## horizontal line



### **Flavour Fusion: AI-Driven Recipe Blogging**

09.03.2025.

Your Name

SRIKAKULA VENKATA LAKSHMANA SWAMY,

Branch:Computer science of Engineering(AI),

SVR engineering college,

Ayyalur

# Overview

flavour Fusion: AI-Driven Recipe Blogging is a web application that leverages Google's Generative AI to create unique and customized recipe blogs. The app provides users with the ability to input a topic and specify the desired word count for their recipe blog. Using the specified parameters, the AI generates detailed and engaging recipe content. Additionally, the app includes a fun feature where it tells a programmer joke to entertain users while the AI is generating the content.

Scenario 1: Creating a Vegan Recipe Blog

Scenario 2: Crafting a Quick Weeknight Dinner Recipe Blog

Scenario 3: Developing a Gluten-Free Baking Recipe Blog

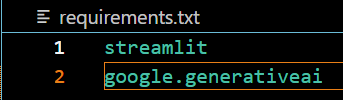
# Architecture:

# Project Flow:

* 1. Users input a topic and specify the desired length of the blog post through the Streamlit UI.
  2. The input topic and length are sent to the Gemini 1.5 Flash language model, which is integrated into the backend.
  3. Gemini 1.5 Flash processes the input and generates a blog post based on the user's specifications.
  4. The model autonomously creates a well-structured, engaging blog post tailored to the specified topic and word count.
  5. The generated blog post is sent back to the frontend for display on the Streamlit app.
  6. Users can customize the blog post further if desired and export or copy the content for their use.
* To accomplish this, we have to complete all the activities listed below,
  1. Initialize Gemini 1.5 Flash:
  2. Generate Gemini 1.5 Flash API
  3. Initialize the pre-trained model
  4. Interfacing with Pre-trained Model
     + Blog Generation
  5. Model Deployment
     + Deploy the application using Streamlit

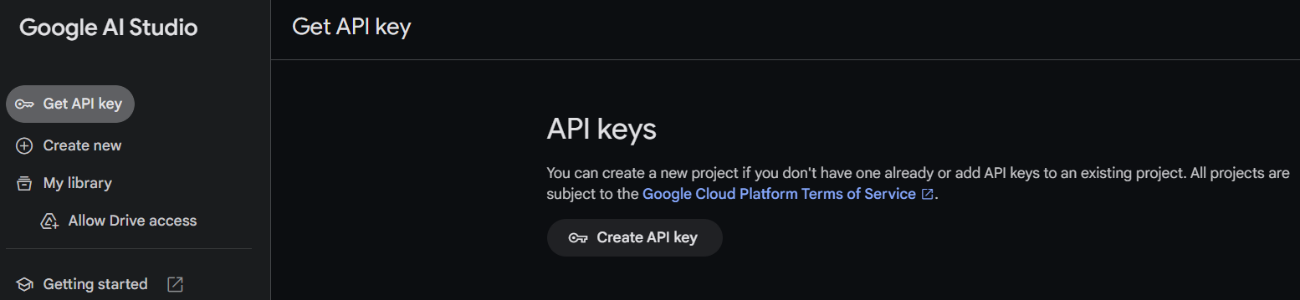
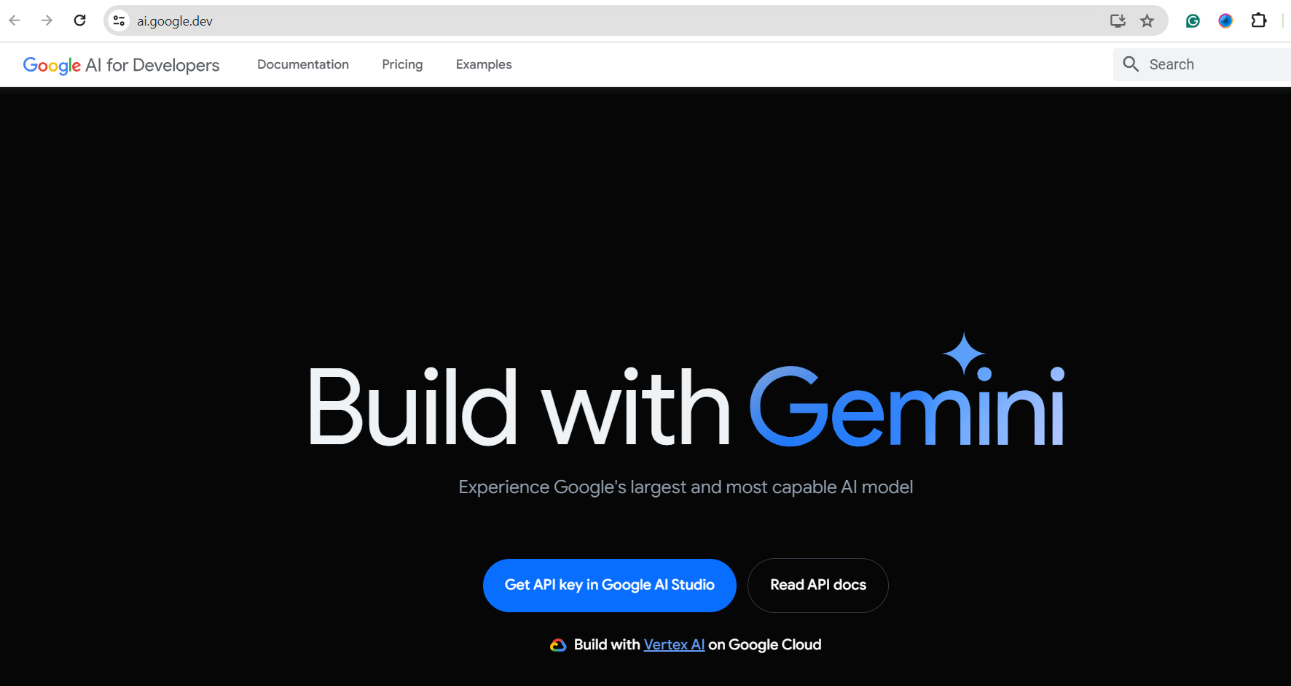
# Requirements Specification:

Specifying the required libraries in the requirements.txt file ensures seamless setup and reproducibility of the project environment



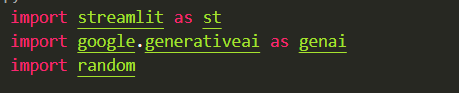
# Generate Google API key:

* Click on the link (<https://developers.generativeai.google/>).
* Then click on “Get API key in Google AI Studio”.
* Click on “Get API key” from the right navigation menu.
* Now click on “Create API key”. (Refer the below images)
* Copy the API key.



# Initialize the pre-trained model:

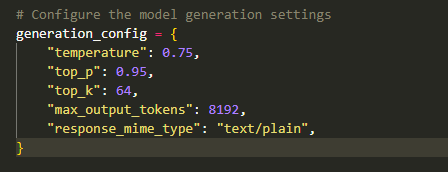
Import necessary files:



## Activity 2.2: Configuration of the Gemini 1.5 Flash API :

Configuring the API key: The configure function is used to set up or configure the Google API with an API key. The provided API key, in this case, is“AIzaSyB5U5-f1edVl99djSKEcqDoFLcI2l6uYyI”

## Activity 2.3: Define the model to be used :



# Joke Generation (get\_joke() Function):

# Function to generate a joke

def get\_joke():

jokes = [

"Why don't programmers like nature? It has too many bugs.",

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

"Why was the JavaScript developer sad? Because he didn't know how to 'null' his feelings.",

"Why don't programmers like nature? It has too many bugs.",

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

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

"Why was the JavaScript developer sad? Because he didn't know how to 'null' his feelings.",

"Why do Python programmers prefer using snake\_case? Because it's easier to read!",

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

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

"Why do programmers always mix up Christmas and Halloween? Because Oct 31 = Dec 25.",

"Why did the programmer get kicked out of the beach? Because he kept using the 'C' language!",

"Why was the computer cold? It left its Windows open."

]

return random.choice(jokes)

# Recipe Generation (recepie\_generation() Function):

def recepie\_generation(user\_input, word\_count):

"""

Function to generate a blog based on user input and word count.

