Bug Report Summary - S9 Improvements

I. Configuration Fixes (config/profiles.yaml)

1. Incorrect MCP Server Working Directory

Issue:

MCP server paths were hardcoded with absolute directories tied to a specific development environment, breaking portability.

Fix:

Replaced with relative paths ("." or "modules/") to ensure configuration portability across systems.

2. Disabled Memory MCP Server

Issue:

The memory server configuration was commented out, disabling long-term memory and history access.

Fix:

Uncommented the memory server entry to restore its functionalities.

II. MCP Server & Tool Enhancements (mcp_server_2.py)

1. Trailing Whitespace in Model Name

Issue:

A trailing space in QWEN_MODEL caused Ollama model invocation failures.

Fix:

Whitespace was removed for correct model recognition.

2. Inefficient Image Captioning in Web Conversion

Issue:

The tool used LLMs to caption downloaded webpage images, causing slowness, errors, and complexity.

Fix:

Refactored to simplify or remove captioning, relying on trafilatura for efficient markdown extraction.

III. Core Agent Logic Improvements (agent.py, core/loop.py, modules/perception.py)

1. input() Causing EOFError or Hangs

Issue:

The input() call would crash or hang after tool executions due to subprocess interference.

Fix:

Moved input() to a thread using asyncio.to_thread(), added error handling, and cleaned up the prompt logic.

2. Incorrect mcp_server_descriptions Structure

Issue:

mcp_server_descriptions was passed as a list instead of a dictionary, causing an AttributeError.

Fix:

Corrected the structure to a dict for AgentContext and a list for MultiMCP.

3. Verbose Tool Outputs Affecting Planning

Issue:

Long tool outputs overwhelmed the LLM and led to poor or repetitive plans.

Fix:

Summarized or truncated tool outputs before sending them to the planning LLM.

4. Agent Loop Ending Prematurely ("Max Steps Reached")

Issue:

Tasks failed early despite being solvable within step limits.

Fix:

- Context is now concise (see III.3)
- Tools are more robust and efficient (see II.2)
- Agent loop better manages states and transitions

IV. General Code Robustness

1. Simplified agent.py Loop

Issue:

Nested while True loops and multiple input() calls caused complexity and double prompts.

Fix:

Refactored to a single clean input call with proper shutdown of MCP servers on exit.