Exno.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.

Name: SINDHUJA P

Register no: 212222220047

Aim:

To 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. Here's a complete guide for developing a **prompt-based application using ChatGPT** to organize daily tasks. The project demonstrates how to build a task management assistant by progressively refining prompts — from simple text inputs to more advanced, structured interactions.

Algorithm:

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.

Objective:

To demonstrate how prompt engineering can be used to create a task management assistant, showcasing how prompt design evolves from simple instructions to structured, multi-step conversations.

Tools Required:

- ChatGPT API (or ChatGPT web interface for demo)
- Optional: Web-based front-end (HTML + JS) or command-line interface
- Optional: Backend (Node.js / Python Flask) to process input-output if building a full app

Procedure

1. Basic Prompt: Simple Task Logging

Prompt:

"Add the following task to my to-do list: 'Buy groceries at 5 PM'."

Output:

Task added: “Buy groceries at 5 PM”

Notes:

- Direct command.
- No structure; ChatGPT simply echoes the action.

2. Enhanced Prompt: Structured Input

Prompt:

“Add a task with the title 'Meeting with HR', set for tomorrow at 10 AM. Mark it as high priority.”

Output:

```
{
  "title": "Meeting with HR",
  "time": "Tomorrow, 10 AM",
  "priority": "High"
}
```

Notes:

- Introduces structured task attributes.
- Output format can now be parsed programmatically.

3. Prompt for Batch Task Entry

Prompt:

“Create my task list for today:

1. Morning workout at 6 AM
2. Team call at 9 AM
3. Complete project report by 2 PM
4. Dinner with Sam at 8 PM”

Output:

```
[
  {"title": "Morning workout", "time": "6 AM"},
  {"title": "Team call", "time": "9 AM"},
  {"title": "Complete project report", "time": "2 PM"},
  {"title": "Dinner with Sam", "time": "8 PM"}
]
```

Notes:

- Batch processing using natural language.
- Could be extended to extract deadlines and categories.

4. Advanced Prompt: Contextual Planning

Prompt:

“Plan my day based on these inputs:

- I have a meeting from 9 to 10 AM
- I need 2 hours for deep work
- I prefer working out in the evening
- Add buffer time before and after meetings”

Output:

```
[
  {"title": "Prepare for meeting", "time": "8:30 AM - 9:00 AM"},
  {"title": "Team meeting", "time": "9:00 AM - 10:00 AM"},
  {"title": "Break", "time": "10:00 AM - 10:15 AM"},
  {"title": "Deep work", "time": "10:15 AM - 12:15 PM"},
  {"title": "Evening workout", "time": "6:00 PM - 7:00 PM"}
]
```

Notes:

- Involves reasoning and time allocation.
- More intelligent, context-aware planning.

5. Interactive Prompt Design (Chat Flow)

Chat Flow:

- User: "Start my daily planning"
- ChatGPT: "Sure! What time do you wake up?"
- User: "7 AM"
- ChatGPT: "How many meetings do you have today?"

- User: "Two: one at 10 AM and one at 4 PM"
- ChatGPT: *(responds with an optimized schedule)*

Output:

Customized plan generated with user input.

Notes:

- Uses dynamic prompts based on user answers.
- Can be implemented as a chatbot with persistent memory (via session or API).

Progression Recap

Stage	Feature	Complexity
1	Simple task entry	Basic
2	Structured task format	Intermediate
3	Multiple tasks with parsing	Intermediate
4	Context-aware scheduling	Advanced
5	Interactive prompt chain	Advanced

Extensions

- Integrate with **Google Calendar or Notion API**
- Use **voice input** with Whisper API
- Enable **daily reminders** via Twilio or email
- Add **priority ranking** and **task categorization**

Outcome

By the end of this, you'll:

- Understand how prompt design affects ChatGPT's output
- Be able to guide ChatGPT through progressively more complex task-handling scenarios
- Build a basic but powerful prompt-driven productivity assistant

Result:

The Prompt is executed successfully