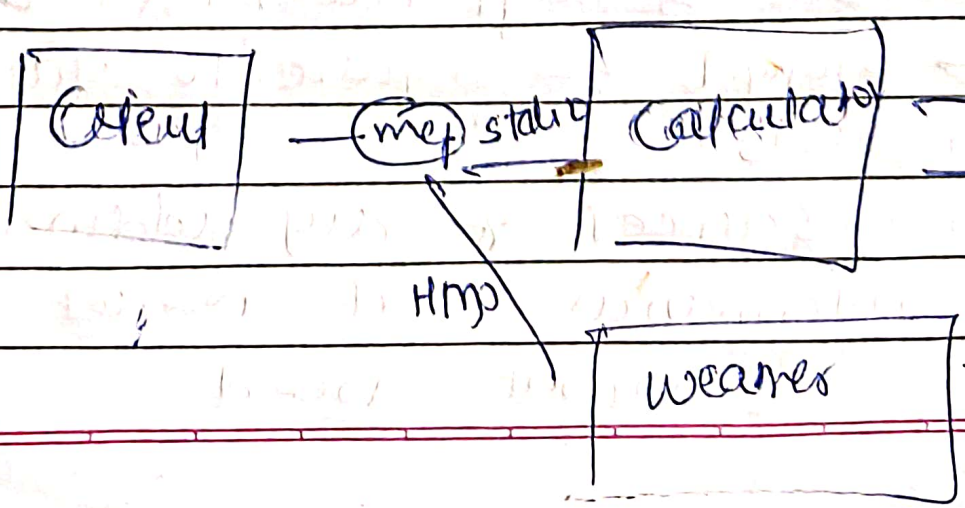
#1
>

PAGE No. 13
DATE 12/

* MCP :- Model Context protocol.
→ an interface to connect LLM & AI model with tools. Follows client server architecture
In traditional method what we do is create AI model & write a code to add tools but the problem is when you are using external API as you don't have control on it if any updation take place your code breaks that can be simplified with mcp If add a unified layer between LLM and tools
→ easy to maintain tools

