

Git Project (2.5 Hours)

Objective:

You will work on a project involving a simple website. You will learn and practice various Git concepts including branching, merging, handling merge conflicts, rebasing, pulling, versioning, and rolling back changes. This project is designed to be completed in 1 hour.

Project Setup (15 minutes)

1. **Install Git:** Ensure Git is installed on your system. Verify with `git --version`.

```
einfochips@AHMLPT1108:~/Desktop/OneDrive$ git --version
git version 2.25.1
einfochips@AHMLPT1108:~/Desktop/OneDrive$
```

Set Up Git: Configure your Git username and email:

```
git config --global user.name "Your Name"
git config --global user.email your.email@example.com
```

```
einfochips@AHMLPT1108:~/Desktop/OneDrive$ git config --global user.name "Maaz"
einfochips@AHMLPT1108:~/Desktop/OneDrive$ git config --global user.email "maazptl240602@gmail.com"
einfochips@AHMLPT1108:~/Desktop/OneDrive$ cat .gitconfig
cat: .gitconfig: No such file or directory
einfochips@AHMLPT1108:~/Desktop/OneDrive$ cd
einfochips@AHMLPT1108:~$ cat .gitconfig
[user]
    name = Maaz
    email = maazptl240602@gmail.com
[filter "lfs"]
    clean = git-lfs clean -- %f
    smudge = git-lfs smudge -- %f
    process = git-lfs filter-process
    required = true
[http]
    postBuffer = 524288000
einfochips@AHMLPT1108:~$
```

2. **Create a GitHub Repository:**

- Go to GitHub and create a new repository named `website-project`.

Clone the repository to your local machine:

```
git clone https://github.com/your-username/website-project.git
```

```
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2$ git clone https://github.com/maazpatel24/website-project.git
Cloning into 'website-project'...
Username for 'https://github.com': maazpatel24
Password for 'https://maazpatel24@github.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 873 bytes | 291.00 KiB/s, done.
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2$ ls
website-project
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2$
```

3. Initialize the Project:

Navigate to the project directory:

```
cd website-project
```

-

Create initial project structure:

```
mkdir src
touch src/index.html
echo "<!DOCTYPE html><html><head><title>My
Website</title></head><body><h1>Welcome to my
website!</h1></body></html>" > src/index.html
```

```
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ mkdir src
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ touch src/index.html
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ tree
.
|-- README.md
'-- src
    |-- index.html

1 directory, 2 files

einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ echo '<!DOCTYPE html><html><head><title>My Website</title></head><body><h1>Welcome to my website!</h1></body></html>' > src/index.html
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ cat src/index.html
<!DOCTYPE html><html><head><title>My Website</title></head><body><h1>Welcome to my website!</h1></body></html>
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$
```

Commit and push the initial project structure:

```
git add .
git commit -m "Initial commit: Added project structure and index.html"
git push origin main
```

```

einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ ls
README.md  src
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      src/

nothing added to commit but untracked files present (use "git add" to track)
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git add .
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git status
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:   src/index.html

einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git commit -m "Initial commit: Added p
project structure and index.html"
[main 99a76e2] Initial commit: Added project structure and index.html
 1 file changed, 1 insertion(+)
 create mode 100644 src/index.html
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git push origin main
Username for 'https://github.com': maazpatel24
Password for 'https://maazpatel24@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 438 bytes | 438.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/maazpatel24/website-project.git
 2094145..99a76e2  main -> main
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$

```

Exercise 1: Branching and Basic Operations (10 minutes)

Create a New Branch:

```
git checkout -b feature/add-about-page
```

```

einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git checkout -b feature/add-about-page
Switched to a new branch 'feature/add-about-page'

einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git branch
* feature/add-about-page
main

```

4. Add a New Page:

Create `about.html`:

```

touch src/about.html
echo "<!DOCTYPE html><html><head><title>About
Us</title></head><body><h1>About Us</h1></body></html>" >
src/about.html

```

```
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ touch src/about.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ echo '<!DOCTYPE html><html><head><title>About Us</title></head><body><h1>About Us</h1></body></html>' > src/about.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ cat src/about.html
<!DOCTYPE html><html><head><title>About Us</title></head><body><h1>About Us</h1></body></html>
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$
```

