Name : Mahak Farhan

Roll no : SU92-BSAIM-F23-068

Semester project

Session 2023-2027

BS Artificial Intelligence

Department of Software Engineering

Faculty of Computer Science & Information Technology

Superior University, Lahore

Fall 2025

ChefBot

Objective

The goal of this project is to build an intelligent web application that:

- Generates a creative recipe and cooking instructions based on user-provided ingredients.
- Uses natural language processing (Groq API with LLaMA-3 model) to create the recipe.
- Uses image generation (Stable Diffusion) to visualize the dish.
- Allows users to save the recipe for later use.

Tools & Technologies Used

Tool/Technology	Purpose
Flask	Web framework for building the application
Groq API (LLaMA-3)	Generate recipe name and instructions
Stable Diffusion	Generate food images from text prompts
pyngrok	Make the local Flask app publicly available
CSV	Store saved recipes

Working

Step 1: Input Ingredients

The user enters a list of ingredients on the homepage.

Step 2: Generate Recipe

- A prompt is sent to **Groq's LLaMA-3** model via API.
- The model returns a creative **recipe name** and **step-by-step instructions**.

Step 3: Generate Image

- The recipe name is sent as a prompt to **Stable Diffusion**.
- A realistic dish image is generated and shown on the homepage.

Step 4: Display Recipe

- The recipe name, image, and steps are displayed in a Bootstrap-styled HTML page.
- The user can either:
  - **Save** the recipe to a CSV file.
  - **Try another** recipe.

Code

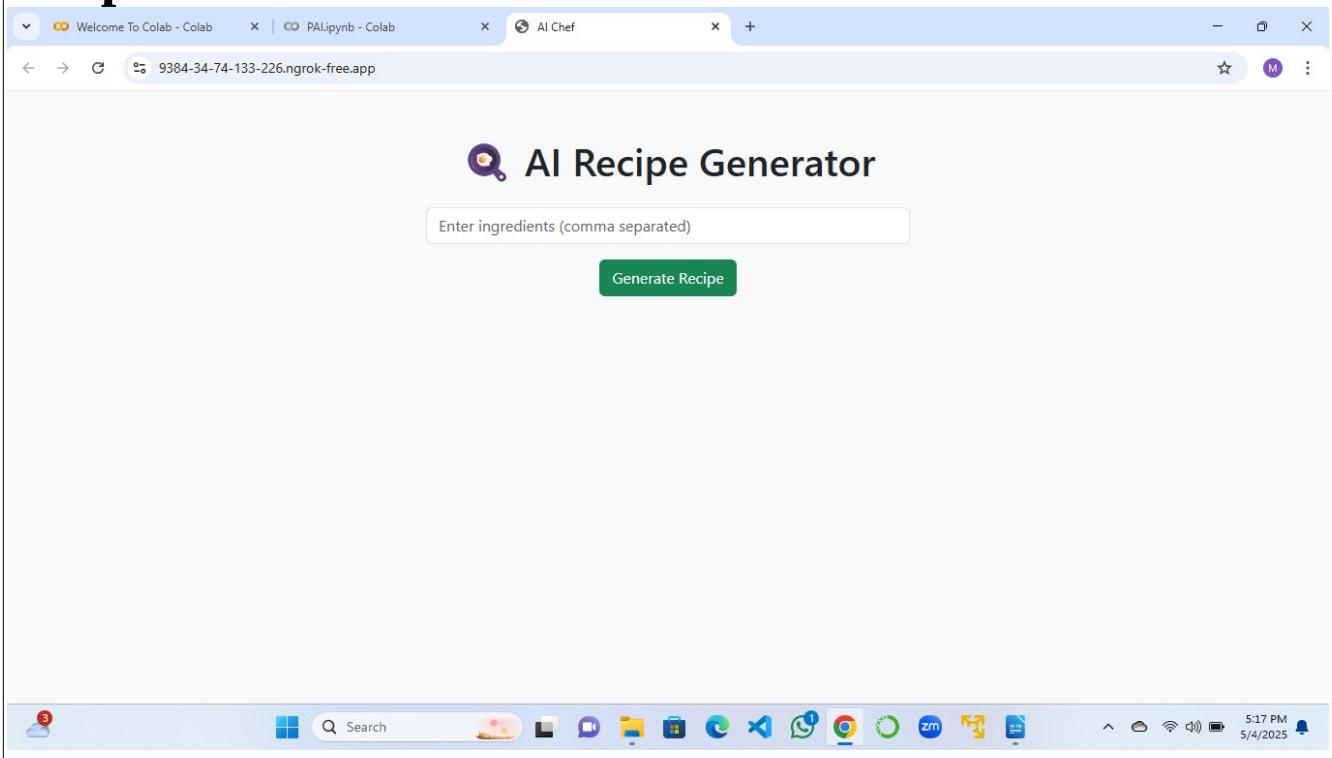
```
import os
from pyngrok import ngrok, conf
from flask import Flask, request, render_template_string
import torch, csv, requests
from diffusers import StableDiffusionPipeline

# Set tokens
```

