EXPRIMENT -04

SHILPA SHREE.R
314CS20048

GIT OPERATIONS

Git

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Creating a repository

Step 1: Go to the official website: https://github.com/

Step 2: Sign up to the Github using email address.

Step 3: Enter username and create a password.

Step 4: You will be able to login to Github successfully.

Step 5: Create a new repository.

Step 6: Enter your repository name and enter the description.

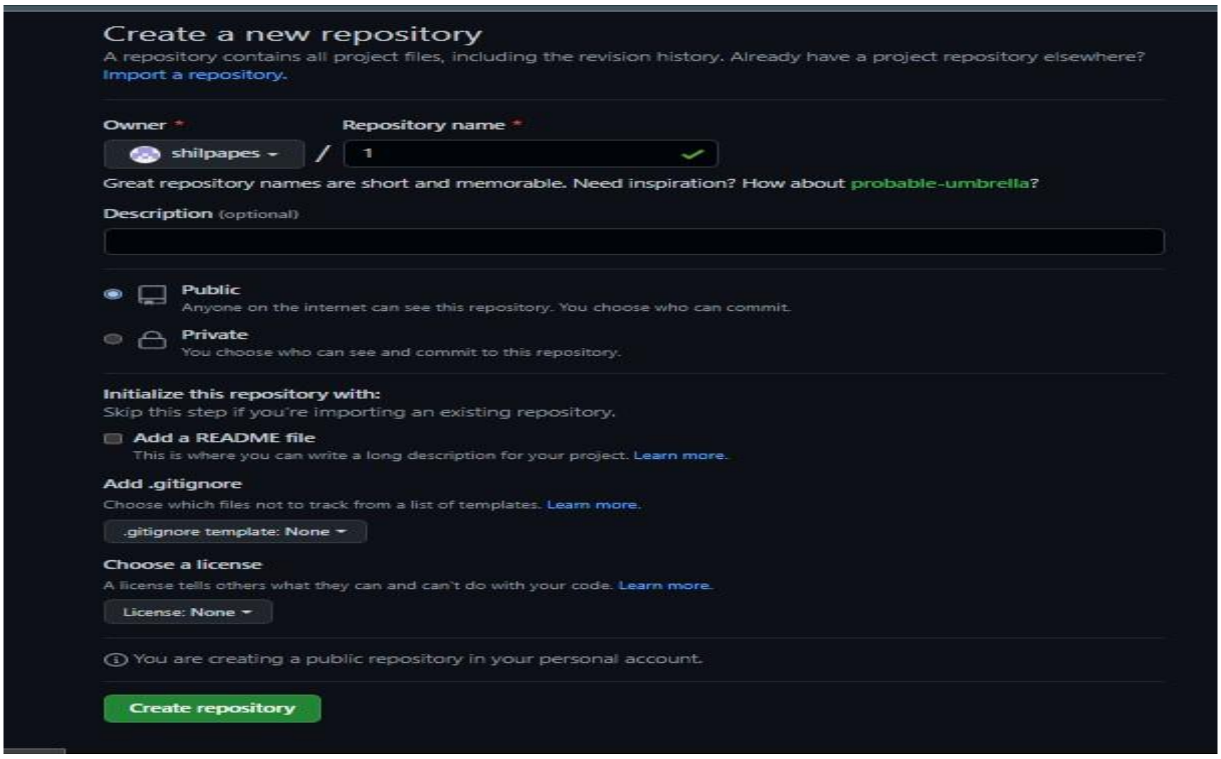
Step 7: You can choose who can commit by selecting public or private.

Step 8: Initialize the repository with choosing add a README file and create the repository.

EXPRIMENT -04

SHILPA SHREE.R
314CS20048

GIT OPERATIONS



Create a new repository
A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * shilpapes / Repository name * 1

Great repository names are short and memorable. Need inspiration? How about probable-umbrella?

Description (optional)

● Public — Anyone on the internet can see this repository. You choose who can commit.
○ Private — You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.
☐ Add a README file — This is where you can write a long description for your project. Learn more.

