🔍 Assignment: Build a Simple AI Agent

🧠 Objective:

To assess your **curiosity, learning agility, and problem-solving approach**, not just your technical skills.

Even if you're new to coding or AI, this is your chance to show how you **think, explore, and work with AI tools like ChatGPT** to build something meaningful.

📦 What You'll Build:

Design a **simple AI Agent** (like a chatbot or a task assistant) using the 4-step architecture model. You don't need to code everything perfectly. Instead, show how far you can go by:

- Asking the right questions
- Iterating on prompts
- Learning through ChatGPT

💻 Reference Materials:

Use these resources to understand how AI Agents are designed and prompted:

1. **Thread on AI Architecture (ChatGPT Share Link)**
   📜 Read Here
2. **4-Step AI Agent Design Approach (Google Doc)**
   📄 View Here
3. **Sample Prompts for Each Step (Web Development Use Case)**
   📋 Reference Here

🛠️ Your Task:

1. **Choose a Use Case:**
   Pick a *simple problem* that an AI Agent can help with.
   Example:
   - "AI Agent to suggest daily writing prompts for bloggers."
   - "AI Agent to help students revise for an exam."

2. **Design the 4 Layers of Prompts:**
   For your use case, create prompts for:

   - **Input Understanding** (What is the user asking?)
   - **State Tracker** (What context/state is remembered?)
   - **Task Planner** (What steps does the agent take?)
   - **Output Generator** (How does it respond clearly?)

3. 💡 Use the reference documents to guide you. You can modify the prompts or create your own.

4. **Iterate & Build with ChatGPT:**
   Use your prompts inside ChatGPT.
   - Ask follow-up questions if the output is not as expected.
   - Keep improving your prompts and logic step-by-step.

5. **Submit an Approach Document:**
   This is the most important part.

   Your document should include:
   - Your **chosen use case**.
   - Prompts for each of the 4 steps.
   - Snapshots or text logs of your **ChatGPT interactions**.
   - **What you learned**, where you got stuck, and how you overcame it.
   - Any code samples or system messages if generated.

6. 📌 *Think of this as a journal of your journey with ChatGPT.*

AI Agent Assignment – Submission Format

Note: This document is both your **journal** and **artifact of thought**. We're evaluating how you *think, build, and debug*, not just what you build.

SECTION 1: BASIC DETAILS

Name: Sakshi Rajendra Patil

AI Agent Title / Use Case:
InfoMate AI : *Your Intelligent Companion for Learning & Content Creation*

SECTION 2: PROBLEM FRAMING

1.1. What problem does your AI Agent solve?

InfoMate AI helps users quickly find structured, reliable, and concise research information on any topic — especially factual and numeric queries like GDP, statistics, or trends — which often require time-consuming manual searches.

1.2. Why is this agent useful?

blog writing + information assistant like WriteWise. This AI agent simplifies the content creation process by helping users research any topic, summarize key insights, and generate well-structured blog content in minutes.

1.3. Who is the target user?

Students, content writers, researchers, and data-driven professionals who need fast, structured information to support academic, analytical, or writing work.

1.4. What *not* to include?

The agent avoids generating opinions, speculative data, or irrelevant conversational banter. It doesn't support long-term memory or deeply personalized interactions for now — the focus is on factual research.

SECTION 3: 4-LAYER PROMPT DESIGN

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

3.1 INPUT UNDERSTANDING

Prompt:

what can I search for you today. "What is the topic or concept you'd like tore search today? Please be as specific as possible."

What is this prompt responsible for?

It captures the user's intent clearly and helps focus the research on a specific, factual query.

Example Input + Output:

Input: "Tell me about unemployment rate in India in 2024"
Output: Topic identified: *Unemployment Rate in India, 2024*

🔹 3.2 STATE TRACKER

Prompt:
"The user is researching: {topic}. Store this context. Use appropriate tools to gather structured data."
You are a research assistant that answers user queries with summarized and structured data.\n

How does this help the agent "remember"?
This sets the task context using system messages and guides the agent to retain the research topic during the planning and output stages.