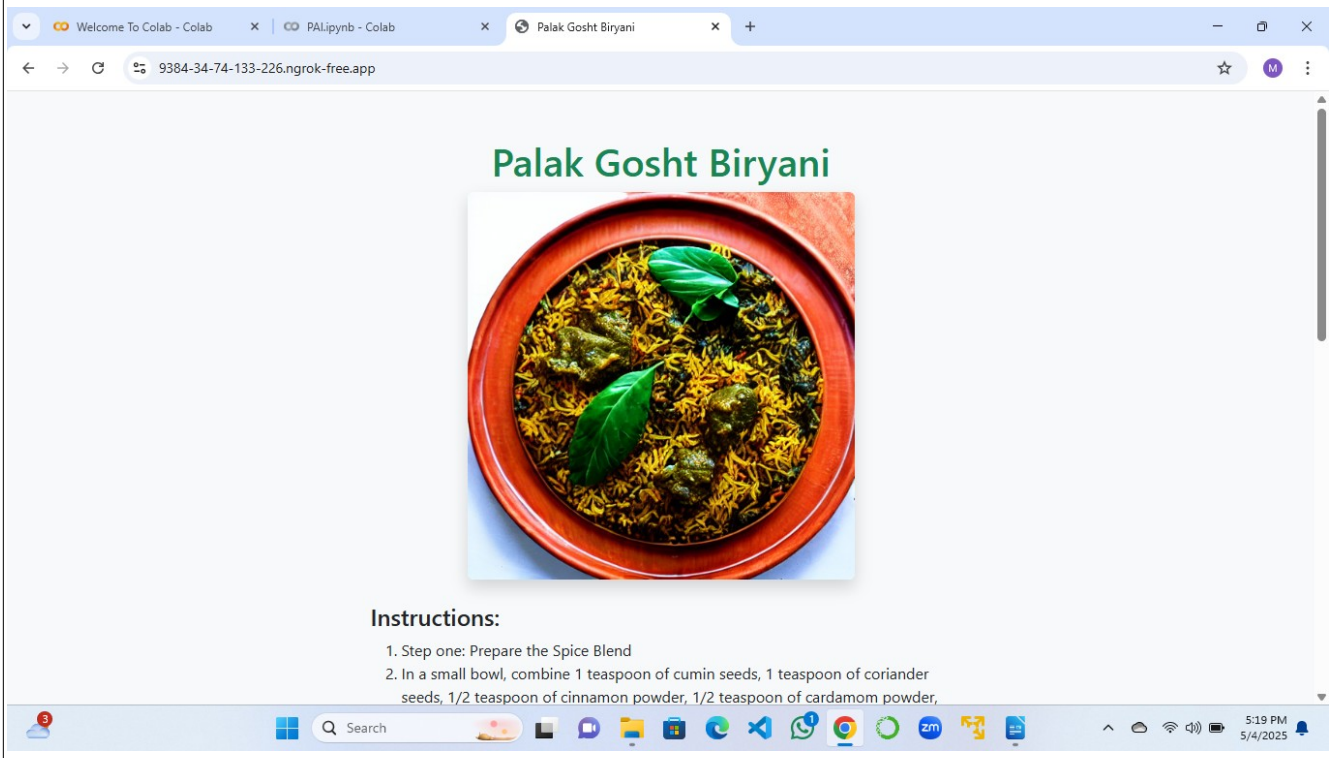
```
os.environ['GROQ_API_KEY'] = "Your_Groq_Key"
os.environ['HUGGINGFACE_TOKEN'] = "Your_HF_Token"
conf.get_default().auth_token = "Your_auth_token"
# Flask app
app = Flask(__name__)

# Load image model
pipe = StableDiffusionPipeline.from_pretrained(
    "runwayml/stable-diffusion-v1-5",
    torch_dtype=torch.float16 if torch.cuda.is_available() else torch.float32,
    use_auth_token=os.getenv("HUGGINGFACE_TOKEN")
).to("cuda" if torch.cuda.is_available() else "cpu")

def generate_recipe(ingredients):
    prompt = f"""
    You are a chef assistant. Create a human friendly dish name from: {', '.join(ingredients)}.
    Then write cooking steps.

    Format:
    Recipe Name: <name>
    Instructions:
    Step one...
    """
    res = requests.post(
        "https://api.groq.com/openai/v1/chat/completions",
        headers={
            "Authorization": f"Bearer {os.getenv('GROQ_API_KEY')}",
            "Content-Type": "application/json"
        },
        json={
            "model": "llama3-8b-8192",
            "messages": [{"role": "user", "content": prompt}]
        }
    )
    content = res.json()["choices"][0]["message"]["content"]
    lines = [l.strip() for l in content.split('\n') if l.strip()]
    name, steps = "", []
    for l in lines:
        if l.lower().startswith("recipe name:"):
            name = l.split(":", 1)[1].strip()
        elif not l.lower().startswith("instructions:"):
            steps.append(l)
    return name, steps
```

