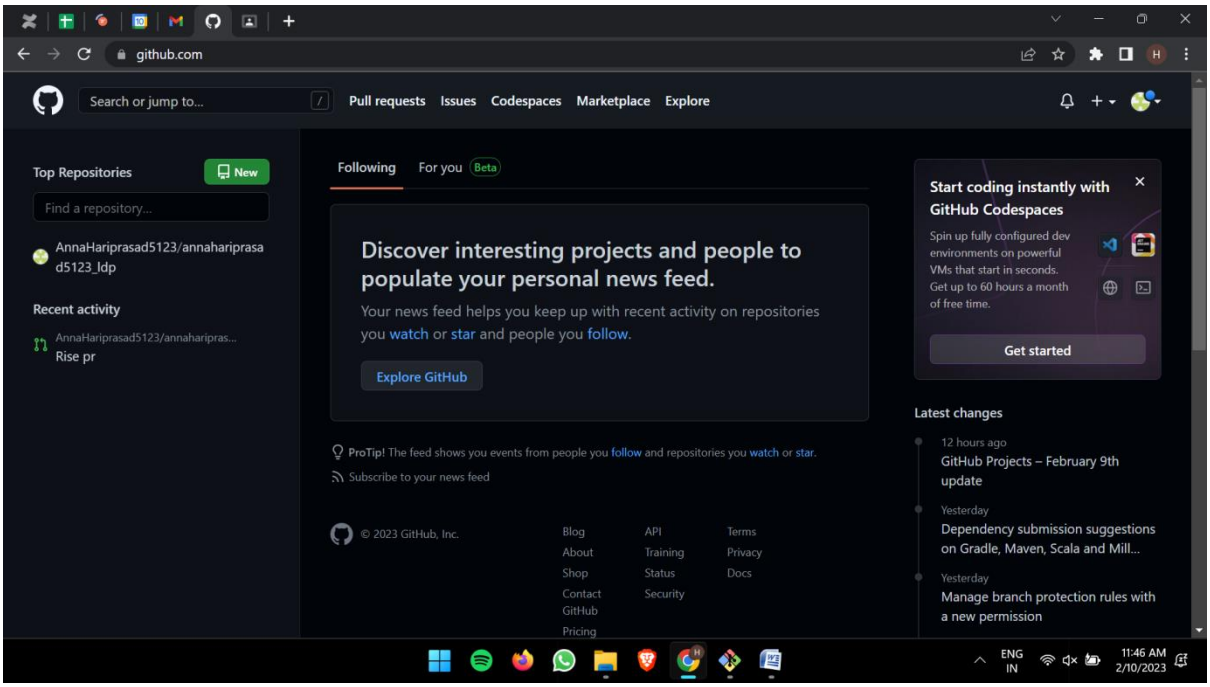
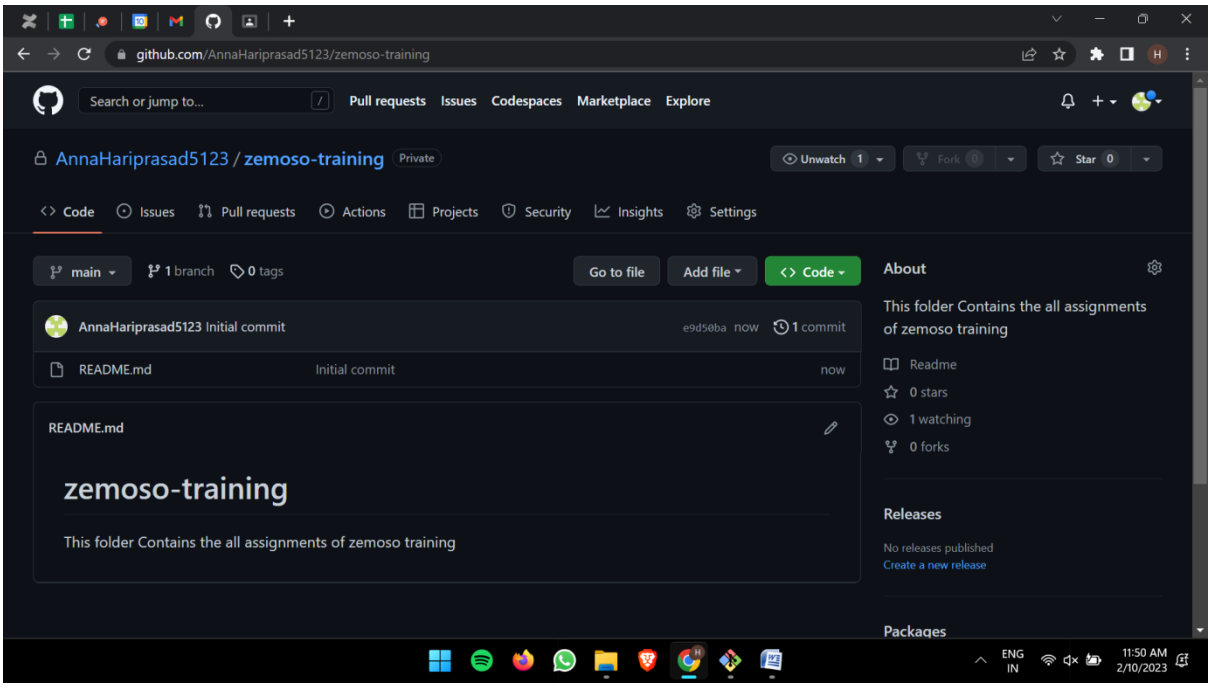
Git Assignment

February 10, 2023

- Sign up for GitHub.