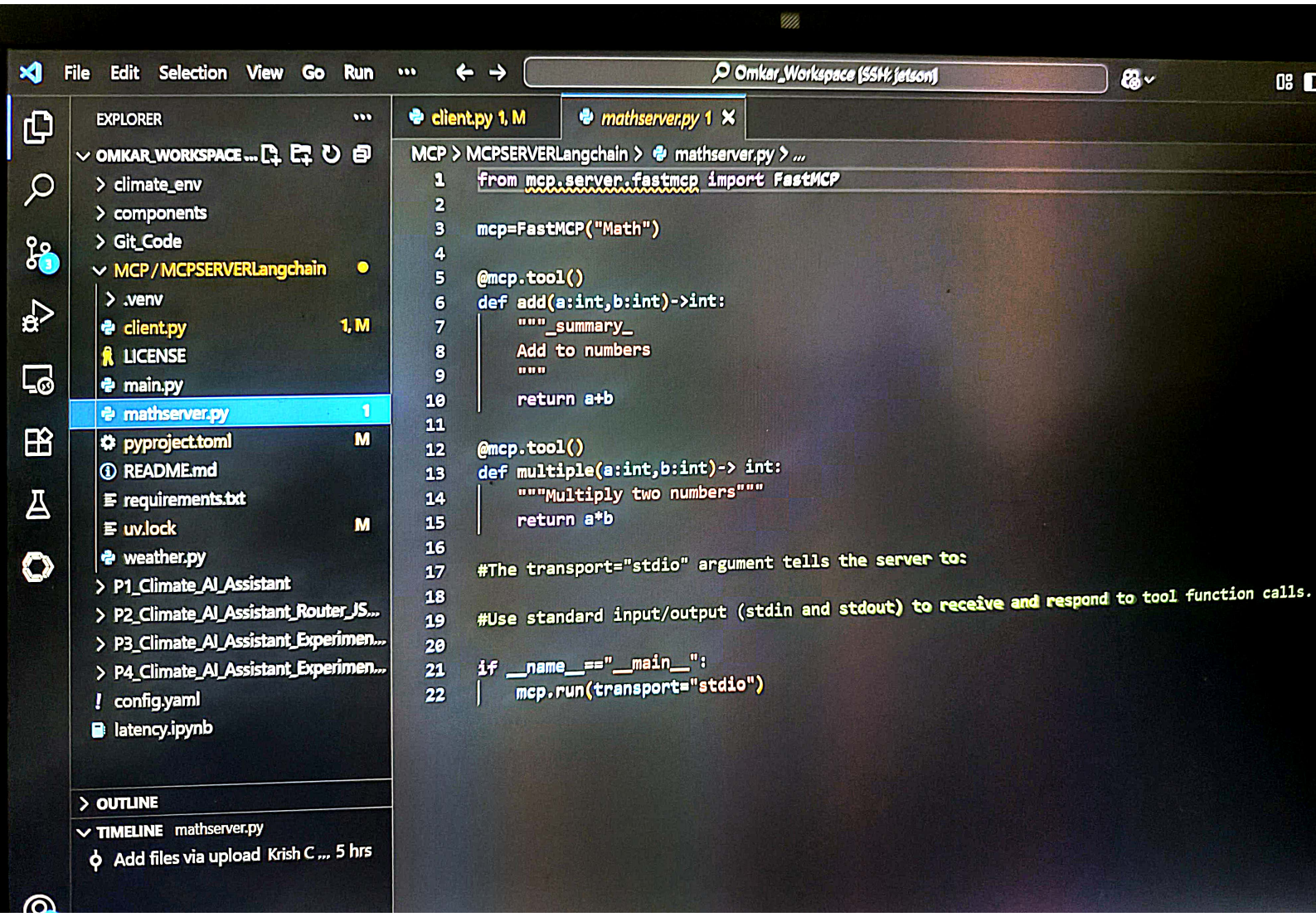


