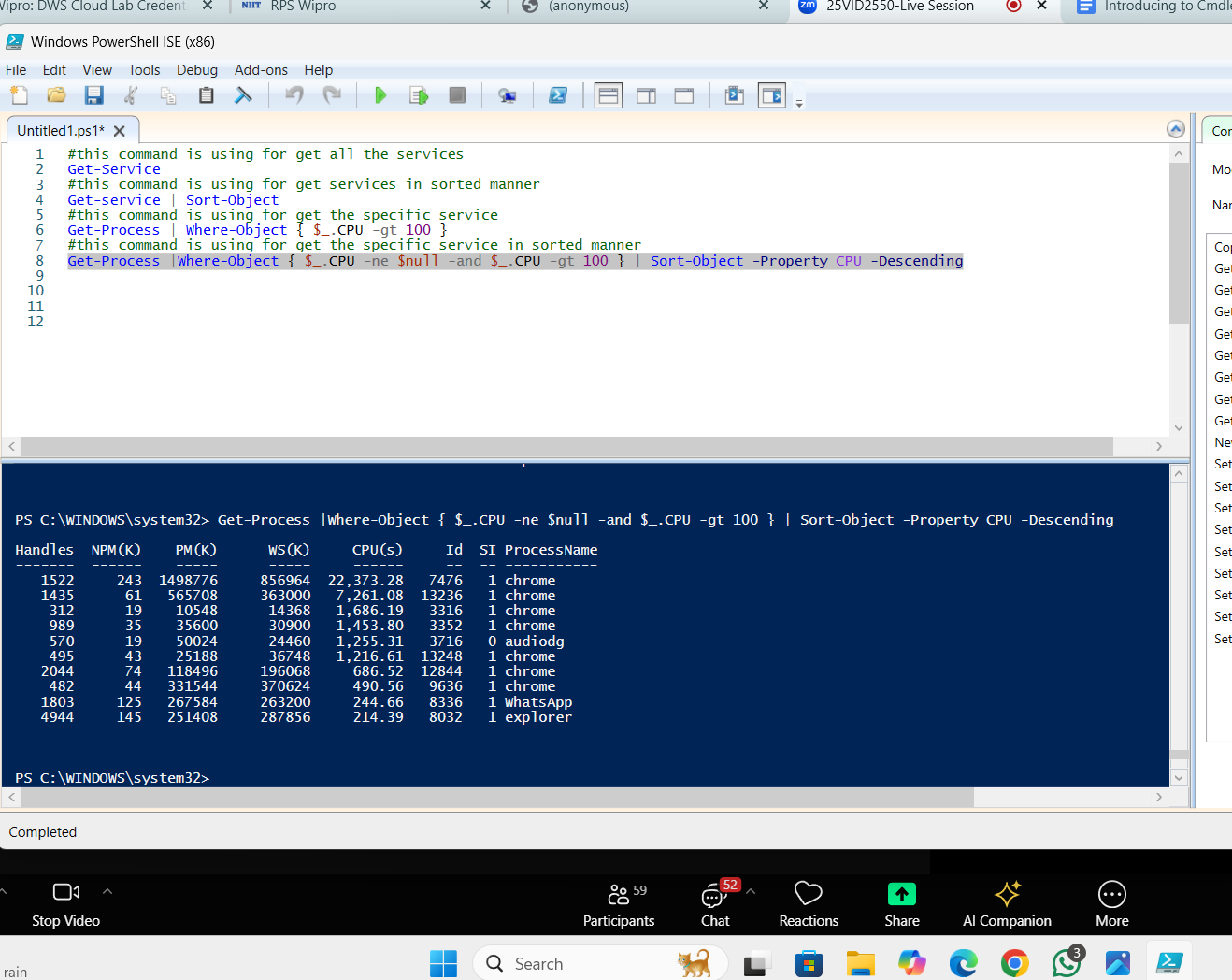
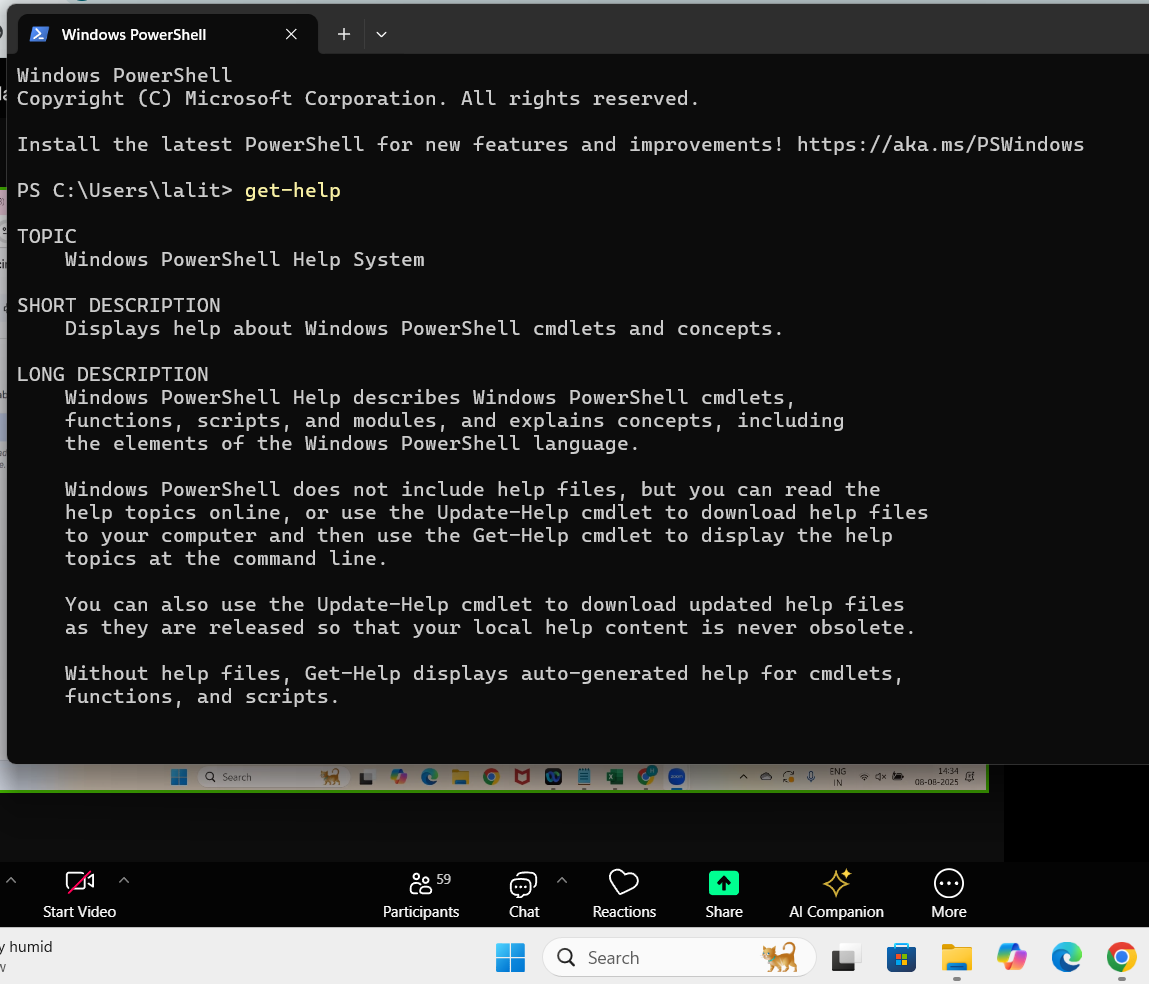
Assignment of Date :08-08-2025   
Name:Lalit Joshi

