Implementing Basic Version Control with Git

Project Selection: Simple Webpage
For this demonstration, we will create a simple webpage project and implement basic version control using Git. The webpage will consist of an index.html file.

Step-by-Step Guide

1. Setting Up the Project Directory
First, let's create a directory for our project and navigate into it.

```
mkdir simple-webpage
cd simple-webpage
```

2. Initializing a Git Repository
Initialize a new Git repository in the project directory.

```
git init
```
This command sets up the necessary files and directories that Git uses to track changes.

3. Creating the Project Files
Create an index.html file for our webpage.

```
echo "<!DOCTYPE html>
<html>
<head>
    <title>Simple Webpage</title>
</head>
<body>
    <h1>Welcome to my simple webpage!</h1>
```

</body>
</html>" > index.html

```
\Assignment 5.1\simple-webpage> echo "<!DOCTYPE ht
ml>
>> <html>
>> <head>
>>     <title>Simple Webpage</title>
>> </head>
>> <body>
>>     <h1>Welcome to my simple webpage!</h1>
>> </body>
>> </html>" > index.html
```

4. Adding Files to the Repository

Add the index.html file to the staging area.

git add index.html

The git add command tells Git to start tracking changes to the specified file.

```
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\A
ssignment 5.1\simple-webpage> git add .
```

5. Committing the Changes

Commit the changes to the repository with a descriptive message.

git commit -m "Initial commit: Add index.html with basic webpage structure"

The git commit command saves the changes to the repository history. The -m option allows us to add a commit message inline.

```
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\A
ssignment 5.1\simple-webpage> git commit -m "Initial
 commit: Add index.html with basic webpage structure
"
[master (root-commit) d3264d1] Initial commit: Add i
ndex.html with basic webpage structure
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 index.html
```

6. Making Changes and Tracking Versions

Let's make some changes to the index.html file to demonstrate version tracking. Add a paragraph to the body of the HTML.

```
echo "<!DOCTYPE html>
<html>
<head>
    <title>Simple Webpage</title>
</head>
<body>
    <h1>Welcome to my simple webpage!</h1>
    <p>This is a paragraph added to demonstrate version control.</p>
</body>
</html>" > index.html
```

Add the changes to the staging area and commit them.

```
git add index.html
git commit -m "Add paragraph to the body of the HTML"
```

```
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\Assignment 5.1\simple-webpage> echo "<!DOCTYPE html>
>> <html>
>> <head>
>>     <title>Simple Webpage</title>
>> </head>
>> <body>
>>     <h1>Welcome to my simple webpage!</h1>
>>     <p>This is a paragraph added to demonstrate version control.</p>
>> </body>
>> </html>" > index.html
```

7. Viewing the Commit History

You can view the commit history to see the changes over time.

```
git log
```

This command shows a list of all commits, along with their messages, author information, and timestamps.

```
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\Assignment 5.1\simple-webpage> git log
commit 9aea7fb1a00aab7cabc041b4f74d93b1983f79b3 (HEAD -> master)
Author: Somendra singh <93323212+SomenSin@users.noreply.github.com>
Date:   Sat May 25 20:30:08 2024 +0530

    Add paragraph to the body of the HTML

commit d3264d182c206ba3849c5b184c9bcad6c0f3ee80
Author: Somendra singh <93323212+SomenSin@users.noreply.github.com>
Date:   Sat May 25 20:29:24 2024 +0530

    Initial commit: Add index.html with basic webpage structure
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\Assignment 5.1\simple-webpage>
```

Rationale and Benefits of Version Control

Rationale:

- Tracking Changes: Git allows us to track changes in the project files over time. Each change is recorded with a commit message that describes the modification, making it easier to understand the project's history.
- Collaboration: Multiple team members can work on the same project simultaneously without overwriting each other's work. Git merges changes from different sources seamlessly.
- Reversibility: If a mistake is made, Git allows us to revert to a previous state of the project, preventing potential loss of work.

Benefits:

- Accountability: Every change is recorded with information about who made the change and why it was made. This ensures accountability within a team.
- Branching and Merging: Developers can work on new features in isolated branches and merge them into the main project once they are complete. This allows for parallel development and cleaner integration of new features.
- Backup: The repository can be cloned to other machines, ensuring that there is always a backup of the project.

Steps to Connect to a Remote Repository

8. Create a Remote Repository on GitHub

1. Log in to GitHub: Go to GitHub and log in to your account.
2. Create a New Repository:

- Click on the "+" icon in the upper right corner of the page and select "New repository".
- Enter a repository name, e.g., simple-webpage.
- Optionally, add a description.
- Choose between public or private visibility.
- Do not initialize the repository with a README, .gitignore, or license (we will add these later if needed).
- Click "Create repository".

