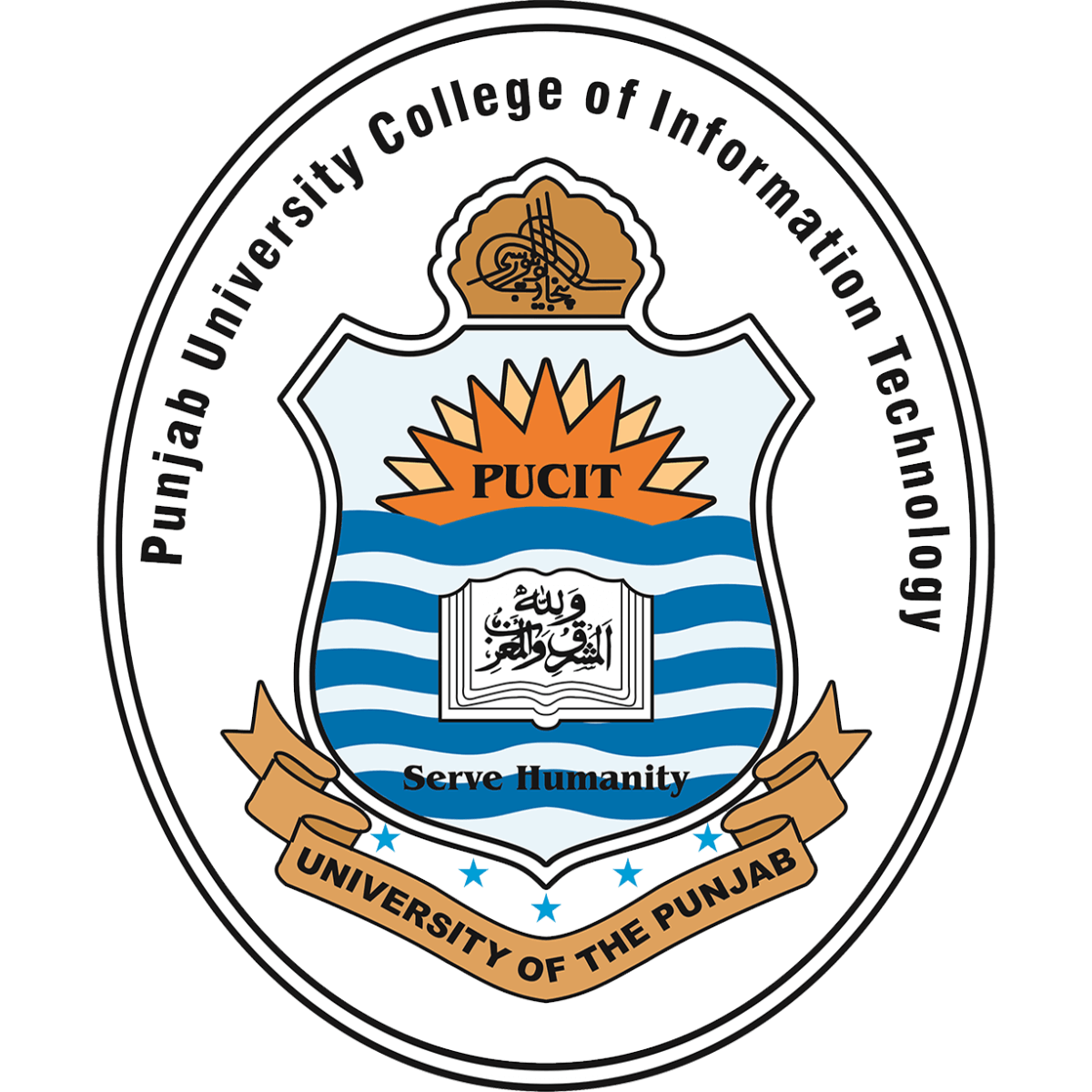
**PUNJAB UNIVERSITY COLLEGE OF INFORMATION AND TECHNOLOGY**

****

**Submitted to: Haq Nawaz**

**Submitted by: Sami Ul Haq**

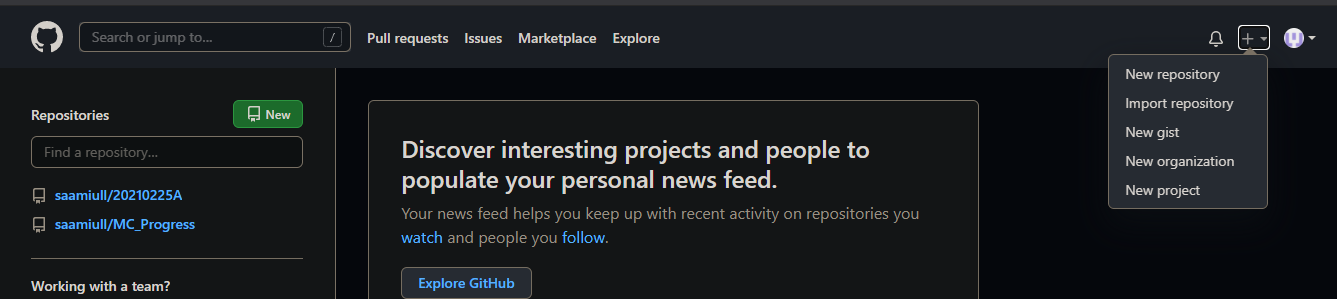
**Roll No: BSEF18A028**

**PUNJAB UNIVERSITY COLLEGE OF INFORMATION TECHNOLOGY**

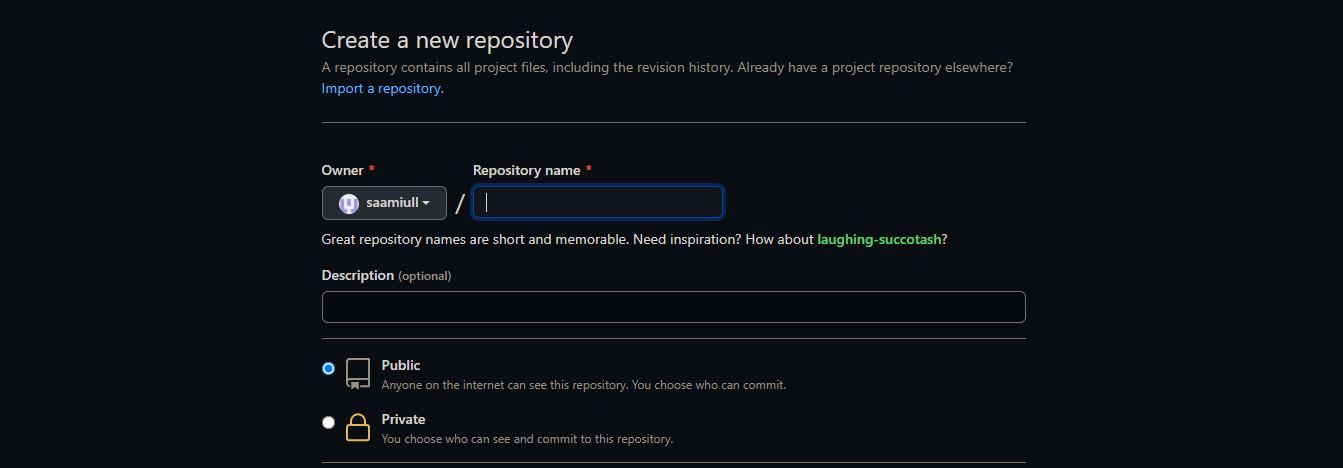
