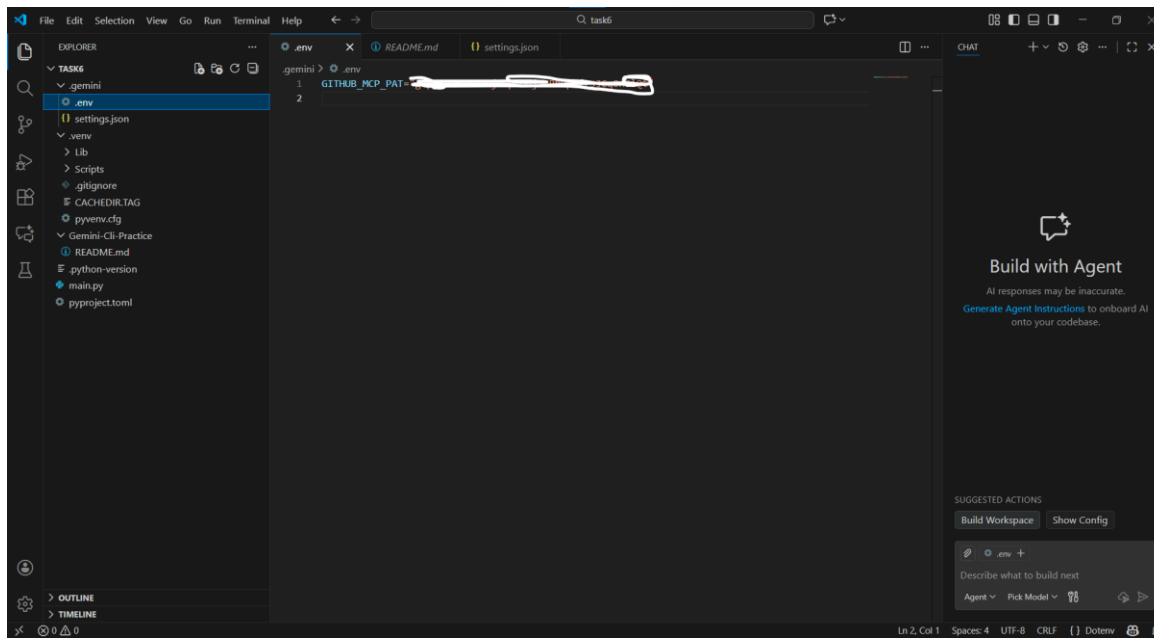


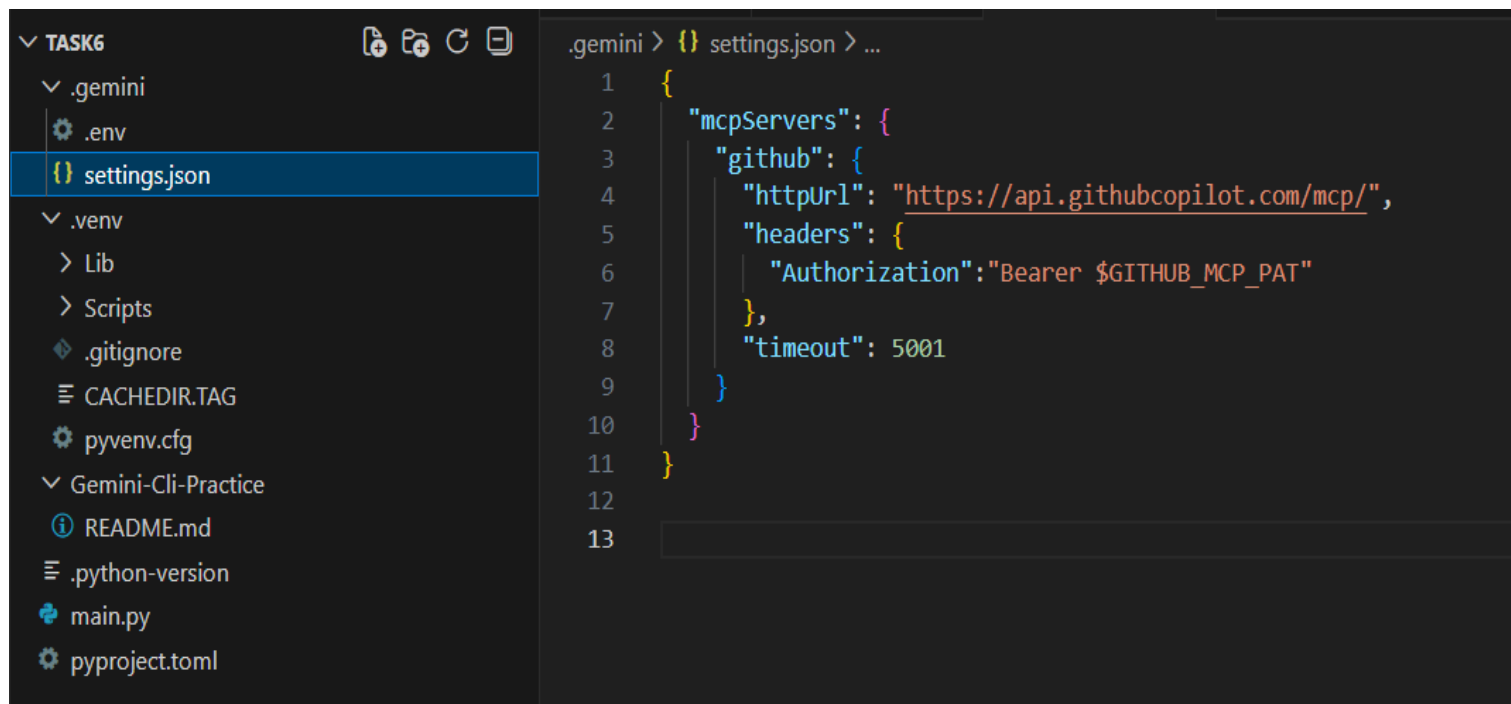
TASK-6

GITHUB MCP INTEGRATION

1. CREATE PERSONAL ACCESS TOKEN (PAT) THROUGH GITHUB AND SAVE IN .env file



2. CONFIGURE GEMINI TO GITHUB MCP SERVER



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a file named `settings.json` selected. The code editor shows the content of `settings.json`, which is a JSON object with a `mcpServers` property containing a `github` server configuration.

```
.gemini > {} settings.json > ...
1  {
2    "mcpServers": {
3      "github": {
4        "httpUrl": "https://api.githubcopilot.com/mcp/",
5        "headers": {
6          "Authorization": "Bearer $GITHUB_MCP_PAT"
7        },
8        "timeout": 5001
9      }
10   }
11 }
12
13
```

3. MCP SERVER CONNECTED WITH GITHUB SUCCESSFULLY.

```
C:\> Gemini - task6
3. Create GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> /mcp list

Configured MCP servers:

[ ] github - Ready (40 tools, 2 prompts)
Tools:
- add_comment_to_pending_review
- add_issue_comment
- assign_copilot_to_issue
- create_branch
- create_or_update_file
- create_pull_request
- create_repository
- delete_file
- fork_repository
- get_commit
- get_file_contents
- get_label
- get_latest_release
- get_me
- get_release_by_tag
- get_tag
- get_team_members
- get_teams
- issue_read
- issue_write
- list_branches
```

- issue_read
- issue_write
- list_branches
- list_commits
- list_issue_types
- list_issues
- list_pull_requests
- list_releases
- list_tags
- merge_pull_request
- pull_request_read
- pull_request_review_write
- push_files
- request_copilot_review
- search_code
- search_issues
- search_pull_requests
- search_repositories
- search_users
- sub_issue_write
- update_pull_request
- update_pull_request_branch

Prompts:

- AssignCodingAgent
- IssueToFixWorkflow

Using: 1 MCP server

GEMINI CREATED FOLDER AND MOVE IT INTO GITHUB REPOSITORY

```
om
d
Directory: C:\Users\kk\Desktop\Darakshan\quater4\task6

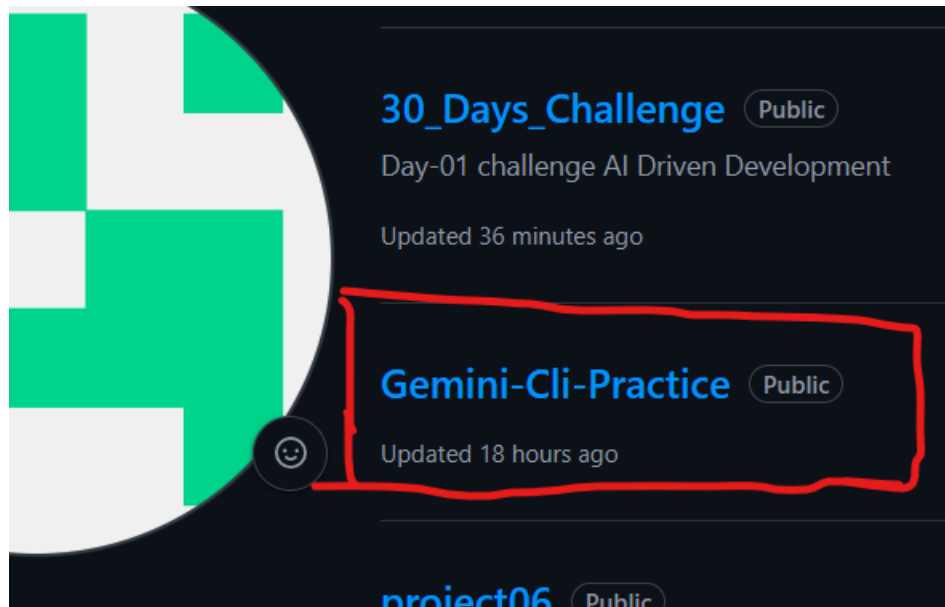
Mode                LastWriteTime         Length Name
----                -
d-----          27-Nov-2025   4:04 PM             Gemini-Cli-Practice

Shell cd Gemini-Cli-Practice [current working directory C:\Users\kk\Desktop\Darakshan\quater4\task6]

I've created the Gemini-Cli-Practice folder and moved into it. What's next?

Using: 1 MCP server

> █ Type your message or @path/to/file
```



```
21 ## About
22 This project aims to streamline the process of interacting with and extracting information from Markdown-based
  documentation using a powerful MCP server backend and the intelligent capabilities of Google's Gemini models.
  It's built with extensibility and performance in mind, offering a foundational component for AI-assisted
  development workflows.
23
24 ## Resources
25 - [GitHub Repository (Placeholder)](https://github.com/owner/repo) - Replace with actual repository link
26 - [Gemini API Documentation (External)](https://ai.google.dev/docs)
27 - [Model Context Protocol (MCP) Specification (External)](https://github.com/google/model-context-protocol)
28
29 ## GitHub Metrics (Placeholder)
30 ![GitHub Stars](https://img.shields.io/github/stars/owner/repo.svg?style=social)
31 ![GitHub Watchers](https://img.shields.io/github/watchers/owner/repo.svg?style=social)
32 ![GitHub Forks](https://img.shields.io/github/forks/owner/repo.svg?style=social)
33 ![GitHub Releases](https://img.shields.io/github/downloads/owner/repo/total.svg)
34 ![GitHub Packages](https://img.shields.io/github/packages/owner/repo/total.svg)
35
36 ## Footer
37 © 2025 Your Name or Organization. All rights reserved.
38 Generated by Gemini CLI.
```

README.md is done. Task complete.

📁 ReadFolder .

Listed 7 item(s).

The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The main window displays a file named 'README.md' within a project called 'Gemini-CLI-Practice'. The file content is as follows:

```
1 # GitHub MCP Server and Gemini CLI
2
3 ## Introduction
4 This project focuses on providing a robust Model Context Protocol (MCP) server that
5 integrates seamlessly with the Gemini CLI. It processes Markdown content, parsing and
6 transforming it, centralizing the interface for Markdown file handling. The core
7 functionality involves ingesting Markdown files and preparing their content for further
8 manipulation or conversion, ultimately enabling advanced interactions through the Gemini
9 CLI. This solution bridges the gap between structured documentation and AI-driven
10 processing, offering a powerful tool for developers and content creators.
11
12 ## Features
13 - **Markdown Processing:** Efficiently parses and transforms Markdown content into various
14   formats, including HTML.
15 - **Configurable Markdown Rendering:** Supports basic and advanced Markdown rendering
16   using libraries like `marked` and `Prism` for syntax highlighting.
17 - **Content Extraction:** Extracts key information from Markdown such as titles,
18   descriptions, example snippets, and schema arguments.
19 - **MCP Server Integration:** Acts as an MCP server, exposing tools for interaction.
20 - **Gemini Model Integration:** Leverages Gemini models for advanced AI capabilities,
21   including content generation and understanding.
22 - **Schema Conversion:** Converts OpenAPI schemas to Gemini Schema objects for consistent
23   tool definitions.
24 - **README Content Management:** Fetches and parses `README.md` files from GitHub
25   repositories, extracting structured data.
26
27 ## Gemini CLI Integration
28 The project is designed to be a backend service for the Gemini CLI, enabling it to interact
29 with and process various types of content, especially Markdown documentation.
30 - **Model Interaction:** Utilizes `Gemini` and `GeminiModel` classes to interact with
31   different Gemini models (`gemini-1.5-flash`, `gemini-2.0-flash-001`, etc.) for content
32   generation and understanding.
33 - **Tool Definition:** Translates tool schemas into Gemini-compatible formats using
34   `gemini_to_json_schema` for effective tool calling within the Gemini ecosystem.
35 - **Asynchronous Operations:** Supports asynchronous content generation and real-time
36   streaming with Gemini models.
37
38 ## About
```

The left sidebar shows the Explorer view with the project structure: 'Gemini-CLI-Practice' (containing '.env', 'settings.json', '.venv', 'python-version', 'main.py', 'pyproject.toml') and 'OUTLINE'/'TIMELINE' views. The bottom status bar indicates 'Ln 5, Col 1', 'Spaces: 4', 'UTF-8', 'LF', and 'Markdown' mode. The system tray at the bottom shows the date and time as '4:17 PM'.