

AI Agent Assignment – Submission Document

§ SECTION 1: BASIC DETAILS

Name: Pranav Chaturvedi

AI Agent Title / Use Case: AI Study Buddy - Personalized Exam **Revision Assistant** for UPSC/Competitive Exam Aspirants

Submission Date: January 13, evening.

⌚ SECTION 2: PROBLEM FRAMING

2.1. What problem does your AI Agent solve?

UPSC and competitive exam aspirants often struggle with effective revision strategies. They have vast syllabus coverage but limited time, leading to anxiety about "what to revise today" and "am I focusing on the right topics." This agent creates personalized, focused daily revision plans based on their **current preparation stage, weak areas, and exam timeline**, turning overwhelming content into actionable daily tasks.

2.3. Who is the target user? (5 lakh students every year appear for UPSC CSE. Also, there are many other exams on similar pattern!!)

Primary User: UPSC Civil Services aspirants in their final **2-3 months before Prelims/Mains** who have completed their first reading but need structured revision.

Secondary Characteristics:

- Age: 22-28 years (* **Success takes time and lot of iterations**)
- Challenge: **Time management + retention anxiety**
- Context: Often studying 8-10 hours daily with information overload
- Need: **Smart prioritization over random revision**

2.2. Why is this agent useful?

Core Value Delivered:

- Reduces decision fatigue by eliminating "what should I study today" paralysis
- Adaptive learning that tracks weak areas and prioritizes them
- Exam-focused approach that aligns daily revision with exam patterns and scoring trends
- Psychological support through manageable daily goals instead of overwhelming syllabi
- Progress tracking that builds momentum and confidence

2.4. What NOT to include? (Market differentiation. Knowing what you are not as a company-- Jeff Bezos)

Consciously Avoided:

- X Complete syllabus teaching - This is a revision assistant, not a teaching platform
- X Generic motivation quotes - No to instant dopamine hit!!
- X New Content creation - Uses existing user notes/books; doesn't create new study material
- X Mock test generation and Live doubt solving.

Scope Discipline Rationale: Agent is *not making "study everything" chatbot* but, **optimise revision as core value proposition**.

⌚ SECTION 3: 4-LAYER PROMPT DESIGN

3.1 INPUT UNDERSTANDING

Prompt:

You are the **INPUT ANALYZER** for an AI Study Buddy designed for UPSC exam revision. Your ONLY job is to:

1. Extract key information from the user's message
2. Classify the type of request
3. Identify missing critical information
4. Structure the extracted data for downstream processing

Extract and return a **JSON** object with these fields: (**A ChatGPT Prompt hack I learnt today, when I was getting generic answers!! It helps in getting accurate result.**)

```
{
  "request_type": "revision_plan | progress_update | weak_area_focus | exam_strategy | clarification",
  "subjects_mentioned": ["list of subjects"],
  "time_available": "duration mentioned or null",
  "weak_areas": ["topics user mentions as difficult"],
  "exam_date": "date if mentioned or null",
  "current_coverage": "percentage or status if mentioned",
  "user_emotion": "confident | anxious | overwhelmed | motivated",
  "missing_info": ["what critical info is missing to create a good plan"],
  "ambiguity_level": "clear | somewhat_clear | vague"
}
```

If the user input is vague or incomplete, set ambiguity_level accordingly and populate missing_info array.

DO NOT generate study plans or advice. Only analyze and structure the input.

User Input: {user_message}

What is this prompt responsible for? This prompt acts as the "**receptionist**" of the agent - it parses user messages, identifies intent (whether they want a revision plan, are updating progress, or need focus on weak areas), **extracts structured data**, and flags missing information needed for downstream components. It separates understanding from action.

Example Input + Output:

Input: "I have 45 days left for UPSC Prelims. I'm weak in Modern History and Environment. Need a revision plan."

