**PUNJAB UNIVERSITY OF INFORMATION AND TECHNOLOGY**



**Submitted to: Prof. Haq Nawaz**

**Submitted by: Muhammad Tayyab**

**Roll no.: mcsf19a037**

**Mobile Computing Classwork**

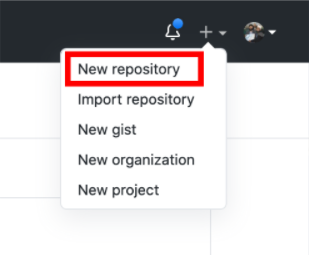
**Lecture 1**

**Task 1:** Creating GitHub Repository, simple text file and uploading it on GitHub.

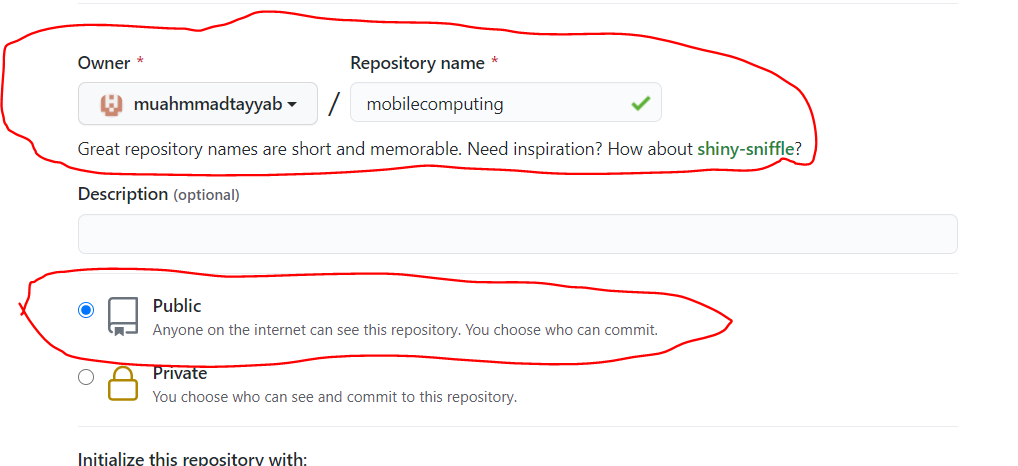
**Steps: -**

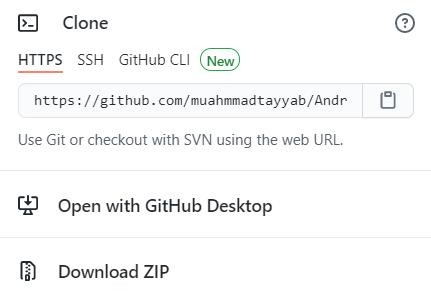
**Step 1:** Create an Account on github.com.

**Step 2:** Create a Repository on GitHub.

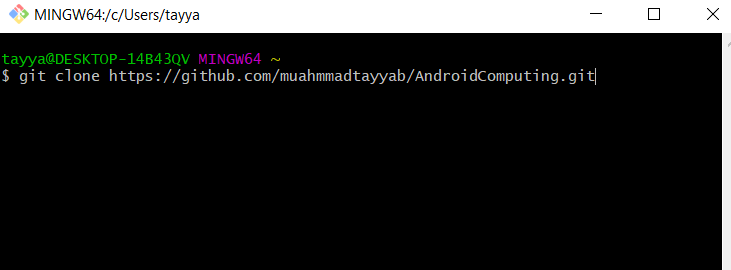


**Step 3**: Provide details of repository on your account.

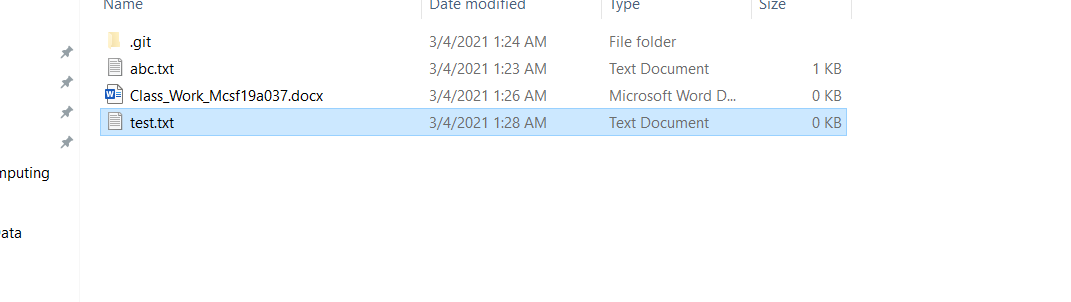


**Step 4**: URL of repository on your account.

**Step 5:** Clone a repository



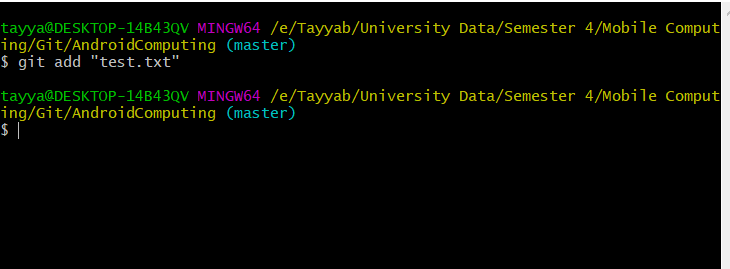
**Step 6:** Create simple text file by clicking right click and then click on New and then Text Document.

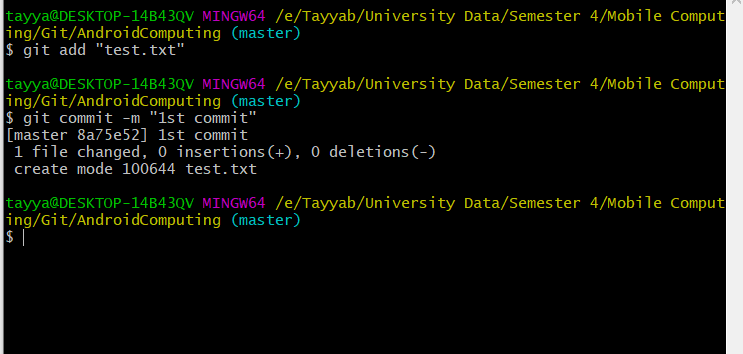


**Step 7:** Now open git from start menu by searching git.

****

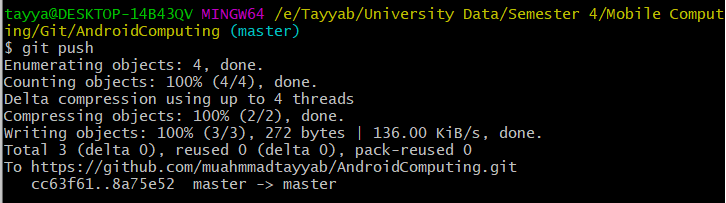
**Step 8:** Now Add text file in git by writing command “git add test.txt”.