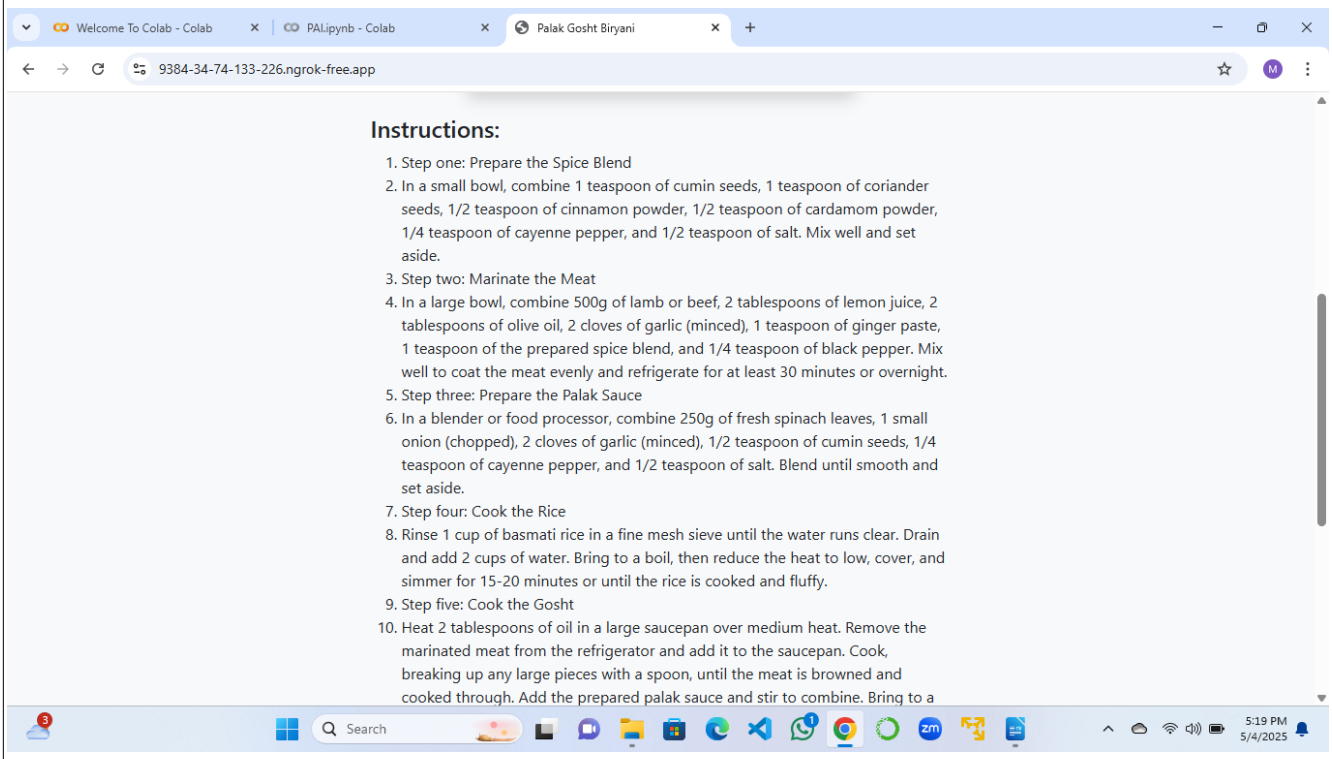
```
def generate_image(prompt):
    image = pipe(prompt).images[0]
    path = "static/recipe_image.png"
    os.makedirs("static", exist_ok=True)
    image.save(path)
    return path
@app.route("/", methods=["GET", "POST"])
def home():
    if request.method == "POST":
        if "save" in request.form:
            recipe_name = request.form["recipe_name"]
            ingredients = request.form["ingredients"]
            steps = request.form.getlist("steps")
            with open('recipes.csv', 'a', newline='') as f:
                writer = csv.writer(f)
                writer.writerow([recipe_name, ingredients, ' | '.join(steps)])
            return f"<h2 class='text-success'>✅ Recipe Saved Successfully!</h2><a href='/'>Back</a>"

        try:
            ingredients = request.form["ingredients"].split(',')
            ingredients = [i.strip() for i in ingredients]
            recipe_name, steps = generate_recipe(ingredients)
            image_path = generate_image(recipe_name + ' on a plate')
            ingredients_str = ', '.join(ingredients)
        except Exception as e:
            return f"<h3>Error generating recipe: {e}</h3><a href='/'>Try again</a>"

        return render_template_string("""
        <!DOCTYPE html>
        <html>
        <head>
            <title>{{ recipe_name }}</title>
            <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
        </head>
        <body class="bg-light text-center p-5">
            <div class="container">
                <h1 class="text-success">{{ recipe_name }}</h1>
                <img src="/{{ image_path }}" class="img-fluid rounded shadow" width="400"><br><br>
                <div class="text-start mx-auto" style="max-width: 600px;">
                    <h4>Instructions:</h4>
                    <ol>
                        {% for step in steps %}
                        <li>{{ step }}</li>
```