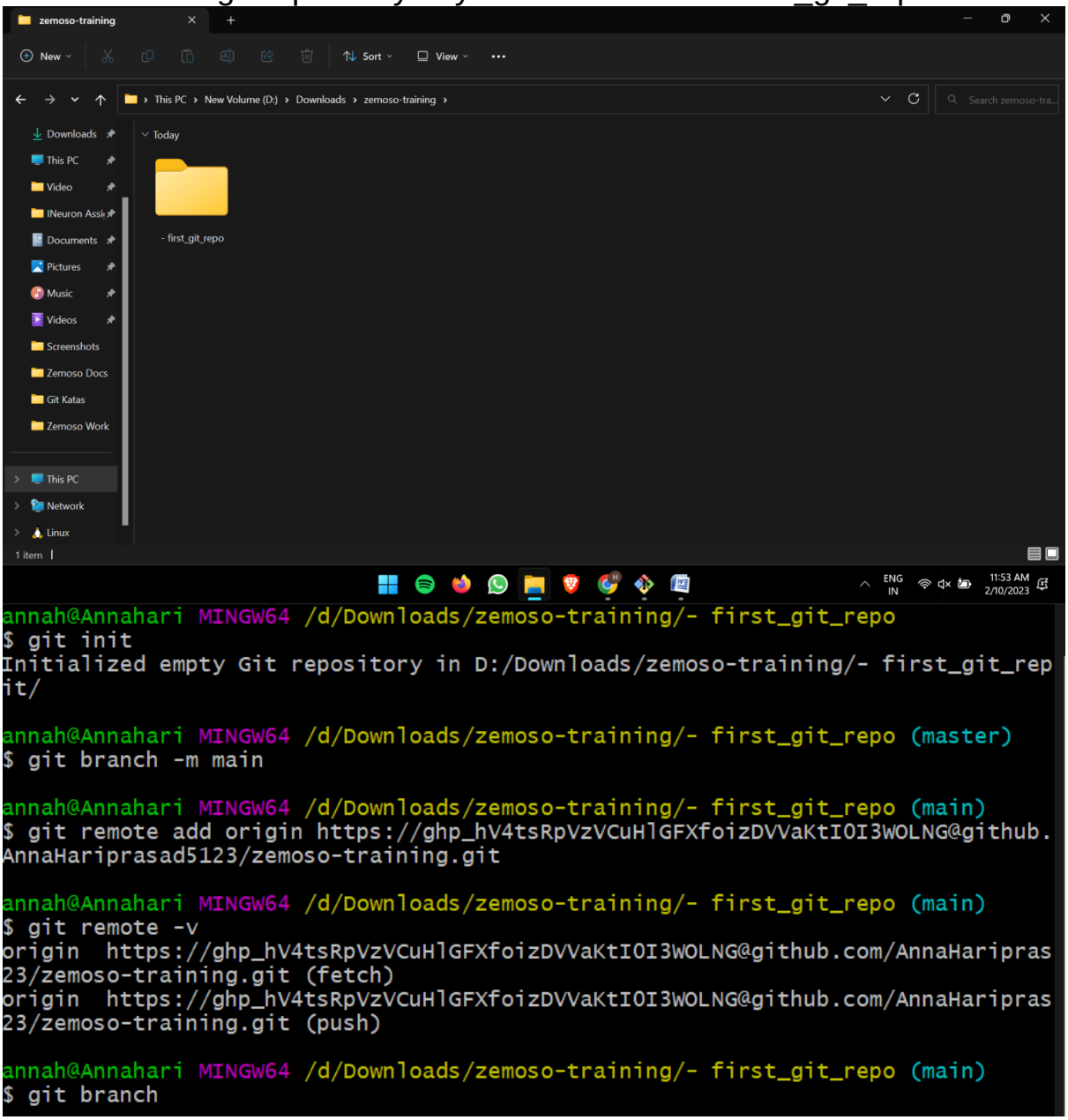
- Create repository zemoso-training in your account.