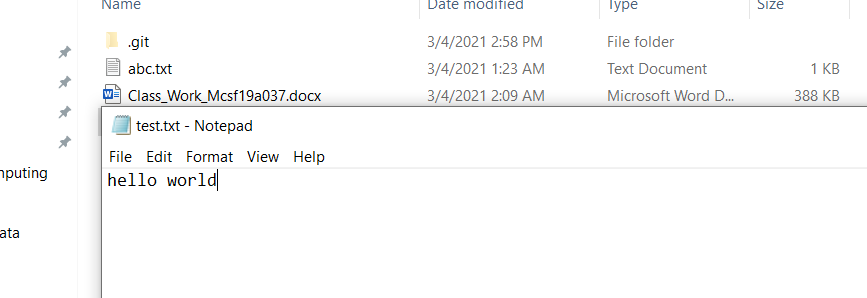


**Step 9:** Changes made with message to track.

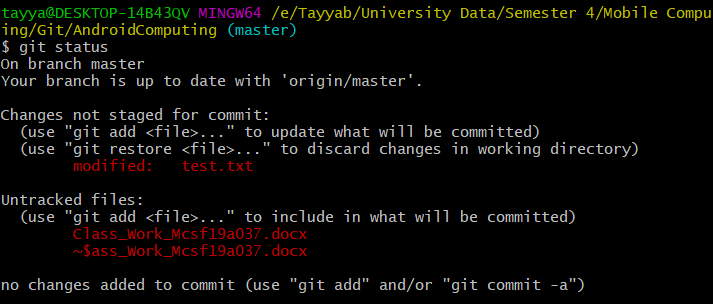
**Step 10**: Now push changes to server.



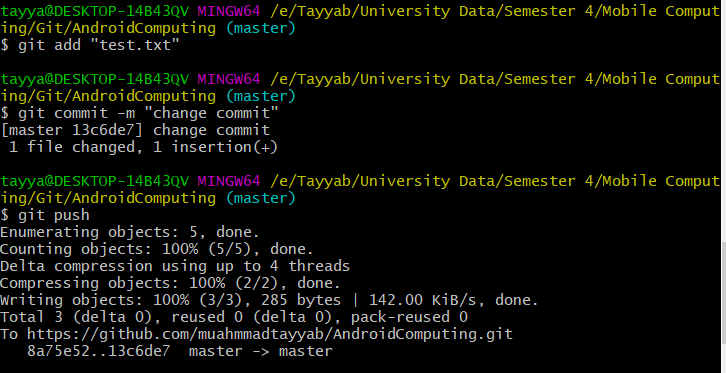
**Step 11**: Now made some changing in text file.



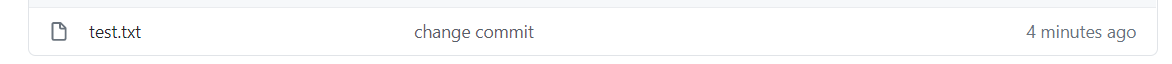
**Step 12**: Now check status on git by writing command “git status”.

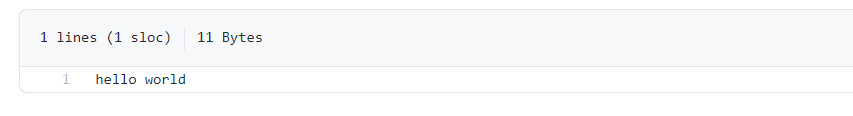


**Step 13**: Now again add file to git and commit change made with message to track and then push it to server git.



**Step 14**: view of repository after push by opening git website on browser.



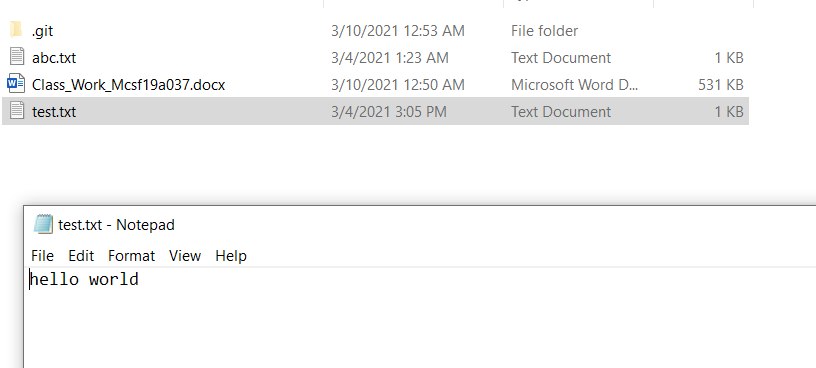
**Step 15**: Content of file in repository.

**Lecture 2**

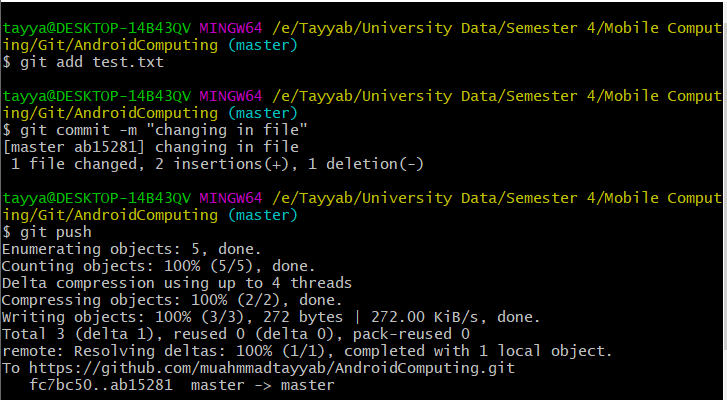
**Task 2:** To solve merge conflict.

**Steps: -**

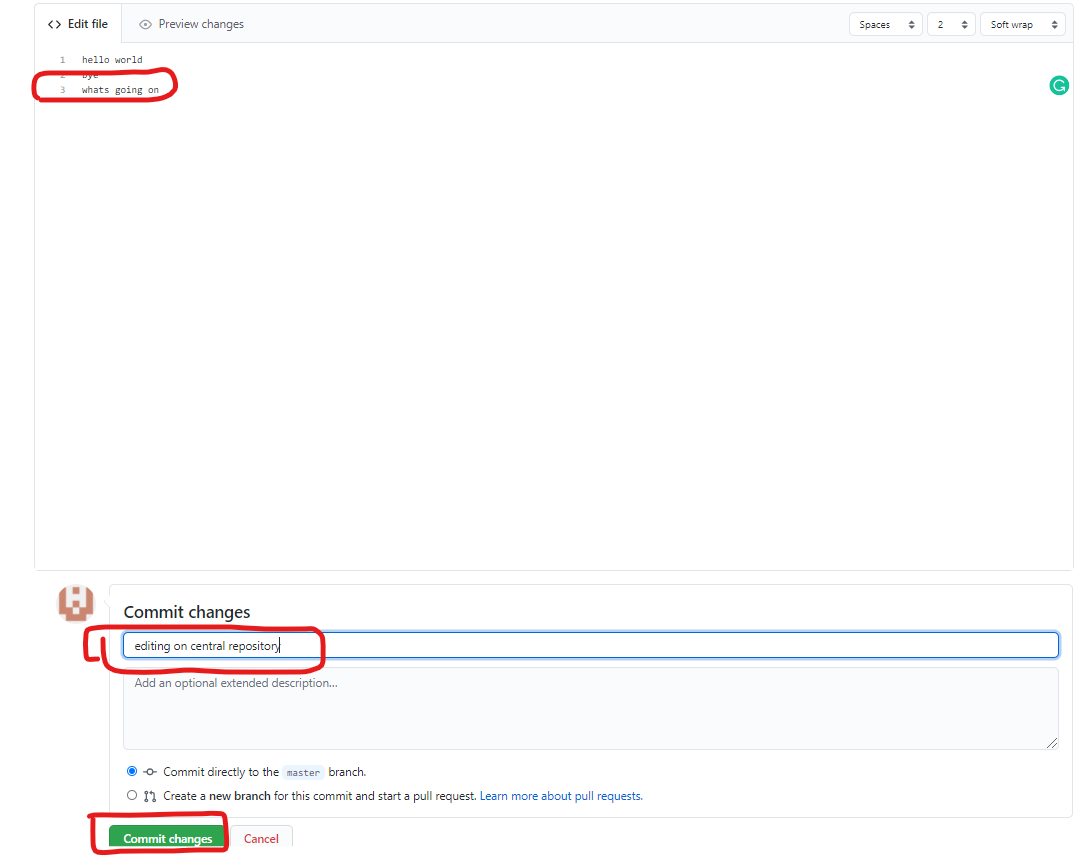
**Step 1:** Open text file.