Commit and push changes:

```
git add src/about.html
```

```
git commit -m "Added about page"
```

```
git push origin feature/add-about-page
```

```
<!DOCTYPE html><html><head><title>About Us</title></head><body><h1>About Us</h1></body></html>
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git add src/about.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git commit -m "Added about page"
[feature/add-about-page 5d2b220] Added about page
 1 file changed, 1 insertion(+)
 create mode 100644 src/about.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git push origin
feature/add-about-page  main          origin/main
HEAD                   origin/HEAD
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git push origin feature/add-about-page

Username for 'https://github.com': maazpatel24
Password for 'https://maazpatel24@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 418 bytes | 209.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'feature/add-about-page' on GitHub by visiting:
remote:   https://github.com/maazpatel24/website-project/pull/new/feature/add-about-page
remote:
To https://github.com/maazpatel24/website-project.git
 * [new branch]   feature/add-about-page -> feature/add-about-page
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$
```

Exercise 2: Merging and Handling Merge Conflicts (15 minutes)

Create Another Branch:

```
git checkout main
```

```
git checkout -b feature/update-homepage
```

```
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git branch
  feature/add-about-page
* main
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git checkout -b feature/update-homepage
Switched to a new branch 'feature/update-homepage'
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git branch
  feature/add-about-page
* feature/update-homepage
  main
```

5. Update the Homepage:

Modify `index.html`:

```
echo "<p>Updated homepage content</p>" >> src/index.html
```

-

Commit and push changes:

```
git add src/index.html
git commit -m "Updated homepage content"
git push origin feature/update-homepage
```

```
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ echo '<p>Updated homepage content</p>'
>> src/index.html
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git add src/index.html
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git commit -m "Updated homepage conten
t"
[feature/update-homepage f7939c5] Updated homepage content
 1 file changed, 1 insertion(+)
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git push origin feature/update-homepag
e
Username for 'https://github.com': maazpatel24
Password for 'https://maazpatel24@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 387 bytes | 387.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'feature/update-homepage' on GitHub by visiting:
remote:   https://github.com/maazpatel24/website-project/pull/new/feature/update-homepage
remote:
To https://github.com/maazpatel24/website-project.git
 * [new branch]      feature/update-homepage -> feature/update-homepage
einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$
```

6. Create a Merge Conflict:

Modify `index.html` on the `feature/add-about-page` branch:

```
git checkout feature/add-about-page
echo "<p>Conflict content</p>" >> src/index.html
git add src/index.html
git commit -m "Added conflicting content to homepage"
git push origin feature/add-about-page
```

```
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git checkout feature/add-about-page
Switched to branch 'feature/add-about-page'
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git branch
* feature/add-about-page
  feature/update-homepage
  main
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ echo '<p>Conflict content</p>' >> src/
index.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ cat src/index.html
<!DOCTYPE html><html><head><title>My Website</title></head><body><h1>Welcome to my website!</h1></bod
y></html>
<p>Conflict content</p>
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git add src/index.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git commit -m "Added conflicting conte
nt to homepage"
[feature/add-about-page 8bbd237] Added conflicting content to homepage
 1 file changed, 1 insertion(+)
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git push origin feature/add-about-page

Username for 'https://github.com': maazpatel24
Password for 'https://maazpatel24@github.com':
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 421 bytes | 421.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/maazpatel24/website-project.git
   5d2b220..8bbd237  feature/add-about-page -> feature/add-about-page
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$
```

7. Merge and Resolve Conflict:

Attempt to merge `feature/add-about-page` into `main`:

```
git checkout main
git merge feature/add-about-page
```

```

9020220710bb0237 feature/add-about-page -> feature/add-about-page
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git branch
  feature/add-about-page
  feature/update-homepage
* main
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$ git merge feature/add-about-page
Updating 99a76e2..8bbd237
Fast-forward
 src/about.html | 1 +
 src/index.html | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 src/about.html
einfochips@AHMLPT1108:~/DevOps_Training/Day-2/website-project$

```

Resolve the conflict in `src/index.html`, then:

```

git add src/index.html
git commit -m "Resolved merge conflict in homepage"
git push origin main

```

Exercise 3: Rebasing (10 minutes)

8. Rebase a Branch:

Rebase `feature/update-homepage` onto `main`:

```

git checkout feature/update-homepage
git rebase main

```

-
- Resolve any conflicts that arise during rebase.

