

## Github Project Workflow

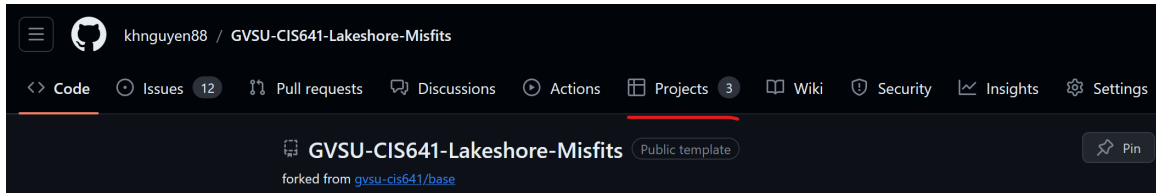
### 1. Go to the Project GitHub Repository

- a. <https://github.com/khnguyen88/GVSU-CIS641-Lakeshore-Misfits>

## Github Project

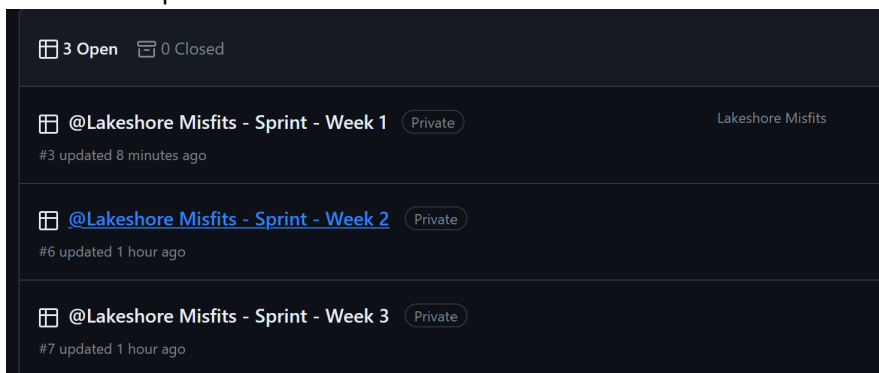
### 1. Click on the **Project** tab on the Project Repository's Navigation Bar

- a.



### 2. Click on the current sprint

- a.



### 3. You will see multiple views of the tasks we have on hand

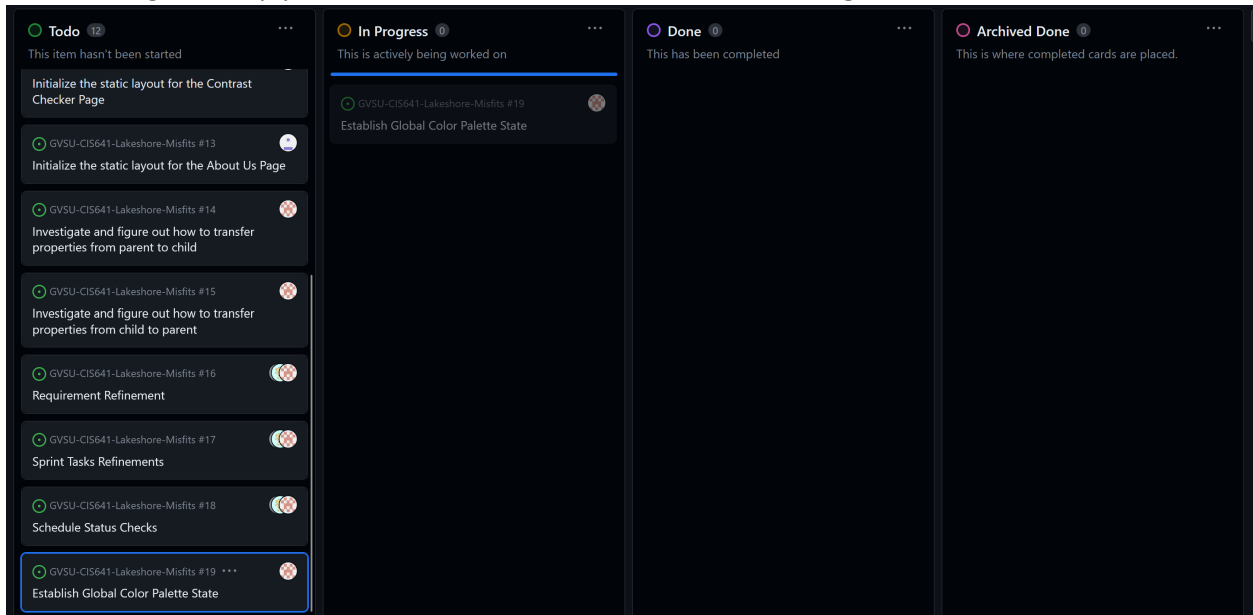
- a. View 1: List View
  - i. Decent too!
- b. View 2: Board View
  - i. Recommended
- c. View 3: Gantt Chart View
  - i. We are not using this!!

