**NAME:** - ANSHUL SAINI  
**MAIL:** - [sainianshul7565@gmail.com](mailto:sainianshul7565@gmail.com)

**User ID**: - 34732

**Date: -** 08/08/2025

**Batch: -** DWS\_B5\_25VID2550

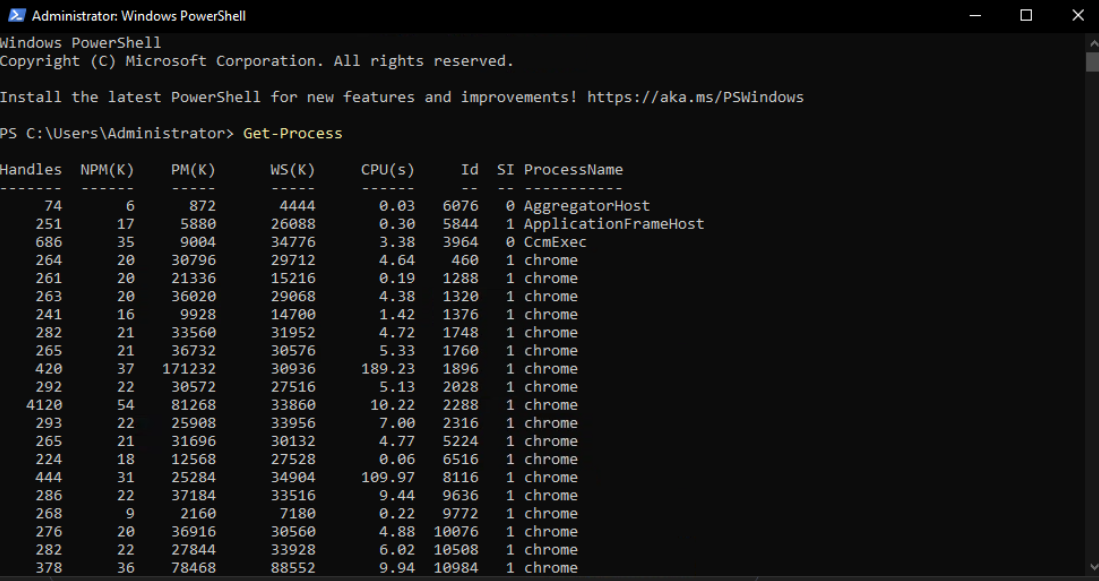
**Topic Assignment: -**

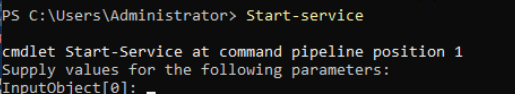
* Introducing to Cmdlets
* The PowerShell Pipeline
* Key Cmdlets
* Pipeline Filtering & Operators
* Scripting Overview
* Project: Exploring Cmdlet Syntax
* Project: Automate a Task with a Cmdlet Script
* Project: Create a PowerShell Cmdlet Cheat Sheet
* **Introducing to Cmdlets: -**

Cmdlets are small commands used in Windows PowerShell to perform specific tasks.

They follow a "Verb-Noun" format, like **Get-Process** or **Start-Service**, which makes their purpose easy to understand. Cmdlets work with the pipeline, meaning you can connect them together to do complex jobs step by step. Instead of just handling text, they deal with objects, making scripts more powerful. PowerShell includes many built-in cmdlets, and users can create their own. Cmdlets aren’t separate programs—they’re part of PowerShell itself. They take input, process it, and give output that can be used by other cmdlets in the pipeline.

**Practice Activity: -**

****



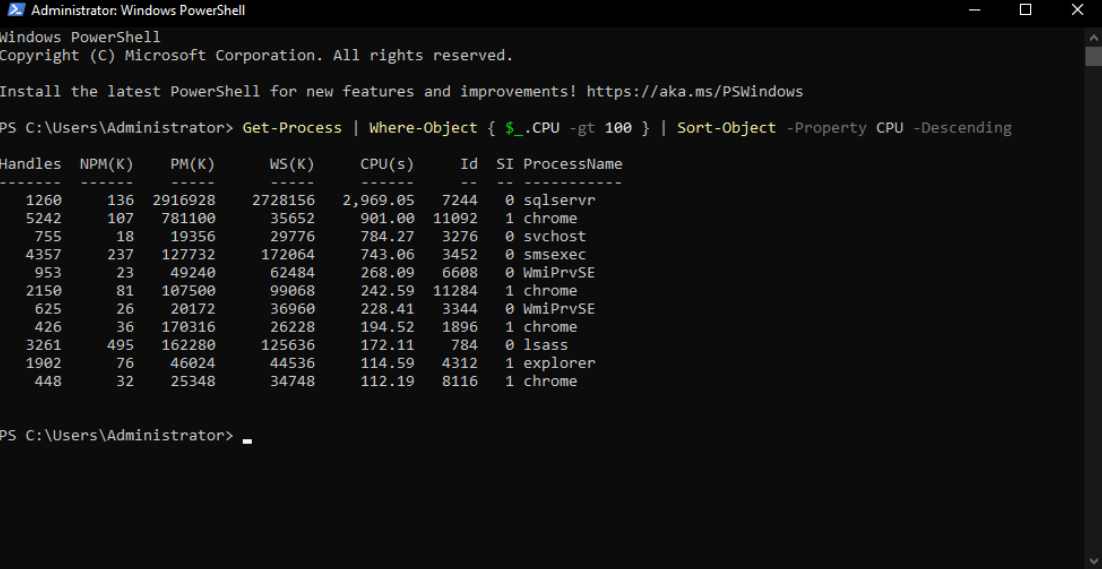
* **The PowerShell Pipeline: -**

The PowerShell pipeline is a way to connect commands so they work together, passing data from one to the next. It uses the | symbol to link cmdlets. For example, **Get-Process | Sort-Object CPU** gets a list of processes and sorts them by CPU usage. Each command sends its output as objects to the next command as input. This makes tasks easier and more efficient, letting you build powerful scripts step by step. Instead of handling plain text, the pipeline deals with rich data, making it flexible and useful for managing systems, files, and services in Windows PowerShell.