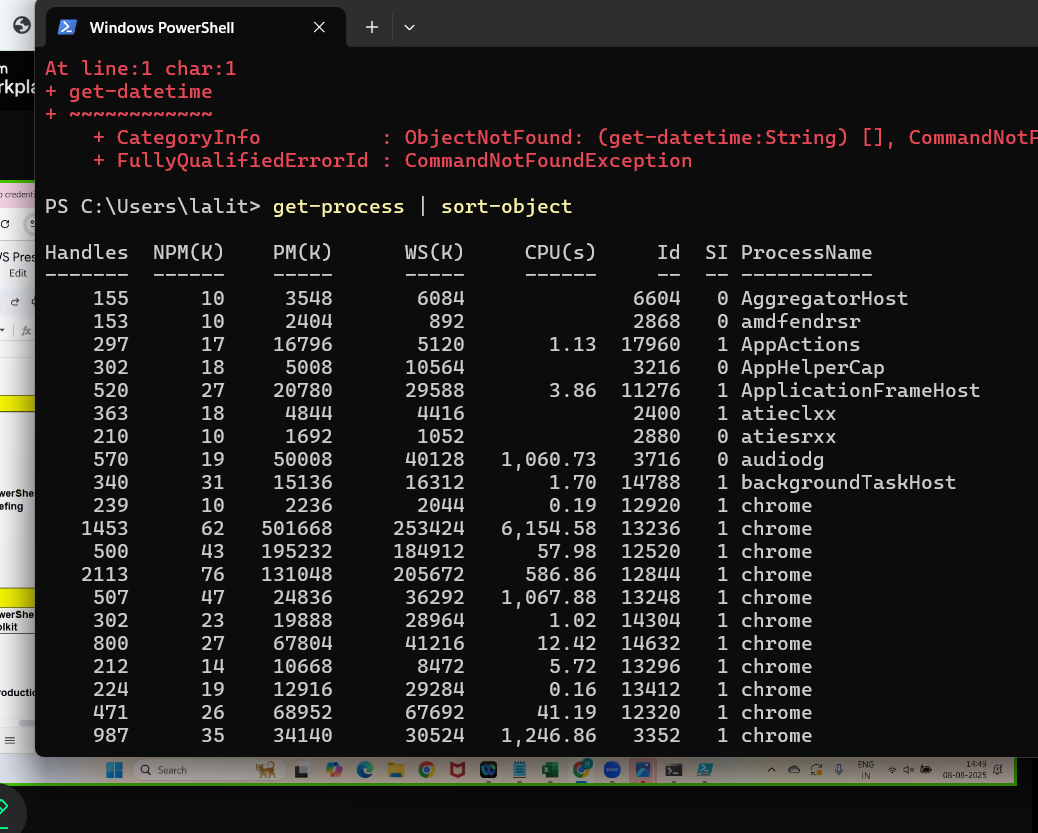
Email :Joshilalit2275@gmail.com

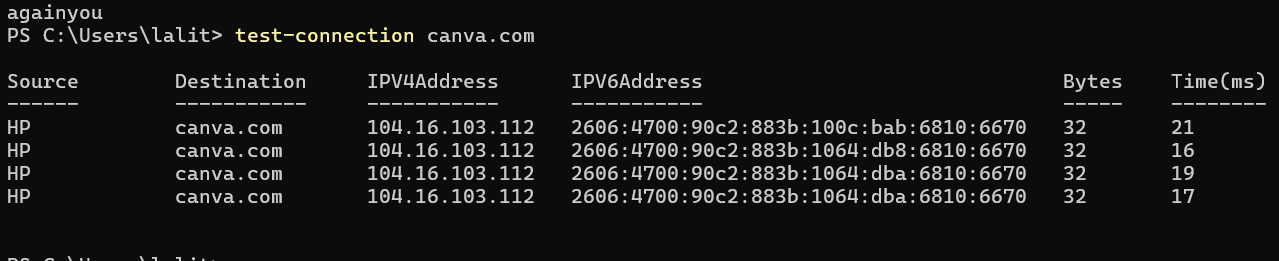
**Topic 1:** Introducing to Cmdlets and The PowerShell Pipeline

