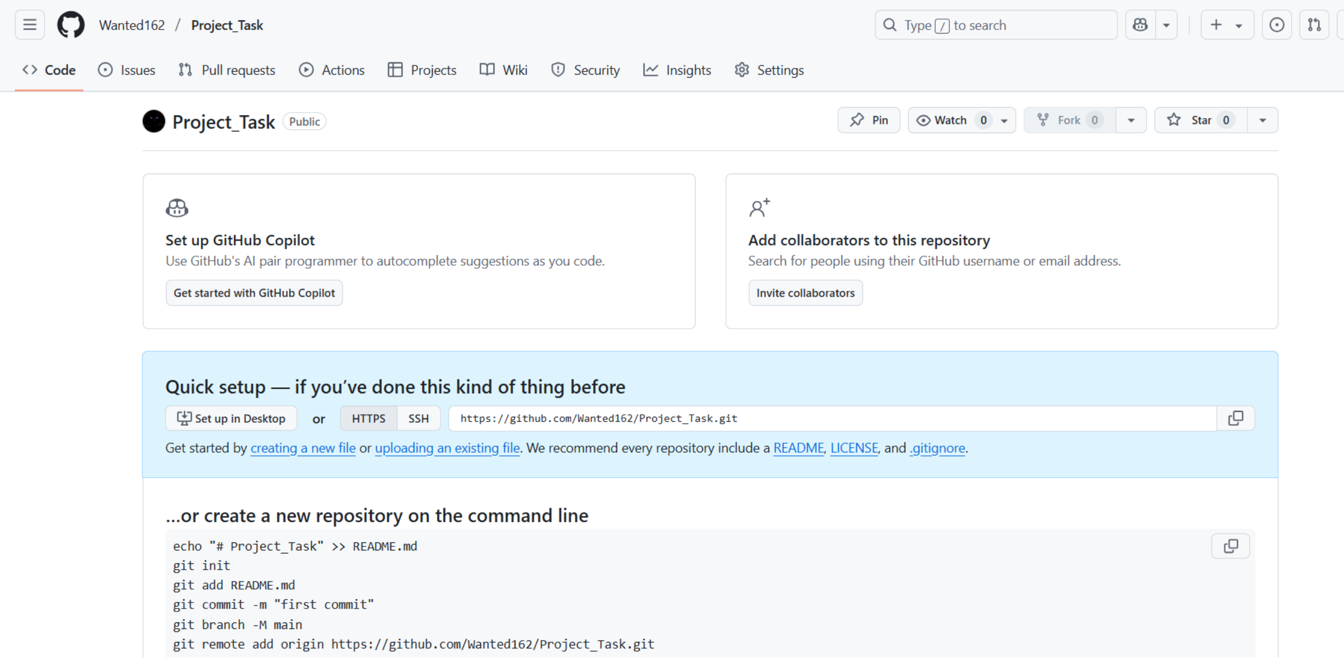
VCS Task

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.