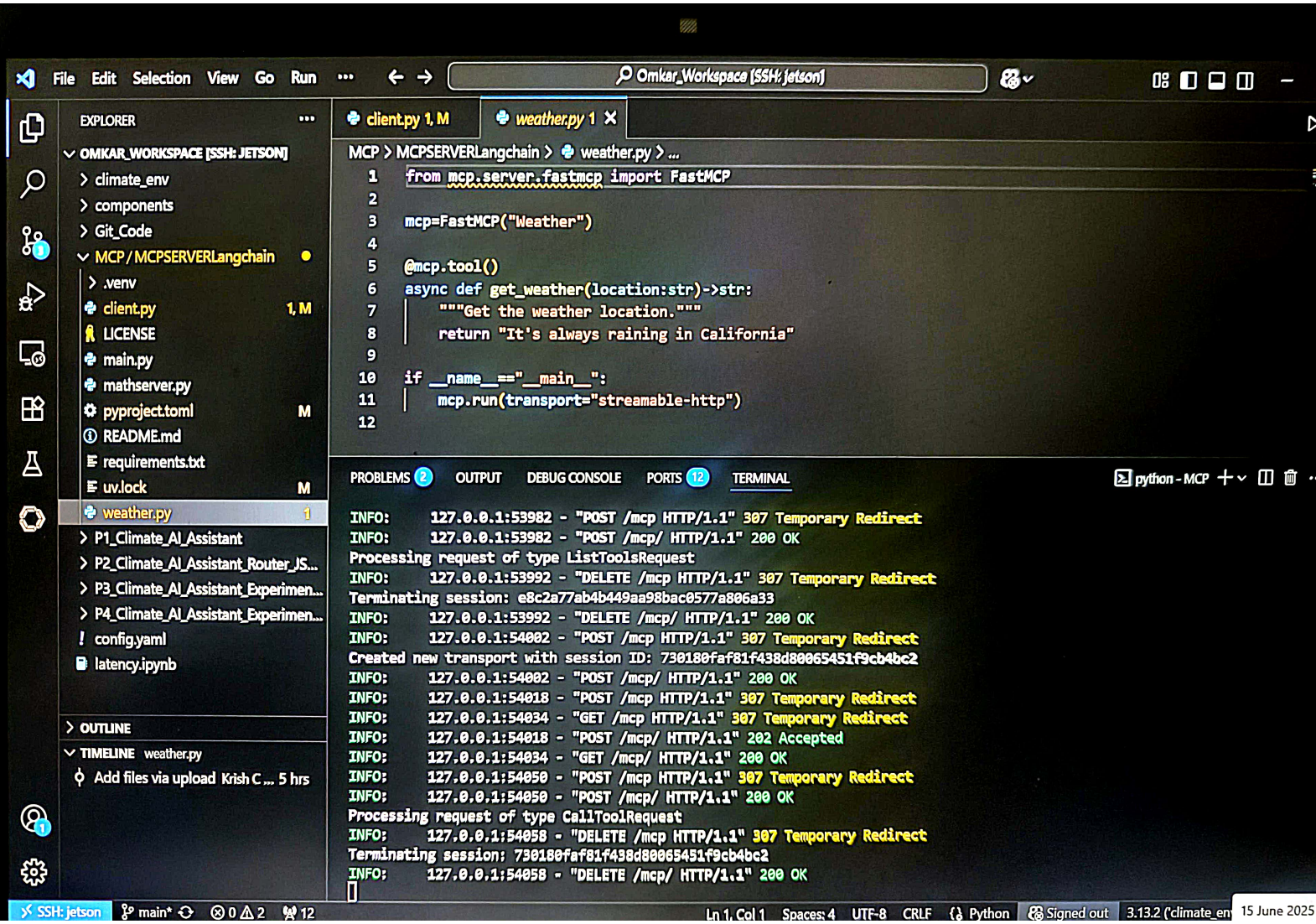
E)

