

AIDD 30 Days Challenge Task: 4

MCP Server:

MCP (Model Context Protocol) is a tool which we will be using to give our CLI Agent more context (information/data).

A practical example, Gemini CLI can't build agents so, for solving this problem we will be providing the OpenAI Agent SDK docs to Gemini by connecting MCP Server with our CLI. It basically provides tools to our CLI.

Advantages:

- Using MCP Server we can make our CLI more efficient because it provides the data with the CLI don't have as a built-in like OpenAI Agents SDK.
- It can be added through few easy steps (We'll see below).
- There are multiple tools that we can get through MCP Server and we can make our development smooth.
- It helps the beginner that they don't have to write a complete backend for this they can just connect the MCP Server.

Context 7:

To solve the above mentioned problems we have a tool **Context 7**.

Context 7 has more than **54,000+ tools** which we can connect with our CLI Agent.

Its a single MCP Server contains docs of Python, OpenAI Agents SDK, Vercel/Nextjs and many more. And the best part is that the docs also auto-updates.

The screenshot shows the Context 7 MCP Server dashboard. At the top, there's a navigation bar with a logo, the text "Context7", and links for "Personal" (with a dropdown arrow), "Dashboard", "Plans", "Learn", "Try Live", "Install", and a green button "+ Add Docs". Below the navigation, the text "Up-to-date Docs for LLMs and AI code editors" is displayed in green. A sub-section below it says "Copy latest docs & code — paste into Cursor, Claude, or other LLMs". At the bottom, there are two input fields: "Search a library (e.g. Next, React)" and "Chat with Docs", separated by the word "or".

Task: 4

Connecting Context7 MCP Server to Gemini CLI

I have successfully connected the MCP Server with my Gemini CLI.

```
> /mcp

Configured MCP servers:

[+] context7 - Ready (2 tools)
Tools:
- get-library-docs
- resolve-library-id

Using: 1 MCP server

> [ ] Type your message or @path/to/file

F:\AIDD\project-1
```

Practical Task - PDF Summarizer & Quiz Generator

Tech Stack:

- OpenAI Agents SDK
- Streamlit
- PyPDF
- Gemini CLI
- Context 7 (MCP Server / Tool Provider)

App Link: <https://pdf-summarizer-quiz-generator-task.streamlit.app>

How it works:

- Open the Link:

PDF Summarizer + Quiz Generator (Gemini Version)

Upload a PDF



Drag and drop file here

Limit 200MB per file • PDF

[Browse files](#)

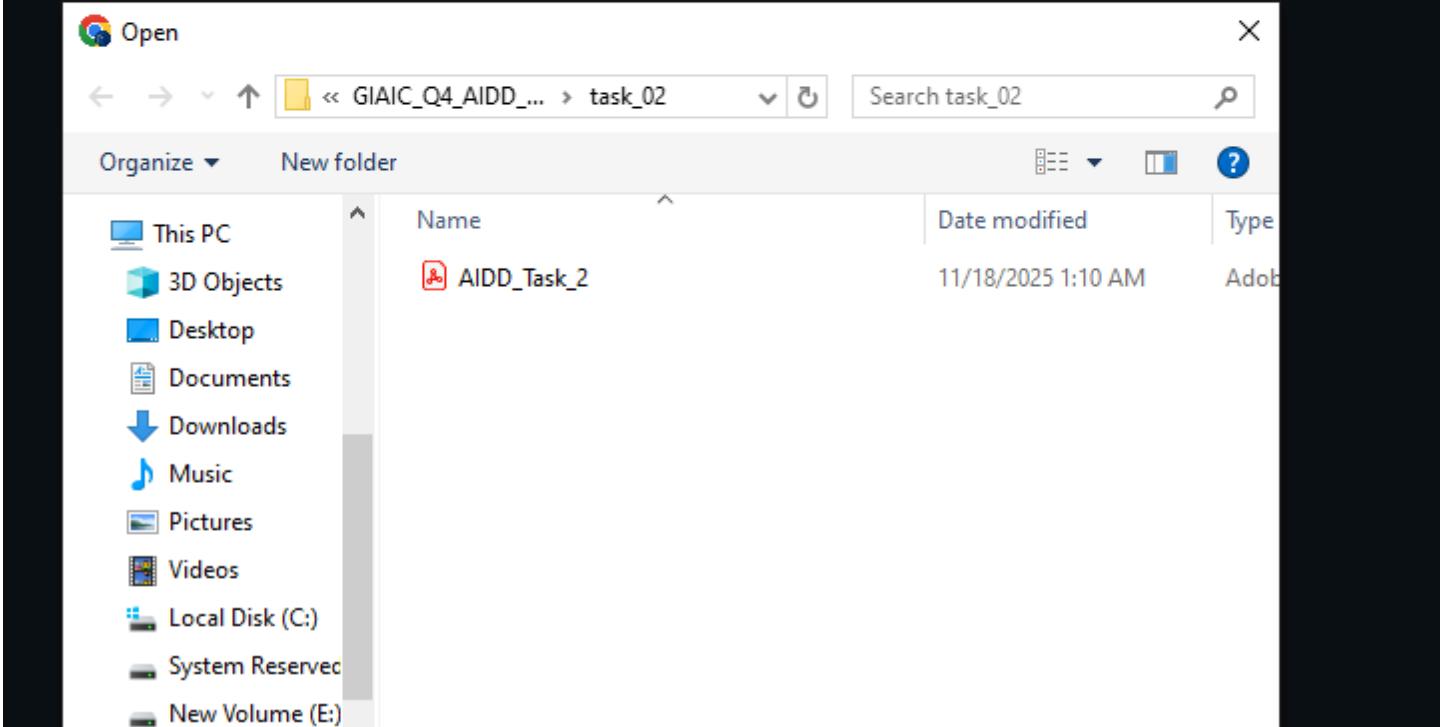
- Add any PDF file from your device:

PDF Summarizer + Quiz Generator (Gemini Version)

Upload a PDF

Drag and drop file here
Limit 200MB per file • PDF

[Browse files](#)



PDF Summarizer + Quiz Generator (Gemini Version)

Upload a PDF



Drag and drop file here

Limit 200MB per file • PDF

[Browse files](#)



AIDD_Task_2.pdf 77.8KB



PDF loaded successfully!