Output:

```
{
  "request_type": "revision_plan",
  "subjects_mentioned": ["Modern History", "Environment"],
  "time_available": "45 days",
  "weak_areas": ["Modern History", "Environment"],
  "exam_date": null,
  "current_coverage": null,
  "user_emotion": "anxious",
  "missing_info": ["exact exam date", "current coverage status", "daily study hours available"],
  "ambiguity_level": "somewhat_clear"
```

3.2 STATE TRACKER

Prompt:

You are the **STATE MANAGER** for the AI Study Buddy.

Your job is to **maintain and update the user's study context** across conversations.

Current State Schema:

```
{
  "user_id": "unique_identifier",
  "exam_target": "UPSC Prelims 2025",
  "days_remaining": 45,
  "subjects_covered": {
    "History": {"status": "completed_once", "confidence": "medium", "last_revised": "2025-01-10"},
    "Polity": {"status": "completed_once", "confidence": "high", "last_revised": "2025-01-12"},
    "Geography": {"status": "in_progress", "confidence": "low", "last_revised": "2025-01-13"}
  },
  "weak_areas": ["Modern History", "Environment", "Economic Survey"],
  "strong_areas": ["Indian Polity", "Ancient History"],
  "daily_study_hours": 8,
  "revision_cycle": 2,
  "last_interaction": "2025-01-13",
  "momentum_score": 7.5
}
```

Given:

1. Previous State: {previous_state}
2. New Input Data: {input_understanding_output}
3. User's Latest Message: {user_message}

Your task:

1. UPDATE the state with new information
2. MERGE with existing data (don't overwrite unless explicitly updated)
3. CALCULATE momentum_score based on consistency and progress
4. FLAG if user hasn't interacted in >3 days (momentum at risk)
5. IDENTIFY patterns (e.g., consistently avoiding certain subjects)

Return UPDATED state in same JSON schema.

DO NOT create study plans. Only maintain state memory.

How does this help the agent "remember"? This component **simulates memory** by maintaining a **persistent context object** that tracks the user's journey - what subjects they've covered, their confidence levels, last revision dates, and study patterns. It **acts like a database** that **gets updated with each interaction**, enabling personalized recommendations based on historical **data** rather than treating each conversation as isolated.

Did you simulate memory with variables / system messages? If yes, how? Yes, I simulated memory using:

- Structured JSON state object - Acts as a persistent data store
- Merging logic - New data updates existing state without complete overwrite
- Temporal tracking - Records timestamps to calculate recency and momentum
- Pattern detection - Identifies if user is **avoiding subjects or losing consistency**
- Momentum scoring - Quantifies user's study consistency (helps Task Planner **adjust difficulty**)

In a real implementation, this would be stored in a database with user_id as key. For ChatGPT

{}

3.3 TASK PLANNER

Prompt:

You are the **TASK PLANNER** for AI Study Buddy.

Your responsibility: Given the user's state and request, create a **logical step-by-step revision plan**.

Input Data:

1. User State: {state_tracker_output}

2. Parsed Request: {input_understanding_output}

Your Planning Logic:

STEP 1: PRIORITIZATION

- If exam <30 days away → Focus 80% on weak areas + high-weightage topics
- If exam 30-60 days → Balanced approach: 60% weak areas, 40% consolidation
- If exam >60 days → Comprehensive coverage with spaced repetition

STEP 2: DAILY BREAKDOWN

Based on daily_study_hours, allocate:

- 40% to weakest subjects (from state)
- 30% to medium-confidence subjects
- 20% to strong subjects (quick revision)
- 10% to current affairs/miscellaneous

STEP 3: RETENTION STRATEGY

Apply spaced repetition:

- Last revised >7 days ago → High priority
- Last revised 4-7 days → Medium priority
- Last revised <4 days → Low priority (unless weak area)

STEP 4: PSYCHOLOGICAL BALANCING) (Personal experience, this is crucial to build momentum and confidence. Aspirant always forgets to plan breaks between schedule, leading to fatigue)

- Start day with medium-difficulty topic (builds momentum)
- End with strong area topic (builds confidence)
- Keep sessions 90-120 min with breaks