**Lecture – 02**

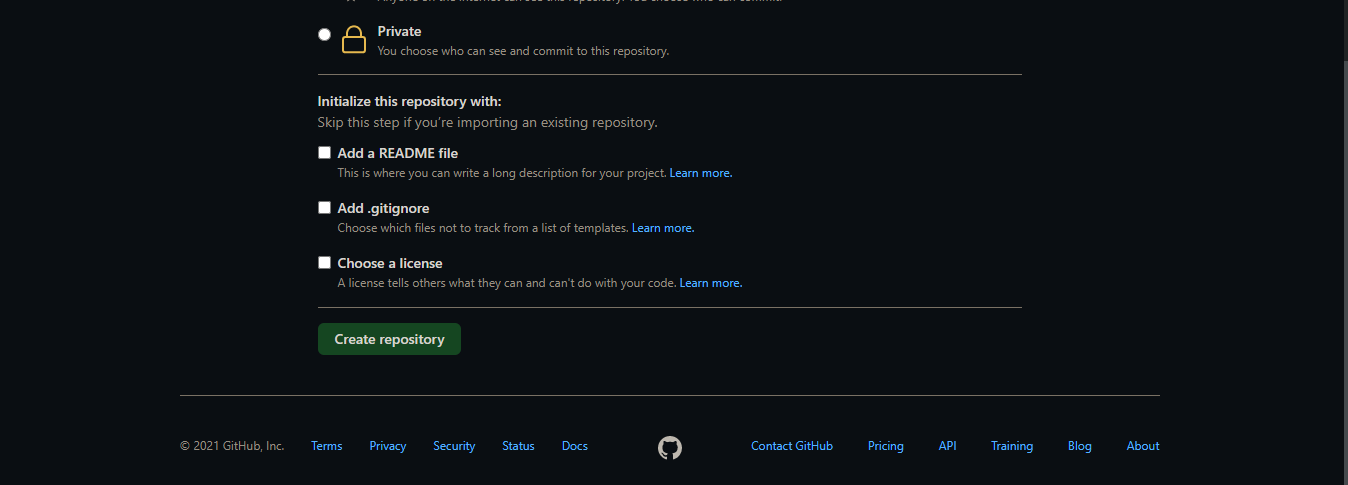
**Vision Control System**

* **Make a new Repository and upload any file on repository**
* **Step 01:** On the top right corner you see a plus drop-down button click and select a new repository.

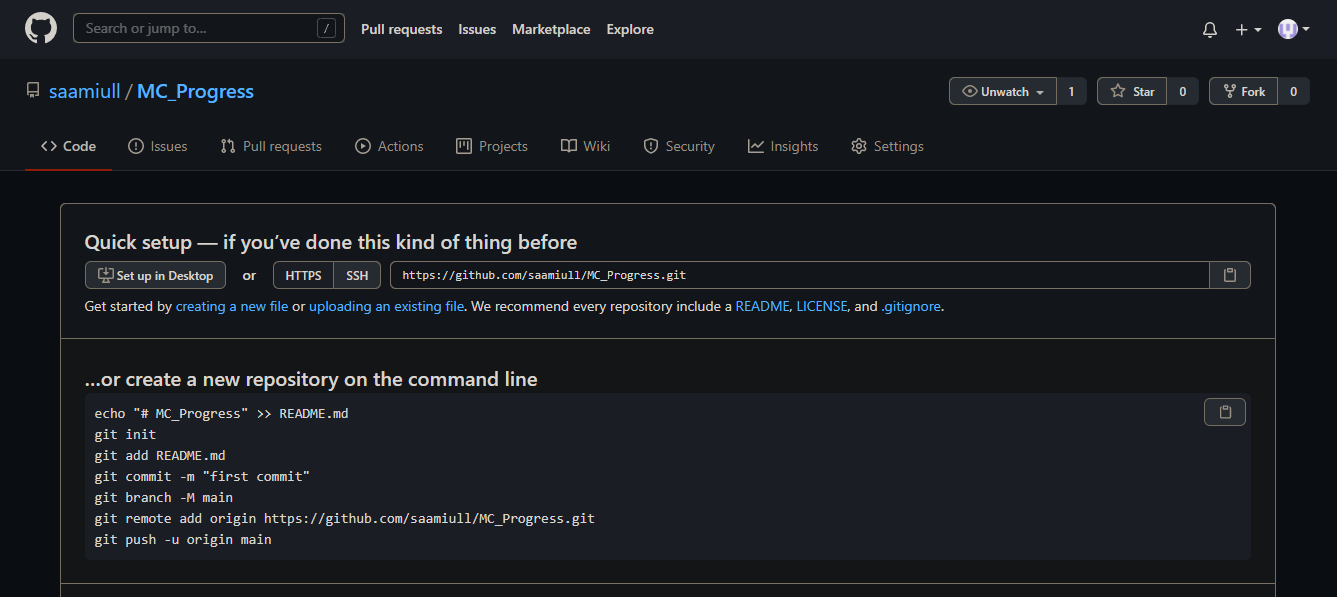


* **Step 02:** Write the of your repository and go down and select Create repository.

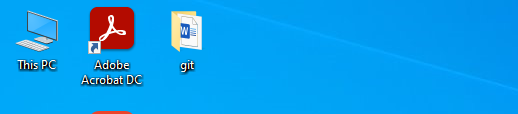




* **Step 03:** New repository is created. My repository is shown below:



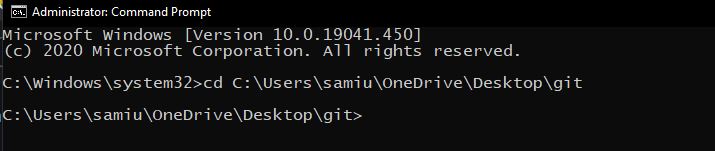
* **Add any file on Git hub**
* **Step 01:** Make any folder in any directory. In my case I make a folder git on Desktop.