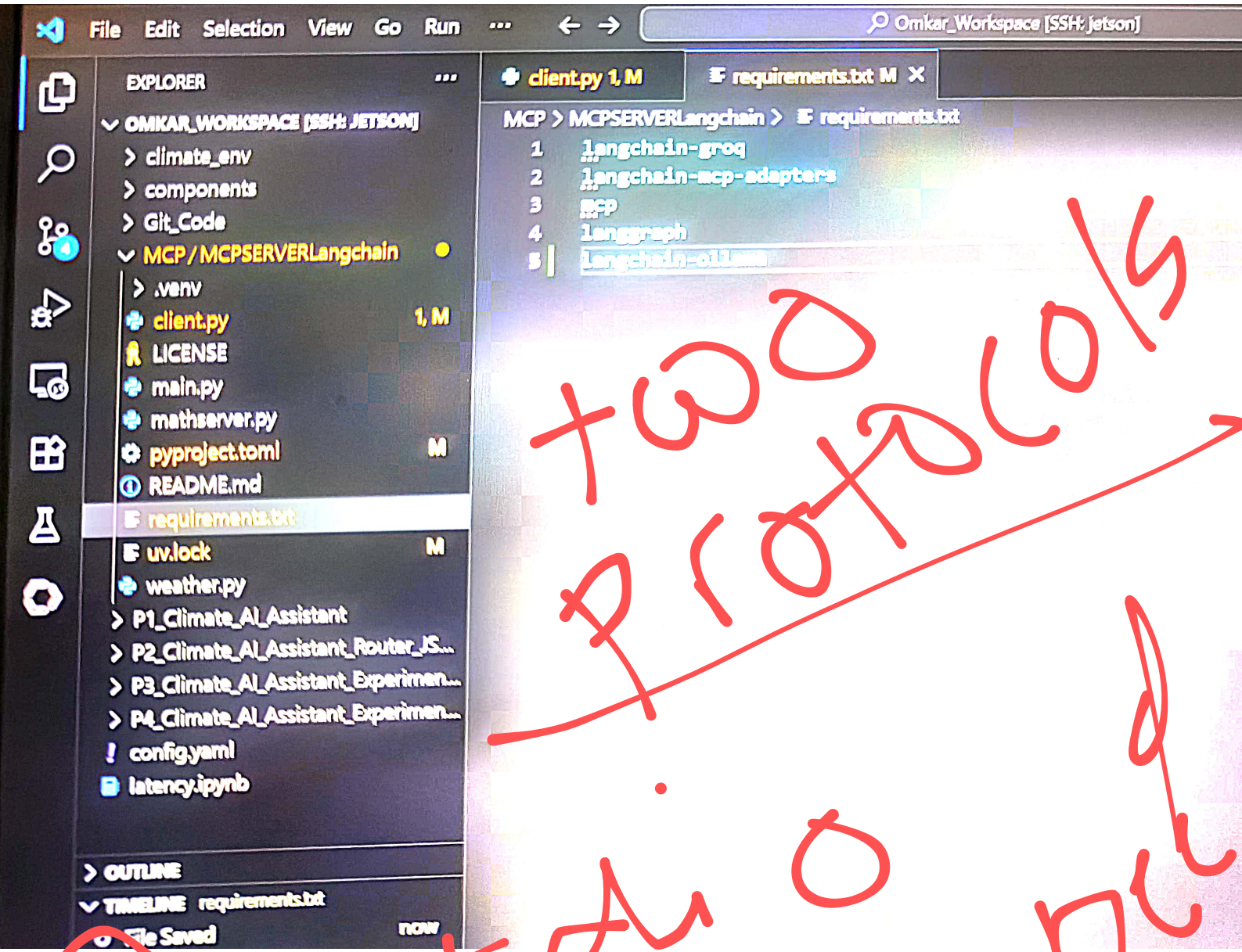
In case of stdio you can direct connect client to server but in case of HTTP you need to first run your server file then client file

Advantage & easy to onboard role on your server you don need to make more changes as compared to prisma code where you use binding now
→ easy to maintain tools, code.

There are 3 things comes in this
→ prompt :- require to guide model
→ Tools
→ resources & any additional information that model might need


File Edit Selection View Go Run ...
Omkar_Workspace [SSH: jetson]
EXPLORER
OMKAR_WORKSPACE ...
climate_env
components
Git_Code
MCP / MCPSERVERLangchain
.venv
client.py 1, M
LICENSE
main.py
mathserver.py 1
pyproject.toml M
README.md
requirements.txt
uv.lock M
weather.py
P1_Climate_AI_Assistant
P2_Climate_AI_Assistant_Router_JS...
P3_Climate_AI_Assistant_Experimen...
P4_Climate_AI_Assistant_Experimen...
config.yaml
latency.ipynb
OUTLINE
TIMELINE mathserver.py
Add files via upload Krish C ... 5 hrs
client.py 1, M
mathserver.py 1
MCP > MCPSERVERLangchain > mathserver.py > ...
1 from mcp.server.fastmcp import FastMCP
2
3 mcp=FastMCP("Math")
4
5 @mcp.tool()
6 def add(a:int,b:int)->int:
7 """_summary_
8 Add to numbers
9 """
10 return a+b
11
12 @mcp.tool()
13 def multiple(a:int,b:int)-> int:
14 """Multiply two numbers"""
15 return a*b
16
17 #The transport="stdio" argument tells the server to:
18
19 #Use standard input/output (stdin and stdout) to receive and respond to tool function calls.
20
21 if __name__=="__main__":
22 mcp.run(transport="stdio")


