6. Using terminal and basic shell commands (Windows Operating System)

What the Windows Terminal Actually Is

Most beginners think the terminal is “just a black screen.” That’s wrong. Treat it as a command parser that translates your text instructions into OS actions.

Windows provides several shells:

Primary Shells

1. **Command Prompt (cmd.exe)** — legacy shell, widely compatible.
2. **PowerShell** — modern, more powerful, object-based.
3. **Windows Terminal** — a UI wrapper that can run both cmd and PowerShell tabs.

For foundational learning, PowerShell offers cleaner, more consistent behavior.
But cmd philosophy is simpler, so we’ll blend both where useful.

First Principles: How Commands Work

Every command follow this structure:

```
Command [arguments] [options / switches]
```

Example;

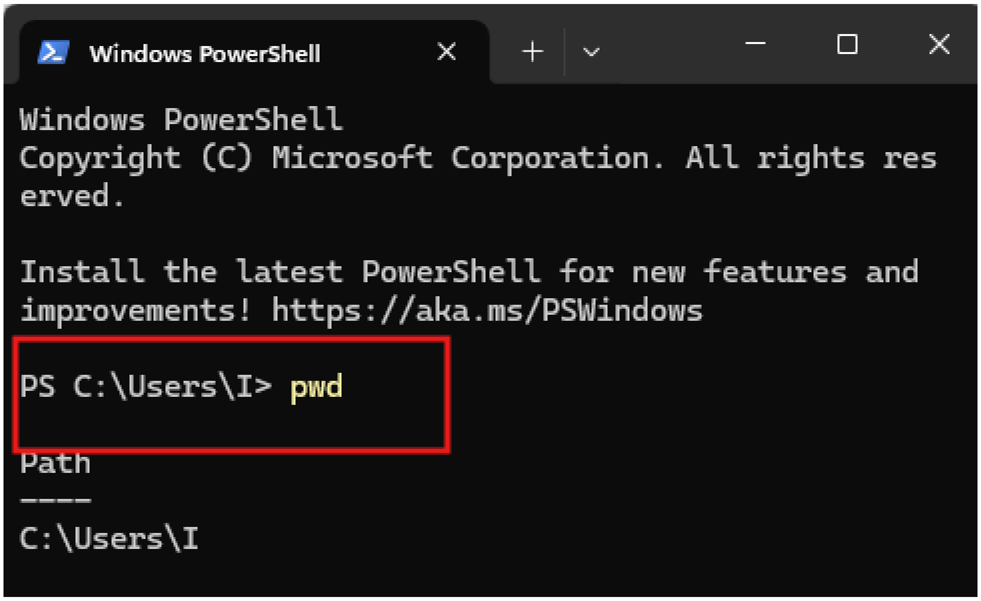
```
cd Documents
dir /a
mkdir “New Folder”
```

Key Truth:

The shell always executes commands relative to your current working directory. If you don't control your working directory, you can't control the terminal.

Essential Concepts Before Commands

Current Working Directory



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights res
erved.

Install the latest PowerShell for new features and
improvements! https://aka.ms/PSWindows

