Set your username and email in git config
Create a new branch named "feature-branch" and switch to it
git checkout -b feature-branch

List all branches in the repository.
git checkout -b feature-branch

Delete the branch “feature-branch”
git checkout -b feature-branch

How do you undo the last commit
git checkout -b feature-branch

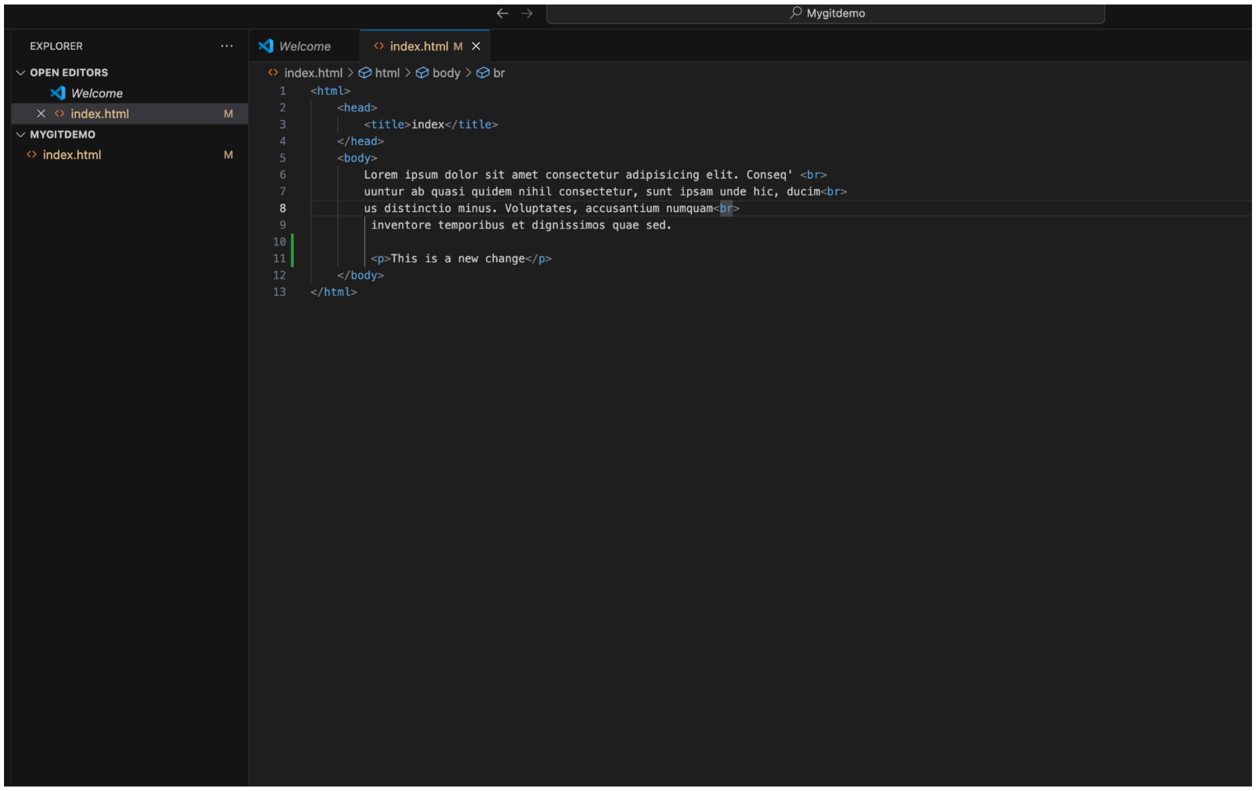
Create a new branch names “conflict-branch”
git checkout -b conflict-branch

Create a another branch named “feature1”
git checkout -b feature-branch

Make some changes in to feature1 branch
Merge “feature1” branch into main branch
git checkout master
git merge feature1

