AIM

To develop a **prompt-based application using ChatGPT** that organizes daily tasks, schedules reminders, suggests wellness tips, and answers general queries, demonstrating the **progression from simple to advanced prompt designs** and their corresponding outputs.

AI TOOLS REQUIRED

- **ChatGPT (Large Language Model)**
- Computer system with internet access

EXPLANATION

Large Language Models (LLMs) like ChatGPT can be used to build **prompt-based applications** that interact with users through natural language.
This experiment focuses on designing a **personal productivity assistant** that helps users manage daily activities, improves creativity, and supports practical problem-solving.

The assistant evolves from:

- **Simple prompts** (basic task input)
  to
- **Advanced prompts** (context-aware, adaptive, and personalized responses).

GIVEN PROMPT

Prompt:
"Design a personal productivity assistant that can help manage daily tasks, schedule reminders, suggest wellness tips, and answer general queries. The assistant should interact using natural language and be adaptable to the user's changing preferences over time."

PROCEDURE

Step 1: Define Core Requirements

The personal productivity assistant should be able to:

- Manage daily tasks
- Schedule reminders and events

- Provide wellness tips
- Answer general user queries
- Adapt to user preferences over time

Step 2: Construct Prompts for Each Task

1. Daily Task Management Prompt

Simple Prompt:

"Add a task for today."

Refined Prompt:

"Add the task 'Submit assignment' with high priority and remind me at 8 PM today."

2. Smart Scheduling Prompt

Simple Prompt:

"Schedule a meeting."

Refined Prompt:

"Schedule a meeting with my project team tomorrow from 10 AM to 11 AM and alert me 15 minutes before it starts."

3. Wellness Tips Prompt

Simple Prompt:

"Give me a health tip."

Refined Prompt:

"Suggest a short wellness tip for a student who spends long hours studying on a computer."

4. General Query Prompt

Simple Prompt:

"What should I do today?"

Refined Prompt:

"Based on my tasks and schedule today, suggest a balanced plan including study time, breaks, and exercise."

Step 3: Simulate Natural User Interaction

The assistant interacts via **natural language**, similar to a chat interface or command-line input:

User:

"Remind me to call mom at 6 PM."

Assistant:

"Got it! I'll remind you to call your mom today at 6 PM."

Step 4: Collect Feedback and Adapt Responses

User Feedback Example:

"I prefer short wellness tips in the evening."

Adapted Response:

"Noted! I'll share brief wellness tips with you in the evenings."

(Simulated preference adaptation using prompt memory)

EXPECTED OUTPUT

(Example Response by LLM)

Personal Productivity Assistant Features

1. Daily Task Manager

- Accepts tasks via natural language
  - Example: *"Remind me to call mom at 6 PM"*
- Organizes tasks by:
  - Priority
  - Deadline

- Provides:
  - Daily summaries
  - Pending task lists

2. Smart Scheduler

- Schedules events using contextual understanding
- Sets reminders automatically
- Alerts users about:
  - Overlapping appointments
  - Free time slots

3. Wellness Tips Generator

- Suggests daily wellness advice such as:
  - Hydration reminders
  - Stretching or exercise tips
  - Screen-time breaks
- Adapts suggestions based on:
  - User preferences
  - Previous responses

RESULT

The lab exercise resulted in the successful creation of a **prototype concept for a personal productivity assistant** powered by ChatGPT.

Students were able to:

- Understand how to tailor **LLM prompts** for real-life applications
- Foster **creativity** by designing personalized features
- Learn **prompt engineering techniques** for optimal AI interaction
- Experience the **versatility and usefulness** of generative AI in daily problem-solving

CONCLUSION

This experiment demonstrates how **prompt-based applications** can effectively leverage large language models like ChatGPT to manage daily tasks, improve productivity, and enhance user well-being. By progressing from simple prompts to advanced, context-aware designs, the assistant becomes more intelligent and personalized. Prompt engineering plays a crucial role in maximizing the usefulness of AI tools for practical, everyday applications.