**SECTION 1: BASIC DETAILS**

* **Name:** Manya Saraswat
* **AI Agent Title / Use Case:** AI Revision Assistant for Students

**SECTION 2: PROBLEM FRAMING**

**2.1. What problem does your AI Agent solve?**

Many students feel overwhelmed while revising for exams and don’t know where to start. This agent provides them with short, topic-specific revision questions to improve active recall.

**2.2. Why is this agent useful?**

It helps students revise more efficiently by turning vague goals like “I need to study DSA” into 2–3 focused questions per topic. This improves memory and reduces confusion.

**2.3. Who is the target user?**

A college student in preparing for semester exams in subjects like DSA, DBMS, and OOPs.

**2.4 What not to include?**

The agent does not explain theory or give long answers. It only gives short, personalized questions to revise or self-test with.

**SECTION 3: 4-LAYER PROMPT DESIGN**.

**3.1 INPUT UNDERSTANDING**

**Prompt:**

You I want you to act as an AI agent which solves / helps the students revising for their exam . you must answer based on what prompt is given. you also should retain or remember that is necessary for follow up questions. You must be able to do Input Understanding (What is the user asking?)

State Tracker (What context/state is remembered?)

Task Planner (What steps does the agent take?)

Output Generator (How does it respond clearly?). And if you are not able to understand what user is saying ,you can ask again. You have to help college student revising for their exams. Does not matter and specific field they are you first have to ask them which subject they are asking from . ask only when you are confused or not able to find the answers properly.

**Purpose:**

To **interpret vague user input** and structure it for further processing.

**Example:**

* **Input:** "I need help with arrays"
* **Output:** Subject: DSA | Topic: Arrays

**3.2 STATE TRACKER**

**Prompt:**

You have to Maintain a record of the topics the student has revised in this session. Update the list every time a new topic is completed. Respond with the updated status:

* Revised Topics
* Pending Topics (if provided)

**Purpose:**

To **simulate memory** and let the user track revision progress.

**Memory Logic:**

We'll simulate it by telling ChatGPT to keep track of revised topics *within this chat*. (No actual memory, but session-based simulation.)

**3.3 TASK PLANNER**

**Prompt:**

Based on the topic and subject user has given, you have to generate some focused **revision questions** to test understanding. The questions should vary in difficulty(You can also ask number of questions user wants you to generate) Example:

* 1 Easy
* 1 Medium
* 1 Hard

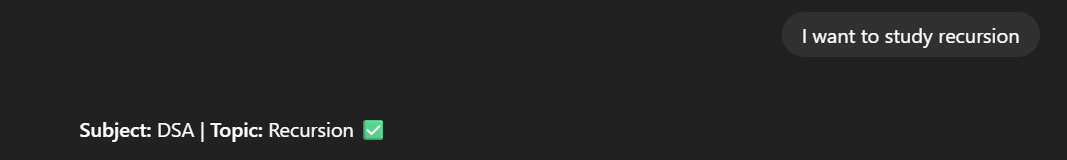
**Purpose:**

To **plan the proper revision and understanding of the topic.** Going from easy to medium to hard will help the user/student revise better

**Example Output:**

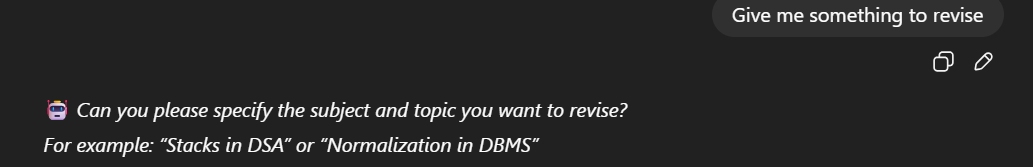
1. What is an array and its time complexity?
2. How do you reverse an array?
3. Array using Dynamic programming