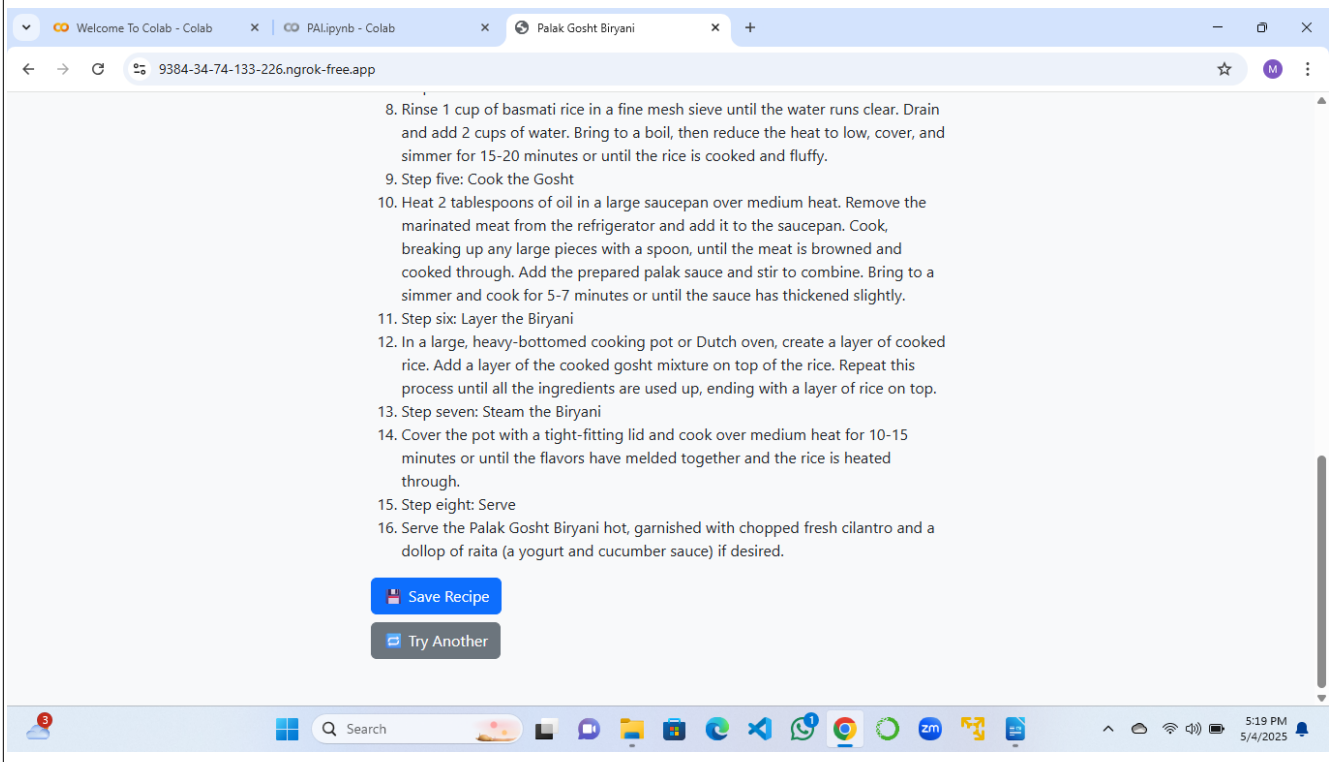
```
                {% endfor %}
            </ol>
            <form method="post">
                <input type="hidden" name="recipe_name" value="{{ recipe_name }}">
                <input type="hidden" name="ingredients" value="{{ ingredients }}">
                {% for step in steps %}
                <input type="hidden" name="steps" value="{{ step }}">
                {% endfor %}
                <button name="save" value="1" class="btn btn-primary">💾 Save Recipe</button>
            </form>
            <a href="/" class="btn btn-secondary mt-2">🔁 Try Another</a>
        </div>
    </div>
</body>
</html>
""", recipe_name=recipe_name, steps=steps, image_path=image_path,
ingredients=ingredients_str)
    return '''
    <!DOCTYPE html>
    <html>
    <head>
        <title>AI Chef</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
    </head>
    <body class="bg-light text-center p-5">
        <div class="container">
            <h1 class="mb-4">🍳 AI Recipe Generator</h1>
            <form method="post" class="mx-auto" style="max-width: 500px;">
                <div class="mb-3">
                    <input name="ingredients" class="form-control" placeholder="Enter ingredients (comma
separated)" required>
                </div>
                <button type="submit" class="btn btn-success">Generate Recipe</button>
            </form>
        </div>
    </body>
    </html>
    '''
public_url = ngrok.connect(5000)
print("Your app is live at:", public_url)
app.run(port=5000)
```