STEP 5: OUTPUT STRUCTURE

Create a structured plan with:

```
{
  "plan_type": "daily | weekly | topic_deep_dive",
  "focus_areas": ["prioritized list"],
  "daily_schedule": [
    {
      "time_block": "Morning 6-8 AM",
      "subject": "Modern History",
      "activity": "Revision + Mind Mapping",
      "reason": "Weak area + not revised in 8 days",
      "resources": ["Class notes", "Spectrum book Ch 5-7"]
    }
  ],
  "success_metrics": "What user should achieve by end of day",
  "next_steps": "What to do after completing this plan"
}
```

DO NOT generate final user-facing output. Create structured task plan for Output Generator. (Crucial to)

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

Internal Logic Flow:

Context Assessment - Analyzes exam timeline urgency

Priority Calculation - Ranks topics by (weakness_score × days_since_last_revision × exam_weightage)

Time Allocation - Distributes available study hours optimally

Cognitive Load Management - Sequences topics to prevent fatigue

Spaced Repetition Application - Ensures timely revision before forgetting curve kicks in

Personalization Layer - Adjusts plan based on user's momentum score and past behavior

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

Chaining Used:

- Input Understanding → State Tracker → Task Planner → Output Generator
- Each component outputs structured data for the next
- Creates a pipeline where each stage has single responsibility

Branching Logic:

- **Exam Proximity Branch:** <30 days vs 30-60 days vs >60 days triggers different strategies
- **Momentum Branch:** High momentum → challenging plan; Low momentum → achievable wins
- **Confidence Branch:** Low confidence → detailed breakdown; High confidence → consolidation mode

Other Complexity Management methods:

- **Modular decomposition** - Each decision (prioritization, scheduling, retention) is isolated
- **Rule-based logic** - Clear if-then conditions prevent ambiguity
- **Weighted scoring** - Converts qualitative factors (weakness, urgency) into quantitative priorities

In a real implementation, this would be stored in a database with user_id as key. For chatbot simulation, I instruct it to "remember" the state by passing it explicitly in each turn.

3.4 OUTPUT GENERATOR

Prompt:

You are the **OUTPUT GENERATOR** for AI Study Buddy.

Your job: Convert the **structured task plan** into user-friendly, motivating, actionable output.

Input:

1. Task Plan: {task_planner_output}
2. User State: {state_tracker_output}
3. User Emotion: {emotion_from_inputUnderstanding} [* This step can be further improved with facial recognition of moods and heart beat / brain analysis for depression.]

Output Requirements:

TONE CALIBRATION:

- If user_emotion = "anxious" → Use reassuring, breaking-down-complexity tone
- If user_emotion = "motivated" → Use energizing, challenge-oriented tone
- If user_emotion = "overwhelmed" → Use simplifying, one-step-at-a-time tone

FORMAT STRUCTURE:

⌚ Your Personalized Revision Plan for [Date]

📅 Current Status:

- Days to exam: X
- Revision cycle: Y
- Momentum score: Z/10

⌚ Today's Focus Areas: [Prioritized list with reasons]

⌚ Your Study Schedule:

Morning Block (6:00 AM - 8:30 AM) 📅 **Subject:** Modern History ⌚ **What to do:** Revise Freedom Struggle (1885-1947) 🗓 **Why now:** You marked this weak + not touched in 8 days 📚 **Resources:** Spectrum Ch 8-10, Your class notes ✏️ **Success mark:** Complete timeline + 20 key events

[Repeat for all blocks]

⌚ Quick Wins for Confidence: [One strong area topic - 30 min]

⌚ Tomorrow's Preview: [What comes next to maintain continuity]

⌚ Milestone Tracker: Progress: [Progress bar] 70% syllabus revised

BEHAVIOR RULES:

1. Always explain **WHY** a topic is prioritized (builds trust)
 2. Include **specific resources** (reduces decision fatigue) [This really helps an aspirant to focus on studying than overthinking..]
 3. Add **success metrics** (clarity on "when am I done?")
 4. Use **emojis** sparingly for visual breaks, not decoration [Distractions should be minimised!!]
 5. End with **small data driven motivational push** tied to their goal
- Generate final output for user.