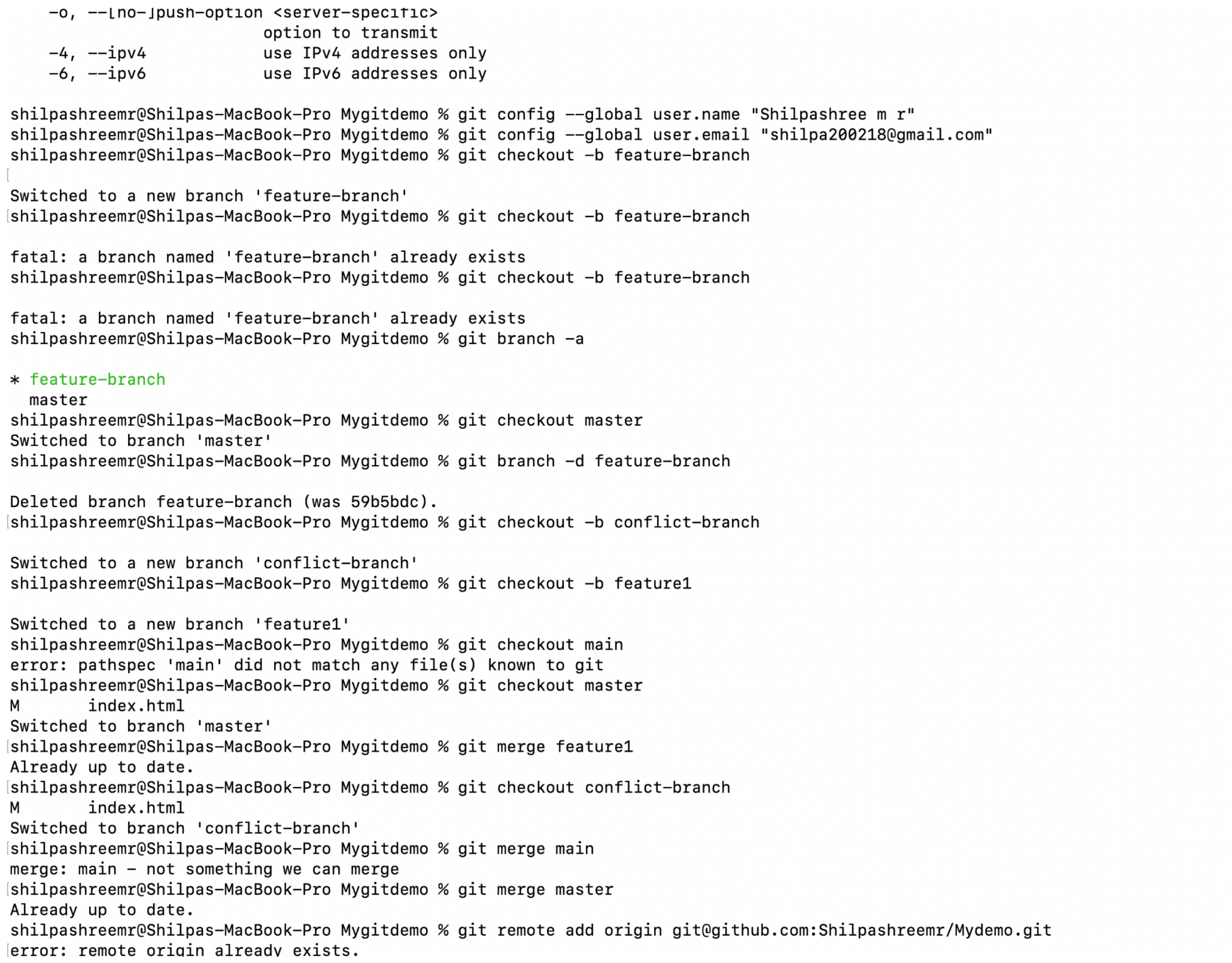
Make changes in “conflict-branch”, in the same file and line that you had made changes in feature1
git checkout main
git merge feature1

Merge master into conflict-branch [Attach screenshot of terminal & file]



```
<html>
    <head>
        <title>index</title>
    </head>
    <body>
        Lorem ipsum dolor sit amet consectetur adipisicing elit. Conseq' <br>
        uuntur ab quasi quidem nihil consectetur, sunt ipsam unde hic, ducim<br>
        us distinctio minus. Voluptates, accusantium numquam<br>
         inventore temporibus et dignissimos quae sed.

         <p>This is a new change</p>
    </body>
</html>
```



```
    -o, --[no-]push-option <server-specific>
                          option to transmit
    -4, --ipv4            use IPv4 addresses only
    -6, --ipv6            use IPv6 addresses only

shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git config --global user.name "Shilpashree m r"
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git config --global user.email "shilpa200218@gmail.com"
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout -b feature-branch

Switched to a new branch 'feature-branch'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout -b feature-branch

fatal: a branch named 'feature-branch' already exists
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout -b feature-branch

fatal: a branch named 'feature-branch' already exists
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git branch -a

* feature-branch
  master
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout master
Switched to branch 'master'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git branch -d feature-branch

Deleted branch feature-branch (was 59b5bdc).
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout -b conflict-branch

Switched to a new branch 'conflict-branch'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout -b feature1

Switched to a new branch 'feature1'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout main
error: pathspec 'main' did not match any file(s) known to git
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout master
M       index.html
Switched to branch 'master'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git merge feature1
Already up to date.
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git checkout conflict-branch
M       index.html
Switched to branch 'conflict-branch'
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git merge main
merge: main - not something we can merge
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git merge master
Already up to date.
shilpashreemr@Shilpas-MacBook-Pro Mygitdemo % git remote add origin git@github.com:Shilpashreemr/Mydemo.git
error: remote origin already exists.
```