**Commands: -**

Get-Process | Where-Object { $\_.CPU -gt 100 } | Sort-Object -Property CPU -Descending

**Practice Activity: -**

**Practice Activity: -**

* **Key Cmdlets: -**

**Get-Process –** Shows all running processes on your system.

**Get-Service –** Lists all services and their status.

**Stop-Service –** Stops a running service.

**Start-Service –** Starts a stopped service.

**Get-EventLog –** Views events from Windows logs.

**Get-Item –** Shows details of a specific file or folder.

**Set-Location –** Changes the current working directory.

**Copy-Item –** Copies a file or folder to another location.

**Move-Item –** Moves a file or folder to another location.

**Remove-Item –** Deletes a file or folder.

**Rename-Item –** Renames a file or folder.

**Get-Content –** Reads the content of a file.

**Set-Content –** Writes or replaces content in a file.

**Select-String –** Searches text in files (like grep).

**Get-ADUser –** Retrieves information about an Active Directory user.

**New-ADUser –** Creates a new Active Directory user.

**Get-ComputerInfo –** Displays detailed system information.

**Set-Date –** Changes the system date and time.

**Where-Object –** Filters results based on a condition.

**Sort-Object –** Sorts output by a property.

**Set-ExecutionPolicy –** Changes PowerShell script execution rules.

**Get-Help –** Shows help for commands and concepts.

**Get-ChildItem –** Lists files and folders in a directory.

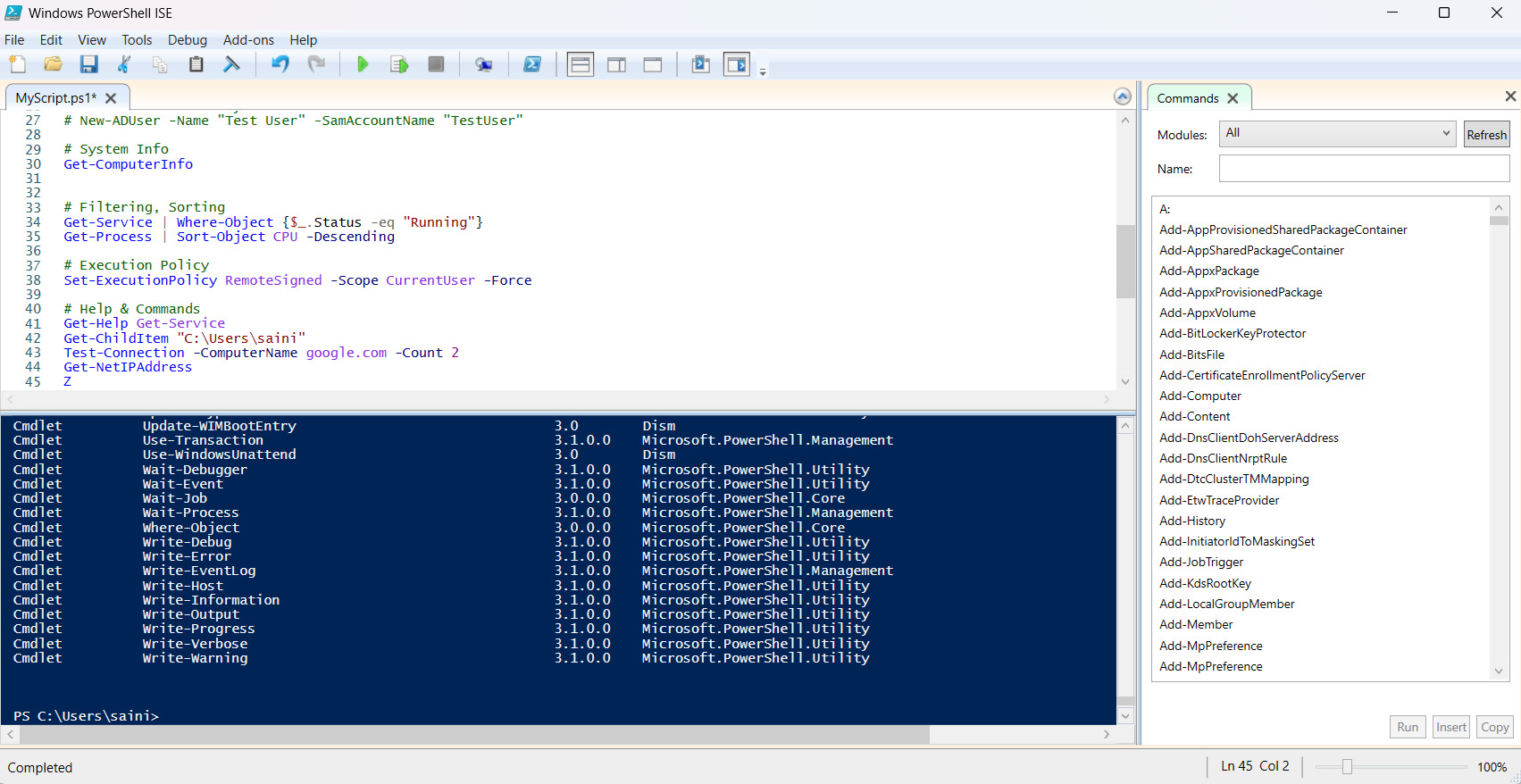
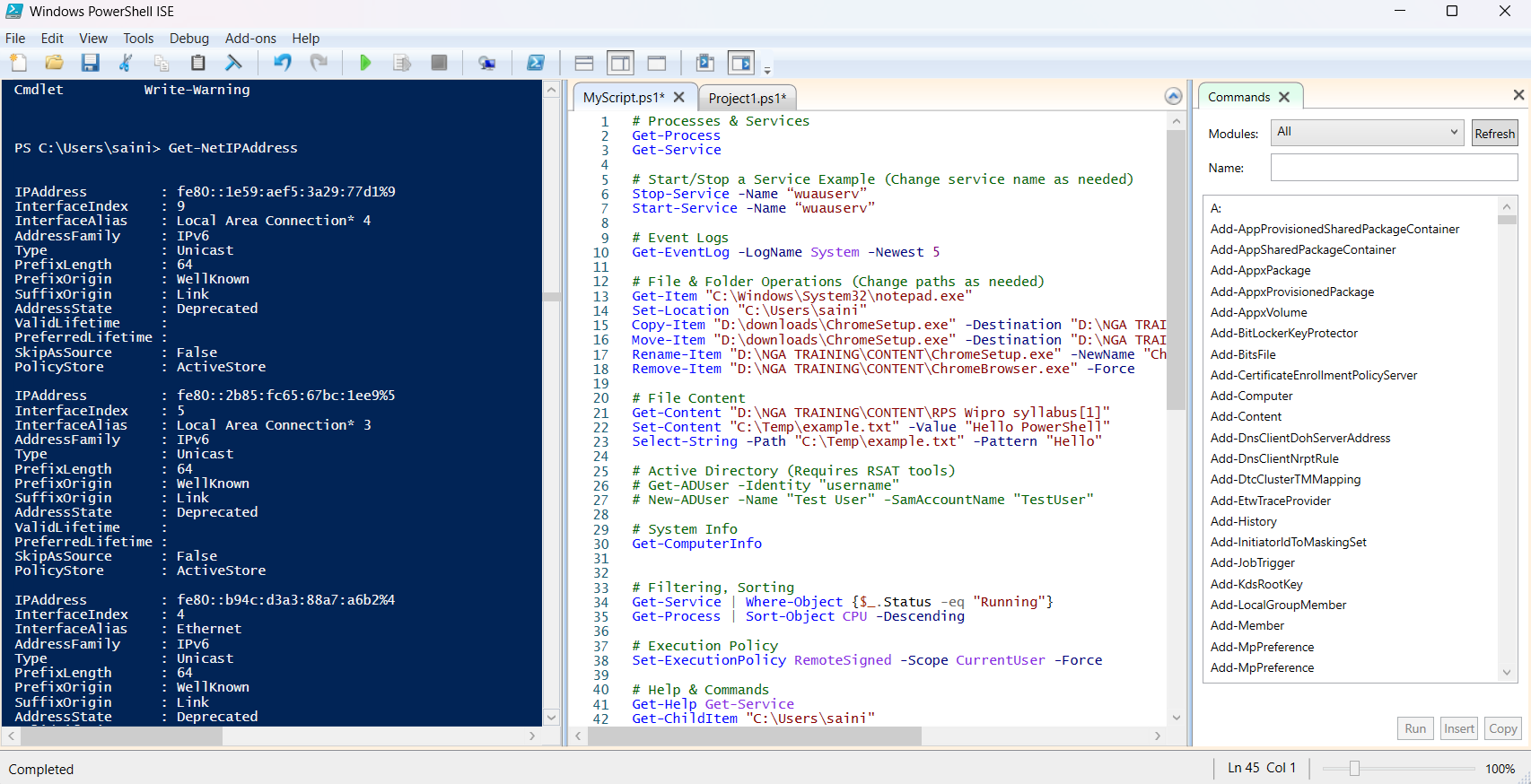
**Remove-Item –** (Duplicate) Deletes a file or folder.