- ➔ Step1: Open a GitHub Repository



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository

Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

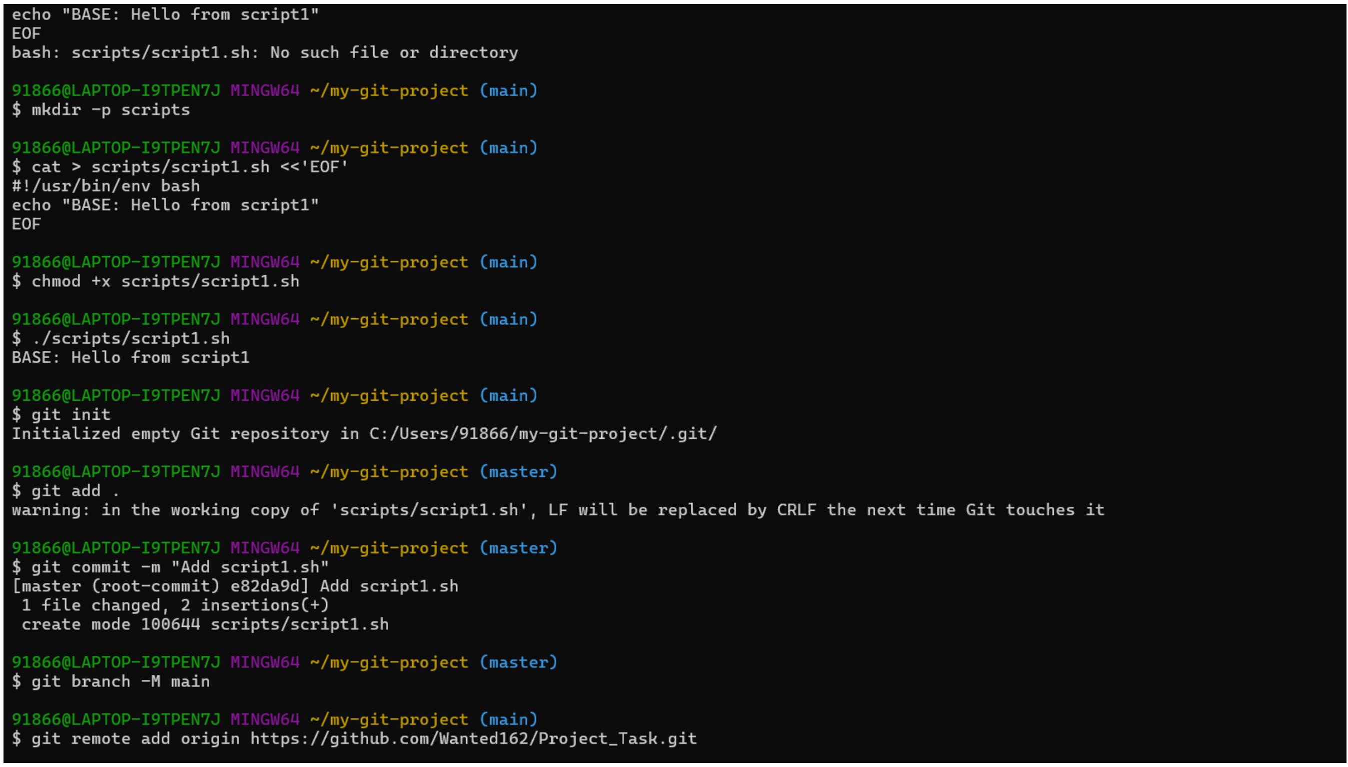
https://github.com/Wanted162/Project_Task.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

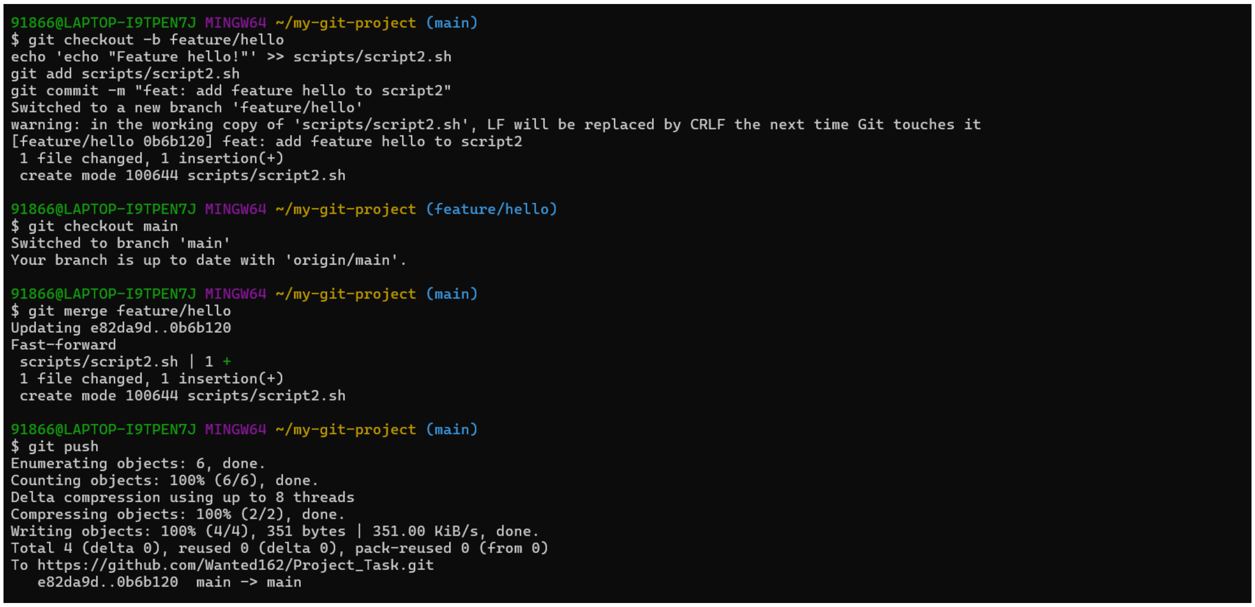
```
echo "# Project_Task" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Wanted162/Project_Task.git
```

- ➔ Step2: In your GitBash open up a directory and load scripts and link that directory to the GitHub Repository that we just created.