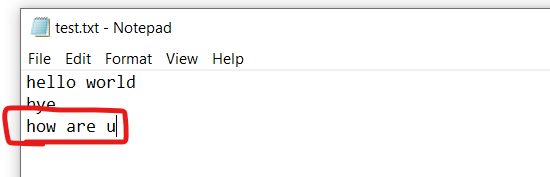
**Step 2:** Add some text. After that, add this file to git, then run commit and then push it into central repository.



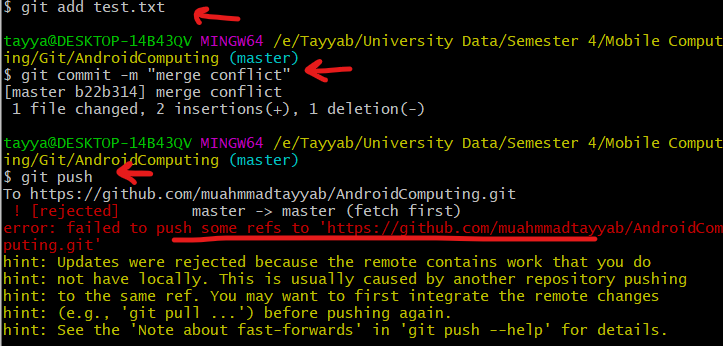
**Step 3:** Now, to merge central and local repository, first, edit code into central repository and then commitchanges.



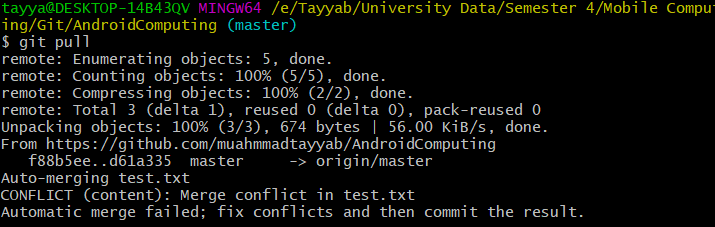
**Step 4:** Now, without pulling central data, just go into local repository and add some text to same line at the same time.



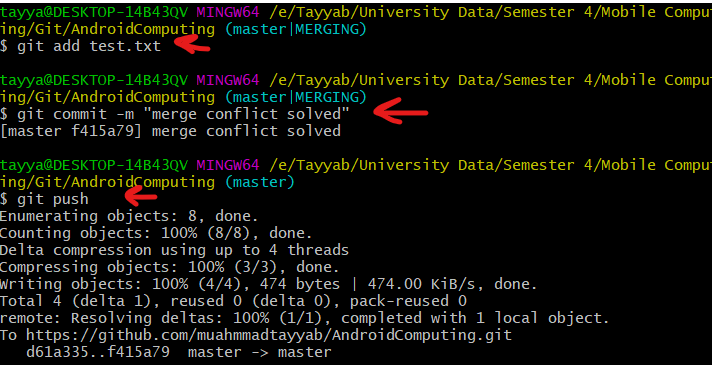
**Step 5:** Then add that file into git and commit, now, when you push it, it gives an error to firstly pull central repository data into it.

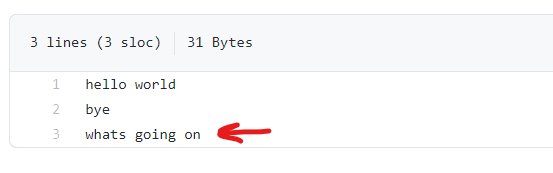


**Step 6:** So, run git pull command and it gives an error which tells you that there is a conflict in merging.

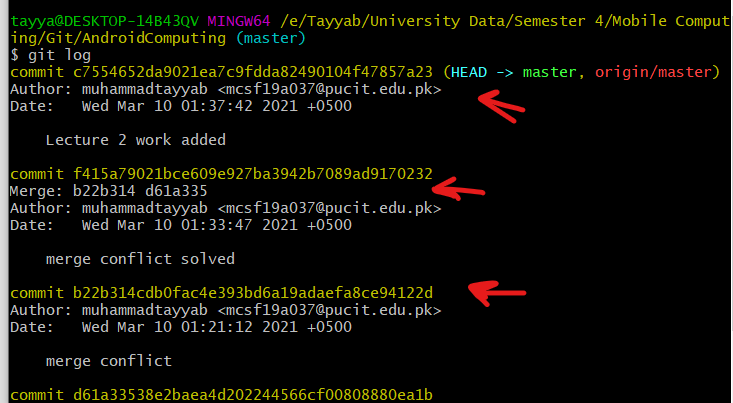


**Step 7:** So, open your file from local repository, you can see there are two statements showing (one of local and 1 of central repository). Select the data you want to add and del other one, then save. After Selection error resolve.



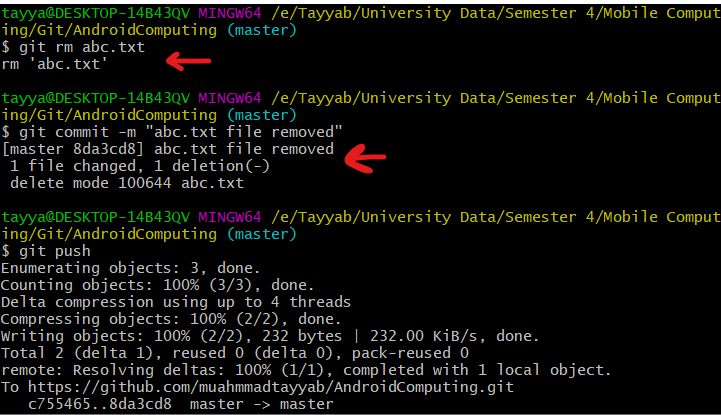
**Step 8:** so, now, the new changes can be seen on repository.

