Agentic AI Mini Project: Smart Follow-Up Assistant

Objective

Build a chatbot + AI agent combo that assists users in managing follow-ups automatically. The goal is to demonstrate reasoning, planning, and autonomous decision-making using freely available AI and automation APIs.

Problem Statement

You are tasked with creating a Smart Follow-Up Assistant that helps a user manage communication and reminders.

The chatbot should understand user messages (emails, chat logs, or notes) and identify which ones require a follow-up. The AI agent should then determine what action to take, when to follow up, and record or automate that action.

Example:

Input:
"Hi, can you send me the project report by Monday?"
"Let's revisit this topic after client feedback."
"Reminder: update the dashboard weekly."

Output:
Follow-up needed: Project report – due Monday
Follow-up: Client feedback pending – check in next week
Ongoing task: Update dashboard weekly

Requirements

Use any one of the following frameworks: LangChain, LangGraph, or N8n.
The solution must include the following components:

1. A Chatbot interface (can be CLI, notebook-based, or web UI).
2. An Agent workflow that:
   - Reads sample messages (from JSON, text file, or list).
   - Decides if a follow-up is needed.
   - Suggests or performs next actions.

Suggested Free APIs

You may use any of the following APIs/services to enhance your project (all offer free tiers or community editions):

- OpenAI API (GPT-4o-mini or GPT-3.5-turbo) via LangChain or direct HTTP call.
- Hugging Face Inference API (for text classification or summarization).

- N8n.io (for workflow orchestration – free self-hosted or cloud trial).
- Google Sheets API (for storing follow-up data).
- SendGrid or Mailgun sandbox API (optional – to simulate email follow-ups).
- Python libraries: requests, langchain, pandas, schedule, flask (for lightweight chatbot UI).

Deliverables

1 - Problem statement and approach
  - Tools and APIs used
  - System architecture diagram
  - Agent reasoning and workflow explanation

2. Working Demo:
  - Jupyter Notebook, N8n workflow, or simple web app
  - Include screenshots or short video (optional)

3. Reflection:
  - What worked well
  - What you would improve
  - How you could scale this to a real-world use case