10. Add the Remote Repository to Your Local Repository

Now that we have a remote repository, we need to link our local repository to this remote one.

```
# Navigate to your project directory if not already there
cd simple-webpage

# Add the remote repository
git remote add origin https://github.com/SomenSin/simple-webpage-.git
```

Replace your-username with your actual GitHub username.

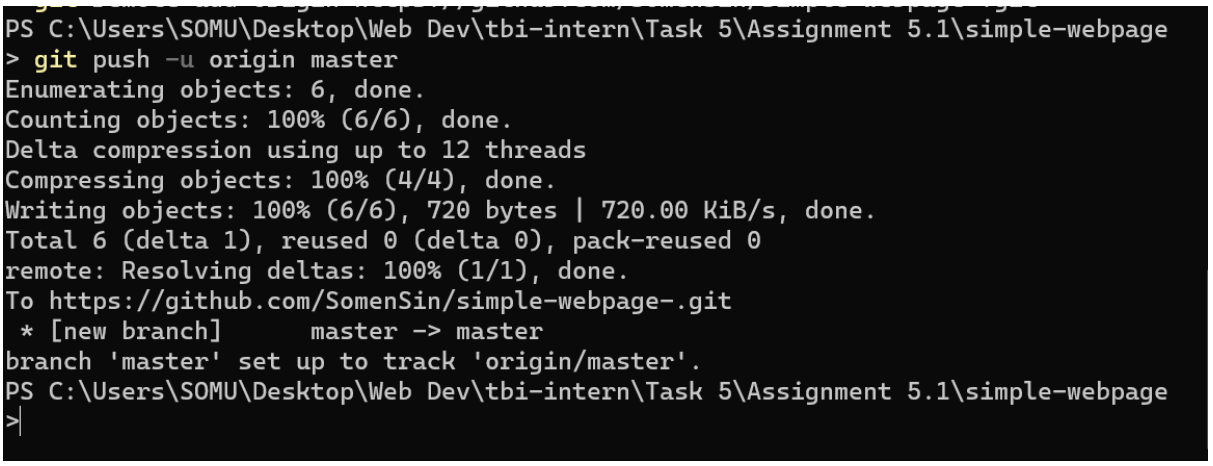
```
PS C:\Users\SOMU\Desktop\Web Dev\tbi-intern\Task 5\Assignment 5.1\simple-
webpage> git remote add origin https://github.com/SomenSin/simple-webpage
-.git
```

11. Push Your Local Repository to GitHub

Push your local commits to the remote repository on GitHub. This will upload your code to the remote repository.

```
git push -u origin master
```

The -u flag sets the upstream tracking for the master branch, so future pushes can be done simply with git push.



Detailed Explanation of Commands

- git remote add origin <URL>: This command adds a new remote repository with the name origin and the specified URL. origin is a conventional name, but you can use any name.
- git push -u origin master: This command pushes your local commits to the master branch of the origin remote repository. The -u flag sets the upstream branch, which simplifies future pushes and pulls.

Benefits of Connecting to a Remote Repository

1. Collaboration: Multiple developers can work on the same project, share code, and merge their changes.
2. Backup: The remote repository acts as a backup for your project. If your local machine fails, you can clone the project from the remote repository.
3. Access from Anywhere: You can access the project from any machine by cloning the remote repository.
4. Continuous Integration: Many CI/CD tools integrate with remote repositories to automatically build and test your code when changes are pushed.

Rationale and Benefits of Version Control

Rationale

Tracking Changes: Git allows us to track changes in the project files over time. Each change is recorded with a commit message that describes the modification, making it easier to understand the project's history.

Collaboration: Multiple team members can work on the same project simultaneously without overwriting each other's work. Git merges changes from different sources seamlessly.
Reversibility: If a mistake is made, Git allows us to revert to a previous state of the project, preventing potential loss of work.

Benefits
Accountability: Every change is recorded with information about who made the change and why it was made. This ensures accountability within a team.
Branching and Merging: Developers can work on new features in isolated branches and merge them into the main project once they are complete. This allows for parallel development and cleaner integration of new features.
Backup: The repository can be cloned to other machines, ensuring that there is always a backup of the project.

Summary
By setting up a Git repository for our simple webpage project, we have implemented a basic version control system. We tracked changes, committed them with descriptive messages, and viewed the history of our project. This process highlights the importance and benefits of using version control in project management and collaboration, ensuring that development is efficient, organized, and error-free

GitHub Repository Link:- https://github.com/SomenSin/simple-webpage-