Git Assignment

February 10, 2023

- Create a local git repository in your home folder - first_git_repo



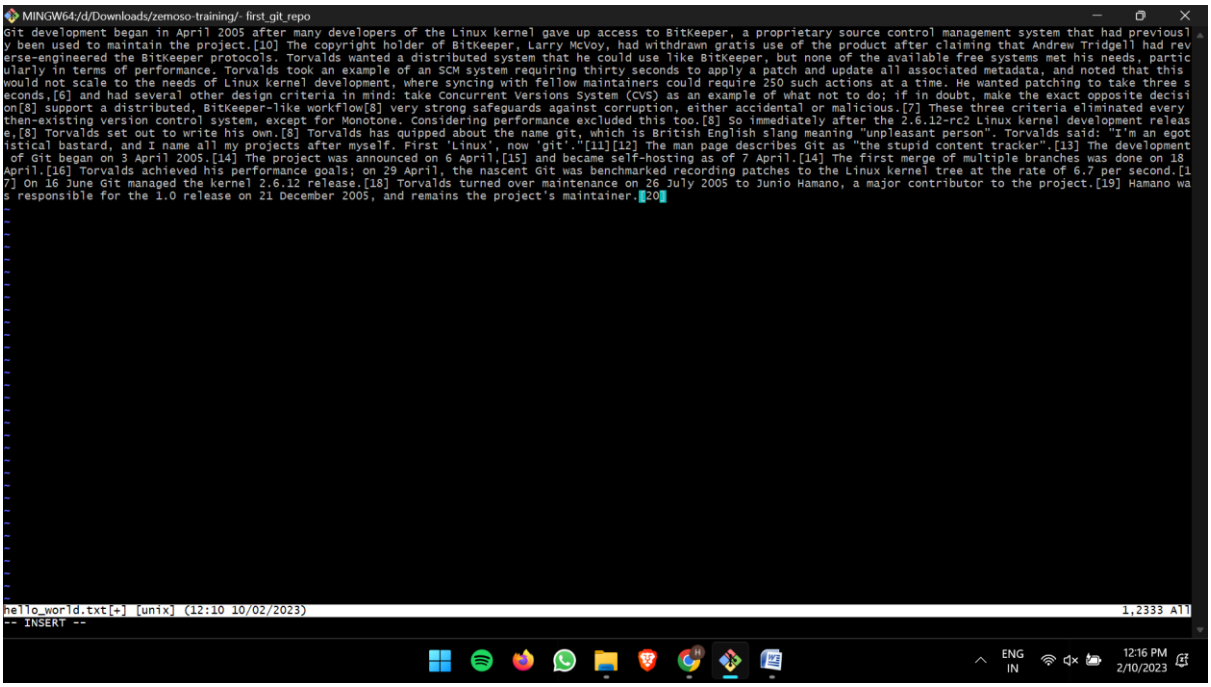
- Create a text file named hello_world and copy text as below.

Git development began in April 2005 after many developers of the Linux kernel gave up access to BitKeeper, a proprietary source control management system that had previously been used to maintain the project.[10] The copyright holder of BitKeeper, Larry McVoy, had withdrawn gratis use of the product after claiming that Andrew Tridgell had reverse-engineered the BitKeeper protocols. Torvalds wanted a distributed system that he could use like BitKeeper, but none of the available free systems met his needs, particularly in terms of performance. Torvalds took an example of an SCM system requiring thirty seconds to apply a patch and update all associated metadata, and noted that this would not scale to the needs of Linux kernel development, where syncing with fellow maintainers could require 250 such actions at a time. He wanted patching to take three seconds,[6] and had several other design criteria in mind: take Concurrent Versions System (CVS) as an example of what not to do; if in doubt, make the exact opposite decision[8] support a distributed, BitKeeper-like workflow[8] very strong safeguards against corruption, either accidental or malicious.[7] These three criteria eliminated every then-existing version control system, except for Monotone. Considering performance excluded this too.[8] So immediately after the 2.6.12-rc2 Linux kernel development release,[8] Torvalds set out to write his own.[8] Torvalds has quipped about the name git, which is British English slang meaning "unpleasant person". Torvalds said: "I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'."[11][12] The man page describes Git as "the stupid content tracker".[13] The development of Git began on 3 April 2005.[14] The project was announced on 6 April,[15] and became self-hosting as of 7 April.[14] The first merge of multiple branches was done on 18 April.[16] Torvalds achieved his performance goals; on 29 April, the nascent Git was benchmarked recording patches to the Linux kernel tree at the rate of 6.7 per second.[17] On 16 June Git managed the kernel 2.6.12 release.[18] Torvalds turned over maintenance on 26 July 2005 to Junio Hamano, a major contributor to the project.[19] Hamano was responsible for the 1.0 release on 21 December 2005, and remains the project's maintainer.[20]

Git Assignment

February 10, 2023

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ touch hello_world.txt

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git config --global core.editor C:/Program Files/Notepad++/notepad++.exe

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ vi hello_world.txt
```



- Commit to your local repository and push it to the Github repository.

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello_world.txt

nothing added to commit but untracked files present (use "git add" to track)

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git add .
warning: LF will be replaced by CRLF in hello_world.txt.
The file will have its original line endings in your working directory

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   hello_world.txt


annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git commit -m "hello_world file is added"
[main (root-commit) 1f131f8] hello_world file is added
 1 file changed, 1 insertion(+)
 create mode 100644 hello_world.txt
```

