

**TASK 06**  
**30 DAYS CHALLENGE**  
**FRIDAY BATCH (6:00 PM to 9:00 PM)**

**Submitted by:**

**Darakhshan Imran**

**Roll No. 00430657**

## GITHUB – MCP INTEGRATION

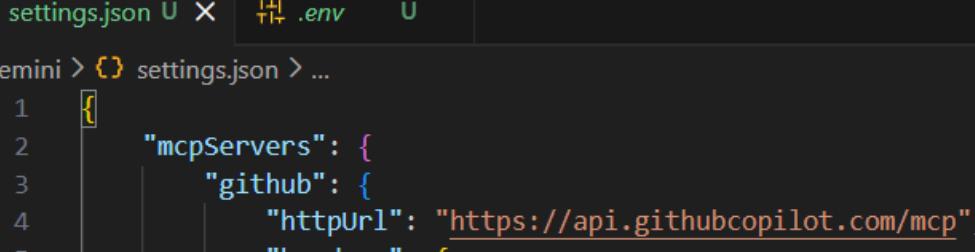
### Step 1 - Create Your GitHub Personal Access Token (PAT)

The screenshot shows the GitHub Developer Settings page under Personal access tokens (classic). A message at the top encourages users to copy their token as it won't be shown again. Below is a list of tokens, with one entry highlighted in green. A tooltip explains that personal access tokens function like OAuth tokens for Git over HTTPS or API authentication. The bottom of the page includes standard GitHub footer links.

### Step 2 - Store Your Token Securely

The screenshot shows a code editor with a .env file open. The file contains a single line: `GITHUB_MCP_PAT = [REDACTED]`. The code editor interface includes tabs for settings.json and .env, and a terminal tab labeled .gemini > .env.

## Step 3 - Configure Gemini to Use GitHub MCP Server



```
settings.json .env

.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          }
9      }
10 }
```

#### **Step 4 – Verify the Connection setup via /mcp list**

```
Gemini - Task 6

> /mcp list

Configured MCP servers:

  [github] 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
  Prompts:
    - AssignCodingAgent
    - IssueToFixWorkflow

  - get_teams
  - 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
```

## Step 5 – Server Runs Successfully

```
> list my github repositories

? get_me (github MCP Server) {}

MCP Server: github
Tool: get_me

Allow execution of MCP tool "get_me" from server "github"?

● 1. Yes, allow once
   2. Yes, always allow tool "get_me" from server "github"
   3. Yes, always allow all tools from server "github"
   4. No, suggest changes (esc)
```

⌚ Waiting for user confirmation...

⌚ The following are your GitHub repositories:

- cli-number-guessing-game
- milestone5-shareable-pdf-resume-darakhshan.imran
- first-aid-game
- milestone1-static-resume-darakhshan.imran
- agent-ai-learning
- car-dynamic-frontend
- furniro
- myportfolio
- assignments-04
- ecommerce7-hackathon-2
- OOP\_01
- milestone2-3-dynamic-resume-builder-darakhshan.imran
- food-riddle-game
- assignment6-oop
- template-7-api
- fitness-tracker
- streamlit-lms
- milestone-4-editable-resume-darakhshan-imran
- password-strength-meter
- cli-library-management-system
- Login-Form
- Hackathon-3
- smart-unit-converter
- Registration-Form
- Asynchronous
- lms-streamlit-app
- AIDD-30-DAYS-CHALLENGE
- class-15-code
- Forms-navigation-feature

Let me know if you need to perform any actions on these repositories.