File Edit Selection View Go Run ···
Omkar_Workspace [SSH: jetson]
EXPLORER
OMKAR_WORKSPACE [SSH: JETSON]
climate_env
components
Git_Code
MCP / MCPSERVERLangchain
.venv
client.py 1, M
LICENSE
main.py
mathserver.py
pyproject.toml M
README.md
requirements.txt
uv.lock M
weather.py 1
P1_Climate_AI_Assistant
P2_Climate_AI_Assistant_Router_JS...
P3_Climate_AI_Assistant_Experimen...
P4_Climate_AI_Assistant_Experimen...
config.yaml
latency.ipynb
OUTLINE
TIMELINE weather.py
Add files via upload Krish C ... 5 hrs
client.py 1, M
weather.py 1
MCP > MCPSERVERLangchain > weather.py > ...
1 from mcp.server.fastmcp import FastMCP
2
3 mcp=FastMCP("Weather")
4
5 @mcp.tool()
6 async def get_weather(location:str)->str:
7 """Get the weather location."""
8 return "It's always raining in California"
9
10 if __name__=="__main__":
11 mcp.run(transport="streamable-http")
12
PROBLEMS 2 OUTPUT DEBUG CONSOLE PORTS 12 TERMINAL
python - MCP
INFO: 127.0.0.1:53982 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:53982 - "POST /mcp/ HTTP/1.1" 200 OK
Processing request of type ListToolsRequest
INFO: 127.0.0.1:53992 - "DELETE /mcp HTTP/1.1" 307 Temporary Redirect
Terminating session: e8c2a77ab4b449aa98bac0577a806a33
INFO: 127.0.0.1:53992 - "DELETE /mcp/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:54002 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
Created new transport with session ID: 730180faf81f438d80065451f9cb4bc2
INFO: 127.0.0.1:54002 - "POST /mcp/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:54018 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:54034 - "GET /mcp HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:54018 - "POST /mcp/ HTTP/1.1" 202 Accepted
INFO: 127.0.0.1:54034 - "GET /mcp/ HTTP/1.1" 200 OK
INFO: 127.0.0.1:54050 - "POST /mcp HTTP/1.1" 307 Temporary Redirect
INFO: 127.0.0.1:54050 - "POST /mcp/ HTTP/1.1" 200 OK
Processing request of type CallToolRequest
INFO: 127.0.0.1:54058 - "DELETE /mcp HTTP/1.1" 307 Temporary Redirect
Terminating session: 730180faf81f438d80065451f9cb4bc2
INFO: 127.0.0.1:54058 - "DELETE /mcp/ HTTP/1.1" 200 OK
SSH: jetson main* 0 2 12
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python Signed out 3.13.2 ('climate_en
15 June 2025