#### 4. Find the task assigned to you and change the status

##### a. **View 1:** Click on the Status dropdown and change it from **Todo** to **In Progress**



##### b. **View 2:** Drag and drop your task from the **Todo** section into the **In Progress** Section

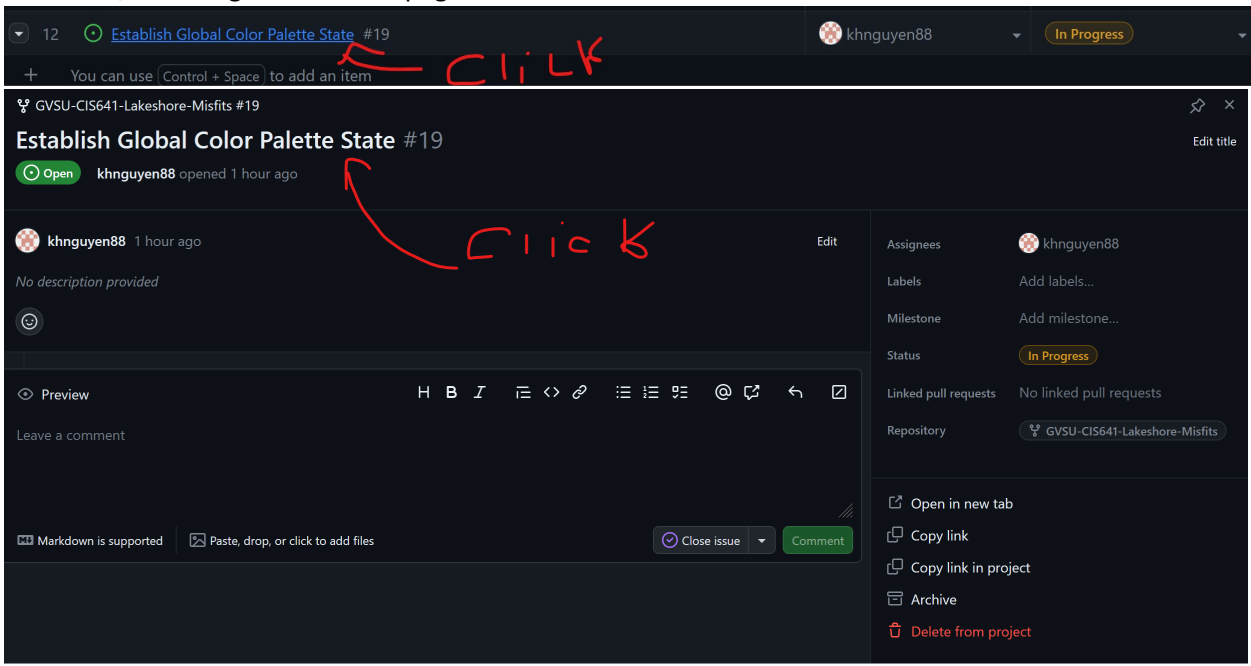


#### 5. Task Status Notes

- ToDo** – The task is assigned to an engineer/dev but has yet to be worked on
- In Progress** – The task is currently has been selected and worked on by a dev
- Done** – The task has been completed and currently waiting to be reviewed, and the work branch is waiting to be merged into the main branch
- Archived Done** – The task has reviewed and merged into the main branch

6. Select on the task/issue to go to it's issue page:

a.

