

## VCS Task

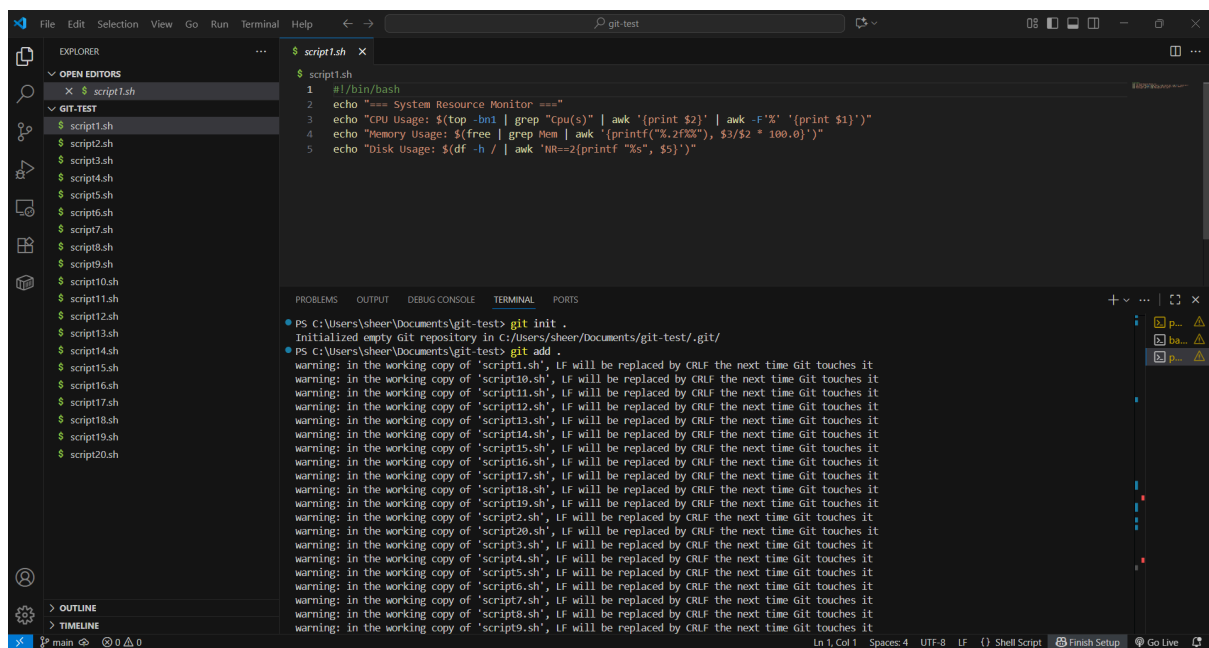
### Task Description:

Establish a new directory, populate it with script files, initiate an empty repository on GitHub, convert the local directory into a Git repository, and link it to GitHub for pushing the code into the repository. Perform merge, rebase, stash commands in following github repo.

### Techstacks needs to be used :

- Shell (AWS, WSL, Vbox)
- Git (Gitbash)

### Setting up a GIT and Github



```
File Edit Selection View Go Run Terminal Help
git-test

EXPLORER
OPEN EDITORS
X $ script1.sh
GIT-TEST
$ script1.sh
$ script2.sh
$ script3.sh
$ script4.sh
$ script5.sh
$ script6.sh
$ script7.sh
$ script8.sh
$ script9.sh
$ script10.sh
$ script11.sh
$ script12.sh
$ script13.sh
$ script14.sh
$ script15.sh
$ script16.sh
$ script17.sh
$ script18.sh
$ script19.sh
$ script20.sh

$ script1.sh
$ script1.sh
1 #!/bin/bash
2 echo "=== System Resource Monitor ==="
3 echo "CPU Usage: $(top -bn1 | grep "cpu(s)" | awk '{print $2}' | awk -F'%' '{print $1}')"
4 echo "Memory Usage: $(free | grep Mem | awk '{printf("%2f%%"), $3/$2 * 100.0}')"
5 echo "Disk Usage: $(df -h / | awk 'NR==2{printf "%s", $5}')"

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\sheer\Documents\git-test> git init .
Initialized empty Git repository in C:\Users\sheer\Documents\git-test\.git\
PS C:\Users\sheer\Documents\git-test> git add .
warning: in the working copy of 'script1.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script18.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script11.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script12.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script13.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script14.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script15.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script16.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script17.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script18.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script19.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script20.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script28.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script3.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script4.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script5.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script6.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script7.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script8.sh', LF will be replaced by CRLF the next time git touches it
warning: in the working copy of 'script9.sh', LF will be replaced by CRLF the next time git touches it
```

```
File Edit Selection View Go Run Terminal Help git-test
```