Git Assignment

February 10, 2023

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git push -f origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 1.39 KiB | 1.39 MiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/AnnaHariprasad5123/zemoso-training.git
 + e9d50ba...1f131f8 main -> main (forced update)
```

- Replace all "to" words by your name. Check git diff to make sure all to words have been replaced and make another commit and push it to Github.

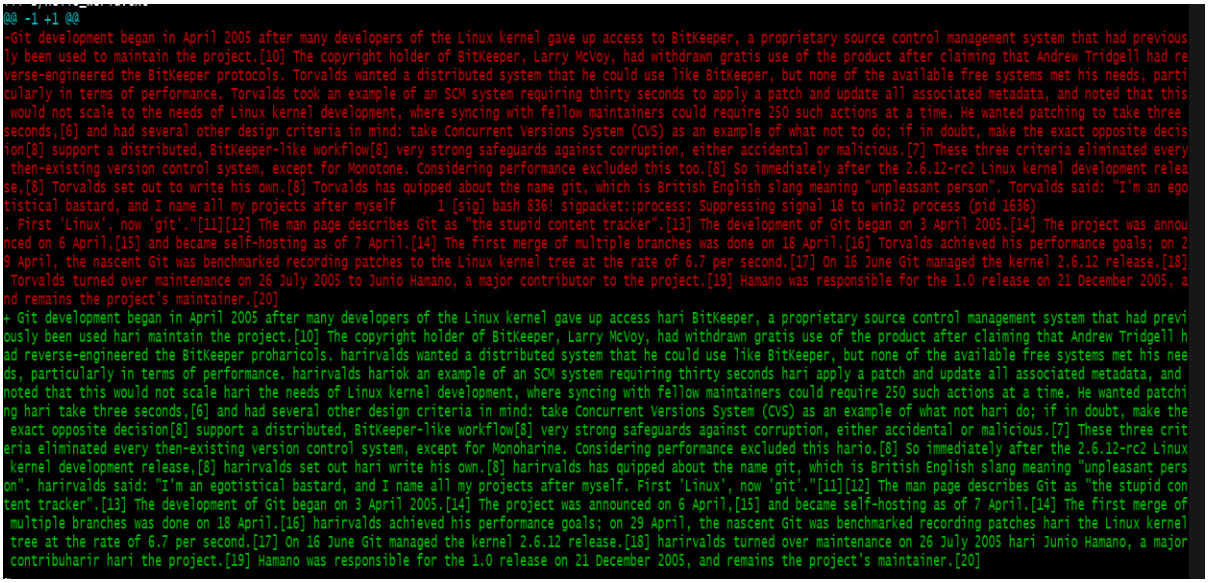
```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ vi hello_world.txt
```

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ cat hello_world.txt
 Git development began in April 2005 after many developers of the Linux kernel
 gave up access hari BitKeeper, a proprietary source control management system
 that had previously been used hari maintain the project.[10] The copyright ho
lder of BitKeeper, Larry McVoy, had withdrawn gratis use of the product after
claiming that Andrew Tridgell had reverse-engineered the BitKeeper proharicols
. harirvalds wanted a distributed system that he could use like BitKeeper, but
 none of the available free systems met his needs, particularly in terms of pe
rformance. harirvalds hariok an example of an SCM system requiring thirty seco
nds hari apply a patch and update all associated metadata, and noted that this
 would not scale hari the needs of Linux kernel development, where syncing wit
h fellow maintainers could require 250 such actions at a time. He wanted patch
ing hari take three seconds,[6] and had several other design criteria in mind:
 take Concurrent Versions System (CVS) as an example of what not hari do; if i
n doubt, make the exact opposite decision[8] support a distributed, BitKeeper-
like workflow[8] very strong safeguards against corruption, either accidental
or malicious.[7] These three criteria eliminated every then-existing version c
ontrol system, except for Monoharine. Considering performance excluded this ha
rio.[8] So immediately after the 2.6.12-rc2 Linux kernel development release,[
8] harirvalds set out hari write his own.[8] harirvalds has quipped about the
name git, which is British English slang meaning "unpleasant person". harirval
ds said: "I'm an egotistical bastard, and I name all my projects after myself.
 First 'Linux', now 'git'."[11][12] The man page describes Git as "the stupid
content tracker".[13] The development of Git began on 3 April 2005.[14] The pr
oject was announced on 6 April,[15] and became self-hosting as of 7 April.[14]
 The first merge of multiple branches was done on 18 April.[16] harirvalds ach
ieved his performance goals; on 29 April, the nascent Git was benchmarked reco
rding patches hari the Linux kernel tree at the rate of 6.7 per second.[17] On
 16 June Git managed the kernel 2.6.12 release.[18] harirvalds turned over mai
ntenance on 26 July 2005 hari Junio Hamano, a major contribuharir hari the pro
ject.[19] Hamano was responsible for the 1.0 release on 21 December 2005, and
remains the project's maintainer.[20]
```

Git Assignment

February 10, 2023

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git diff
warning: LF will be replaced by CRLF in hello_world.txt.
The file will have its original line endings in your working directory
diff --git a/hello_world.txt b/hello_world.txt
index bab22f2..926ea1f 100644
--- a/hello_world.txt
+++ b/hello_world.txt
```




```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git status
On branch main
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:   hello_world.txt

no changes added to commit (use "git add" and/or "git commit -a")

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git add .
warning: LF will be replaced by CRLF in hello_world.txt.
The file will have its original line endings in your working directory
g
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git commit -m "hello_world file is updated"
[main 3fe7ea5] hello_world file is updated
 1 file changed, 1 insertion(+), 1 deletion(-)
 rewrite hello_world.txt (100%)

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git push -f origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 459 bytes | 459.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/AnnaHariprasad5123/zemoso-training.git
   1f131f8..3fe7ea5  main -> main
```

- Revert back to Step 5 and check in again.

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git log --oneline
3fe7ea5 (HEAD -> main, origin/main) hello_world file is updated
1f131f8 hello_world file is added
```

```
MINGW64:/d/Downloads/zemoso-training/- first_git_repo
Revert "hello_world file is reverted"

This reverts commit 3fe7ea58b2b15f8313587d1df910305ee7bf148f.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# On branch main
# Changes to be committed:
#       modified:   hello_world.txt
#
~
```

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git revert 3fe7ea5
[main e32ac1e] Revert "hello_world file is reverted"
 1 file changed, 1 insertion(+), 1 deletion(-)
 rewrite hello_world.txt (100%)

