1. BASIC DETAILS

***Name*:** G.Suvishal Xavier Joy
***AI Agent Title / Use Case*:** "AI Agent to help students revise for an exam."

2. *PROBLEM FRAMING*

1.1. What problem does your AI Agent solve?
A lot of students have challenges studying for tests because they have loads of material, lack focus, or have no idea what to study next. This AI Agent helps by breaking down revision into smaller, more manageable tasks and providing students personalized, interactive study sessions.

1.2. Why is this agent useful?

It provides students unique, adaptive revision assistance, such as quizzes, summaries, worksheets, and explanations of concepts, depending upon their progress and the subject they're studying.

1.3. Who is the target user?
The target users are high school or college students preparing for subject-specific exams (e.g., CBSE 12th board exams, engineering entrance exams, or university semester exams).

1.4. What *not* to include?
This agent does not create comprehensive study guides, manage lengthy tutoring sessions, or perform irrelevant productivity duties. To keep lightweight and focused, it stays away from live video/audio interactions and only uses brief, text-based editing tools.

4-LAYER PROMPT DESIGN

3.1 INPUT UNDERSTANDING
Prompt:

I want you to act like the input analyzer of an AI study assistant. When a student types something like: 'I need help for my physics test, especially Newton's laws. I only have 1 hour,'
break this into 6 fields:

1. Subject
2. Urgency (high/medium/low)

3. Study type (review, practice, memorization)
4. Time available
5. Difficulty (high/medium/low)
6. Specific topics (max 3)
   Return this as a JSON object.

What is this prompt responsible for?

This helps the AI to understand the user's questions by dividing the data into organized sections that the agent may utilize to organize the study assistance.

Example Input + Output:

Input : Now I want you to simulate memory for my AI agent.
Every time a student sends a request, add their subject and urgency to a 'studyHistory' list.
If the difficulty is high, mark that subject in 'weakAreas'.
These will help personalize future responses."

```
{
 "subject": "biology",
 "urgency": "high",
 "studyType": "review",
 "timeAvailable": "2hours",
 "difficulty": "high",
 "specificTopics": ["photosynthesis"]
}
```

This became the base for all future layers in my AI agent.

STATE TRACKER
Prompt:

"Now, please provide my AI agent a memory simulation.
Add the student's subject and urgency to a "studyHistory" list each time they submit a request.
If the subject is really tough, mark it under "weakAreas."
Future answers will be more tailored thanks to this.

How does this help the agent "remember"?

It creates a short-term memory system that keeps track of what topics the student has already revised — and what they struggle with — during a session.

Did I simulate memory with variables/system messages?

Yes. I used variables like:

```
js
Copy code
userState = {
  studyHistory: [],
  weakAreas: []
}
```

Each session builds on this using the data from the input understanding step.

TASK PLANNER

Prompt:

"Now assist me in making an intelligent task planner.
In accordance with the retrieved input (time, studyType, subject, urgency, etc.):

Choose a technique, such as "intensive review," "spaced repetition," "practice-heavy," etc.

List three to five steps to take.

Divide the time into blocks of activities, such as 20 minutes for review and 30 minutes for practice.

Provide four to five helpful resources.
The output need to be neat and organized.

What steps does my agent take internally to solve the problem?

1. Pick the right study strategy
2. Plan detailed steps for the session
3. Break up the available time logically
4. Suggest helpful study materials (videos, notes, flashcards)

Did I use chaining? Branching? How did I manage complexity?

Branching according to studyType and urgency is what I did. For each section, I developed distinct functions, such as determineStrategy(), generateSteps(), allocateTime(), etc. The information remains readable and modular.

OUTPUT GENERATOR

Prompt:

Now assist me with the final output design.
Employ a tidy layout with:
Title, topic, approach, and length
An action-step list with numbers
A remark that inspires
A section with reference links, notes, and advice
Finish with choices to "Start New Session," "Export Plan," and so on.
Keep the tone lighthearted and student-centered.

What kind of output formatting or phrasing did I aim for?

1. Friendly, clear, motivational tone
2. Easy-to-scan structure (like a coaching app)
3. Colorful visual formatting (when implemented in HTML)
4. Encourages students to take action

Any special behavior?

