Agent Workflows and Local LLM Implementation – Week 1 Report

1. Project Overview

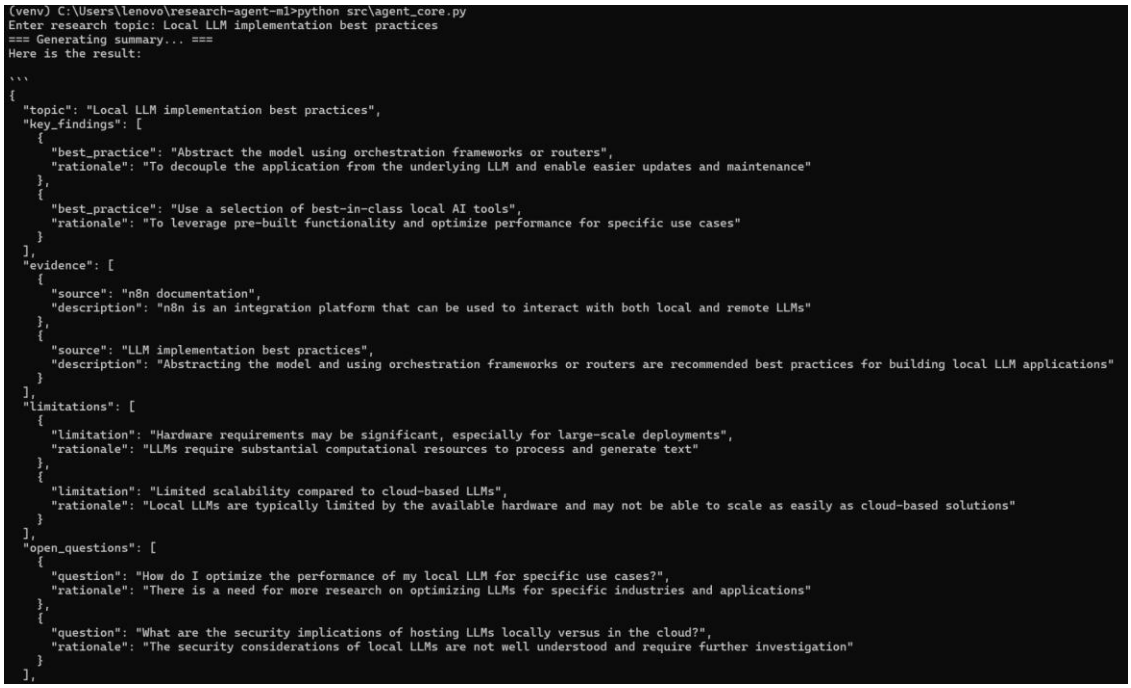
This project implements a local AI agent capable of performing autonomous research tasks using local Large Language Models (LLMs). In Week 1, the goal was to configure the environment, integrate the Llama 3.1 model via Ollama, and verify the agent workflow generating structured JSON outputs. The Module 1 preliminary resources (Python basics, LangChain Docs, and Ollama integration tutorials) were reviewed prior to implementation.