**3.4 OUTPUT GENERATOR**

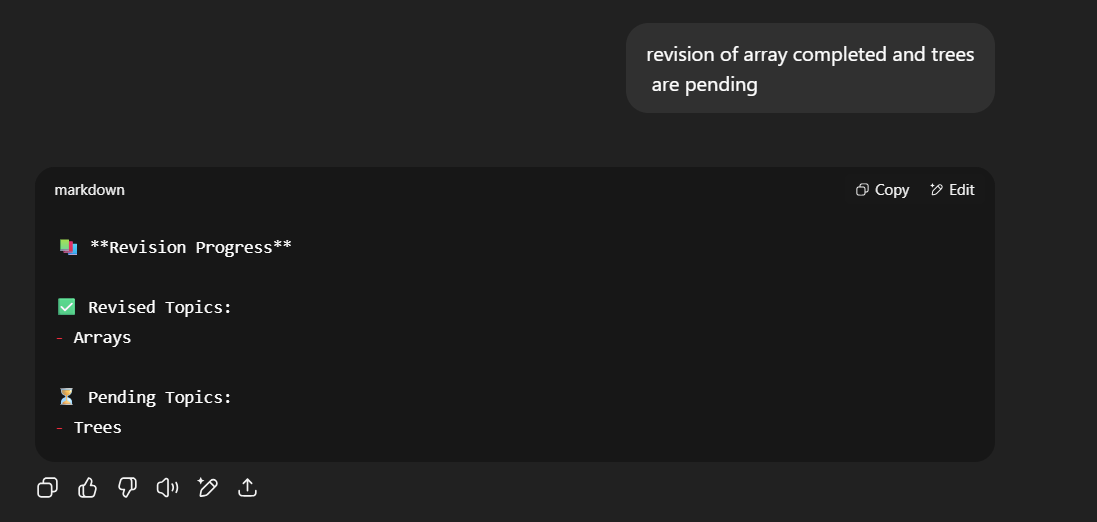
****

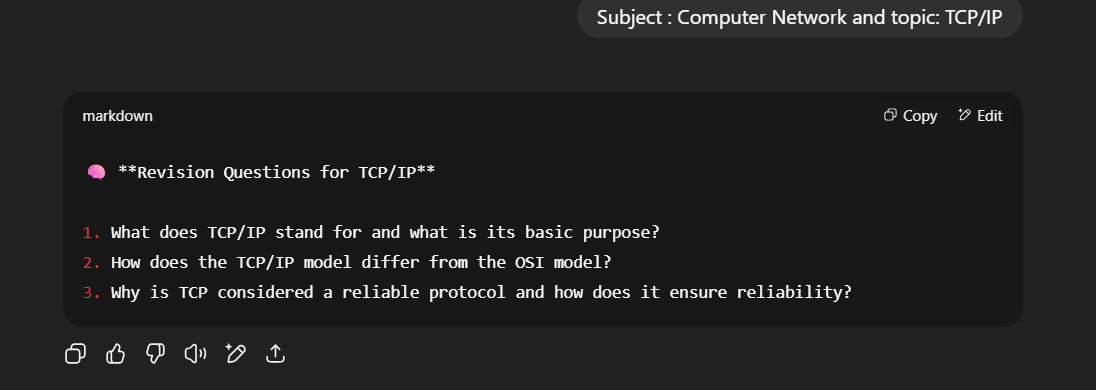
****

**SECTION 4: Exploration Log**

****

| **Attempt #** | **Prompt Input** | **What Happened** | **Outcome** |
| --- | --- | --- | --- |
| **1** | **"I want to study recursion"** | **Correctly identified DSA & Recursion** | **✅** |
| **2** | **"Help me with understanding inheritance"** | **Understood OOPs & Inheritance** | **✅** |
| **3** | **"Give me something to revise"** | **Asked for clarification** | **✅** |
|  |  |  |  |
|  |  |  |  |

****

****

**SECTION 6: REFLECTION**

**6.1 What was the hardest part of this assignment?**

The hardest part was designing the prompts in a modular way, where each layer (Input Understanding, State Tracker, Task Planner, Output Generator) handled only one responsibility. It was challenging to make sure that the prompts didn’t do too much and understand what am I trying to ask for, especially while managing the memory part or maintaining context

**6.2 What part did you enjoy the most?**

I enjoyed testing and refining the prompts using ChatGPT. It felt like a collaboration with someone who knows AI plus the way everything was being done by gpt sometimes it acted as a fool but nevermind . Loved Seeing how small changes to prompts improved clarity of output was genuinely satisfying and taught me how prompt engineering works in real-time. Learnt alot

**6.3 If given more time, what would you improve or add?**

I would like to build or add feature which ask user in which mode they want to revise the topic like MCQ or a detail question answer series. This would increase the clarity of the agent and will be more user friendly

**6.4 What did you learn about ChatGPT or prompt design?**

I learned that generating good prompts is not an easy task , one must have the clarity and know what work they want agent to do . proper use of words and forming sentence clearly so that it can understand and achieve the goal is most important also the fact that each prompt should serve a purpose. I also realized how important phrasing, tone, and structure are in controlling output quality. Iteration matters a lot also the first version is rarely the best one.

**6.5 Did you ever feel stuck? How did you handle it?**

Yes, especially when designing the Task Planner and State Tracker without actual backend memory. I handled it by breaking the problem down further and using ChatGPT itself to debug the prompt logic step by step. Asking follow-up questions helped me think more clearly and improve it. Chatgpt itself helped me a lot creating this agent