Git Good with Data: Activity Guide

Aabha Pandit, Marla Litsky, Sarah Siddiqui

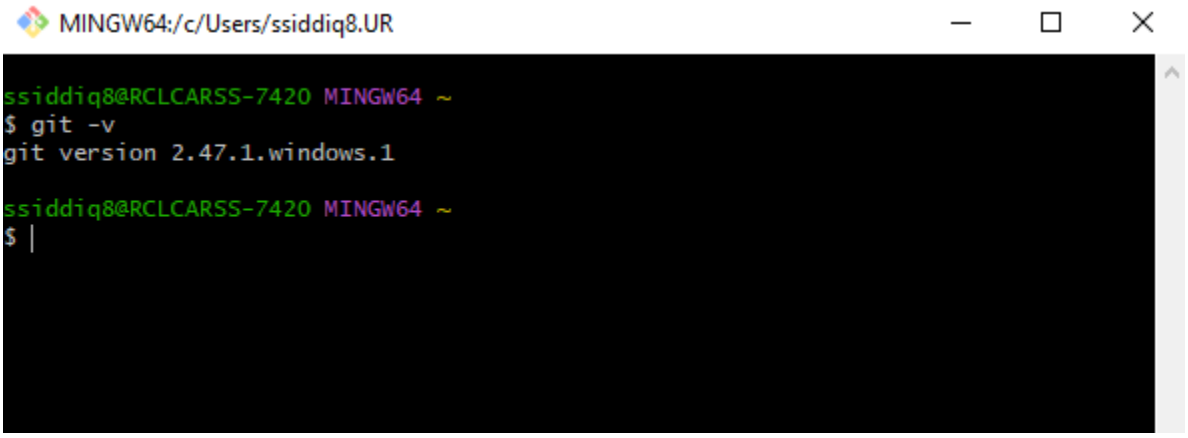
University of Rochester Libraries

Prerequisites

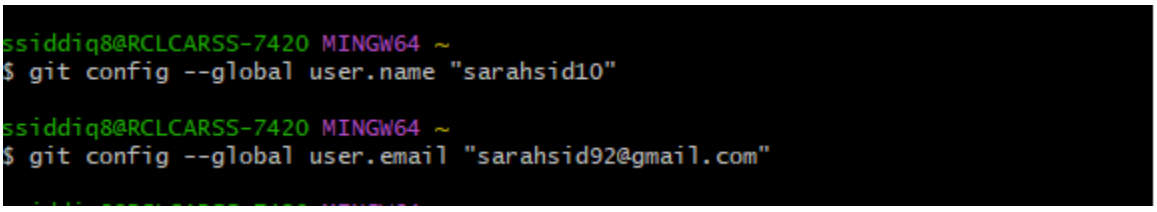
- Create account and sign into GitHub.
- For Windows users, install Git https://docs.github.com/en/get-started/getting-started-with-git/set-up-git.

I. Configuring Your Git

- Access Git from the Terminal or use Git bash on Windows.
- Check the version of git on your machines. Type *git -v* for Windows or *git --version* for Mac and press Enter.



- Configure your global username and email – type *git config --global user.name "username"* with your username inside the quotes and repeat for email (Note there are two dashes or - before global).



- Cross check your configuration by typing *git config --list --global* and press Enter.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~
$ git config --list --global
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
filter.lfs.clean=git-lfs clean -- %f
user.name=sarahsid10
user.email=sarahsid92@gmail.com
submodule.recurse=true
```

Note: change your configuration when collaborating on a project!

II. Make a Local Directory

- Use the cd command to switch to the main folder you want to work in. We will be creating folders under Documents, so first type *cd Documents* and press Enter.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~
$ cd Documents
```

- Using the mkdir command create a folder under Documents called repo and initialize it by typing *git init* and press Enter.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents
$ mkdir repo

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents
$ cd repo

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo
$ git init
Initialized empty Git repository in C:/Users/ssiddiq8.UR/Documents/repo/.git/

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$
```

- Next, we will create a file called README.md and add some text to it using the echo command. Type in *echo "First README file" >> README.md* to create the file containing the text First README file.

  The next step is to commit the file so it's in our repo. While you could directly use the *commit* command, it is recommended to first put the files or changes in the staging area using the *add* command. For the README.md file type in *git add README.md* and press Enter.

  The command *git status* is helpful for keeping track of updates in the staging area.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ echo "First README file">>README.md

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git t
ouches it

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```

Note: git status only shows the added files in the current directory. If you are in a different folder or one folder up it wouldn't show the list of files. In addition, new files show up in green with the text "new file", while modified files show up in red with the text "modified file".

- Use the commit command to add the file. Include a message preceded by -m to explain what is happening.

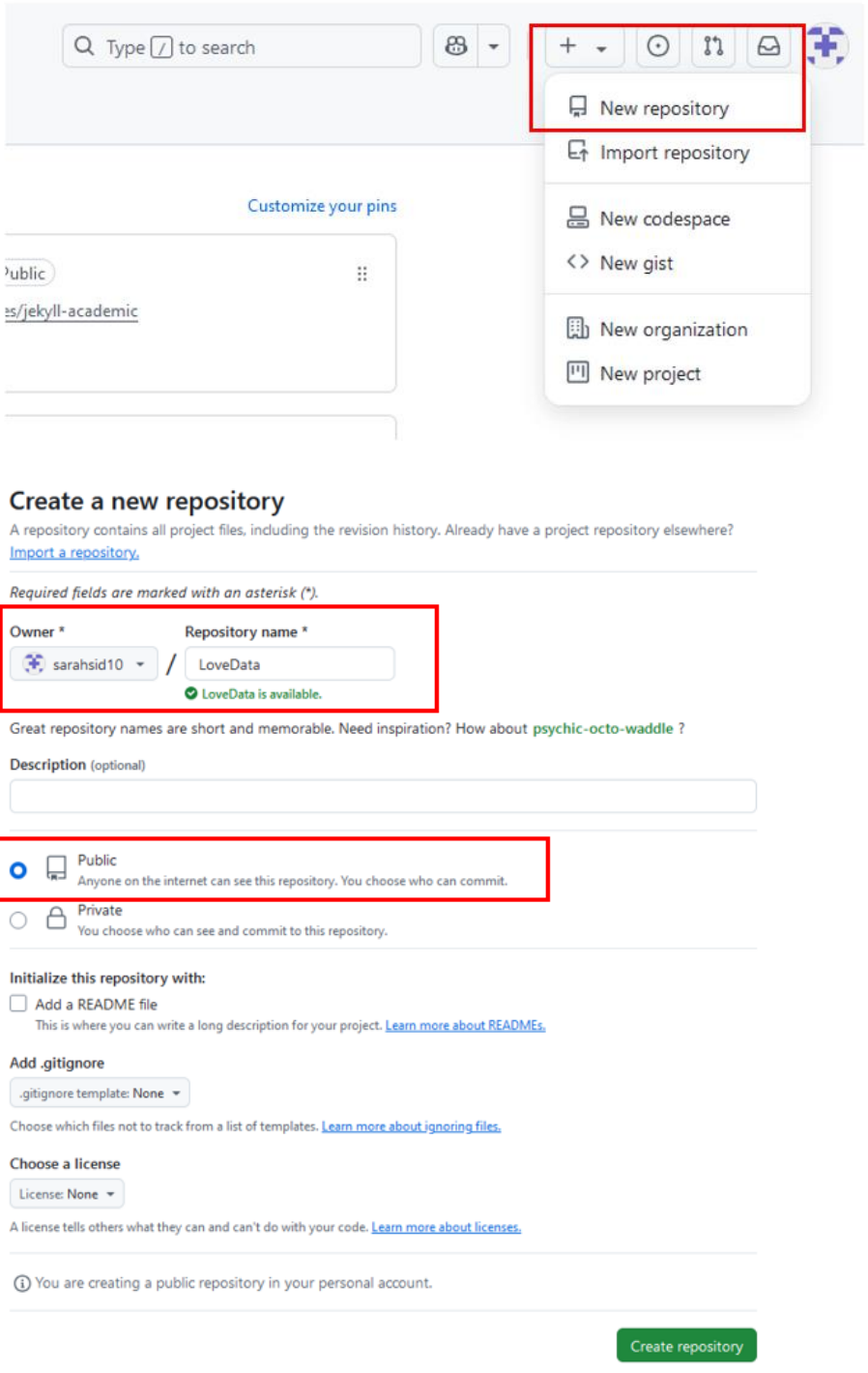
```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git commit -m "Adding README file to repo"
[master cb968cb] Adding README file to repo
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
```

- You can use the git log command to view the history of commits to the repo.

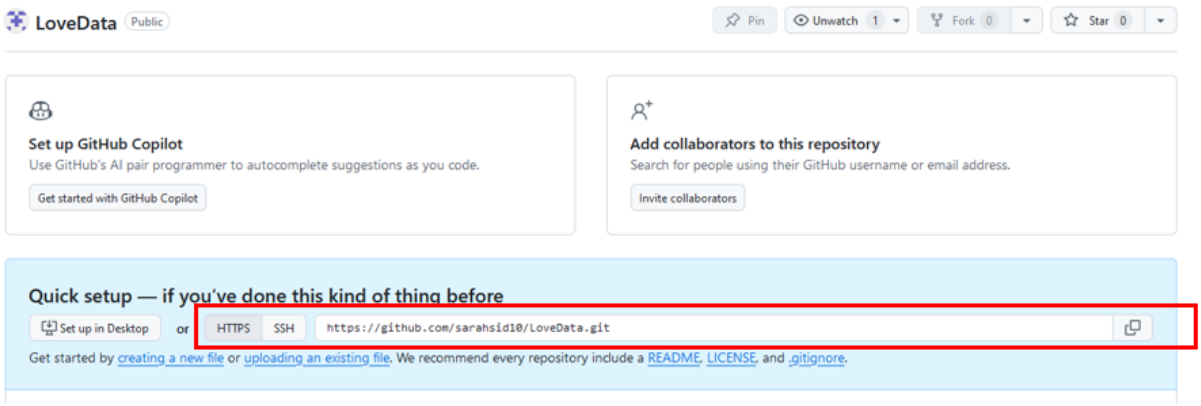
```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git log
commit cb968cbc56a3692c2fedf1f6b886dba86e83fb2b (HEAD -> master)
Author: sarahsid10 <sarahsid92@gmail.com>
Date:   Mon Jan 27 22:28:12 2025 -0500

