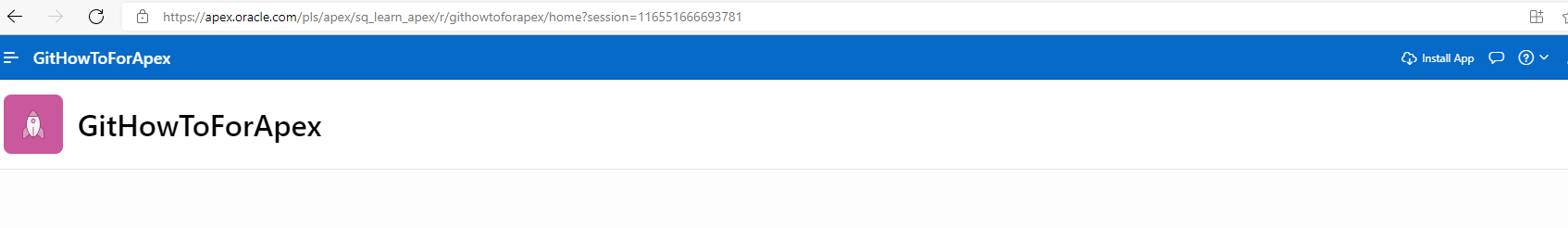
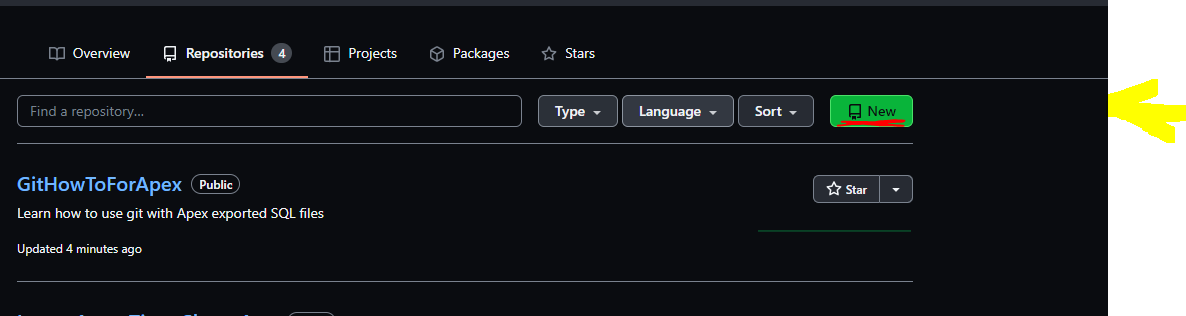
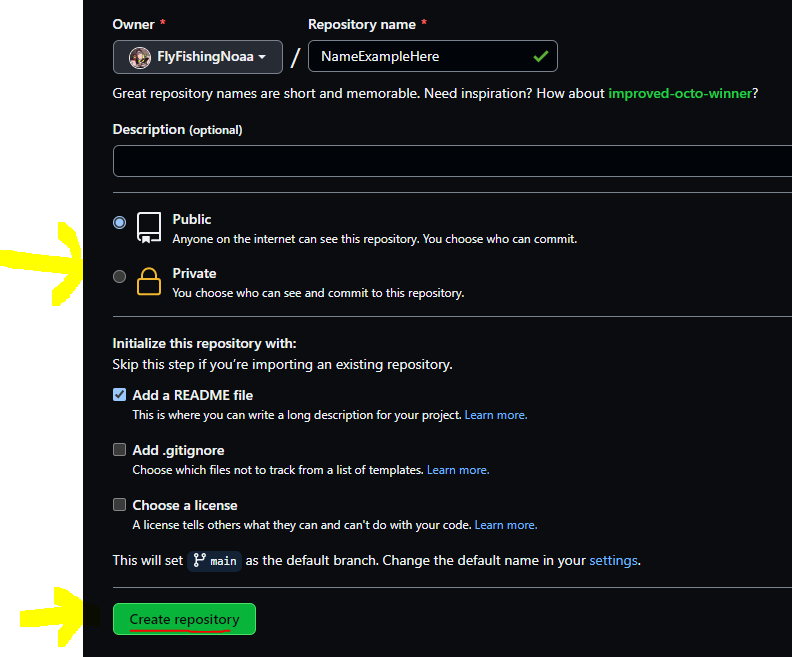
GitHub How to Using Apex/SQL

Disclaimer: This is an example of how Git might be used to work with Oracle Apex/SQL files and this is not an official NOAA process and more of a high level overview of how to branch and merge your code.  
  
Note: There are 3rd party apps and plug-ins that do a lot of the work for you – this example is working with sql files manually.

