We are creating a Agent using gemini cli that does the following:

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

first we are gonna make a account on Context7 then generate a API key for your project then we wil create **.gemini** folder at the root of our project then copy the setting.json file from the **global .gemini** folder and paste it in the **project .gemini** folder next we will paste the json for our Context7 mcp server which is provided by Context7
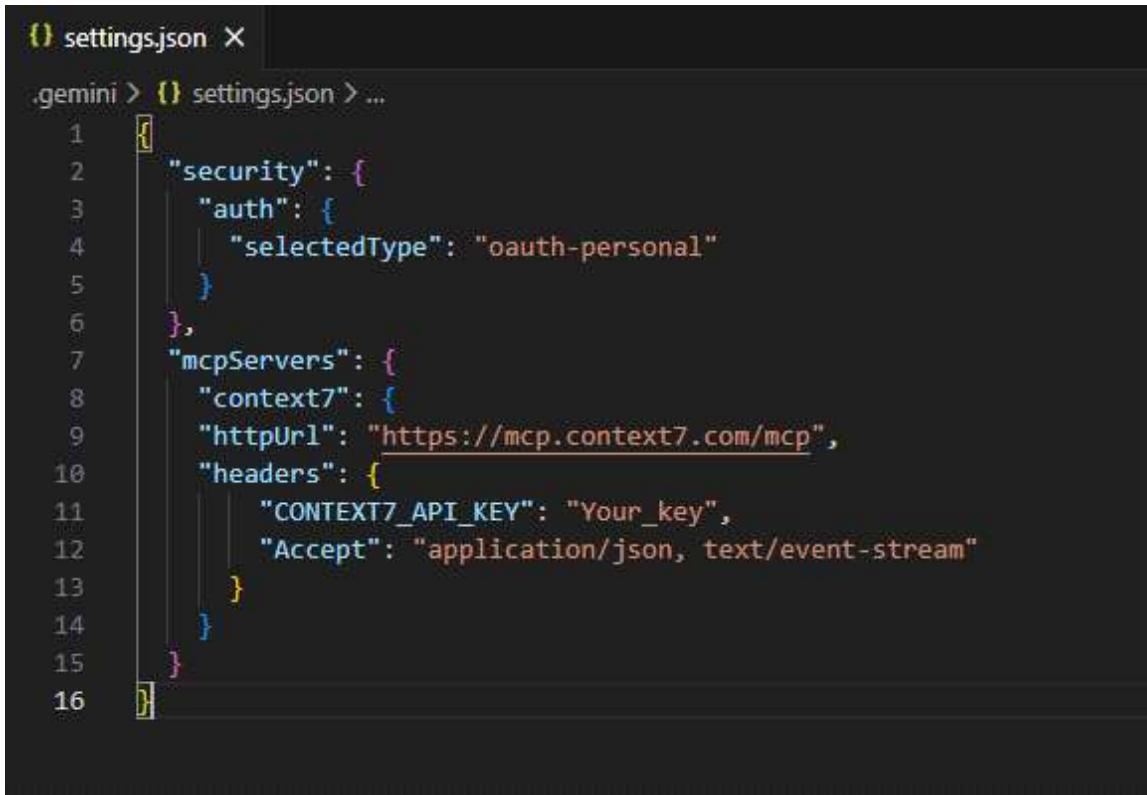
G Gemini CLI

Open `~/.gemini/settings.json` and add:

Remote Server:

```json
{
  "mcpServers": {
    "context7": {
      "httpUrl": "https://mcp.context7.com/mcp",
      "headers": {
        "CONTEXT7_API_KEY": "YOUR_API_KEY",
        "Accept": "application/json, text/event-stream"
      }
    }
  }
}
```

after adding json correctly your file your file will look like this

```json
{} settings.json X

.gemini > {} settings.json > ...
  1   {
  2       "security": {
  3         "auth": {
  4           "selectedType": "oauth-personal"
  5         }
  6       },
  7       "mcpServers": {
  8         "context7": {
  9         "httpUrl": "https://mcp.context7.com/mcp",
 10         "headers": {
 11             "CONTEXT7_API_KEY": "Your_key",
 12             "Accept": "application/json, text/event-stream"
 13           }
 14         }
 15       }
 16   }
```
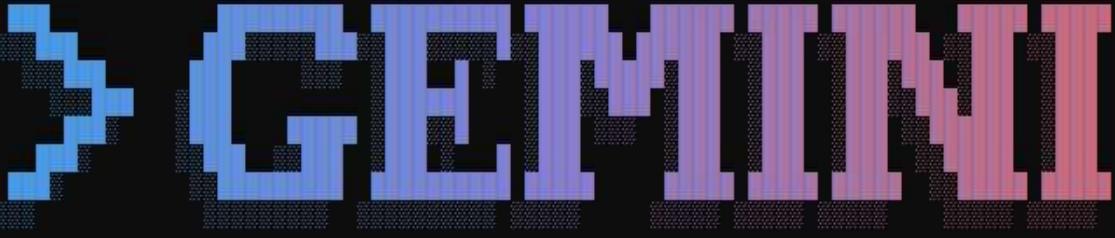
replace the "Your_key" to your actual generated Context7 API key

then we will open the gemini cli in the exact project folder

change the model to 3 pro if available by **/model** command

then we will run the "**/mcp list"** command to check if mcp has set up successfully

we can see that gemini has successfully recognized Context7 mcp server

now we will create a GEMINI.md file in the root of our project folder

in the file we will describe the structure and initial prompt of our project

you can describe what type of project you are trying to build in it

I have described the project structure of our project in GEMINI.md file

next i will give gemini this starting prompt

"Please read the entire project specification contained in the file named **GEMINI.md located in the current directory, and begin generating the complete `PDF_Study_Agent` project code based on those detailed instructions."**

it will give gemini signal to read the GEMINI.md file of our project instead of

taking prompts from command line and starts building it from the GEMINI.md

this is the first question

```
> Please read the entire project specification contained in the file named **GEMINI.md** located in the current directory, and begin generating the complete
  `PDF_Study_Agent` project code based on those detailed instructions.

  ☰  ReadFile GEMINI.md

  ?  WriteFile Writing to requirements.txt                                                                                                              ←

  1 streamlit
  2 pypdf
  3 openai

  Apply this change?

  ● 1. Yes, allow once
    2. Yes, allow always
    3. Modify with external editor
    4. No, suggest changes (esc)

☰ Waiting for user confirmation...
```

it created a requirements.txt file with all the packages that are going to be installed

it did a mistake not writing the correct package name

i will fix it to openai-agents and then we will initialize UV in the project

after initializing UV and activating virtual environment in the project

i will run the following command uv add -r requirements.txt

which will install all the packages in the UV .venv folder

first it created the app without even using the mcp then i specifically told him to use context7 mcp then after some tweaks and changes it successfully created the web app

while giving the starting prompt not the pre-written prompt tell the gemini cli to use the mcp server which is Context7 for our case so it can actually use the mcp server

After hitting on some errors it successfully created the web app