**Test-Connection –** Checks if a device is reachable (like ping).

**Get-NetIPAddress –** Shows IP address information.

**Get-Command –** Lists all available PowerShell commands.

**Practice Activity: -**



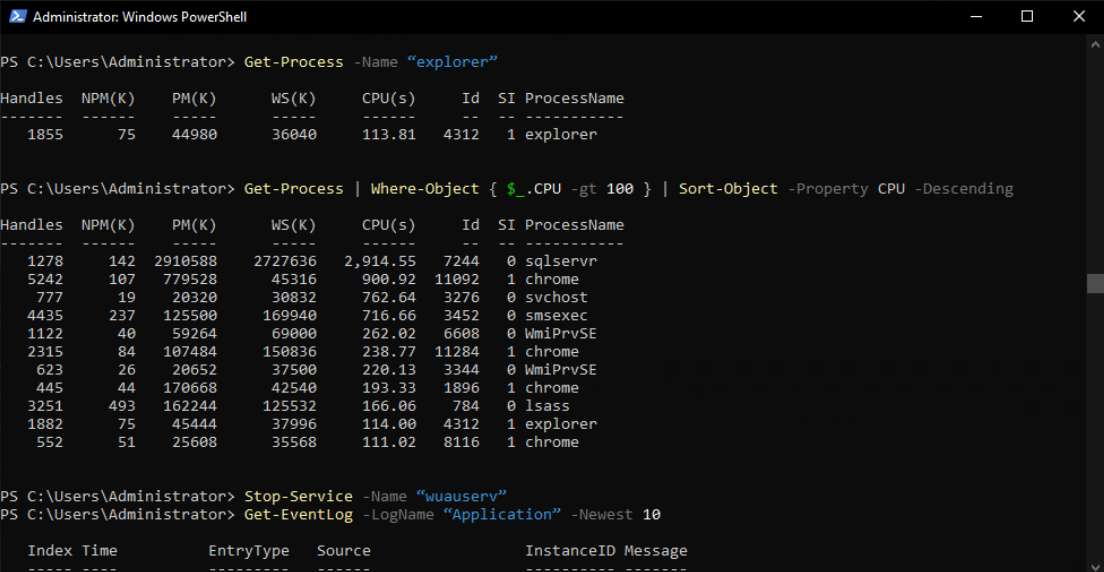
* **Pipeline Filtering & Operators**

Filtering in PowerShell pipelines helps narrow down results based on conditions. The most common cmdlet used is **Where-Object**, which lets you apply logic to filter objects.

Example:

Get-Process | Where-Object { $\_.CPU -gt 100 }

**Practice Activity: -**



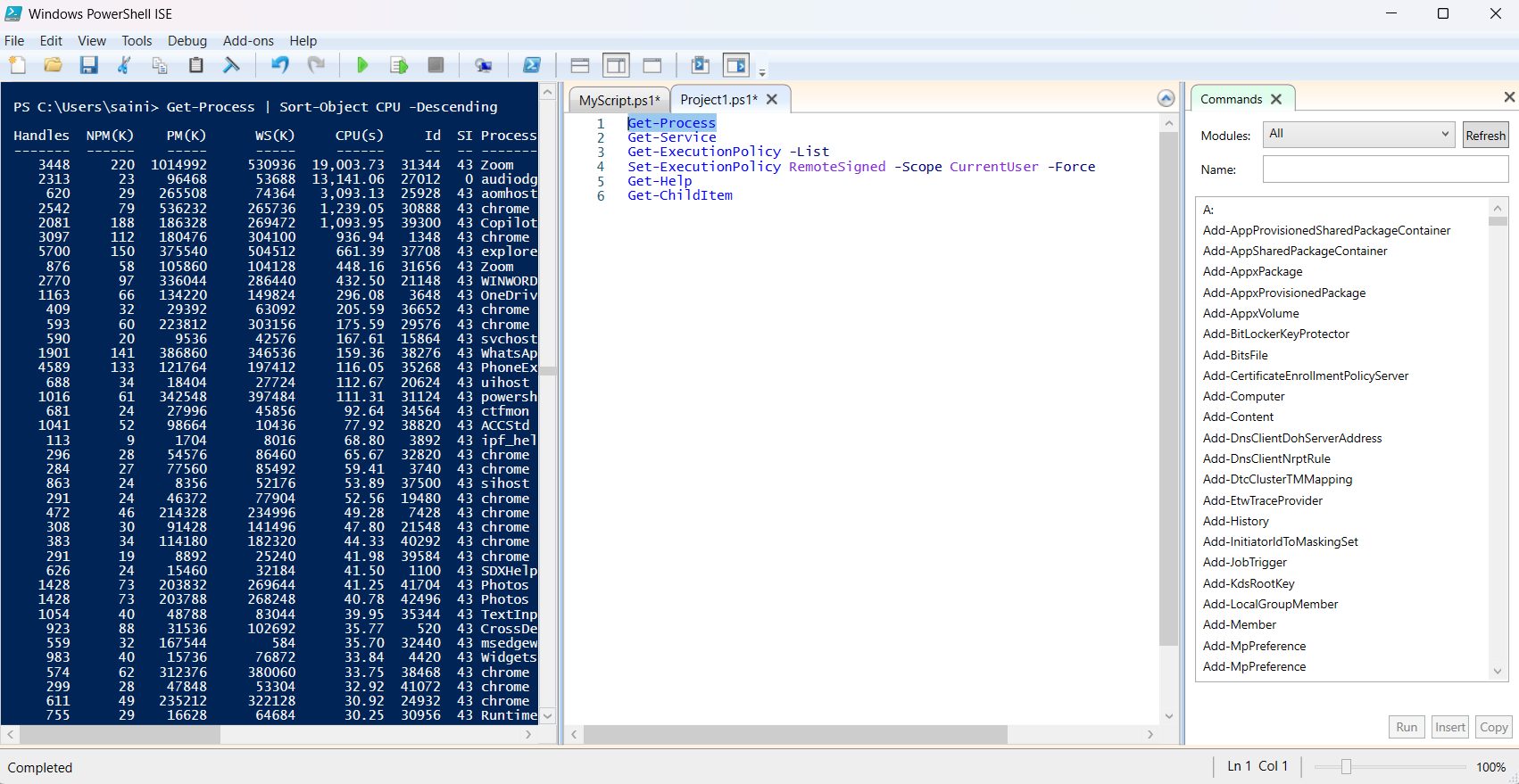
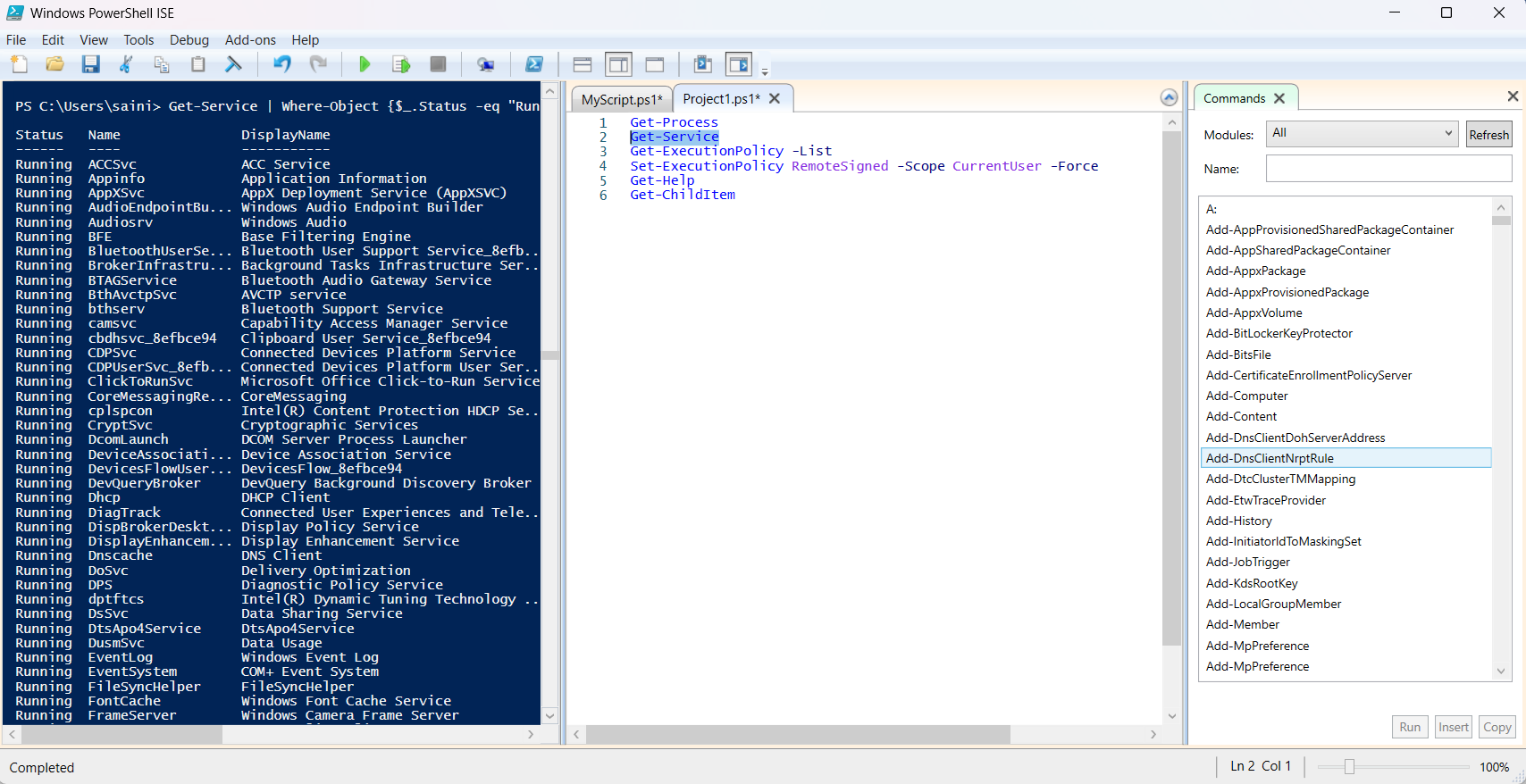
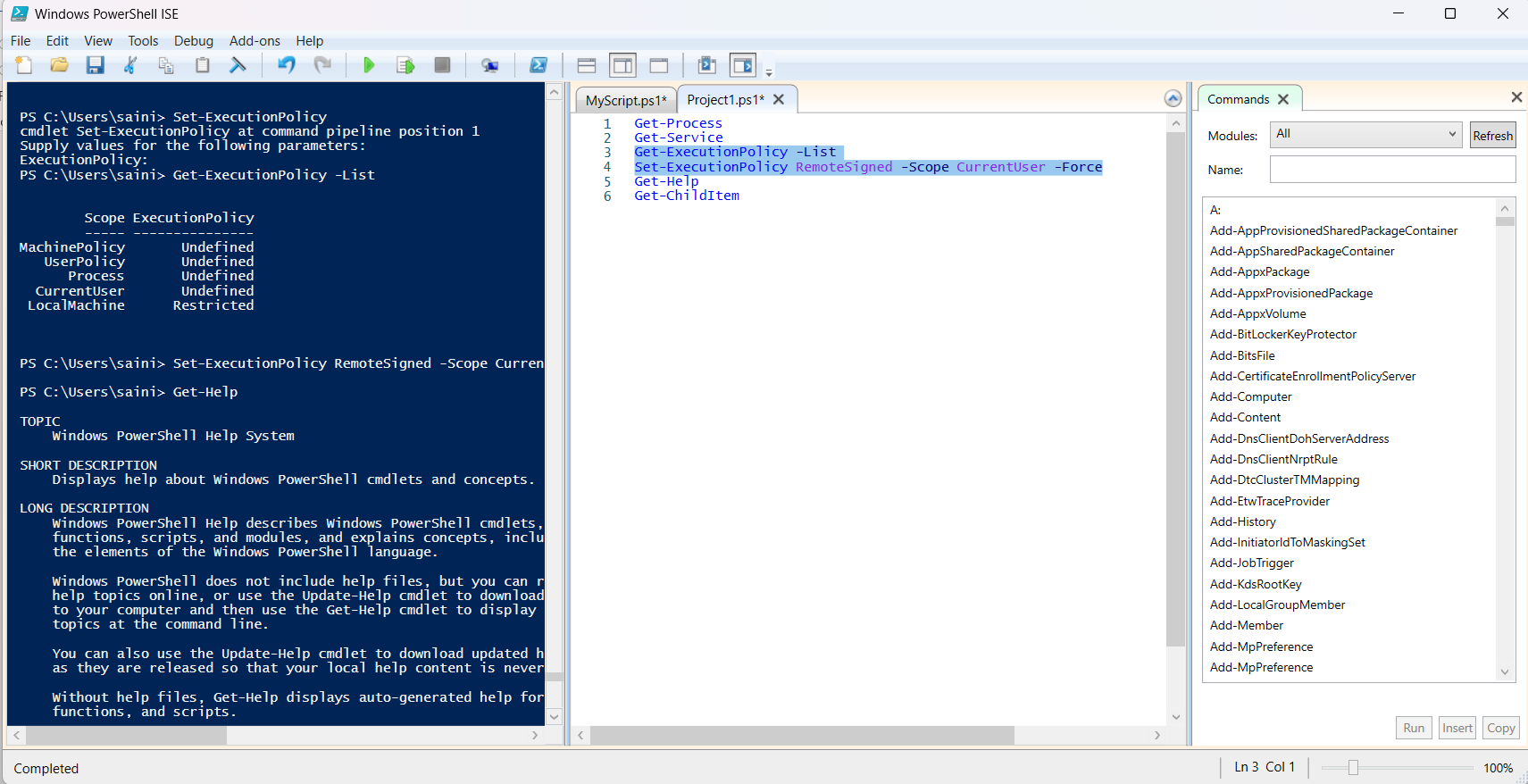
* **Scripting Overview**