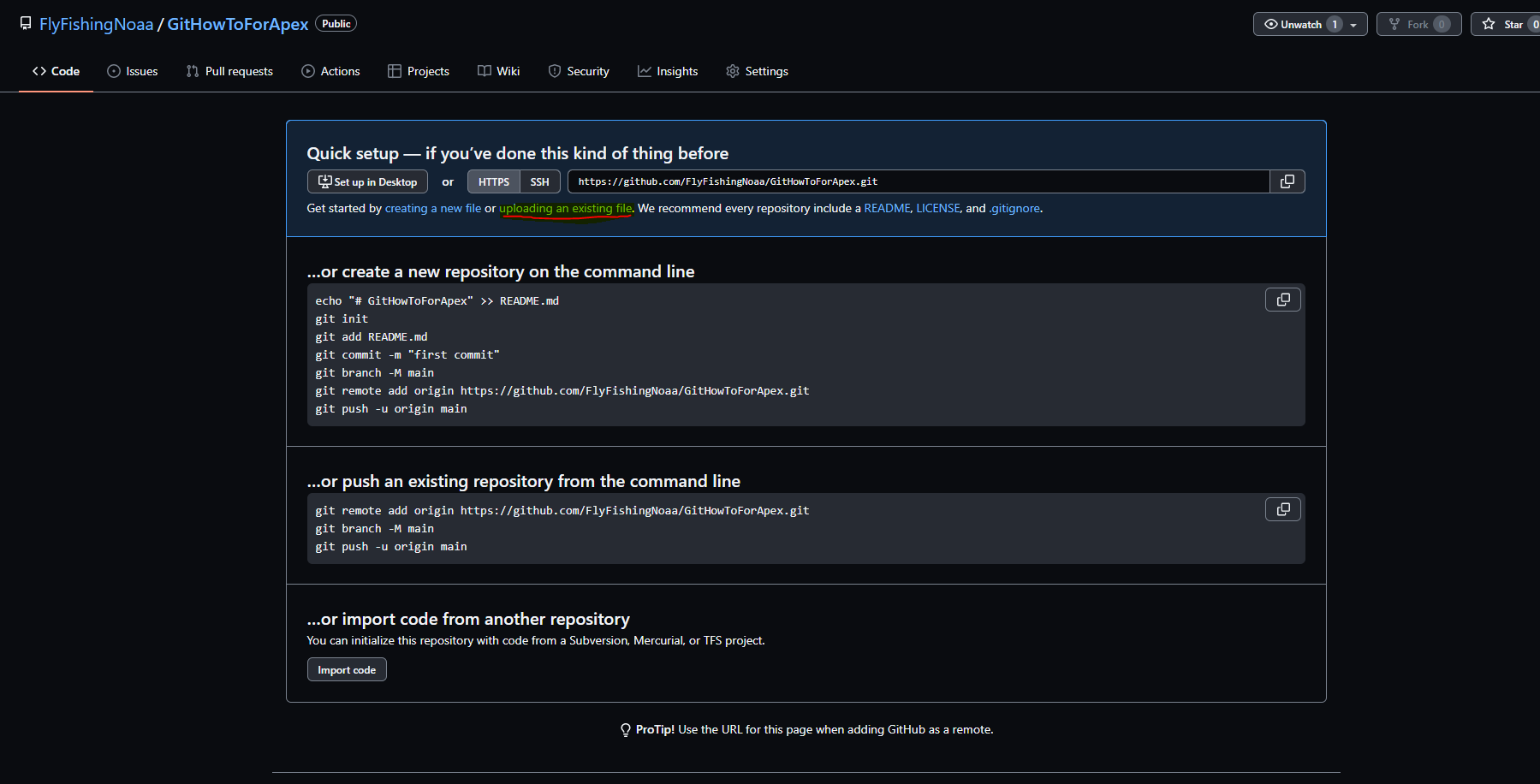
Have an oracle apex application ready (or a sql file from SQL Dev for example)– here is a very simple shell of a project that has been set up. Export the application as a SQL file.  


To create a new Repository, in GitHub click on the Repositories tab and then select New:  


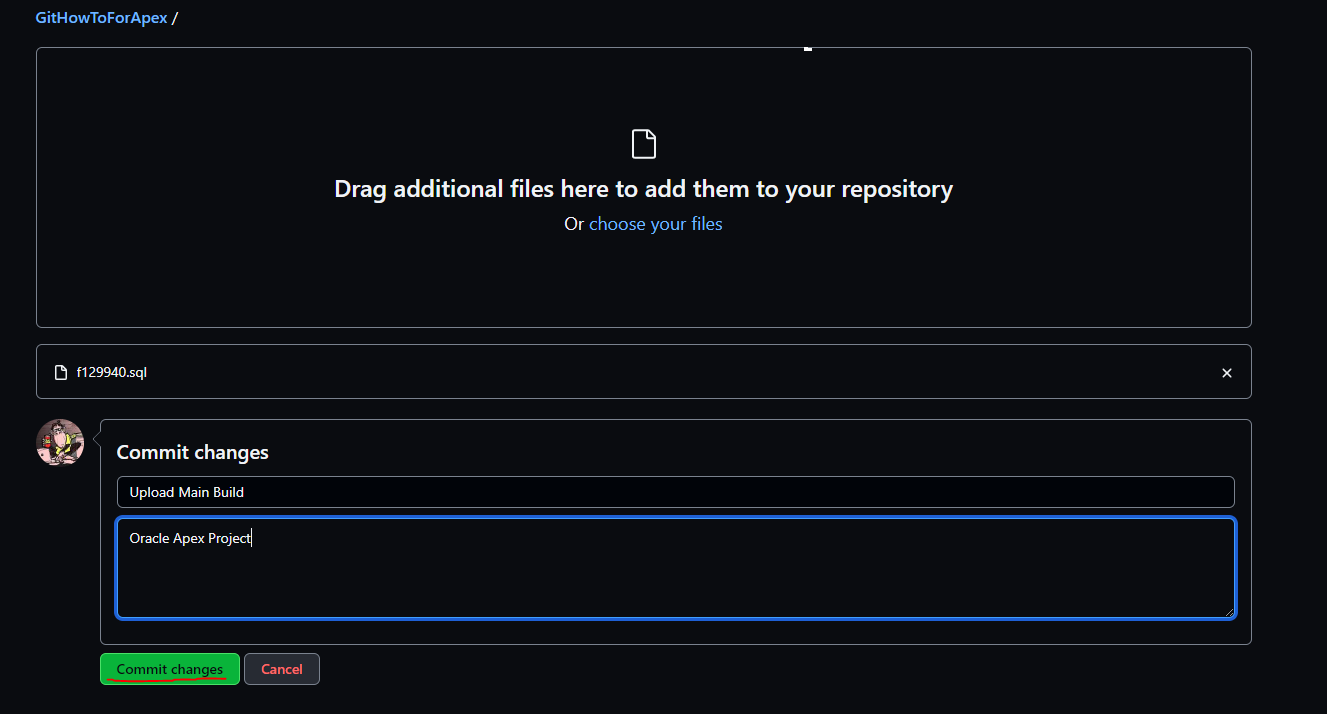
Give your Repository a name in the name field, select the options at the bottom, take extra notice if you want your repository to be Public or Private, if Public anyone on the web view it. Private will be just for you or users you invite to view it.