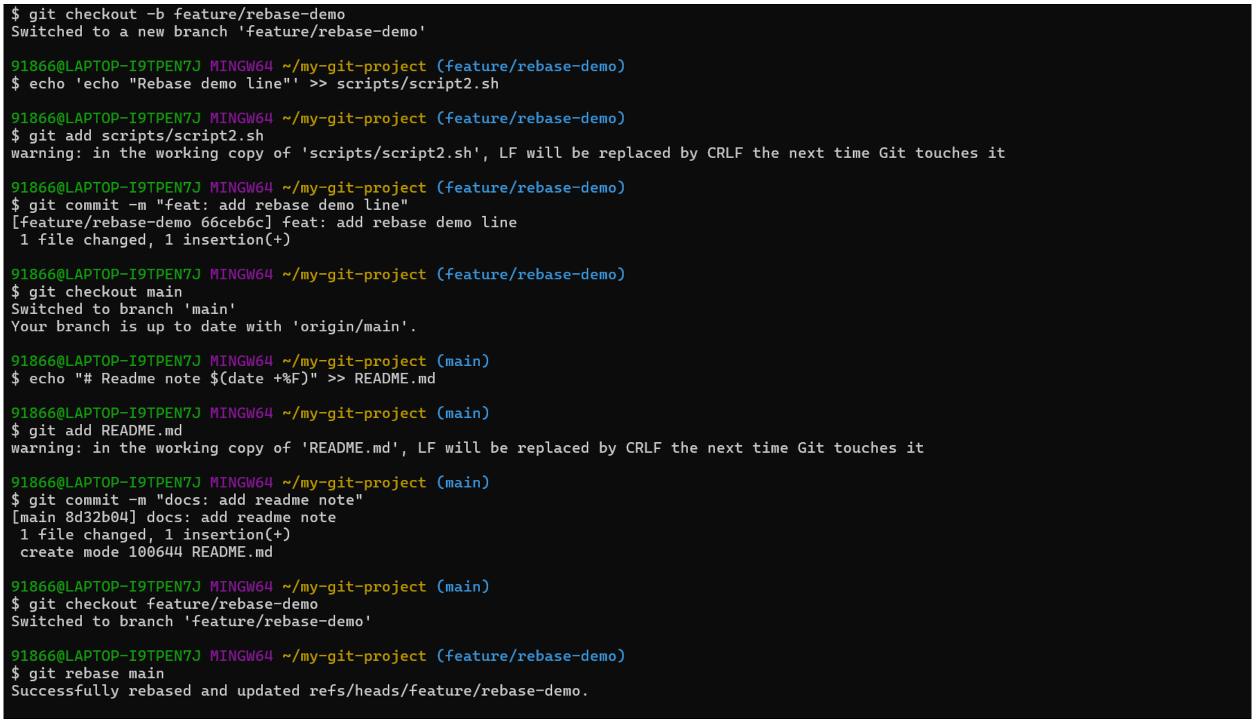
```
echo "BASE: Hello from script1"
EOF
bash: scripts/script1.sh: No such file or directory

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ mkdir -p scripts

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ cat > scripts/script1.sh <<'EOF'
#!/usr/bin/env bash
echo "BASE: Hello from script1"
EOF

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ chmod +x scripts/script1.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ ./scripts/script1.sh
BASE: Hello from script1

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git init
Initialized empty Git repository in C:/Users/91866/my-git-project/.git/

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (master)
$ git add .
warning: in the working copy of 'scripts/script1.sh', LF will be replaced by CRLF the next time Git touches it

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (master)
$ git commit -m "Add script1.sh"
[master (root-commit) e82da9d] Add script1.sh
 1 file changed, 2 insertions(+)
 create mode 100644 scripts/script1.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (master)
$ git branch -M main

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git remote add origin https://github.com/Wanted162/Project_Task.git
```