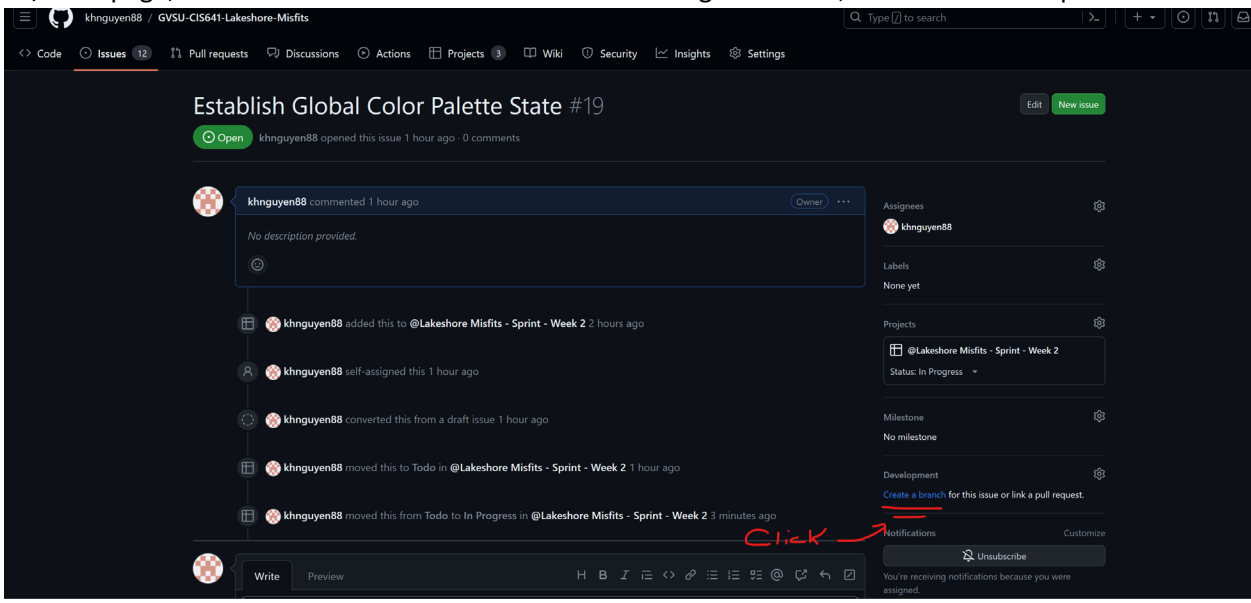


b.

