**Register no: 212222040170**

# Ex no.7-Develop a prompt-based application tailored to their personal needs, fostering creativity and practical problem-solving skills while leveraging the capabilities of large language models.

## Aim:

To develop a prompt-based application using ChatGPT - To demonstrate how to create a prompt-based application to organize daily tasks, showing the progression from simple to more advanced prompt designs and their corresponding outputs.

## AI Tools Required:

CHATGPT

## Procedure:

1: Define Functional Requirements

2: Choose Your Technology Stack

3: Design the System Architecture

4: Set Up the Chat Interface

5: Implement Task Management

6: Build Reminder and Scheduler Module

7: Add Wellness Tips Feature

8: Enable General Query Answering

9: Build Preference Learning Logic

10: Integrate Calendar & Notifications

11: Test & Refine

OUTPUT:

Here's a structured plan to **develop a prompt-based application using ChatGPT** for organizing daily tasks. The goal is to demonstrate **how prompt engineering evolves**, from simple to advanced, and how each level enhances the application's usefulness.

---

**Objective**

Build a ChatGPT-driven app that helps users manage daily tasks using prompts — progressing from basic command interpretation to context-aware task planning.

---

**1. Define the Core Use Case**

**Prompt-Based Task Organizer**: A user types a natural language prompt, and ChatGPT organizes it into structured to-do lists, prioritizes tasks, and offers scheduling suggestions.

---

**2. Simple Prompt Design**

**Prompt Example:**

> "List my tasks for today: email client, team meeting at 3 PM, grocery shopping."

**✅Output:**

text
CopyEdit
Today's Tasks:
1. Email client
2. Team meeting at 3 PM
3. Grocery shopping

**Explanation:**

- ✓ No time analysis or prioritization.
- ✓ Raw listing of user-entered items.

---

### 3. Intermediate Prompt Design

**Prompt Example:**

> "I have to send an email to the client, attend a team meeting at 3 PM, and go grocery shopping. Help me prioritize."

**✅Output:**

text
CopyEdit
Prioritized Tasks:
1. Send email to client (Urgent - Do first)
2. Team meeting at 3 PM (Scheduled)
3. Grocery shopping (Low priority - Evening)

Suggested Plan:
- 9:00 AM: Email client
- 3:00 PM: Attend team meeting
- 6:00 PM: Grocery shopping

**Explanation:**

- ✓ Adds **prioritization and scheduling**.
- ✓ Assumes urgency and time-sensitivity.

---

### 4. Advanced Prompt Design with Context

**Prompt Example:**

> "Based on my calendar, I have a doctor appointment at 11 AM, a report due by 5 PM, and I need to pick up my kids at 4 PM. I also want to find 30 minutes for exercise. Can you create a schedule?"

**✅Output:**

text
CopyEdit
Personalized Daily Schedule:
- 9:00 AM – 10:30 AM: Work on report
- 10:30 AM – 11:00 AM: Travel to doctor
- 11:00 AM – 12:00 PM: Doctor appointment

- 12:00 PM – 12:30 PM: Exercise
- 12:30 PM – 3:30 PM: Continue report
- 4:00 PM – 4:30 PM: Pick up kids
- 4:30 PM – 5:00 PM: Finalize and submit report

**Explanation:**

- ✓ Incorporates **calendar awareness**, **time constraints**, and **personal goals** (like exercise).
- ✓ Acts like a **virtual assistant**.

---

## 5. Advanced Prompt Engineering Techniques

| Technique | Description | Example |
|---|---|---|
| Few-shot prompting | Include examples to guide output | "Here's how I like my tasks listed..." |
| Chain-of-thought | Ask the model to reason step by step | "Help me think through how to prioritize these tasks." |
| Role-based prompting | Instruct the model to act like a productivity coach | "You are my task manager. Help me plan the day efficiently." |

---

## 6. Implementation Options

**Frontend:**

- ✓ Simple Web UI using HTML/CSS/JS or React

**Backend:**

- ✓ Python (Flask/FastAPI) to handle requests
- ✓ OpenAI API to access ChatGPT

**Example API Call:**

python
CopyEdit

```
import openai

response = openai.ChatCompletion.create(
  model="gpt-4",
  messages=[{"role": "user", "content": "Organize my tasks for today: ..."}]
)
```

---

**7. Possible Add-ons**

- ✓ Integration with Google Calendar / Outlook
- ✓ Task reminders via email or notification
- ✓ Voice input support
- ✓ Mood-aware task adjustment ("I feel tired today...")

