# Round 2 Tasks - Lynq

All the best!

## Level 1: Exploration & Setup

#### Agent Setup Challenge

To follow along, you will need the following:

A code editor like Visual Studio Code (VS Code). A programming language to work with in VS Code. We'll use Python for this. UV, a Rust-based Python package manager.

LLMs you can use:

Frontier models – API

1. GPT API from OpenAI
2. Claude API from Anthropic
3. Gemini API from Google (Free, recommended for now)
4. DeepSeek API from DeepSeek AI

**Step 1:** make simple calls to any one of the LLMs listed above, to answer any general prompt(not agentic just yet). - a building step to help you progress smoothly.

**Step 2:** from pypdf import PdfReader - use pdf reader to read and analyse a pdf of your choice. Bonus: make a small UI chat interface.

## Level 2: Building Agent Behaviors

#### Mini-Agent Build

Create MCP Tool Server (weather_mcp.py)

(Use FastMCP)
- Add a function get_weather(city: str)
- Inside, call a free weather API (like OpenWeather) OR just return mock data like "Sunny, 30°C".
- Register it as a tool with @mcp.tool()
- Run MCP server. It will start listening for tool requests.
- Connect with Client (client_agent.py)
- Your client agent already supports MCP connections.
- Point it at the weather MCP server.

Specific Deliverable:

Implement this workflow so that:

If you ask the AI: “Is it raining in Hyderabad today?”
You get a response like:
"According to the weather API, it's cloudy with light rain, 27°C."

## Submission Instructions

Push your code to a public GitHub repository.

**And submit on this forms: https://forms.gle/P91FejpjH9dnrJtR9**

Suggested structure:

```
/agentic-challenge
    ├── level1/
    │   ├── llm_call.py
    │   └── pdf_reader.py
    ├── level2/
    │   ├── weather_mcp.py
    │   └── client_agent.py
    ├── requirements.txt
    └── README.md
```

In your README.md, include:

- Setup instructions (how to run your scripts).
- Which LLM API you used.
- Sample inputs/outputs.
- Submit your GitHub link as your final deliverable.

Resources

1. Gemini CLI installation
2. Firecrawl Webscraping MCP
3. Setting MCP server with Gemini CLI
4. Prompt Engineering Basics
5. Newspaper3k docs