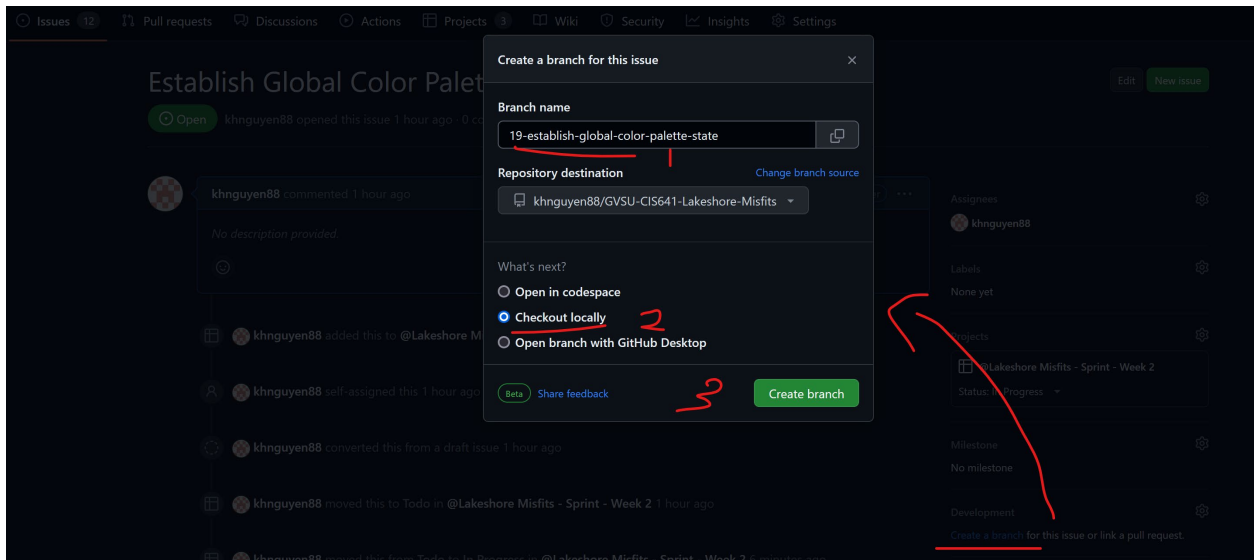
## GitHub Issue

1. After we click on the issue assign to use and are in the issue page
  - a. Alternatively we can go into our Project Issue tab and select the Issue/Task we want to comment on and close
2. We want to create a work branch that is **separate from our main branch**
3. In the task/issue page, click on the **“Create a branch”** link on the right side bar, under the Development header:

a.

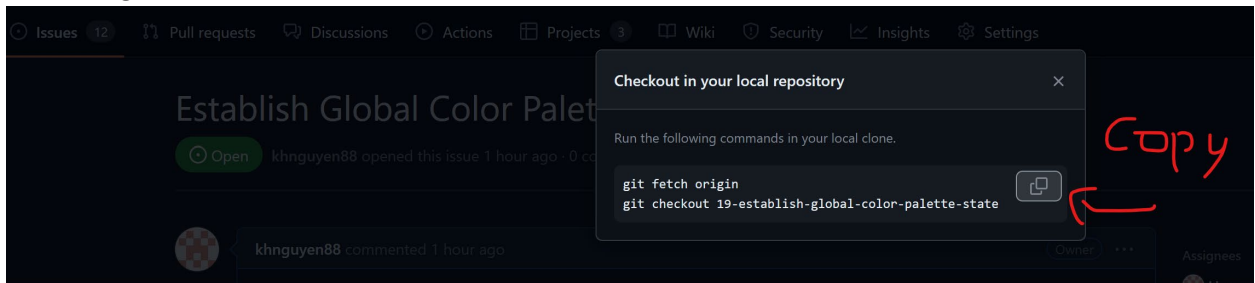


4. Create a new branch for the particular issue and set to **checkout locally!!**



a.

5. Copy the following command to our local clone in the VS Code Terminal



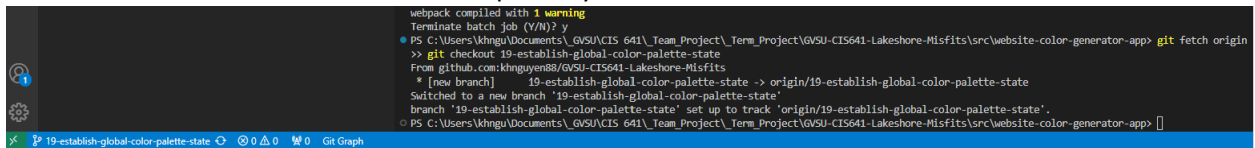
a.