Add .gitignore
Choose which files not to track from a list of templates. Learn more.
.gitignore template: None

Choose a license
A license tells others what they can and can't do with your code. Learn more.
License: None

You are creating a public repository in your personal account.

Create repository

Cloning a repository.

Step 1: create an empty folder named ‘git’ and save it on desktop.

Step 2: open the gitbash

Step 3: type ‘cd’ and press enter

Step 4: type cd Desktop and press enter

Step 5: type cd git and press enter

Step 6: Type the following command

git config --global user.name “ enter user name>“---- (press enter)

git config --global user.email enter email address----- (press enter)

git clone <enter the address that we get from the code section of the repository>

EXPRIMENT -04

SHILPA SHREE.R
314CS20048

GIT OPERATIONS

Step 7: open the git folder -> open your repository -.> create a text file.

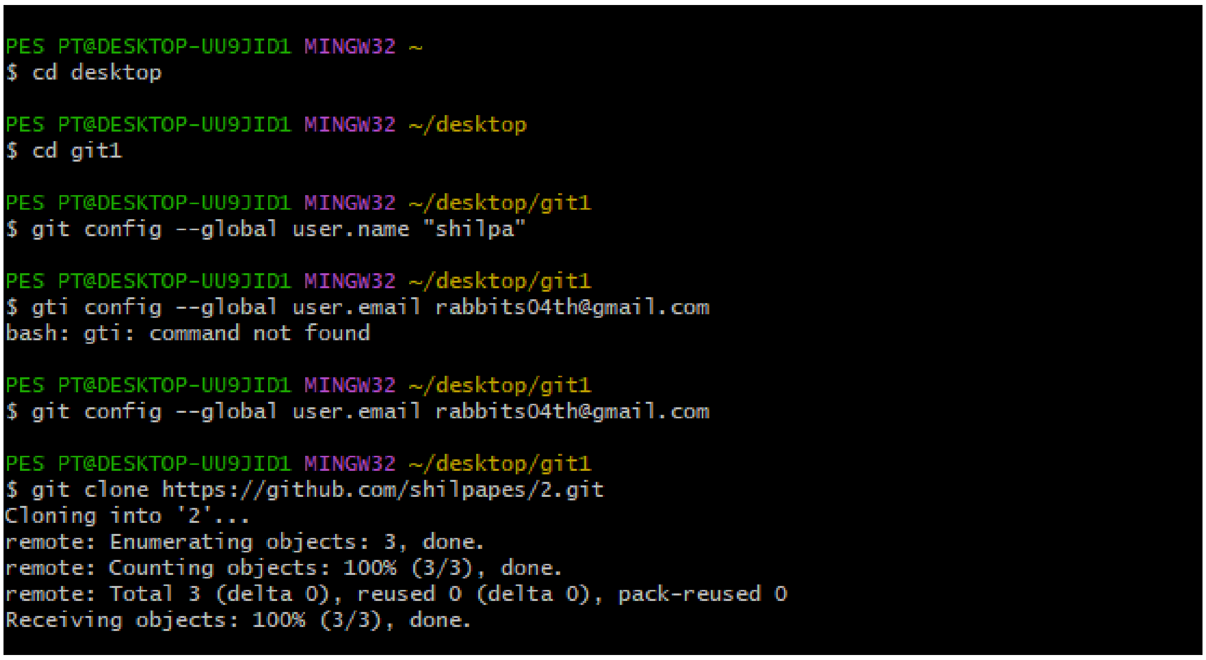
Step 8: come back to git bash and execute the below
commandGit add <text file you created> (press enter)
Git status (press enter)

Git Commit -m “first commit” (press enter)

Git push –u origin main (press enter) (Enter your username and password)

Step 10: You will get the successful output

Step 11: Go to your repository of github and text file will be added to the repository.