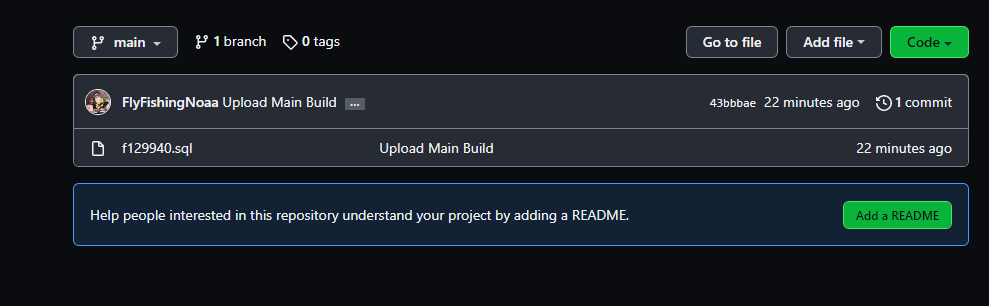
After creating the new repository – click on “uploading an existing file.”



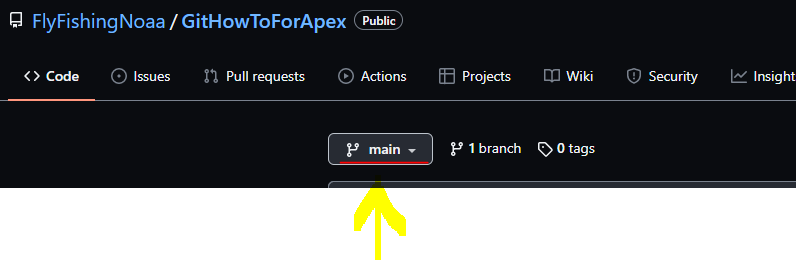
Click on Commit changes after upload:

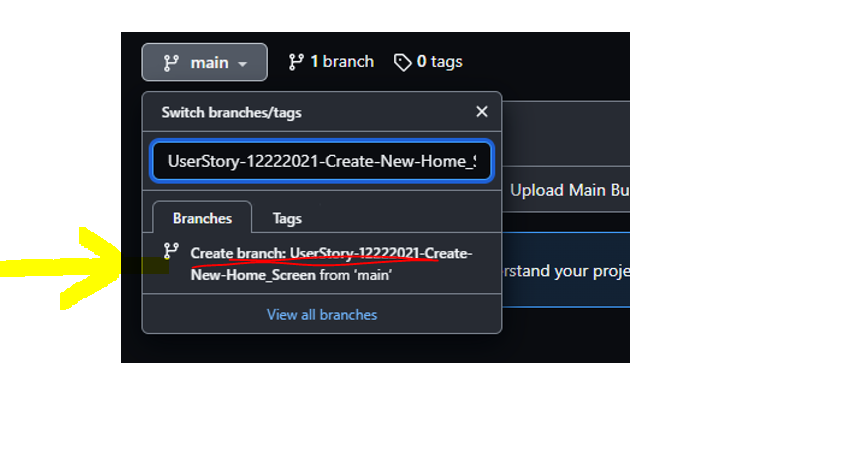


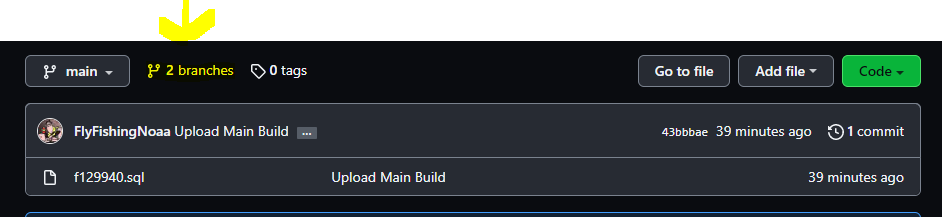
You now have your main branch set up with the initial project file.

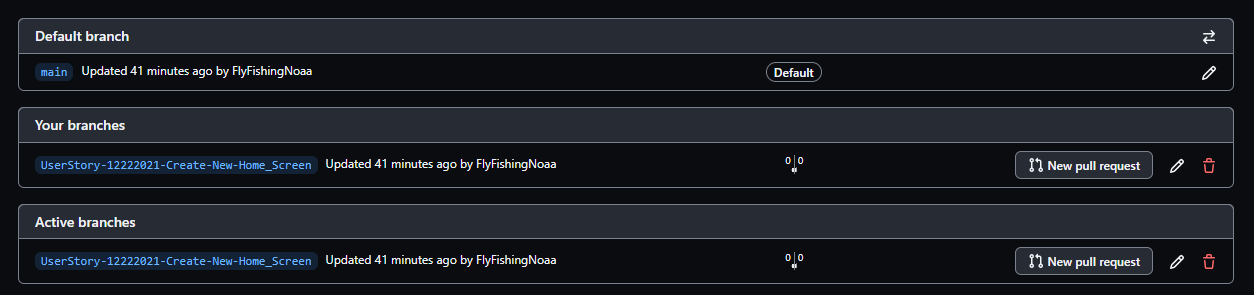


**Example of a change to your project - a user requests updates to the home screen.**

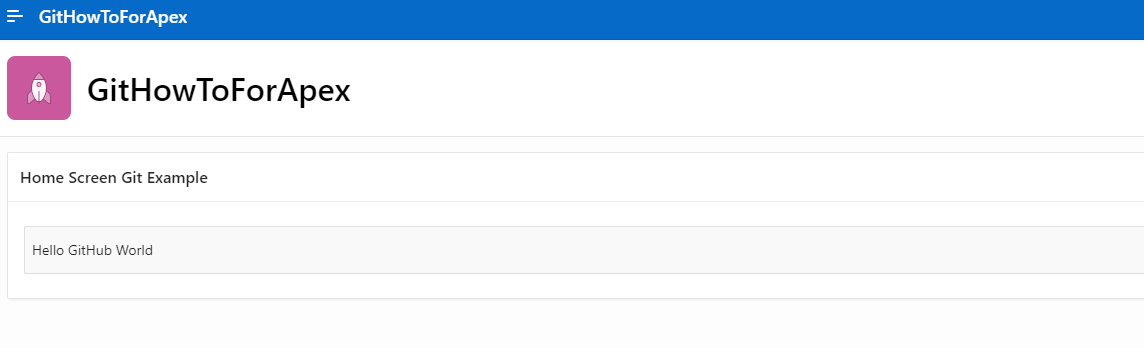
In GitHub – click on the main branch button.  