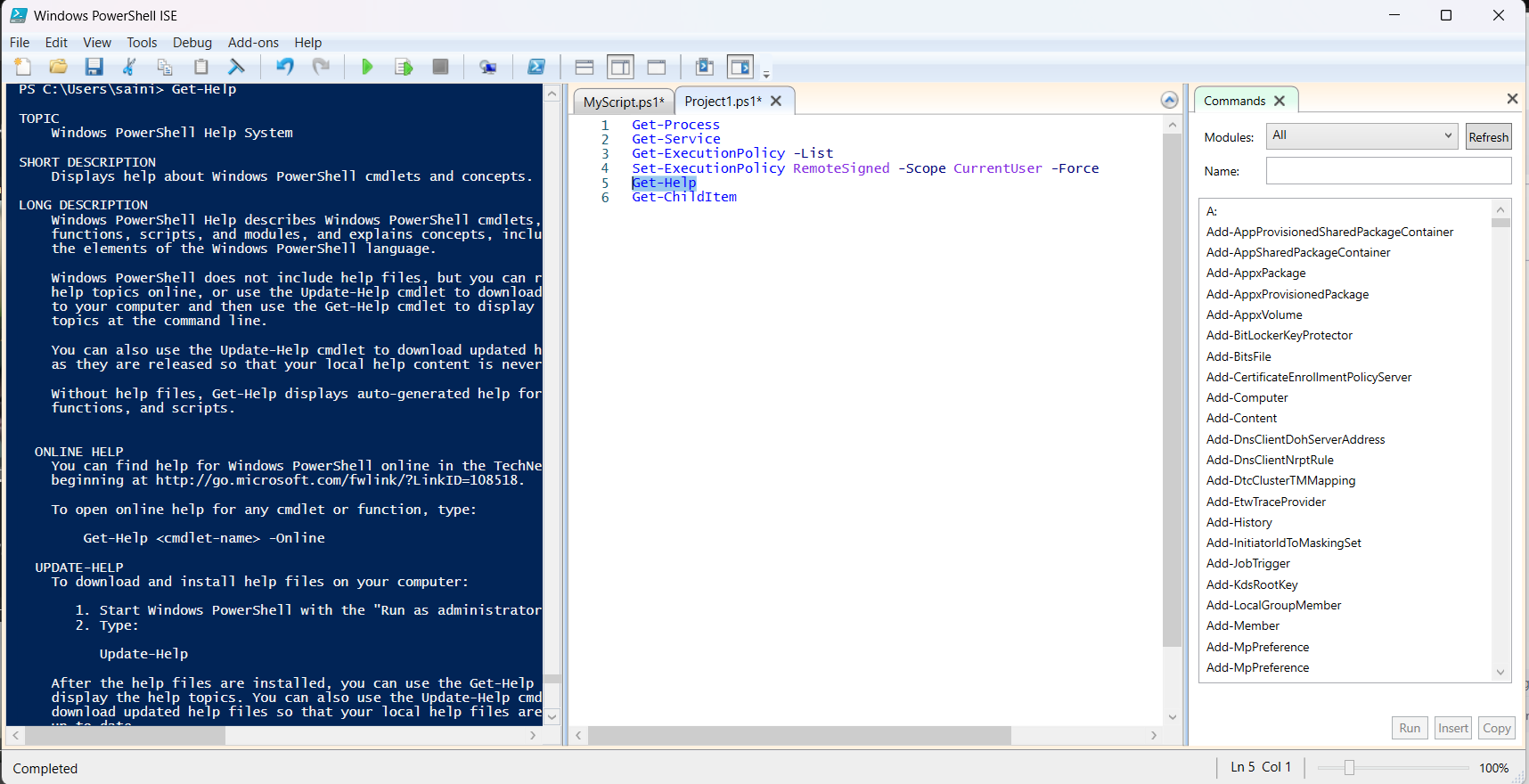
PowerShell scripting allows you to:

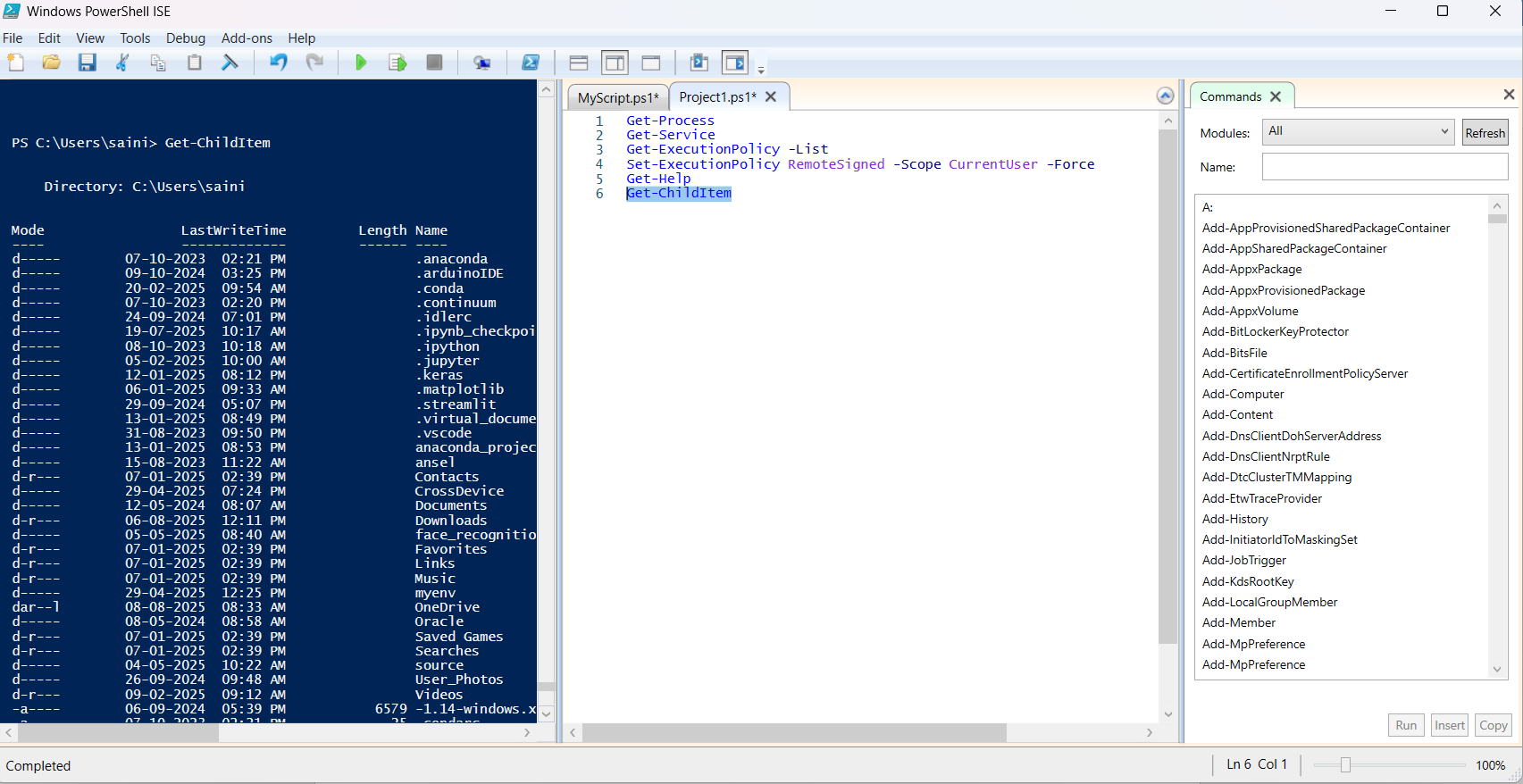
* Automate repetitive tasks
* Manage system configurations
* Build and deploy solutions (e.g., CI/CD pipelines)
* Share reusable code with others
* **Project: Exploring Cmdlet Syntax**