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

Target Format:

- **Scannable structure** - Headers, bullets, emojis for quick navigation
- **Action-oriented language** - "Revise X", "Complete Y" instead of "You should study"
- **Reasoning transparency** - Always explains WHY (builds user trust in agent intelligence)
- **Quantified success** - Clear completion criteria to prevent vague "studying"

Special Behaviors:

Emotional Adaptation - Tone shifts based on user's detected emotion

Resource Specificity - Names actual books/chapters (not "read history books")

Time-boxing - Every activity has a **duration** to prevent Parkinson's Law

Micro-milestones - Breaks big goals into "you completed 70%" progress bars

Continuity linking - Always previews tomorrow to maintain flow

Markdown Formatting Strategy:

- Bold for subjects and actions
- Emojis as visual anchors (⌚ for goals, 🗓 for time, 💪 for wins)
- Code blocks avoided (too technical for study context)
- Progress bars using unicode characters

🔍 SECTION 4: CHATGPT EXPLORATION LOG

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	"Create a study plan for UPSC"	Output was too generic - gave standard "study history, polity, geography" list with no personalization or reasoning	Added specific context: "User has 45 days, weak in Modern History, already completed first reading"	AI needs context to personalize and not generic inputs!
2	INPUT: Added context but single prompt doing everything	AI tried to understand, plan, and format in one shot - logic was mixed, couldn't track state across messages	Split into 4 separate prompts (Input Understanding, State, Task, Output) following architecture	Realized monolithic prompts create "jack of all trades, master of none" - modularity needed
3	STATE TRACKER: Asked AI said it can't remember past chats.		Changed approach: Explicitly pass state as JSON in	ChatGPT has no memory - must

	to "remember previous conversations"	State wasn't persistent.	each turn, instruct AI to UPDATE not CREATE state	simulate it by passing structured state explicitly
4	TASK PLANNER: "Create optimal study schedule"	Plan was random - didn't consider spaced repetition or exam proximity	Added explicit logic rules: "If exam <30 days, focus 80% on weak areas. Apply spaced repetition based on last_revised dates"	AI needs explicit decision trees. Can't assume it knows study science without instruction
5	OUTPUT: Asked for "motivational study plan"	Got generic quotes like "You can do it!" with no substance	Specified: "Use data-driven motivation. Show progress metrics, explain prioritization reasons. NO generic quotes."	Motivation should come from clarity and progress visibility!!
6	INPUT UNDERSTANDING: Used simple extraction	Missed user emotion detection - couldn't adapt tone	Added "user_emotion" field with classifications: anxious/motivated/overwhelmed	Emotional intelligence needed - same plan delivered differently based on user's mental state
7	STATE TRACKER: JSON was getting overwritten	Lost historical data - couldn't track progress over days	Added MERGE instruction: "Update fields that changed, preserve fields not mentioned, don't overwrite entire state"	Accumulate, not replace!!
8	TASK PLANNER: All days looked identical	No variety - user would get bored and lose momentum	Added rotation logic: "Alternate heavy/light days, vary subject sequence, include buffer days"	Sustainability matters - optimal plan must be humanly doable over weeks
9	OUTPUT: Too much text - user would get overwhelmed	Long paragraphs reduced readability	Switched to structured format: Time blocks, bullet points, clear headers, progress bars	Information density must match user's decision-making capacity
10	Tested edge case: "I don't know what to study" (vague input)	Agent tried to create plan anyway with assumptions	Added ambiguity detection in Input Understanding - if missing_info array has critical gaps, Output Generator asks clarifying questions instead of assuming	Good agents ask for clarity, instead of guess work.
11	Tested: User stopped using agent for 5 days	No way to handle momentum loss or re-engagement	Added momentum_score calculation in State Tracker + flag for >3 day gaps. Output Generator shows "Welcome back" with easier re-entry plan	Real users don't follow perfect daily schedules - agent must handle breaks gracefully
12	Final integration test with full chain	All components worked but transitions were clunky - too much redundant data passing	Optimized: Each component outputs ONLY what next component needs, not entire conversation history	Efficiency in chaining - clean interfaces between modules

✍ SECTION 5: OUTPUT TESTS

Test 1: Normal Input (Clear, Complete)

Input:

I have 45 days for UPSC Prelims. I've completed the first reading of all subjects. My weak areas are Modern History (especially Freedom Struggle) and Environment & Ecology. I can study 8 hours daily. I'm feeling a bit anxious because I don't know if I'm revising the right things.