- Shows “layer progress” visually (in my UI design)
- Reuses memory to avoid repeating topics
- Includes motivational notes and realistic study suggestions
- Supports features like plan export and session reset

Section 4:

CHATGPT EXPLORATION LOG

Attempt	Prompt Variant	What Happened	What I Changed
1	“Now I want you to simulate memory...”	A userState object with studyHistory and weakAreas was appropriately generated by Cha tGPT.	No modification is required. For memory simulation, Prompt performed admirably.
2	(Added new input with low difficulty)	As anticipated, ChatGPT added to weakAreas but did not update studyHistory.	verified that the difficulty-based filtering logic was sound.

SECTION 5:

OUTPUT TESTS

Test 1: Normal input

Input: "I want a journaling prompt about career confusion"

Output:

Math Study Plan – Intensive Review (2 hours)

Steps:

1. Review key algebra formulas – 20 min
2. Practice trigonometry identities – 30 min
3. Take a quiz on both topics – 30 min
4. Analyze mistakes – 20 min
5. Final revision flashcards – 20 min

Motivational Note:

"You've got this! Focus on one concept at a time. Small wins matter."

- **Test 2:** Vague input

  Input: "Give me something to think about"

  Output:

Study Plan – General Review (1 hour)

Since you didn't mention the subject or topic, here's a flexible review plan:

- Review notes from your latest class – 20 min
- Practice any recent assignments or quizzes – 25 min
- Reflect on areas you felt unsure about – 15 min

Tip:

Try being more specific next time! Example: "I have a science test on ecosystems."

- **Test 3:** Invalid input or challenge

  Input: ""

  Output:

  ❗ **Oops! I didn't catch that.**

  Could you tell me what subject or topic you're trying to revise?

  Example: *"Help me study for biology – I struggle with genetics."*

6.1. What was the hardest part of this assignment?

Creating prompts that were sufficiently explicit to enable ChatGPT to reliably extract structured data from ambiguous student input was the most challenging aspect. To ensure that the model produced expected results, such as subject, duration, and difficulty, I had to thoroughly test and modify the language. Additionally, it took a lot of trial and error to simulate memory and determine what should be stored in studyHistory or weakAreas.

6.2. What part did you enjoy the most?

When the AI agent transformed a disorganized student communication into a customized, understandable study schedule, I truly loved seeing it come to life. I felt as though I was working with the AI as I saw the results change as I improved each layer. The experience was also made innovative and enjoyable with the addition of inspirational messages and a visually appealing interface design.

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

I would include follow-up memory between sessions, such as performance or long-term subject tracking. Giving the learner a choice between several study methods (visual, text-based, or quiz-heavy) might also be beneficial. The agent would also be more comprehensive if it included real learning materials or included links to practice sets (such as those on YouTube or Khan Academy).

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

Prompt design, I discovered, is similar to plain-language programming: the more accurate you are, the better the AI works. Even little phrase variations might have wildly different outcomes. It's best to break things down into layers or phases and urge the AI gradually, I also learnt, rather than attempting to acquire the ideal response all at once.

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

Yes, I felt stuck when the agent gave me inconsistent outputs, such as unexpected topic names or JSON data that were missing. In order to deal with it, I made the prompt simpler, provided examples, and gave the AI additional structure (such as a key-value output format). I was also able to debug more efficiently by testing with edge situations and asking ChatGPT why it did anything.

SECTION 7

HACK VALUE

1. Did you simulate multiple users?

No, I didn't simulate multiple users in this version of the agent. The focus was on a single-user experience to keep the memory and logic clean within one session. However, the current structure could be extended to handle multiple users in future versions by assigning each a unique ID or session.

2. Did you add logic for memory or role play?

Yes, I added simulated memory using two key elements: `studyHistory` and `weakAreas`. This allowed the agent to personalize its responses based on what the user studied before and which subjects they struggled with. It didn't just react to the latest message — it adapted its responses over time within the session.

3. Did you explore chaining multiple agents?

No, I didn't explore chaining multiple agents. Instead, I focused on building a single agent using a layered architecture — Input Understanding, State Tracker, Task Planner, and Output Generator — which allowed the agent to act in stages and remain structured, almost like an internal pipeline.

This shows that while the agent stays within scope, it still demonstrates extended thinking with memory simulation and a clean modular design.