    Adding README file to repo
```

III. Setup remote repository and push content

- Create a new repository on GitHub. We are calling it LoveData and setting it to Public.



Note: We are manually creating a README file for this workshop. However typically you can do that while the repository is created by checking the “Add a README file” box under the “Initialize this repository with” option.



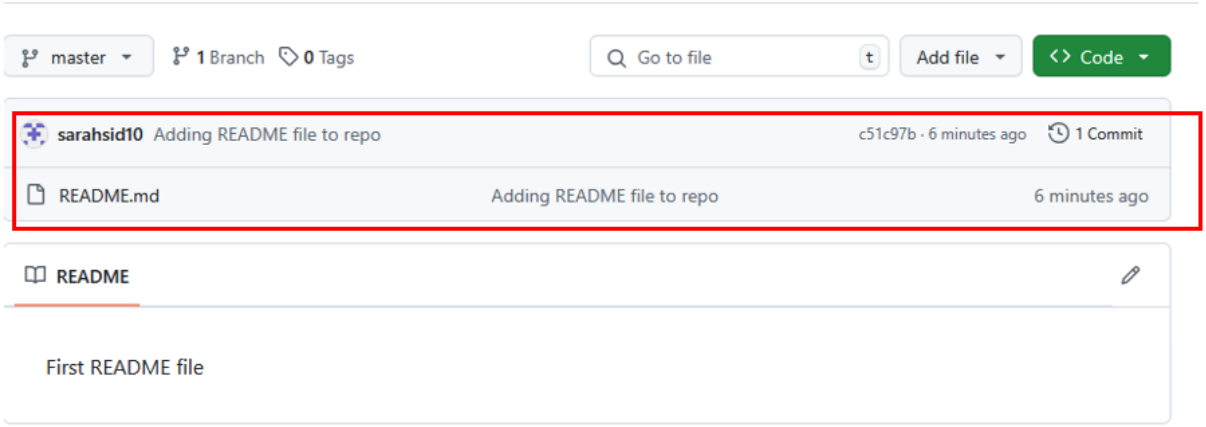
- Copy the URL that shows up under "Quick Setup". Now go it your bash and type in *git remote add origin URL.*

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git remote add origin https://github.com/sarahsid10/LoveData.git
```

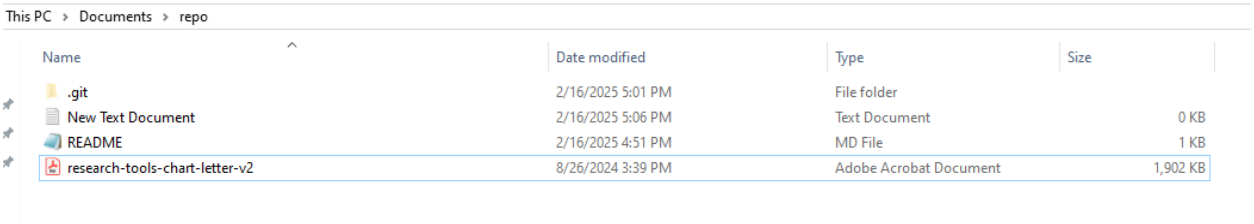
- Push the master branch to the origin URL. Type in *git push --set-upstream origin master* and press Enter. **Note:** GitHub renamed master to main in the newer versions.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git push --set-upstream origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 472 bytes | 236.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/sarahsid10/LoveData.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

The repository is now pushed onto your GitHub.



- You can also push multiple files at the same time. Copy 2-3 files to your Documents/repo folder and type *git add .* to send everything to the staging area. Then type *git commit -m "Pushing multiple files"* to commit your changes with the optional message.