**Task 3**: Check History of command and action you perform using git log command.

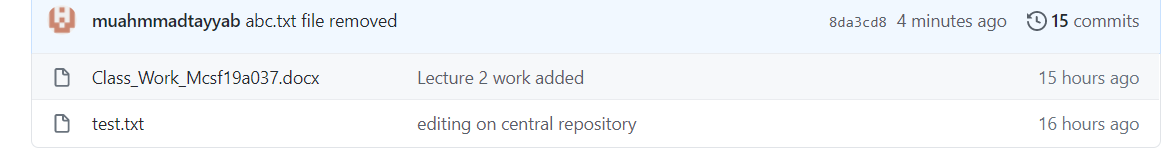


**Task 4**: Delete file using rm command.

git rm <filename>

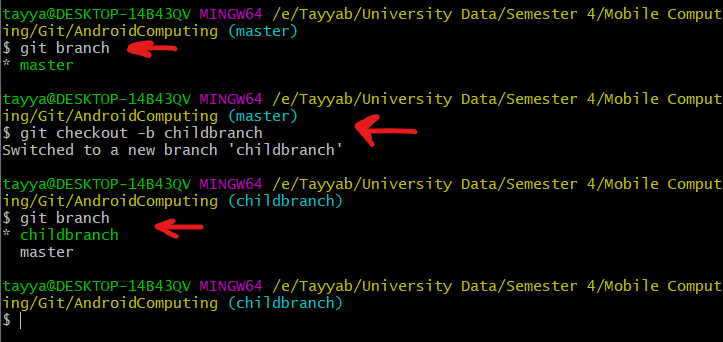


Here you can see file deleted successfully.

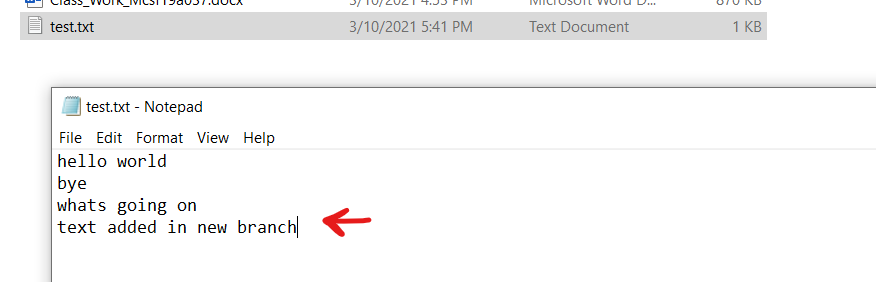


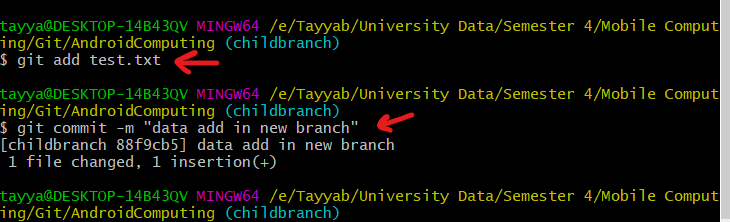
**Task 5:** Creating Branches, Merging Branches

**Step 1:** At first, check your branch status which is master branch by default, then create a new branch using git checkout –b newbranch command. Then again check your branch status**,** it will be changed.

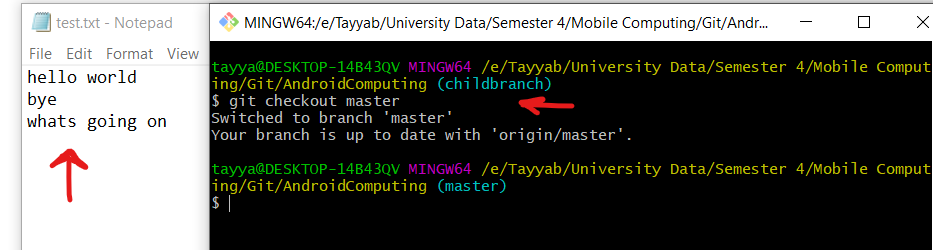


**Step 2:** Now apply some changes in text file. Then add file to git, commit it and push

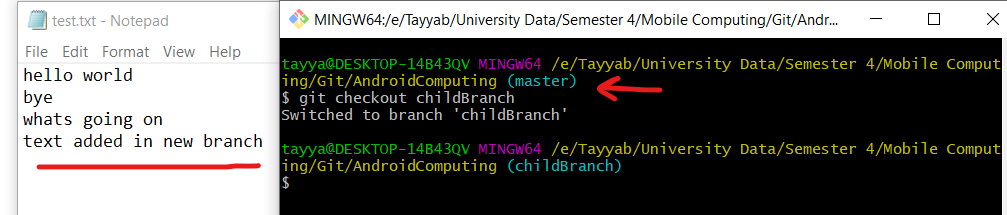




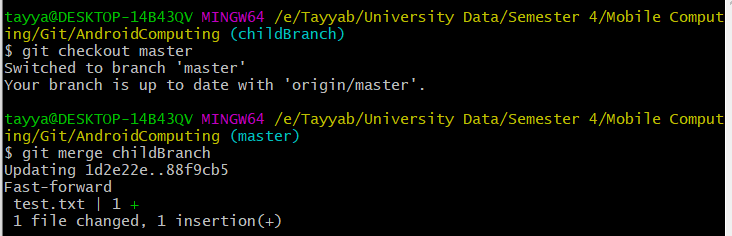
**Step 3:** Now change branch to master branch using command git checkout master. After changing here, you can see data in master branch.



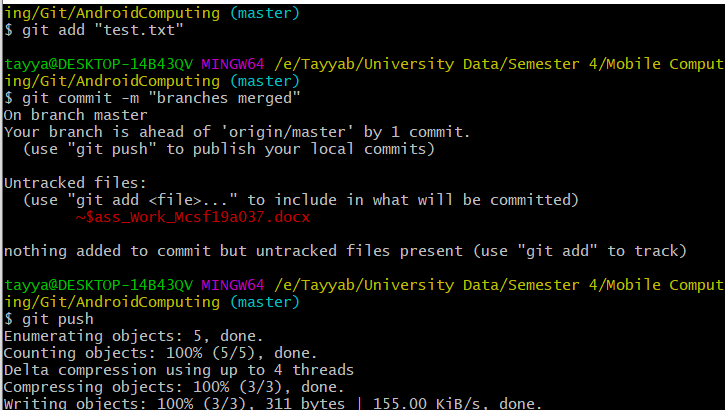
**Step 4:** Now again change branch to childBranch and you will see childBranch data.



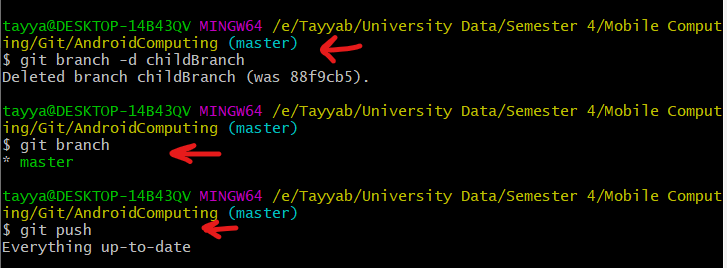
**Step 5:** Now to merge both branches switch to that branch first in which you want to merge the data and then merge data using git merge [branch\_name] command.



**Step 6:** After merging add file to git, commit it and then push it to git central repository to update.



**Step 7:** After pushing delete childBranch with command git branch -d childbranch, and check branch status by git branch command, and then push.



So, that’s all the git practice.