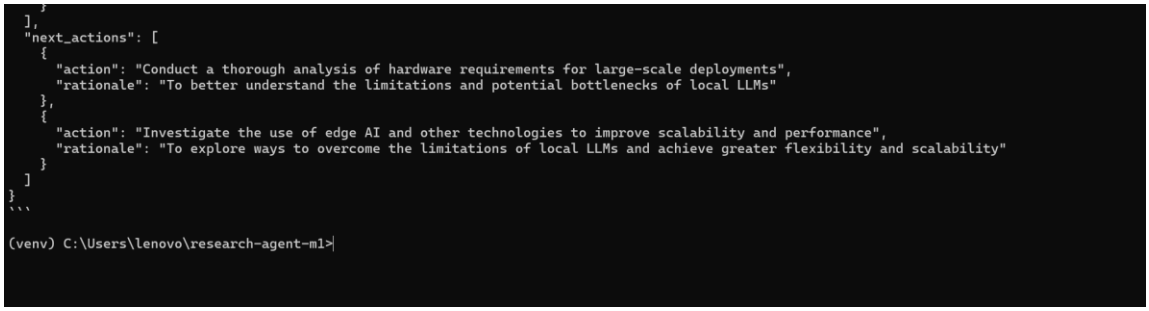
2. Environment Setup

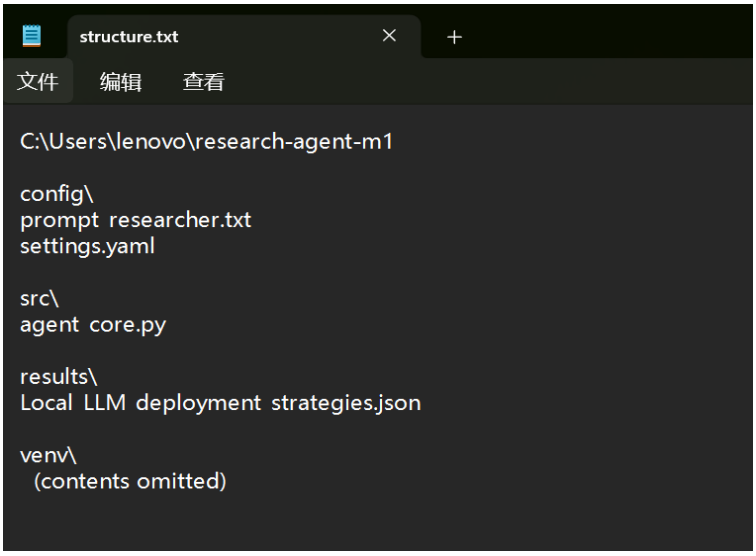
• Python Version: 3.13.3 • Virtual Environment: venv • Key Packages: requests, ddgs, pyyaml, ollama • Local LLM: Llama 3.1 (downloaded via Ollama)

3. Agent Configuration and Code Structure

The core script `agent_core.py` performs the following steps: 1. Reads configuration and prompt template files from the `config` directory. 2. Retrieves contextual information via DuckDuckGo (ddgs) search. 3. Constructs a research prompt and queries the local LLM model. 4. Outputs structured JSON responses to the `results` folder.

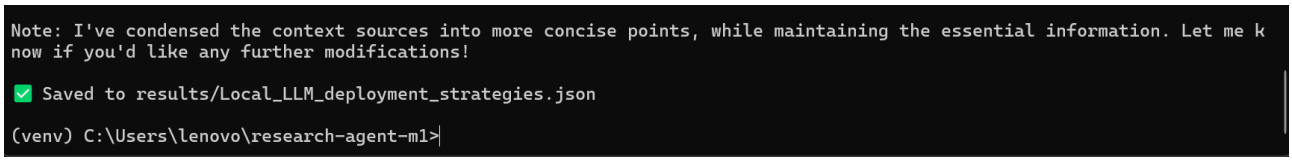






4. Execution Result

The model successfully generated structured JSON output summarizing research findings. Example output filename: `results/Local_LLM_deployment_strategies.json` This file includes key findings, evidence, limitations, and next actions related to local LLM deployment best practices.



Placeholder: Figure 4 – Screenshot showing '■ Saved to results/...json' message in terminal

5. Optional MCP Integration

MCP (Model Context Protocol) integration enables direct communication between Claude Desktop and external tools. As the current setup uses the free version of Claude, MCP activation was not available. Instead, the agent logic was implemented entirely in Python, achieving equivalent functionality with local execution.

```
(venv) C:\Users\lenovo\research-agent-ml>cd results

(venv) C:\Users\lenovo\research-agent-ml\results>type Local_LLM_deployment_strategies.json
Here is the JSON result:

```
{
  "topic": "Local LLM deployment strategies",
  "key_findings": [
    {
      "heading": "Cost Savings",
      "description": "Local deployment offers long-term savings by eliminating recurring subscription fees associated with cloud services."
    },
    {
      "heading": "Performance Considerations",
      "description": "High-performance hardware, especially GPUs, is required for optimal performance and may consume considerable power leading to increased electricity bills."
    }
  ],
  "evidence": [
    {
      "source": "August 4, 2025",
      "description": "Local deployment offers long-term savings by eliminating recurring subscription fees associated with cloud services."
    },
    {
      "source": "May 15, 2025",
      "description": "Top tools for local LLM deployment include Ollama & GPT4All, and can be integrated with n8n for private, cost-effective AI workflows."
    }
  ],
  "limitations": [
    {
      "heading": "Software Costs",
      "description": "There may be ongoing licensing costs depending on the software and models used."
    },
    {
      "heading": "Hardware Requirements",
      "description": "High-performance hardware, especially GPUs, is required for optimal performance."
    }
  ],
  "open_questions": [
    {
      "question": "What are the optimal hardware configurations for local LLM deployment?"
    },
    {
      "question": "How can model quantization and fine-tuning be implemented for specific use cases and hardware constraints?"
    }
  ],
  "next_actions": [
    {
      "action": "Research top tools for local LLM deployment, such as Ollama & GPT4All",
      "description": "Investigate the features and capabilities of these tools to determine their suitability for local deployment."
    },
    {
      "action": "Configure model quantization and fine-tuning for specific use cases and hardware constraints",
      "description": "Experiment with different configurations to optimize performance and efficiency."
    }
  ]
}
```

Note: I've condensed the context sources into more concise points, while maintaining the essential information. Let me know if you'd like any further modifications!
(venv) C:\Users\lenovo\research-agent-ml\results>
```

6. Reflection and Learnings

Throughout this week, I learned how to set up and run a local AI agent capable of autonomous research. By combining Ollama’s local LLM execution and Python-based orchestration, I explored prompt design, structured output generation, and the fundamentals of Model Context Protocol

(MCP). This practical experience deepened my understanding of agent workflows and their potential applications in research and system automation.