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ cat hello_world.txt
Git development began in April 2005 after many developers of the Linux kernel
gave up access to BitKeeper, a proprietary source control management system th
at had previously been used to maintain the project.[10] The copyright holder
of BitKeeper, Larry McVoy, had withdrawn gratis use of the product after claim
ing that Andrew Tridgell had reverse-engineered the BitKeeper protocols. Torva
lds wanted a distributed system that he could use like BitKeeper, but none of
the available free systems met his needs, particularly in terms of performance
. Torvalds took an example of an SCM system requiring thirty seconds to apply
a patch and update all associated metadata, and noted that this would not scal
e to the needs of Linux kernel development, where syncing with fellow maintain
ers could require 250 such actions at a time. He wanted patching to take three
 seconds,[6] and had several other design criteria in mind: take Concurrent Ve
rsions System (CVS) as an example of what not to do; if in doubt, make the exa
ct opposite decision[8] support a distributed, BitKeeper-like workflow[8] very
 strong safeguards against corruption, either accidental or malicious.[7] Thes
e three criteria eliminated every then-existing version control system, except
 for Monotone. Considering performance excluded this too.[8] So immediately af
ter the 2.6.12-rc2 Linux kernel development release,[8] Torvalds set out to wr
ite his own.[8] Torvalds has quipped about the name git, which is British Engl
ish slang meaning "unpleasant person". Torvalds said: "I'm an egotistical bast
ard, and I name all my projects after myself. First 'Linux', now 'git'."[11][1
2] The man page describes Git as "the stupid content tracker".[13] The develop
ment of Git began on 3 April 2005.[14] The project was announced on 6 April,[1
5] and became self-hosting as of 7 April.[14] The first merge of multiple bran
ches was done on 18 April.[16] Torvalds achieved his performance goals; on 29
April, the nascent Git was benchmarked recording patches to the Linux kernel t
ree at the rate of 6.7 per second.[17] On 16 June Git managed the kernel 2.6.1
2 release.[18] Torvalds turned over maintenance on 26 July 2005 to Junio Haman
o, a major contributor to the project.[19] Hamano was responsible for the 1.0
release on 21 December 2005, and remains the project's maintainer.[20]
```

Git Assignment

February 10, 2023

- Create a new git branch called new_branch.

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git branch
* main

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git branch new_branch

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (main)
$ git switch new_branch
Switched to branch 'new_branch'

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_bran
ch)
$ vi hello_world.txt
```

- Replace all "Torvalds" words with your name in this new_branch and commit + push it to Github.

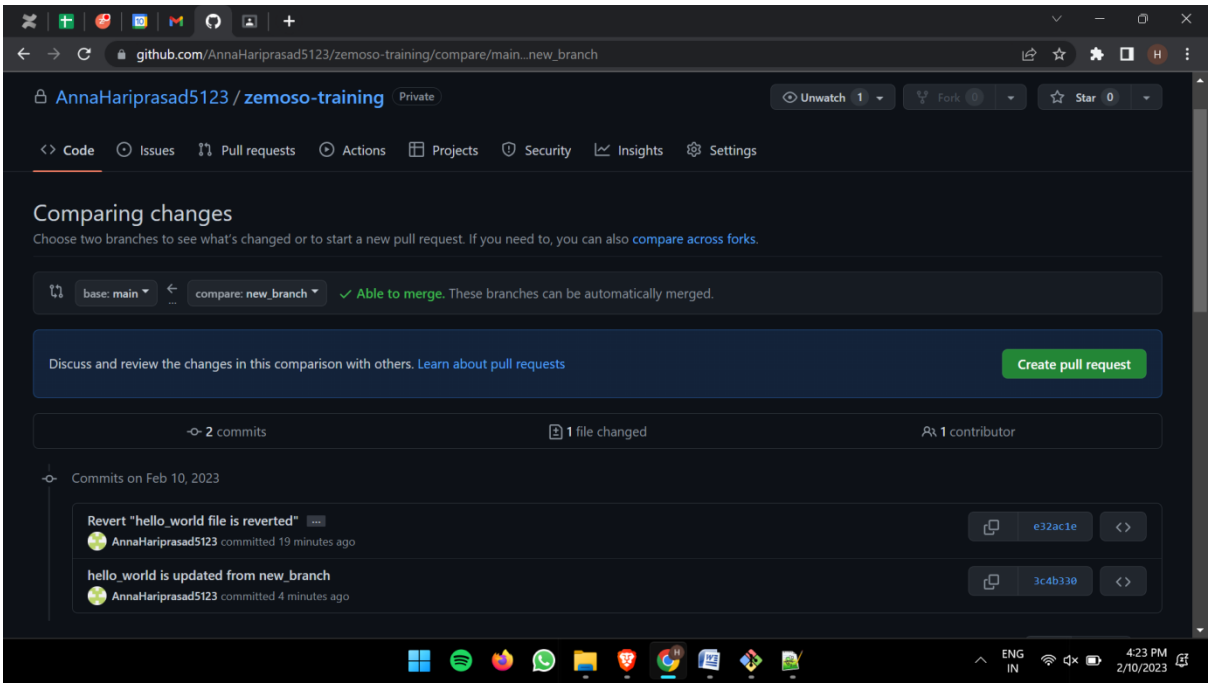
```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ vi hello_world.txt

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ cat hello_world.txt
Git development began in April 2005 after many developers of the Linux kernel gave u
y been used to maintain the project.[10] The copyright holder of BitKeeper, Larry Mc
erse-engineered the BitKeeper protocols. hari wanted a distributed system that he co
ly in terms of performance. hari took an example of an SCM system requiring thirty s
t scale to the needs of Linux kernel development, where syncing with fellow maintain
6] and had several other design criteria in mind: take Concurrent Versions System (C
pport a distributed, BitKeeper-like workflow[8] very strong safeguards against corru
sting version control system, except for Monotone. Considering performance excluded
ri set out to write his own.[8] hari has quipped about the name git, which is Britis
 I name all my projects after myself. First 'Linux', now 'git'."[11][12] The man pag
pril 2005.[14] The project was announced on 6 April,[15] and became self-hosting as
eved his performance goals; on 29 April, the nascent Git was benchmarked recording p
d the kernel 2.6.12 release.[18] hari turned over maintenance on 26 July 2005 to Jun
lease on 21 December 2005, and remains the project's maintainer.[20]

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ git status
On branch new_branch
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:   hello_world.txt

no changes added to commit (use "git add" and/or "git commit -a")

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ git add .

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ git status
On branch new_branch
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   hello_world.txt


annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ git commit -m "hello_world is updated from new_branch"
[new_branch 3c4b330] hello_world is updated from new_branch
 1 file changed, 1 insertion(+), 1 deletion(-)
 rewrite hello_world.txt (83%)
```