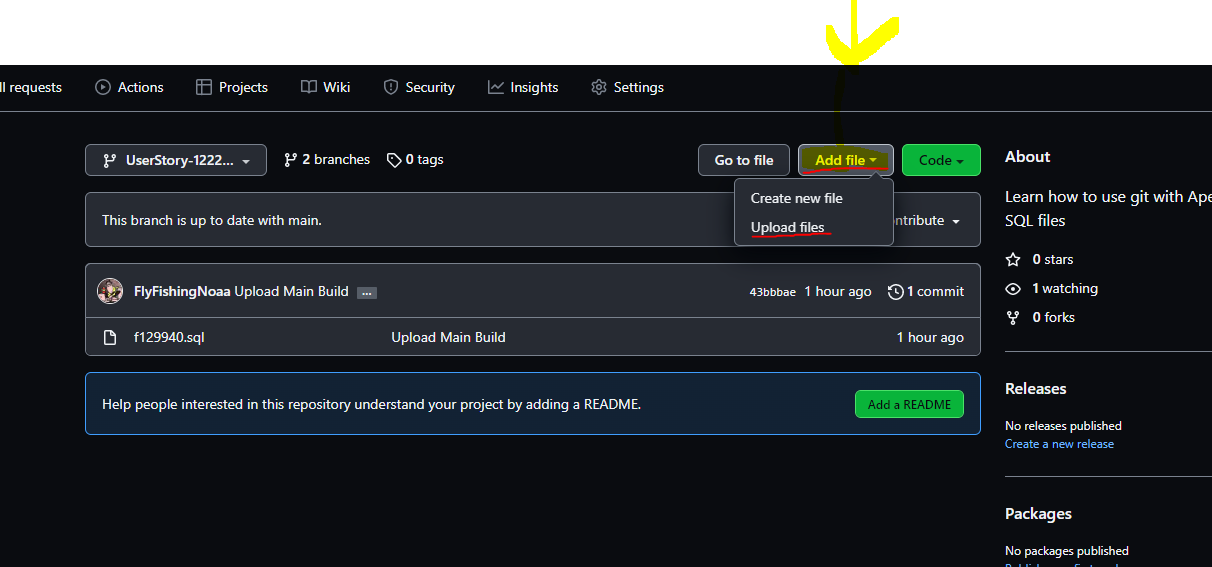
Add name for new branch and then click on Create Branch for the new request:  


Notice now it says 2 branches:  


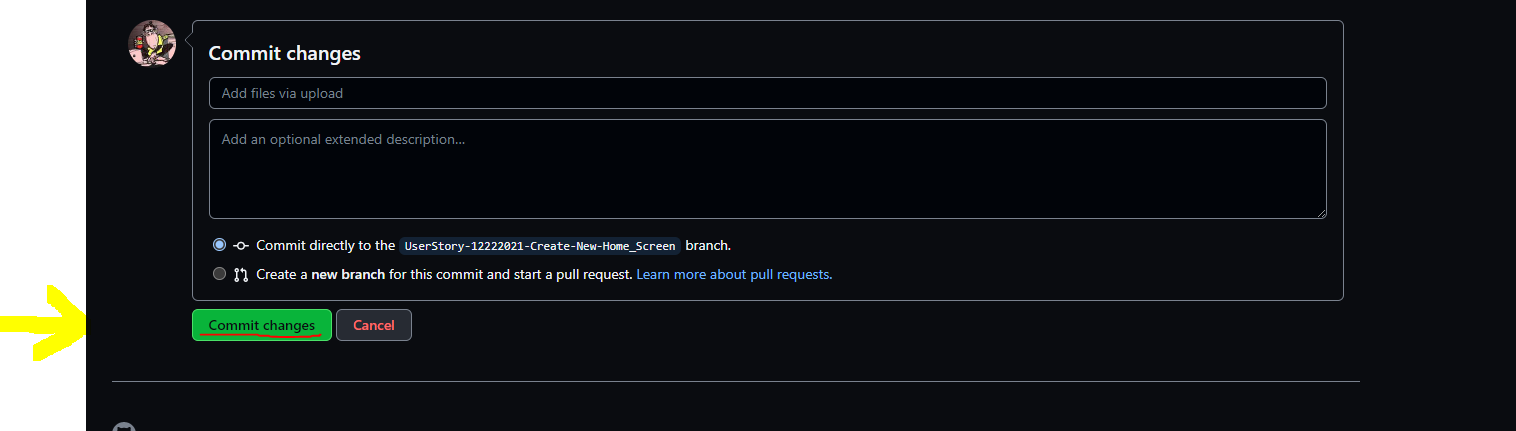
Click on the 2 branches item and you will now see the newly created branch:  


Go back to your apex project and make some updates for your new branch then export your project, here we added a new region with the title Home Screen Git Example. Export your application as SQL.