PS C:\Users\I> pwd

Path
----
C:\Users\I
```

This tells you where you are in the filesystem.

Paths (Relative vs Absolute)

Example absolute path:
C:\Users\Mehedi\Desktop

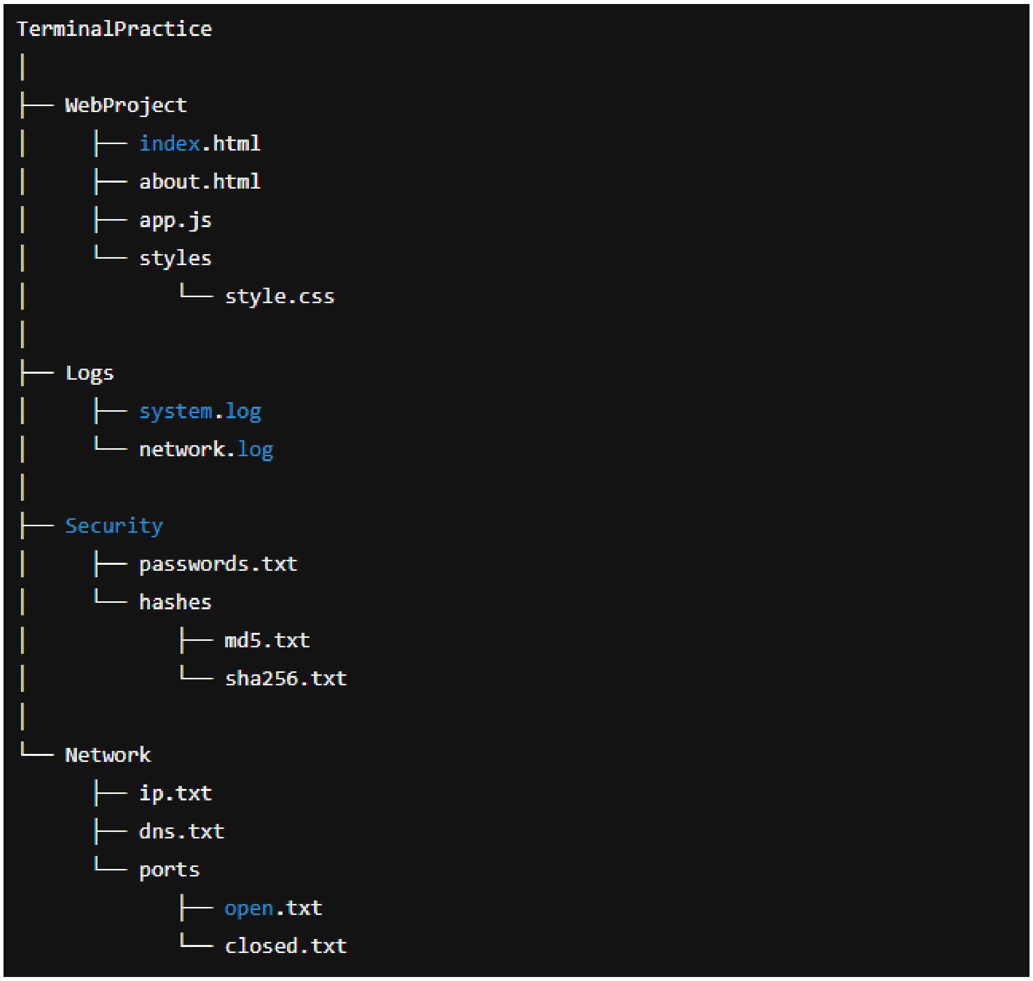
Relative path:
cd ..\Documents

Memory trick:

- Absolute = starts from drive root
- Relative = from your current folder

PRACTICE ENVIRONMENT using PowerShell

You will create a root folder called:
C:\TerminalPractice



```
TerminalPractice
│
├── WebProject
│     ├── index.html
│     ├── about.html
│     ├── app.js
│     └── styles
│            └── style.css
│
├── Logs
│     ├── system.log
│     └── network.log
│
├── Security
│     ├── passwords.txt
│     └── hashes
│            ├── md5.txt
│            └── sha256.txt
│
└── Network
      ├── ip.txt
      ├── dns.txt
      └── ports
             ├── open.txt
             └── closed.txt
```

Create the Entire Structure

Create a root directory

- ➔ ***mkdir C:\TerminalPractice***
- ➔ ***cd C:\TerminalPractice***

Create subfolders

- ➔ *mkdir WebProject*
- ➔ *mkdir Logs*
- ➔ *mkdir Security*
- ➔ *mkdir Network*

Now create nested folders:

- ➔ *mkdir WebProject\styles*
- ➔ *mkdir Security\hashes*
- ➔ *mkdir Network\ports*

Create files (PowerShell)

- ➔ *New-Item WebProject\index.html*
- ➔ *New-Item WebProject\about.html*
- ➔ *New-Item WebProject\app.js*
- ➔ *New-Item WebProject\styles\style.css*
- ➔
- ➔ *New-Item Logs\system.log*
- ➔ *New-Item Logs\network.log*
- ➔
- ➔ *New-Item Security\passwords.txt*
- ➔ *New-Item Security\hashes\md5.txt*
- ➔ *New-Item Security\hashes\sha256.txt*
- ➔
- ➔ *New-Item Network\ip.txt*
- ➔ *New-Item Network\dns.txt*
- ➔ *New-Item Network\ports\open.txt*
- ➔ *New-Item Network\ports\closed.txt*

PRACTICE TASKS

List everything:

➔ `ls -Recurse`

Move around:

➔ `cd WebProject`

➔ `cd ..`

➔ `cd Security\hashes`

➔ `cd \`

➔ `cd C:\TerminalPractice`

Copy files:

Copy *index.html* to logos:

➔ `Copy-Item WebProject\index.html Logs\index_backup.html`

Move files

Move *network.log* to Security

➔ `Move-Item Logs\network.log Security`

Rename

Rename app.js -> main.js

➔ `Rename-Item WebProject\app.js main.js`

Delete unwanted files

Delete *closed.txt*

➔ `Remove-Item Network\ports\closed.txt`

Show file contents

➔ `Get-Content Security\passwords.txt`

Write your IP inside *ip.txt*

➔ `ipconfig > Network\ip.txt`

DNS lookup:

➔ `nslookup google.com > Network\dns.txt`

List open ports:

➔ *netstat -ano > Network\ports\open.txt*

List users:

➔ *net user > Security\users.txt*

List firewall rules:

➔ *Get-NetFirewallRule > Security\firewall.txt*

Scan DNS cache:

➔ *ipconfig /displaydns > Logs\dns_cache.txt*

System Information test

➔ *systeminfo > Logs\system_details.txt*

Create a backup of the whole project

➔