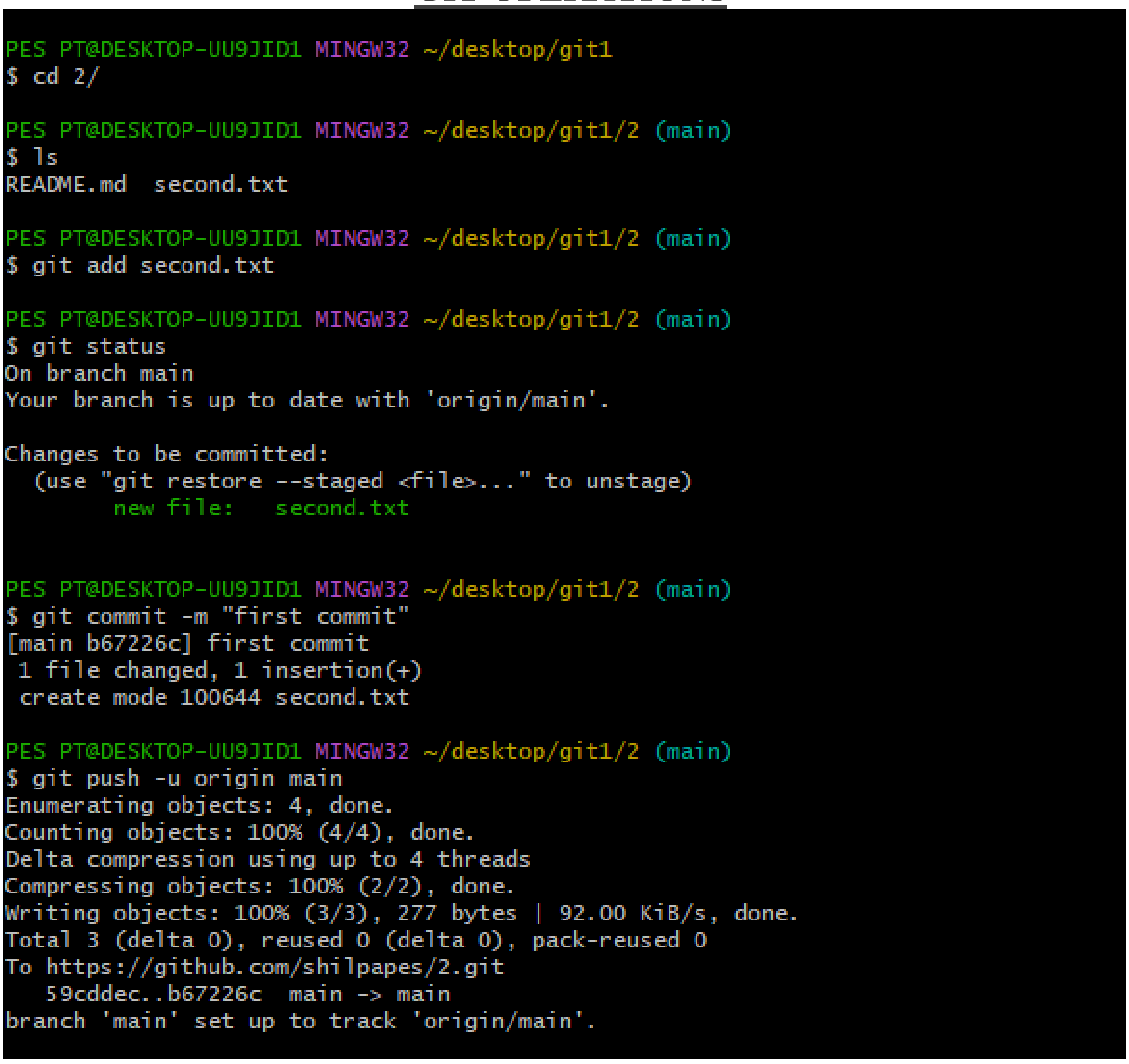


```
PES PT@DESKTOP-UU9JID1 MINGW32 ~
$ cd desktop

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop
$ cd git1

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1
$ git config --global user.name "shilpa"

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1
$ gti config --global user.email rabbits04th@gmail.com
bash: gti: command not found

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1
$ git config --global user.email rabbits04th@gmail.com

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1
$ git clone https://github.com/shilpapes/2.git
Cloning into '2'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

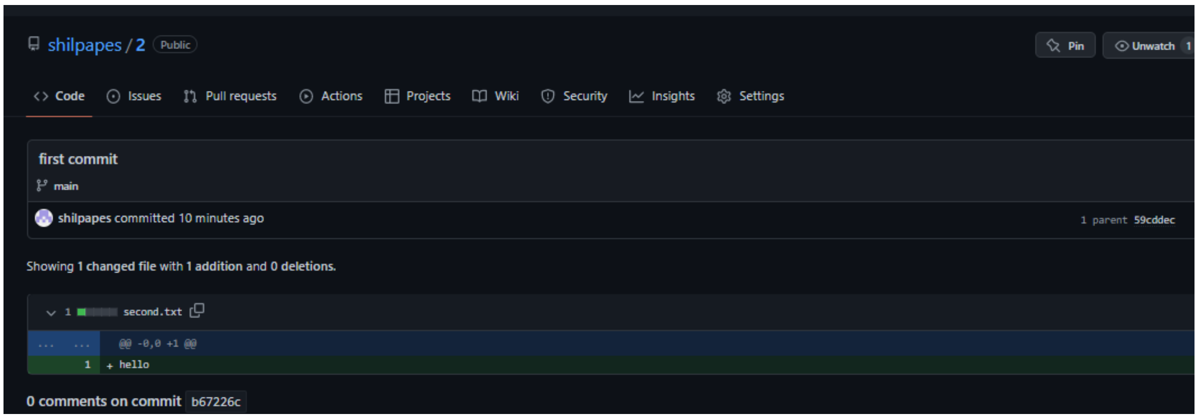
EXPRIMENT -04

SHILPA SHREE.R
314CS20048

GIT OPERATIONS