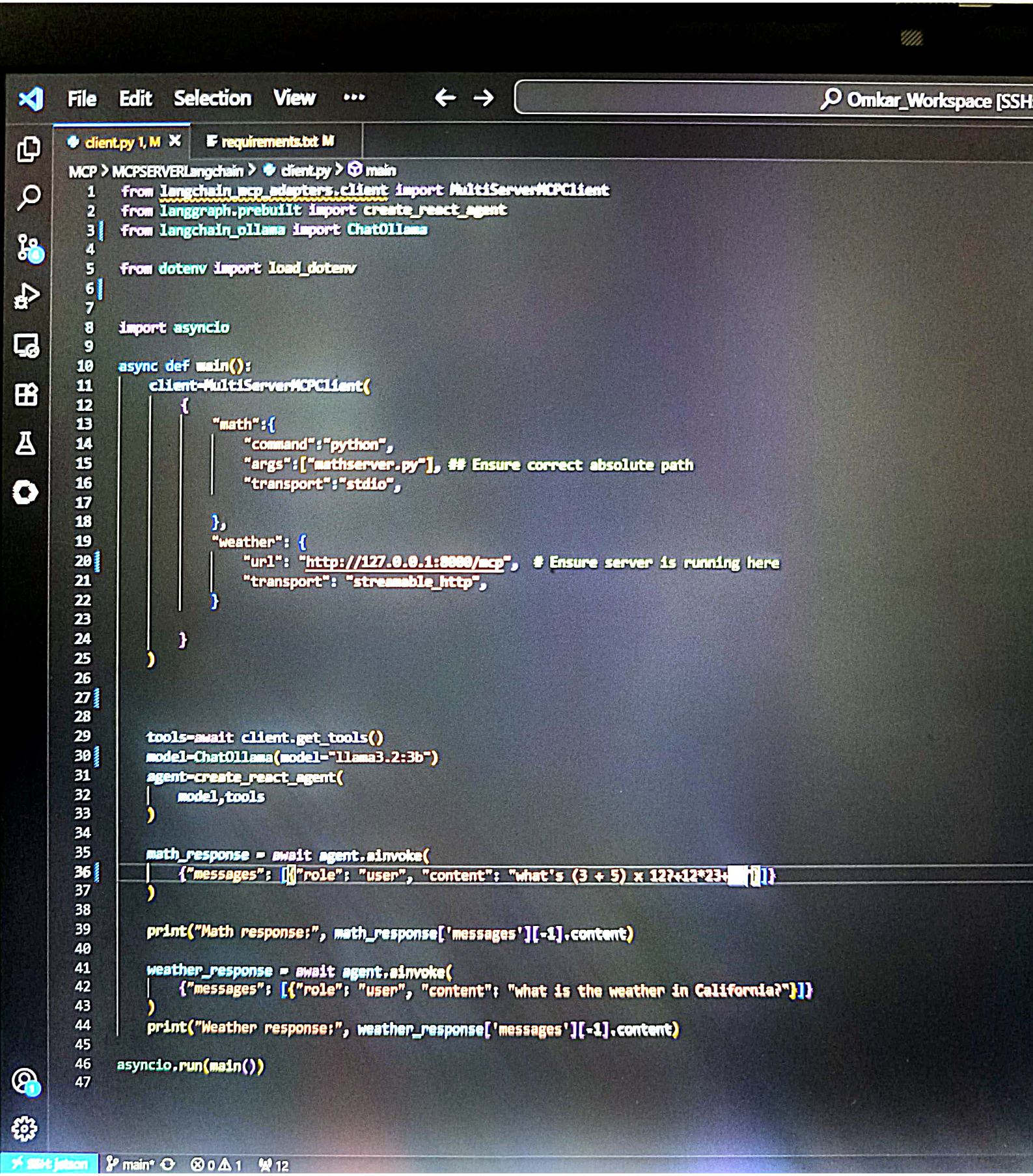
two protocols

① stdio

② HTTP streamable

to run file

to get url



File Edit Selection View ··· Omkar_Workspace [SSH

client.py 1, M | requirements.txt M

MCP > MCPSERVERLangchain > client.py > main

```python
from langchain_mcp_adapters.client import MultiServerMCPClient
from langgraph.prebuilt import create_react_agent
from langchain_ollama import ChatOllama

from dotenv import load_dotenv


import asyncio

async def main():
    client=MultiServerMCPClient(
        {
            "math":{
                "command":"python",
                "args":["mathserver.py"], ## Ensure correct absolute path
                "transport":"stdio",

            },
            "weather": {
                "url": "http://127.0.0.1:8000/mcp",  # Ensure server is running here
                "transport": "streamable_http",
            }

        }
    )



    tools=await client.get_tools()
    model=ChatOllama(model="llama3.2:3b")
    agent=create_react_agent(
        model,tools
    )

    math_response = await agent.ainvoke(
        {"messages": [{"role": "user", "content": "what's (3 + 5) x 12?+12*23+[redacted]"}]}
    )

    print("Math response:", math_response['messages'][-1].content)

    weather_response = await agent.ainvoke(
        {"messages": [{"role": "user", "content": "what is the weather in California?"}]}
    )
    print("Weather response:", weather_response['messages'][-1].content)

asyncio.run(main())
```

main* ⊗ 0 ⚠ 1 12