⭐AI-Driven Development 30-Day Challenge-Task 4📝

Marks: 10 | Deadline: 48 Hours

Class Slot: Friday — 6:00 PM to 9:00 PM

Instructor: Sir Hamzah Syed

1. What Are MCP Servers?

MCP (Model Context Protocol) servers act like a bridge between your AI model or your CLI and the tools you want it to use.
They give your model controlled access to things like:

- Files
- APIs
- Local functions
- External systems like Github, Firebase etc

In simple terms:

An MCP server gives Gemini CLI "tools" so it can actually *do things* instead of just replying with text.

---

2. Why MCP Servers Are Useful

- They let you add abilities to an AI model instantly.
- They follow a standard format, so you can plug them into different systems easily.

- They remove complexity — you don't have to manually wire every tool.
- They make your setup more modular and maintainable.
- Students don't need deep backend code; they just connect to a server.

---

3. The Problem

Gemini CLI **cannot create full agents by itself.**
It doesn't have strong agent-building support.
So creating complete agents directly inside Gemini CLI becomes frustrating and limited.

---

4. The Solution — Context7✨

There is a platform called **Context7**.

Link: https://context7.com

What Context7 Provides

Context7 is **one complete MCP server**.
It is not a collection of MCP servers — it is **one** MCP server that exposes powerful tools and documentation.

It includes:

- Documentation for Python
- Documentation for OpenAgents SDK
- Documentation for Supabase
- Documentation for FastAPI
- Documentation for all modern frameworks
- Auto-updating documentation

(So if OpenAgents SDK updates → Context7 updates too.)

Why This Is Perfect

Because when you ask Gemini CLI to build an agent using the OpenAgents SDK:

- It will not produce errors
- It will follow the correct documentation
- It will understand the updated workflow
- Students don't have to keep checking new docs
- The whole system stays fresh and compatible

This solves the frustration of Gemini CLI not knowing how to build agents.

---

5. Task 4 — Connecting Context7 MCP Server to Gemini CLI

For today's task, you will connect the **Context7 MCP server** to **your Gemini CLI**.

You can find the full step-by-step instructions here:

Guide Link:

https://www.notion.so/Personalization-Chatbot-with-Chainlit-2b2644e5197680728913dc57ee7df803

This guide explains:

- How to add the MCP server of Context7 with Gemini CLI

(Connect the MCP Server first before proceeding to a practical task!)

(We did not rewrite those steps here. Students will follow that link.)

---

6. Practical Task — Build the Study Notes Summarizer & Quiz Generator Agent

After Context7 is connected, you will create an agent using:

- **OpenAgents SDK**
- **Streamlit** (recommended for UI, but HTML/CSS is allowed your choice)
- **PyPDF** (for PDF text extraction)
- **Gemini CLI**
- **Context7 MCP** (tool provider)

What the Agent Will Do

A. PDF Summarizer

- User uploads a PDF.
- Text is extracted using PyPDF.
- Agent generates a clean, meaningful summary.
- Summary can appear in any UI style students choose (card, block, container, etc.).

B. Quiz Generator

- After summarization, the user can click **Create Quiz**.
- The agent reads the **original PDF** (not the summary).
- It generates:
  - MCQs
  - Or mixed-style quizzes

This is the core required functionality but students can add more features if they want, but these two are the minimum.

After completing the task, update it in the github and also provide a screenshot of the prompt given to the gemini cli while creating the pdf summarizer and quiz agent in the same github file too!

7. Important Links

- **Context7 Website:**
  https://context7.com

- **MCP Server Setup Guide:**

  https://www.notion.so/Personalization-Chatbot-with-Chainlit-2b2644e5197680728913dc57ee7df803

Prepared by Asma Yaseen
Supervised by Sir Hamzah Syed

✨ *Good luck*✨