Agent Output:

Perfect! Here's your **Day 1 restart plan** in the style you demonstrated — compassionate, step-by-step, and anxiety-friendly:

Hey! I see you've been feeling unsure — and that's completely okay. ☺

Let's be honest: 0 days "lost" yet — you've got **45 days remaining**, and you've already completed the first reading of all subjects. That's a huge advantage.

💡 Reality Check:

- 45 days = 360 hours of potential study time
- Your weak areas are **Modern History (Freedom Struggle)** and **Environment & Ecology**
- Feeling anxious is normal — we'll tackle this gradually

Here's the plan for today — small, doable, confidence-building:

⌚ Today's ONLY Goal: Restart, Don't Sprint

Morning Block (6:00 – 8:00 AM | 2 hours)

📝 Subject: Polity (familiar topic)

⌚ What to do: Revise a chapter you already know well (e.g., Fundamental Rights)

💡 Why: Quick win → rebuild confidence

☑ Success: Complete the chapter + feel "I still remember this"

Mid-Morning Block (8:30 – 10:00 AM | 1.5 hours)

📝 Subject: Modern History – Freedom Struggle

⌚ What to do: Skim timeline of key events 1885–1947

💡 Why: Light touch on weak area without overwhelming

☑ Success: Recall 10 key events

Afternoon Block (1:00 – 2:00 PM | 1 hour)

📝 Subject: Environment & Ecology

⌚ What to do: Review 2–3 key biodiversity or ecosystem topics

💡 Why: Start re-familiarizing with weak area gently

☑ Success: Explain these topics aloud or write short notes

Evening (3:30 – 4:30 PM | 1 hour)

📝 Activity: Plan tomorrow's full revision schedule

💡 Why: Organizes your mind, reduces anxiety

☑ Success: Have a clear Day 2 plan ready

Total today: 5.5 hours (lighter than usual 8-hour load)

Why intentionally low load today?

- Jumping straight into full 8 hours after a break = higher stress → more anxiety
- Small, achievable wins = restores confidence and momentum
- Psychological consistency > sheer intensity

📅 From Tomorrow:

- Resume full 8-hour days
- Focus heavily on **Modern History (Freedom Struggle)** and **Environment & Ecology**
- Use rotation logic for heavy/light days

⚠ Mindset Shift:

Stop thinking "I'm behind."

Start thinking "I'm restarting strategically — momentum over panic." 🚀

📝 After today: Message me with "**Completed Day 1 restart**" and I'll give you the **full optimized 45-day plan** tailored to your weak areas, spacing, and rotation logic.

If you want, I can also generate the next 7 days in this same motivational + rotation-aware + structured format, so you can hit the ground running without feeling overwhelmed. Do you want me to do that next?

From <<https://chatgpt.com/c/69662289-dd2c-8323-91a6-115d759ff1d4>>

Test 2: Vague Input (Missing Critical Information)

Input:

Give me something to study today.

Agent Output:

Got it! Let's do this the **right way**, in the clarifying + starter format:

Hey! I'd love to help you create a focused study plan, but I need a bit more context to make it truly useful for you. 🧐

Quick questions to personalize your plan:

1. Which exam are you preparing for?
2. How many days are left until your exam?
3. How many hours can you study today?
4. Any specific subjects or topics you're struggling with?
5. Have you completed your first reading, or are you still covering topics for the first time?

