Mawlana Bhashani Science and Technology University

Santosh, Tangail - 1902

Department of Information and Communication Technology

Lab Report

Lab Report No: 06

Lab Report On: Setup and Practical Implementation of GitHub Copilot and Gemini CLI in Personal Portfolio Project.

Course Title: Internet & Web Programming Lab

Course Code: ICT-4108

Submitted by: Name: Shusmita Ghosh Shoili ID: IT-21006 Session: 2020-21 Year: 4th Semester: 1st Dept of ICT, MBSTU	**Submitted to:** Md. Anowar Kabir Lecturer Dept. of ICT MBSTU

Date of Performance: 09/07/2025 Date of Submission: 31/07/2025

Lab Report Title : Setup and Practical Implementation of GitHub Copilot and Gemini CLI in your Personal Portfolio Project.

Introduction:

With the rapid advancement of artificial intelligence in software development, AI-assisted coding tools have become increasingly valuable in enhancing productivity and code quality. GitHub Copilot and Gemini CLI represent two cutting-edge technologies that provide intelligent code suggestions and interactive development capabilities. This lab focuses on the setup and practical implementation of GitHub Copilot and Gemini CLI within the context of developing a personal portfolio project. The primary objective is to examine how these tools can streamline the coding process, facilitate efficient HTML and CSS development, and support real-time interaction through Visual Studio Code.

Objective:

- To set up and configure GitHub Copilot and Gemini CLI within the Visual Studio Code environment.

- To understand the core functionalities of AI-assisted coding tools in real-time code generation and editing.

- To develop and enhance a personal portfolio website using HTML and CSS with the assistance of GitHub Copilot and Gemini CLI.

- To evaluate the effectiveness of these tools in improving development workflow and reducing manual coding effort.

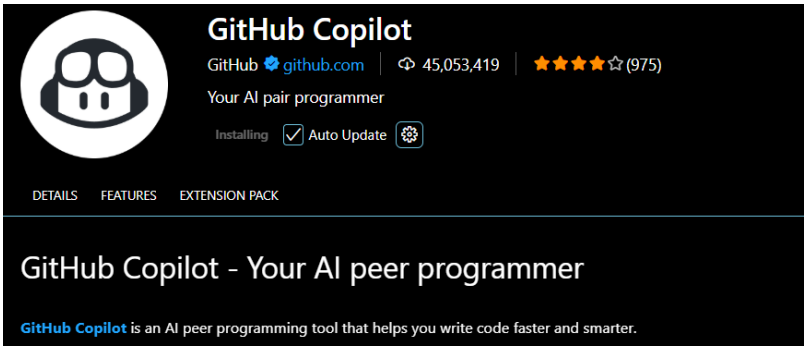
Software and Tools:

- **Visual Code:** Source-code editor used for writing and editing HTML, CSS, and JavaScript files.
- **GitHub Co-Pilot:** AI-powered code completion tool integrated into VS Code.
- **Gemini CLI:** Command Line Interface tool for interactive AI-assisted web development.
- **Node.js:** Required runtime environment for executing Gemini CLI.
- **Git:** Version control system used for managing source code and integrating with GitHub.
- **Web Browser:** Used to preview and test the portfolio website (e.g., Google Chrome, Opera Mini)

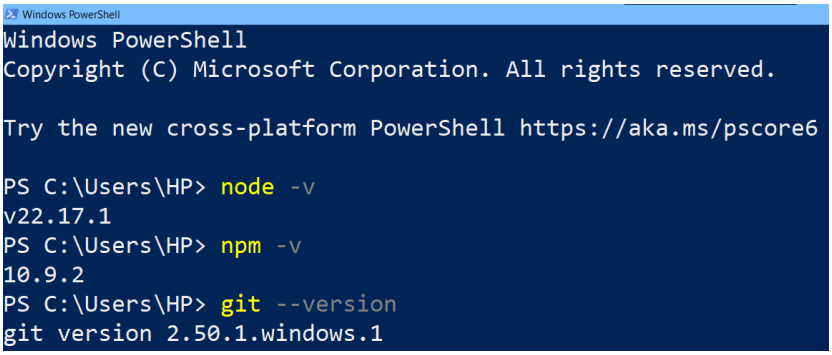
Working Procedure:

- **Portfolio Creation and Uploading Project to GitHub:** A personal portfolio website is first developed using HTML and CSS in Visual Studio Code. It features essential sections like Home, About, Education, Skills, and Contact. After completion, the project was uploaded to GitHub for version control and backup. A remote repository is created on GitHub, and the local files are pushed using Git commands after proper initialization and committing.

- **Installing GitHub Copilot:** GitHub Copilot is installed as an extension in Visual Studio Code. After installation, it is authenticated via GitHub account login. This enables AI-powered code suggestions within the editor.



- **Installing Required Tools:** Node.js and Git are downloaded and installed. Their successful installation is verified using the following commands in PowerShell:

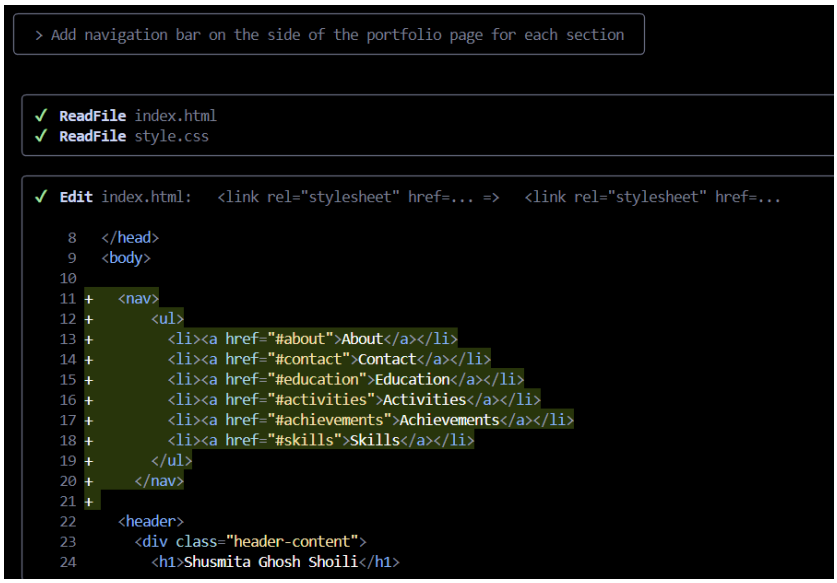


- **Setting Up Gemini CLI:** Gemini CLI is initialized in the terminal by executing a command on the new terminal of the folder where the portfolio is saved:

- **Authentication Process:** Upon running the command, a browser window will open for Google account authentication. Once access is granted, Gemini CLI is authorized to make changes in the local workspace.

- **Command-Based Editing with Gemini:** After successful authentication, natural language commands is to be entered directly into the terminal. Gemini CLI will interpret the commands and make appropriate edits to the portfolio files.

  For example, we have given a command "Add navigation bar on the side of the portfolio page for each section" to make changes to the original portfolio file.



- **Tracking Changes Before and After using Gemini Commands:**

Before:

After:

Result:

The portfolio website is successfully created and enhanced using GitHub Copilot and Gemini CLI. Copilot assists in generating and refining code efficiently, while Gemini CLI enables real-time editing through natural language commands. The final output is a well-structured, responsive, and interactive portfolio hosted on GitHub.

Conclusion:

This lab demonstrates the effective integration of GitHub Copilot and Gemini CLI in a real-world web development project. By combining AI-assisted coding with natural language command execution, the development process becomes faster, more efficient, and user-friendly. These tools not only enhance productivity but also support developers in improving the quality and interactivity of their web applications.