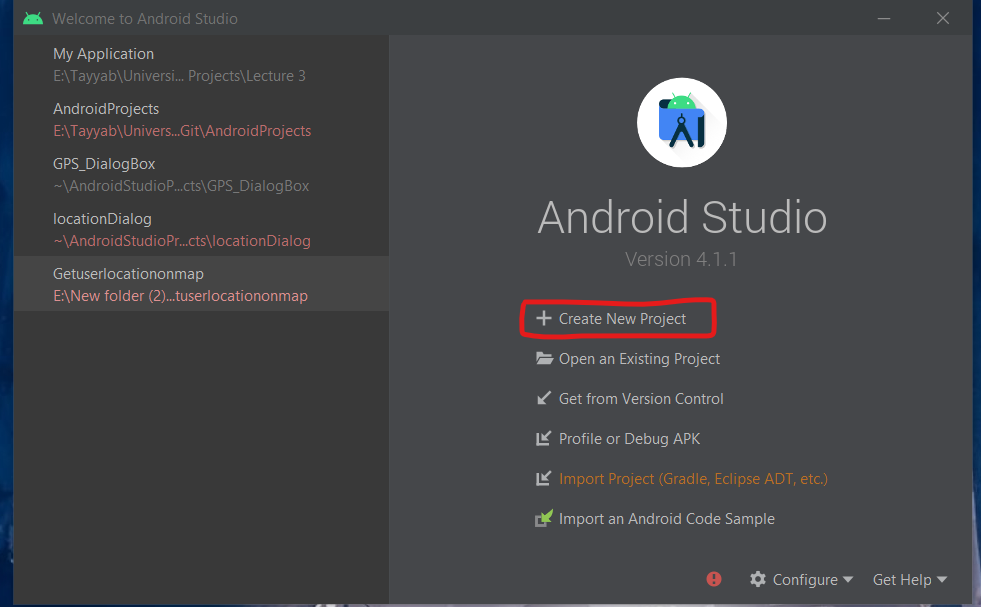
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Lecture 3**

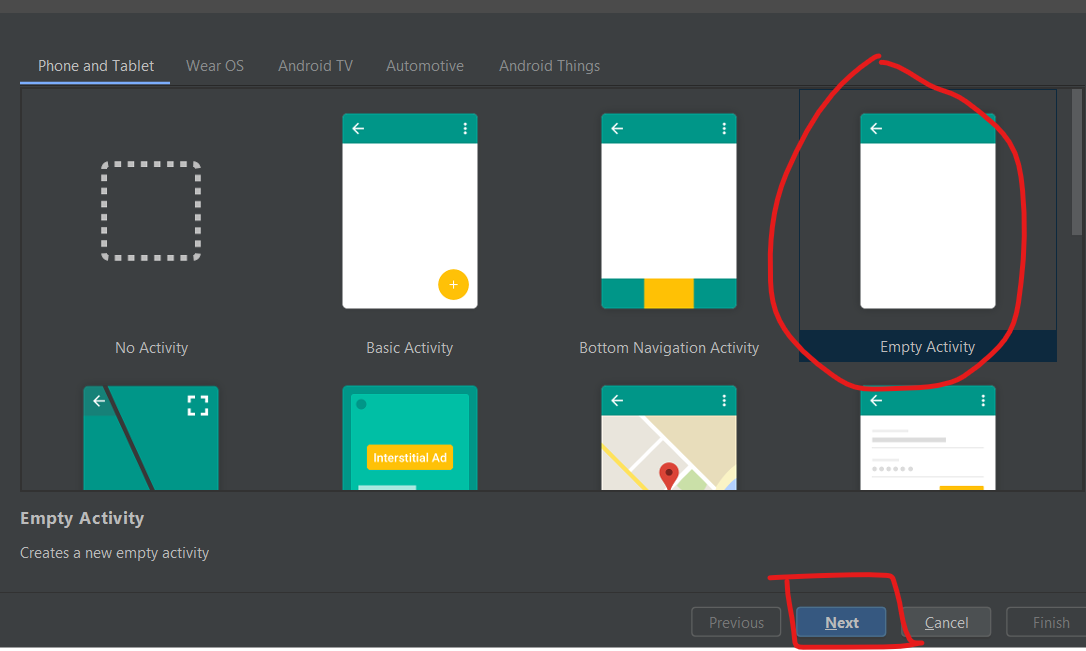
Task 1: Create Basic three screen Android Application.

Steps 1: Open Android Studio from Star Menu.

Step 2: Now click on Create New Project.

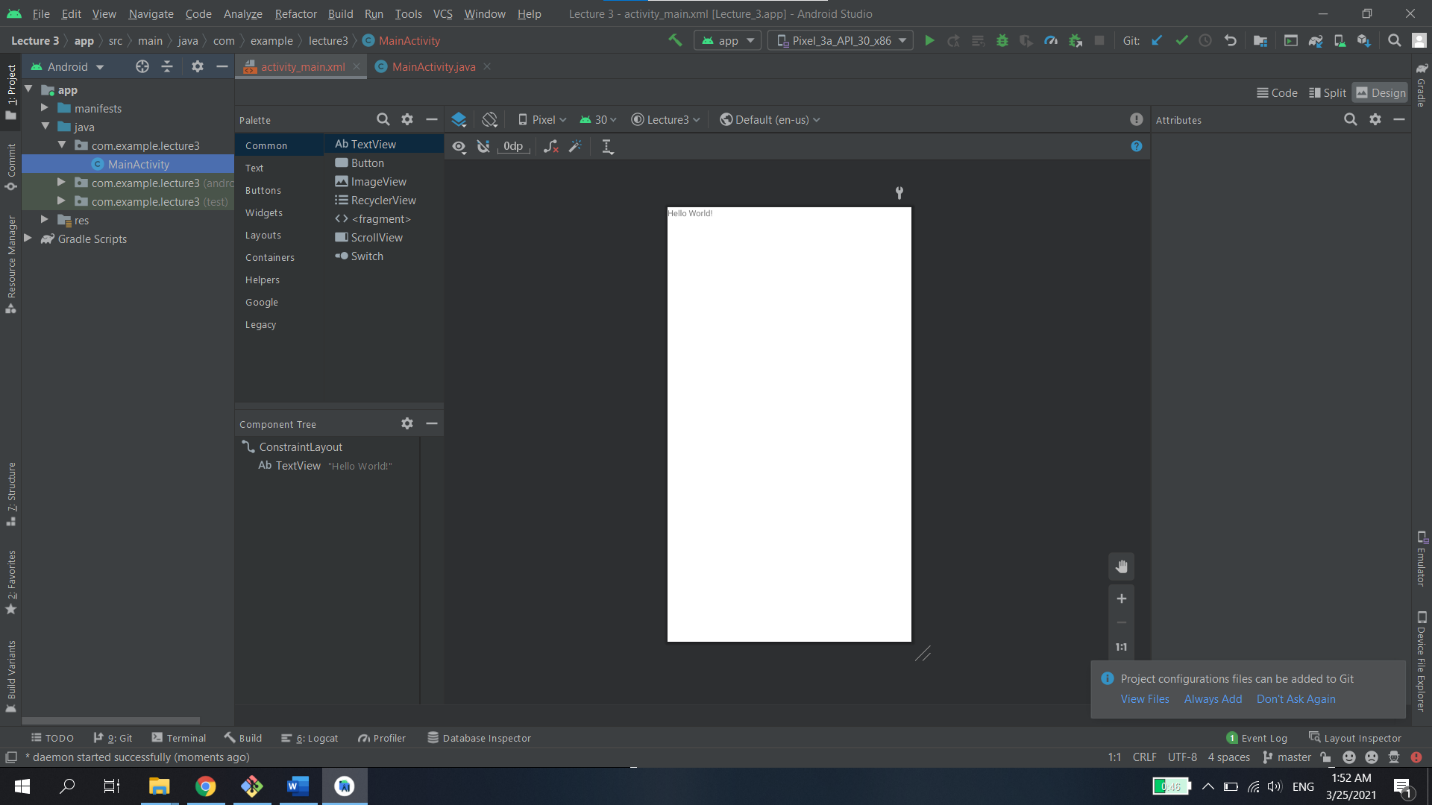


Step 3: Then select Empty Activity and thin click on Next.