Choose five commonly used cmdlets (e.g., Get-Process, Get- Service, Set-ExecutionPolicy, Get-Help, and Get-ChildItem).

**Practice Activity: -**







* **Project: Automate a Task with a Cmdlet Script**

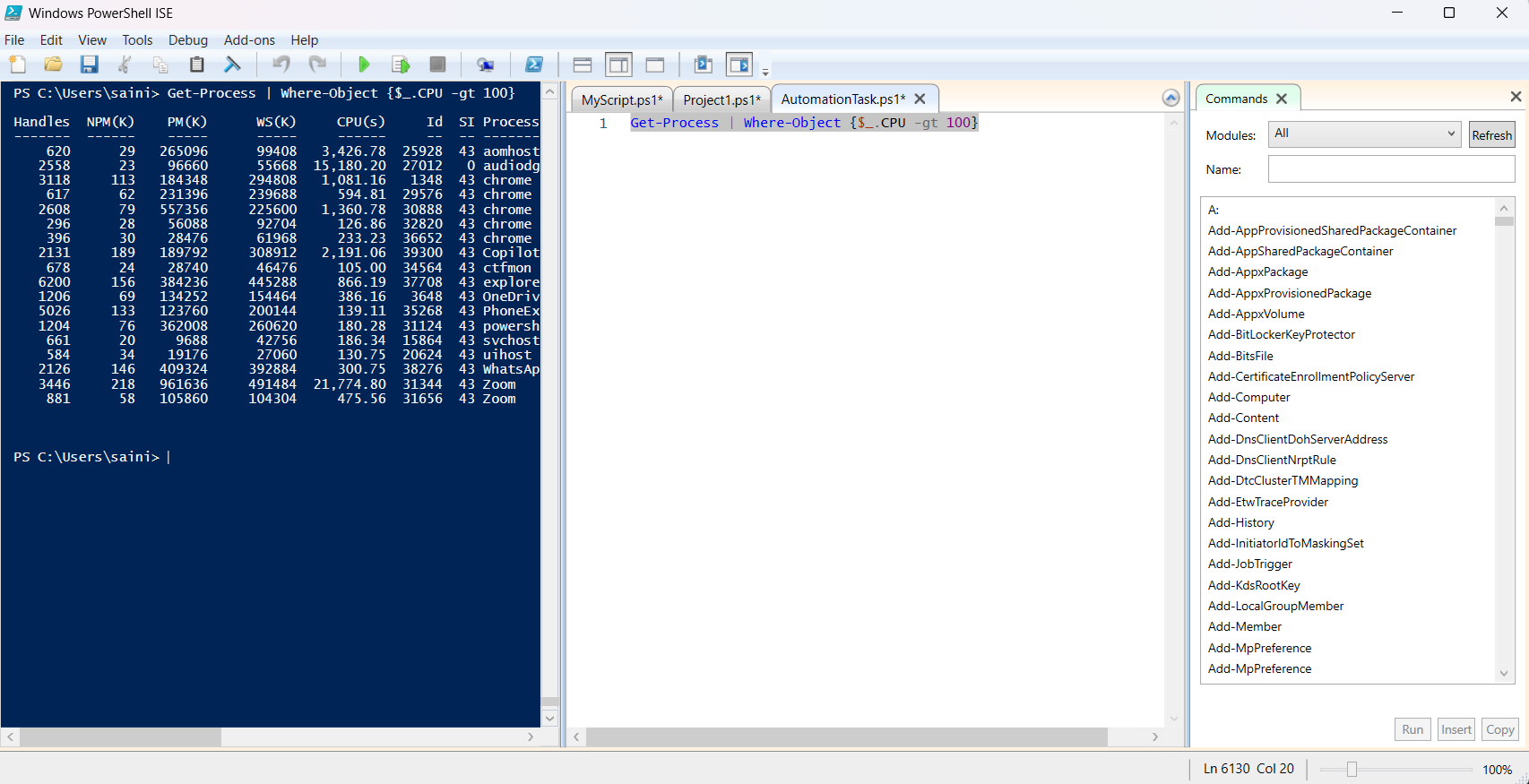
Create a new script file named AutomationTask.ps1.