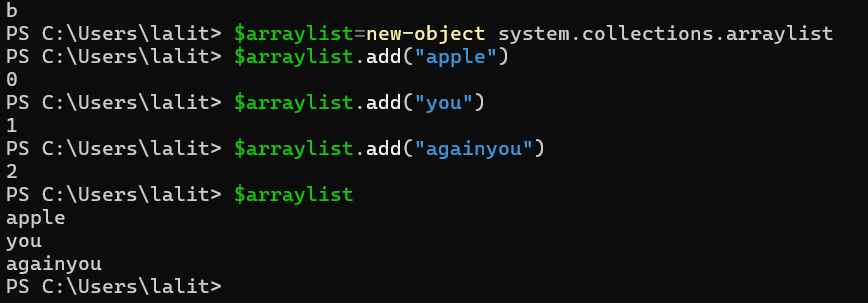
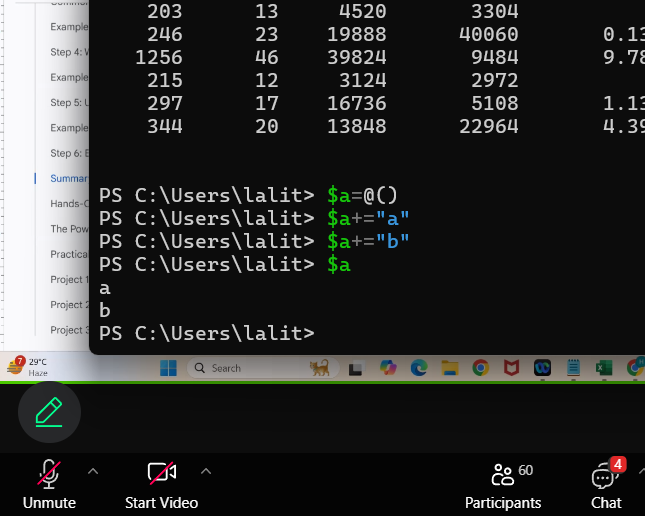
Cmdlets (pronounced *command-lets*) are lightweight PowerShell commands built into the framework, designed to perform specific tasks. They follow a Verb-Noun naming convention (like Get-Process or Set-Date) for clarity and consistency. Unlike traditional commands in CMD or Unix shells, cmdlets are .NET objects, which means they work with structured data rather than plain text, making automation and scripting more powerful and flexible.

The PowerShell pipeline allows you to pass the output of one cmdlet directly into another, enabling a smooth flow of data. Each stage in the pipeline works with objects instead of text, which means you can filter, sort, and manipulate data without losing structure. This design makes complex tasks easier by chaining simple commands together into a logical sequence.