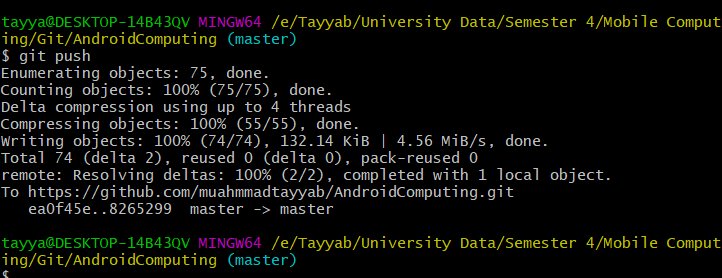
Step 4: Then give an appropriate name to app and then click on Finish.

Project Created Successfully.

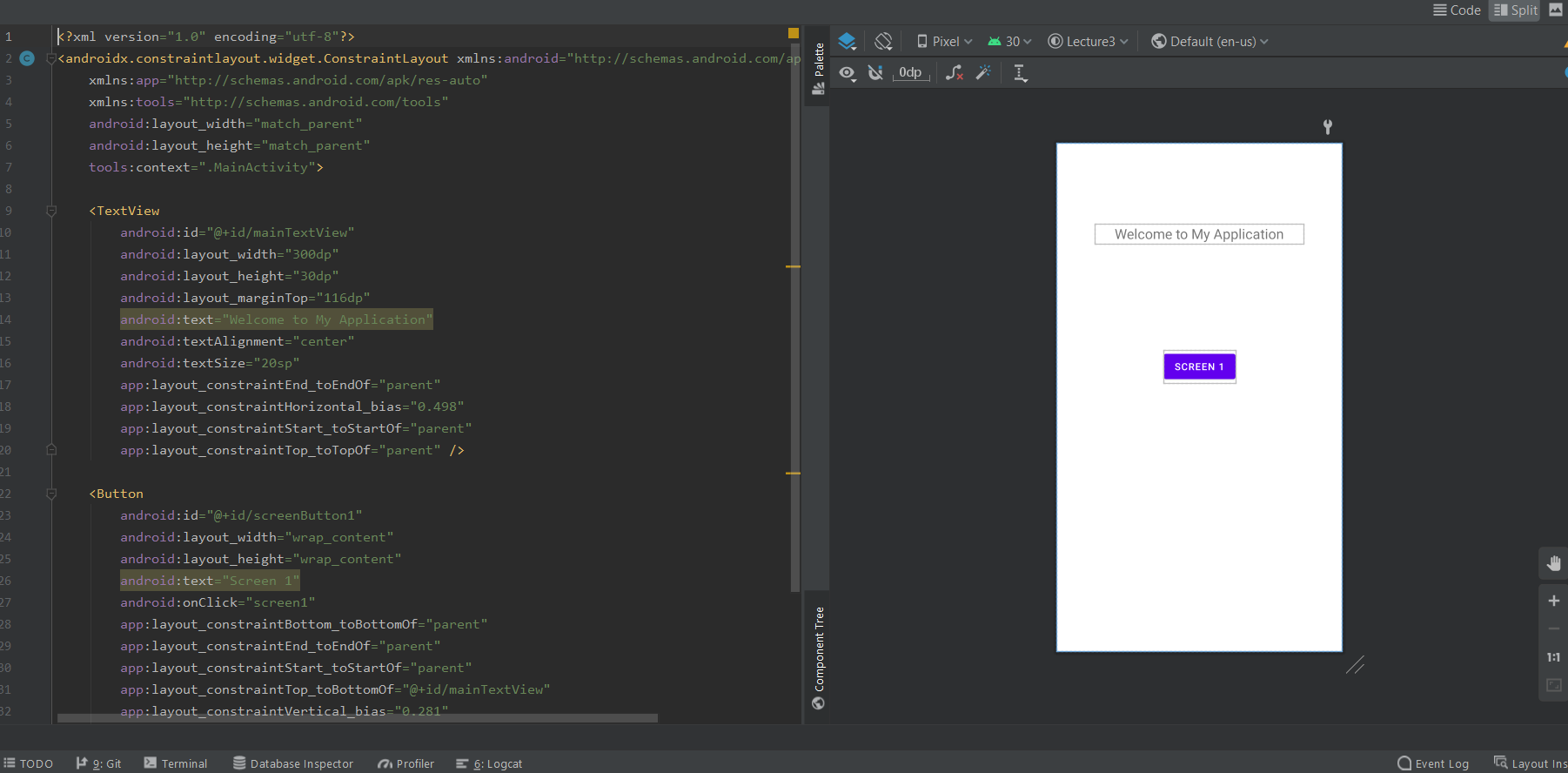


Step 5: Now push it to repository ass you know how to push on repository.

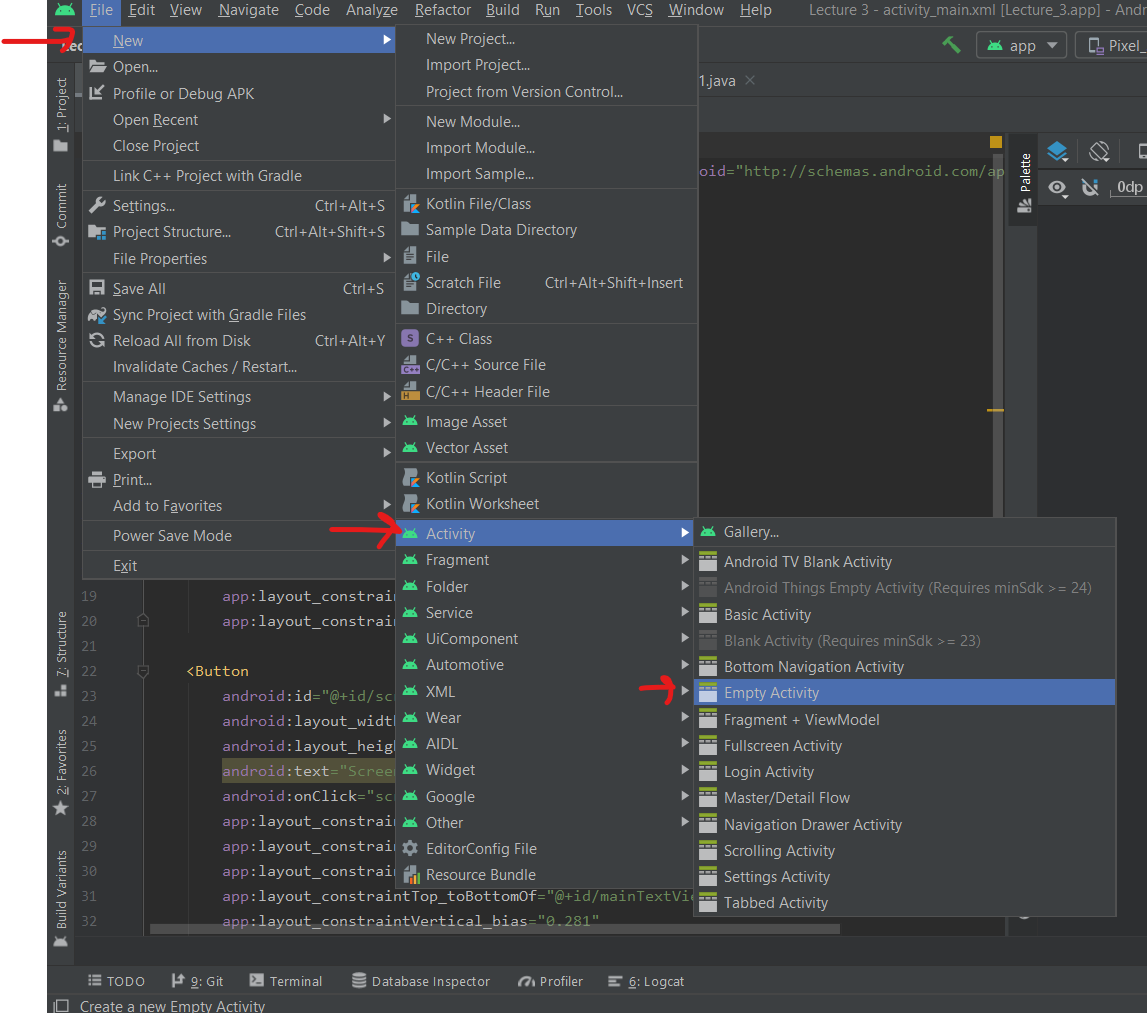
Project Uploaded on Repository successfully.