➔ Merging the script files



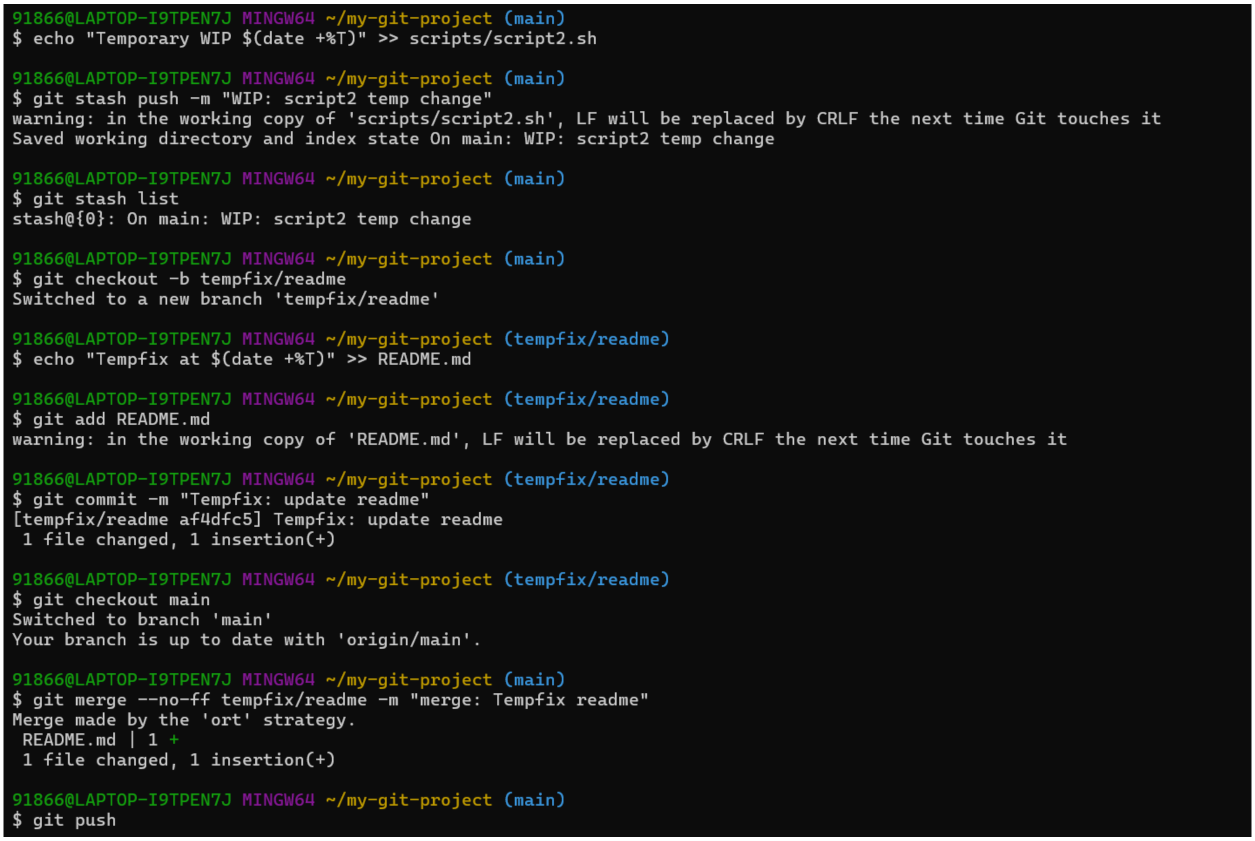
```
91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git checkout -b feature/hello
echo 'echo "Feature hello!"' >> scripts/script2.sh
git add scripts/script2.sh
git commit -m "feat: add feature hello to script2"
Switched to a new branch 'feature/hello'
warning: in the working copy of 'scripts/script2.sh', LF will be replaced by CRLF the next time Git touches it
[feature/hello 0b6b120] feat: add feature hello to script2
 1 file changed, 1 insertion(+)
 create mode 100644 scripts/script2.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/hello)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git merge feature/hello
Updating e82da9d..0b6b120
Fast-forward
 scripts/script2.sh | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 scripts/script2.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 351 bytes | 351.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Wanted162/Project_Task.git
   e82da9d..0b6b120  main -> main
```

