AI Agent Assignment

SECTION 1: BASIC DETAILS

Name: Nithesh Raj R
AI Agent Title / Use Case: AI Agent to Help College Students Revise for Exams

SECTION 2: PROBLEM FRAMING

1.1. What problem does your AI Agent solve?

College students often feel overwhelmed during exam preparation and don't know where to start, what to revise first, or how to manage limited time. This agent helps students structure their revision based on urgency, subject difficulty, and available time.

1.2. Why is this agent useful?

The agent reduces confusion and stress by breaking revision into small, actionable steps. It adapts responses based on the student's exam timeline and weak areas, making revision more focused and efficient.

1.3. Who is the target user?

A college student preparing for an upcoming exam (e.g., a fourth year engineering student revising core subjects before semester exams).

1.4. What *not* to include?

- **No full syllabus coverage**
- **No performance analytics or grading**
- **No external integrations (LMS, calendars)**
  These were avoided to keep the agent simple and focused.

🧱 SECTION 3: 4-LAYER PROMPT DESIGN

Create a subsection for each of the 4 components of the agent architecture:

◇ 3.1 INPUT UNDERSTANDING

Prompt:

You are an AI agent helping students revise for exams.

Analyze the user's input and identify:

1. Subject or topic

2. Exam date or urgency

3. Student level (beginner/intermediate)

4. Type of help requested (revision, explanation, questions)

If any information is missing, ask a clarifying question before proceeding.

What is this prompt responsible for?

It interprets what the student is actually asking and extracts key details needed for revision planning.

Example Input + Output:

Input: "I have an OS exam tomorrow"
Output: Subject = Operating Systems, Urgency = High, Missing info = weak topics

◇ 3.2 STATE TRACKER

Prompt:

Maintain a short memory of the conversation:

- Subject being revised

- Exam timeline

- Topics already covered

- Topics the student finds difficult

Update this state after every user response and use it for future replies.

How does this help the agent "remember"?

It allows the agent to avoid repeating content and tailor future responses based on what the student has already studied.

Did you simulate memory with variables / system messages? If yes, how?

Memory was simulated using a structured list in the system message that gets updated after each interaction.

◇ 3.3 TASK PLANNER

Prompt:

Based on the current state, plan the next steps:

1. Decide whether to revise or explain concepts

2. Prioritize topics based on urgency

3. Break topics into small chunks

4. Decide whether to include examples or practice questions

Output the plan as a numbered list.

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

The agent uses step-by-step reasoning (chaining) to decide what to do next instead of jumping directly to an answer.

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

Only linear chaining was used to keep the logic simple and readable.

3.4 OUTPUT GENERATOR
Prompt:

Generate a clear, student-friendly response:

- Use simple language

- Prefer bullet points

- Keep explanations under 150 words

- End with a follow-up question to guide the next step

- Maintain a calm and motivating tone

What kind of output formatting or phrasing did you aim for?
Any special behavior? (e.g., examples, markdown formatting, tone control, etc.)

Short explanations, structured formatting, and a supportive tone to reduce exam anxiety.

🔍 SECTION 4: CHATGPT EXPLORATION LOG

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	"Help me revise OS"	Output too generic	Added urgency & weak-topic detection	To make advice more relevant
2	Added task planner prompt	Output became structured	Added word limit	To avoid long explanations

3	Added follow-up question rule	Conversation flowed better	Improved output formatting	To guide next interaction

🧪 SECTION 5: OUTPUT TESTS (Optional but Recommended)

Include 2–3 sample outputs from your agent. Try edge cases.

Test 1: Normal input
Input: "I have an OS exam tomorrow and memory management is weak"
Output:

- Focus on paging and segmentation
- Revise key formulas and diagrams
- Practice 2 numerical problems
  Follow-up: "Do you want a quick summary or practice questions?"

- **Test 2:** Vague input
  Input: "Help me study"
  Output: "Sure! Which subject are you studying and when is your exam?"

- **Test 3:** Invalid input or challenge
  Input: ""
  Output: "I didn't receive any input. Please tell me which subject you want to revise."

🔁 SECTION 6: REFLECTION

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

  The hardest part of this assignment was separating the AI agent's behavior into distinct layers instead of writing a single long prompt. Initially, it felt unnatural to think in terms of input understanding, memory, planning, and output separately. However, once the layers were defined, the agent's behavior became more controlled and predictable. This helped me understand how structured thinking improves AI responses.

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

I enjoyed designing the task planner the most because it involved breaking down a problem into logical steps. It felt similar to teaching the agent how to reason instead of just answering questions. Creating a clear sequence of actions made the agent's responses more helpful and organized. This step made the agent feel more intelligent and purposeful.

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

  If given more time, I would add topic-wise progress tracking so the agent can monitor what has already been revised. I would also include adaptive difficulty, where explanations become simpler or more advanced based on student understanding. This would make the agent more personalized and effective over long study sessions. It would improve continuity across multiple interactions.

- **6.4. What did you learn about ChatGPT or prompt design?**
  I learned that small constraints in prompt design, such as word limits, tone control, and structured formatting, significantly improve response quality. Clear instructions help reduce generic or overly long answers. Prompt iteration is essential to guide the model toward consistent behavior. Even minor changes in phrasing can produce noticeably better results.

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

  Yes, I felt stuck when the agent began giving repetitive or generic responses. To resolve this, I improved the state tracker so the agent could remember previously covered topics. This reduced repetition and made the conversation more natural. Iterating on prompts helped me regain control over the agent's behavior.

🧠 SECTION 7: HACK VALUE (Optional)

Did you go beyond the brief in any way?

Yes, I went slightly beyond the brief by simulating memory using a state-tracking mechanism that retained subject, urgency, and covered topics across interactions. I also enforced step-by-step reasoning through a task-planning layer, which helped control the agent's internal logic. This made the agent more consistent and context-aware without adding unnecessary complexity.

.