TASK-04

NAME : Ayaan Ahmed Khan

SLOT : FRIDAY (6:00 TO 9:00)

INSTRUCTOR : SIR HAMZA SYED

AI-DRIVEN DEVELOPMENT — 30 DAYS CHALLENGE

1. What Are MCP Servers?

MCP (Model Context Protocol) servers act as a bridge between your AI model and external tools.

They allow your AI or CLI to safely access:

• Files

• APIs

• Local functions

• Databases

• External systems (GitHub, Firebase, Supabase, etc.)

In simple terms:

An MCP server gives tools to your AI model so it can DO tasks instead of only talking.

2. Why MCP Servers Are Useful?

• Instantly add new abilities to an AI model

• Works with a standard format

• No need to manually connect every tool

• System becomes modular and easy to maintain

• Students don’t need deep backend coding

• Connecting the server makes the AI model much more powerful

3. The Problem

Gemini CLI cannot build full agents itself.

It lacks strong agent-building support.

This causes:

• Errors

• Incorrect SDK usage

• Outdated workflows

• Limited capabilities

4. The Solution — Context7

Context7 is a complete MCP server providing:

• Python documentation

• OpenAgents SDK documentation

• Supabase docs

• FastAPI docs

• Modern frameworks documentation

• Auto-updating docs

This ensures Gemini CLI always works with the latest and correct documentation.

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

Students must follow the official MCP server setup steps to connect Context7 before starting the practical task.

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

Project folder structure:

/study-notes-agent

■■■ app.py

■■■ agent.py

■■■ pdf_utils.py

■■■ requirements.txt

■■■ README.md

■■■ screenshots/

■ ■■■ gemini_prompt.png

requirements.txt:

streamlit

pypdf

openagents

google-generativeai

pdf_utils.py:

```
from pypdf import PdfReader

def extract_pdf_text(pdf_file):

reader = PdfReader(pdf_file)

text = ""

for page in reader.pages:

text += page.extract_text() + "\n"

return text
```

agent.py:

```
from openagents import Agent

import google.generativeai as genai

genai.configure(api_key="YOUR_API_KEY")

agent = Agent(

name="pdf_study_agent",

instructions="""

You are a Study Notes Summarizer & Quiz Generator Agent.

1. Extract PDF content

2. Generate structured summary

3. Create MCQs and quizzes

""",

model="gemini-1.5-flash"

)

def generate_summary(text):

prompt = f"Summarize this PDF text:\n{text[:15000]}"

return agent.run(prompt)
```

def generate_quiz(text):

prompt = f"Generate MCQs and short questions:\n{text[:15000]}"

return agent.run(prompt)

app.py (Streamlit UI):

• Upload PDF

• Generate Summary

• Generate Quiz

Prepared by: Asma Yaseen

Supervised by: Sir Hamzah Syed