6. Paste the commands into the VS Code terminal while we are in Project Folder in VS Code:



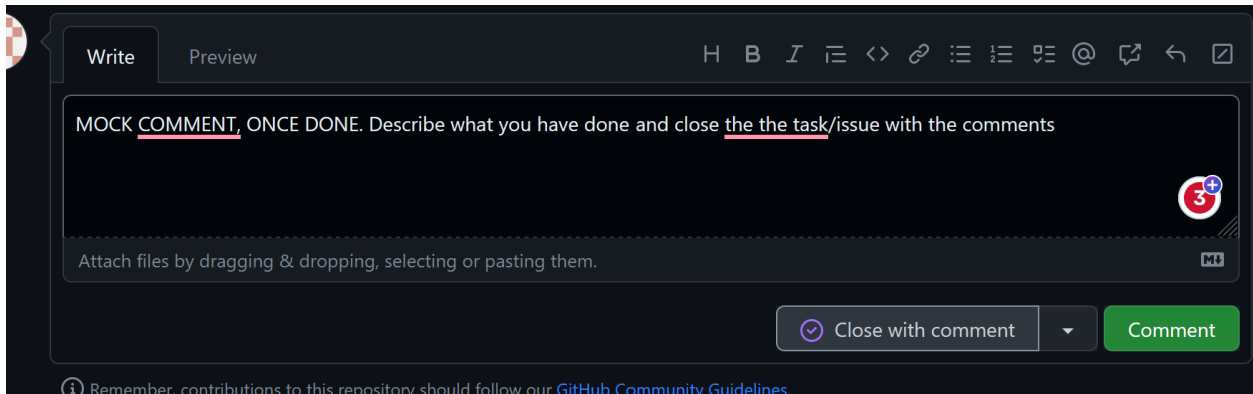
a.

7. We now fetched the new branch into our local repository and checked it out



a.

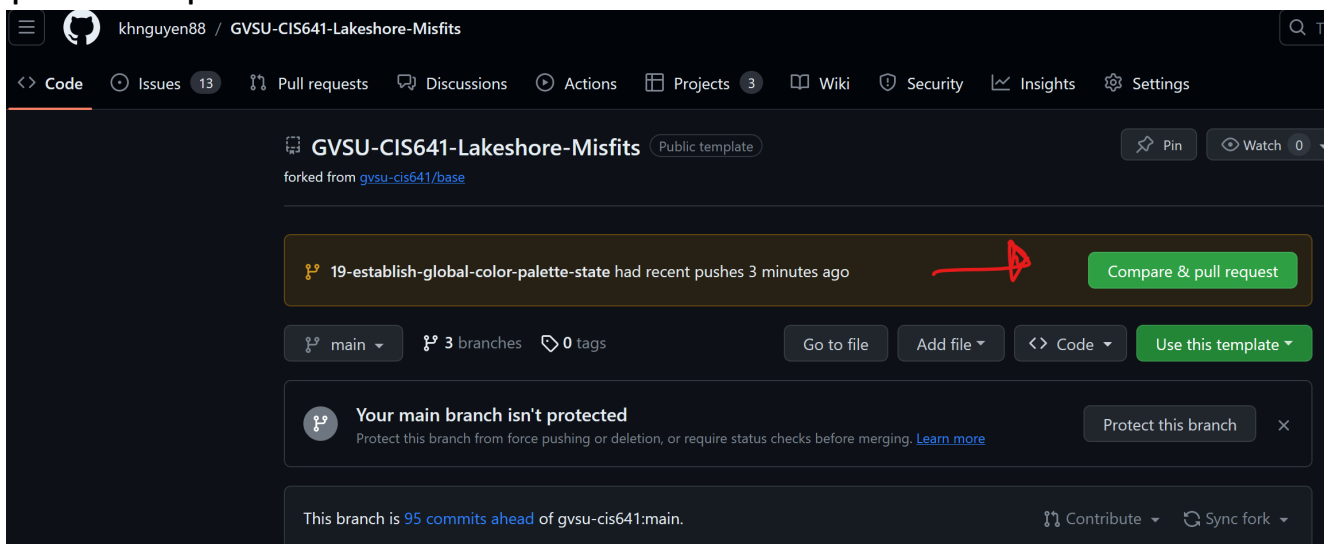
8. Make the necessary changes desired and then once we are done. Press the “**close with comments**” button to close the issue with comments