Push the Rebased Branch:

```

git push -f origin feature/update-homepage

```

Exercise 4: Pulling and Collaboration (10 minutes)

9. Pull Changes from Remote:

Ensure the `main` branch is up-to-date:

```
git checkout main
git pull origin main
```

-

10. Simulate a Collaborator's Change:

- Make a change on GitHub directly (e.g., edit `index.html`).

11. Pull Collaborator's Changes:

Pull the changes made by the collaborator:

```
git pull origin main
```

```
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git pull
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 1.02 KiB | 1.02 MiB/s, done.
From https://github.com/maazpatel24/website-project
   2478305..158fdbb  main       -> origin/main
Updating 2478305..158fdbb
Fast-forward
 src/index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-p
○ einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$
```

Exercise 5: Versioning and Rollback (15 minutes)

12. Tagging a Version:

Tag the current commit as `v1.0`:

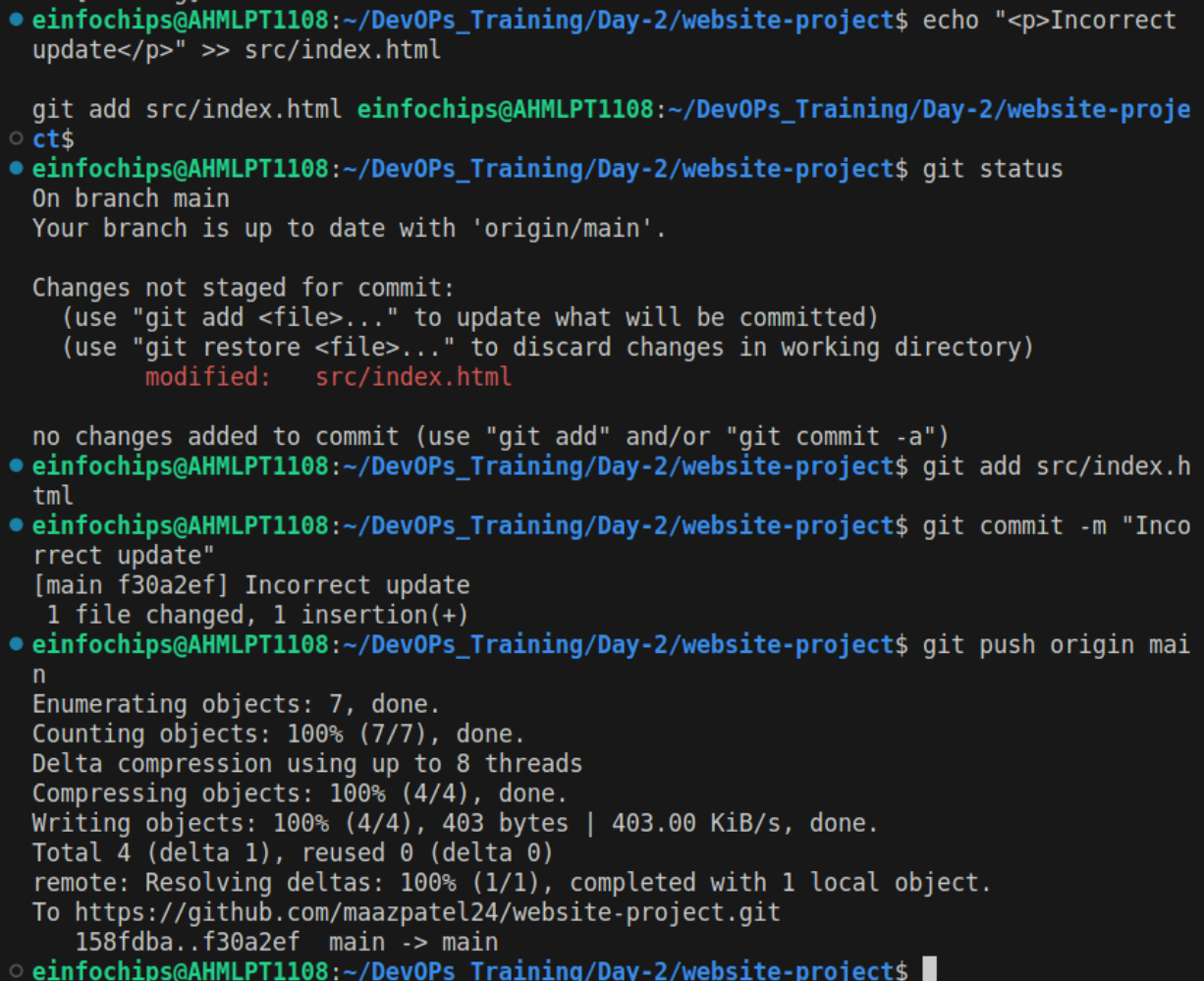
```
git tag -a v1.0 -m "Version 1.0: Initial release"
git push origin v1.0
```

```
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git push origin v1.0
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 174 bytes | 174.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/maazpatel24/website-project.git
 * [new tag]         v1.0 -> v1.0
○ einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$
```