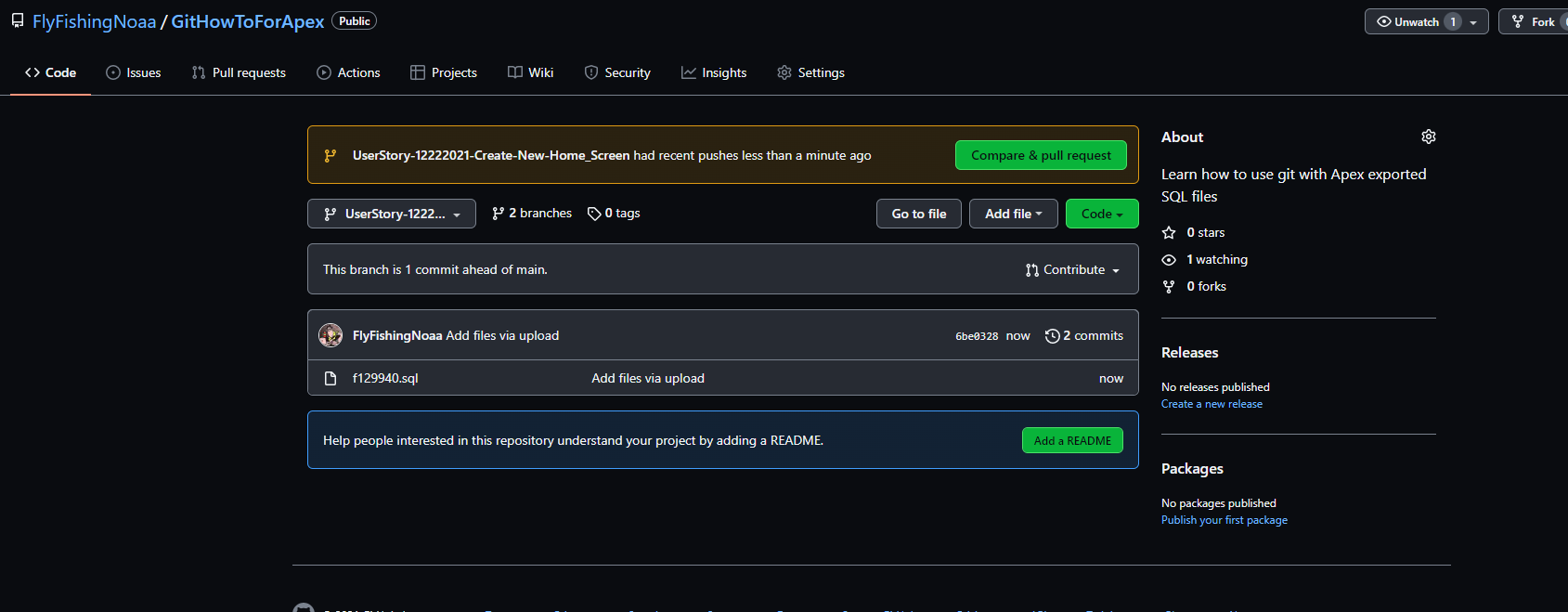


Go back to github – click on branches and open up the new branch that was just created. Then click on add file and then Upload files to upload your exported file.

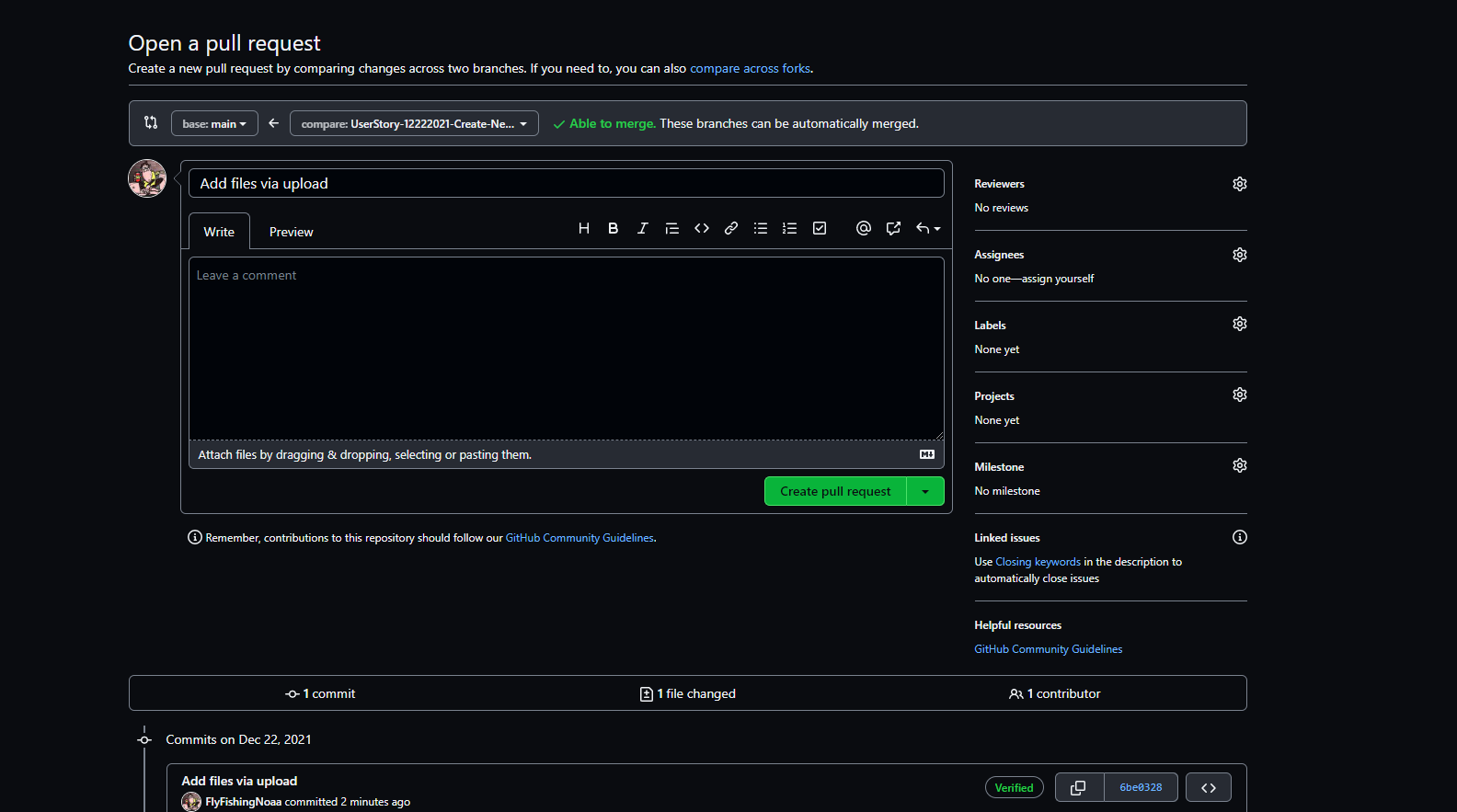
Now click on Commit changes to “check in” your code.