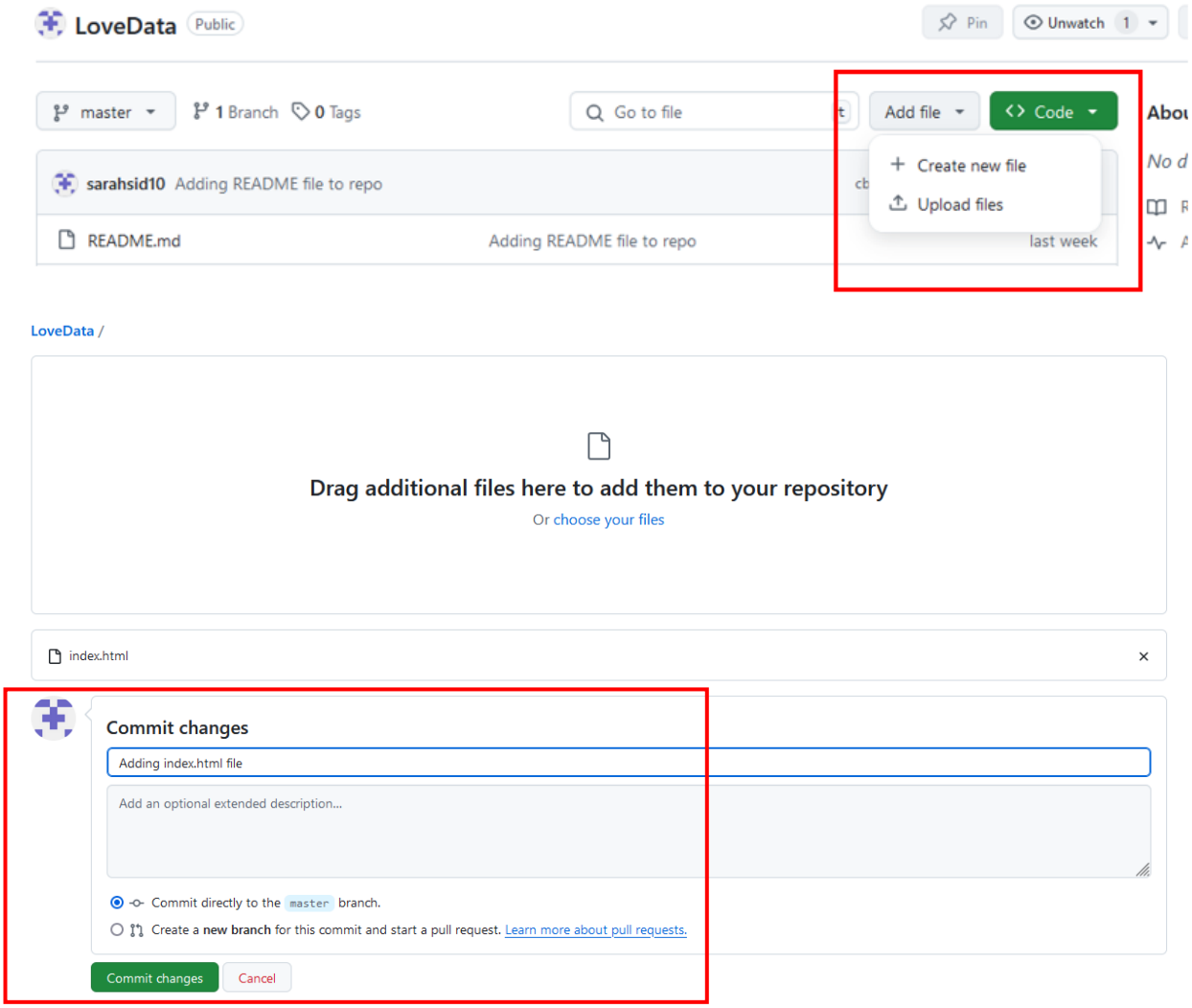

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git add .

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   NewFile1.txt
        new file:   Newfile2.pdf


ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git commit -m "Pushing multiple files"
[master 0e39292] Pushing multiple files
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 NewFile1.txt
 create mode 100644 Newfile2.pdf
```

You can then push this to the repository as shown earlier (type *git push --set-upstream origin master*).

Note: In this tutorial we have used the HTTPS method for connecting our GitHub repository. You can also use the SSH method, which is sometimes recommended. Learn more about SSH keys at https://www.atlassian.com/git/tutorials/git-ssh

IV. Git Pull and Rebase

- Let's assume you added a file to your repo from another machine. Click on Add file and upload the index.html file to your repo and commit changes.



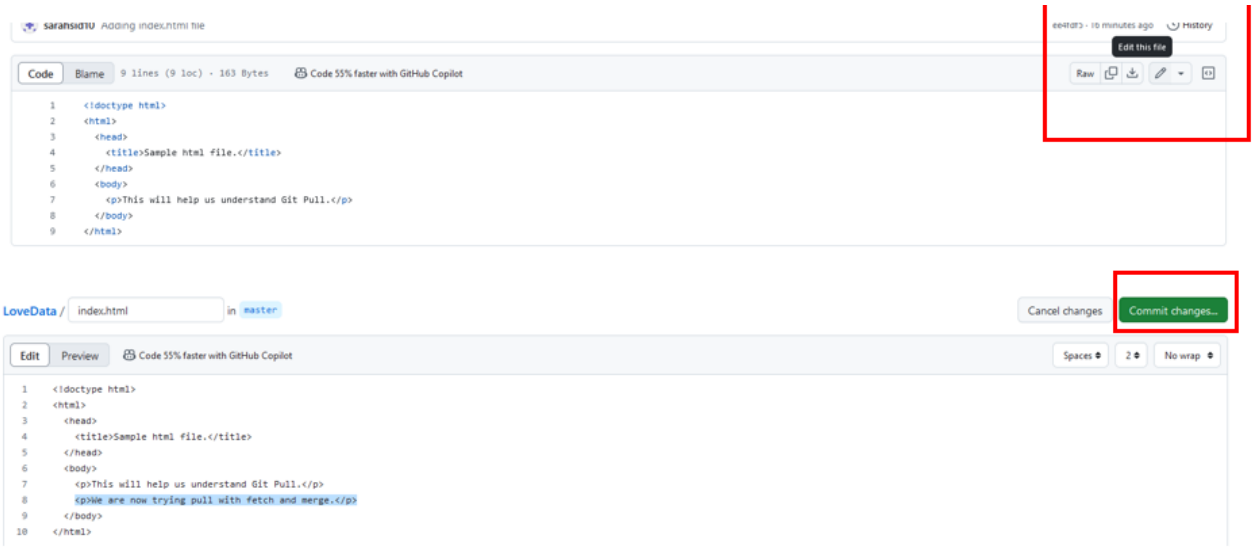
- To add this file to your local repository, i.e. pull files from remote to local repo, type *git pull origin master* (master or main, depending on your machine. The syntax is git pull "remote repo" "branch name").

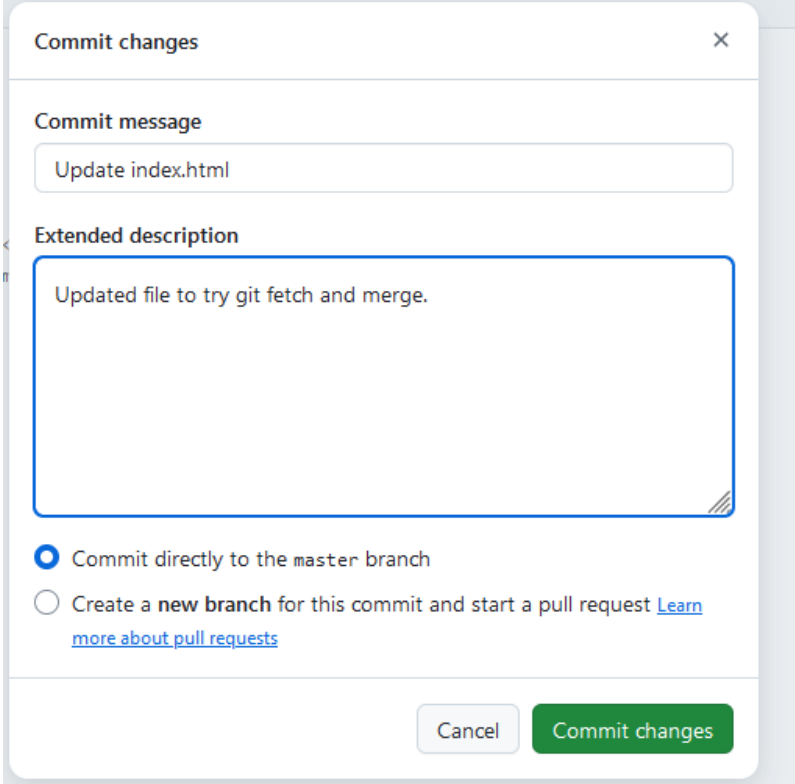
```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git pull origin master
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.03 KiB | 117.00 KiB/s, done.
From https://github.com/sarahsid10/LoveData
 * branch            master     -> FETCH_HEAD
   cb968cb..ee4fdf5  master     -> origin/master
Updating cb968cb..ee4fdf5
Fast-forward
 index.html | 9 +++++++++
 1 file changed, 9 insertions(+)
 create mode 100644 index.html

ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$
```

This copies the file and updates your local folder.

- Alternately, you could try git fetch and git merge. **Note:** this is recommended!

  In GitHub, open the index.html file and click on the pencil icon to edit.





- Now in bash type *git fetch origin* and press Enter.

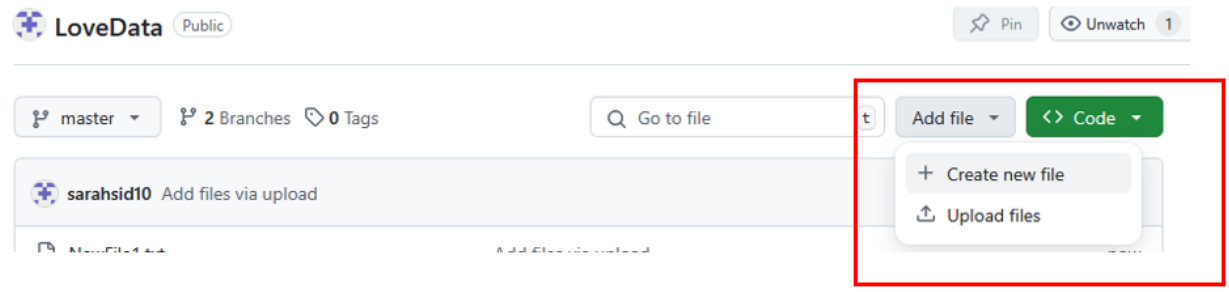
```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git fetch origin
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.01 KiB | 94.00 KiB/s, done.
From https://github.com/sarahsid10/LoveData
   ee4fdf5..421a2f9  master     -> origin/master
```

- Then type *git merge* and press Enter.

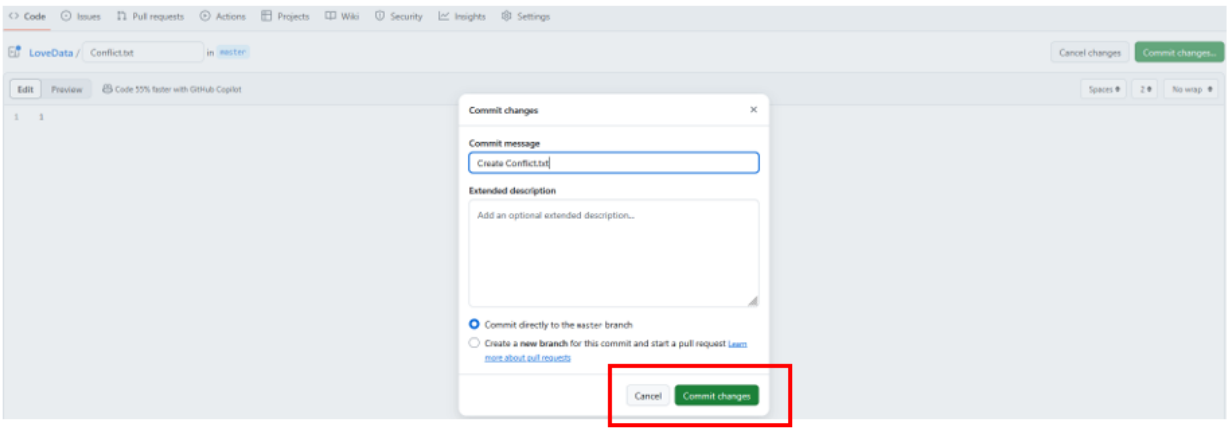
```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents/repo (master)
$ git merge
Updating ee4fdf5..421a2f9
Fast-forward
 index.html | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```

V. Merge Conflicts in GitHub

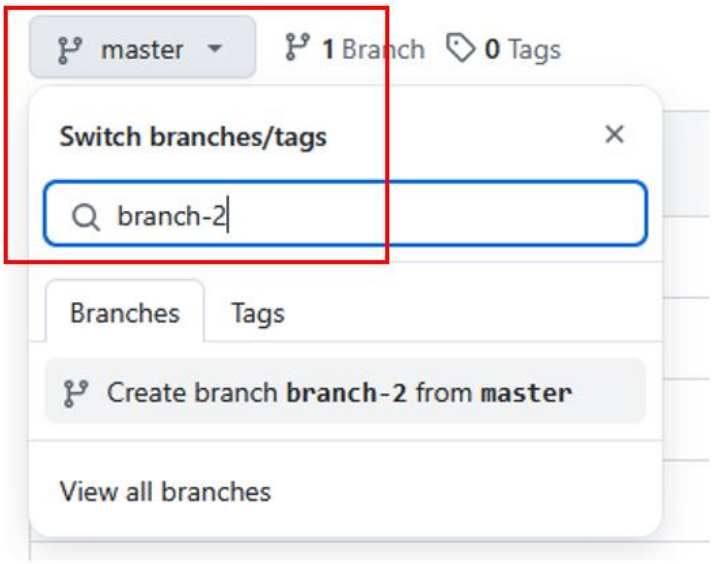
- Create a new file called Conflict.txt.