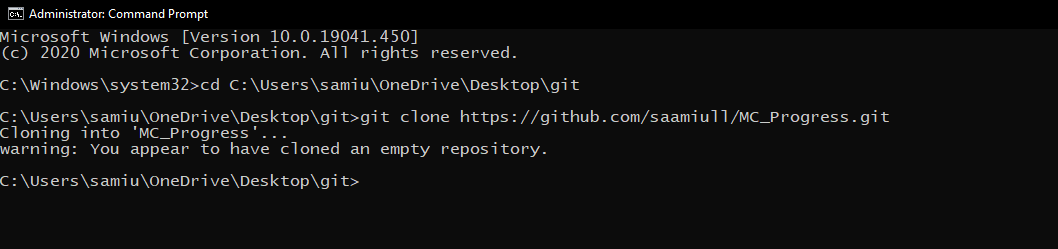
* **Step 02:** Make a clone of this repository. For this we use command prompt. Open the command prompt and run the following command.

git clone [URL]

firstly, set the path of folder which create last time. In my case I set the path of git folder.

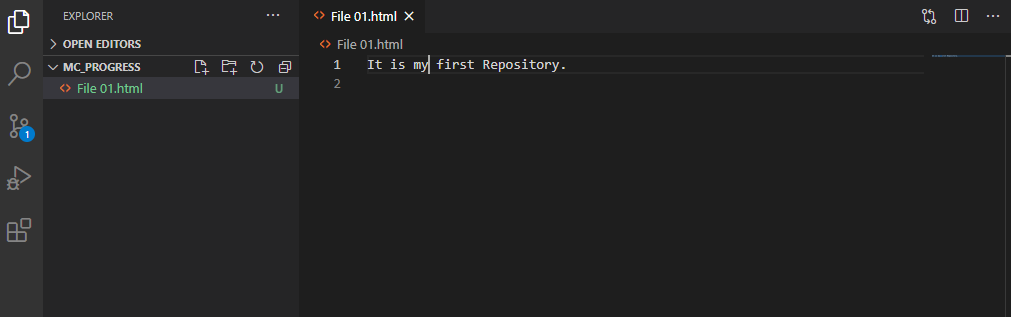


URL in this case is the URL of Repository which we can create before this.



So, empty repository clone is created because repository is empty.

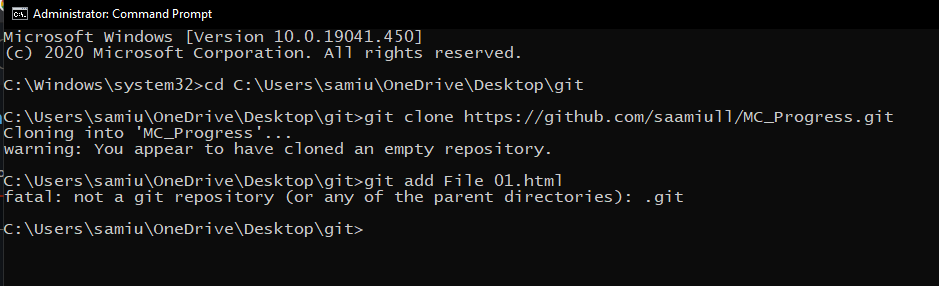
* **Step 03:** Make any file in clone folder. In my case a make a html file which name is File 01. And write something in it.



