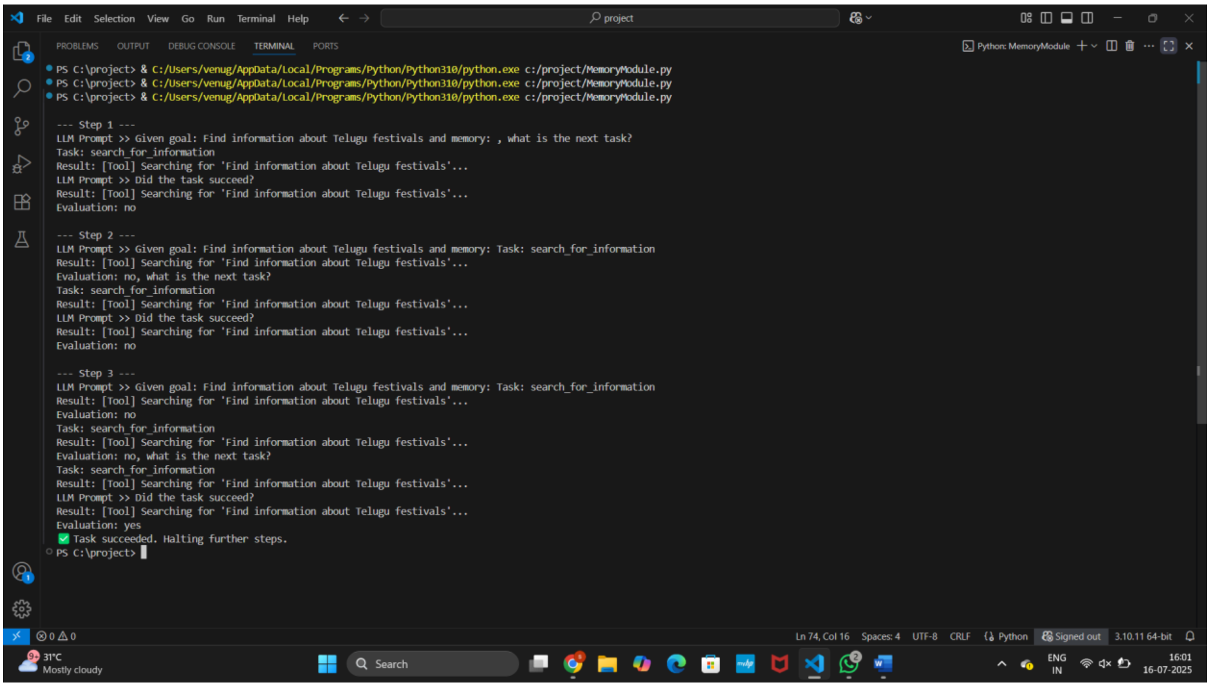
SIMPLE AGENTIC AI

```
import random
def llm(prompt):
    print("LLM Prompt >>", prompt)
    if "next task" in prompt:
        return "search_for_information"
    elif "Did the task succeed" in prompt:
        return random.choice(["yes", "no"])
    elif "search_for_information" in prompt:
        return "I found some useful articles on the topics."
    elif "replan" in prompt:
        return "try_different_way"
    else:
        return "task completed"


class Memory:
    def __init__(self):
        self.logs = []
    def store(self, entry):
        self.logs.append(entry)
    def recall(self):
        return "\n".join(self.logs[-5:])


class Tools:
    def search_for_information(self, query):
        return f"[Tool] Searching for '{query}'..."
    def try_different_way(self):
        return "[Tool] Trying alternative search method..."

class Critic:
    def evaluate(self, result):
        return llm(f"Did the task succeed?\nResult: {result}")

class Planner:
    def decide_next_task(self, goal, memory):
        return llm(f"Given goal: {goal} and memory: {memory}, what is the next task?")

class Agent:
    def __init__(self, goal):  # ✅ FIXED from _init
        self.goal = goal
        self.memory = Memory()
        self.tools = Tools()
        self.critic = Critic()
        self.planner = Planner()

    def run(self):
        for step in range(5):
            print(f"\n--- Step {step + 1} ---")
            mem = self.memory.recall()
            task = self.planner.decide_next_task(self.goal, mem)
            action = getattr(self.tools, task, None)
            if action:
                result = action(self.goal)
            else:
                result = llm(task)
            print(f"Task: {task}\nResult: {result}")
            self.memory.store(f"Task: {task}\nResult: {result}")
            success = self.critic.evaluate(result)
            print(f"Evaluation: {success}")
            self.memory.store(f"Evaluation: {success}")
            if success.lower() == "yes":
                print("✅ Task succeeded. Halting further steps.")
                break
        else:
            print("❌ Task did not succeed in given steps.")
# ✅ Correct main block
if __name__ == "__main__":
    agent = Agent(goal="Find information about Telugu festivals")
    agent.run()
```

OUTPUT:



```
PS C:\project> & C:/Users/venug/AppData/Local/Programs/Python/Python310/python.exe c:/project/MemoryModule.py
PS C:\project> & C:/Users/venug/AppData/Local/Programs/Python/Python310/python.exe c:/project/MemoryModule.py
PS C:\project> & C:/Users/venug/AppData/Local/Programs/Python/Python310/python.exe c:/project/MemoryModule.py

--- Step 1 ---
LLM Prompt >> Given goal: Find information about Telugu festivals and memory: , what is the next task?
Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
LLM Prompt >> Did the task succeed?
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: no

--- Step 2 ---
LLM Prompt >> Given goal: Find information about Telugu festivals and memory: Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: no, what is the next task?
Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
LLM Prompt >> Did the task succeed?
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: no

--- Step 3 ---
LLM Prompt >> Given goal: Find information about Telugu festivals and memory: Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: no
Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: no, what is the next task?
Task: search_for_information
Result: [Tool] Searching for 'Find information about Telugu festivals'...
LLM Prompt >> Did the task succeed?
Result: [Tool] Searching for 'Find information about Telugu festivals'...
Evaluation: yes
✅ Task succeeded. Halting further steps.
PS C:\project>
```