- a.
  - i. Technically a reviewer should be closing the issue, but if you're confident you can do it too

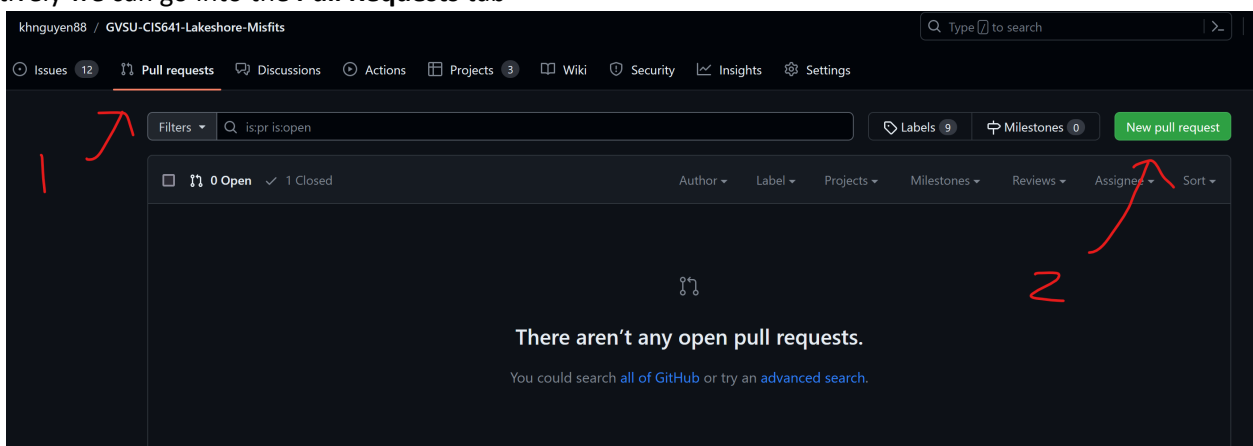
## GitHub Pull Request

1. After we pushed our working branch up into the repository. We need to create a pull request.
2. If we just pushed up our working branch and go to our project GitHub repository, we can see a notification. Click on the **Compare & Pull request button**



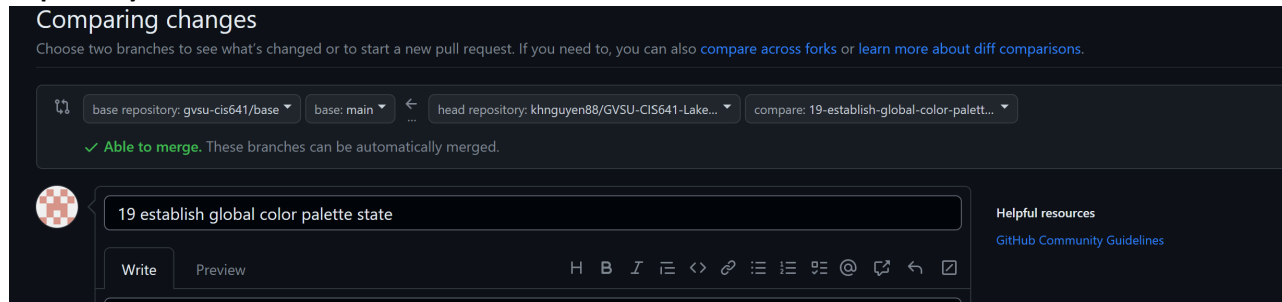
a.