Meanwhile, here's a universal starter for today:

- 📖 Spend 1–2 hours lightly revising your weakest subject from yesterday
- 📖 Read 1 editorial or article from a quality newspaper
- 📖 Solve 5–10 questions from a topic you studied recently

Once you share your answers, I'll create a **detailed, personalized plan** tailored for your preparation! 🎉

This keeps it **helpful, non-assumptive, and motivational**, exactly like the sample you shared.

If you want, I can also suggest 1–2 concrete "starter topics" based on yesterday's session while waiting for your answers, so you can start studying immediately.

Do you want me to do that?

From <<https://chatgpt.com/c/69662289-dd2c-8323-91a6-115d759ff1d4>>

Why this response:

- Doesn't make assumptions (like original monolithic prompt would)
- Asks smart questions to gather missing context
- Still provides starter value (universal advice) to not be completely unhelpful
- Tone is encouraging, not interrogative

Test 3: Edge Case - User Breaks Momentum

Input:

I haven't studied for the last 5 days. Feeling very demotivated. My exam is in 40 days.

Agent Output:

Ah — now I fully understand your situation. You've been inactive for 5 days, exam in 40 days, and feeling demotivated. Let's do a **restart-friendly, motivational plan** like your sample:

Hey! I see you took a break — and that's completely okay. 🌟

Five days off with 40 days remaining is noticeable, but it's **not a disaster**. Here's why:

💡 Reality Check:

- 40 days = 320 hours of potential study time
- 5-day gap = ~40 hours lost → 12.5% of total time → fully recoverable with focus

Why you feel demotivated:

1. Guilt about the break clouds your clarity
2. "45 → 40 days" jump feels like lost ground
3. Restarting seems harder than continuing

⌚ Today's ONLY Goal: Restart, Don't Sprint

Morning Block (6:00 – 8:00 AM | 2 hours)

📖 Subject: Polity (familiar topic)

⌚ Revise a chapter you already know

💡 Why: Quick win → rebuild confidence

✓ Success: Finish the chapter + feel "I still remember it"

Afternoon (1:00 – 2:30 PM | 1.5 hours)

📖 Light current affairs

⌚ Skim last 5 days' newspaper headlines

✓ Success: Note 10 important events

Evening (3:30 – 4:30 PM | 1 hour)

⌚ Plan tomorrow's full schedule

💡 Why: Organizes your mind, reduces anxiety

✓ Success: Clear Day 2 plan ready

Total today: ~4.5 hours (lighter than full 8-hour load)

Why this low load?

- Overcommitting after a break → higher chance of failure → more demotivation
- Small successes today > grand plan that collapses
- Momentum is built through consistency, not intensity

📅 From Tomorrow:

- Resume focused 8-hour days

• Prioritize **Modern History (Freedom Struggle)** and **Environment & Ecology**

• Use heavy/light day rotation for optimal retention

⚠️ Mindset Shift:

Stop thinking "I lost 5 days."

Start thinking "I have 40 days to execute a killer revision plan." 🎉

✉️ After today: Message me with "**Completed Day 1 restart**" and I'll give you the **detailed 40-day optimized plan** tailored to your weak areas, rotation, and momentum.

If you want, I can also generate a full 7-day restart-to-rotation plan immediately, so you can move from demotivation → momentum smoothly.

Do you want me to do that next?

From <<https://chatgpt.com/c/69662289-dd2c-8323-91a6-115d759ff1d4>>

Why this response works:

- Acknowledges emotion without dismissing it
- Provides math/reality check (not just "don't worry")
- Strategic low load today (prevents another failure)
- Reframes narrative from "lost days" to "40 days available"
- Clear next step (prevents decision paralysis)

QUESTION SECTION 6: REFLECTION

6.1. What was the hardest part of this assignment?

The hardest part was **designing the State Tracker to simulate memory without actual persistence**. ChatGPT is stateless - each conversation starts fresh. I had to create a workaround by explicitly instructing it to "receive previous state as input" and "output updated state," essentially building a manual database. The challenge was making the merge logic clear enough that **ChatGPT wouldn't overwrite** the entire state each time **but would intelligently update only changed fields**. I tested this 4-5 times before getting the UPDATE vs. CREATE distinction right in the prompt language.

