

Comprehensive Workflow with Forced Pushes and Recovery - Task 10

➤ Initializing a repo:

```
C:\Users\karur>mkdir adv-git-workflow

C:\Users\karur>cd adv-git-workflow

C:\Users\karur\adv-git-workflow>git init
Initialized empty Git repository in C:/Users/karur/adv-git-workflow/.git/

C:\Users\karur\adv-git-workflow>echo "Git Workflow demo" > README.md

C:\Users\karur\adv-git-workflow>git add README.md

C:\Users\karur\adv-git-workflow>git commit -m "Initial commit"
[master (root-commit) 2d558f9] Initial commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

➤ Creating branches and files:

```
C:\Users\karur\adv-git-workflow>git checkout -b feature/login
Switched to a new branch 'feature/login'

C:\Users\karur\adv-git-workflow>echo //login feature > login.js

C:\Users\karur\adv-git-workflow>git add login.js

C:\Users\karur\adv-git-workflow>git commit -m "Added login feature"
[feature/login d9ad240] Added login feature
1 file changed, 1 insertion(+)
create mode 100644 login.js
```

```
C:\Users\karur\adv-git-workflow>git checkout -b bugfix/login-typo
Switched to a new branch 'bugfix/login-typo'

C:\Users\karur\adv-git-workflow>echo // typo fixed >> login.js

C:\Users\karur\adv-git-workflow>git add login.js

C:\Users\karur\adv-git-workflow>git commit -m "Fixing typo in login feature"
[bugfix/login-typo aa224ac] Fixing typo in login feature
1 file changed, 1 insertion(+)
```

```
C:\Users\karur\adv-git-workflow>git checkout -b release/v1.0
Switched to a new branch 'release/v1.0'

C:\Users\karur\adv-git-workflow>echo v1.0 release notes > release-notes.md

C:\Users\karur\adv-git-workflow>git add release-notes.md

C:\Users\karur\adv-git-workflow>git commit -m "Adding release notes for v1.0"
[release/v1.0 dfbb334] Adding release notes for v1.0
1 file changed, 1 insertion(+)
create mode 100644 release-notes.md
```

➤ **Rewriting history:**

```
C:\Users\karur\adv-git-workflow>git log --oneline
0d7a452 (HEAD -> feature/login) Update login UI
d9ad240 Added login feature
2d558f9 (master) Initial commit
```

```
C:\Users\karur\adv-git-workflow>git rebase -i HEAD~2
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/karur/adv-git-workflow/.git/rebase-merge/git-rebase-todo to DOS format...
dos2unix: converting file C:/Users/karur/adv-git-workflow/.git/rebase-merge/git-rebase-todo to Unix format...
hint: Waiting for your editor to close the file... unix2dos: converting file C:/Users/karur/adv-git-workflow/.git/COMMIT_EDITMSG to DOS format...
dos2unix: converting file C:/Users/karur/adv-git-workflow/.git/COMMIT_EDITMSG to Unix format...
[detached HEAD 8eff1f6] Added login feature
Date: Sat Apr 12 12:02:47 2025 +0530
1 file changed, 2 insertions(+)
create mode 100644 login.js
Successfully rebased and updated refs/heads/feature/login.

C:\Users\karur\adv-git-workflow>git log --oneline
8eff1f6 (HEAD -> feature/login) Added login feature
2d558f9 (master) Initial commit
```

➤ **Force push:**

```
C:\Users\karur\adv-git-workflow>git remote add origin https://github.com/Karu08/adv-git-workflow
error: remote origin already exists.

C:\Users\karur\adv-git-workflow>git push -u origin feature/login
Username for 'https://github.com': karu08
Password for 'https://karu08@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 527 bytes | 87.00 KiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote: https://github.com/Karu08/adv-git-workflow.git
To https://github.com/karu08/adv-git-workflow.git
 * [new branch] feature/login -> feature/login
branch 'feature/login' set up to track 'origin/feature/login'.