EXPLORER

- OPEN EDITORS
  - script1.sh
- GIT-TEST
  - script1.sh
  - script2.sh
  - script3.sh
  - script4.sh
  - script5.sh
  - script6.sh
  - script7.sh
  - script8.sh
  - script9.sh
  - script10.sh
  - script11.sh
  - script12.sh
  - script13.sh
  - script14.sh
  - script15.sh
  - script16.sh
  - script17.sh
  - script18.sh
  - script19.sh
  - script20.sh

script1.sh

```
1 #!/bin/bash
2 echo "=== System Resource Monitor ==="
3 echo "CPU Usage: $(top -bn1 | grep "Cpu(s)" | awk '{print $2}' | awk -F '%' '{print $1}')"
4 echo "Memory Usage: $(free | grep Mem | awk '{printf("%.2f%%"), $3/$2 * 100.0}')"
5 echo "Disk Usage: $(df -h / | awk 'NR==2{printf "%s", $5}')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sheer\Documents\git-test> git add .
PS C:\Users\sheer\Documents\git-test> git commit -m "my first commit"
[master (root-commit) 3c2ed8d] my first commit
20 files changed, 137 insertions(+)
create mode 100644 script1.sh
create mode 100644 script10.sh
create mode 100644 script11.sh
create mode 100644 script12.sh
create mode 100644 script13.sh
create mode 100644 script14.sh
create mode 100644 script15.sh
create mode 100644 script16.sh
create mode 100644 script17.sh
create mode 100644 script18.sh
create mode 100644 script19.sh
create mode 100644 script2.sh
create mode 100644 script20.sh
create mode 100644 script3.sh
create mode 100644 script4.sh
create mode 100644 script5.sh
create mode 100644 script6.sh
create mode 100644 script7.sh
create mode 100644 script8.sh
```

```
File Edit Selection View Go Run Terminal Help git-test
```

EXPLORER

- OPEN EDITORS
  - script1.sh
- GIT-TEST
  - script1.sh
  - script2.sh
  - script3.sh
  - script4.sh
  - script5.sh
  - script6.sh
  - script7.sh
  - script8.sh
  - script9.sh
  - script10.sh
  - script11.sh
  - script12.sh
  - script13.sh
  - script14.sh
  - script15.sh
  - script16.sh
  - script17.sh
  - script18.sh
  - script19.sh
  - script20.sh

script1.sh

```
1 #!/bin/bash
2 echo "=== System Resource Monitor ==="
3 echo "CPU Usage: $(top -bn1 | grep "Cpu(s)" | awk '{print $2}' | awk -F '%' '{print $1}')"
4 echo "Memory Usage: $(free | grep Mem | awk '{printf("%.2f%%"), $3/$2 * 100.0}')"
5 echo "Disk Usage: $(df -h / | awk 'NR==2{printf "%s", $5}')
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\sheer\Documents\git-test> git commit -m "my first commit"
create mode 100644 script7.sh
create mode 100644 script8.sh
create mode 100644 script9.sh
PS C:\Users\sheer\Documents\git-test> git remote add origin https://github.com/Sheersh123/VCS.git
PS C:\Users\sheer\Documents\git-test> git status
On branch master
nothing to commit, working tree clean
PS C:\Users\sheer\Documents\git-test> git switch main
fatal: invalid reference: main
PS C:\Users\sheer\Documents\git-test> git branch -m main
PS C:\Users\sheer\Documents\git-test> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\sheer\Documents\git-test> git push .
Everything up-to-date
PS C:\Users\sheer\Documents\git-test> git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Everything up-to-date
```

