Lab 4 – Feature branching with Visual Studio Code

Objectives:

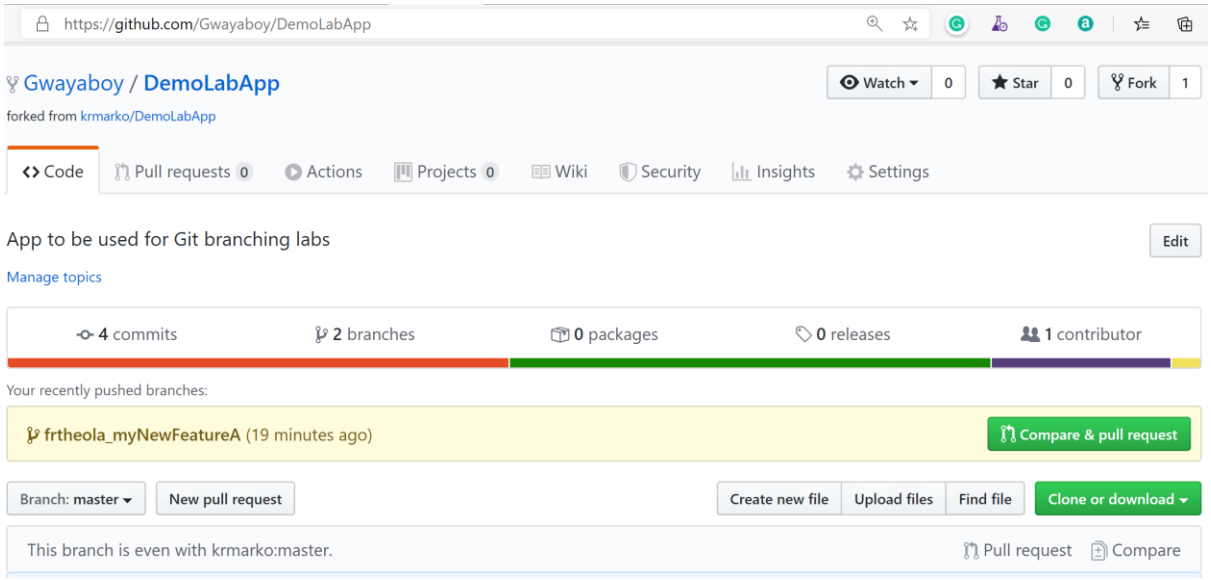
After completing this lab, you will be able to work with a remote repository using Visual Studio Code:

- Clone a remote repository
- Create local feature branches to makes changes and push them up to the remote repository

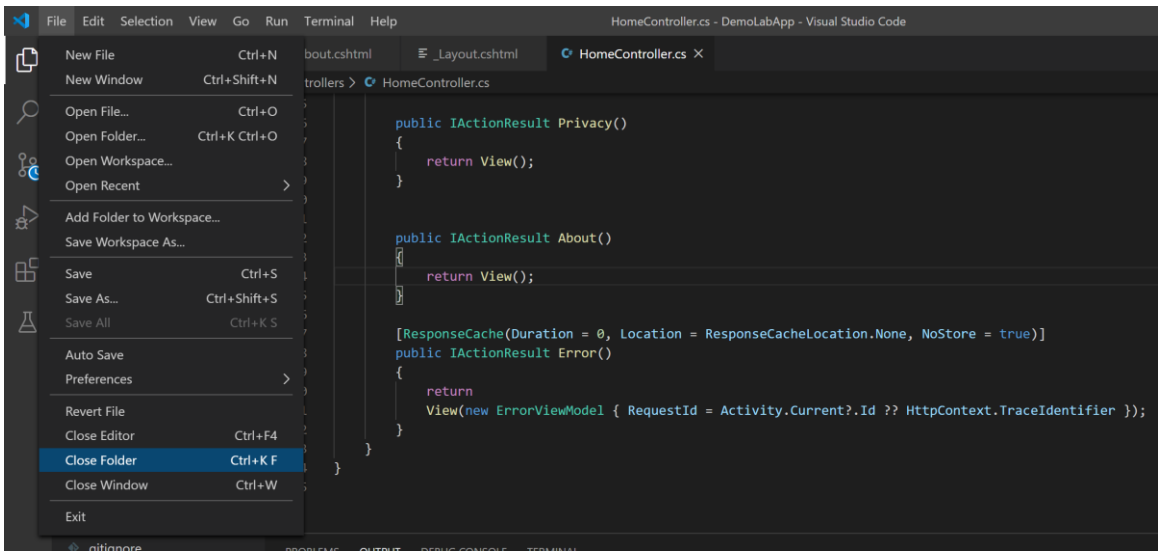
Clone Remote Git Repository

We've already forked to our personal GitHub Repo and should now have 2 branches (master and our feature branch). You can verify this by navigating to *https://github.com/<yourGitHubUserName>/DemoLabApp.*