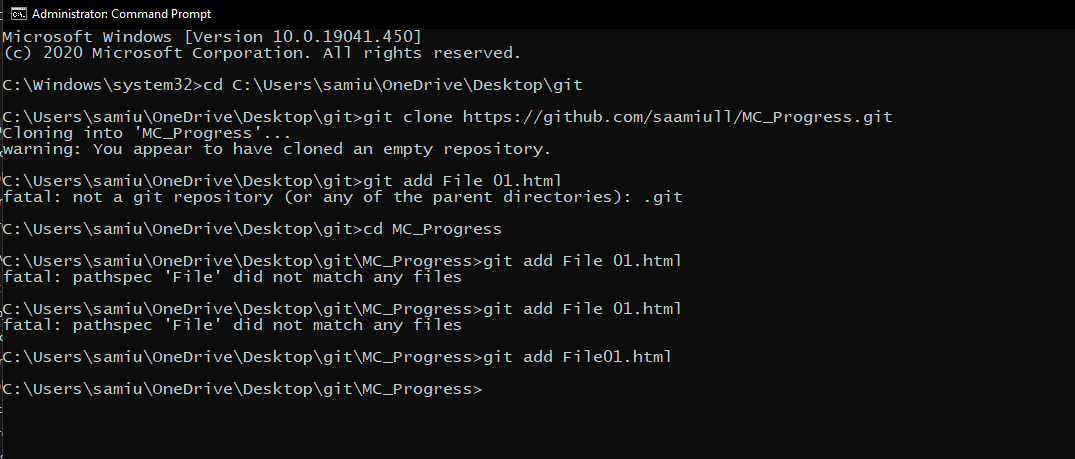
* **Step 04:** Add file by use of following command.

git add filename

in my case it is git add File01.html



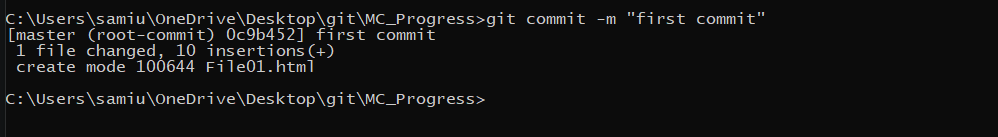
It displays an error because we are not in such directory in which file is save so first move to this directory. And again, apply this command.



* **Step 05:** Commit this file by use of following command.

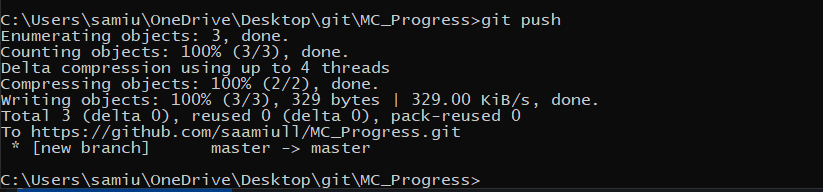
git commit -m "Message"

in my case it is git commit -m "first commit".

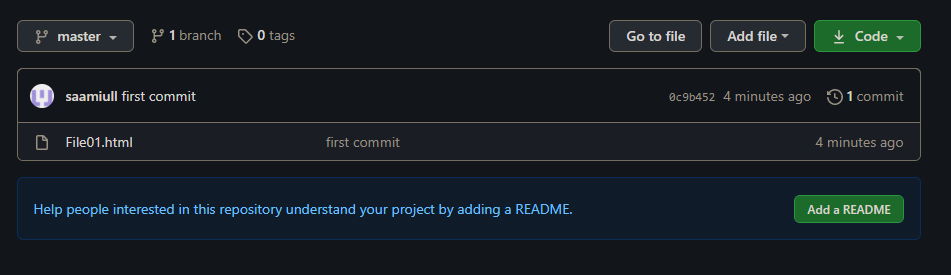


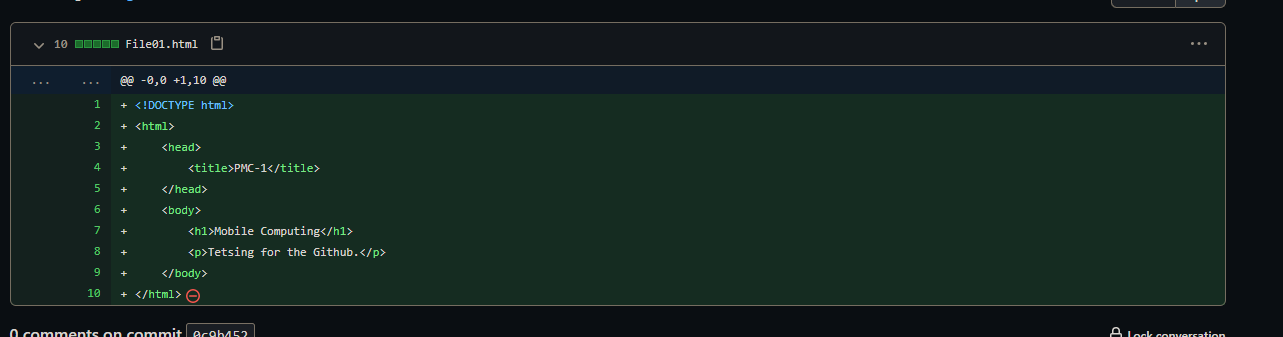
* **Step 06:** Push this file by use of following command.