[Generate Summary](#)

- Now click on Generate Summary:

Summary

The provided document, "AIDD 30 Days Challenge Task: 2," explores key concepts in AI-Driven Development (AIDD) through theoretical questions, a practical AI CLI task, and multiple-choice questions.

AI Agents and System Architecture: Using AI Development Agents, such as Gemini CLI, for repetitive setup tasks is crucial for a system architect's growth. This approach frees developers from mundane coding, allowing them to focus on high-level architectural design, feature development, scalability, and strategic thinking, fostering a collaborative partnership with AI.

The Nine Pillars and M-Shaped Developers: The Nine Pillars of AIDD act as a blueprint for building advanced solutions with AI. Key pillars include Specification-Driven Development (SDD) for precise product specifications, MCP Standard for AI comprehension, AI CLI Coding Agents for code generation, and AI First IDEs for automated file and folder management. This methodology enables developers to become "M-Shaped," possessing deep skills across various domains like frontend, backend, and design, rather than being limited to a single area.

Vibe Coding vs. Specification-Driven Development (SDD): Vibe coding, often an unstructured approach to using AI for code generation, typically leads to issues within a week. Developers quickly lose understanding of the AI-generated code, making debugging and maintenance difficult. In contrast, SDD involves providing AI with a comprehensive, "syllabus-like" guide detailing the final product's requirements. This ensures the AI produces well-understood, maintainable code that aligns with clear specifications.

Architecture-First and Layered Thinking in AIDD: In AIDD, architecture-first thinking transforms the developer's role from a manual coder to a system designer. Developers prioritize creating complete blueprints and directing AI agents to execute tasks, including instant bug resolution. This shift allows for more strategic and high-level engagement. Furthermore, thinking in layers and systems (e.g., specifications, logic, AI coding, testing) is vital for optimal product development. This approach defines clear responsibilities, reduces risks, accelerates development, and enhances overall product quality.

Practical Task: Email Validation Specification: For the practical task, an AI CLI tool was used to generate a one-paragraph specification for an email validation function. The prompt required checks for the "@" symbol, validation of common domain extensions (e.g., .com, .net), and the return of clear, human-readable error messages for specific validation failures. The Gemini CLI successfully produced a technical yet readable specification that met all specified criteria.

Conclusion: The assignment concluded with multiple-choice questions reinforcing the theoretical concepts. Completing the task significantly enhanced the participant's understanding of AIDD principles, including the Nine Pillars, the characteristics of an M-Shaped Developer, and the critical importance of Specification-Driven Development over unstructured vibe coding.

[Create Quiz](#)

- After the summary is generated click on the Create Quiz button:

Quiz

Here is a 10-question MCQ quiz based on the provided PDF text:

Question 1: How does using AI Development Agents, like Gemini CLI, for repetitive setup tasks primarily contribute to a system architect's growth? A. By allowing the architect to focus on more manual coding tasks. B. By eliminating the need for any architectural design. C. By freeing the architect's brain to think at a high level about architecture, features, and scalability. D. By making the architect completely dependent on the AI for all strategic decisions. **Correct Answer: C**

Question 2: According to the text, what do the Nine Pillars of AIDD fundamentally represent for a developer? A. A set of coding standards for basic tasks. B. A blueprint or key concepts for building advanced projects with an AI Agent. C. A method for completely replacing human developers with AI. D. A historical overview of AI development. **Correct Answer: B**

Question 3: In Spec-Driven Development (SDD), what is the main purpose of creating complete specifications for a product? A. To speed up the debugging process. B. To provide the AI Agent with a complete work plan and understanding of the product. C. To reduce the amount of documentation required later. D. To allow for flexible coding without strict guidelines. **Correct Answer: B**

Question 4: What is the primary role of the AI CLI Coding Agent within the Nine Pillars framework, after understanding specifications? A. To generate new specifications for the product. B. To provide financial estimates for the project. C. To translate the understood specifications into actual code and create documentation. D. To manage client communication and feedback. **Correct Answer: C**

Question 5: Which characteristic defines AI First IDEs, as described in the text, to assist developers? A. They strictly prohibit AI agents from coding within them. B. They automate the creation of files and management of folder structures. C. They are exclusively used for running compiled code. D. They require developers to manually configure every setting. **Correct Answer: B**

Question 6: How is an M-Shaped Developer characterized in the context of AIDD? A. As someone with limited knowledge in only one specialized domain. B. As a developer with deep skills across multiple related domains (e.g., frontend, backend, design) who works with an AI partner. C. As an individual who only works on design aspects of a project. D. As a developer who avoids using any AI tools or partners.

Here is how the app works.

By the End of this Assignment:

By the end of this assignment I really learned how to use CLI for development. The correct way of writing Specification. It was a very good assignment did a hands-on practice of AIDD.