Did you simulate memory with variables / system messages? If yes, how?
Yes, memory is simulated by passing the topic variable through the lang-chain and explicitly referencing it across prompts.

🔹 3.3 TASK PLANNER

Prompt:
"Break the user's query into steps:

1. Search structured data using tools (e.g., for numbers, statistics)
2. Summarize key findings
3. List sources and mention tools used"

What steps does your agent take internally to solve the problem?
The agent performs tool-augmented searches (like Wikipedia, Google Search, and numeric lookups), summarizes the result, and formats the output using a structured schema.

Did you use chaining? Branching? How did you manage complexity?
Yes, tool chaining was used through LangChain's create_tool_calling_agent logic. Each step was modular — tools fetch data → parser structures → agent composes.

🔹 3.4 OUTPUT GENERATOR

Prompt:
"You must return the response in this JSON structure:

- topic:
- summary:
- sources:
- tools_used:

Be clear, concise, and source-based. Avoid speculation."

What kind of output formatting or phrasing did you aim for?
The agent uses JSON formatting for clarity and ease of parsing, with concise, professional summaries. Tone is neutral, factual, and helpful.

Any special behavior? (e.g., examples, markdown formatting, tone control, etc.)
Yes — includes:

- Markdown styling for improved readability
- Source citations
- Tool usage transparency

SECTION 4: CHATGPT EXPLORATION LOG

You may structure this as a table or bullet list.

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	*"Give research on topic"*	Output was vague and unfocused	Asked for specific topic	To improve precision
2	*"Summarize any topic using structured format"*	Output was inconsistent	Added JSON schema	To enforce structure
3	*Used system message + tools*	*Output improved but lacked citations*	*Added requirement to show sources*	*For credibility and trust*
4	*Added Pydantic OutputParser*	*Output became well-structured*	*Finalized schema format*	*To lock the format and allow validation*

SECTION 5: OUTPUT TESTS (Optional but Recommended)

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

- **Test 1:** Normal input
  Input: "I want a journaling prompt about career confusion"
  Output: *Reflect on your skills, interests, and values to identify potential career paths. Explore your fears and anxieties around career choices. Consider seeking guidance from mentors or career counselors."*

- **Test 2:** Vague input
  Input: "Give me something to think about"
  Output: *"topic": "The Fermi Paradox",*
  "summary": "The Fermi Paradox is the contradiction between the high probability of extraterrestrial civilizations existing and the lack of any contact with them. It prompts us to consider the vastness of the universe, the potential for life beyond Earth, and the possible reasons why we haven't encountered other intelligent species. This paradox leads to fascinating discussions about the nature of life, technology, and the potential challenges of interstellar travel and communication.",

- **Test 3:** Invalid input or challenge
  Input: ""
  Output: *Input should be a valid string*

SECTION 6: REFLECTION

Respond to each question briefly (3–5 sentences each):

- **6.1. What was the hardest part of this assignment?**
- The hardest part was designing prompts that balance both creativity and structure. Since the agent needed to produce blog-worthy content while maintaining factual accuracy, it was challenging to guide the model in generating information-rich yet engaging text. Another tricky area was configuring the tool integration so the agent could fetch relevant data without overloading the response or breaking flow.

- **6.2. What part did you enjoy the most?**
- I enjoyed seeing how small prompt adjustments could drastically improve the quality of the agent's writing output. It was fascinating to experiment with tone, structure, and formatting — and then watch the AI evolve into a tool that could genuinely assist with creative tasks like blog writing. Integrating tools and designing prompts that felt like giving instructions to a co-writer made the process both fun and insightful.

- **6.3. If given more time, what would you improve or add?**
- If given more time, I would add multi-turn conversation capability, allowing users to refine their blog drafts over multiple interactions. I'd also integrate a tone selector (e.g., formal, casual, persuasive) and keyword-based SEO suggestions to make the content more customizable and web-optimized. Finally, I'd include a memory feature that remembers a user's writing style or past topics to offer more personalized suggestions.