13. Make a Change that Needs Reversion:

Modify `index.html`:

```
echo "<p>Incorrect update</p>" >> src/index.html
git add src/index.html
git commit -m "Incorrect update"
git push origin main
```



```
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ echo "<p>Incorrect
update</p>" >> src/index.html

git add src/index.html einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-proje
ct$
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git status
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:   src/index.html

no changes added to commit (use "git add" and/or "git commit -a")
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git add src/index.h
tml
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git commit -m "Inco
rrect update"
[main f30a2ef] Incorrect update
1 file changed, 1 insertion(+)
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git push origin mai
n
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 403 bytes | 403.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/maazpatel24/website-project.git
158fdbba..f30a2ef  main -> main
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$
```

14. Revert to a Previous Version:

Use `git revert` to undo the last commit:

```
git revert HEAD
git push origin main
```

```

• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git revert HEAD
[main 7e492a4] Revert "Incorrect update"
 1 file changed, 1 deletion(-)
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git push origin mai
n
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 409 bytes | 409.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/maazpatel24/website-project.git
   f30a2ef..7e492a4  main -> main
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$

```

Alternatively, reset to a specific commit (use with caution):

sh

Copy code

```
git reset --hard v1.0
```

```
git push -f origin main
```

Extra Activities (10 minutes)

15. Stashing Changes:

Make some local changes without committing:

```
echo "<p>Uncommitted changes</p>" >> src/index.html
```

Stash the changes:

```
git stash
```

Apply the stashed changes:

```
git stash apply
```

```

• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ echo "<p>Uncommitt
d changes</p>" >> src/index.html
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ git stash
Saved working directory and index state WIP on main: 7e492a4 Revert "Incorrect upd
ate"
• einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ 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:   src/index.html

no changes added to commit (use "git add" and/or "git commit -a")
○ einfochips@AHMLPT1108:~/DevOPs_Training/Day-2/website-project$ █

```

16. Viewing Commit History:

Use `git log` to view commit history:

```
git log --oneline
```

•

17. Cherry-Picking Commits:

Create a new branch and cherry-pick a commit from another branch:

```
git checkout -b feature/cherry-pick
git cherry-pick <commit-hash>
git push origin feature/cherry-pick
```

•

18. Interactive Rebase:

Use interactive rebase to squash commits:

```
git checkout main
git rebase -i HEAD~3
```

Collaborative Blogging Platform

Objective:

You will work on a project to collaboratively develop a simple blogging platform. You will practice various Git concepts including branching, merging, handling merge conflicts, rebasing, pulling, versioning, rolling back changes, stashing, and cherry-picking commits. The project is designed to be completed in 1.5 Hours

19. Create a GitHub Repository:

- Go to GitHub and create a new repository named **blogging-platform**.

Clone the repository to your local machine:

```
git clone https://github.com/your-username/blogging-platform.git
```

-

20. Initialize the Project:

Navigate to the project directory:

```
cd blogging-platform
```

-

Create initial project structure:

```
mkdir src
```

```
touch src/index.html
```

```
echo "<!DOCTYPE html><html><head><title>Blogging
Platform</title></head><body><h1>Welcome to the Blogging
Platform!</h1></body></html>" > src/index.html
```

-

Commit and push the initial project structure:

```
git add .
```

```
git commit -m "Initial commit: Added project structure and index.html"
```

```
git push origin main
```

-

Exercise 1: Branching and Adding Features (20 minutes)

