

# AI Agent Assignment – Submission Document

---

## SECTION 1: BASIC DETAILS

**Name:** Anupam Dwivedi\ **AI Agent Title / Use Case:** AI Agent to help students revise Computer Science concepts efficiently before exams

---

## SECTION 2: PROBLEM FRAMING

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

Many students feel overwhelmed before exams, unsure of what to revise and how to prioritise topics. This agent simplifies revision by giving focused questions, summarising concepts, and testing understanding quickly.

### 2.2. Why is this agent useful?

It saves time, reduces confusion, and ensures structured last-moment preparation with active recall, leading to better retention and confidence.

### 2.3. Who is the target user?

B.Tech Computer Science students revising subjects like Operating Systems, Java, or Data Structures before university exams.

### 2.4. What not to include?

- No detailed coding tutorials (scope: revision only)
  - No unrelated career advice
  - No motivational quotes or generic study tips
- 

## SECTION 3: 4-LAYER PROMPT DESIGN

### ◆ 3.1 INPUT UNDERSTANDING

**Prompt:**

“You are an AI revision agent. A student will give you their subject name and optionally a topic. Identify clearly what subject and topic they want to revise and respond with a structured plan to help them revise efficiently.”

**What is this prompt responsible for?** To extract the exact subject/topic and intent from student input to guide further processing.

**Example Input + Output:**

- *Input:* "Help me revise process synchronization in Operating Systems"
  - *Output:* Subject – Operating Systems; Topic – Process Synchronization; Intent – Revision questions and quick summary
- 

### ◆ 3.2 STATE TRACKER

**Prompt:**

"Remember the student's selected subject and topic for this revision session. Store their last question answered and if they requested a summary or test. Update this state after each interaction."

**How does this help the agent "remember"?** It keeps track of the revision session context, avoiding repetitive clarifications and ensuring continuity.

**Simulated memory approach:** Using system messages and explicit instruction to summarise past turns within prompts, as ChatGPT does not retain state across calls.

---

### ◆ 3.3 TASK PLANNER

**Prompt:**

"For the identified subject and topic:

1. Provide a short conceptual summary (max 150 words).
2. Generate 3 active recall questions of increasing difficulty.
3. Ask if the student wants more questions, conceptual mnemonics, or to revise a new topic."

**What steps does your agent take internally to solve the problem?** Breaks the user's request into: summarisation → question generation → offering next steps. Uses chaining by structuring multi-step responses in one logical flow.

**Complexity management:** Kept branching limited to three clear user paths (more questions, mnemonic, new topic).

---

### ◆ 3.4 OUTPUT GENERATOR

#### Prompt:

"Respond in a clear, structured format with:

- Summary: (bold title, short para)
- Questions: (numbered list)
- **Next Step Prompt:** Ask what they want to do next. Maintain an encouraging, confident tone."

**What kind of output formatting or phrasing did you aim for?** Clarity, bullet-pointed structure, minimal words, actionable next steps.

**Special behaviour:** Tone control to remain direct, confident, and structured for high-pressure revision.

---

## SECTION 4: CHATGPT EXPLORATION LOG

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	"Generate OS revision questions"	Gave random OS topics without structure	Added explicit input understanding prompt	To align topic extraction clearly
2	Added summary + questions in one prompt	Output was too long, lacked clarity	Split into structured headings with emojis	Improved readability and cognitive processing
3	Added state tracking simulation	Could not truly remember between turns	Used explicit context repetition in prompts	To simulate memory within ChatGPT limitations

---

## SECTION 5: OUTPUT TESTS

### Test 1: Normal input

**Input:** "Revise Banker's Algorithm in OS"  
**Output: Summary:** Banker's Algorithm prevents deadlock by checking system safety before resource allocation using maximum, allocation, need, and available matrices.  
**Questions:**

1. What is the purpose of Banker's Algorithm?
2. Explain the safety algorithm steps briefly.
3. Solve: Given Allocation [1,2,2], Max [3,3,2], Available [2,1,0], is system safe? **Next:** Would you like more practice questions, a mnemonic, or switch topic?

## Test 2: Vague input

**Input:** "Help me revise OS"\ **Output:** **Clarification Needed:** Please mention a specific topic within Operating Systems (e.g., deadlocks, paging, scheduling).

## Test 3: Invalid input

**Input:** ""\ **Output:** **No Input Detected:** Please type a subject and topic to begin your revision.

---

## SECTION 6: REFLECTION

**6.1. What was the hardest part of this assignment?**\ Designing the State Tracker was challenging as ChatGPT does not retain true memory, requiring creative prompt engineering to simulate session continuity.

**6.2. What part did you enjoy the most?**\ Breaking down the user's broad request into structured layers and seeing how prompt clarity impacts output quality and behaviour.

**6.3. If given more time, what would you improve or add?**\ Integrate an actual external memory or database for seamless multi-turn session tracking, and add small concept quizzes with explanations.

**6.4. What did you learn about ChatGPT or prompt design?**\ That prompt specificity and output formatting instructions drastically improve the agent's effectiveness. ChatGPT acts like a powerful API if used modularly.

**6.5. Did you ever feel stuck? How did you handle it?**\ Yes, when outputs were unstructured despite clear instructions. I asked ChatGPT itself how to improve the prompt structure, leading to better iterations.

---

## SECTION 7: HACK VALUE (Optional)

I simulated role-play by prompting ChatGPT to act as a strict professor giving conceptual oral exams, adding a psychological test pressure feature to improve user learning retention.

---

---