EXPERIMENT No. 01: Setting Up and Basic Commands

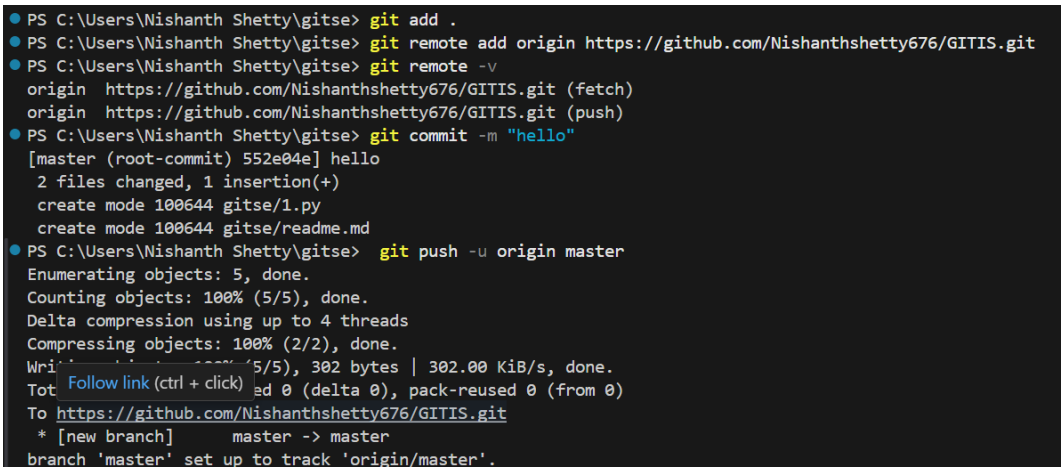
AIM:
Initialize a Git repository, create files, stage and commit changes, link with a remote repository, and push the changes.

COMMANDS:

1. git init
2. git add .
3. git commit -m "Initial"
4. git remote add origin <repo-link>
5. git remote -v
6. git push -u origin main

PROCEDURE AND RESULTS:

- Create a new repository on GitHub (just give the repo name and create it).
- Copy the repo link.
- Create a new folder locally and open it in VS Code.
- Inside the folder, create any file and add some content.
- Open the terminal and initialize Git using git init.
- Stage the file using git add ..
- Commit the changes using git commit -m "Initial".
- Add the remote origin using git remote add origin <repo-link>.
- Confirm the remote using git remote -v.
- Push the code to GitHub using git push -u origin main.



EXPERIMENT No. 02: Creating and Managing Branches

AIM:
Create and manage branches using Git. Perform merge operations and use stash to temporarily save and apply changes.

COMMANDS:

Branch Creation and Merge:

1. git branch feature-branch
2. git checkout feature-branch
3. git add .
4. git commit -m "Initial"
5. git checkout main
6. git merge feature-branch
7. git push

Stash Operation:

1. git stash
2. git checkout main
3. git stash apply

PROCEDURE AND RESULTS:

Branch Creation and Merge:

- Create a new branch using git branch feature-branch.
- Switch to the new branch with git checkout feature-branch.
- Make changes (e.g., create a file), then stage and commit using git add . and git commit -m "Initial".
- Switch back to the main branch using git checkout main.
- Merge the feature branch using git merge feature-branch.
- Push the updated main branch to GitHub using git push.

Stash Operation:

- While on the feature-branch, make some changes without committing.
- Use git stash to temporarily save the changes.
- Switch to the main branch using git checkout main.

- After performing other operations or creating files, run git stash apply to re-apply the stashed changes to the working directory.

```
PS C:\Users\Nishanth Shetty\gitse> git branch feature-branch
PS C:\Users\Nishanth Shetty\gitse> git checkout feature-branch
Switched to branch 'feature-branch'
PS C:\Users\Nishanth Shetty\gitse> git branch -v
* feature-branch 552e04e hello
  master         552e04e hello
PS C:\Users\Nishanth Shetty\gitse> git add .
PS C:\Users\Nishanth Shetty\gitse> git commit -m "csc"
[feature-branch 774bf5f] csc
 1 file changed, 1 insertion(+)
 create mode 100644 gitse/2.py
PS C:\Users\Nishanth Shetty\gitse> git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
PS C:\Users\Nishanth Shetty\gitse> git branch -v
  feature-branch 774bf5f csc
* master         552e04e hello
PS C:\Users\Nishanth Shetty\gitse> git merge feature-branch
Updating 552e04e..774bf5f
PS C:\Users\Nishanth Shetty\gitse> git merge feature-branch
Updating 552e04e..774bf5f
Updating 552e04e..774bf5f
Fast-forward
 gitse/2.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 gitse/2.py
```

```
PS C:\Users\Nishanth Shetty\gitse> git checkout feature-branch

M       gitse/1.py
Switched to branch 'feature-branch'
PS C:\Users\Nishanth Shetty\gitse> git branch feature-branch
fatal: a branch named 'feature-branch' already exists
PS C:\Users\Nishanth Shetty\gitse> git branch -v
* feature-branch 774bf5f csc
  master         774bf5f [ahead 1] csc
PS C:\Users\Nishanth Shetty\gitse> git add .
PS C:\Users\Nishanth Shetty\gitse> git stash
Saved working directory and index state WIP on feature-branch: 774bf5f csc
PS C:\Users\Nishanth Shetty\gitse> git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
PS C:\Users\Nishanth Shetty\gitse> git stash apply
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   testStash.py
```

EXPERIMENT No. 03: Collaboration and Remote Repositories

AIM:
Clone remote repositories and manage updates with fetch, rebase, and merge.