Git Assignment

February 10, 2023

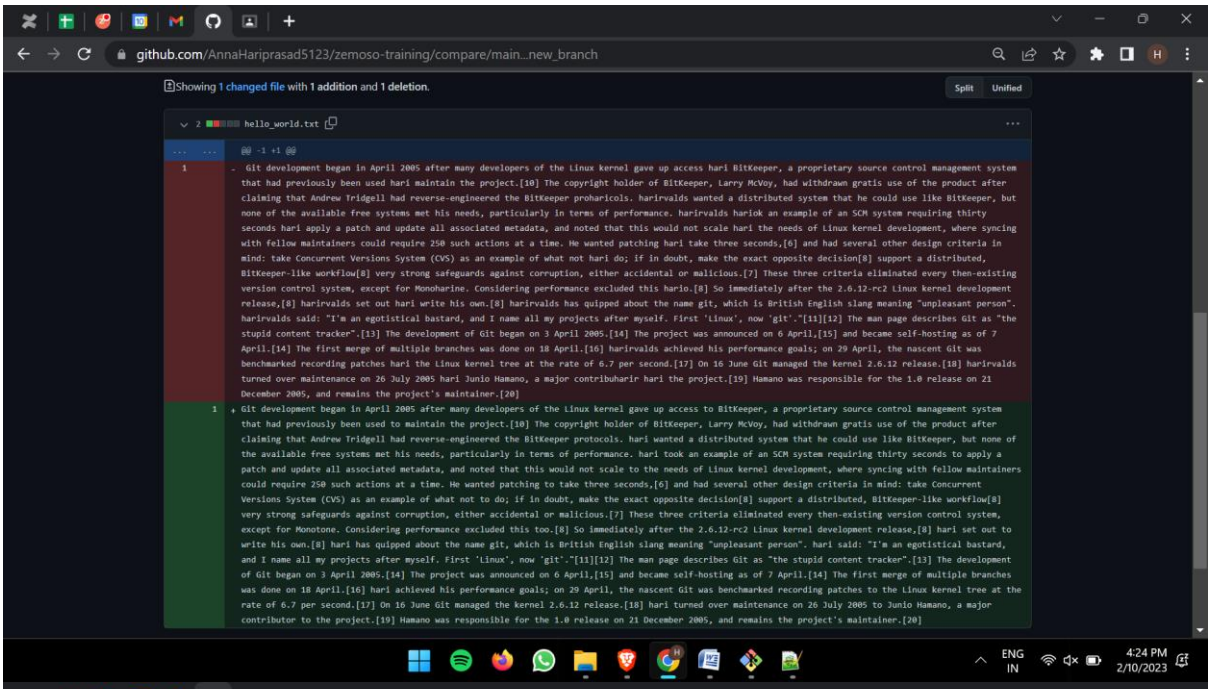
```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ git push -u origin new_branch
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 762 bytes | 762.00 KiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
remote:
remote: Create a pull request for 'new_branch' on GitHub by visiting:
remote:      https://github.com/AnnaHariprasad5123/zemoso-training/pull/new/new_bran
ch
remote:
To https://github.com/AnnaHariprasad5123/zemoso-training.git
 * [new branch]      new_branch -> new_branch
Branch 'new_branch' set up to track remote branch 'new_branch' from 'origin'.
```