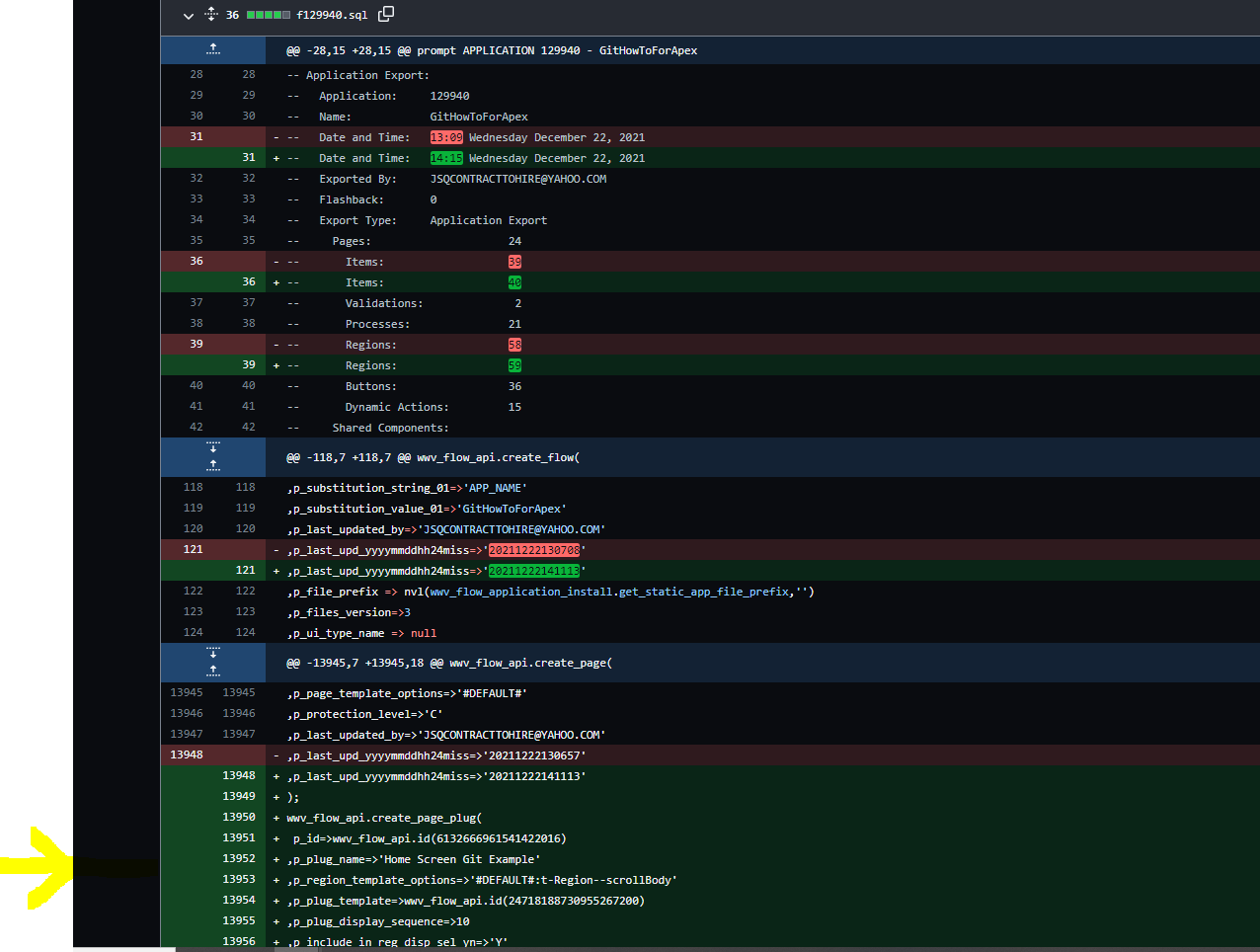
Now click on Compare and Pull request.

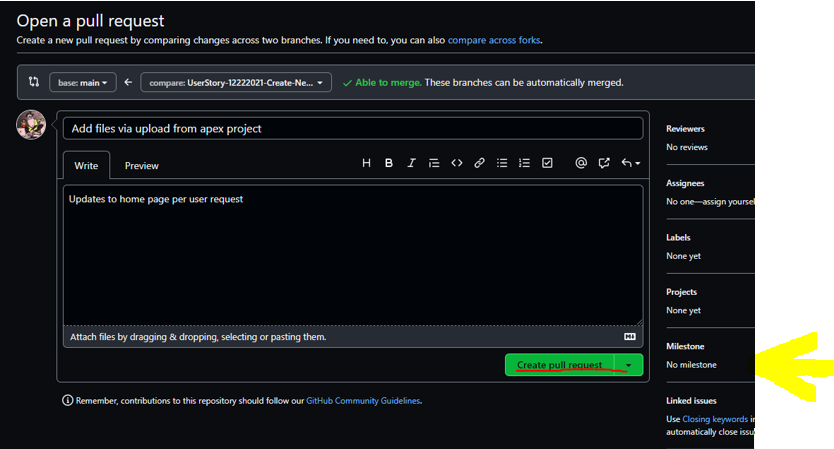


You will now be on the Create pull request page – scroll the page down so you can see the changes being made in the comparison window.