git push

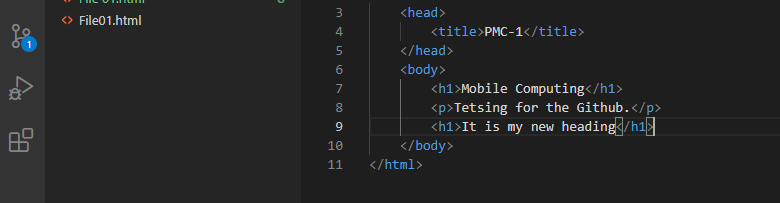


* **Step 07:** Refresh the page of browser and we will see the file in our repository.





* **Step 08:** Add new heading in file and see the changes.

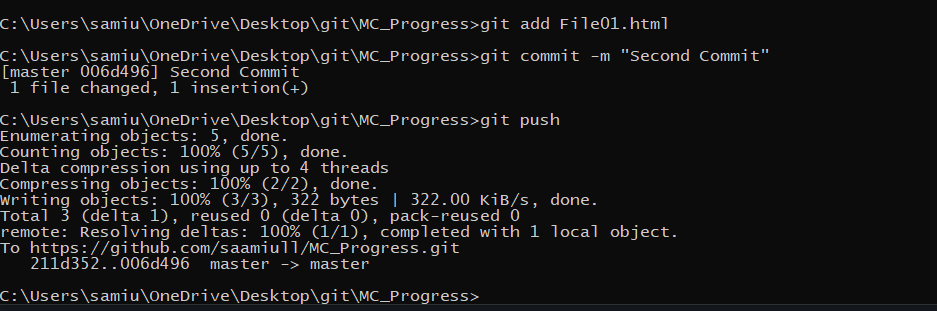


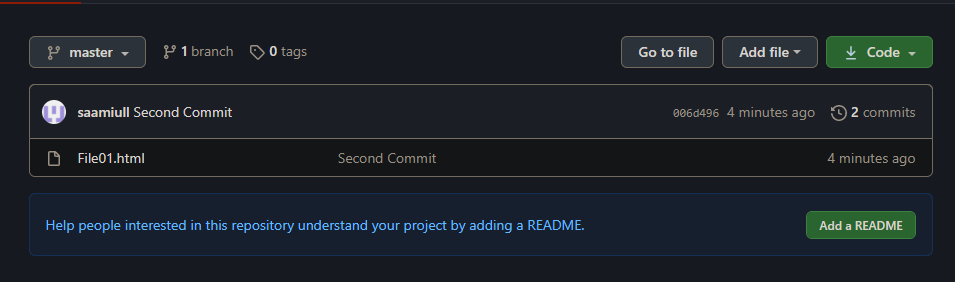
* **Step 09:** Again, add this file, commit and push.

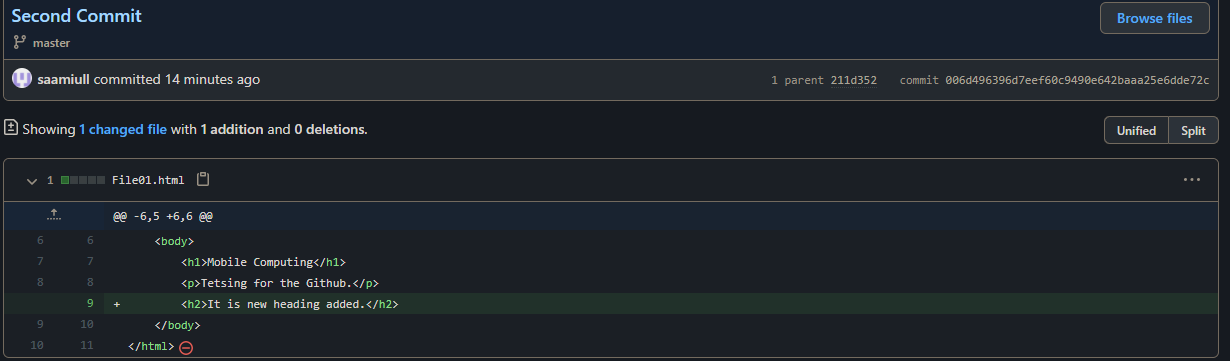
git add File01.html

git commit -m "second commit"