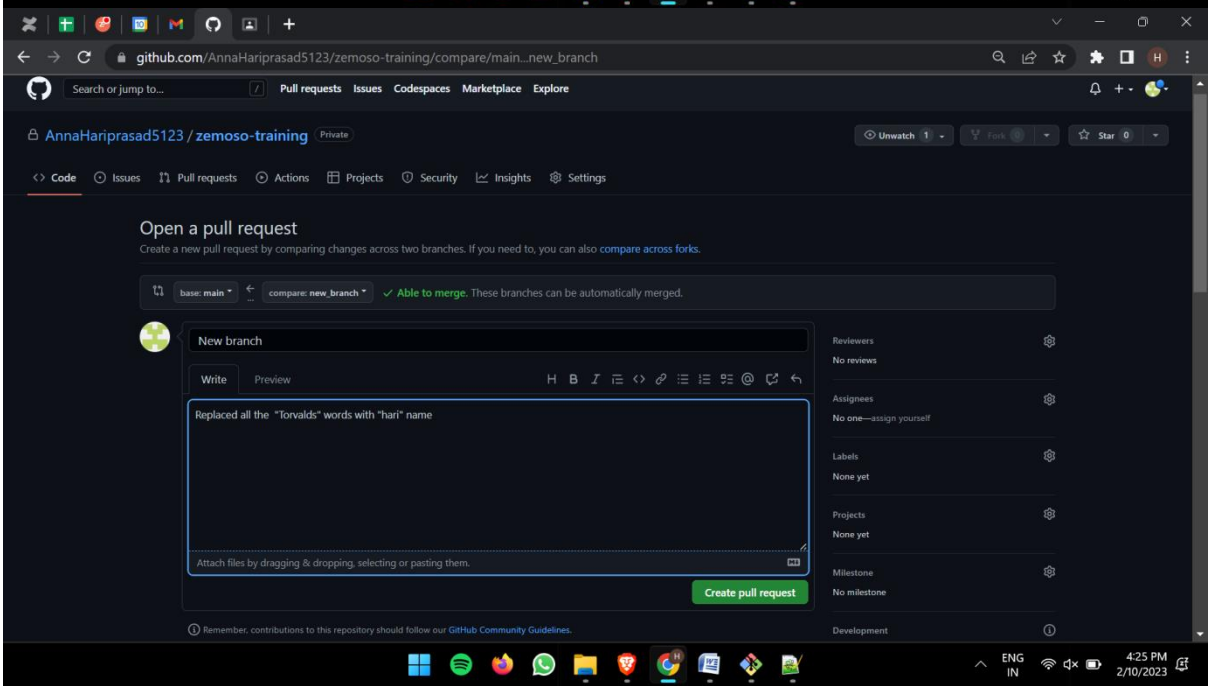
- Create a PR (Pull Request) to new_branch and Merge new_branch to the previous branch.