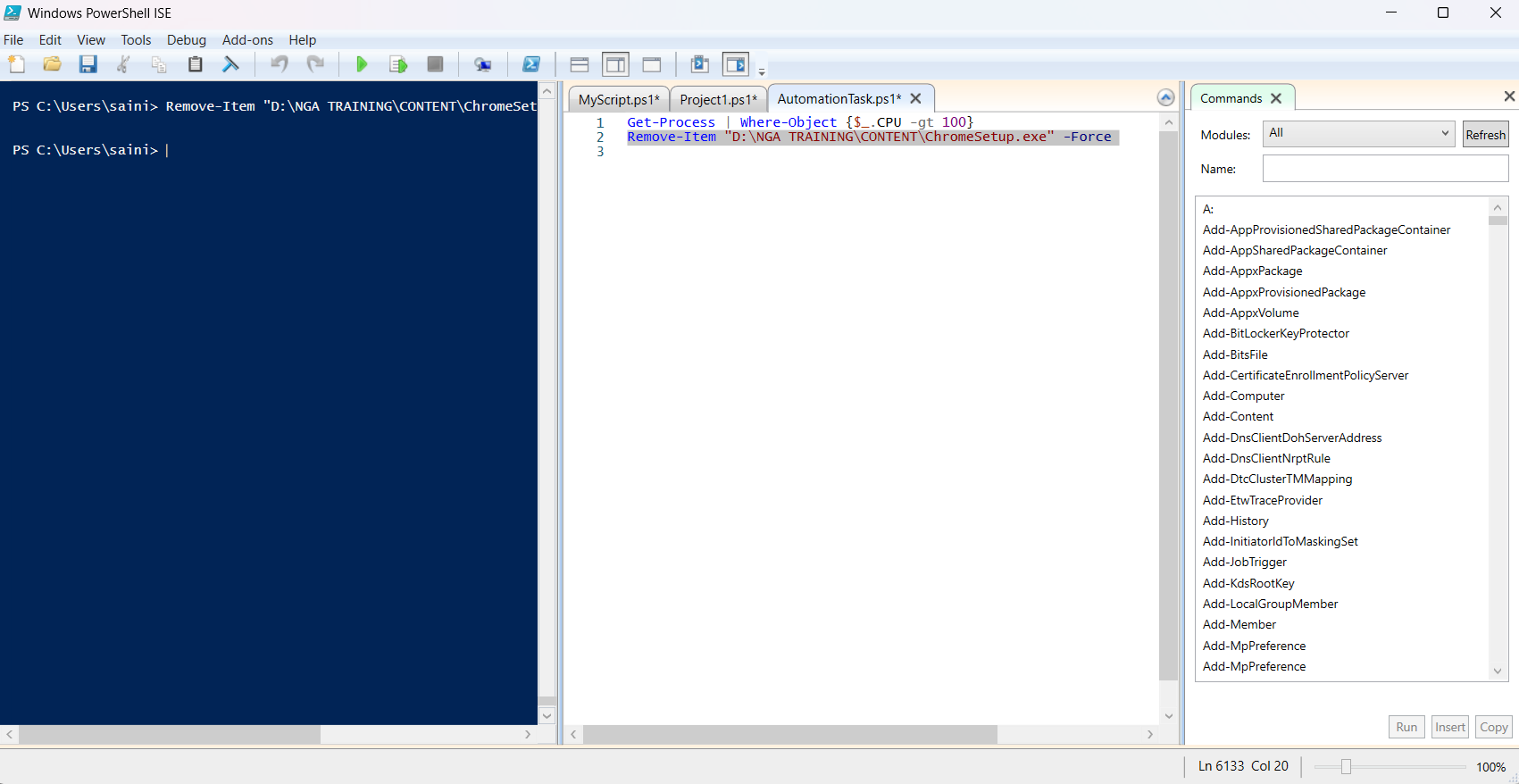
**Use cmdlets like:**

**●** Get-Process | Where-Object {$\_.CPU -gt 100} to find processes consuming significant CPU.

● Use Remove-Item to delete files in a temporary directory.

**Practice Activity: -**

****



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

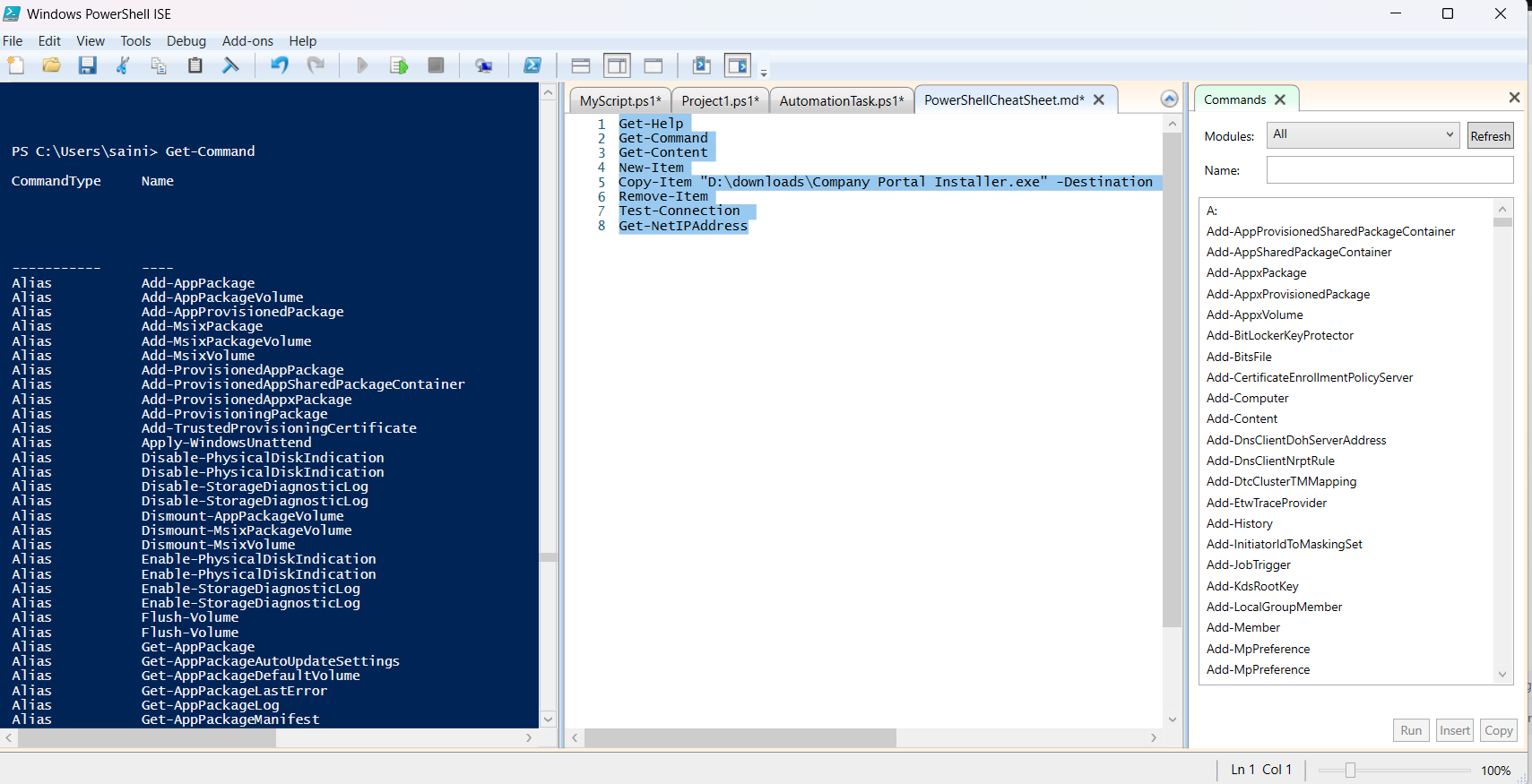
Create a new Markdown file named PowerShellCheatSheet.md.

