

## AI Agent Development Assignment – Exam Revision Use Case

### SECTION 1: BASIC DETAILS

Name: Udai Pratap Singh

AI Agent Title / Use Case: AI Agent to help students revise for exams

### SECTION 2: PROBLEM FRAMING

#### 2.1 Problem:

Students often struggle to revise effectively before exams. They don't know which questions to practice, how to test themselves, or how to cover all subjects in a structured way.

#### 2.2 Why is this agent useful?

The AI Agent helps by generating practice questions, quizzes, and summaries based on subjects or topics. It makes revision faster, more interactive, and less overwhelming.

#### 2.3 Who is the target user?

College students preparing for semester exams.

#### 2.4 What not to include?

- Full textbook-length notes.
- Automated grading of answers.
- Deep integrations with learning apps.

### SECTION 3: 4-LAYER PROMPT DESIGN

#### 3.1 INPUT UNDERSTANDING

Prompt: "You are an input interpreter. Read the student's request and classify it into one of these: {question request, summary request, quiz request}. Extract the subject and topic if mentioned."

Responsibility: Understand if the student wants practice questions, a summary, or a quiz, and identify the subject/topic.

Example Input: "Give me 5 questions on DBMS normalization"

Example Output: {intent: question request, subject: DBMS, topic: normalization}

#### 3.2 STATE TRACKER

Prompt: "You are a state tracker. Keep track of the last 3 topics the student studied. Store subject and topic, and update context after each query."

Responsibility: Simulate memory so the agent can suggest follow-up topics or avoid repeating the same ones.

#### 3.3 TASK PLANNER

Prompt: "You are a planner. Based on the intent, decide the steps:

- If question request → generate 5 practice questions.
- If summary request → generate 1-paragraph summary.
- If quiz request → generate 5 MCQs with 4 options each.

Output the steps clearly before execution."

Responsibility: Break down the task into logical actions before generating output.

#### 3.4 OUTPUT GENERATOR

Prompt: "You are a response generator. Based on the task plan, create a clear and concise response for the student. Format using markdown (e.g., numbered lists for questions, bullet points for summaries, MCQ format for quizzes)."

Responsibility: Produce final polished answers that are easy to revise.

Example Output: 5 Practice Questions on DBMS Normalization:

1. Define 1NF, 2NF, and 3NF with examples.
2. Explain the purpose of BCNF.
3. Why is normalization important in database design?
4. Differentiate between denormalization and normalization.

5. Give an example of a violation of 2NF.

#### SECTION 4: CHATGPT EXPLORATION LOG

Attempt 1: Prompt: "Generate exam questions" | What Happened: Output too generic | Change: Added subject context | Why: To make questions relevant

Attempt 2: Prompt: "Generate quiz" | What Happened: Answers missing | Change: Added 'provide correct answers separately' | Why: Students need answer key

Attempt 3: Prompt: "Summarize topic" | What Happened: Too long | Change: Asked for 1-paragraph only | Why: Keeps revision quick

#### SECTION 5: OUTPUT TESTS

Test 1: Input: "Give me 5 questions on OS Deadlocks."

Output:

1. Define deadlock with an example.
2. Explain Coffman's conditions for deadlock.
3. What are deadlock prevention techniques?
4. Differentiate between deadlock avoidance and detection.
5. Describe the Banker's algorithm.

Test 2: Input: "Give me something to revise in DBMS."

Output: Suggested Topic: Transactions & ACID Properties

Summary: ACID stands for Atomicity, Consistency, Isolation, Durability. These properties ensure reliable transaction processing in DBMS.

Test 3: Input: "asdfghjkl"

Output: "I couldn't understand your request. Try asking for questions, a summary, or a quiz on a specific subject (e.g., OS, DBMS, CN)."

#### SECTION 6: REFLECTION

6.1 Hardest Part: Designing prompts specific enough but flexible.

6.2 Enjoyed Most: Seeing exam-like questions instantly.

6.3 Improvements: Add scheduling to create daily revision plan.

6.4 Learned: Prompt design is like programming – small changes matter.

6.5 Stuck: Outputs too generic. Solved with subject+topic constraints.

#### SECTION 7: HACK VALUE

- Added answer key in quizzes.

- Simulated short-term memory with last 3 topics.

- Tested vague and invalid inputs for robustness.