COMMANDS:

1. git clone <link>
2. git fetch
3. git rebase origin/main
4. git merge feature-branch --no-ff -m "Merge"

PROCEDURE AND RESULTS:

- Clone an existing GitHub repository using git clone <link>.
- Fetch new updates from the remote using git fetch.
- Rebase the local branch using git rebase origin/main.
- Merge the feature branch with a message using git merge feature-branch --no-ff -m "Merge".

```
PS C:\Users\Nishanth Shetty\gitse> cd cloneFolder
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git clone https://github.com/Nishanthshetty676/GITIS.git
Cloning into 'GITIS'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 5 (delta 0), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git fetch
```

```
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport\cloneFolder> git rebase origin/main
Current branch main is up to date.
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport\cloneFolder> git merge feature-branch --no-ff -m "Merge"
Already up to date.
```

EXPERIMENT No. 04: Git Tags and Releases

AIM:
Use Git tags to mark versions and create GitHub releases.

COMMANDS:

1. git tag v1.0

PROCEDURE AND RESULTS:

- Create a local tag using git tag v1.0.
- Go to the GitHub repository.
- Click on **Releases > Create a new release**.
- Set the tag version as v1.0 and give a release title.

```
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git tag v1.0
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git show v1.0
commit 774bf5f71849f08da2ebc337150b134bf5d61b8f (HEAD -> master, tag: v1.0, feature-branch)
Author: Nishanthshetty676 <shettyynishanth676@gmail.com>
Date:   Tue May 20 18:34:18 2025 +0530

    csc

diff --git a/gitse/2.py b/gitse/2.py
new file mode 100644
index 0000000..db8980d
--- /dev/null
+++ b/gitse/2.py
@@ -0,0 +1 @@
+print("Hey")
\ No newline at end of file
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git checkout feature-branch
M       gitse/1.py
M       gitse/2.py
Switched to branch 'feature-branch'
```

EXPERIMENT No. 05: Advanced Git Operations – Cherry Pick

AIM:
Use cherry-pick to apply specific commits from one branch to another.

COMMANDS:

1. git log --oneline
2. git cherry-pick <commit-id>

PROCEDURE AND RESULTS:

- On the feature-branch, create a file and commit changes.
- Use git log --oneline to get the commit ID.
- Switch to the main branch using git checkout main.
- Use git cherry-pick <commit-id> to apply a specific commit.

```
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> git checkout feature-branch
Switched to branch 'feature-branch'
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> git add .
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> git commit -m "Feature"
[feature-branch 9c0e5e6] Feature
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 advTest.py
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> git log --oneline
9c0e5e6 (HEAD -> feature-branch) Feature
05c3007 (origin/main, origin/HEAD) Initial
61648b4 Initial
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> 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)
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport> git cherry-pick 9c0e5e6
[main 092b58e] Feature
 Date: Tue May 20 07:42:00 2025 +0530
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 advTest.py
PS C:\Users\LENOVO\OneDrive\Desktop\gitReport>
```

EXPERIMENT No. 06: Analysing and Changing Git History

AIM:
Analyze commit logs and revert specific changes.

COMMANDS:

1. git show <commit-id>
2. git log --author="Varsha" --after="YYYY-MM-DD" --before="YYYY-MM-DD"
3. git log -n 5
4. git revert <commit-id>

PROCEDURE AND RESULTS:

- Use git show <commit-id> to view commit details.
- Use author and date filters to check commit history.
- View last five commits using git log -n 5.
- Revert a specific commit using git revert <commit-id>.

```
PS C:\Users\Nishanth Shetty\gitse\cloneFolder> git show 9a75ce3
commit 9a75ce36e6d9a84260f914f130ac38cd5b2b6964 (feature-branch)
Author: Nishanthshetty676 <shettyynishanth676@gmail.com>
Date:   Tue May 20 20:20:41 2025 +0530

    Feature

diff --git a/gitse/cloneFolder/GITIS b/gitse/cloneFolder/GITIS
new file mode 160000
index 0000000..552e04e
--- /dev/null
+++ b/gitse/cloneFolder/GITIS
@@ -0,0 +1 @@
+Subproject commit 552e04e1fbe7170145538ed38979efadf1761acc
PS C:\Users\Nishanth Shetty\gitse\cloneFolder>
```