Parameters:

user\_input (str): The topic for the blog.

word\_count (int): The desired number of words for the blog.

Returns:

str: The generated blog content.

"""

#Display a message while the blog is being generated

st.write("# Generating your recepie...")

st.write("While I work on creating your blog, here's a little joke to keep you entertained:\n\n\*\* get\_joke()\*\*")

#Start a chat session with the input supis and surd count

chat\_session = model.start\_chat()

history=[

{

"role": "user",

"parts": [f"Write a recepie based on the input topic: {user\_input} and number of words: {word\_count}\n"]

}

]

#Generate a response for the now έπους

response = chat\_session.send\_message(user\_input)

st.success(" Your recepie is ready!")

return response.text

except Exception as e:

st.error(f"Error generating blog: {e}")

return None

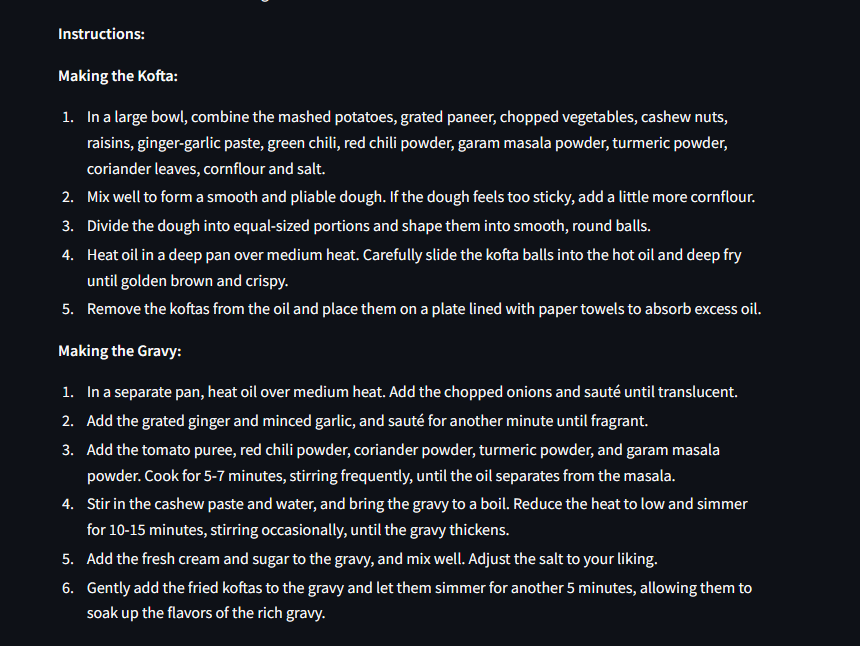
# Procedure for implementation of program:

1. Generates a recipe based on user input and a specified word count.
2. user\_input (str): The topic or theme for the recipe.
3. word\_count (int): The desired length of the recipe in words.
4. Displays a message indicating that the recipe is being generated.
5. Calls get\_joke() to display a joke to the user while waiting.
6. Starts a chat session using the Gemini 1.5 Flash model, passing the user input and word count.
7. Attempts to generate the recipe and returns the generated text if successful.
8. Handles any exceptions by displaying an error message.
9. This function is triggered when the user clicks the "Generate recipe" button in the Streamlit interface.

# outputs :



# 



# 

# Goals:

**1. Develop an AI-Powered Recipe Generation Engine**

**2. Enhance User Engagement Through Interactive Features**

**3. Ensure Nutritional Accuracy and Dietary Compliance**

**4. Streamline Content Creation and Management**

**5. Optimize User Experience with Personalized Recommendations**

**6. Facilitate Seamless Integration with Smart Kitchen Appliances**

# Conclusion:

The "Flavour Fusion: AI-Driven Recipe Blogging" project exemplifies the seamless integration of artificial intelligence into culinary content creation, enabling the generation of unique and customized recipe blogs. By leveraging advanced AI technologies, this initiative not only enhances user engagement but also sets a new standard for innovation in the food blogging industry.

# Thankyou