The screenshot shows a VS Code editor with a terminal window. The terminal displays the output of several scripts (script1.sh to script20.sh) that monitor system resources like CPU, memory, and disk usage. Below the scripts, the terminal shows the output of a `git push origin main` command, indicating a successful push to the main branch.

```
$ script1.sh
1 #!/bin/bash
2 echo "=== System Resource Monitor ==="
3 echo "CPU Usage: $(top -bn1 | grep "Cpu(s)" | awk '{print $2}' | awk -F'%' '{print $1}')"
4 echo "Memory Usage: $(free | grep Mem | awk '{printf("%.2fGB")", $3/$2 * 100.0}')"
5 echo "Disk Usage: $(df -h / | awk 'NR==2{printf "%s", $5}')"

Everything up-to-date
Everything up-to-date
Ps C:\Users\sheer\Documents\git-test> git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Everything up-to-date
Everything up-to-date
Ps C:\Users\sheer\Documents\git-test> git push origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (22/22), 3.63 KiB | 337.00 KiB/s, done.
Total 22 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Sheersh123/VCS.git
 * [new branch]      main -> main
```

## Git Rebase:

The screenshot shows a VS Code editor with a terminal window. The terminal displays the output of a script (script2.sh) that generates a random password. Below the script, the terminal shows the output of a `git rebase` command, indicating a successful rebase of the current branch onto the main branch.

```
$ script2.sh
1 #!/bin/bash
2 length=${1:-12}
3 password=$(openssl rand -base64 32 | head -c $length)
4 echo "Generated Password: $password"

Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<< HEAD (Current Change)
#this is the third commit in main branch
=====
####this is my second commit in dev branch
>>>>>> e925589 (This is the second commit in dev branch) (Incoming Change)
```

The screenshot shows a VS Code window with the Explorer sidebar on the left displaying a directory named 'SAMPLE-SCRIPTS' containing files 'script1.sh' through 'script20.sh'. The 'script2.sh' file is selected. The main area is the 'TERMINAL' tab, showing a series of Git commands and their outputs. The user is in a shell prompt 'sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)'. The commands and outputs are as follows:

```
sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git branch
* main

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git add .
warning: in the working copy of 'Sample-Scripts/script1.sh', LF will be replaced by CRLF the next time Git touches it

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git commit -m "first commit in main branch"
[main 30a8a31] first commit in main branch
1 file changed, 5 insertions(+), 1 deletion(-)

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git checkout dev
Switched to branch 'dev'

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (dev)
$ git add .
warning: in the working copy of 'Sample-Scripts/script2.sh', LF will be replaced by CRLF the next time Git touches it

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (dev)
$ git commit -m "This is the second commit in dev branch"
[dev c925589] This is the second commit in dev branch
1 file changed, 5 insertions(+), 1 deletion(-)

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (dev)
$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
```

The screenshot shows the same VS Code window as the first, but the terminal now shows the continuation of the Git workflow. The user is still in the 'dev' branch. The commands and outputs are as follows:

```
sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (dev)
$ git checkout main

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git add .

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   script2.sh

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git checkout dev
error: Your local changes to the following files would be overwritten by checkout:
       Sample-Scripts/script2.sh
Please commit your changes or stash them before you switch branches.
Aborting

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git commit -m "This is the third commit in main branch"
[main 5ba8592] This is the third commit in main branch
1 file changed, 5 insertions(+), 1 deletion(-)

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (main)
$ git checkout dev
Switched to branch 'dev'

sheen@SheershPC MINGW64 ~\Documents\git-test\Sample-Scripts (dev)
$ git rebase main
Auto-merging Sample-Scripts/script2.sh
CONFLICT (content): Merge conflict in Sample-Scripts/script2.sh
error: could not apply c925589... This is the second commit in dev branch
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
```