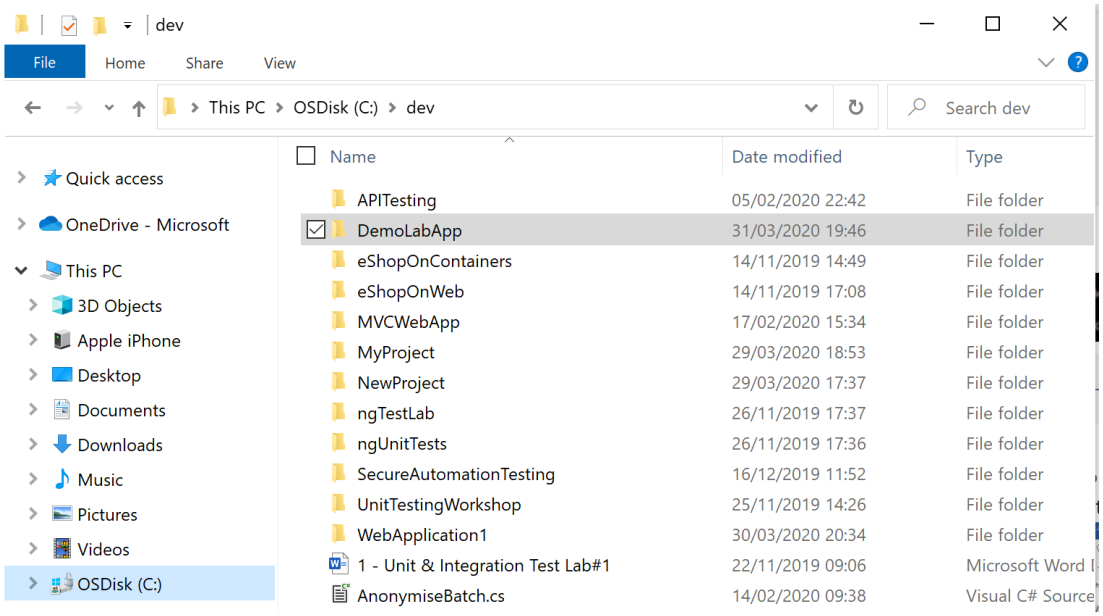
In my case that would be https://github.com/gwayaboy/DemoLabApp



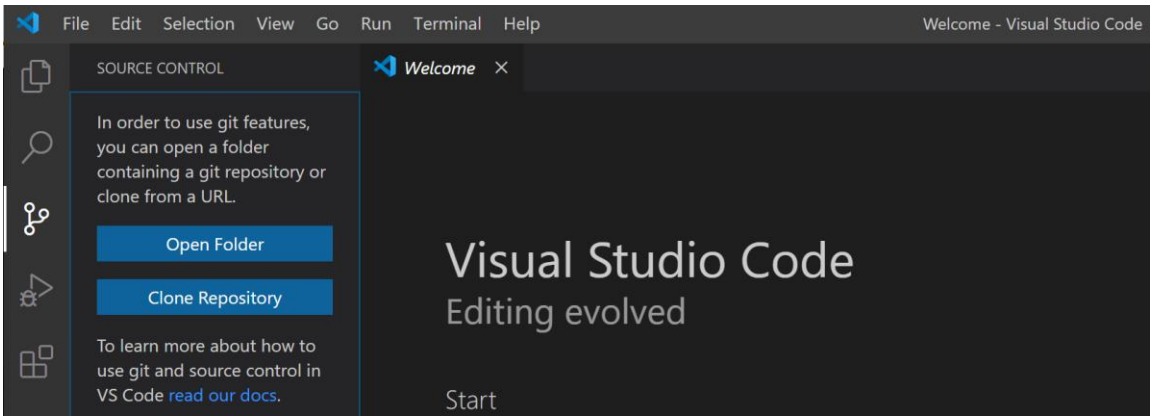
1) Open a Visual Studio Code and close any currently open project or folders.