Output

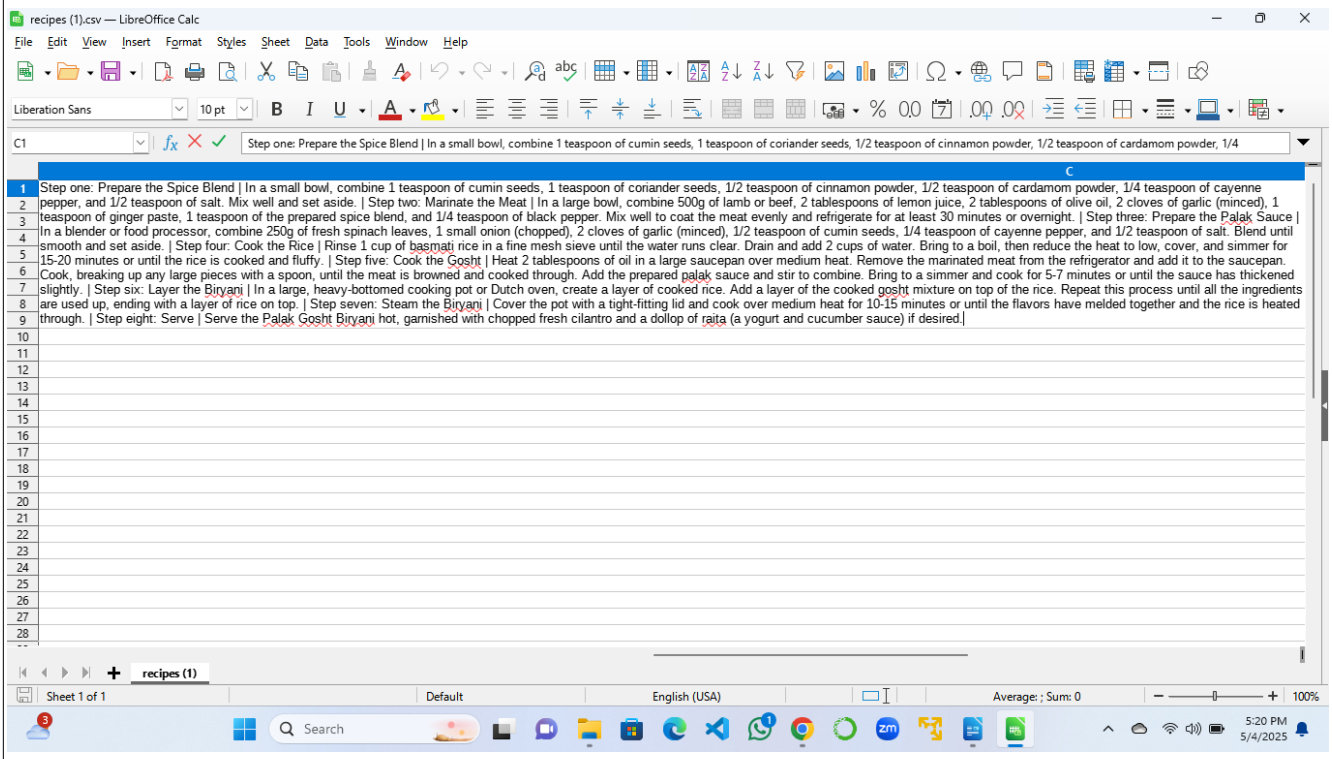

Welcome To Colab - Colab
PAI.ipynb - Colab
AI Chef
9384-34-74-133-226.ngrok-free.app
AI Recipe Generator
Enter ingredients (comma separated)
Generate Recipe
Search
5:17 PM
5/4/2025


Welcome To Colab - Colab
PAI.ipynb - Colab
Palak Gosht Biryani
9384-34-74-133-226.ngrok-free.app
Palak Gosht Biryani
Instructions:
1. Step one: Prepare the Spice Blend
2. In a small bowl, combine 1 teaspoon of cumin seeds, 1 teaspoon of coriander seeds, 1/2 teaspoon of cinnamon powder, 1/2 teaspoon of cardamom powder,
Search
5:19 PM
5/4/2025


Welcome To Colab - Colab
PAI.ipynb - Colab
Palak Gosht Biryani
9384-34-74-133-226.ngrok-free.app
Instructions:
1. Step one: Prepare the Spice Blend
2. In a small bowl, combine 1 teaspoon of cumin seeds, 1 teaspoon of coriander seeds, 1/2 teaspoon of cinnamon powder, 1/2 teaspoon of cardamom powder, 1/4 teaspoon of cayenne pepper, and 1/2 teaspoon of salt. Mix well and set aside.
3. Step two: Marinate the Meat
4. In a large bowl, combine 500g of lamb or beef, 2 tablespoons of lemon juice, 2 tablespoons of olive oil, 2 cloves of garlic (minced), 1 teaspoon of ginger paste, 1 teaspoon of the prepared spice blend, and 1/4 teaspoon of black pepper. Mix well to coat the meat evenly and refrigerate for at least 30 minutes or overnight.
5. Step three: Prepare the Palak Sauce
6. In a blender or food processor, combine 250g of fresh spinach leaves, 1 small onion (chopped), 2 cloves of garlic (minced), 1/2 teaspoon of cumin seeds, 1/4 teaspoon of cayenne pepper, and 1/2 teaspoon of salt. Blend until smooth and set aside.
7. Step four: Cook the Rice
8. Rinse 1 cup of basmati rice in a fine mesh sieve until the water runs clear. Drain and add 2 cups of water. Bring to a boil, then reduce the heat to low, cover, and simmer for 15-20 minutes or until the rice is cooked and fluffy.
9. Step five: Cook the Gosht
10. Heat 2 tablespoons of oil in a large saucepan over medium heat. Remove the marinated meat from the refrigerator and add it to the saucepan. Cook, breaking up any large pieces with a spoon, until the meat is browned and cooked through. Add the prepared palak sauce and stir to combine. Bring to a
Search
5:19 PM
5/4/2025


Welcome To Colab - Colab
PAI.ipynb - Colab
Palak Gosht Biryani
9384-34-74-133-226.ngrok-free.app
8. Rinse 1 cup of basmati rice in a fine mesh sieve until the water runs clear. Drain and add 2 cups of water. Bring to a boil, then reduce the heat to low, cover, and simmer for 15-20 minutes or until the rice is cooked and fluffy.
9. Step five: Cook the Gosht
10. Heat 2 tablespoons of oil in a large saucepan over medium heat. Remove the marinated meat from the refrigerator and add it to the saucepan. Cook, breaking up any large pieces with a spoon, until the meat is browned and cooked through. Add the prepared palak sauce and stir to combine. Bring to a simmer and cook for 5-7 minutes or until the sauce has thickened slightly.
11. Step six: Layer the Biryani
12. In a large, heavy-bottomed cooking pot or Dutch oven, create a layer of cooked rice. Add a layer of the cooked gosht mixture on top of the rice. Repeat this process until all the ingredients are used up, ending with a layer of rice on top.
13. Step seven: Steam the Biryani
14. Cover the pot with a tight-fitting lid and cook over medium heat for 10-15 minutes or until the flavors have melded together and the rice is heated through.
15. Step eight: Serve
16. Serve the Palak Gosht Biryani hot, garnished with chopped fresh cilantro and a dollop of raita (a yogurt and cucumber sauce) if desired.
Save Recipe
Try Another
Search
5:19 PM
5/4/2025

recipes.csv



Conclusion

This project demonstrates the powerful combination of LLMs (like LLaMA-3) and image diffusion models in generating creative and useful content. The application can serve as a fun cooking assistant, helping users explore new dishes with ease.
