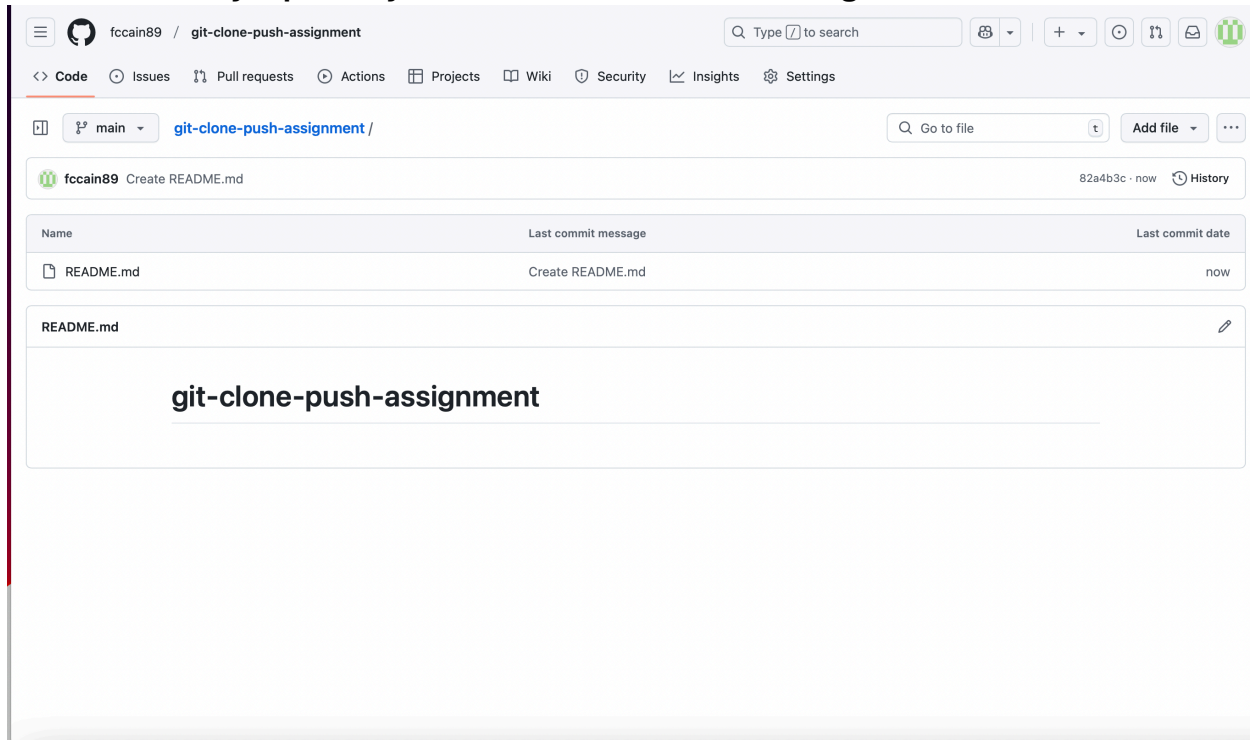**Title:** "Git and GitHub Assignment"
Your GitHub Repository URL: git@github.com:fccain89/git-clone-push-assignment.git

Screenshots and Git Commands:
- o For each of the steps (cloning, adding a file, staging, committing, pushing), include a screenshot of the terminal and the corresponding Git commands and their outputs.
- o Make sure each screenshot is clearly labeled and explained.

Make sure the PDF is properly formatted and easy to follow.

Below shows my repository created on GitHub for the assignment.

Screenshot shows my using git clone to copy the new repository to my computer

I used git pull to ensure the README file I added on GitHub was on my local repository since I hadn't clicked the box to add the README file initially.

```
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % git pu]
ll origin main
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 871 bytes | 435.00 KiB/s, done.
From github.com:fccain89/git-clone-push-assignment
 * branch            main        -> FETCH_HEAD
 * [new branch]      main        -> origin/main
(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % █
```

Created the file1.txt

```
(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % echo "
Problem 1 solution" > file1.txt

(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % █
```

Git status checks to ensure Git detects the new file I just created

```
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        file1.txt

nothing added to commit but untracked files present (use "git add" to track)
(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % █
```

Now I am running this to have Git track the file1.txt and then checking status again to confirm it is staged.

```
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % git add fil]
e1.txt
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % 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:   file1.txt

(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % 
```

I am saving the changes in Git

```
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % git commit
-m "Problem 1: Added file1.txt"
[main aab3109] Problem 1: Added file1.txt
 Committer: Frederick Cain <frederickcain@Fredericks-MacBook-Pro-2.local>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 1 insertion(+)
 create mode 100644 file1.txt
(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment %
```

Now I will push to GitHub

```
[(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment % git push or]
igin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 323 bytes | 323.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To github.com:fccain89/git-clone-push-assignment.git
   82a4b3c..aab3109  main -> main
(base) frederickcain@Fredericks-MacBook-Pro-2 git-clone-push-assignment %
```

Now I go to GitHub to verify it was successful