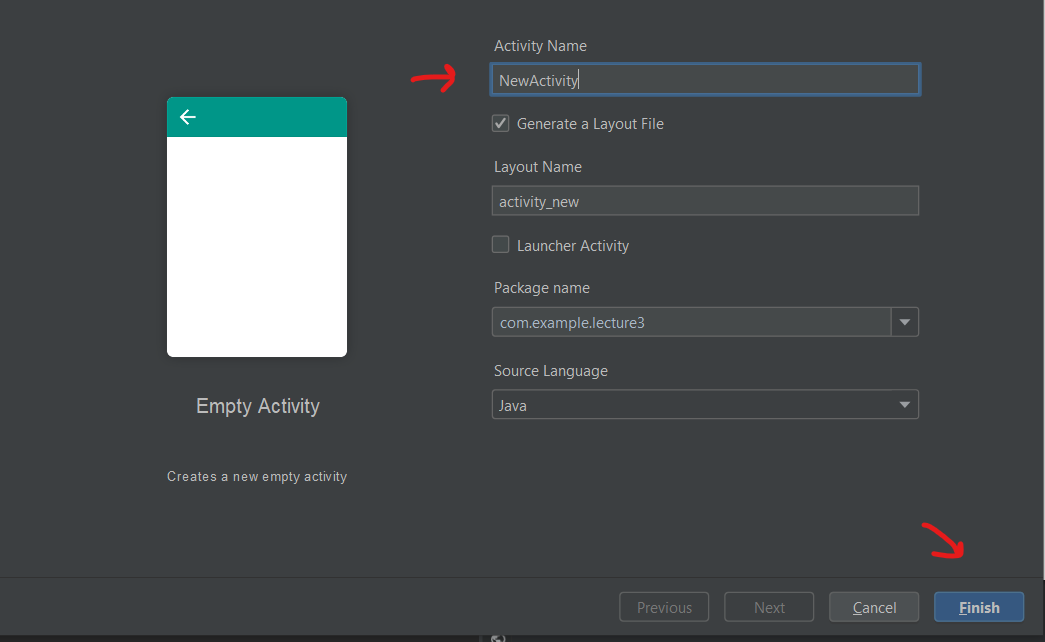
Step 6: Now add one TextView and Button on main screen by add and drop.



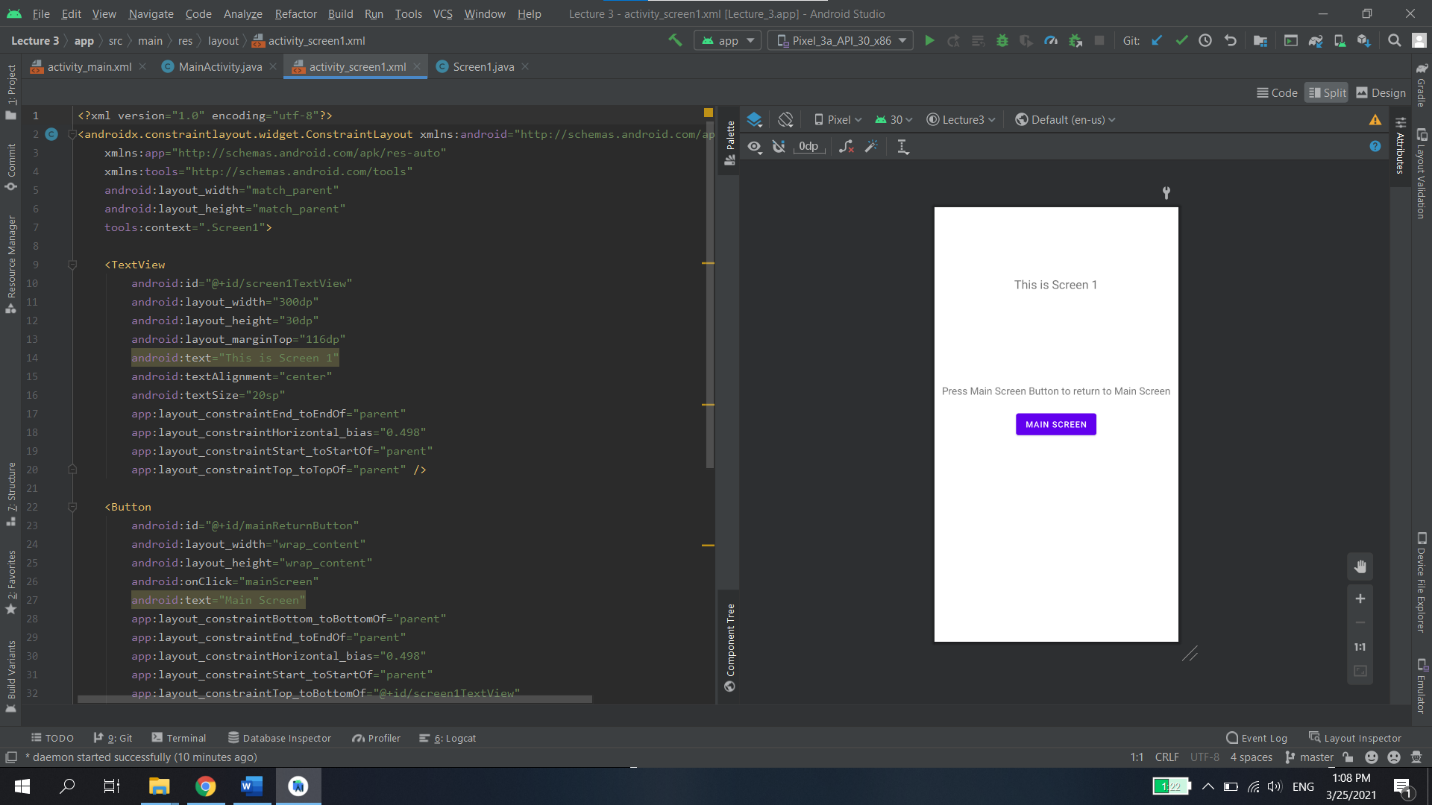
Step 7: Now add one new activity by clicking on File then New then Activity and then click on empty activity.