---

**Demo Flow (Presentation-Ready)**

| Step | Feature | Prompt Example | Output |
|---|---|---|---|
| 1 | Basic Listing | "List tasks: laundry, email boss" | Raw list |
| 2 | Prioritization | "Help me prioritize" | Ordered list |
| 3 | Time awareness | "Schedule my day" | Time-based plan |
| 4 | Context + Preferences | "I feel tired, adjust schedule" | Relaxed plan |

**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."

# AI Personal Productivity Assistant

**Key Objectives:**

- ✓ Manage daily tasks with ease
- ✓ Schedule and remind users of important activities

- ✓ Offer personalized wellness tips
- ✓ Answer general knowledge or lifestyle queries
- ✓ Adapt to changing user preferences over time

---

**Core Features**

**1. Task Management**

**Input:** "Add ‘submit report’ to my task list for tomorrow."

**Output:** “Got it. 'Submit report' added to your tasks for tomorrow.”

**Extras:** Supports tagging (work, personal), deadlines, and priorities.

**2. Smart Reminders & Scheduling**

Sync with Google/Apple Calendar

Natural language input: “Remind me to take medicine at 9 PM daily.”

Notification options: Email, push, or voice assistant integration

**3. Wellness Tips Engine**

Provides daily or situational wellness tips (e.g., hydration, meditation)

Responds to mood/context prompts like:

"I'm feeling overwhelmed."
"What can I eat for energy?"

**4. General Knowledge & Lifestyle Q&A**

Answer user queries about topics like:

- ✓ Productivity techniques (Pomodoro, GTD)
- ✓ Quick recipes
- ✓ Travel recommendations
- ✓ Workouts, etc.

**5. Learning & Adaptation**

Learns from user behavior:

- ✓ Frequently delayed tasks → Suggest time blocks or lighter loads
- ✓ User wakes up later on weekends → Adjust morning routines
- ✓ Tracks completed tasks to adapt priority suggestions

---

**Interaction Flow (Natural Language)**

| User Says | Assistant Responds |
|---|---|
| "What's on my plate today?" | "Here's your day: 9 AM team meeting, 1 PM gym, 4 PM project review." |
| "Push my afternoon call to tomorrow." | "Moved your 2 PM call to the same time tomorrow." |
| "Any tips for staying focused?" | "Try the Pomodoro technique—work 25 minutes, break 5 minutes. Want to start a timer?" |
| "Remind me to drink water every 2 hours." | "Done. You'll get hydration reminders every 2 hours from 8 AM to 8 PM." |

---

**System Architecture**

**Frontend:**

- ✓ Web or mobile app with chat-based UI
- ✓ Voice assistant support (optional)

**Backend:**

- ✓ **ChatGPT API**: Conversational intelligence
- ✓ **Task DB**: Store tasks, reminders, preferences
- ✓ **Scheduler**: Triggers time-based events
- ✓ **Wellness Engine**: Curated tip database + context-based recommendation system

**Tech Stack:**

| Layer | Tools |
| --- | --- |
| Frontend | React / Flutter / Swift |
| Backend | Node.js / Python (Flask or FastAPI) |
| DB | Firebase / PostgreSQL |
| AI | OpenAI API (ChatGPT), optional ML model for preference adaptation |

---

**Adaptive Intelligence**

| Input | Adaptation Example |
| --- | --- |
| Skips workouts on Mondays | Suggests lighter tasks or motivation quotes |
| Responds better to evening planning | Offers task suggestions in the evening |
| Frequently reschedules a task | Flags it and suggests alternatives or breaks it into sub-tasks |

---

**User Preferences and Privacy**

Data encrypted and stored securely

User can export or delete data anytime

Option to opt in/out of adaptation features

---

**Future Add-ons**

Gamified streaks for wellness and task completion

Mood tracking through sentiment analysis

Voice-based journaling or reflection prompts

Cross-platform sync (Slack, Gmail, Notion, etc.)

---

**□ Demo Use Case (Storyboard)**

1. **Morning**:

   "Good morning! You have 3 meetings today. Would you like to review or reschedule anything?"

2. **Afternoon**:

   "Time to stretch! A quick walk will help boost your focus."

3. **Evening**:

   "You completed 5/6 tasks today. Great work! Shall I carry over the last one or reschedule?"

## Conclusion

Designing a **Personal Productivity Assistant** powered by ChatGPT presents an innovative and practical approach to improving daily efficiency and well-being. By combining **natural language understanding**, **task management**, **smart reminders**, **wellness guidance**, and **adaptive learning**, such an assistant offers a holistic solution to modern productivity challenges.

Through a conversational interface, users can manage their schedules effortlessly, receive personalized tips for maintaining mental and physical health, and get instant answers to everyday queries. As the assistant evolves by learning user preferences and behaviors over time, it becomes more intelligent and aligned with individual lifestyles.

Ultimately, this project not only showcases the power of prompt engineering and AI integration but also paves the way for building empathetic, context-aware digital assistants that support users in achieving both their professional goals and personal wellness.

## Result:

The Prompt is executed successfully