```
PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1
$ cd 2/

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ ls
README.md  second.txt

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ git add second.txt

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ 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:   second.txt


PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ git commit -m "first commit"
[main b67226c] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 second.txt

PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ git push -u origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 92.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/shilpapes/2.git
   59cddec..b67226c  main -> main
branch 'main' set up to track 'origin/main'.
```



first commit
main
shilpapes committed 10 minutes ago   1 parent 59cddec

Showing 1 changed file with 1 addition and 0 deletions.

```
1 second.txt
@@ -0,0 +1 @@
1 + hello
```

0 comments on commit b67226c

Making and recording changes and for committing stages:

EXPRIMENT -04

SHILPA SHREE.R
314CS20048

GIT OPERATIONS

Step 1: come back to git bash and execute the below commandGit add <text file you created> (press enter)
Git status (press enter)

Git Commit -m “first commit” (press enter)

Git push –u origin main (press enter) (Enter your username and password)

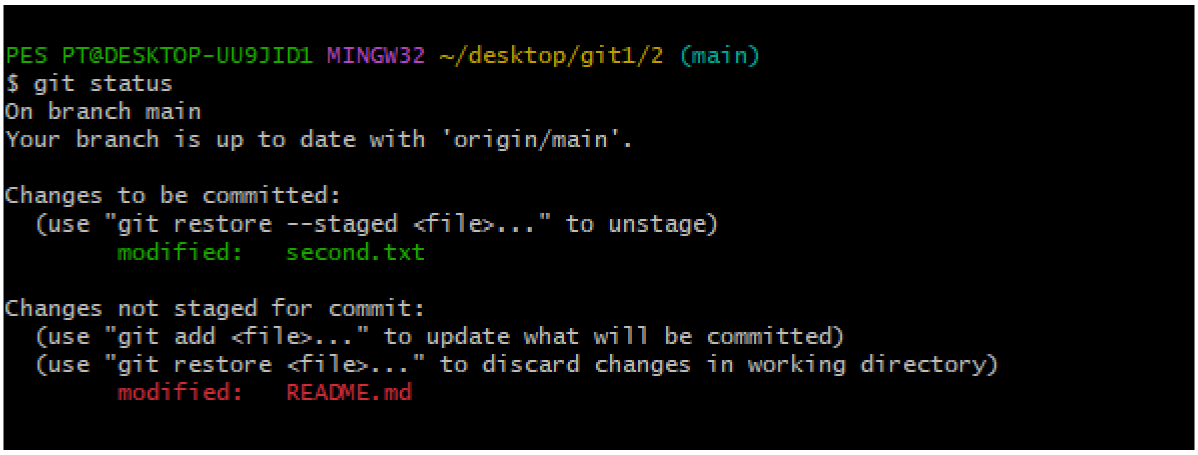
Step 2: You will get the successful output

Step 3: Go to your repository of github and text file will be added to the repository.

Step 4: Change the content in the file which you have added to your repository.

Step 5: In the git bash type Git status <enter>

Step 6: Open the repository and view the changes.