PROJECT -1 : **Exploring Cmdlet Syntax**

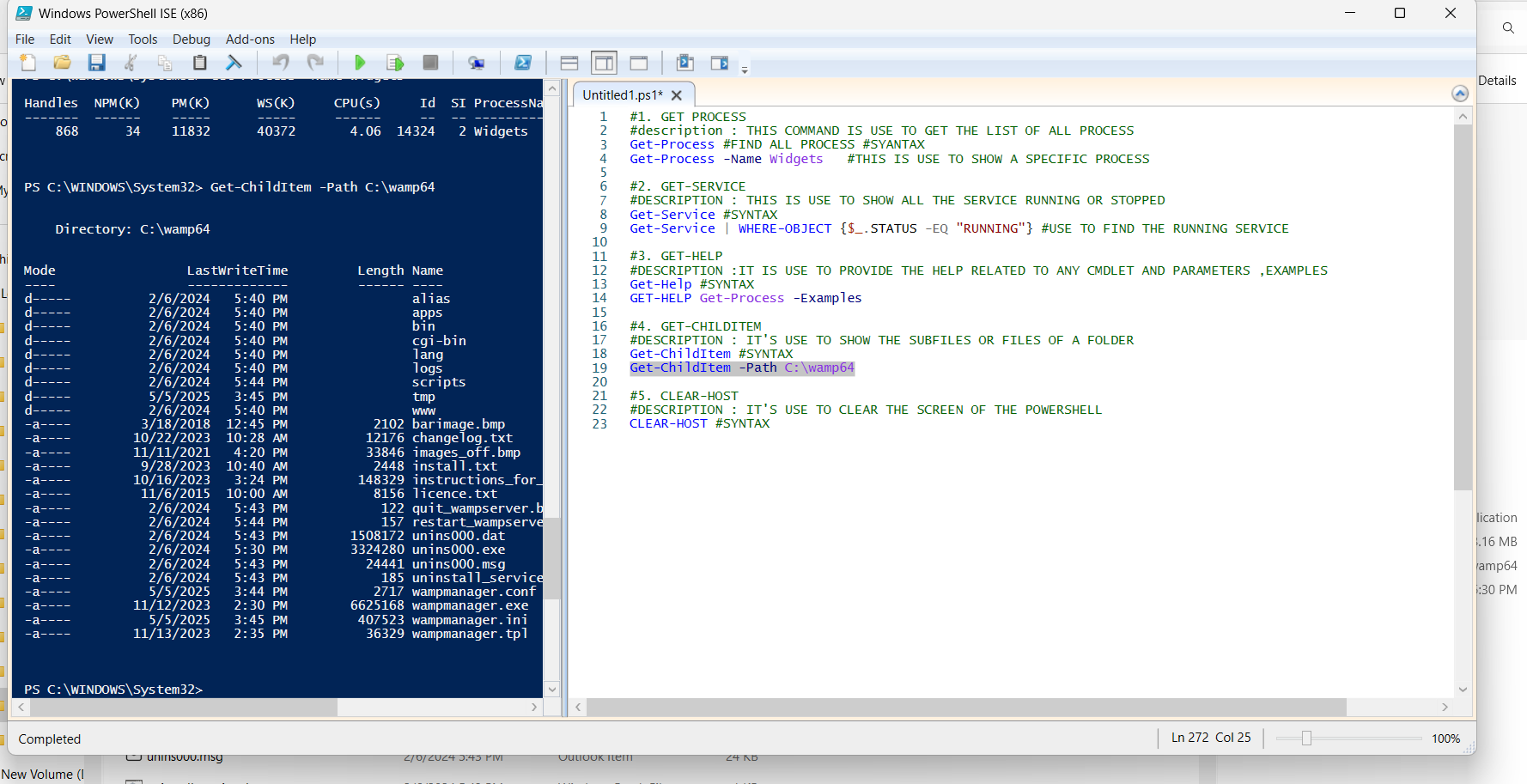
**1) Get-Process  
Get-Process is used to display a list of all processes running on the system. It shows details like process name, ID, CPU usage, and more. You can also view a specific process using the -Name parameter, for example: Get-Process -Name Widgets.**

**2) Get-Service  
Get-Service provides information about all Windows services, whether they are running or stopped. You can filter it to show only running services using Where-Object, for example: Get-Service | Where-Object {$\_.Status -eq "Running"}.**

**3) Get-Help  
Get-Help displays documentation for any PowerShell cmdlet, including its parameters and examples. It’s especially useful for beginners to quickly learn the syntax and practical usage. Example: Get-Help Get-Process -Examples.**