Resolve merge conflicts
Add a remote named "origin" pointing to a GitHub repository.
it remote add origin git@github.com:Shilpashreemr/Mydemo.git

Fork a repository on GitHub and clone it to your local machine.
git clone https://github.com/your-username/repository.git

Create a new branch on your fork, make changes, and open a pull request to the original repository.
git checkout -b new-feature

git add .
git commit -m "Add new feature"
git push origin new-feature

Comment on a PR and suggest improvements

Create a Git alias for the command `git log --oneline` named `gitlol`.

```
git config --global alias.gitlol "log --oneline"
```

Create a pre-commit hook

```
git config --global alias.gitlol "log --oneline"
```

You have made local changes in your branch, but you need to switch to another branch urgently without committing. How would you handle this situation?
Stash your changes

You accidentally deleted a file in your local repository. How do you restore it using Git?

```
# Stage the file you forgot to include
git add forgotten-file

# Amend the last commit
git commit --amend --no-edit
```

You have committed changes to your branch but forgot to include a file. How do you add the file to the last commit without creating a new commit?

```
git show <commit-hash>
```

You want to discard all changes in your working directory and revert to the last commit. What Git command would you use?

```
git show <commit-hash>
```

You need to view a specific commit's changes. What Git command can be used to show the changes introduced by a particular commit?

```
git show <commit-hash>
```

You want to change a commit message, after you have already committed, how do you do so?

```
git commit --amend
```

Your colleague has made changes in their branch, and you want to incorporate those changes into your branch without merging. How do you achieve this?

```
git fetch origin
git checkout your-branch
git rebase origin/colleague-branch
```

You've made several commits on a branch, but you want to club them into a single commit before pushing to the remote repository. How would you do that?

```
git rebase -i HEAD~n
```

You accidentally staged a file that you don't want to commit. How do you unstage it?

```
git reset HEAD path/to/file
```

You don’t want to commit files that have .yml in the end, and also files inside folder config. How do you do that?

```
git add -- :!*.yml
git add -- :!config/
```

You want to see a list of all the files changed in the last commit. What Git command would you use?

```
git show --name-only HEAD
```

You realize that your local branch is outdated, and you want to fetch the latest changes from the remote repository. How do you do this without merging?

```
git fetch origin
git rebase origin/main
```

You accidentally deleted a branch. How do you recover it?

```
git fetch origin
git rebase origin/main
```

You want to remove untracked files and directories from your working directory. What Git command would you use?

```
git clean -fd
```

You have a commit from a feature branch that you want to apply to the main branch without merging the entire feature branch.

```
# Switch to the main branch
git checkout main

# Cherry-pick the commit from the feature branch
git cherry-pick <commit-hash>
```

You mistakenly committed a change to the wrong branch and need to apply that commit to the correct branch.

```
# Switch to the correct branch
git checkout correct-branch

# Cherry-pick the commit from the wrong branch
git cherry-pick <commit-hash>

# Optionally, remove the commit from the wrong branch
git checkout wrong-branch
git reset HEAD~1
```

There is a series of commits on a feature branch, but you only want to cherry-pick a specific range of commits.

You want to clone a GitHub repository onto your local machine, but you only need a specific branch. How can you achieve this?
You've made changes to your local repository and want to push them to your fork on GitHub. What Git commands would you use?

```
git push origin branch-name
```

You want to create a new branch both locally and on GitHub to work on a new feature. What commands would you use?

```
# Create a new branch locally
git checkout -b new-feature

# Push the new branch to GitHub
git push origin new-feature
```

You want to see the commit history of a GitHub repository. How can you do this using Git commands?

```
git log
```

You've accidentally committed sensitive information and want to remove the commit from both your local and remote repositories on GitHub. What commands would you use?

```
git filter-branch --force --index-filter "git rm --cached --ignore-unmatch
path/to/sensitive-file" --prune-empty --tag-name-filter cat -- --all
```

```
git push origin --force --all
```

You want to delete a remote branch on GitHub. What Git command would you use?

```
git push origin --delete branch-name
```

Create a git repository for all your assignments and upload them in it. Ask your peers to code review it, and you need to code review your peers assignments

Create a pull request on any open source library on github, attach the pull request link to the readme file of this project's repository