamy mashed potatoes, tangy coleslaw, fluffy biscuits, or a generous scoop of mac and cheese. You've just created a fried chicken legend in your own kitchen. Go on, take a bite, and savor the moment.

What's your favorite side to pair with this golden delight? Let me know in the comments below – I love hearing from you! Happy frying!

🎉 Recipe generated successfully!

- Markdown download confirmation



fried_chicken_recipe.md
5.4 KB • Done

Generated outputs demonstrated:

- Structured blog format
- Correct word count range
- Professional presentation

8. ADVANTAGES & DISADVANTAGES

Advantages

- Significant time savings
- Structured professional output
- Lightweight architecture
- Easy deployment
- Beginner-friendly UI

Disadvantages

- Dependent on external AI API
- Requires internet connection
- API latency may vary
- No persistent user storage

9. CONCLUSION

Flavor Fusion successfully demonstrates practical integration of Generative AI into a real-world web application.

The system reduces manual effort in content creation while maintaining structured output quality. It validates the feasibility of AI-driven automation for creative blogging tasks.

The project achieves strong problem–solution alignment and demonstrates scalable AI application architecture.

10. FUTURE SCOPE

- Multi-language recipe generation
- SEO keyword optimization
- AI image generation for recipes
- Nutrition analysis integration
- User account and recipe history storage
- Cloud auto-scaling deployment

11. APPENDIX

Source Code (app.py):

```
import streamlit as st

import google.generativeai as genai

import random

# Configure Google Generative AI API key

# Ideally, this should be stored in st.secrets or environment variables

api_key = "AIzaSyDCG7_AE9N3ocS8pT3lQWyP2icEnHsGMCE"

genai.configure(api_key=api_key)

# Generation Configuration

generation_config = {

  "temperature": 0.75,

  "top_p": 0.95,

  "top_k": 64,

  "max_output_tokens": 8192,

  "response_mime_type": "text/plain",

}

# Function to get a random programming joke

def get_joke():

  jokes = [

    "Why do programmers prefer dark mode? Because light attracts bugs.",

    "Why did the developer go broke? Because he used up all his cache.",

    "How many programmers does it take to change a light bulb? None. It's a hardware problem.",

    "What is a programmer's favorite hangout place? Foo Bar.",
        "Why do Java developers wear glasses? Because they don't see sharp.",
        "A SQL query walks into a bar, walks up to two tables and asks... 'Can I join you?'",
        "Why was the JavaScript developer sad? Because he didn't know how to 'null' his feelings.",
        "What do you call a programmer from Finland? Nerdic.",
        "Why did the computer go to the doctor? Because it had a virus!",
        "There are 10 types of people in the world: those who understand binary, and those who don't."
    ]
    return random.choice(jokes)
# Function to generate recipe blog
def recipe_generation(topic, word_count):
    try:
        model = genai.GenerativeModel(
            # model_name="gemini-flash-latest",
            model_name="gemini-2.5-flash",
            generation_config=generation_config,
        )
        prompt = f"""
        You are a professional food blogger. Create a detailed and engaging recipe blog post about "{topic}".
        The blog post should be approximately {word_count} words long.
        Include a catchy title, an introduction, ingredients list, step-by-step instructions, and a conclusion.
        Make it fun and appetizing.
        """
        response = model.generate_content(prompt)
        return response.text
    except Exception as e:
        return f"Error creating recipe: {str(e)}"
# Streamlit UI
def main():
    st.set_page_config(
        page_title="Flavor Fusion",
        page_icon="□",
        layout="wide",
        initial_sidebar_state="expanded"
    )
    st.markdown("""
<style>
.stApp {
    background-color: #F0FFF4;
}
h1 {
    color: #27AE60;
    text-align: center;
    font-family: 'Segoe UI', sans-serif;
    border-bottom: 3px solid #27AE60;
    padding-bottom: 10px;
}
.stButton > button {
    background-color: #27AE60;
    color: white;
    border-radius: 25px;
    padding: 12px;
    border: none;
    font-weight: bold;
    transition: 0.3s;
}
.stButton > button:hover {
    background-color: #1E8449;
    transform: scale(1.05);
}
.stTextInput input {
    border-radius: 12px;
}
.recipe-card {
    background-color: white;
    padding: 30px;
    border-radius: 18px;
    box-shadow: 0 8px 20px rgba(0,0,0,0.08);
    border-left: 8px solid #27AE60;
}
```

```
.joke-box {
    background-color: #FEF9E7;
    color: #856404;
    padding: 15px;
    border-radius: 12px;
    border: 1px solid #F9E79F;
    text-align: center;
    font-style: italic;
    margin-bottom: 20px;
}
</style>
""", unsafe_allow_html=True)
    st.title("□ Flavor Fusion")
    st.markdown("<h3 style='text-align: center; color: #555;'>AI-Driven Recipe Blogging</h3>", unsafe_allow_html=True)
    st.markdown("---")
    # Layout with columns
    col1, col2 = st.columns([1, 2], gap="large")
    with col1:
        st.markdown("### □ Configuration")
        st.info("Customize your recipe request below.")
        topic = st.text_input("□ Recipe Topic", placeholder="e.g., Gluten Free Bread")
        word_count = st.slider("□ Desired Word Count", min_value=100, max_value=2000, value=500, step=50)
        st.markdown("### □ Model Settings")
        st.caption("Using Gemini Flash for fast generation.")
        generate_btn = st.button("✨ Generate Recipe")
    with col2:
        if generate_btn:
            if topic:
                # Joke display
                joke = get_joke()
                # Create a placeholder for the joke
                joke_placeholder = st.empty()
                joke_placeholder.markdown(f"""
                <div class="joke-box">
                    □ <b>Programmer Joke:</b><br>{joke}
                </div>
                """, unsafe_allow_html=True)
                with st.spinner("□□ Chef AI is cooking up your blog post..."):
                    recipe_content = recipe_generation(topic, word_count)
                joke_placeholder.empty()
                if recipe_content and "Error" not in recipe_content:
                    st.markdown(f"""
                    <div class="recipe-card">
                        {recipe_content}
                    </div>
                    """, unsafe_allow_html=True)
                    st.success("□ Recipe generated successfully!")
                    st.download_button(
                        label="□ Download Recipe",
                        data=recipe_content,
                        file_name=f"{topic.replace(' ', '_').lower()}_recipe.md",
                        mime="text/markdown",
                        help="Save this recipe to your computer"
                    )
                else:
                    st.error("□ " + recipe_content)
            else:
                st.warning("□ Please enter a recipe topic first.")
        else:
            st.markdown("""
            <div style='text-align: center; padding: 50px; color: #888;'>
                <h3>□ Ready to cook?</h3>
                <p>Enter a topic in the sidebar and hit Generate!</p>
                <div style='font-size: 50px;'>□ □ □ □</div>
            </div>
            """, unsafe_allow_html=True)
if __name__ == "__main__":
    main()
```

GitHub Repository

https://github.com/PattapuChohitha/Flavour-Fusion-AI-Driven-Recipe-Blogging

Project Demo Link

https://drive.google.com/file/d/1Hz9EQJXTTLktwJt50uUEF7UBd7bYIDuV/view?usp=sharing