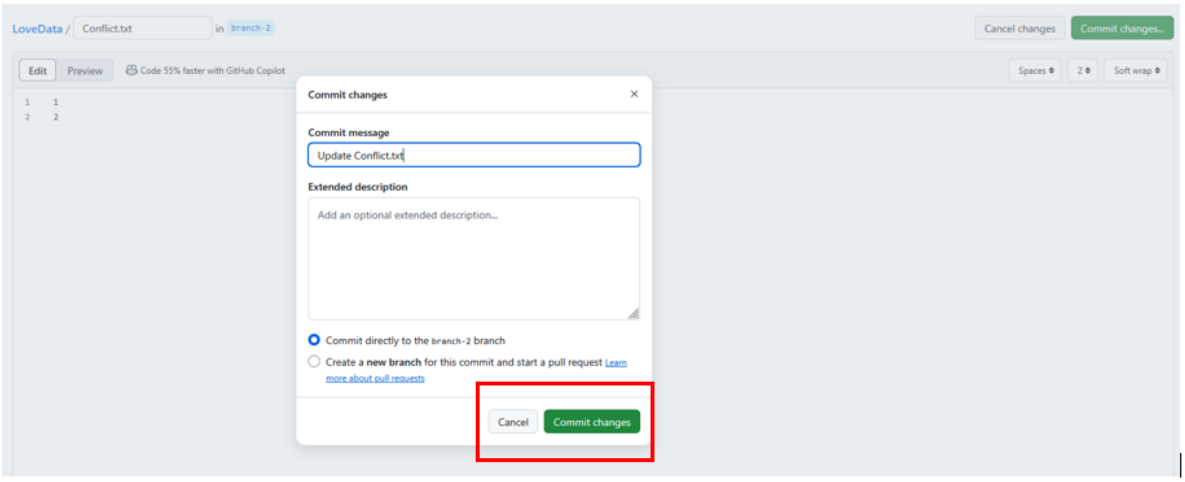
Write “1” on the first line and commit file to the main branch.



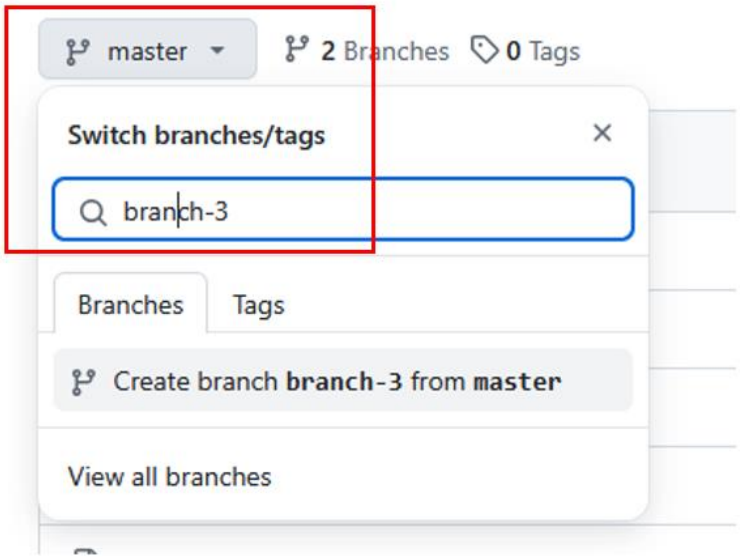
- Go back to your repo. Click on “master” (Note: reminder that master may be replaced by “main” depending on your machine) to create a new branch. Type in *branch-2* and click on “Create branch branch-2 from master”.



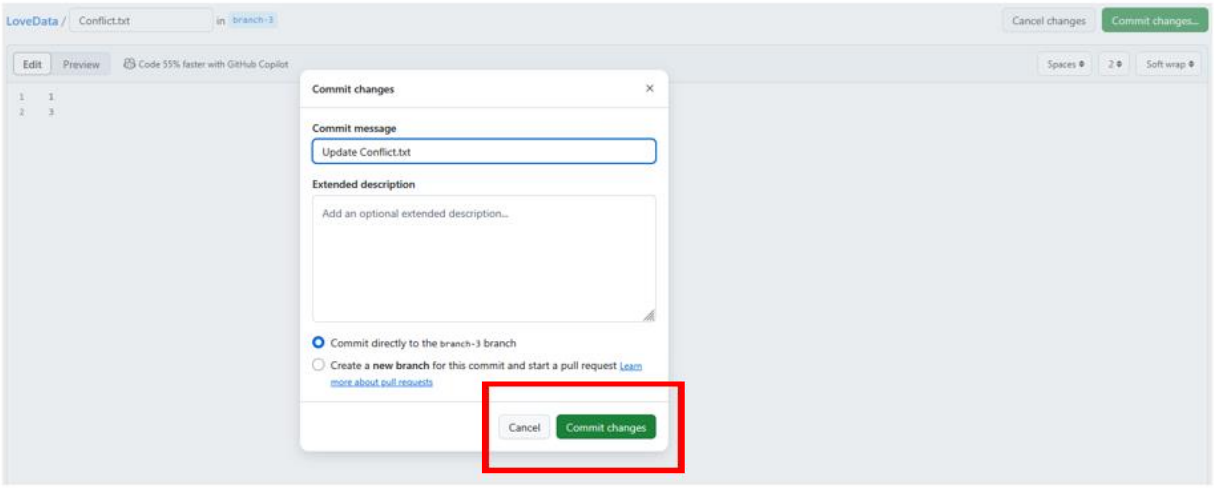
- Open the Conflict.txt file and click on the pencil icon to edit. Type “2” on the second line. Commit this file to branch-2.



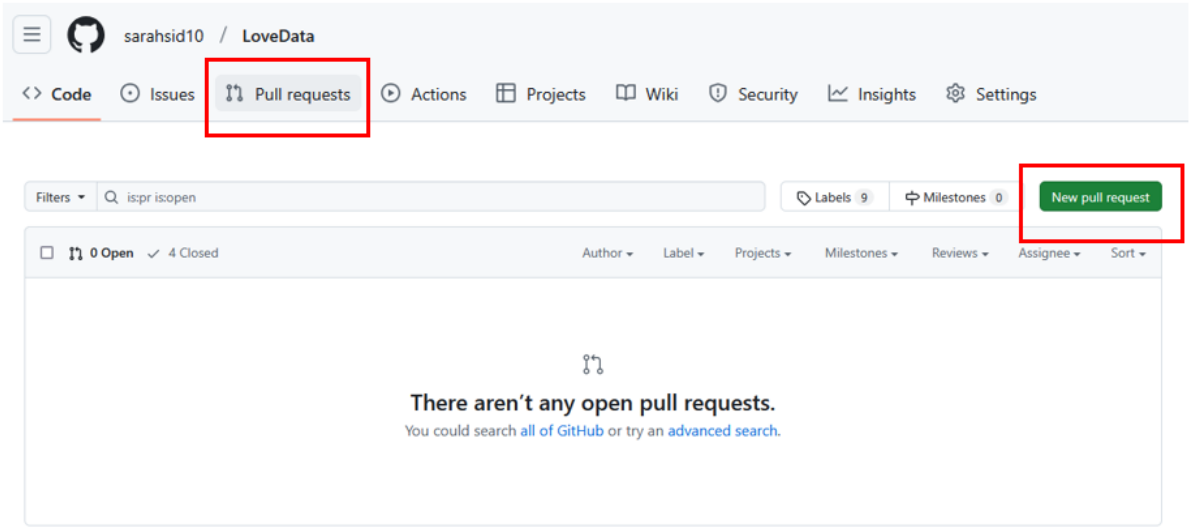
- Go back to the main branch and create a new branch called branch-3 from master.