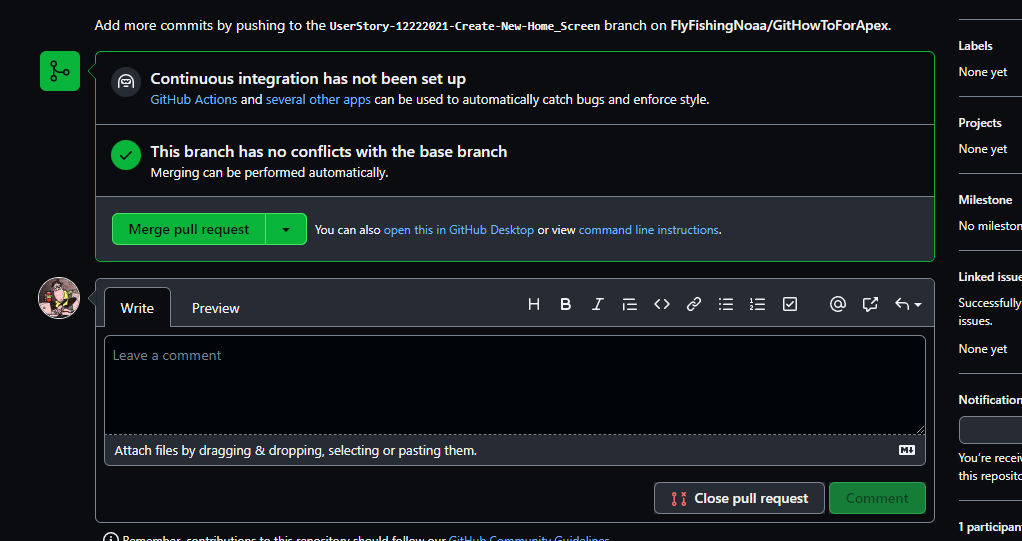


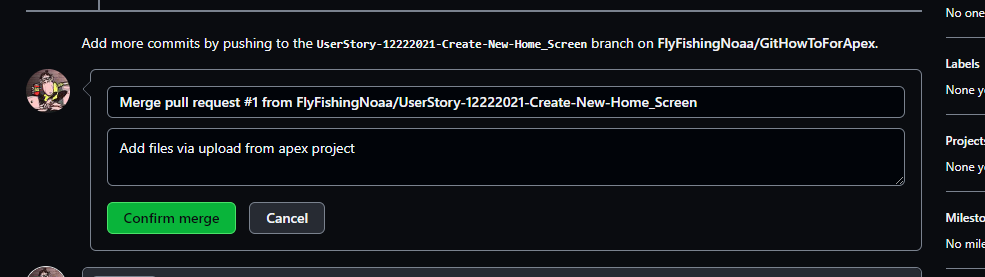
In the compare code window you will see the changes being made from your branch to the main branch.  
Green with a + shows code that is being added, Red with a – is code that is being removed.  
Notice below the arrow showing the changes made with the new title “Home Screen Git Example”



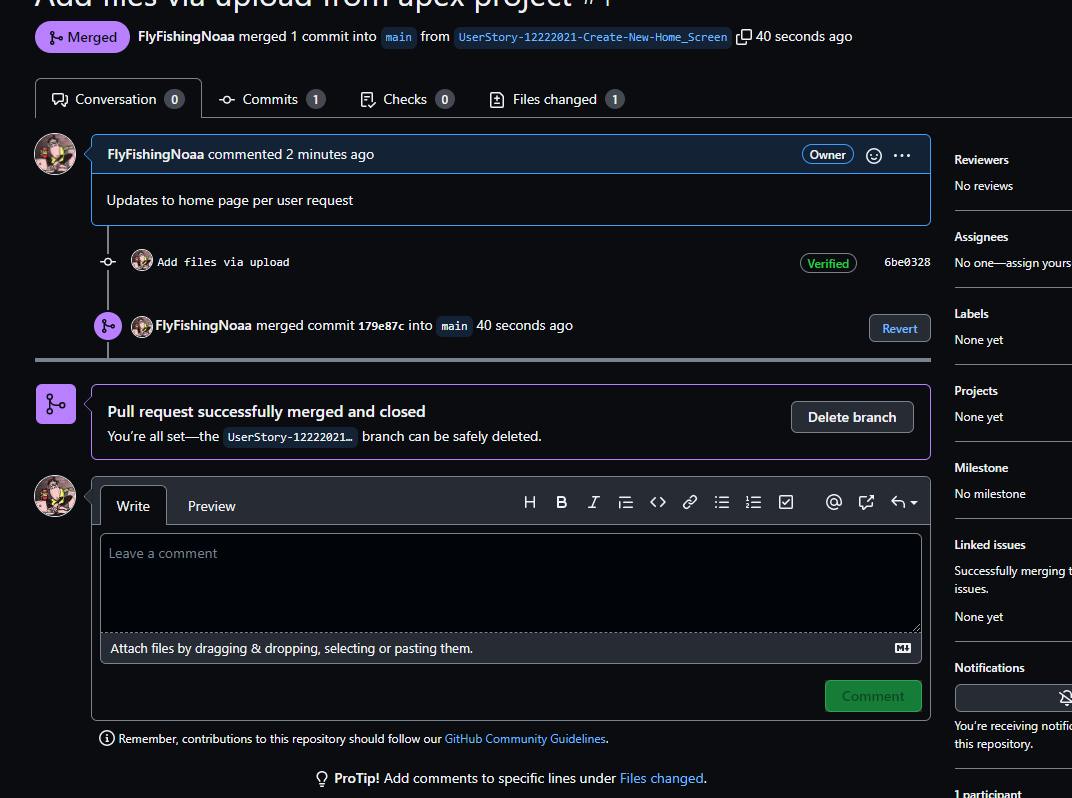
Going back up to the top of the page you can now update the notes and Click on create pull request.  