```
PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ 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)
        modified:   second.txt

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:   README.md
```

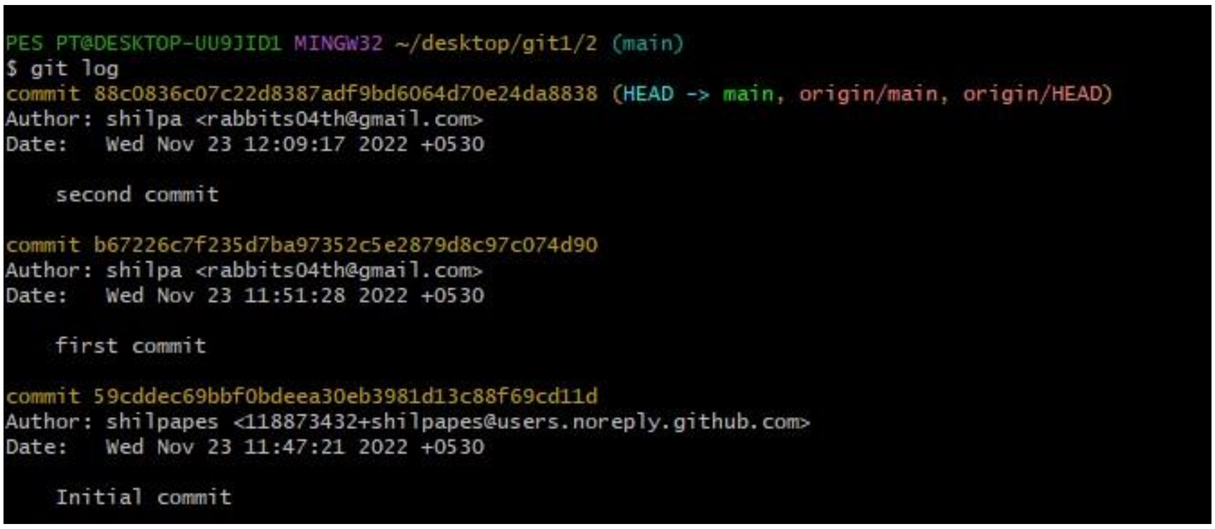
EXPRIMENT -04

SHILPA SHREE.R
314CS20048

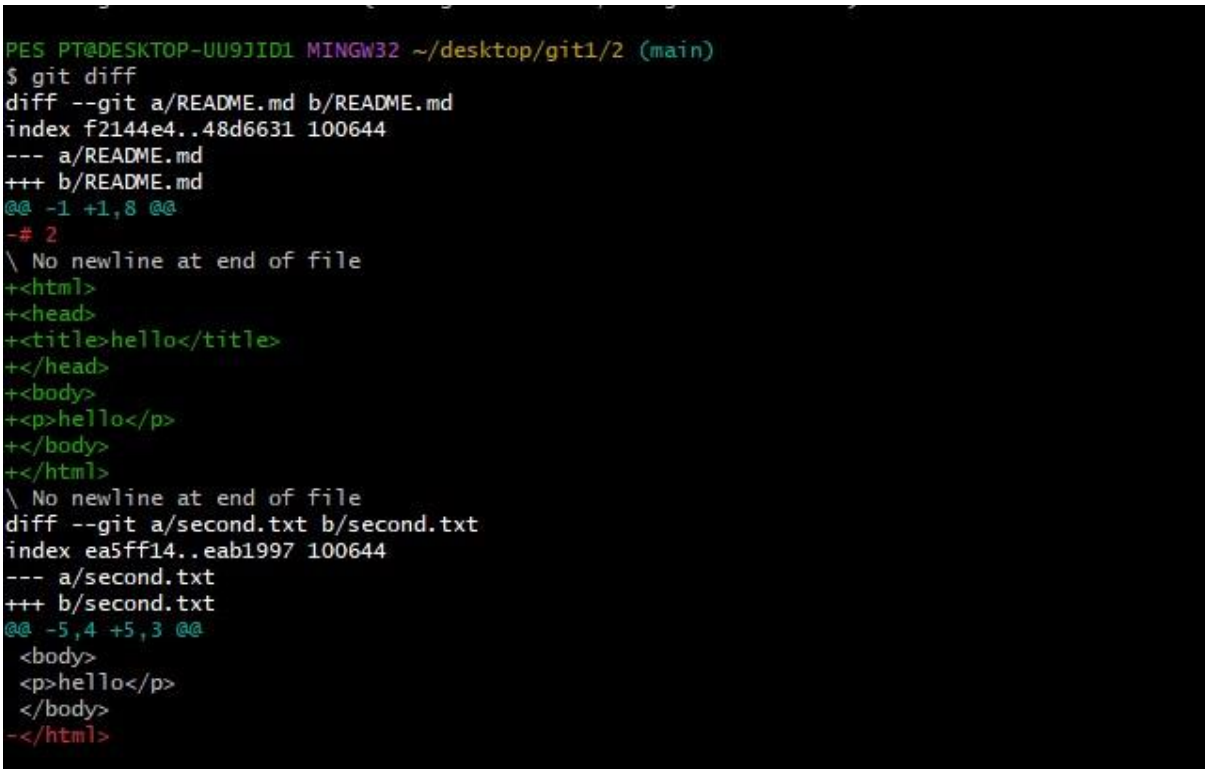
GIT OPERATIONS

For viewing the history of all the changes type the below command:

Git log<enter>