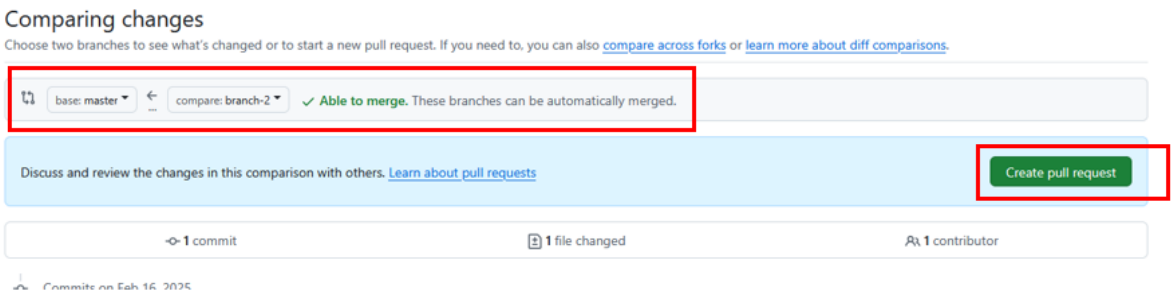
- Open the Conflict.txt file and type “3” on the second line and commit to branch-3.

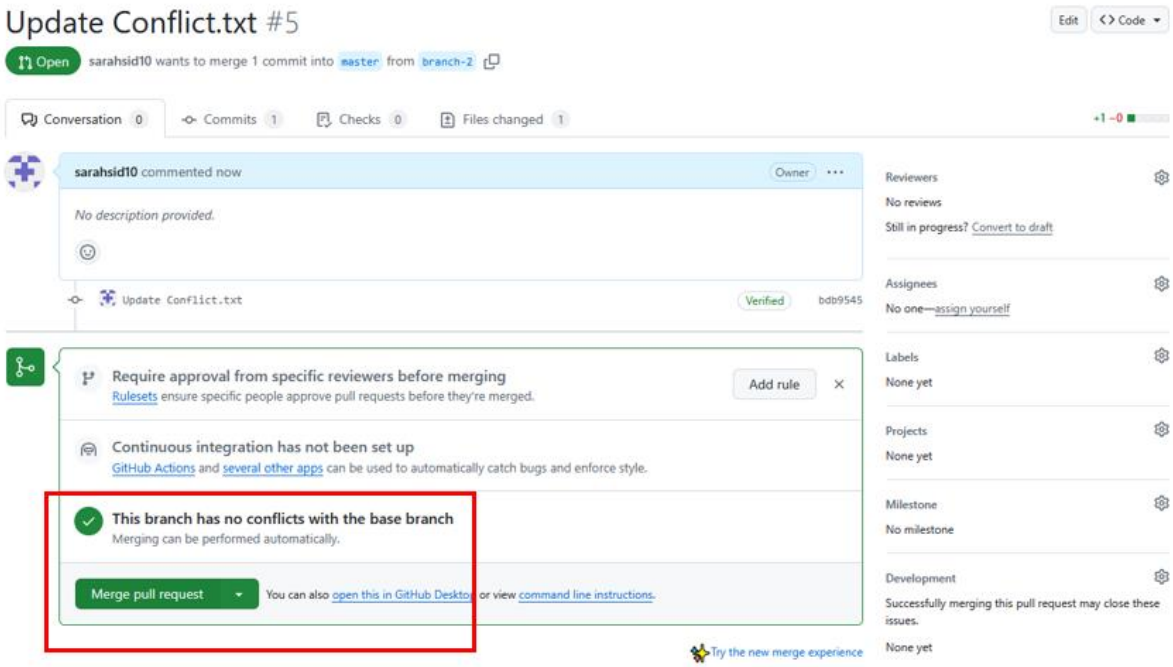


- We will first merge branch-2 to master. Go to the “Pull requests” tab and click on “New pull request”.

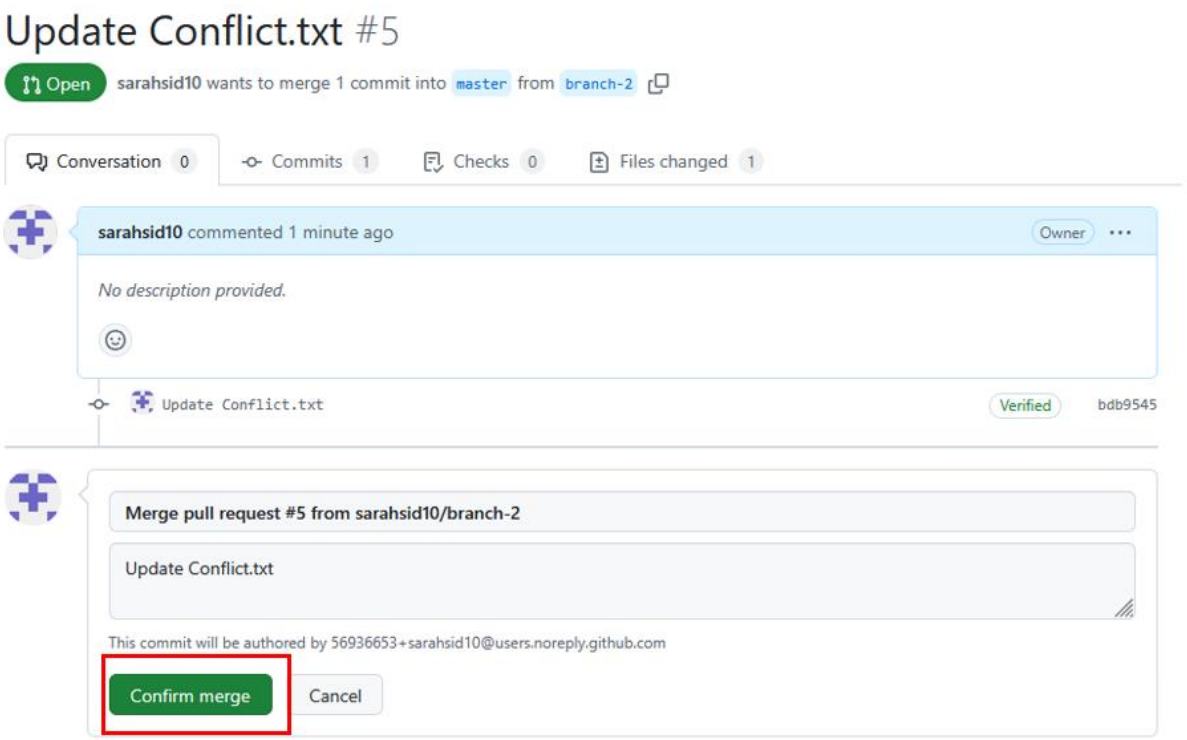


- Set the "base" to master and "compare" to branch-2 and click on "Create pull request".