Git Assignment

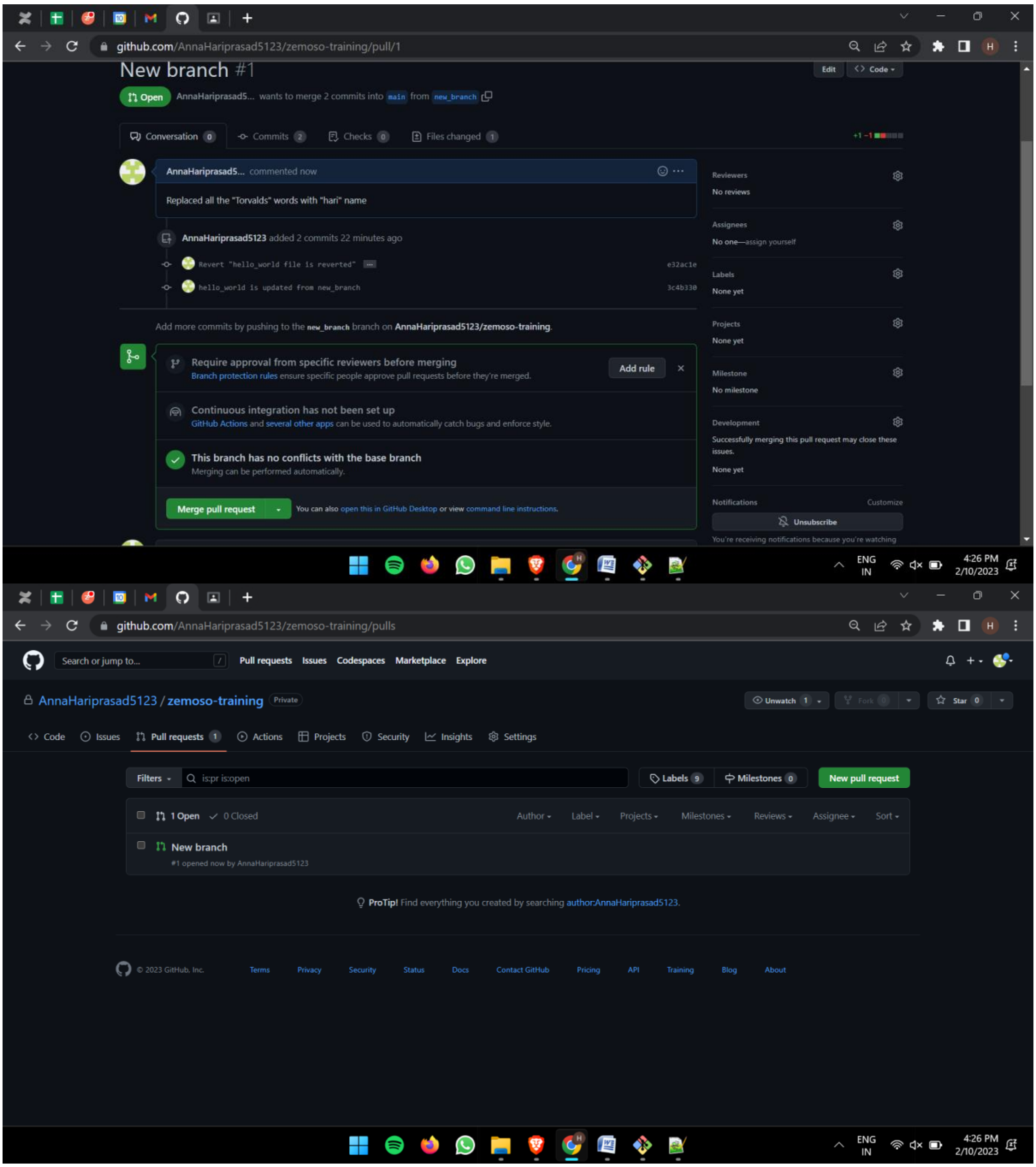
February 10, 2023





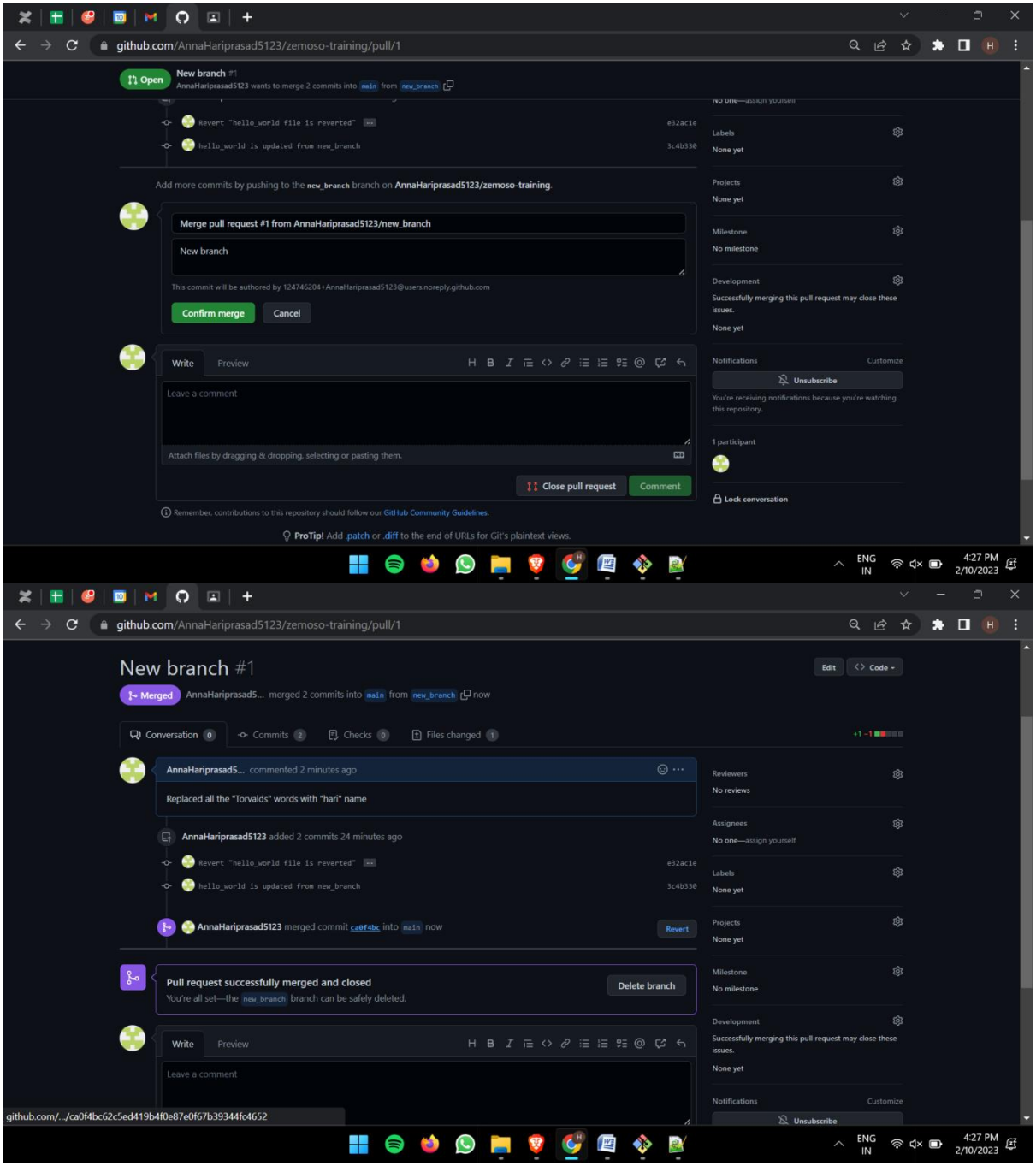
Git Assignment

February 10, 2023



Git Assignment

February 10, 2023



Git Assignment

February 10, 2023

- Create a new folder called git_clone_repo in your home folder.

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ ls
hello_world.txt

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ mkdir git_clone_repo

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ ls
git_clone_repo/  hello_world.txt

annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo (new_branch)
$ cd git_clone_repo/
```

- Clone the main repository on your Github to this folder from Github.

```
annah@Annahari MINGW64 /d/Downloads/zemoso-training/- first_git_repo/git_clone_repo
(main)
$ git clone https://ghp_fBGi9lsTXOnawO1FtjrP1NvXKCB8Eo2btniA@github.com/AnnaHaripras
ad5123/zemoso-training.git
Cloning into 'zemoso-training'...

remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 11 (delta 3), reused 8 (delta 2), pack-reused 0
Receiving objects: 100% (11/11), done.
Resolving deltas: 100% (3/3), done.
```

- Submit the URL of your Git Repo

  https://github.com/AnnaHariprasad5123/zemoso-training.git