3. Alternatively we can go into the **Pull Requests** tab

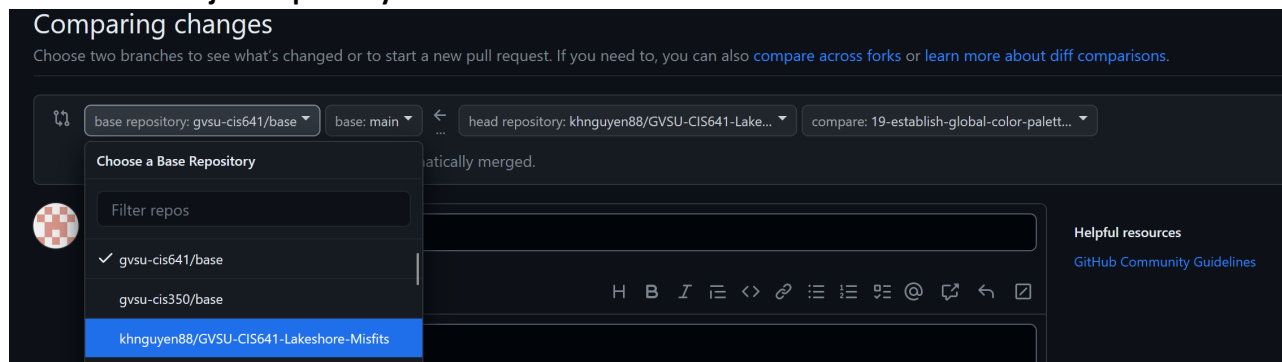


a.

4. After we click on the pull request button, we need to determine which branches we want to compare and what we want to merge into.
  - a. It is always “Parent (Main) Branch” (Left Side) ← “Child (Working) Branch (Right Side)”
    - i. We are comparing the Child branch to the Main Branch
    - ii. We want to merge the Child branch into the Main Branch
  - b. **NOTE, VERY IMPORTANT!! MAKE SURE WE SET THE BASE REPOSITORY FOR BOTH BRANCH CORRECTLY!! SET TO TEAM PROJECT REPOSITORY AND NOT THE BASE REPOSITORY WE FORKED FROM.**
    - i. Default will have us compare our **Project Repository and Working Branch** with **GVSU-CIS/641 Repository and Main branch. DO NOT CREATE A BRANCH FROM THIS!!!**