The screenshot shows the Visual Studio Code interface with the Explorer, Problems, Output, Debug Console, Terminal, and Ports panels. The Explorer panel on the left shows a file tree for 'SAMPLE-SCRIPTS' with files named script1.sh through script20.sh. The Terminal panel on the right shows the following output:

```
sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git rebase main
hint: "git add/rm <conflicted files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
hint: Disable this message with "git config set advice.mergeConflict false"
Could not apply c925589... # This is the second commit in dev branch

sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev|REBASE 2/2)
• $ git status
interactive rebase in progress; onto 5ba8592
Last commands done (2 commands done):
  pick 3f60767 # Branched-changed-file-changed
  pick c925589 # This is the second commit in dev branch
No commands remaining.
You are currently rebasing branch 'dev' on '5ba8592'.
(fix conflicts and then run "git rebase --continue")
(use "git rebase --skip" to skip this patch)
(use "git rebase --abort" to check out the original branch)

Unmerged paths:
  (use "git restore --staged <file>..." to unstage)
  (use "git add <file>..." to mark resolution)
        both modified:   script2.sh

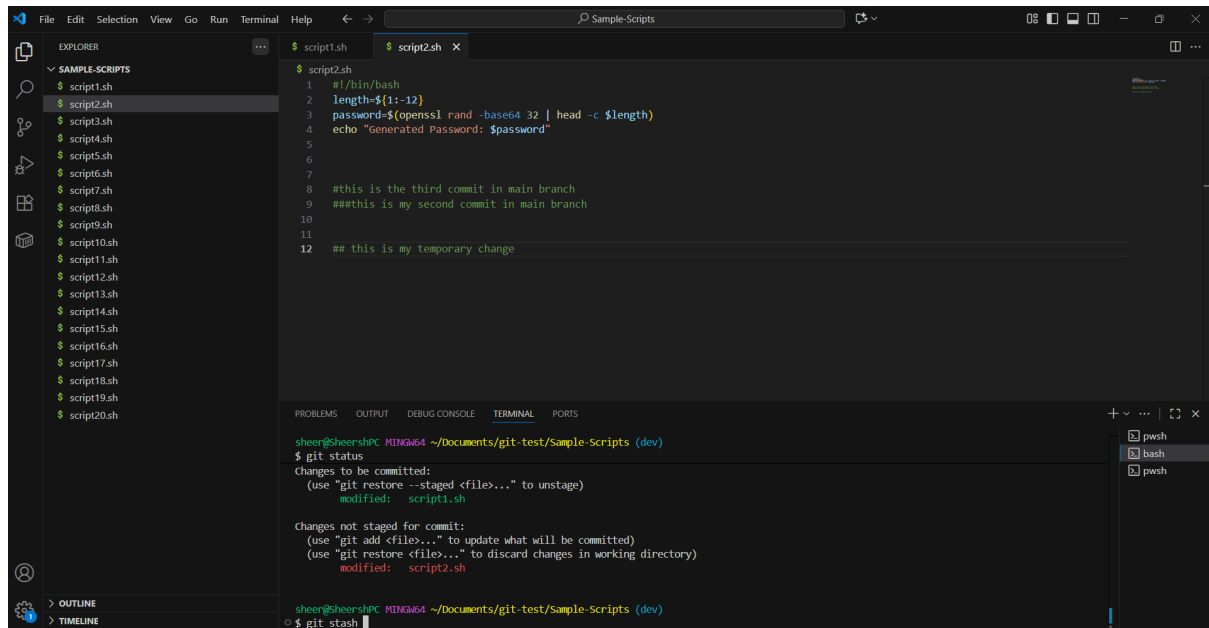
no changes added to commit (use "git add" and/or "git commit -a")

sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev|REBASE 2/2)
$
```

```
sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev|REBASE 2/2)
$ git add .

sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev|REBASE 2/2)
• $ git rebase --continue
[detached HEAD e099a49] This is the second commit in dev branch
 1 file changed, 5 insertions(+), 1 deletion(-)
Successfully rebased and updated refs/heads/dev.
```