2) Open the File explorer and delete the previous repository

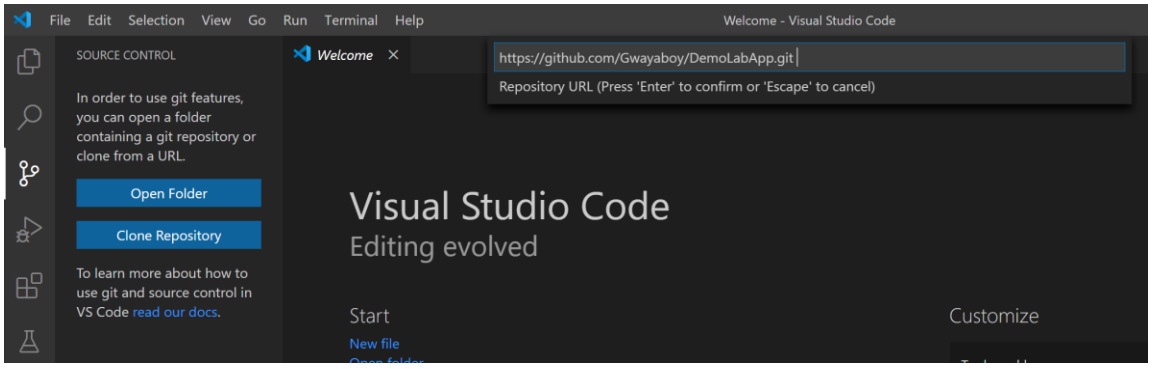


3) Then back to Visual Studio Code, on the left side bar, click on the source control button (3rd button from the top) to bring up the source control manager panel as follow:

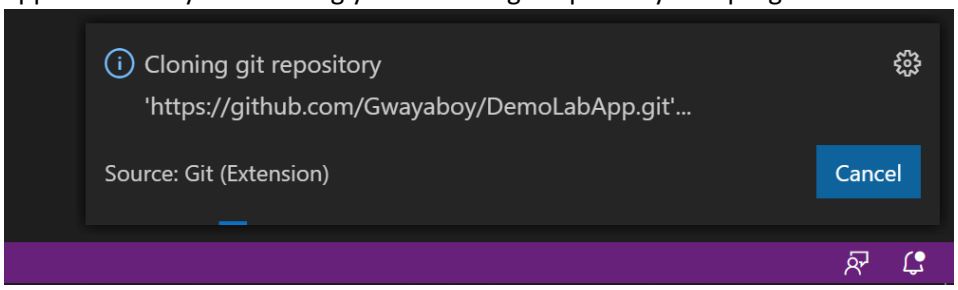


4) Click on the Clone Repository button and enter GitHub remote repository's URL *https://github.com/<yourGitHubUserName>/DemoLabApp.git*

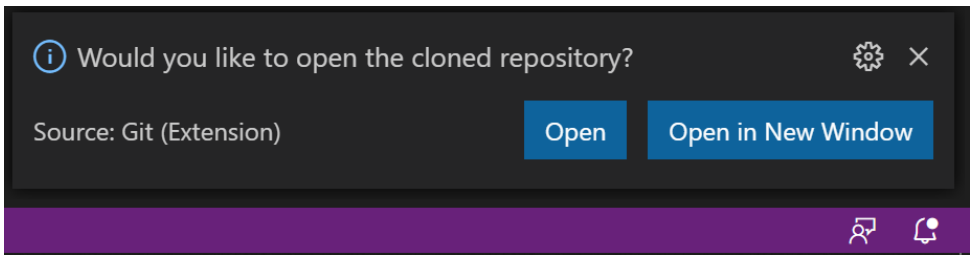
Please replace *<yourGitHubUserName>* with your GitHub username which you can find in the URL. In my case that would be https://github.com/Gwayaboy/DemoLabApp.git



5) Confirm the root folder under which you want to clone the repository. For example, I am setting my current directory to my local c:\dev\ folder.
Once you've selected the desired root folder, on the bottom right corner a pop up will appear to notify that cloning your remote git repository is in progress as below:



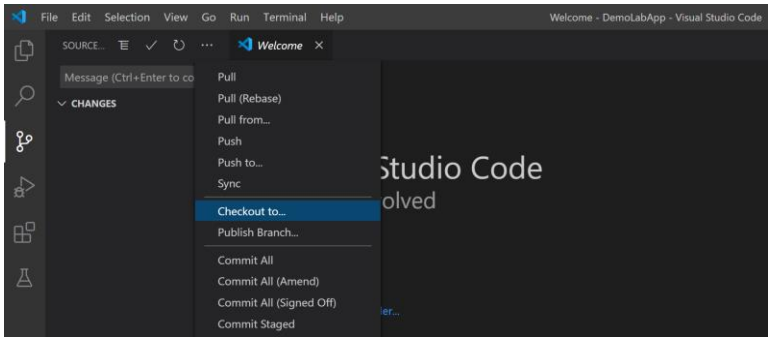
Once the repository was successfully cloned, another pop-up should appear and suggest opening the clone repository in this VS Code window or a new one. Please choose open