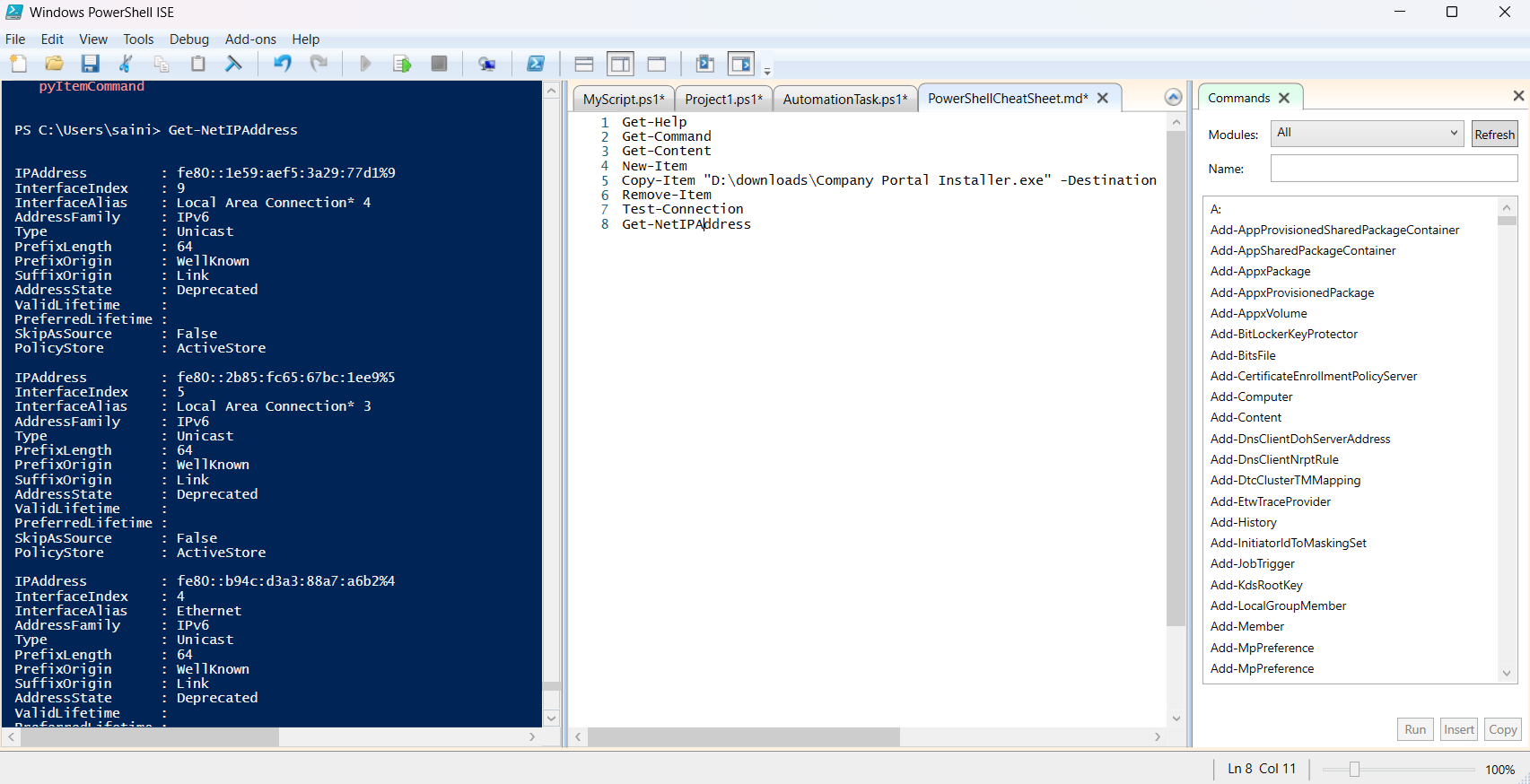
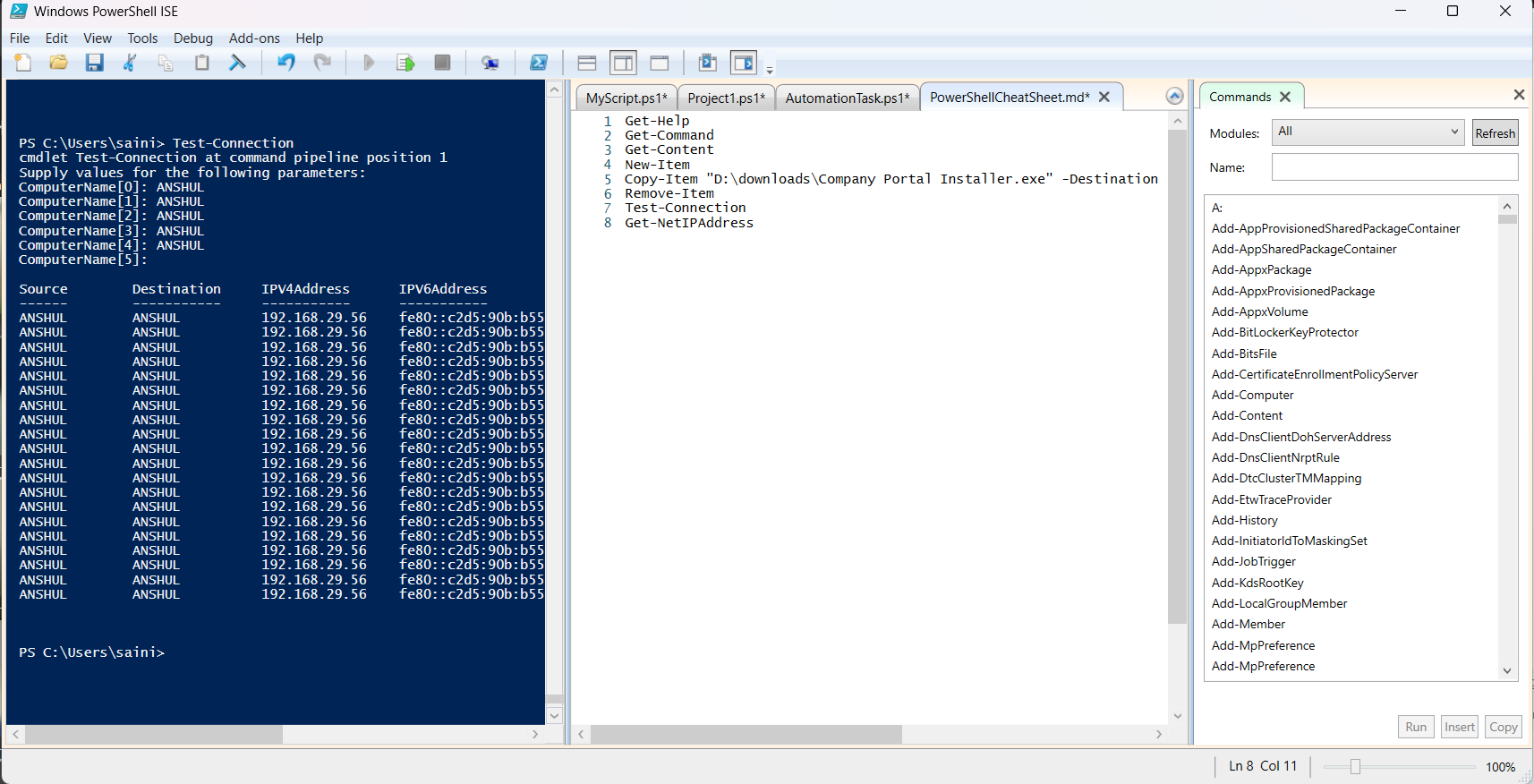
Basic Cmdlets:

List common cmdlets like Get-Help, Get-Command, Get-Content, etc., with a brief description.

● File System Cmdlets: Include cmdlets such as New-Item, Copy-Item, and Remove-Item.

● Network Cmdlets: List cmdlets like Test-Connection and Get-NetIPAddress.

**Practice Activity: -**

****