```
PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ git log
commit 88c0836c07c22d8387adf9bd6064d70e24da8838 (HEAD -> main, origin/main, origin/HEAD)
Author: shilpa <rabbits04th@gmail.com>
Date:   Wed Nov 23 12:09:17 2022 +0530

    second commit

commit b67226c7f235d7ba97352c5e2879d8c97c074d90
Author: shilpa <rabbits04th@gmail.com>
Date:   Wed Nov 23 11:51:28 2022 +0530

    first commit

commit 59cddec69bbf0bdeea30eb3981d13c88f69cd11d
Author: shilpapes <118873432+shilpapes@users.noreply.github.com>
Date:   Wed Nov 23 11:47:21 2022 +0530

    Initial commit
```



```
PES PT@DESKTOP-UU9JID1 MINGW32 ~/desktop/git1/2 (main)
$ git diff
diff --git a/README.md b/README.md
index f2144e4..48d6631 100644
--- a/README.md
+++ b/README.md
@@ -1 +1,8 @@
-# 2
\ No newline at end of file
+<html>
+<head>
+<title>hello</title>
+</head>
+<body>
+<p>hello</p>
+</body>
+</html>
\ No newline at end of file
diff --git a/second.txt b/second.txt
index ea5ff14..eab1997 100644
--- a/second.txt
+++ b/second.txt
@@ -5,4 +5,3 @@
 <body>
 <p>hello</p>
 </body>
-</html>
```