Create a New Branch for Blog Post Feature:

```
git checkout -b feature/add-blog-post
```

21. Add a Blog Post Page:

Create `blog.html`:

```
touch src/blog.html
```

```
echo "<!DOCTYPE html><html><head><title>Blog
Post</title></head><body><h1>My First Blog Post</h1></body></html>" >
src/blog.html
```

-

Commit and push changes:

```
git add src/blog.html
```

```
git commit -m "Added blog post page"
```

```
git push origin feature/add-blog-post
```

-

Exercise 2: Collaborating with Merging and Handling Merge Conflicts (25 minutes)

Create Another Branch for Author Info:

```
git checkout main
```

```
git checkout -b feature/add-author-info
```

22.

23. Add Author Info to Blog Page:

Modify `blog.html`:

```
echo "<p>Author: John Doe</p>" >> src/blog.html
```

-

Commit and push changes:

```
git add src/blog.html
```

```
git commit -m "Added author info to blog post"
```

```
git push origin feature/add-author-info
```

-

24. Create a Merge Conflict:

Modify `blog.html` on the `feature/add-blog-post` branch:

```
git checkout feature/add-blog-post
```

```
echo "<p>Published on: July 10, 2024</p>" >> src/blog.html
```

```
git add src/blog.html
```

```
git commit -m "Added publish date to blog post"
```

```
git push origin feature/add-blog-post
```

-

25. Merge and Resolve Conflict:

Attempt to merge `feature/add-blog-post` into `main`:

```
git checkout main
```

```
git merge feature/add-blog-post
```

-

Resolve the conflict in `src/blog.html`, then:

```
git add src/blog.html
```

```
git commit -m "Resolved merge conflict in blog post"
```

```
git push origin main
```

-

Exercise 3: Rebasing and Feature Enhancement (25 minutes)

26. Rebase a Branch for Comment Feature:

Rebase `feature/add-author-info` onto `main`:

```
git checkout feature/add-author-info
```

```
git rebase main
```

-

- Resolve any conflicts that arise during rebase.

27. Add Comment Section:

Modify `blog.html` to add a comment section:

```
echo "<h2>Comments</h2><p>No comments yet.</p>" >> src/blog.html
```

```
git add src/blog.html
```

```
git commit -m "Added comment section"
```



```
git push origin feature/add-author-info
```

-

Exercise 4: Pulling and Simulating Collaboration (20 minutes)

28. Pull Changes from Remote:

Ensure the `main` branch is up-to-date:

```
git checkout main
```

```
git pull origin main
```

-

29. Simulate a Collaborator's Change:

- Make a change on GitHub directly (e.g., edit `blog.html` to add a new comment).

30. Pull Collaborator's Changes:

Pull the changes made by the collaborator:

```
git pull origin main
```

-

Exercise 5: Versioning and Rollback (30 minutes)

31. Tagging a Version:

Tag the current commit as `v1.0`:

```
git tag -a v1.0 -m "Version 1.0: Initial release"
```

```
git push origin v1.0
```

-

32. Make a Change that Needs Reversion:

Modify `blog.html`:

```
echo "<p>Incorrect comment</p>" >> src/blog.html
```

```
git add src/blog.html
```

```
git commit -m "Incorrect comment update"
```

```
git push origin main
```

-

33. Revert to a Previous Version:

Use `git revert` to undo the last commit:

```
git revert HEAD
```

```
git push origin main
```

-

Alternatively, reset to a specific commit (use with caution):

```
git reset --hard v1.0
```

```
git push -f origin main
```

-

Extra Activities (25 minutes)

34. Stashing Changes:

Make some local changes without committing:

```
echo "<p>Uncommitted changes</p>" >> src/blog.html
```

Stash the changes:

```
git stash
```

Apply the stashed changes:

```
git stash apply
```

35. Viewing Commit History:

Use `git log` to view commit history:

```
git log --oneline
```

36. Cherry-Picking Commits:

Create a new branch and cherry-pick a commit from another branch:

```
git checkout -b feature/cherry-pick
```

```
git cherry-pick <commit-hash>
```

```
git push origin feature/cherry-pick
```

37. Interactive Rebase:

Use interactive rebase to squash commits:

```
git checkout main
```

```
git rebase -i HEAD~3
```