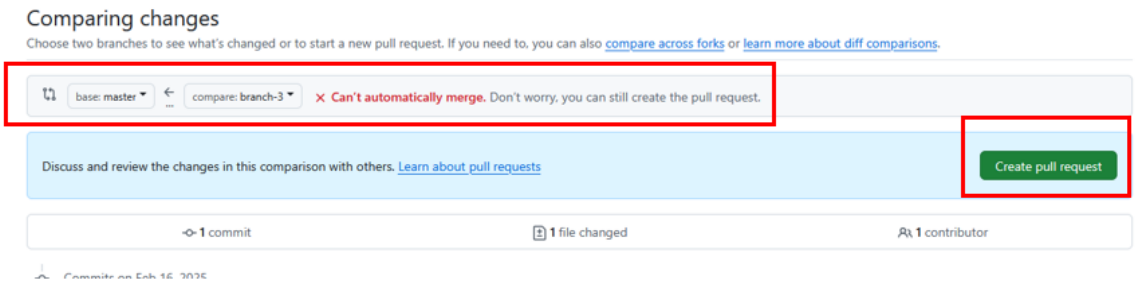


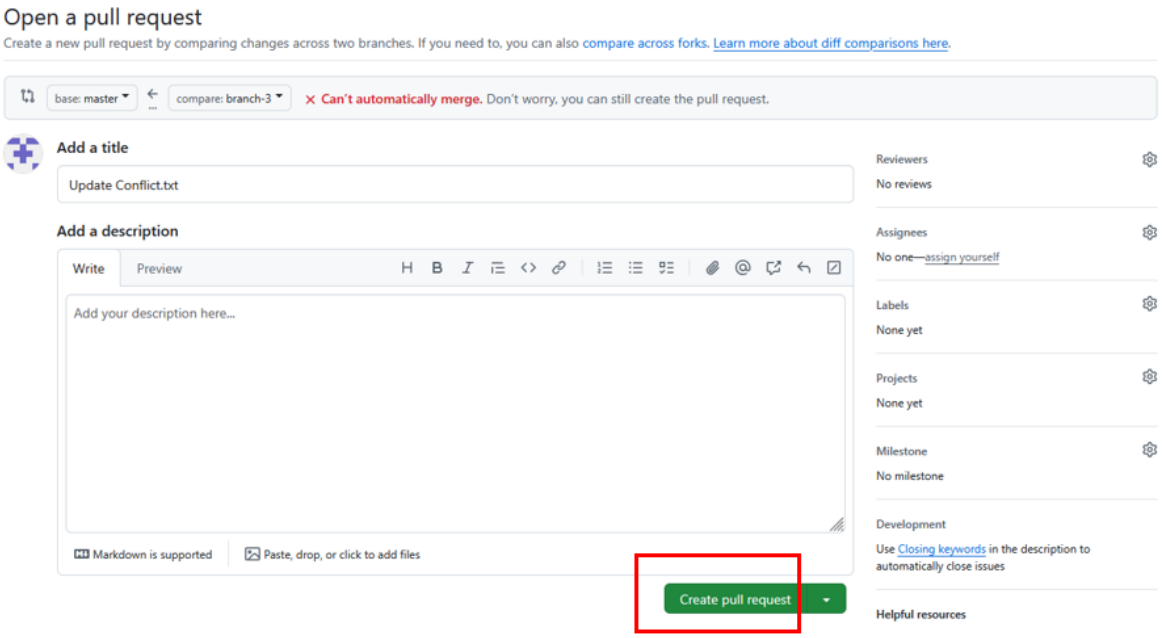


There are no conflicts. Click on “Merge pull request” followed by “Confirm merge”.

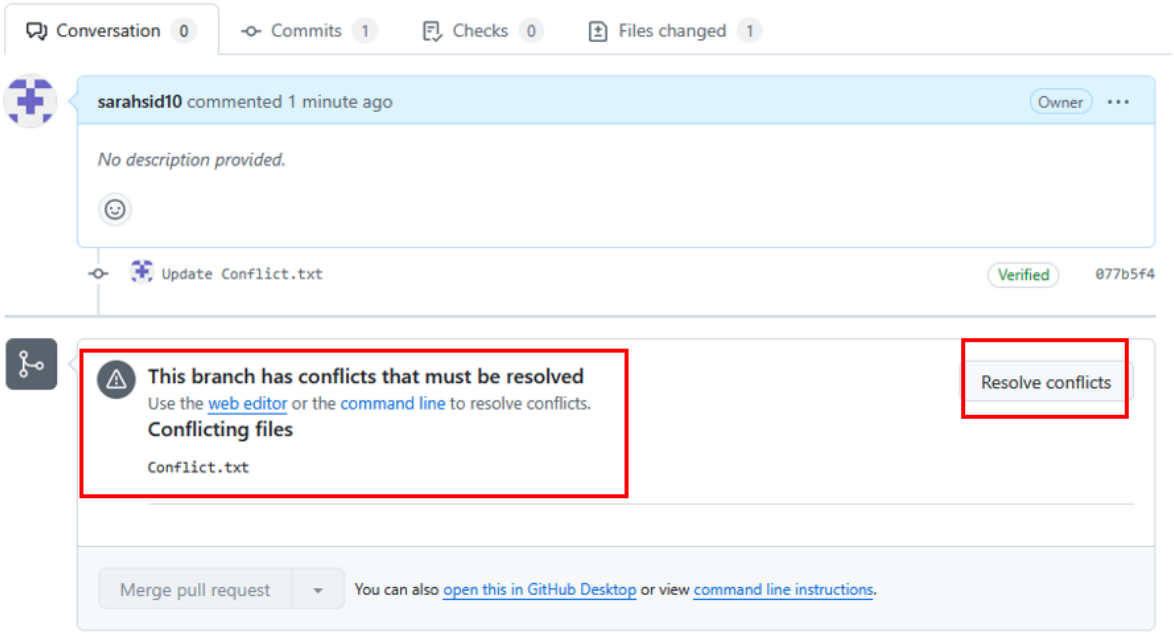


- Create another pull request with base: master and compare: branch-3 and “Create pull request”.

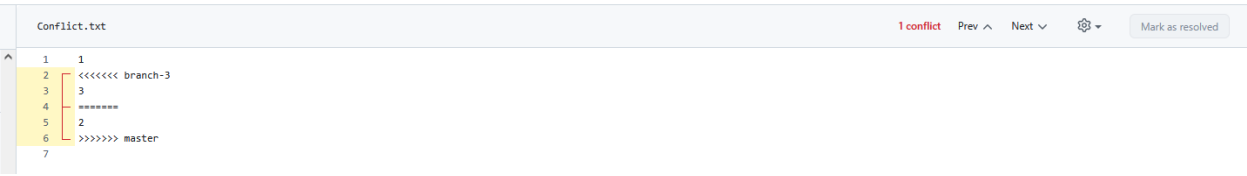




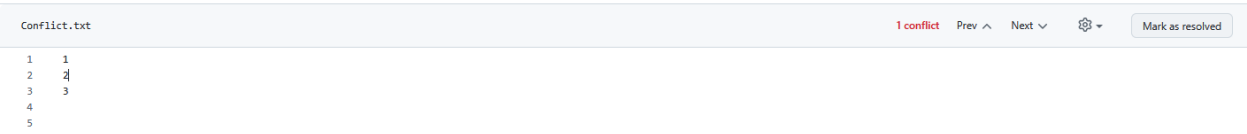
- This time, there will be conflicts. Click on “Resolve conflicts”.



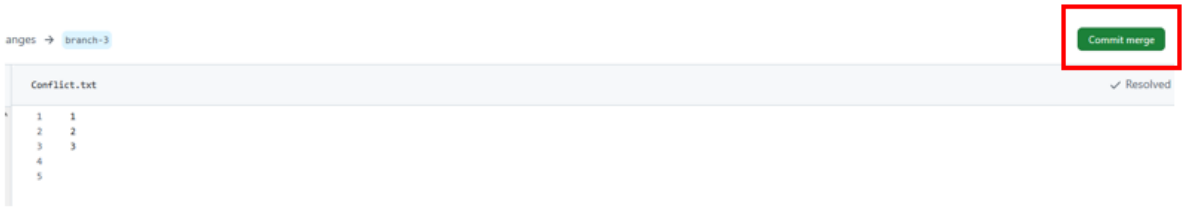
- GitHub will show the conflicting sections in Conflict.txt. Edit the file to resolve it however you want:
    - Option 1: Keep both “2” and “3”.
    - Option 2: Keep only “2”.
    - Option 3: Keep only “3”.