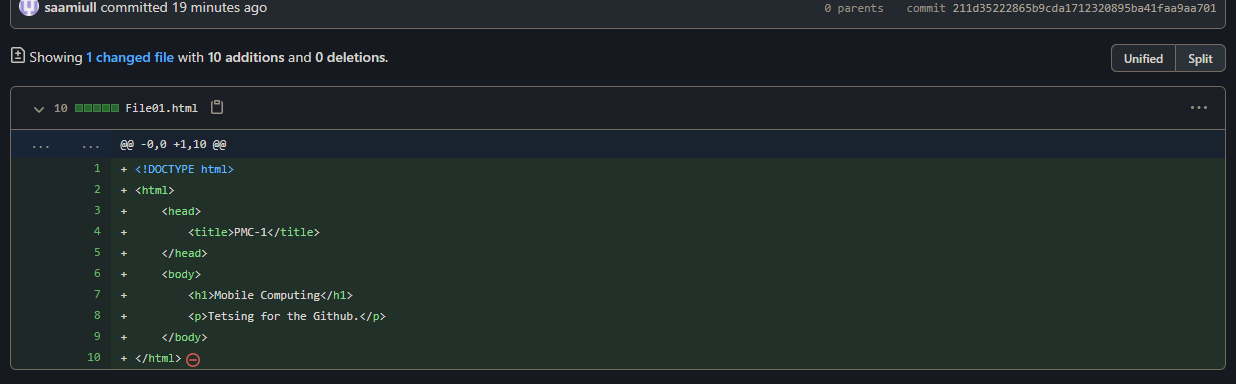
git push





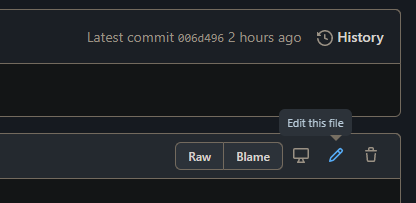


* **Step 10:** By Clicking on parent, you see the previous code.

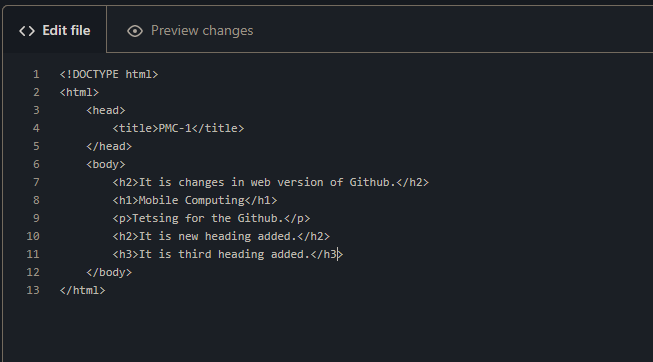


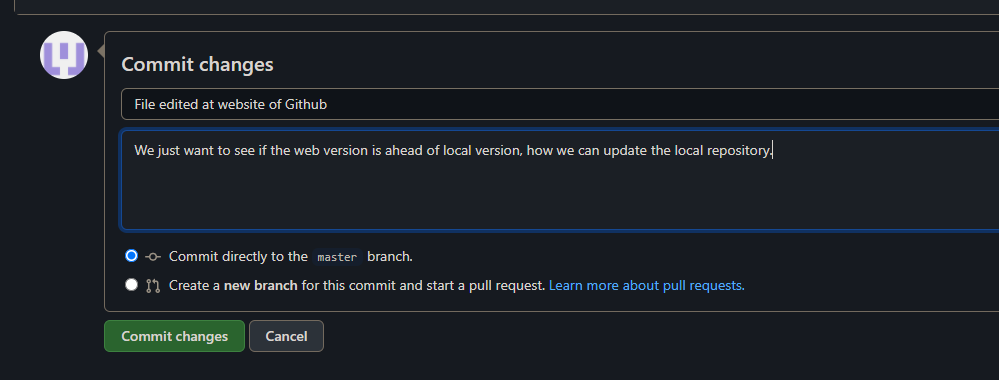
**Lecture – 03**

* **Edit files at Repository**
* **Step 01:** On the top right corner you see Edit button click on it.