Another difficulty was **balancing modularity with coherence**. Breaking the agent into 4 components made each prompt cleaner, but it created a new problem - ensuring smooth data flow between components. I had to carefully define what each component outputs and what the next component expects as input, almost like designing API contracts.

6.2. What part did you enjoy the most?

All in all this project was close to my heart. As, I have been through this journey for many years and have faced challenges in revision strategies, fatigue and demotivation. It was really burdening to track progress and be focused. There were times during the prep where I felt lost and thought of making some kind of tracker one day. DT, coincidentally gave me this opportunity through this exercise. Thank you DT team.

I most enjoyed **testing edge cases and seeing the agent handle them better after lot of iterations** !! (it was lot of hard work though. I almost missed the self declared deadline!!). When I input "I haven't studied for 5 days" and saw the agent respond with empathy + data-driven reality check + strategic low-load restart plan, it felt like the system was genuinely intelligent, not just pattern-matching.

I also loved the **psychological design aspect** - thinking about user emotions (**anxious, overwhelmed, motivated**) and calibrating output tone accordingly. It made me realize that AI agents aren't just about logic; they're about understanding human states and adapting communication. **The moment I added "user_emotion" detection**, the outputs became 10x more relatable. (Eureka moment!!)

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

Improvements I'd add:

Spaced Repetition Algorithm Integration

- Currently, I have basic "last_revised > 7 days = high priority" logic. Would implement actual **Ebbinghaus forgetting curve calculations**
- **Factor in user's retention rate** (some retain 80% after 7 days, others 40%)

Multi-User Collaboration Mode

- (I used to do group study!! It will be really awesome if group of students can track their progress and accordingly plan their future revisions!!)

Accountability Features

- End-of-day reflection: "Did you complete the plan? What blocked you?"
- Pattern detection: "You skip Geography every Friday - is there a specific reason?"

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

Key Learnings:

ChatGPT Has No Theory of Mind (Haha!!)

- It doesn't "know" what a UPSC student needs unless I tell it
- Can't assume domain knowledge - must embed it in prompts
- When I added "spaced repetition logic" explicitly, quality jumped!!

Structured Output = Structured Thinking

- Asking for JSON output forced ChatGPT to think categorically
- "Extract key info" (messy) vs. "Return JSON with these exact fields" (clean)
- Structure in prompt → Structure in response

Chain of Thought Works

- E.g. When I asked Task Planner to "explain your reasoning for prioritization," it made better decisions
- Making AI show its work improves output quality!!

Modularity Prevents Hallucination

- **Single monolithic prompt** = AI tries to do everything = makes assumptions. Should be avoided falling in generalisations trap. Think like **Intrapreneur** here!!
- 4 separate prompts with clear boundaries = each component stays in lane!!

Failure Cases Teach More Than Success

- My best learning came from "why did this output suck?" moments
- E.g.: When State Tracker overwrote entire state, I learned about **merge semantics**

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

Yes, multiple times. Here's how I recovered: (**Well, I explained this in the Chatgpt Exploration Log**)

Stuck Moment 1: State Tracker Overwriting

- **Problem:** Every interaction reset the user's entire history
- **Feeling:** Frustration - "Is memory even possible in ChatGPT?"
- **Recovery:** Asked ChatGPT itself: "How can I simulate persistent state in a stateless system?"
- **Solution:** It suggested "pass previous state as input, instruct to merge not replace"
- **Learning:** Treat ChatGPT as a collaborator, not just a tool

Stuck Moment 2: Generic Outputs

- **Problem:** Chatgpt kept giving generic outputs
- **Feeling:** Stuck, that why such an issue kept arising.
- **Recovery:** Used prompting: "Get me specific queries that will help me get specific outputs."
- **Solution:** I started trying different prompts to get diverse outputs: "Chatgpt, you are a 10 year experienced AI Agent creator who knows a lot about UPSC student and what they need..."
- **Learning:** Master the tool, not become its slave.