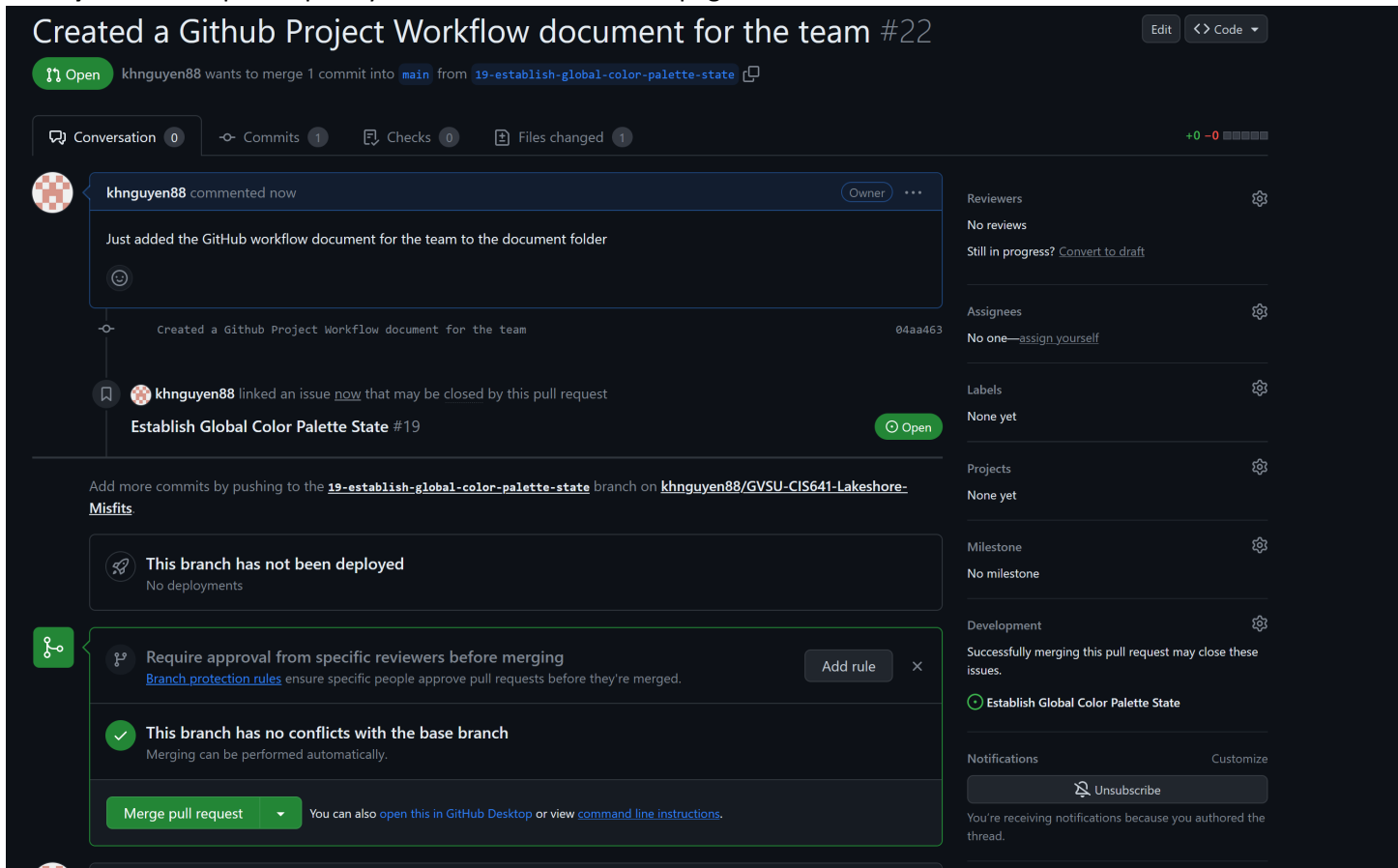


- ii. We need to change the parent repository branch from **GVSU-CIS/641 Repository and Main branch** to **Our Team Project Repository and Main Branch**



5. Now Press the **create pull request** button!!

6. Once you create a pull request you will be directed to this page

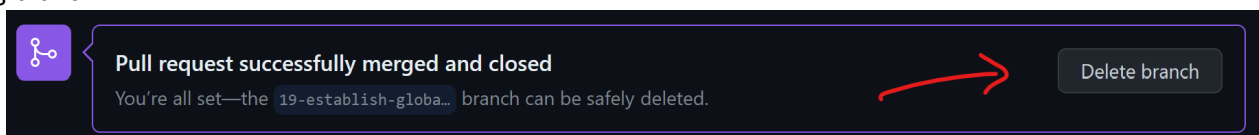


7. Under the working branch pull request page, click on the **Commit** and **File Changed** to review the changes that have been made in the working branch.
8. Once the reviewer is satisfied with the changes, they can merge the pull request. Click the **Merge pull request** button.



- a.
- i. This will merge the working branch into the main branch
  - ii. Add comment and confirm merge

9. Once the merge is successful, the pull request will close. If you no longer need to work on the issue, delete the working branch.



a.

10. Afterward, tell the entire team to pull down the latest main branch so they can merge it into their working branch to receive the latest changes.
  - a. Use the terminal command line: **git merge <main branch> where <working branch>**

#### After Pull Request

1. After you have resolved your issue or completed your task, go into the Issue Tab, select your particular task/issue, add comments on the changes you made, and close it.
  - a. Technically an independent reviewer should do this.
2. Go into the GitHub Project and Sprint Page and move the issue from In **Progress** or **Done** to **Archived Done**