Git and GitHub Workflow Exercise

## 1. Project Initialization

To begin, the project needs to be set up for version control and linked to a GitHub repository. Follow these steps:

**1. Set Up a Local Git Repository:**

* Create a folder for the project:  
    
  mkdir sample-project

cd sample-project

* Initialize a Git repository:  
    
  git init
* Create an initial file (e.g., index.html):  
    
  touch index.html
* Stage and commit the file:  
    
  git add index.html

git commit -m "Initial commit with index.html"

**2.Create a Remote Repository on GitHub:**

Log in to your GitHub account and create a new repository named sample-project.

**3.Link the Local Repository to GitHub:**

* Add the GitHub repository as the remote origin:

git remote add origin https://github.com/your-username/sample-project.git

* Push the initial commit to GitHub:  
  git push -u origin main

## 2. Feature Addition

A new feature requires adding a README.md file to introduce the project.

**1.Create and Edit the README.md File:**

* Create the file:  
    
  touch README.md
* Add the following content to the file:

# Sample Project

This project demonstrates a Git and GitHub workflow for collaborative software development.

**2.Stage and Commit the Changes:**

* Stage the file:  
    
  git add README.md
* Commit the changes:

git commit -m "Added README.md with project introduction"

## 3. Branch Management

To make updates without affecting the main branch, you need to create and manage a new branch.

**1.Create a New Branch:**

* Create and switch to a branch named feature/update-readme:

git checkout -b feature/update-readme

**2.Push the Branch to GitHub:**

* Push the new branch:  
    
  git push -u origin feature/update-readme

## 

## 4. Merge Changes

Once the updates are reviewed, merge the feature/update-readme branch into the main branch.

**1.Switch to the Main Branch:**

git checkout main

**2.Merge the Feature Branch:**  
 git merge feature/update-readme

3.**Push the Merged Changes to GitHub:**  
 git push origin main

## 

## 5. Branch Cleanup

After the merge, delete the feature branch locally and remotely.

**1.Delete the Branch Locally:**  
 git branch -d feature/update-readme

**2.Delete the Branch Remotely:**

git push origin --delete feature/update-readme

## Deliverables

By following this workflow, the following deliverables will be achieved:

1. A GitHub repository named sample-project containing the project files.
2. A branch named feature/update-readme with the updated README.md file.
3. The README.md file merged into the main branch.
4. The feature/update-readme branch deleted both locally and remotely.

## Key Questions Answers

1. **How will you set up the project for version control and push it to GitHub?**

Initialize the repository: git init

Create a remote link: git remote add origin <GitHub URL>

Push changes: git push -u origin main

1. **What steps will you take to create, push, and manage branches effectively?**

Create a branch: git checkout -b feature/update-readme

Push the branch: git push -u origin feature/update-readme

Commit changes regularly and review before merging.

1. **How will you ensure the changes in the feature branch are merged into the main branch?**

Switch to the main branch: git checkout main

Merge the branch: git merge feature/update-readme

Push the changes to GitHub: git push origin main

1. **What commands will you use to clean up unused branches locally and remotely?**

Delete locally: git branch -d feature/update-readme

Delete remotely: git push origin --delete feature/update-readme