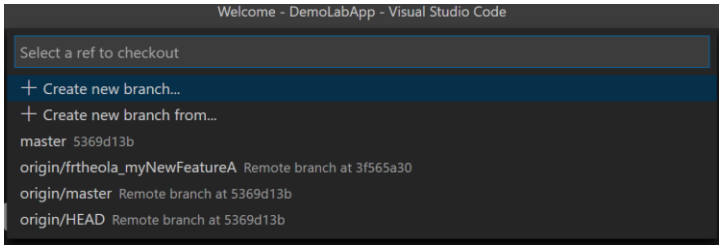
Create a new feature branch

We are going to create another feature branch so that we can make our changes while protecting the integrity of our main or master branch.

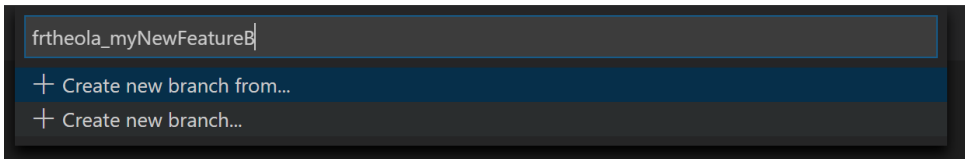
1) Click on the source control button on left side and select the 3 dotted contextual menu to choose the "Checkout to…" item



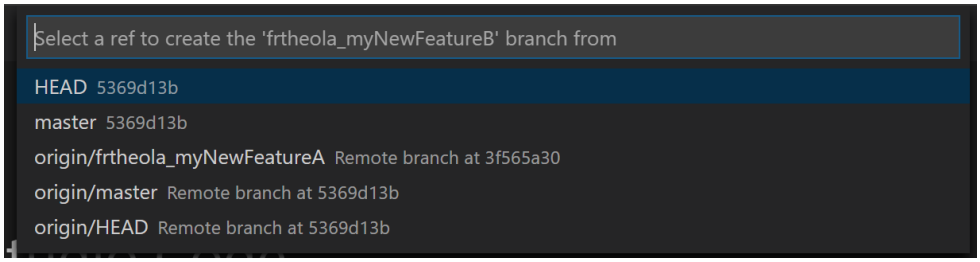
Please notice that you can see all remote (master and feature branch) and local branches (just master)



2) Create a feature branch by selecting “Create new branch from…”
You can name your new feature branch using the first part of your email address as follow:
git checkout *<emailHost>_myNewFeatureB*
For example, mine would be frtheola_myNewFeatureB

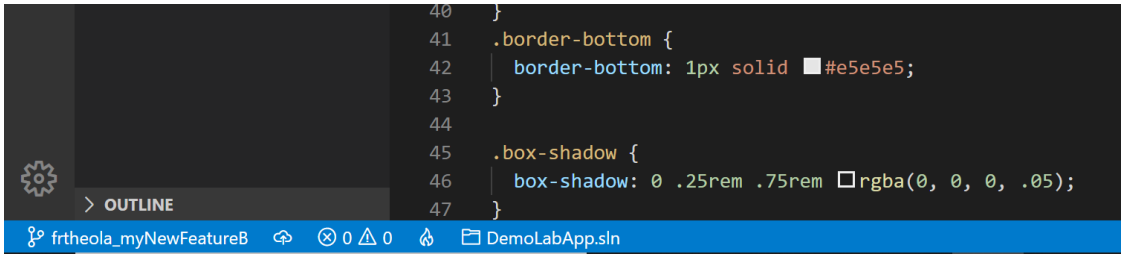


3) Then select the local master branch



Please notice that HEAD, master both local and remote (origin/master & origin/HEAD) all have the same truncated identifier which basically means they are all pointing the snapshot.

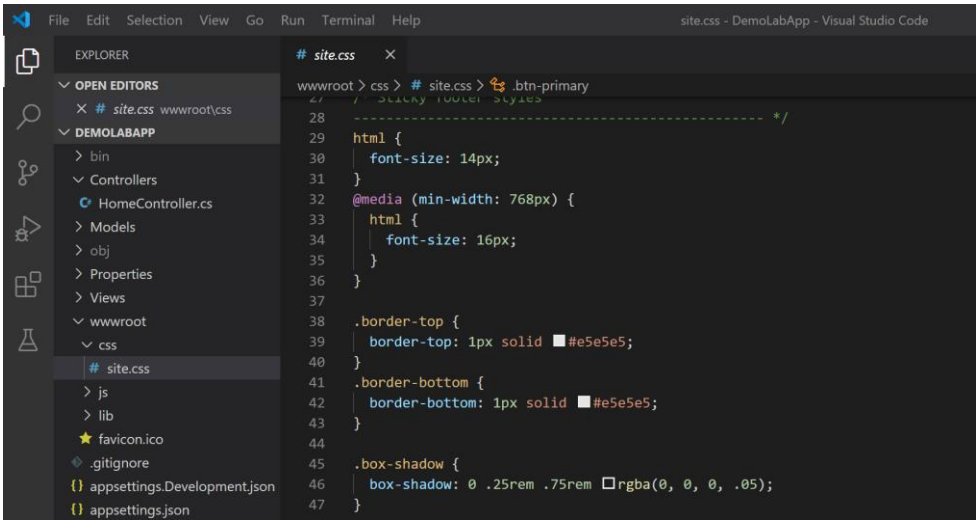
To verify that which branch you are currently working look on the bottom bar left corner to spot that you have indeed switched to *<emailHost>_myNewFeatureB* (In my case frtheola_myNewFeatureB)



Make changes

We are going to amend the style.css file in our new feature branch

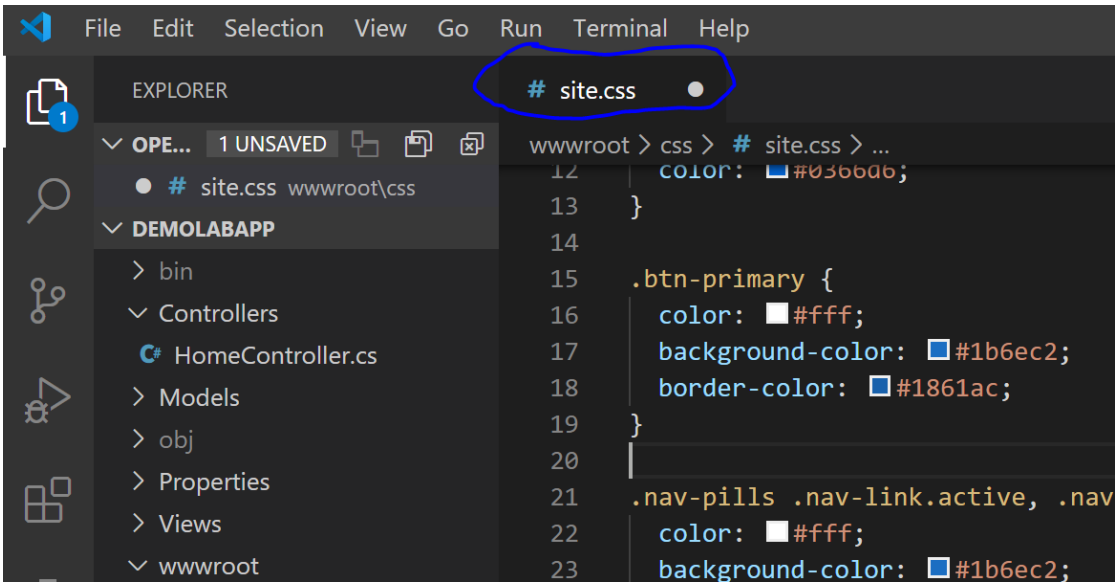
1) The left side explorer, navigate and select site.css under DemoLabApp/wwwroot/css/



2) Then after line 34 in the media section which applies for smaller devices at width less than 800px we are going to change the background colour to grey for these targeted devices (see bold css instructions to add below)

```
html {
  font-size: 14px;
}
@media (min-width: 768px) {
  html {
    font-size: 16px;
    background-color: grey;
  }
}
```

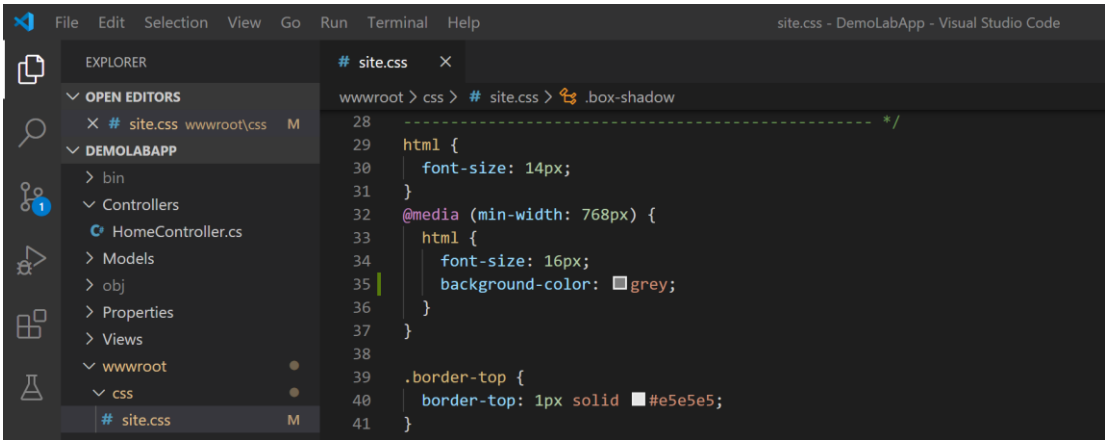
Please note when the file is being modified but still unsaved the tabs shows a dot after the opened file name.



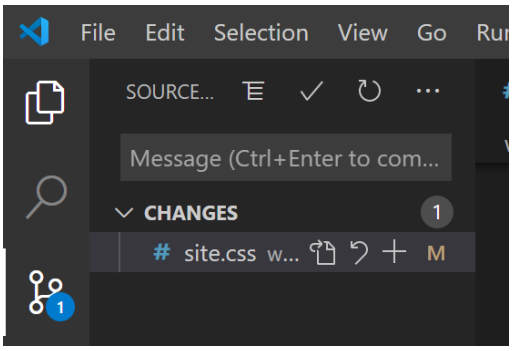
3) Save the file using the File > Save menu item or strike the Ctrl + S key combination.

After saving the file please notice again how Visual Studio Code is keeping track of the changes by highlighting orange the modified file site.css and marking with **M.**

Also take note that the Source Control Management (SCM) icon now mentions one change.

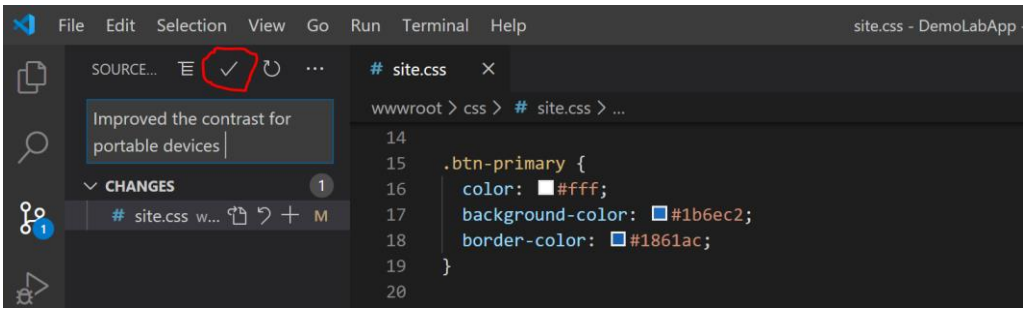


4) Now let's get ready to stage and commit the changes to our local feature branch by selecting the SCM icon on the left side



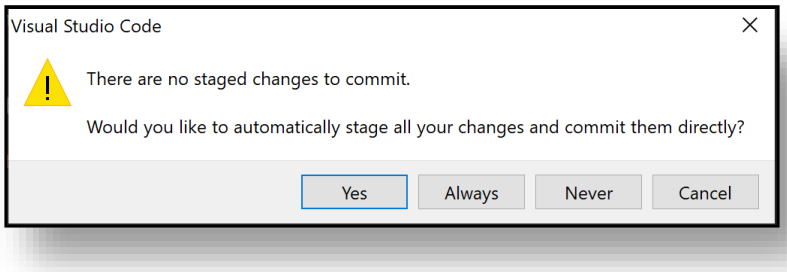
We have now our modified/untracked file listed in the "CHANGES" section. Please notice the 3 icons before the **M.** From left to right, the first icons allow to open the file, the second and middle to undo any changes and last one to stage the changes.

5) We are going to stage and commit in one operation in Visual Studio Code.
Enter a meaningful comment such as "Improved the contrast for portable devices "and click on the top tick icon:



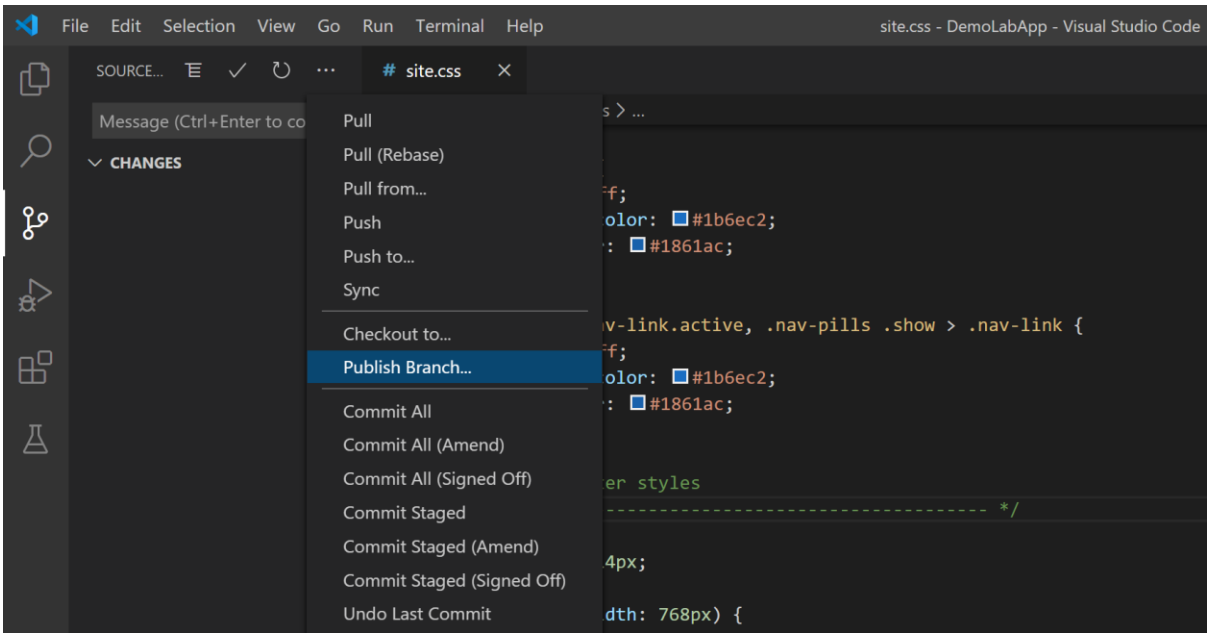
Please notice that you must provide with a comment when committing any staged changes. Also, it is recommended to have meaningful and functionality centric comments rather than a mere description of the technical components or files added/deleted/amended.

Confirm Visual Studio Code's prompt to always or for this one time to stage and commit changes in one operation.



There should be now no changes under the "CHANGES" expanded list.

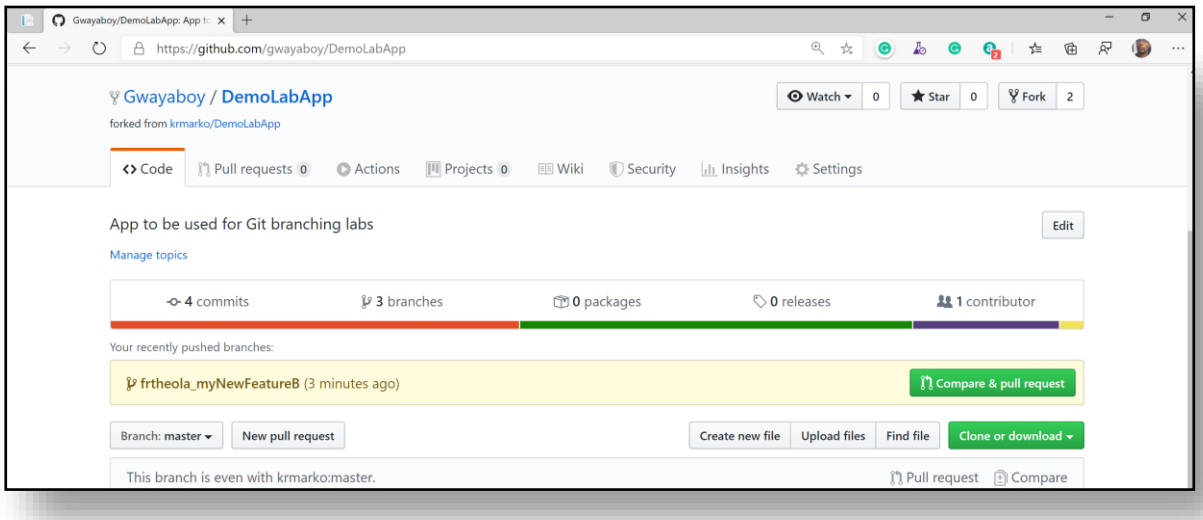
6) Finally let's push our committed changes and our new feature branch up to our remote repository.
Click on the 3 dotted contextual menu and select "Publish Branch…"