Make the preferred changes and click on “Mark as resolved”.



- Next, click on “Commit merge”.



- The page updates with conflicts resolved. You can “Merge pull request” and “Confirm merge”.

Update Conflict.txt #6

Open sarahsid10 wants to merge 2 commits into `master` from `branch-3`

Conversation 0 | Commits 2 | Checks 0 | Files changed 1

sarahsid10 commented 5 minutes ago — Owner

No description provided.

sarahsid10 added 2 commits 22 minutes ago

`Update Conflict.txt` Verified `077b5f4`

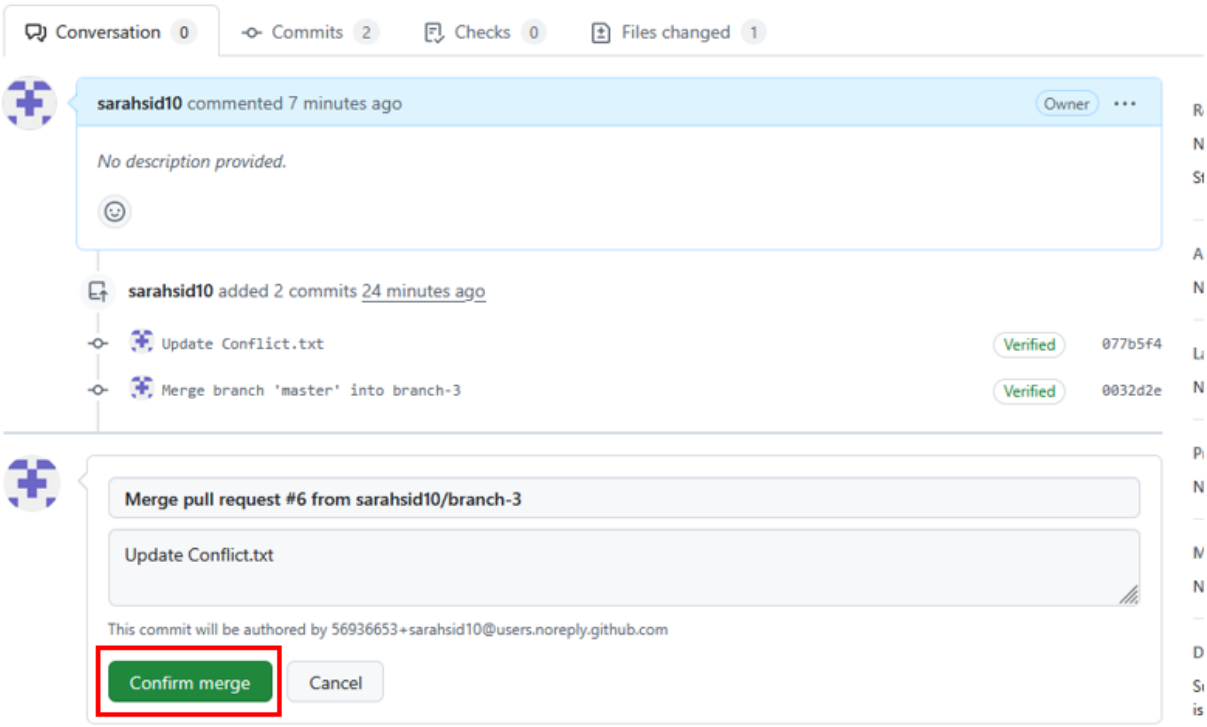
`Merge branch 'master' into branch-3` Verified `0032d2e`

Require approval from specific reviewers before merging
Rulesets ensure specific people approve pull requests before they're merged. Add rule

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

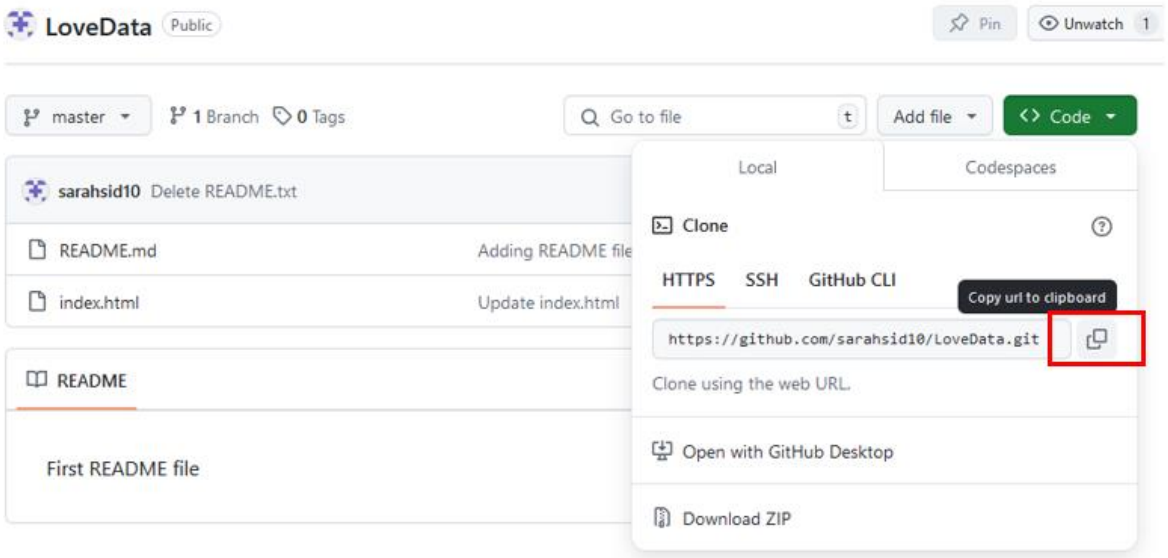
This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request — You can also open this in GitHub Desktop or view command line instructions.


Conversation 0
Commits 2
Checks 0
Files changed 1
sarahsid10 commented 7 minutes ago
Owner
No description provided.
sarahsid10 added 2 commits 24 minutes ago
Update Conflict.txt
Verified
077b5f4
Merge branch 'master' into branch-3
Verified
0032d2e
Merge pull request #6 from sarahsid10/branch-3
Update Conflict.txt
This commit will be authored by 56936653+sarahsid10@users.noreply.github.com
Confirm merge
Cancel

VI. Cloning a Repository

This may be useful when you come across an external repository that can be cloned. Click on the “Code” box and choose the cloning method. Here we are using the HTTPS option. Click on the box next to the link to “Copy url to clipboard”.



In bash, type *git clone URL* (where URL is the text you copied) and press Enter.

```
ssiddiq8@RCLCARSS-7420 MINGW64 ~/Documents
$ git clone https://github.com/sarahsid10/LoveData.git
Cloning into 'LoveData'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 13 (delta 3), reused 4 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (13/13), done.
Resolving deltas: 100% (3/3), done.
```