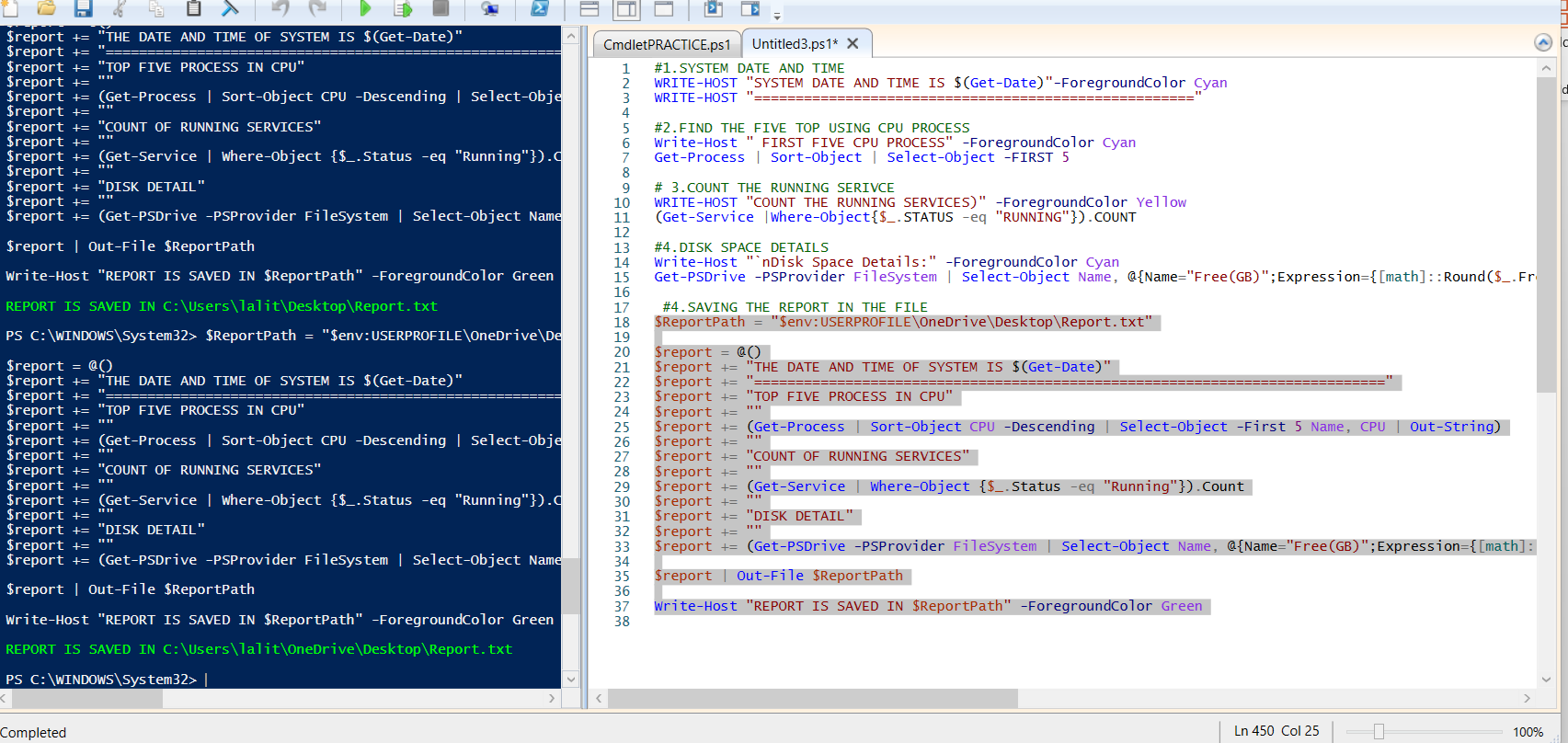
**4) Get-ChildItem  
Get-ChildItem lists all files and subfolders inside a folder. You can specify a path to view its contents, such as Get-ChildItem -Path C:\wamp64. It works similarly to the dir command in Command Prompt or ls in Linux.**