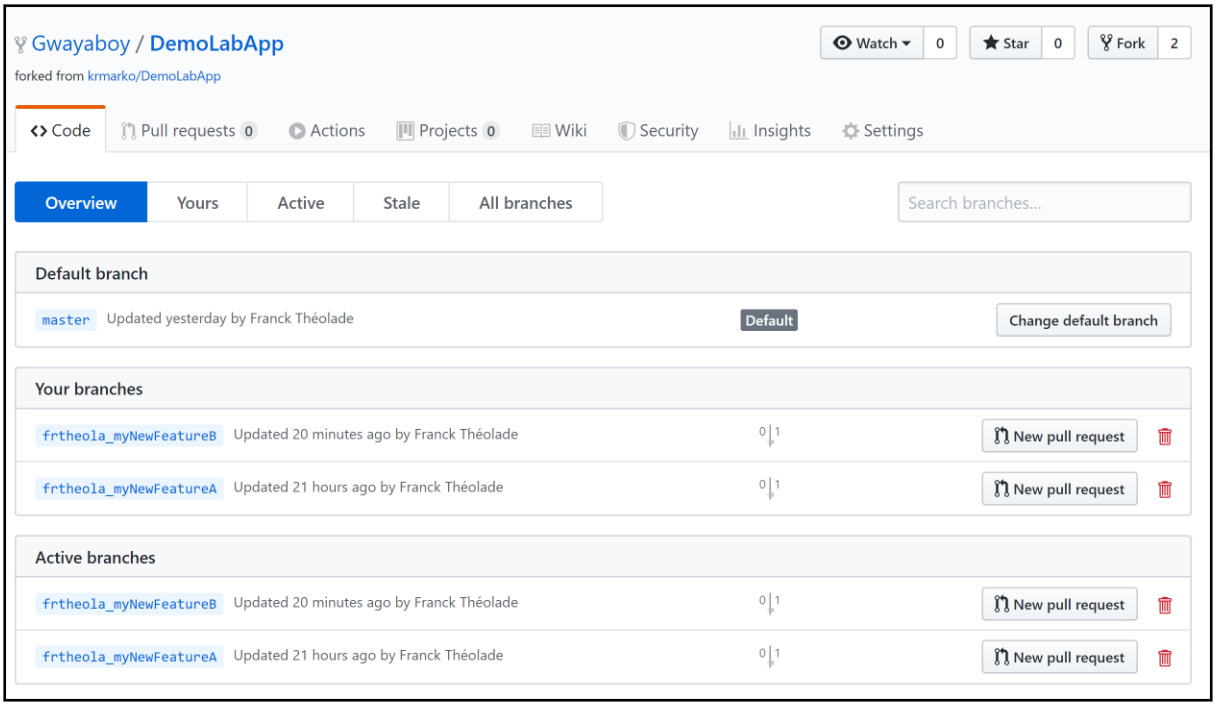
After a few seconds, the progress bar on top of the SCM panel should stop which signify the branch has been successfully pushed remotely.

Let's verify by navigating back to our GitHub Repository to *https://github.com/<yourGitHubUserName>/DemoLabApp.*

In my case that would be https://github.com/gwayaboy/DemoLabApp



7) Click on the branches tab next to the (4) commits tab.
You should now have 3 branches, master as the default branch and 2 active remote branches *<emailHost>_myNewFeatureA and <emailHost>_myNewFeatureB*.
In my case, as below, frtheola_myNewFeatureA and frtheola_myNewFeatureB.



8) To finish just get back to Visual Studio as you may have noticed the built-in Git extensions prompting to auto-fetch.
Please also notice that Visual Studio we commit a set of git commands into one operation such as “sync” which combines in this order git fetch origin, git push origin and git pull origin