# Project 62: Meal Planner

**Description:**

This intelligent meal planner creates daily or weekly menus based on your dietary preferences, restrictions, and goals. Whether you want vegetarian, high-protein, keto, or budget meals—it builds a tasty, practical plan with variety and balance.

---

### meal_planner.py

```
import openai
import os
import gradio as gr

# Load OpenAI API Key
openai.api_key = os.getenv("OPENAI_API_KEY")

# Function to generate a meal plan
def generate_meal_plan(preferences, days, meals_per_day):
    # Prompt the AI to act like a smart nutritionist
    messages = [
        {
            "role": "system",
            "content": (
                f"You are a meal planning assistant. Create a meal plan based
                f"Include breakfast, lunch, dinner, and snacks (if meals_per_d
                f"List the days with meals in bullet points or table style. In
            )
        },
        {
            "role": "user",
            "content": (
                f"Dietary Preferences: {preferences}\n"
                f"Number of Days: {days}\n"
                f"Meals Per Day: {meals_per_day}"
            )
        }
    ]
    try:
        # Call OpenAI to generate the plan
        response = openai.ChatCompletion.create(
            model="gpt-3.5-turbo",  # GPT-4 optional for fancier meal details
            messages=messages
        )

        # Return the meal plan
        return response["choices"][0]["message"]["content"].strip()

    except Exception as e:
        return f"Error: {str(e)}"

# Gradio interface for the meal planner
iface = gr.Interface(
    fn=generate_meal_plan,
    inputs=[
        gr.Textbox(label="Dietary Preferences (e.g. vegetarian, high protein,
        gr.Slider(minimum=1, maximum=14, step=1, label="Plan for How Many Days
        gr.Slider(minimum=1, maximum=5, step=1, label="Meals Per Day")
    ],
    outputs="text",
    title="🥗 AI Meal Planner",
    description="Tell me your preferences and I’ll cook up a meal plan for the
)

# Launch the meal planner
iface.launch()
```