- **6.4. What did you learn about ChatGPT or prompt design?**
- I learned that effective prompt design is less about giving detailed instructions and more about breaking down a task into clear, modular stages. With ChatGPT, even small changes in phrasing can significantly impact the relevance, tone, and structure of the response. I also realized that layering prompts — like separating understanding, planning, and generation — makes the agent more controllable and reliable, especially for creative tasks like writing.

- **6.5. Did you ever feel stuck? How did you handle it?**
- Yes, I felt stuck when the outputs were either too generic or not aligned with the intended tone of a blog. Instead of starting over, I analyzed what the prompt was missing and experimented with small changes — like adding context, structure expectations, or tone guidance. I also asked ChatGPT meta-questions about how to improve the prompt itself, which helped me iterate faster and get more targeted results.

🧠 SECTION 7: HACK VALUE (Optional)

Did you go beyond the brief in any way?

- Simulated multiple users?
- Added logic for memory or role play?
- Explored chaining multiple agents?

If yes, explain in 2–3 lines.

Yes, I went beyond the brief in several ways:

- I implemented tool-based chaining using LangChain to combine search, summarization, and structured output generation into a seamless flow.

- I used Pydantic-based output parsing to enforce a clean, reliable JSON structure — making the agent's responses consistently usable and format-friendly.

- I simulated context tracking by passing state (like the user's topic) across the prompt layers, enabling more coherent and personalized responses.

While not full multi-agent chaining, I experimented with tool switching logic to select the right function (search, wiki, numeric) depending on the query type.
These enhancements made the agent feel more intelligent, flexible, and reliable for real-world blogging and research use.

Rubric for Evaluation (Work Simulation Style)

Dimension	Weight	What Good Looks Like	Red Flags
Problem Framing	20%	Clear, original use-case. Well-scoped agent behavior. Avoided vagueness. Shows user-centric thinking.	Chose a generic or unclear use-case (e.g. "make a chatbot"). Vague problem statement.
Prompt Architecture Quality	25%	Each of the 4 prompts reflects different system responsibilities. Prompts are concise, modular, and testable. Shows understanding of functional decomposition.	Prompts overlap in function, are too verbose, or fail to create modular behavior.
Exploration Quality (AI Collaboration)	20%	Good trail of follow-up questions and prompt debugging. Attempts multiple iterations. Shows ability to collaborate with ChatGPT like a teammate.	One-shot prompts. No iteration. No recovery from errors. No exploration of alternatives.
Output Clarity & Functional Coverage	15%	The generated outputs (agent responses or flow) demonstrate useful behavior. Shows command of output formatting, response variation, or state transitions.	The output is trivial, repetitive, or structurally broken.
Documentation of Process (Approach Doc)	15%	Thoughtfully documents the journey. Annotates prompts with rationale. Reflects on what worked and what didn't. Data trail is clean.	Only dumps the final result. No iterative record. No reflection or analysis.
Initiative / Hack Value	5%	Goes beyond base instructions — e.g. added context memory, simulated user inputs, nested workflows, or tried to test edge cases.	Minimum viable effort with no signs of initiative.

Time Estimate:

8-10 hours, depending on how deep you explore.

Tips for Applicants

These should be shared as part of the brief to guide execution:

Think Like an Agent Architect

- Your job isn't just to "make a chatbot" — you are defining how an **agent breaks down a user's request into logical steps.**
- Try to **separate "understanding", "thinking", and "responding"** like different departments in a company.

Ask Better Questions

- If ChatGPT doesn't give what you want, **don't settle.** Ask:
  - "How can I make this output more modular?"
  - "What if the user gives a bad input?"
  - "How do I retain memory across turns?"

- Your ability to **ask follow-up questions** shows whether you're teachable and resilient.

Design with Clarity

- Good prompts are like good APIs — **each does one job** and can be reused.
- Don't hardcode outcomes. Let the **prompt guide structure**, not content.

Debug Like a Researcher

- When things go wrong, **don't delete and restart.** Tweak. Test. Compare.
- Log what changed and why. We want to see your **diagnostic thought process**.