➔ Rebase – replays your branch commit on the top list

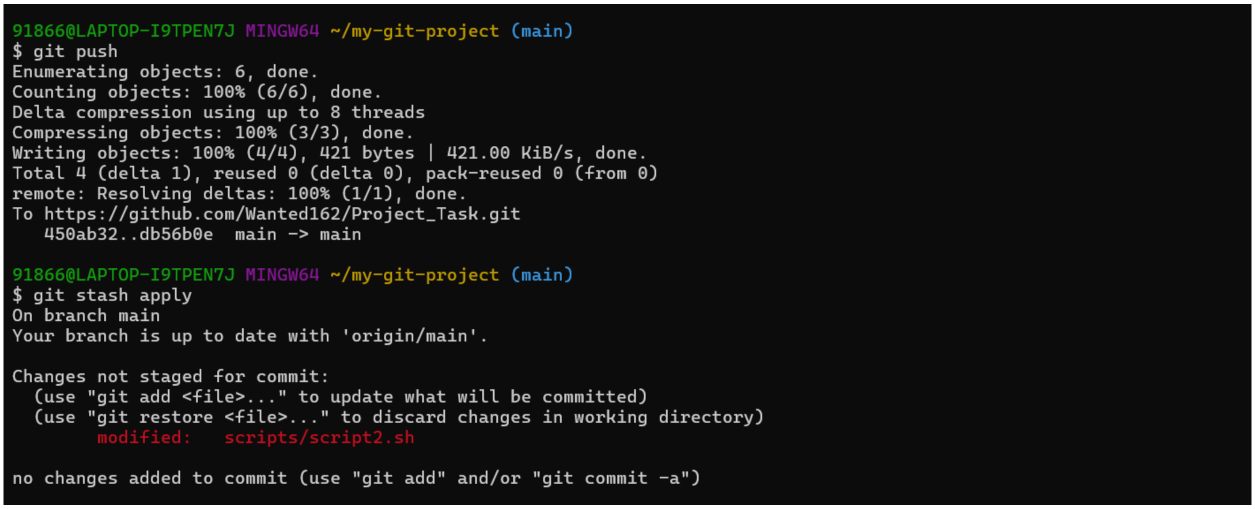


```
$ git checkout -b feature/rebase-demo
Switched to a new branch 'feature/rebase-demo'

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/rebase-demo)
$ echo 'echo "Rebase demo line"' >> scripts/script2.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/rebase-demo)
$ git add scripts/script2.sh
warning: in the working copy of 'scripts/script2.sh', LF will be replaced by CRLF the next time Git touches it

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/rebase-demo)
$ git commit -m "feat: add rebase demo line"
[feature/rebase-demo 66ceb6c] feat: add rebase demo line
 1 file changed, 1 insertion(+)

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/rebase-demo)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ echo "# Readme note $(date +%F)" >> README.md

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git commit -m "docs: add readme note"
[main 8d32b04] docs: add readme note
 1 file changed, 1 insertion(+)
 create mode 100644 README.md

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git checkout feature/rebase-demo
Switched to branch 'feature/rebase-demo'

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (feature/rebase-demo)
$ git rebase main
Successfully rebased and updated refs/heads/feature/rebase-demo.
```

➔ Stash – save work in progress without committing



```
91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ echo "Temporary WIP $(date +%T)" >> scripts/script2.sh

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git stash push -m "WIP: script2 temp change"
warning: in the working copy of 'scripts/script2.sh', LF will be replaced by CRLF the next time Git touches it
Saved working directory and index state On main: WIP: script2 temp change

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git stash list
stash@{0}: On main: WIP: script2 temp change

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git checkout -b tempfix/readme
Switched to a new branch 'tempfix/readme'

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (tempfix/readme)
$ echo "Tempfix at $(date +%T)" >> README.md

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (tempfix/readme)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (tempfix/readme)
$ git commit -m "Tempfix: update readme"
[tempfix/readme af4dfc5] Tempfix: update readme
 1 file changed, 1 insertion(+)

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (tempfix/readme)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git merge --no-ff tempfix/readme -m "merge: Tempfix readme"
Merge made by the 'ort' strategy.
 README.md | 1 +
 1 file changed, 1 insertion(+)

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git push
```



```
91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 421 bytes | 421.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/Wanted162/Project_Task.git
   450ab32..db56b0e  main -> main

91866@LAPTOP-I9TPEN7J MINGW64 ~/my-git-project (main)
$ git stash apply
On branch main
Your branch is up to date with 'origin/main'.

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:   scripts/script2.sh

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