Now Merge the Pull Request:

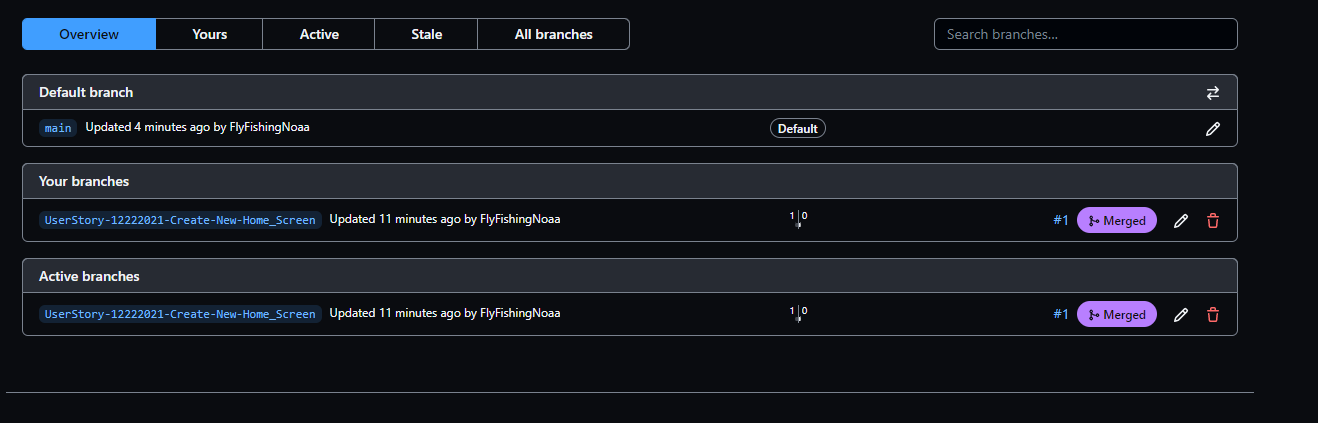


Then Confirm Merge

Indication the merge was completed, you may now delete the branch but you may also keep it for an audit trail of work done.

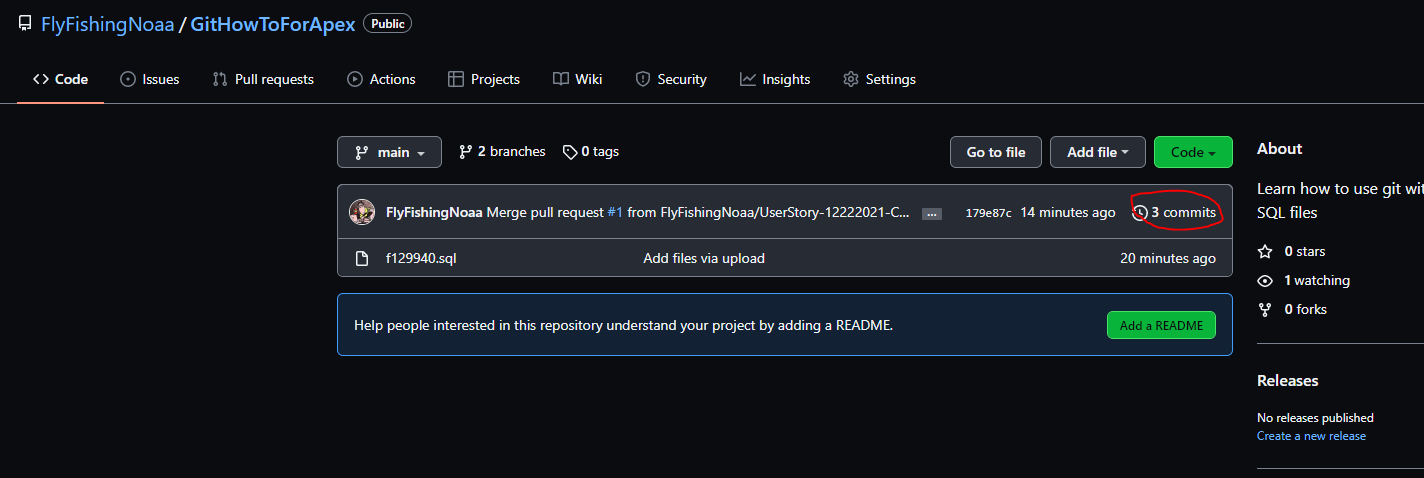


Go back to your github home screen and click on the branches tab and you will see branches with the merged status.

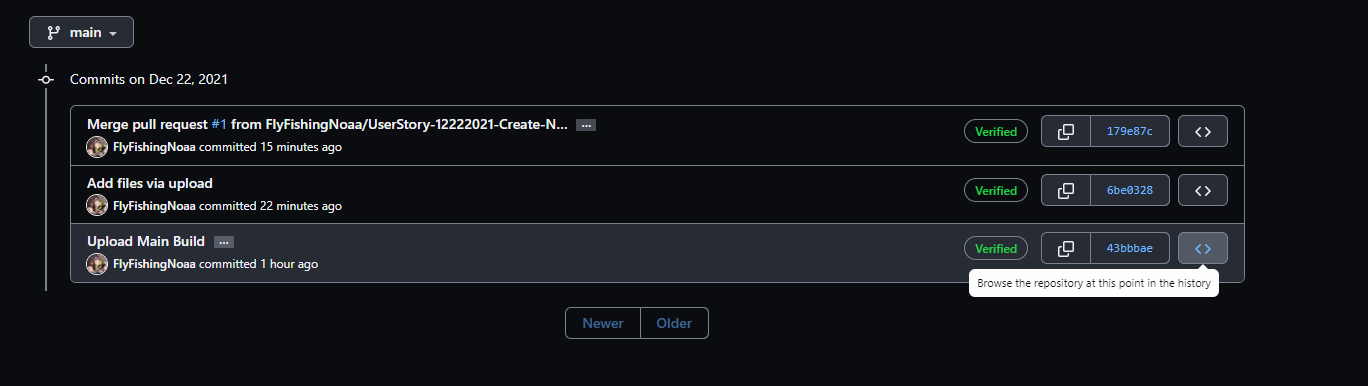


If you need to revert back to older code:

Go back to the main screen and you will see a clock icon with X number of commits.



Click on the <> button for the build you wish to roll back to – in this case we are pointed back to the first main/base line build.



And now you can review the file:

