Table of Contents

PROJECT WORKFLOW

Install the required software and create account:

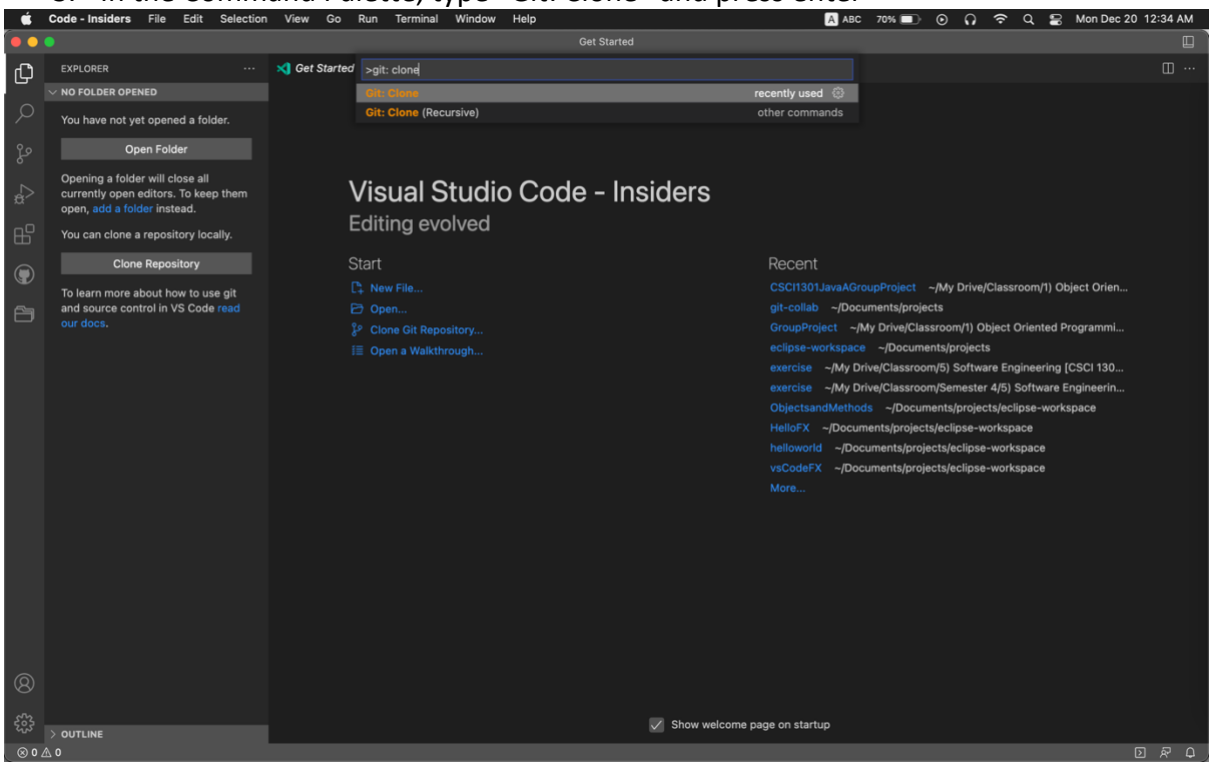
1. Visual Studio Code (VSCode)
2. Git
3. GitHub (create account)

Clone the project from the main repo (ask from Project Manager)

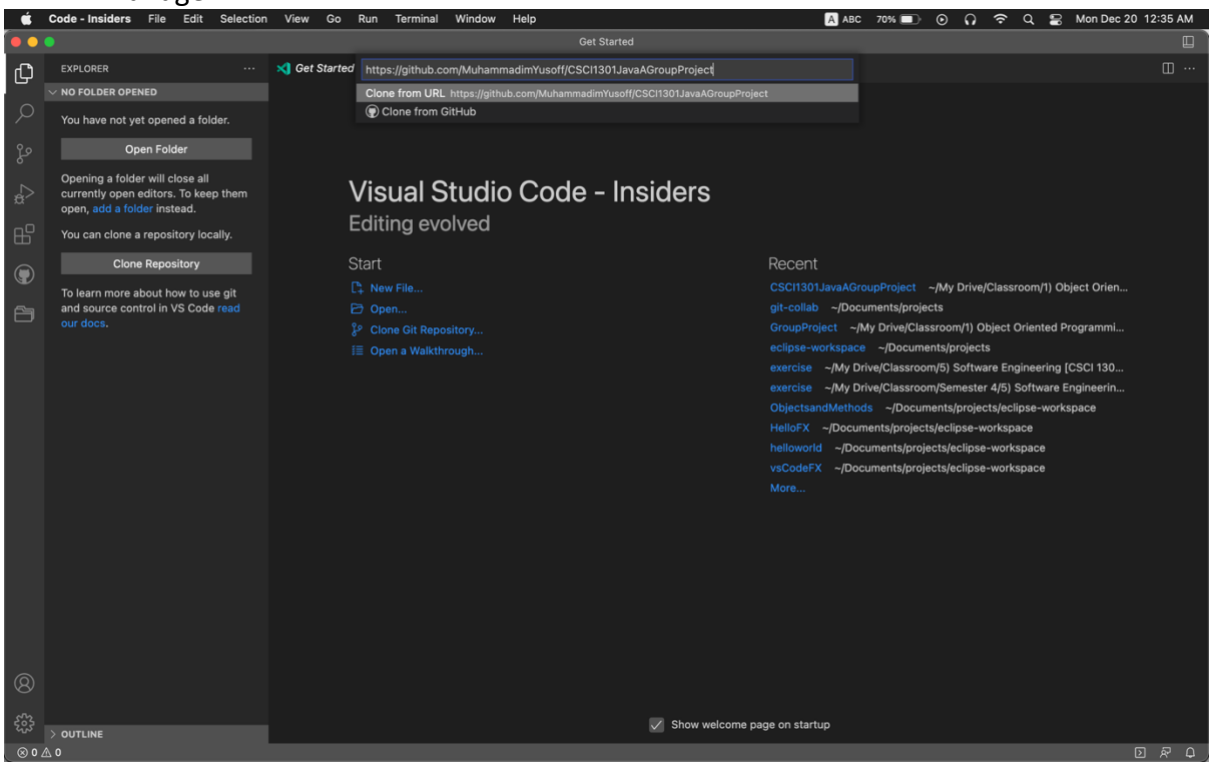
CLONING THE PROJECT

How to Clone from the main repo into your PC using VSCode.

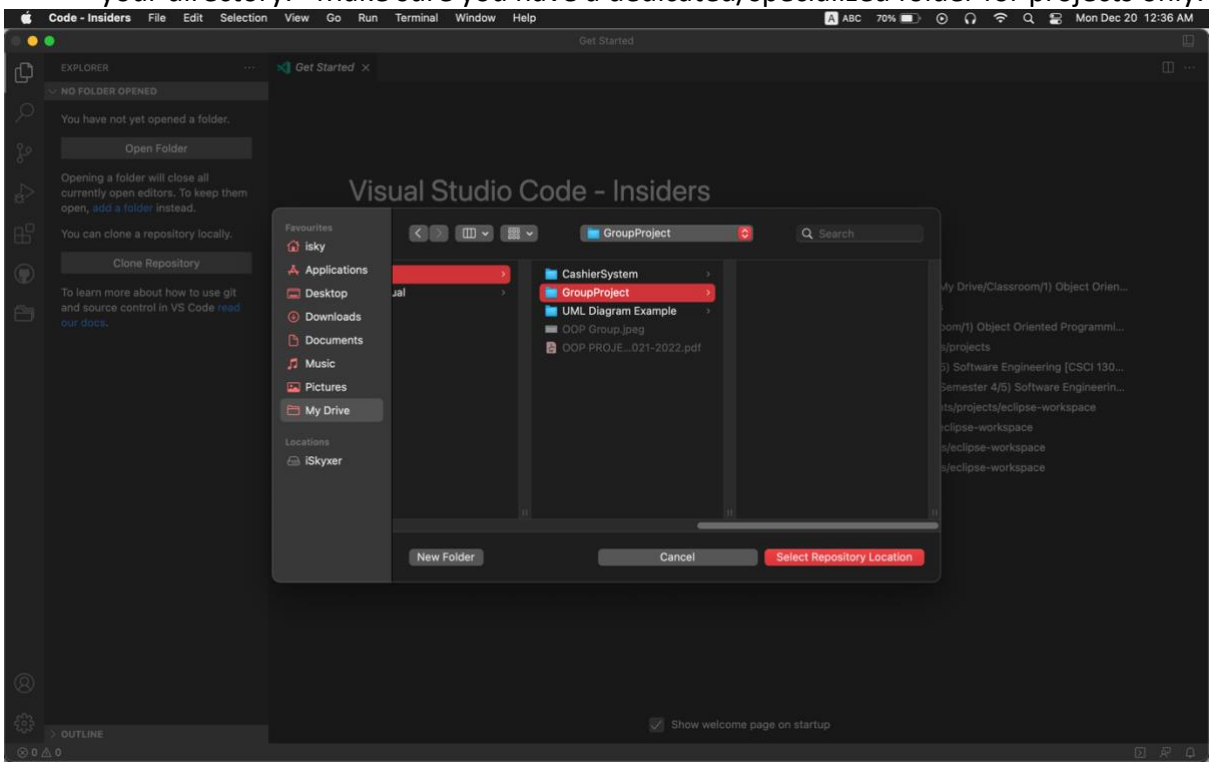
1. Open your VSCode
2. Go to “View” → “Command Pallete…” or press CTRL + SHIFT + P
3. In the Command Palette, type “Git: Clone” and press enter



4. The command will ask for the GitHub URL, paste the URL you got from Project Manager.



5. You will then ask for a directory where the files and folders will be placed, choose your directory. *Make sure you have a dedicated/specialized folder for projects only.

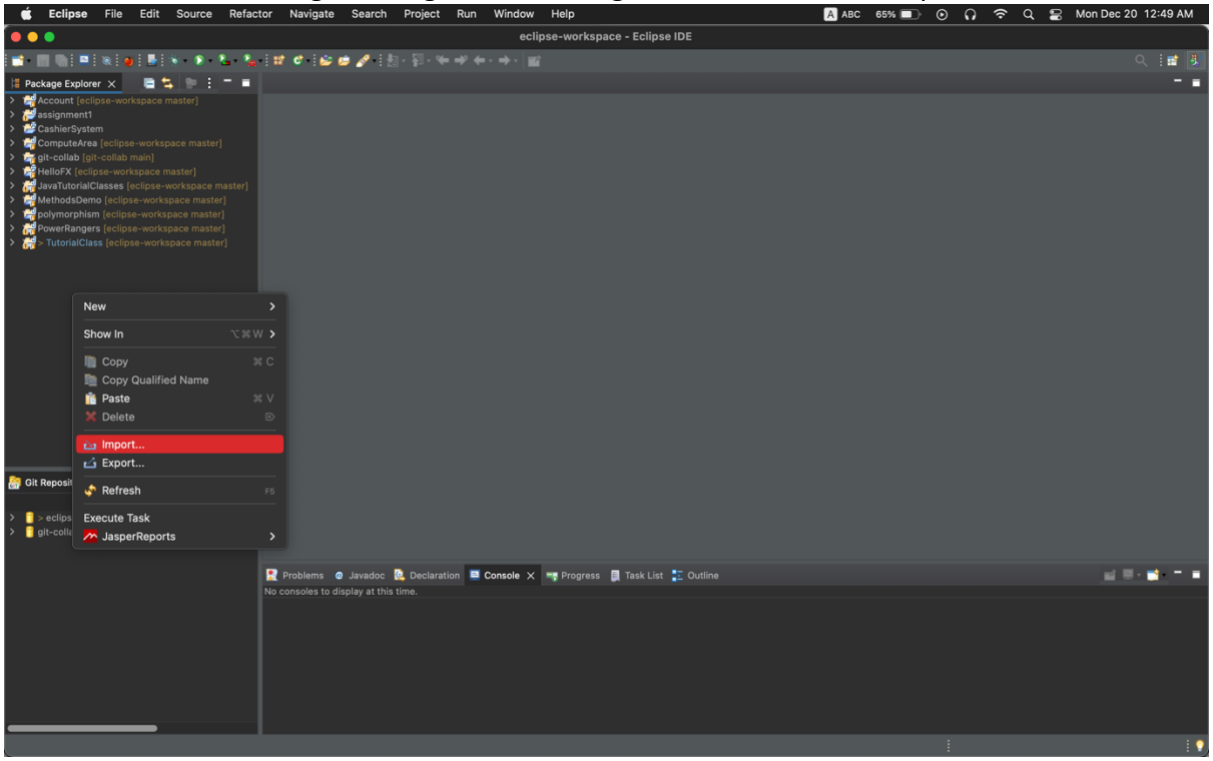


6. Click on “Select Repository Location”.

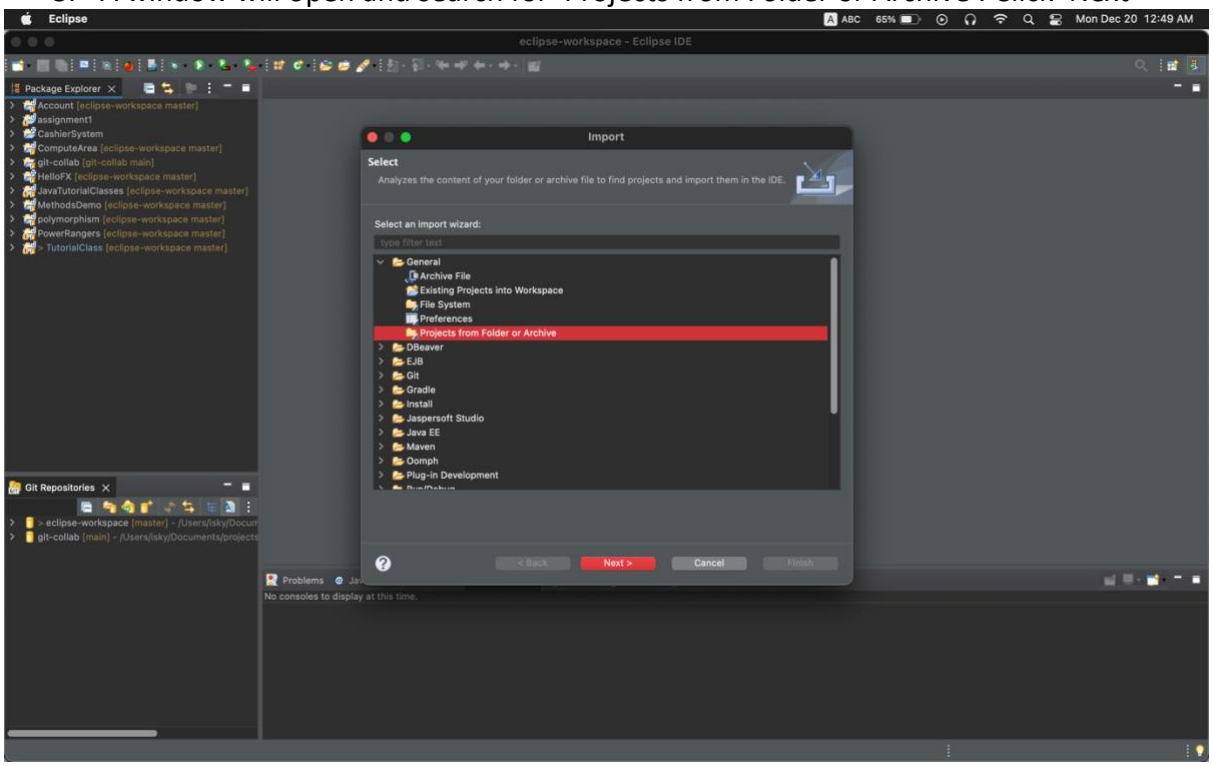
GETTING STARTED

IMPORTING INTO ECLIPSE WORKSPACE (OTHER IDE PLEASE DO YOUR RESEARCH)

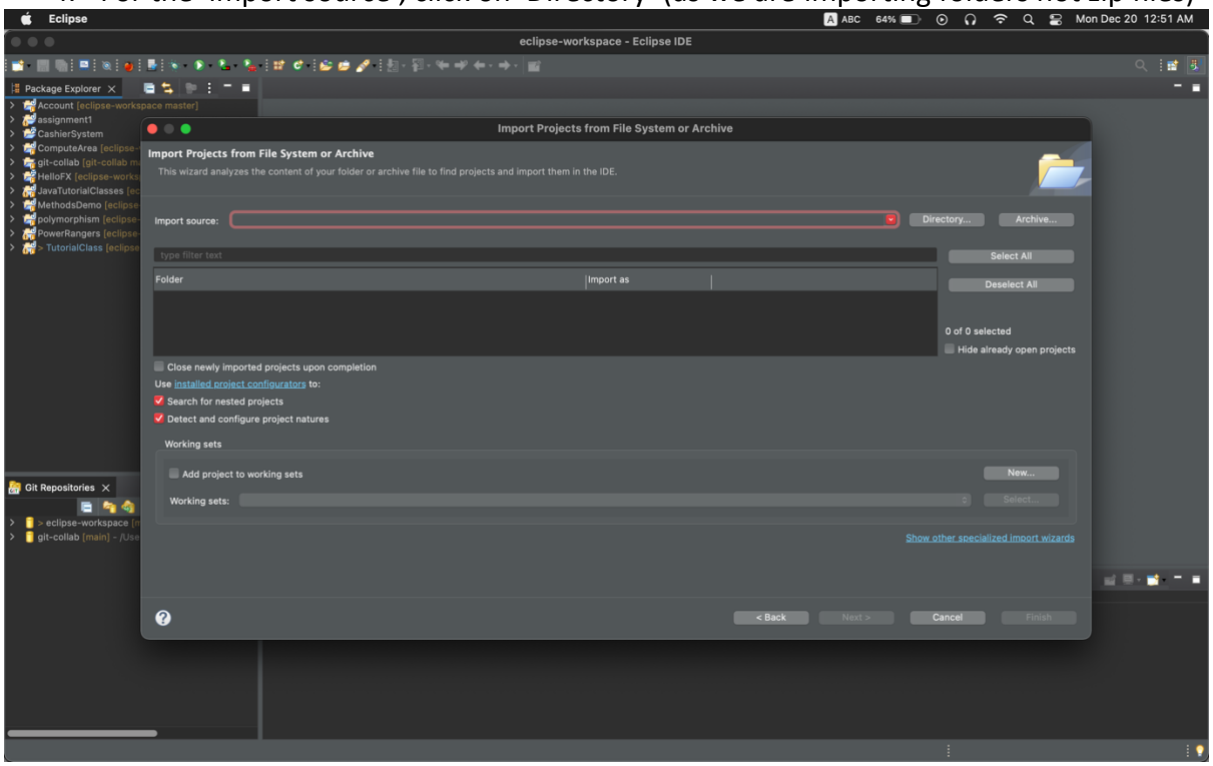
1. Open Eclipse IDE.
2. From the ‘Package Manager’ Section, right click and choose ‘import’.



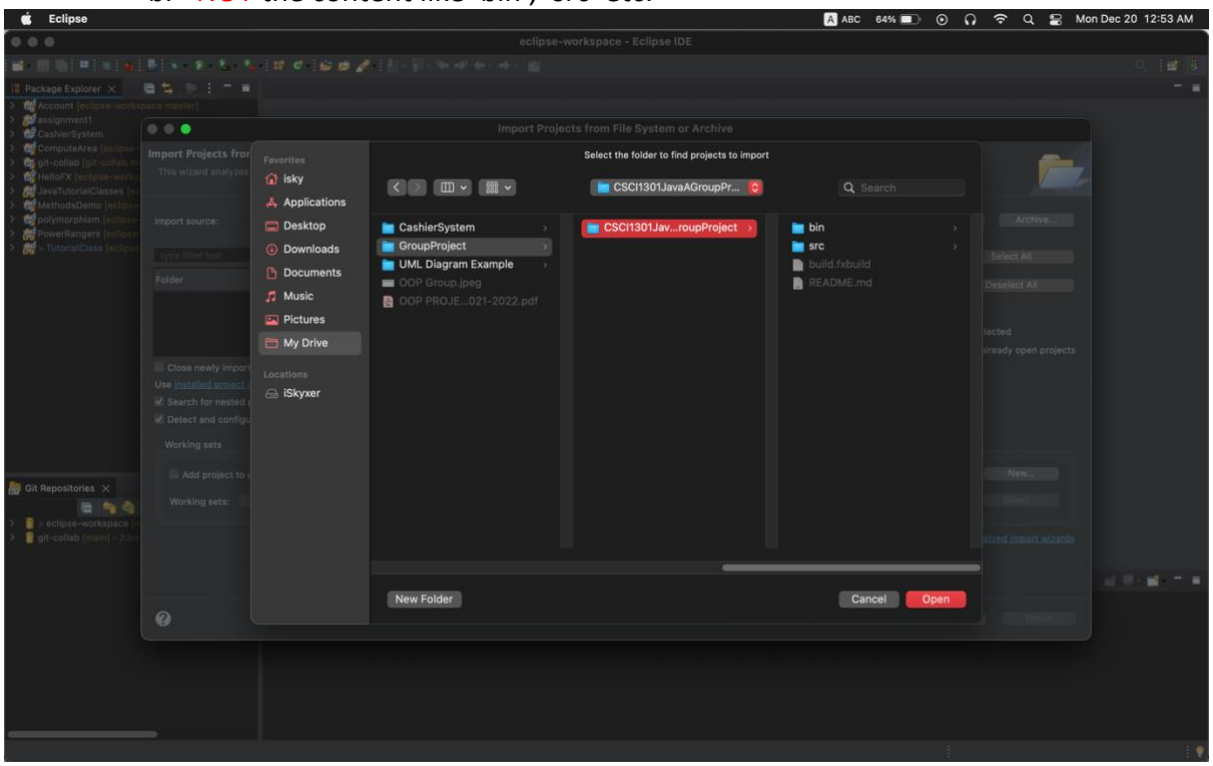
3. A window will open and search for ‘Projects from Folder or Archive’. Click ‘Next’



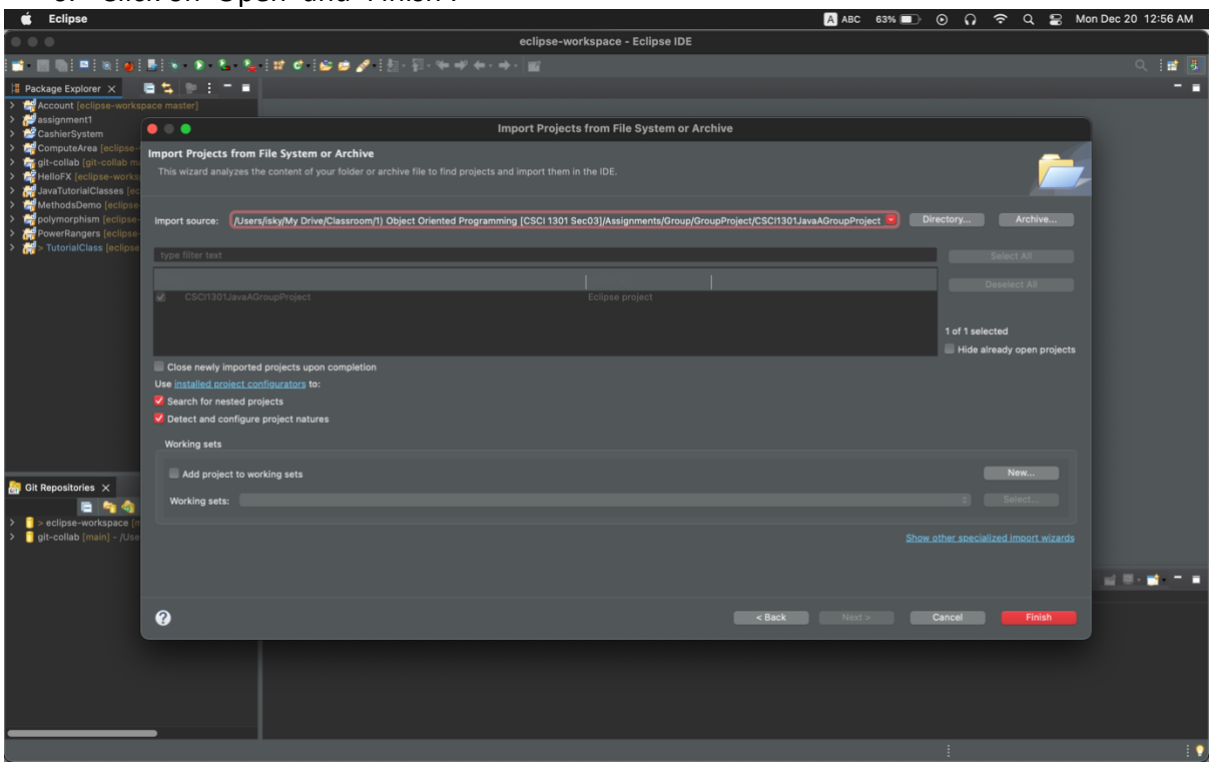
4. For the ‘import source’, click on ‘Directory’ (as we are importing folders not zip files)



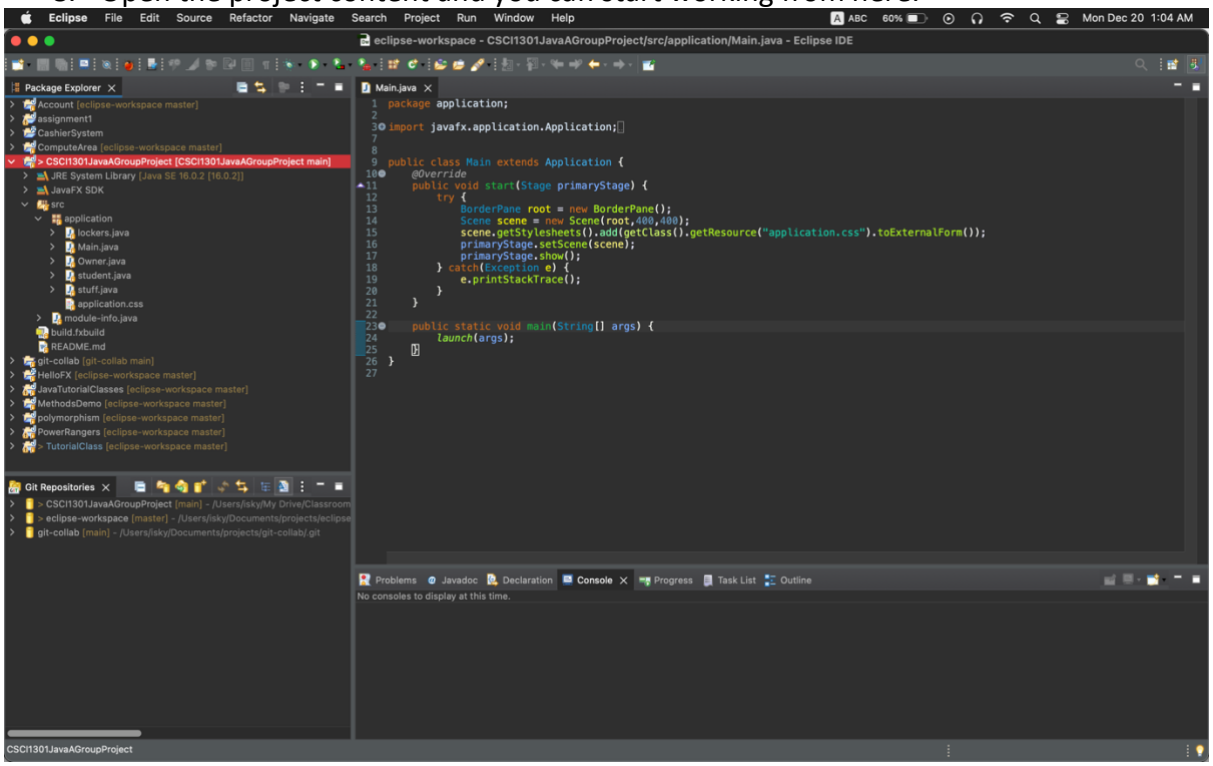
5. Navigate to the directory you choose in VSCode as the project repository.
    a. Choose the <main folder name> (example here CSCI1301…).
    b. NOT the content like ‘bin’, ‘src’ etc.



6. Click on ‘Open’ and ‘Finish’.



7. The project folder will be loaded into your Eclipse workspace as below.
8. Open the project content and you can start working from here.

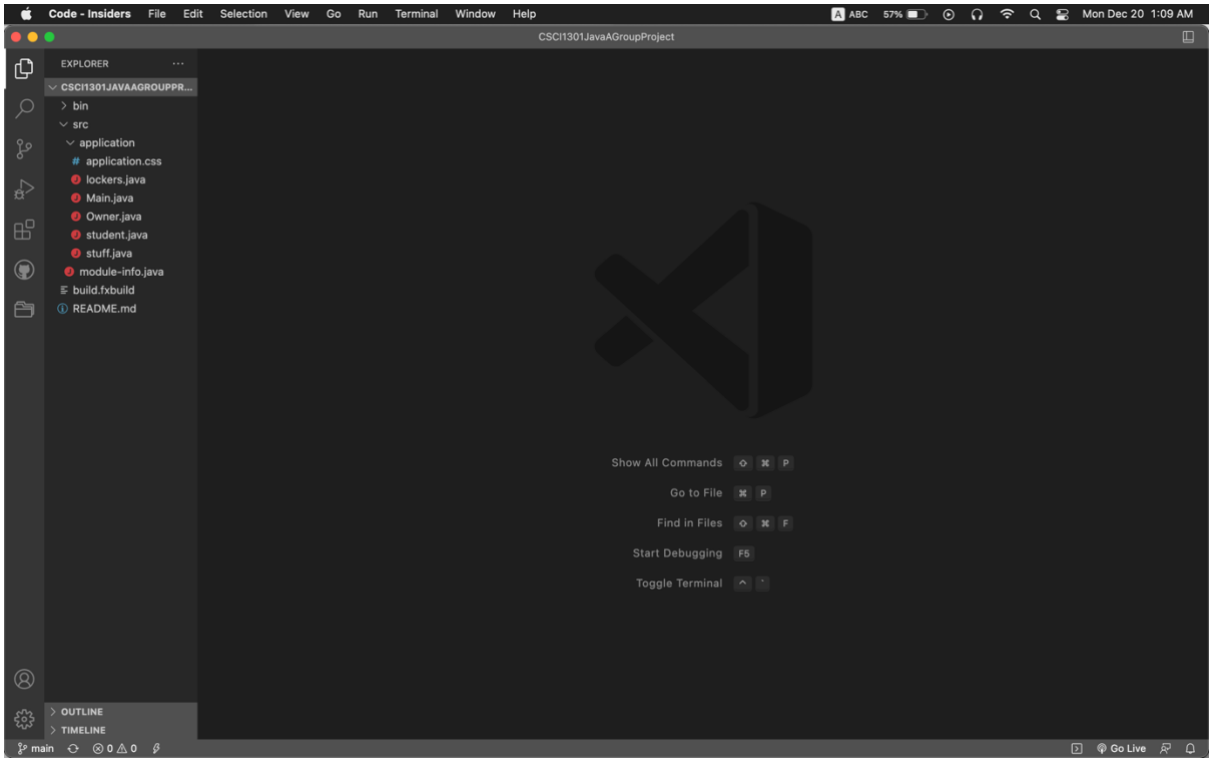


!!!!MAKE SURE:

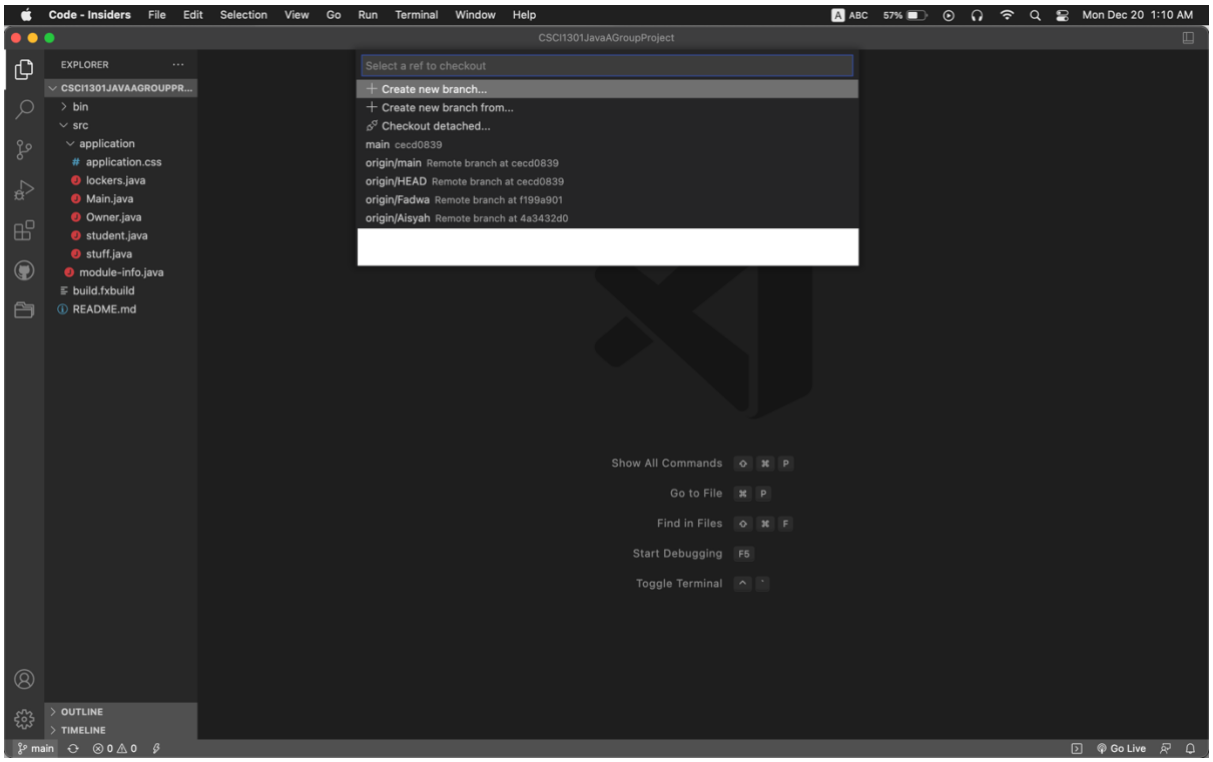
- SAVE YOUR CHANGES. VSCode will automatically pickup any changes in your file.

CREATING YOUR BRANCH (from VSCode)

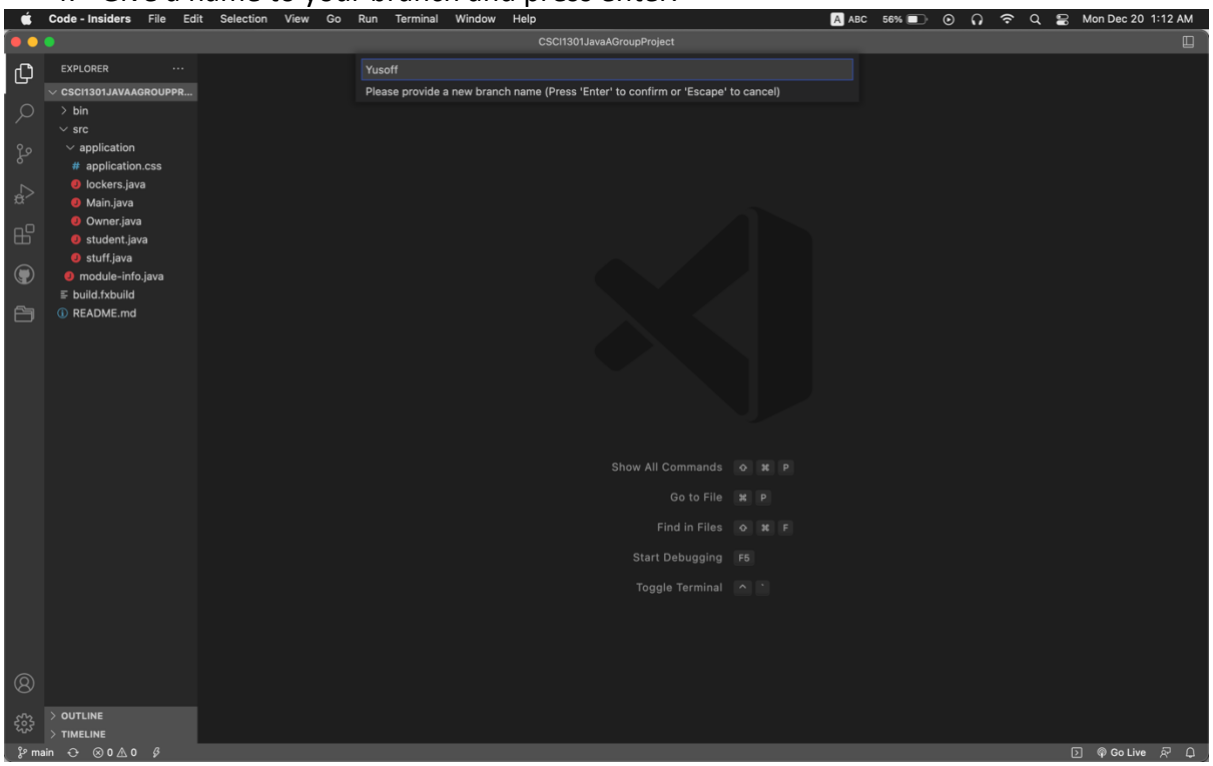
1. Open/Load the Project using your VSCode. You will see the same content as in Eclipse.
2. At the bottom left, VSCode shows Branch Icon with branch name you are in, usually the ‘main’ or ‘master’ branch as name.



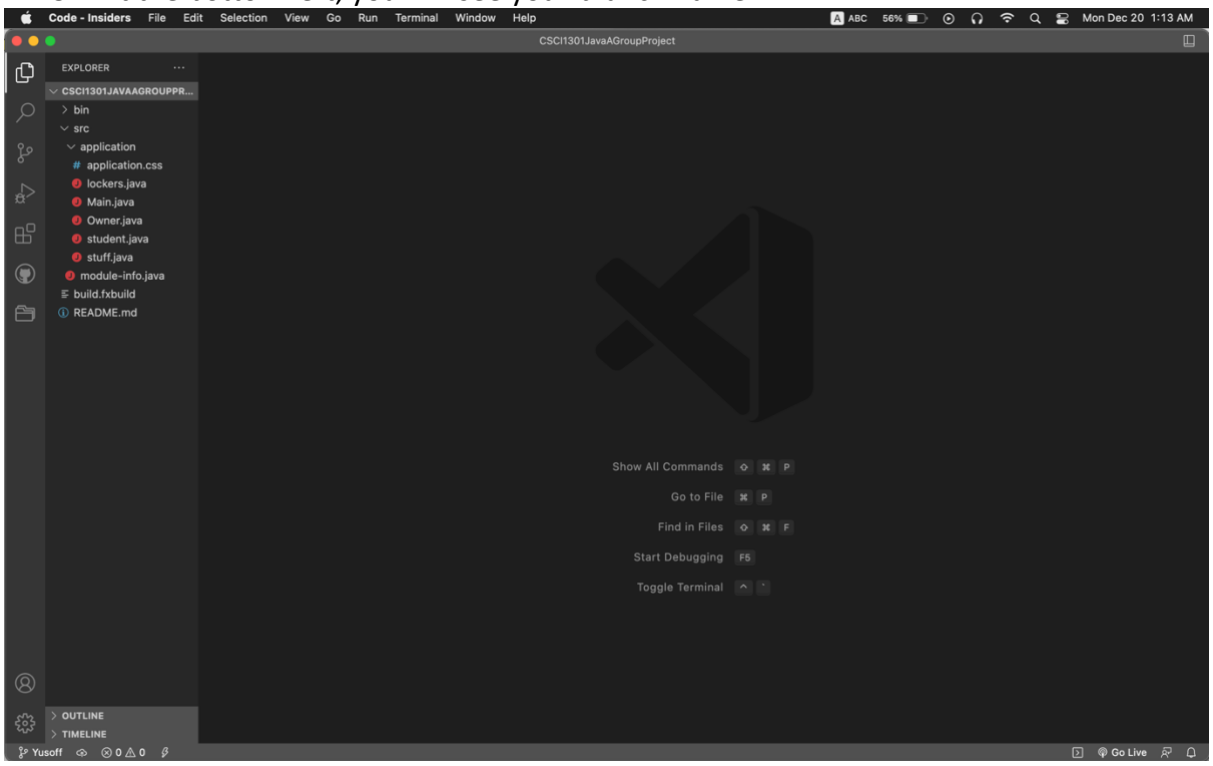
3. Click on the Branch Icon → at the top/Command Pallete you will see “+ Create new branch…” and select it.



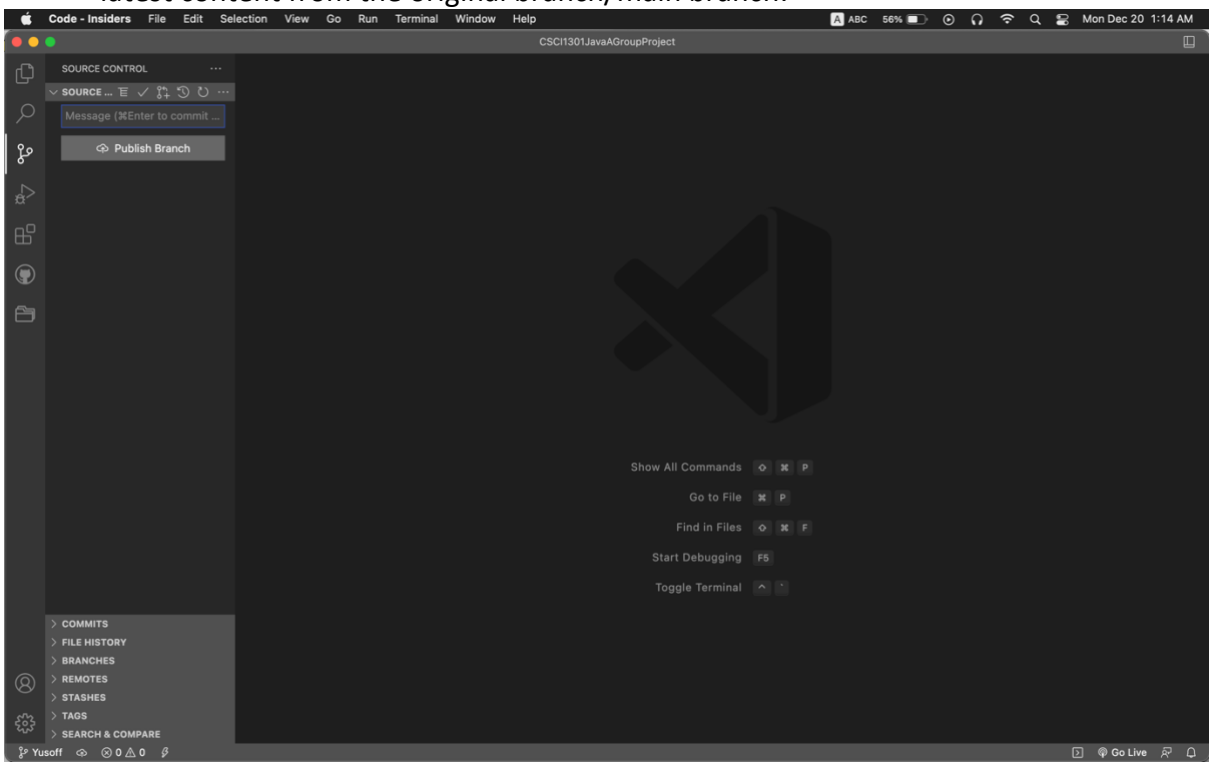
4. Give a name to your branch and press enter.



5. At the bottom left, you will see your branch name.



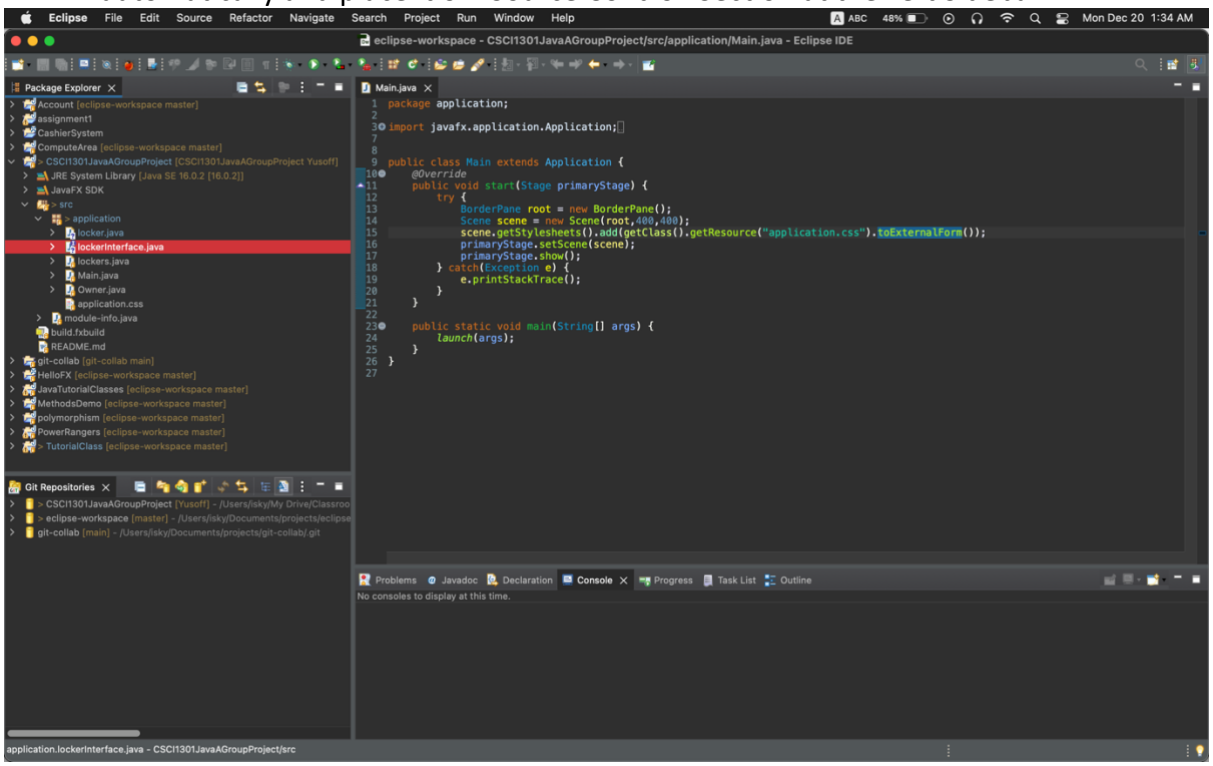
6. Click on the ‘Source Control’ at the Left sidebar
7. You can see ‘Publish Branch’, just click the button.
   - If you see a ‘Pull Request’ related notification just ignore it as you already have the latest content from the original branch/main branch.



8. You are done.

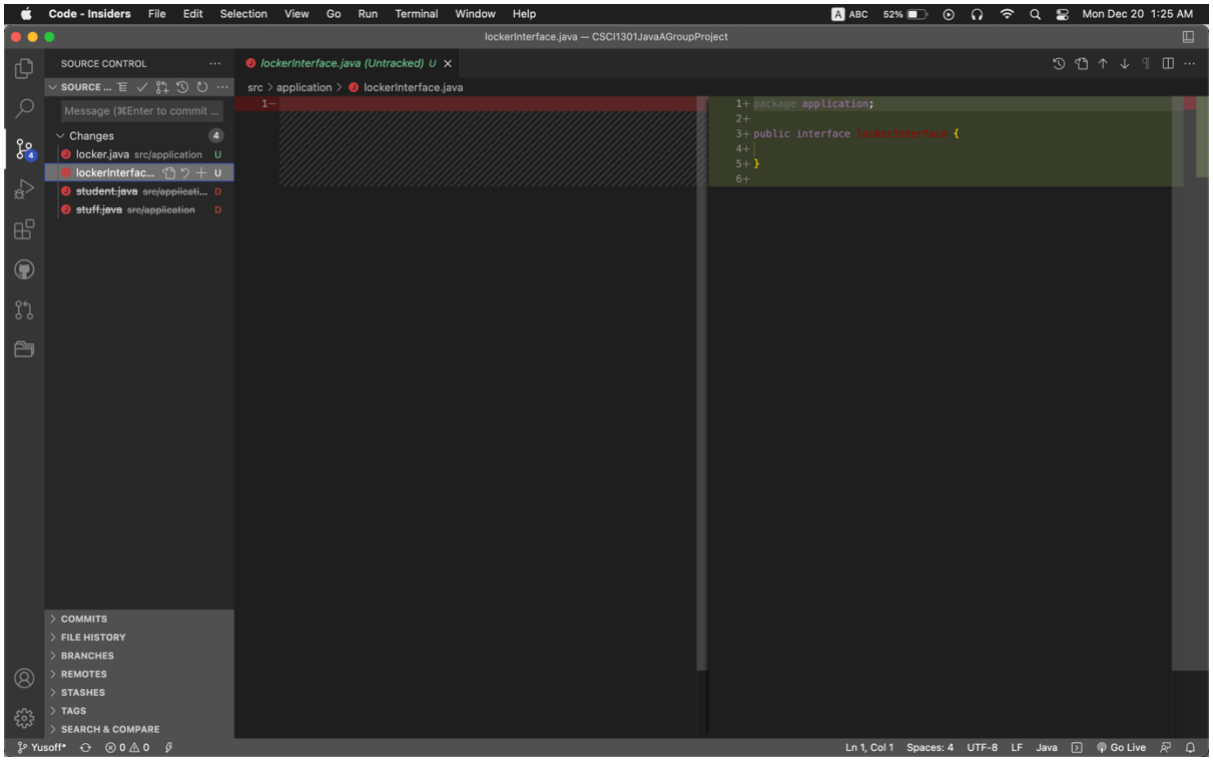
WORKING ON YOUR BRANCH (Eclipse and VSCode)

1. [Using Eclipse IDE] Open a file you wanted to start coding on.
2. Make sure you save your work inside Eclipse. VSCode will pick up the changes automatically and place it on ‘Source Control’ Section at the Left Sidebar.



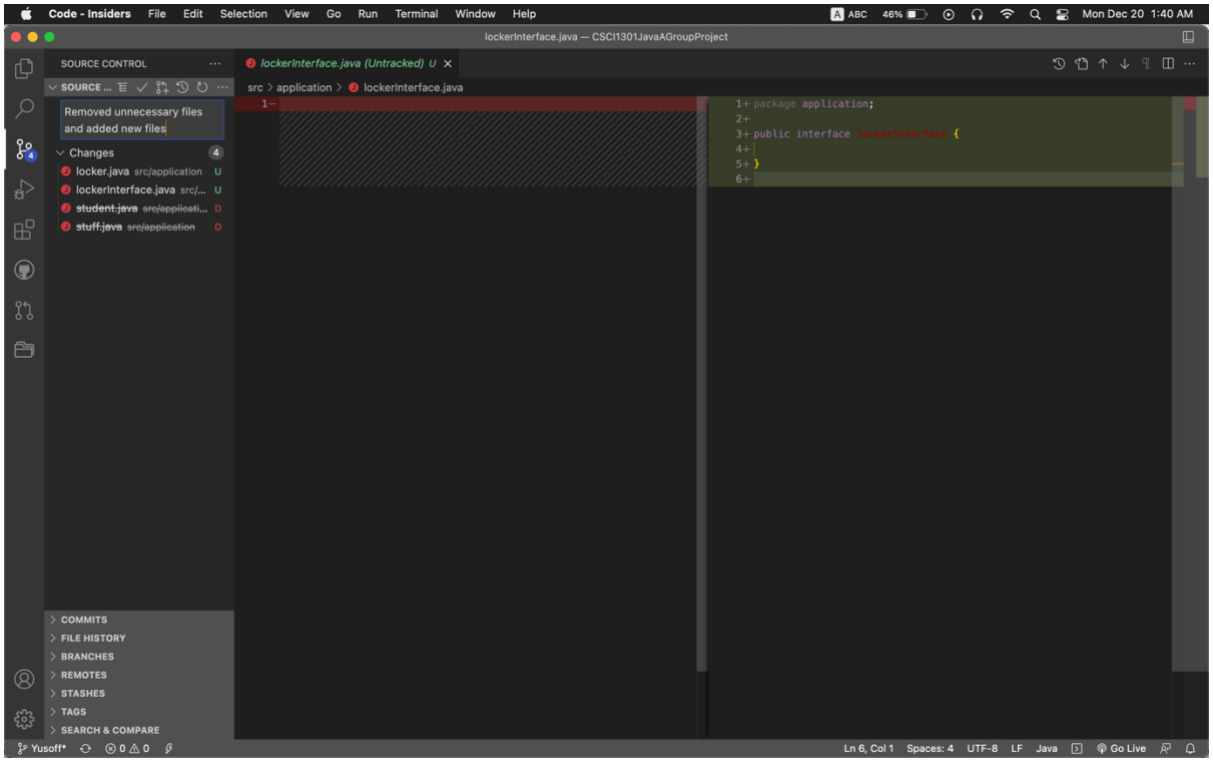
<CONTINUE TOTHE NEXT PAGE>

3. [Using VSCode] Make sure you are on your branch, check it on bottom left side.
4. ‘Version Control’ on the Left sidebar will detect you have made changes to the files, the number denotes how many files have been changed.
5. Click on the Version Control Icon, it will display the files and the changes if you click to any files from there.
6. You can click on any files you have changed and see the changes you have made.
    a. The file changes:
        i. The left side shows the original file content.
        ii. The right side shows the changes you have made.
    b. The File content color
        i. Red: Indicate the codes removed from the line.
        ii. Green: indicate the codes added into the line.



PUSHING TO YOUR BRANCH (COMMIT & PUSH TO YOUR BRANCH)

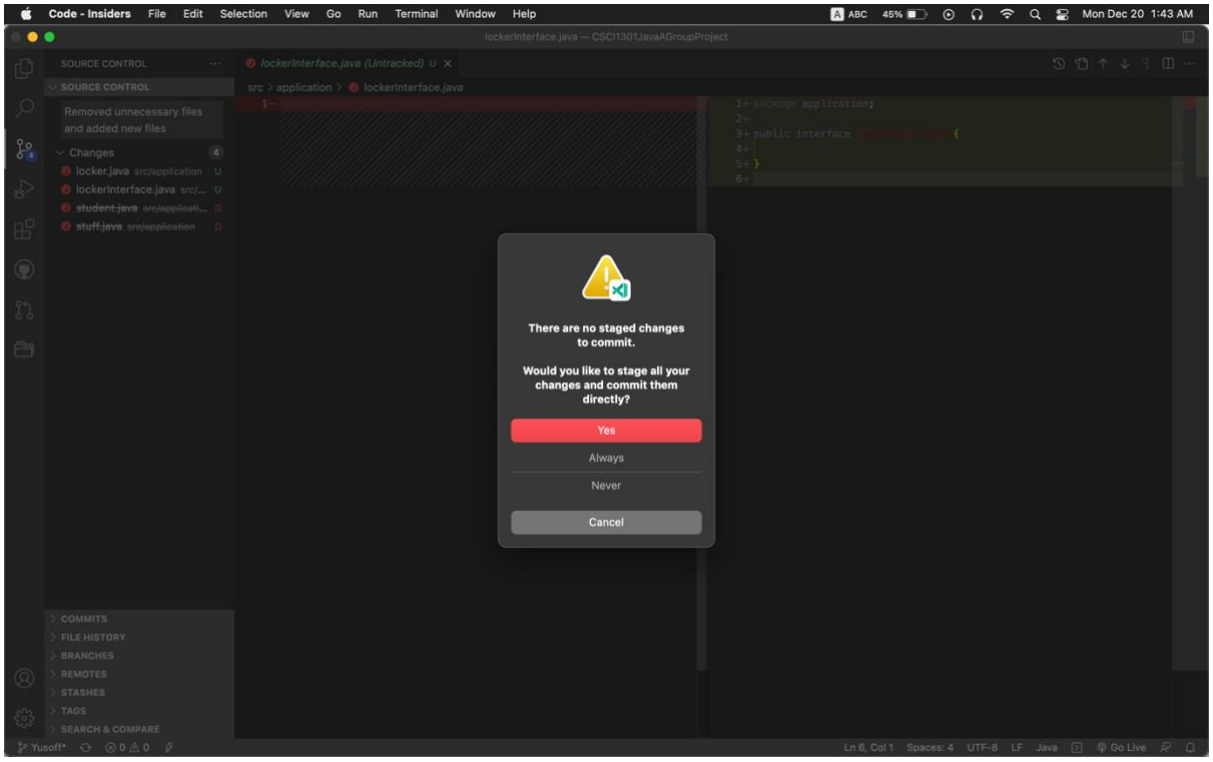
1. Write your ‘commit’ message on the textField, you have max of 50 characters as this is the commit title in GitHub. If you want to make a full commit you might need to follow the GitHub Workflow.



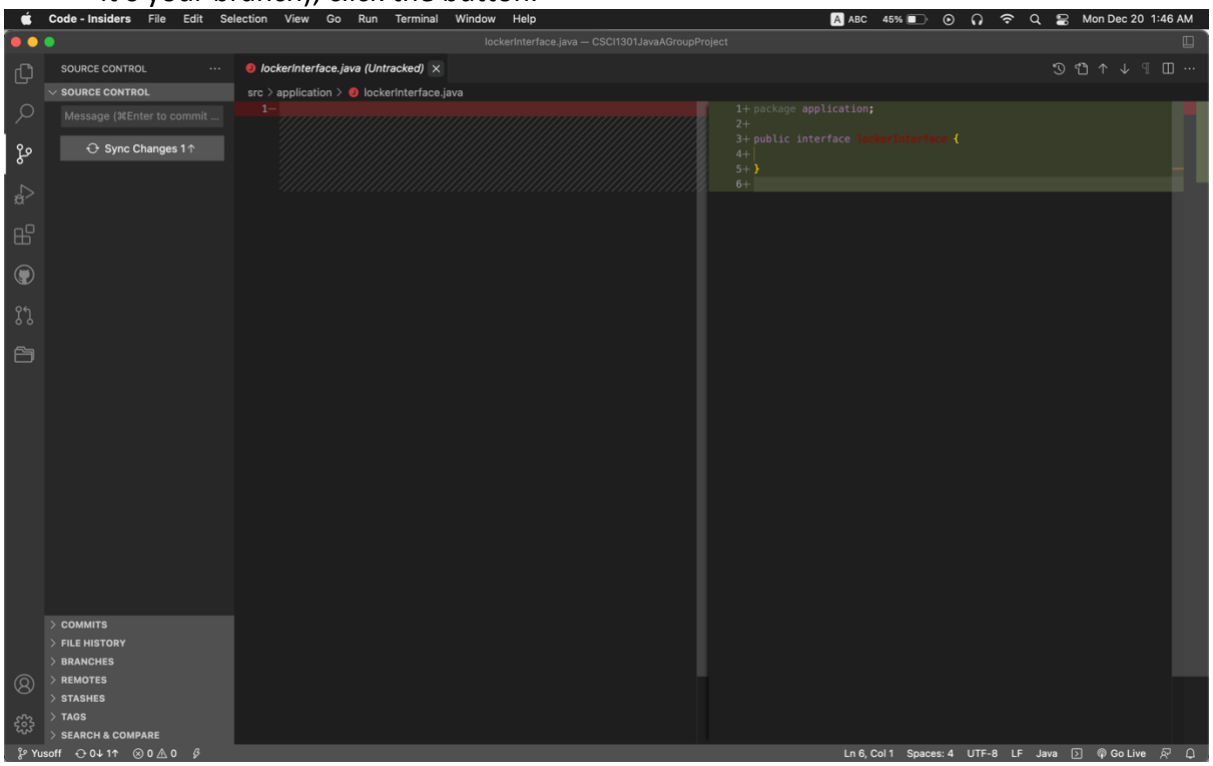
2. Click on the ‘tick’ icon just above the commit message textField. A dialog will appear asking you to stage the changes, just click on the ‘Yes’ option.

- You can choose the ‘Always’ option but not recommended as sometimes you need to make some rush changes before the ‘staging’ phase.

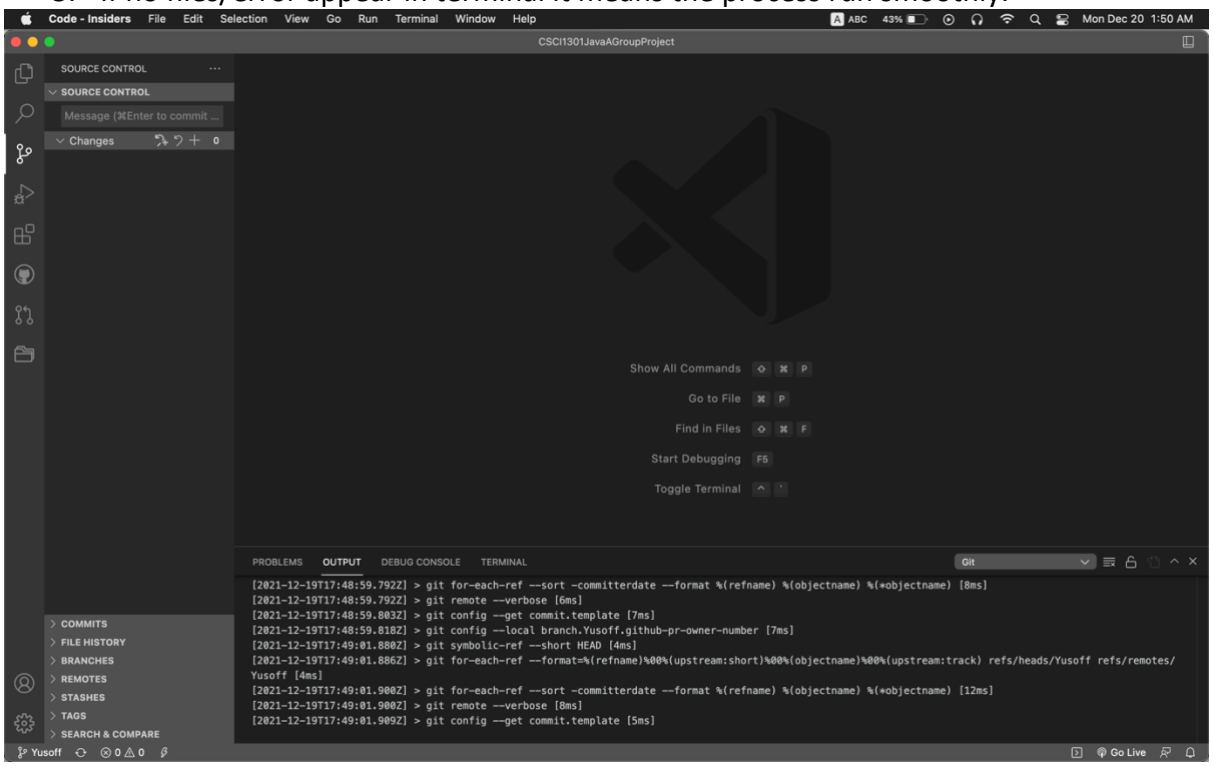
<LOOK AT THE NEXT PICTURE>



3. You will see the files have been ‘staged’ so you can push it to your remote repository (in this case GitHub)
4. A button ‘Sync Changes’ means you will push the files to your remote (for this one it’s your branch), click the button.

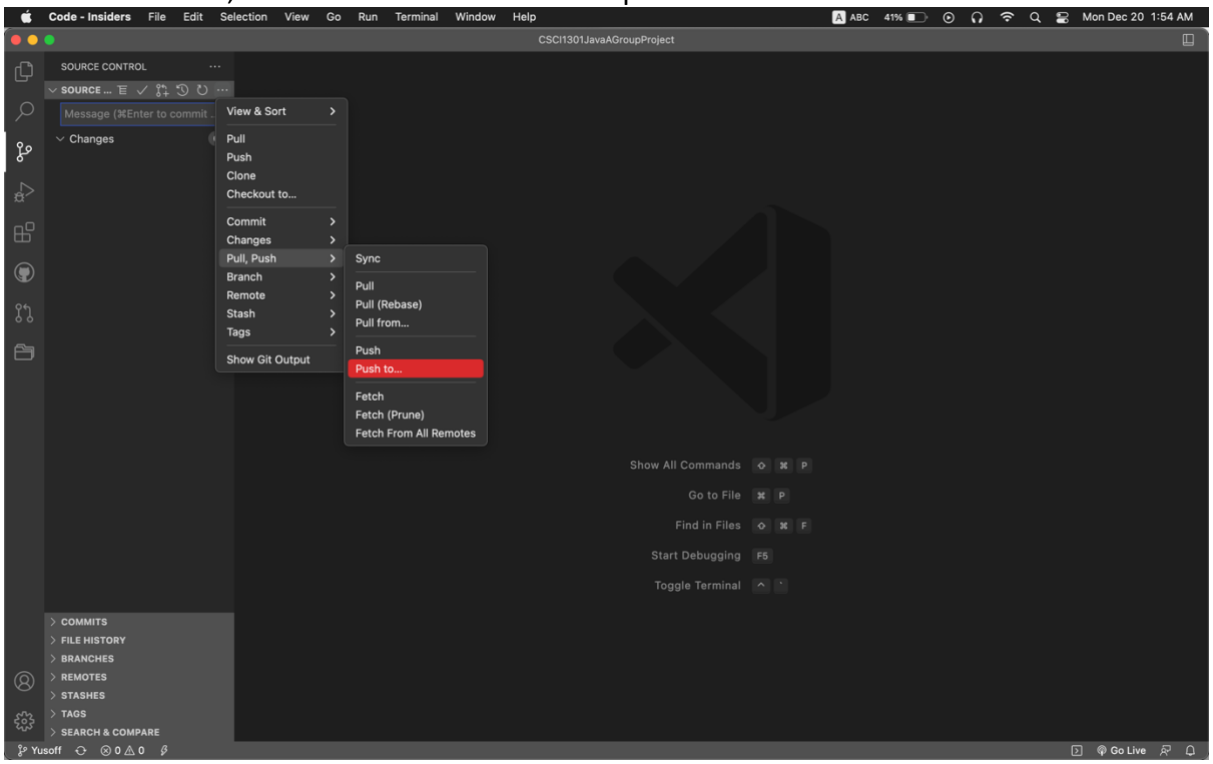


5. If no files/error appear in terminal it means the process run smoothly.

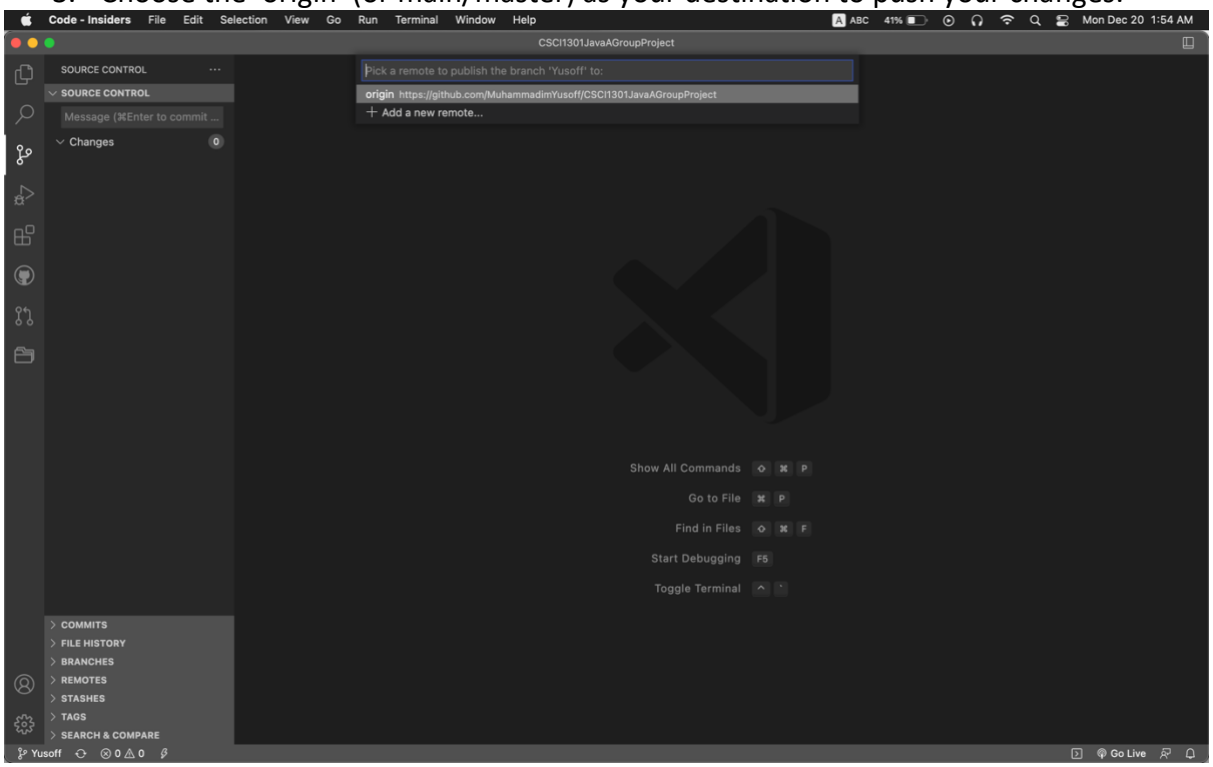


PUSHING TO MAIN BRANCH (COMMIT & PUSH TO MAIN BRANCH)

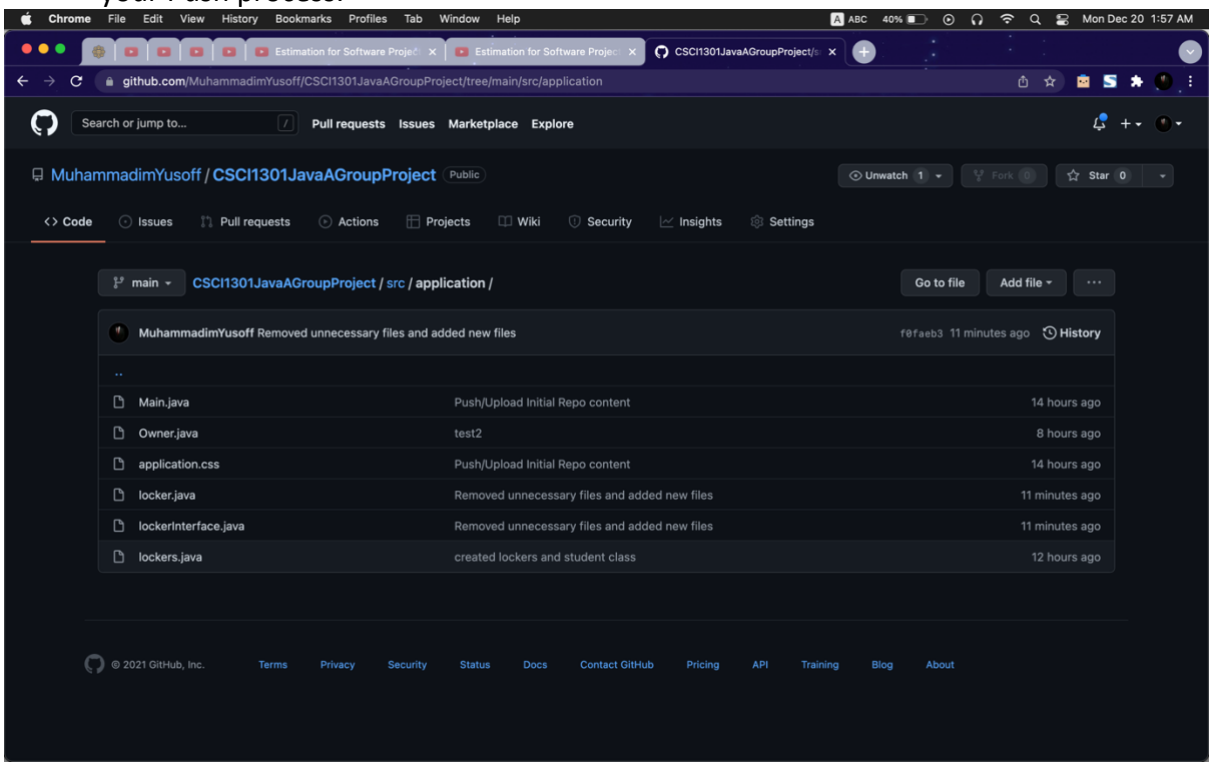
1. Click on the 3 dots besides the ‘Source Control’ just above the ‘message commit’ textField
2. Goto ‘Pull, Push’ → Choose ‘Push to…’ option.



3. Choose the ‘origin’ (or main/master) as your destination to push your changes.

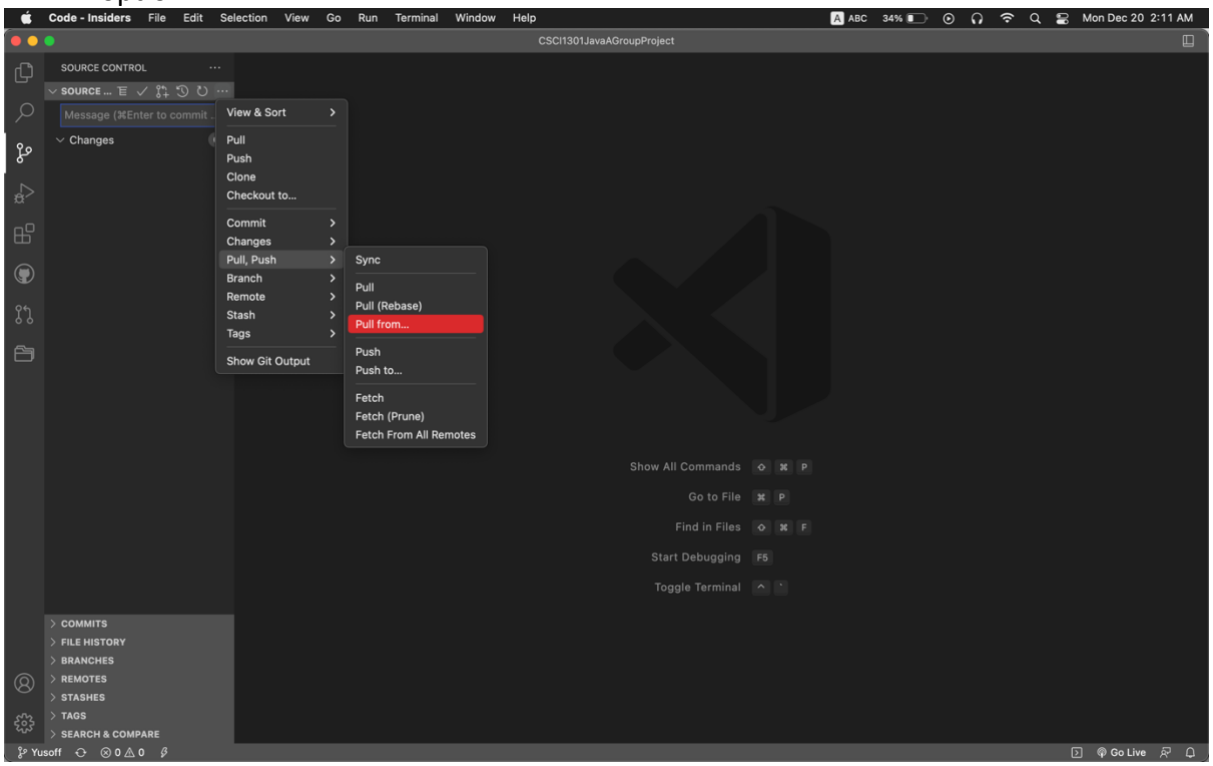


4. After the process finished, you can check the master/main branch on github.com to verify your changes.
5. If it's not appearing, contact the Repository Owner (Project Manager) to approve your Push process.

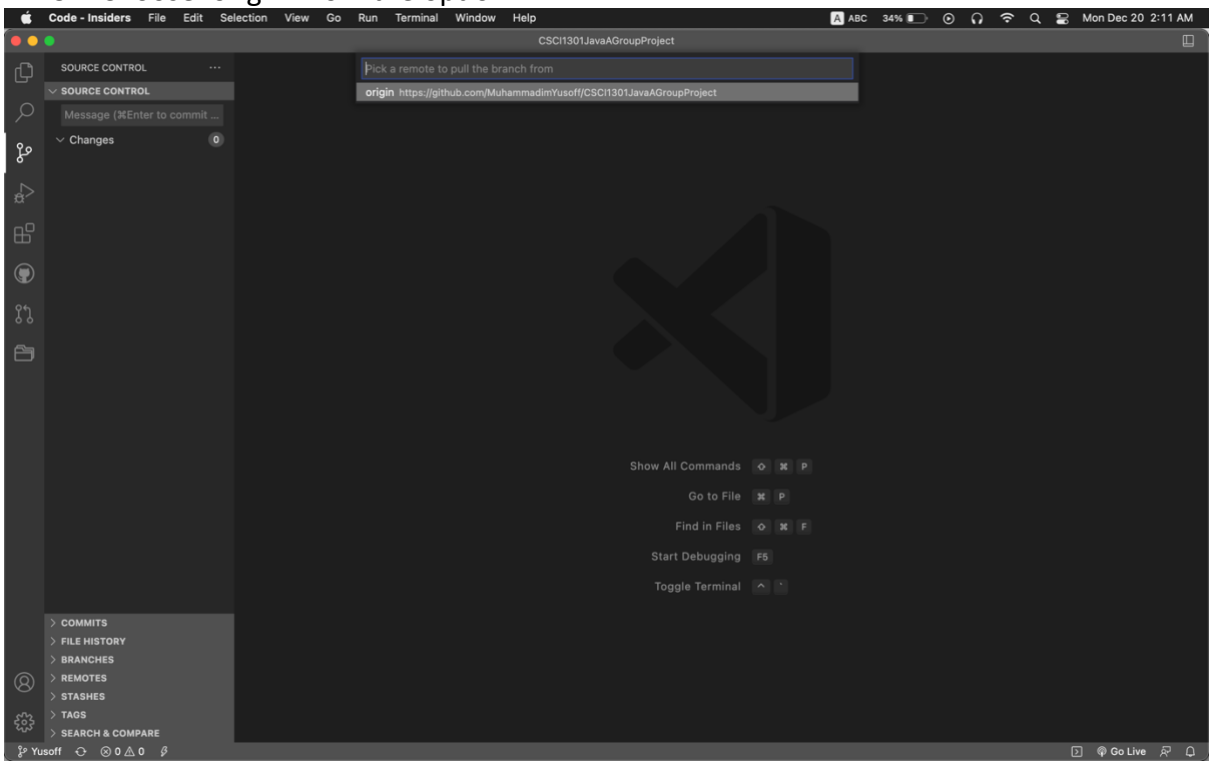


PULLING FROM MAIN BRANCH

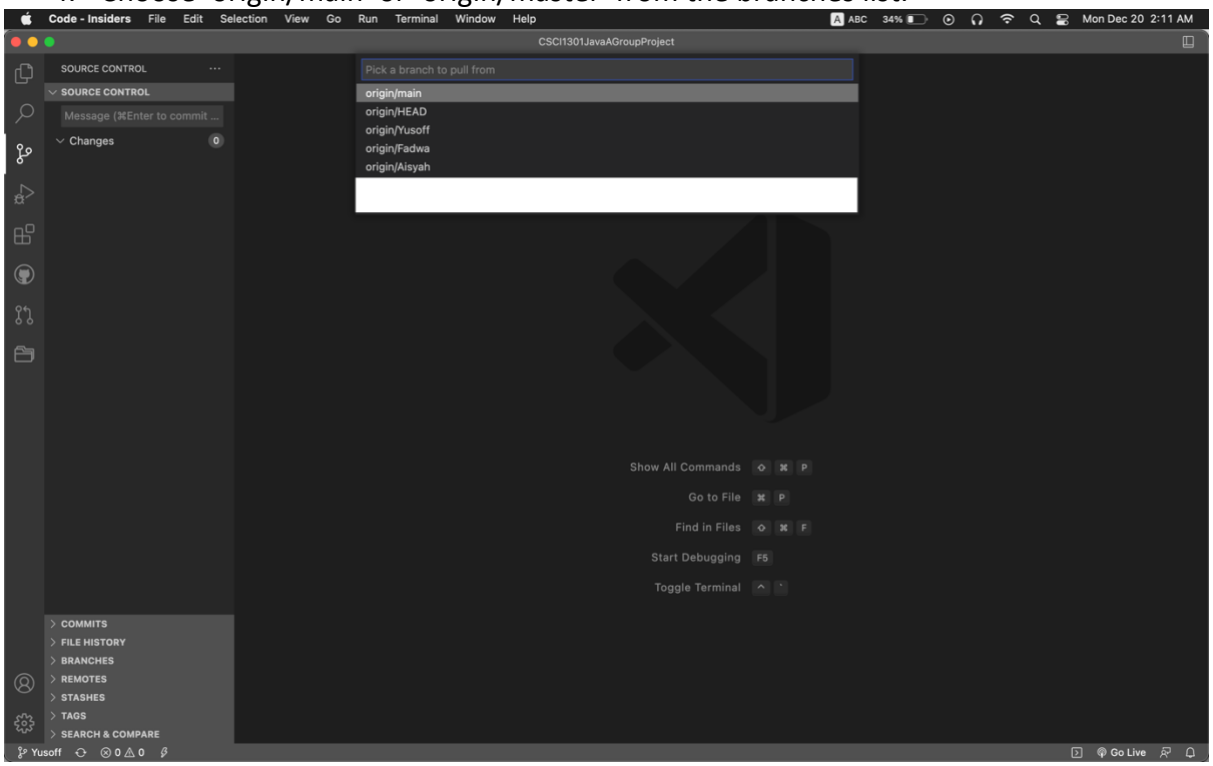
1. [Using VSCode] Open the awesome IDE VSCode and choose the ‘Source Control’ from the left sidebar.
2. Click on 3 dots besides the ‘Source Control’ → ‘Pull, Push’ → Choose ‘Pull from…’ option.