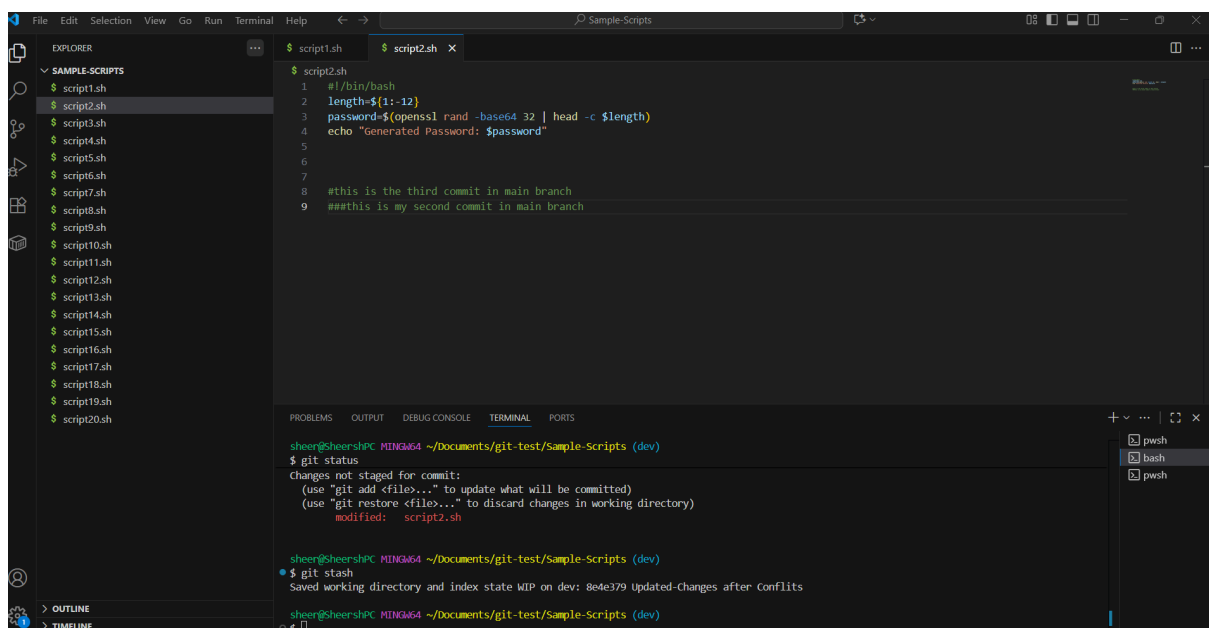
## Git Stash :



```
sheen@sheershpc MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git status
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   script1.sh

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script2.sh

$ git stash
```



```
sheen@sheershpc MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git status
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script2.sh

$ git stash
Saved working directory and index state WIP on dev: 8e4e379 Updated-Changes after Conflicts

sheen@sheershpc MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
```

## Git Stash Pop:

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a list of files named script1.sh through script20.sh. The main editor window shows the content of script2.sh, which includes a shebang, a length variable, a password generation command, an echo statement, and several commit messages. The bottom panel shows the 'TERMINAL' tab with the output of the 'git status' command, indicating changes to script1.sh and script2.sh. The terminal also shows the 'git stash' command being entered.

```
$ script2.sh
1 #!/bin/bash
2 length=${1:-12}
3 password=$(openssl rand -base64 32 | head -c $length)
4 echo "Generated Password: $password"
5
6
7
8 #this is the third commit in main branch
9 ##this is my second commit in main branch
10
11
12 ## this is my temporary change
```

```
sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git status
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   script1.sh

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script2.sh

sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git stash
```

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a list of files named script1.sh through script20.sh. The main editor window shows the content of script2.sh, which includes a shebang, a length variable, a password generation command, an echo statement, and several commit messages. The bottom panel shows the 'TERMINAL' tab with the output of the 'git stash pop' command, indicating that the changes to script1.sh and script2.sh have been restored to the working directory. The terminal also shows the 'git add' command being entered.

```
$ script2.sh
1 #!/bin/bash
2 length=${1:-12}
3 password=$(openssl rand -base64 32 | head -c $length)
4 echo "Generated Password: $password"
5
6
7
8 #this is the third commit in main branch
9 ##this is my second commit in main branch
10
11
12 ## this is my temporary change
```

```
sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git stash pop
On branch dev
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   script1.sh
        modified:   script2.sh

no changes added to commit (use "git add" and/or "git commit -a")
Dropped refs/stash@{0} (1c988802037086fee2sec7c2be1d50201e1a9)

sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (dev)
$ git add
```

## Git merge:

The screenshot shows the VS Code terminal with the output of the 'git merge dev' command. The output indicates that the merge was made by the 'ort' strategy and that 1 file was changed, with 2 insertions and 1 deletion.

```
sheer@SheershPC MINGW64 ~/Documents/git-test/Sample-Scripts (main)
$ git merge dev
Merge made by the 'ort' strategy.
 Sample-Scripts/script2.sh | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