Document Like an Engineer

- Think of your approach doc as a **hand-off artifact** — someone else should be able to reuse or build on your logic.
- Annotate your prompts: "This prompt ensures the agent understands ambiguous user questions by…"
- Don't be afraid to **admit where you got stuck.** We value truth over polish.

Beginner-Friendly WBS: AI Agent Assignment

🎯 Total Estimated Time: **8 to 10 hours**, including orientation to AI terms
🎯 Success = Curiosity + Clarity of Thinking + Documenting the Journey

🧭 PHASE 0: Orientation – "What am I even doing here?" (60–75 min)

Task	Description	Time
0.1	**Read this brief** (assignment, structure, and expected output)	10 min
0.2	Watch/Read 1 basic explainer on "what is a prompt" and "how AI responds" *(shared by DT mentor or link in doc)*	15–20 min
0.3	Read the 3 linked docs (architecture, prompts, and example) very slowly. Take notes.	30–45 min
0.4	Make your own summary: "What are the 4 steps in AI Agent design?" in your words	10 min

🧠 PHASE 1: Choose a Use Case + Imagine the Agent (90–120 min)

Task	Description	Time
1.1	Think of 2–3 situations where an AI Agent could help YOU (in studies, writing, planning, etc.)	20 min
1.2	Choose one. Write a short note: Who is the user? What problem does the agent solve?	20 min
1.3	Write a few user examples — how would someone *talk* to this agent? (e.g., "Hey, give me a writing idea today")	15 min
1.4	Think: What should the agent do *internally* before replying? (no need for code — just logic)	20 min
1.5	Write down what the agent should NOT do (helps define the boundary of work)	10 min

🧱 PHASE 2: Design the Prompts One by One (2–2.5 hours)

You will now create a basic prompt for each layer. The goal is not perfection, but logic.

Task	Description	Time
2.1	INPUT UNDERSTANDING – Try a basic version of "Understand what the user is asking"	20–30 min
2.2	STATE TRACKER – Can the AI remember past info or simulate memory? Try simple instructions.	20–30 min
2.3	TASK PLANNER – How should the AI break the request into steps before answering?	30–40 min
2.4	OUTPUT GENERATOR – What should the final reply look like? Try formatting, tone, variation.	30–40 min

🔬 PHASE 3: ChatGPT Exploration and Debugging (2–2.5 hours)

Task	Description	Time
3.1	Run each prompt in ChatGPT (one layer at a time), copy the output	20–30 min
3.2	Note what's *not working*, and ask ChatGPT how to improve your prompt	30–45 min
3.3	Try 2–3 improved prompts for each layer	45–60 min
3.4	Save all your prompt versions + outputs in a table or doc	30 min

📄 PHASE 4: Write the Final Submission Document (2–2.5 hours)

Task	Description	Time
4.1	Section 1–2: Title, Problem, Use Case, What Agent Does	20–30 min
4.2	Section 3: Final version of all 4 prompts (with comments if possible)	30–40 min
4.3	Section 4: Exploration Log (attempts, changes, reasons)	30–40 min
4.4	Section 5: Output examples (Good/Bad inputs)	20 min
4.5	Section 6: Reflections – What you learned, what was tough	30 min

⏱️ Total Estimated Time: ~9 hours

(Possibly more if done with deep thinking and meaningful iterations.)

💡 Final Tips for First-Time Explorers

Tip	Why It Matters
❓ **If confused, *ask ChatGPT itself* what a prompt should do.**	It can teach you how to work with it if you treat it like a teammate.
🪜 **Build prompt logic step by step.**	Don't try to write perfect prompts from the start. Work like a scientist.
📝 **Keep a scratchpad next to you.**	Helps track how your thinking evolved. Makes documentation easier later.
🚫 **Don't try to "complete the assignment" fast.**	This is a simulation of how DeepThought works. Curiosity > speed.
🔍 **Use analogies to understand prompts.**	Think of each prompt like a department in a company — each doing its job.