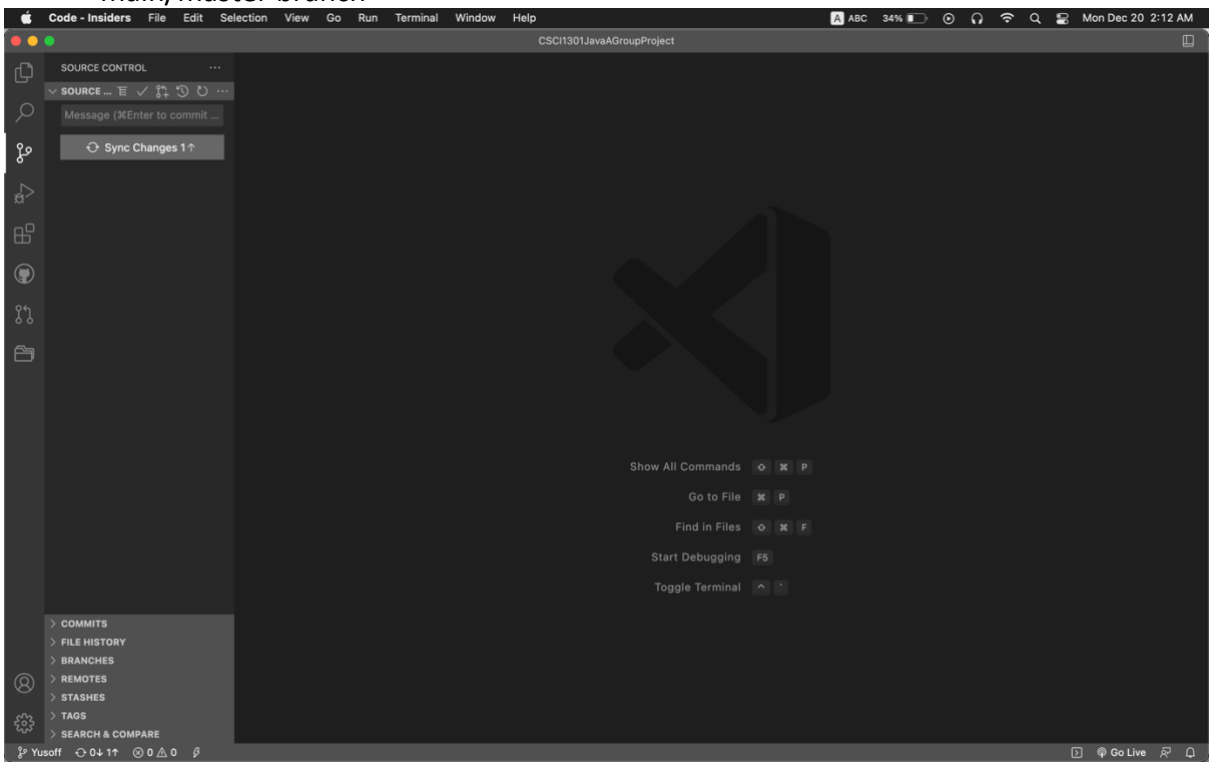
3. Choose ‘origin’ from the option



4. Choose ‘origin/main’ or ‘origin/master’ from the branches list.



5. If you see any ‘Sync Changes’ button, just click on it to sync your branch with the main/master branch

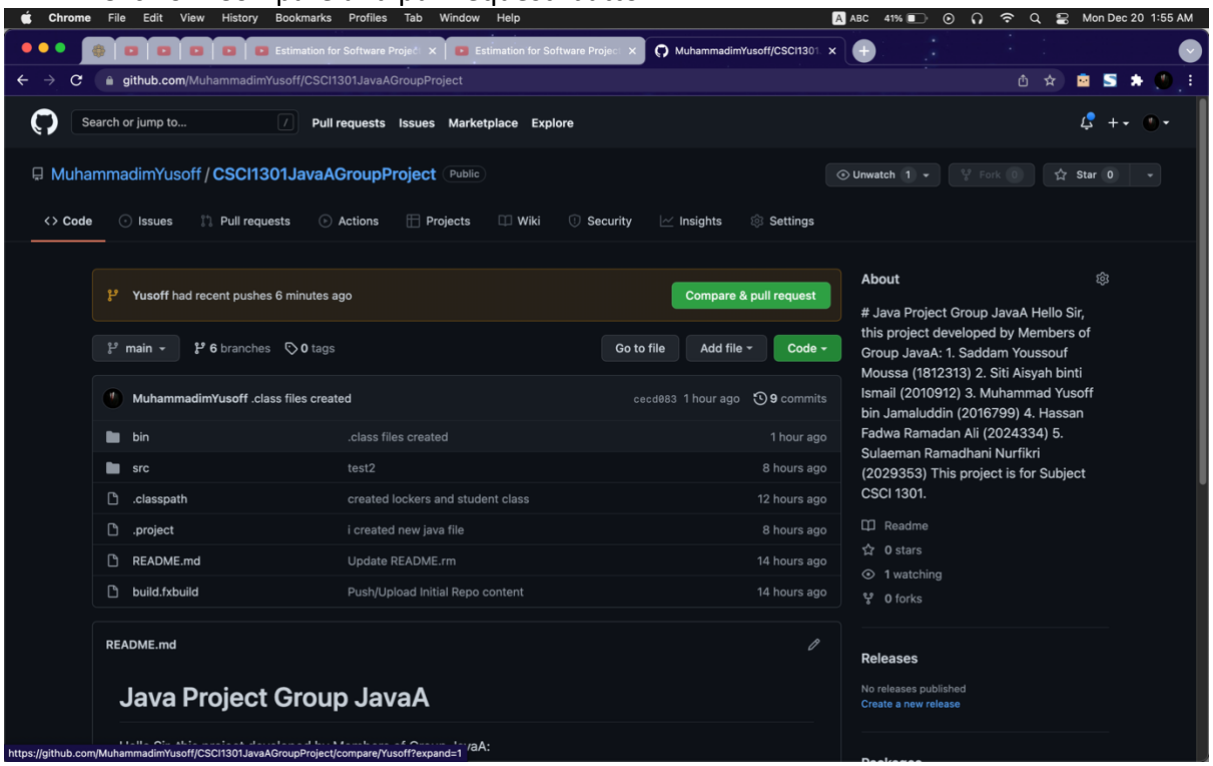


PROJECT MANAGER/REPOSITORY OWNER WORKFLOW

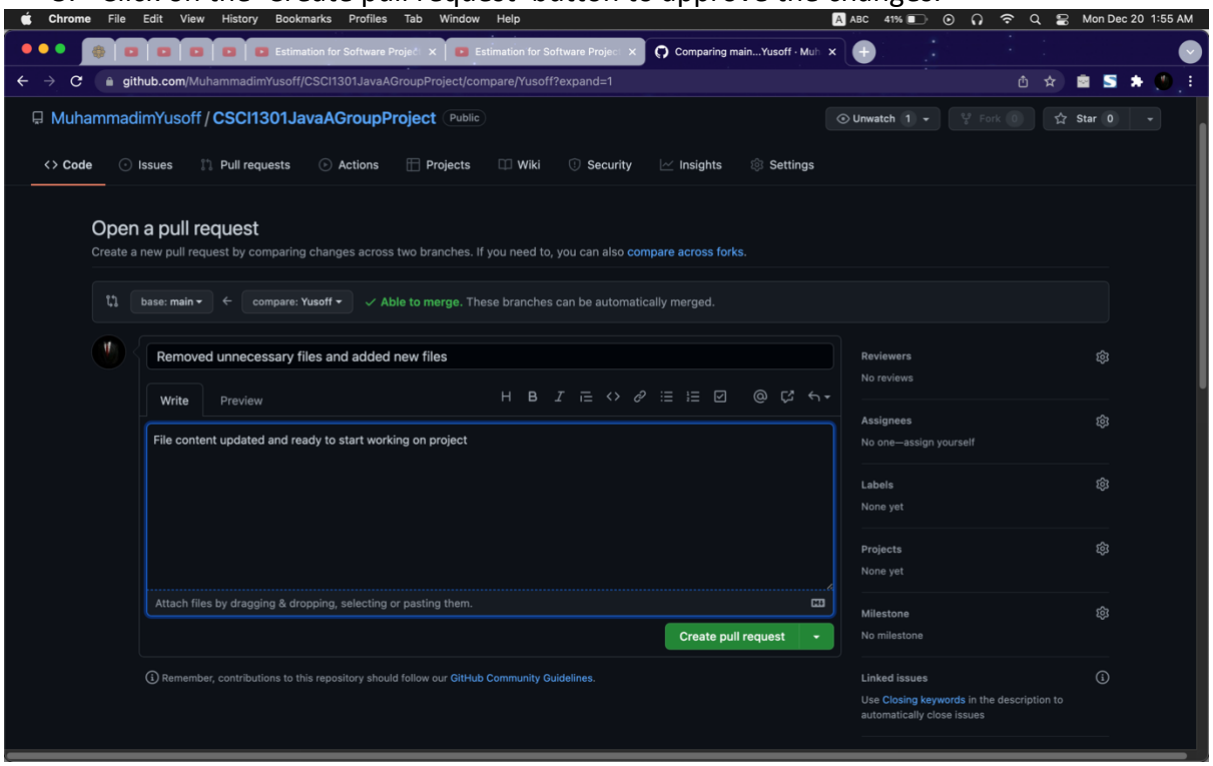
WHEN?

- If any of your Collaborators have problem with their ‘pull’ process, you need to check the ‘pull request’ from github.com.

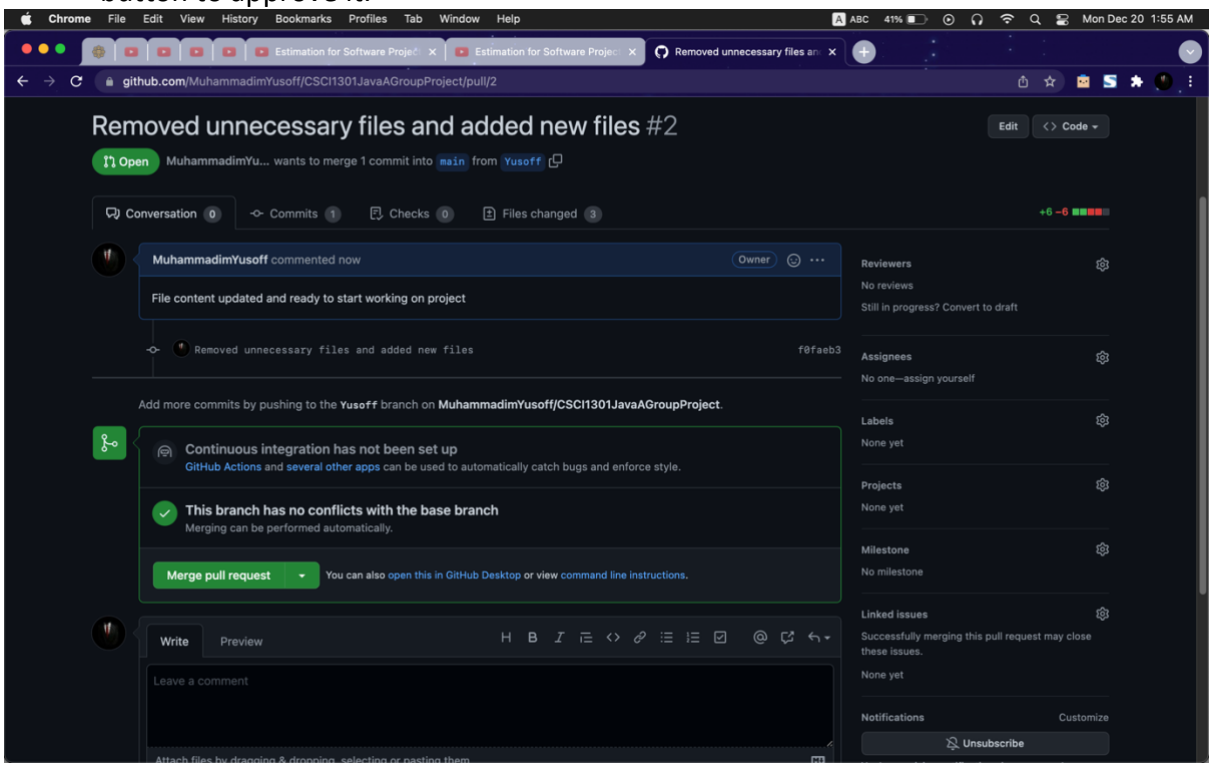
1. Open your GitHub repository and make sure it’s on the main branch. There you will see the pull request for the push your collaborators have made previously (if they have pull process hindered).
2. Click on ‘Compare and pull request’ button.



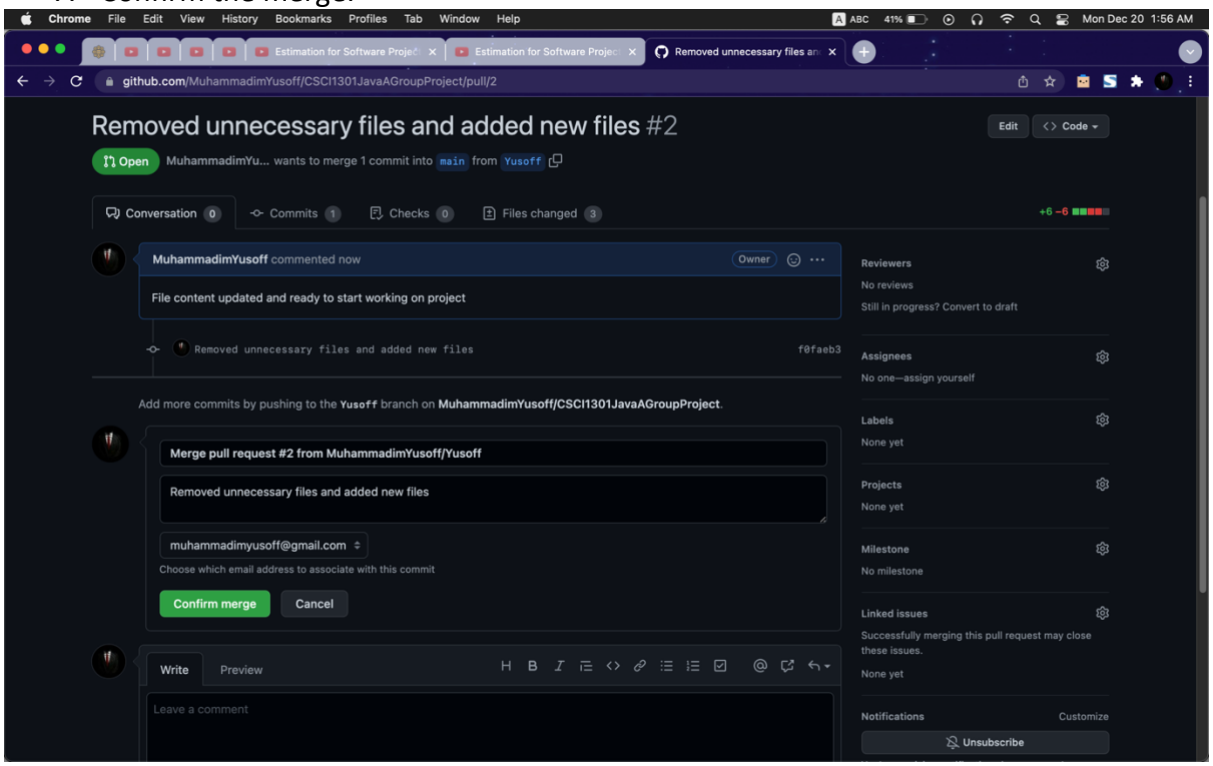
3. Check on the ‘Open a pull request’ content and make sure it’s ‘Able to merge’. This indicates no conflict for the ‘merge’ process.
4. Make sure you have some comments for easy tracking in the future.
5. Click on the ‘Create pull request’ button to approve the changes.



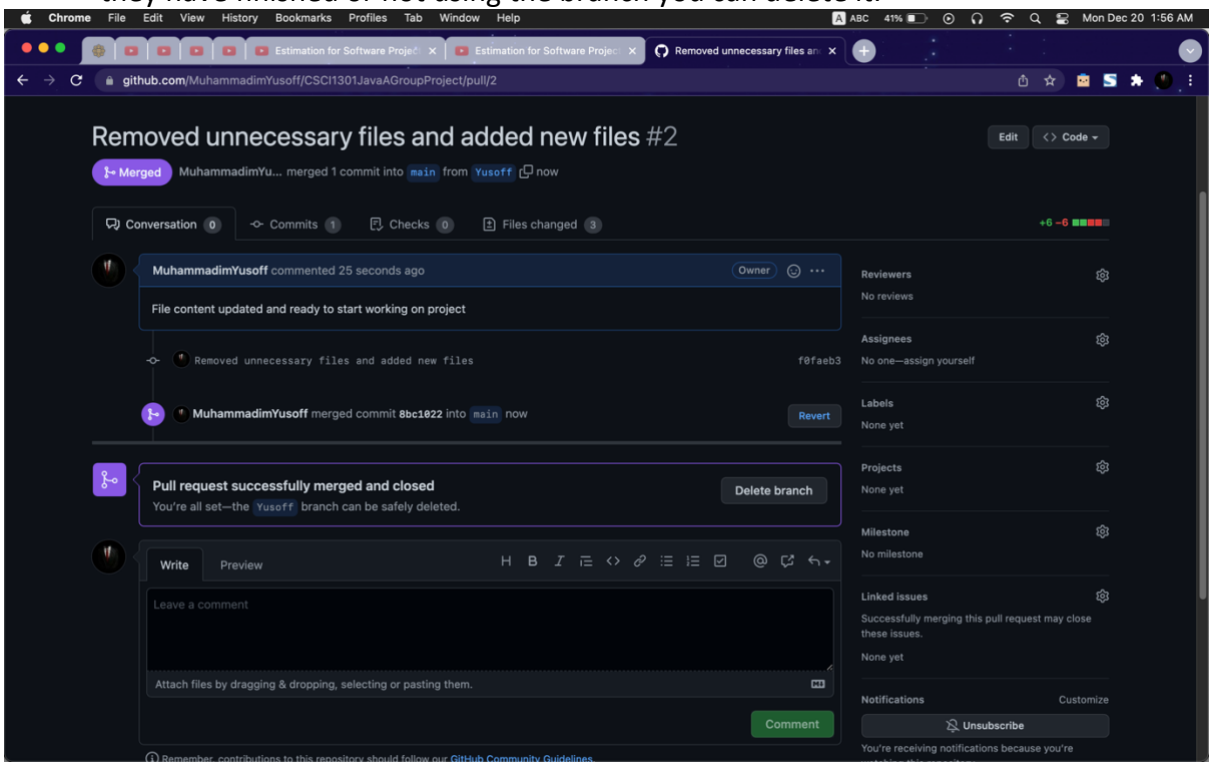
6. You will again be asked to approve for the ‘merge pull request’, just click on the button to approve it.



7. Confirm the merge.



8. If the merge successful, you will get the purple outline, ‘Pull request successfully merged and closed’ and ‘Delete’ branch button
9. You might not want to delete the branch as your collaborators still working on it, if they have finished or not using the branch you can delete it.



10. You can recheck the pushed file from the other branches into the main branch by checking the repo content itself.