C:\Users\karur\adv-git-workflow>git push --force origin feature/login
Username for 'https://github.com': karu08
Password for 'https://karu08@github.com':
Everything up-to-date
```

➤ **Simulating mistake – commit lost:**

```
C:\Users\karur\adv-git-workflow>git reset --hard HEAD~1
HEAD is now at 2d558f9 Initial commit

C:\Users\karur\adv-git-workflow>git push --force origin feature/login
Username for 'https://github.com': karu08
Password for 'https://karu08@github.com':
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: This repository moved. Please use the new location:
remote:  https://github.com/Karu08/adv-git-workflow.git
To https://github.com/karu08/adv-git-workflow.git
+ 8eff1f6...2d558f9 feature/login -> feature/login (forced update)
```

➤ **Recovering lost commit using git reflog:**

```
C:\Users\karur\adv-git-workflow>git reflog
2d558f9 (HEAD -> feature/login, origin/feature/login, master) HEAD@{0}: reset: moving to HEAD~1
8eff1f6 HEAD@{1}: rebase (finish): returning to refs/heads/feature/login
8eff1f6 HEAD@{2}: rebase (squash): Added login feature
d9ad240 HEAD@{3}: rebase (start): checkout HEAD~2
0d7a452 HEAD@{4}: rebase (finish): returning to refs/heads/feature/login
0d7a452 HEAD@{5}: rebase (start): checkout HEAD~2
0d7a452 HEAD@{6}: commit: Update login UI
d9ad240 HEAD@{7}: checkout: moving from release/v1.0 to feature/login
dfbb334 (release/v1.0) HEAD@{8}: commit: Adding release notes for v1.0
aa224ac (bugfix/login-typo) HEAD@{9}: checkout: moving from bugfix/login-typo to release/v1.0
aa224ac (bugfix/login-typo) HEAD@{10}: commit: Fixing typo in login feature
d9ad240 HEAD@{11}: checkout: moving from feature/login to bugfix/login-typo
d9ad240 HEAD@{12}: commit: Added login feature
2d558f9 (HEAD -> feature/login, origin/feature/login, master) HEAD@{13}: checkout: moving from master to feature/login
2d558f9 (HEAD -> feature/login, origin/feature/login, master) HEAD@{14}: commit (initial): Initial commit

C:\Users\karur\adv-git-workflow>git checkout d9ad240
Note: switching to 'd9ad240'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.
```

➤ **Finalizing the recovery:**

```
C:\Users\karur\adv-git-workflow>git switch -c recovered-feature
Switched to a new branch 'recovered-feature'
```

➤ **Best Practices for using *force pushes*:**

- Use git pull --rebase : Helps avoid unnecessary merge commits
- Use git push --force-with-lease : Safer than --force, avoids overwriting others' lease
- Notify your team : Communicating before rebasing or force pushing
- Avoid force pushing to main or release branches : Limit it to feature branches
- Create backups before rewriting history : Use tags or clone if needed
- Enable branch protection rules on GitHub : Prevents forced pushes to sensitive branches

➤ **Best Practices for *Team Collaboration*:**

- Communicate before rebasing or force pushing
- Set clear branch policies
- Use Pull Requests to handle merging and reviews
- Always backup with tags before destructive actions

➤ **Summary of Commands used:**

Command	Purpose
mkdir	Creates a new directory (folder)
cd	Changes the current working directory
git init	Initializes a new Git repository in the current directory
echo	Displays a line of text or writes text to a file
git add	Stages changes (files) to be included in the next commit
git commit -m "message"	Records the staged changes with a descriptive commit message
git checkout -b <branch-name>	Creates a new branch and immediately switches to it
git log --oneline	Shows the commit history in a compact format with one line per commit
git rebase -i HEAD~n	Opens an interactive rebase for the last n commits, allowing you to edit, reorder, squash, or remove commits
git remote add origin <url>	Adds a remote repository (usually named origin) and links it to your local repo using the given URL
git push -f origin <branch-name>	Forces the push of your local branch to the remote repository (origin), overwriting any changes that might have been made on the remote branch
git reset --hard HEAD~1	Resets your working directory, staging area, and commit history to one commit before the current HEAD, effectively removing the latest commit and discarding any changes from that commit
git reflog	Shows the history of references (like commits, branch updates, and resets) in your repository, allowing you to track all movements of HEAD and other branches

git checkout <branch-name>	Switches to the specified branch in your repository
git checkout <file>	Restores a file from the most recent commit, discarding any local changes to that file
git switch -c <branch-name>	Creates a new branch with the specified name and immediately switches to it