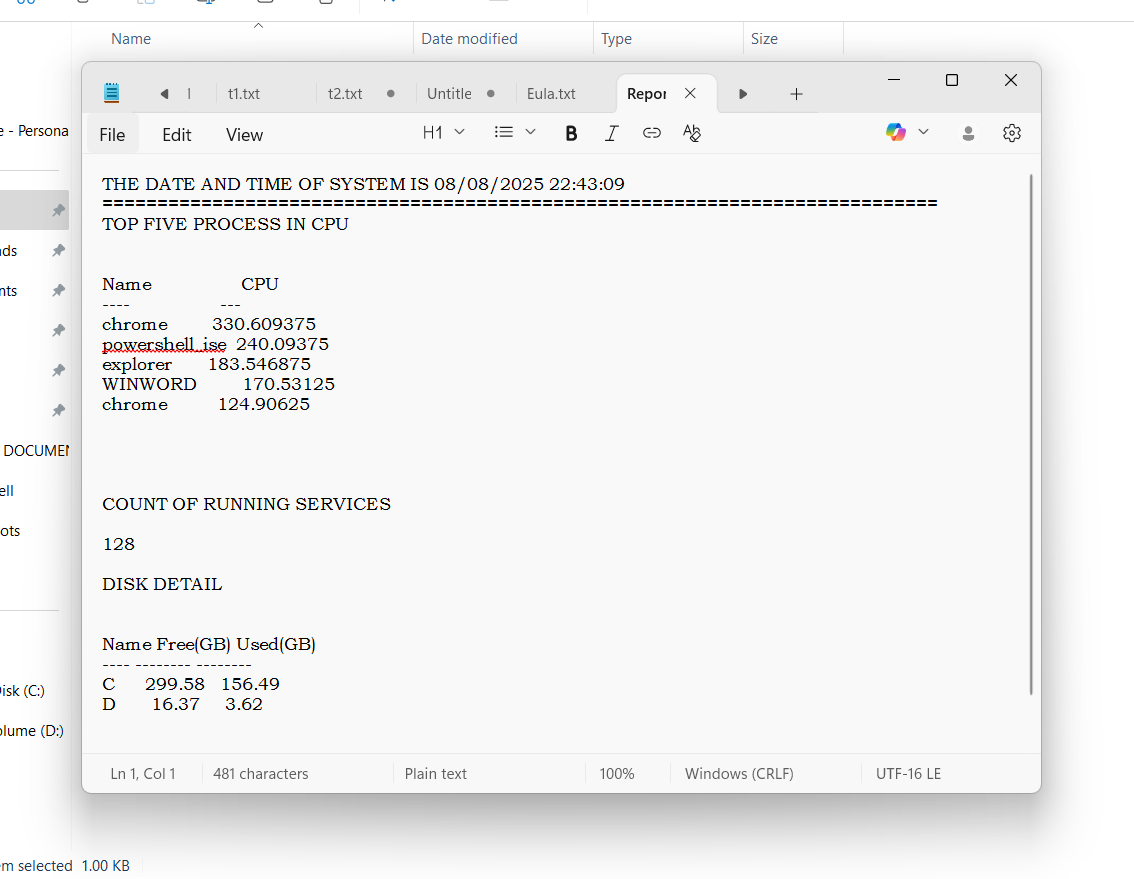
**5) Clear-Host  
Clear-Host clears the PowerShell screen, allowing you to work on a fresh, clean console. You can use either Clear-Host or the short form cls to run it.**



PROJECT -2 : **Automate a Task with a Cmdlet Script**

This PowerShell script is designed to **automatically generate a system health report** and save it as a text file on the desktop. It collects several types of system information, including the **current date and time**, the **top five processes using the most CPU**, the **count of all currently running services**, and **disk space details** (both free and used space in gigabytes). The script stores all this data in a variable as text, then writes it to Report.txt on the user’s desktop using Out-File. By running this script, users can quickly check important system status information without manually typing multiple commands, making monitoring tasks faster and easier.





**Project-3:Create a PowerShell Cmdlet Cheat Sheet**

Creating a PowerShell Cmdlet Cheat Sheet involves compiling a quick reference guide of commonly used cmdlets, their descriptions, syntax, key parameters, and example usage. The goal is to provide a handy resource for quickly recalling how to use essential commands without searching documentation repeatedly. By listing cmdlets such as Get-Process, Get-Service, Get-Help, Get-ChildItem, and Clear-Host along with clear examples, the cheat sheet helps both beginners and advanced users work faster and more efficiently. This practice not only improves command recall but also strengthens understanding of cmdlet structure, parameters, and output formatting.