Step 8: Then give name to Activity and click on Finish. New activity will be created.

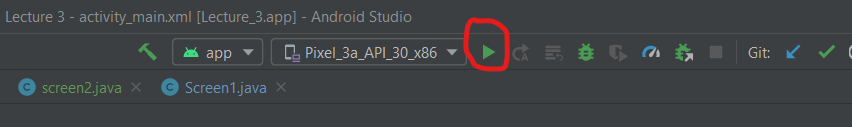


Step 9: Now add one TextView and Button on main screen by add and drop and provide information.

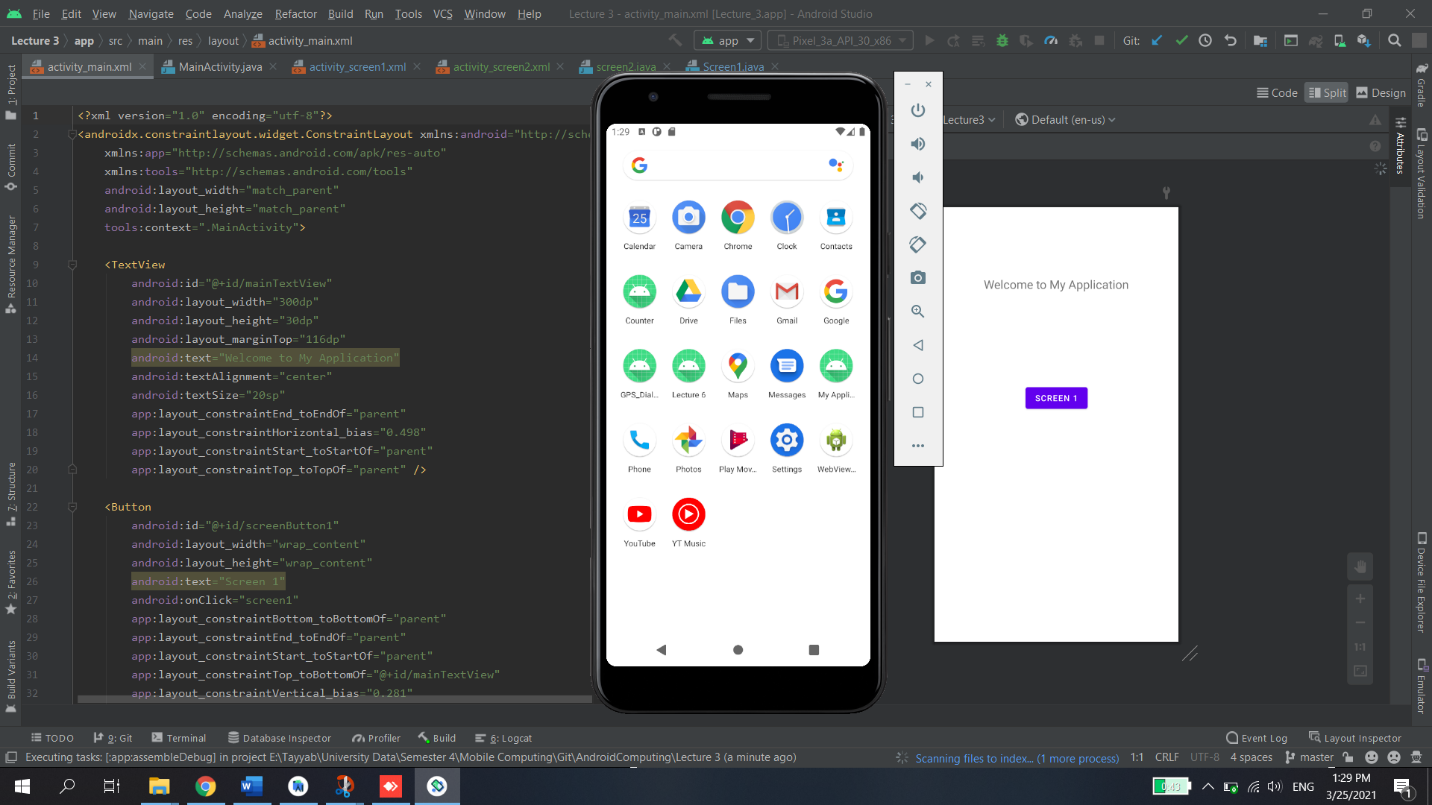


Here you all done it.

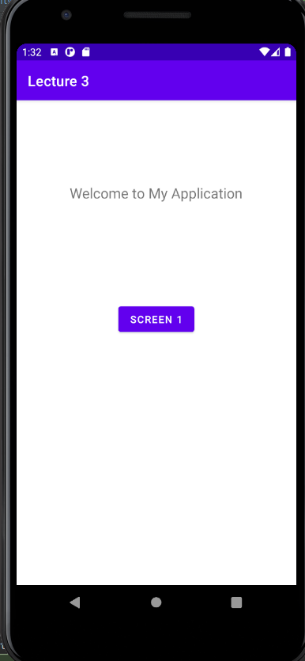
Step 10: Now run app by clicking on Run button or pressing Shift+F10.



Here some background process will start working and in some time app will run on emulator. Here is picture of emulator.

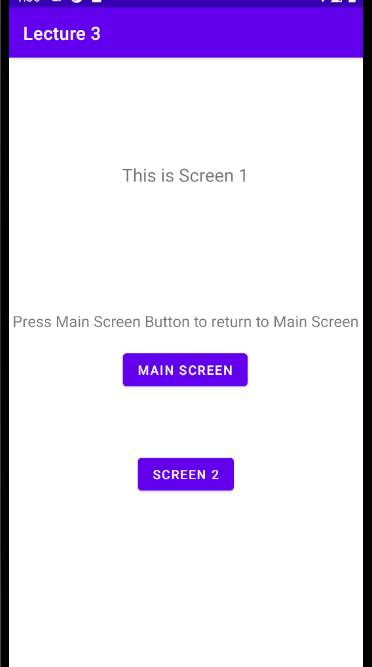


Here application is running and picture of main screen.

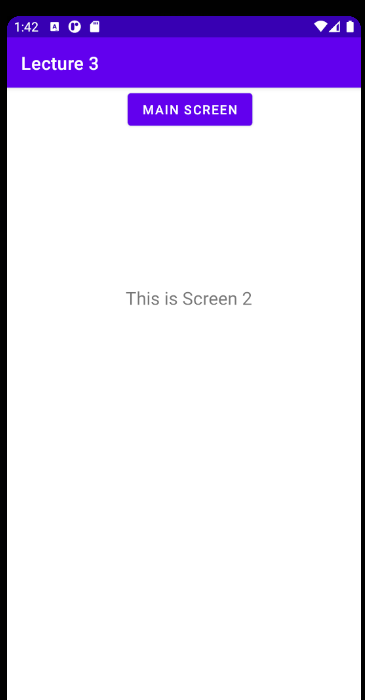


Step 11: Now click on Screen 1 Button to move to screen 1.

Here is screen 1.



Step 12: Now click on Screen 2 Button to move to screen 2.



Push it on Giit as you know how to do it. Here you done all. Task is completed.