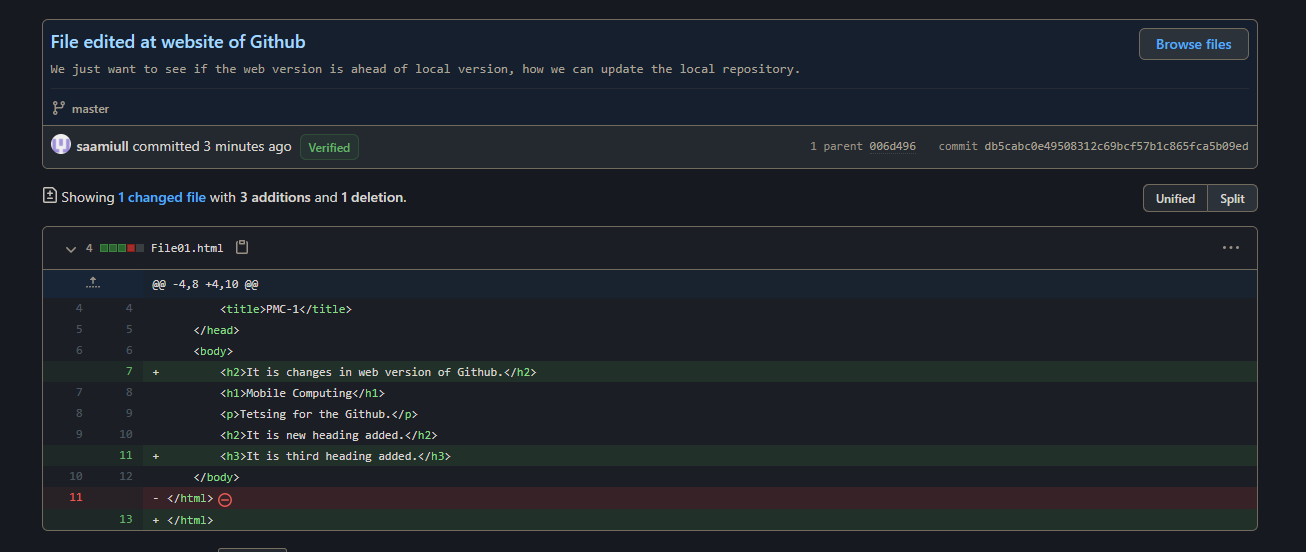
* **Step 02:** Add Changes in file and Commit it. And if you want to add description then add it.





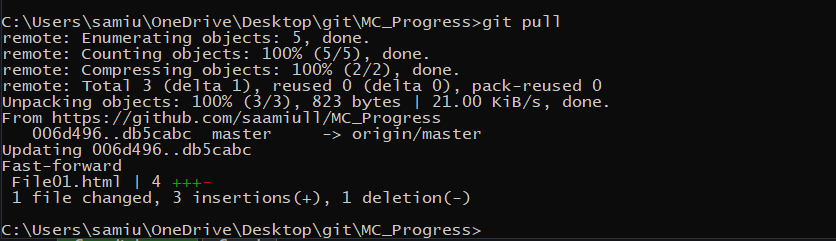
* **Step 03:** Changes added in file and you see your new Commit and its description.



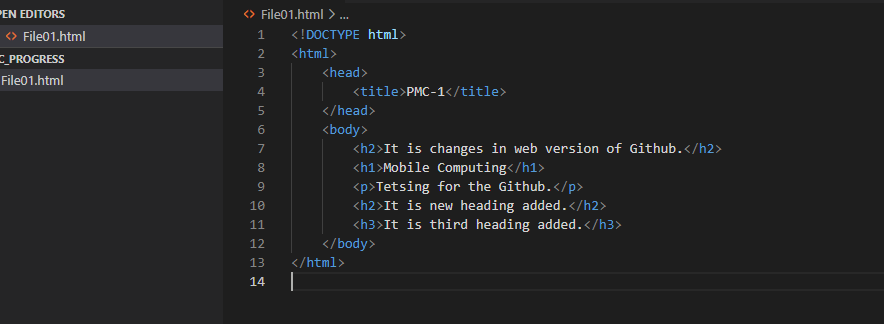


* **Step 04:** To addChanges in local repository add following command on cmd.

git pull



* **Step 05:** Changes added in local repository.



* **Files Deleted at Central Repository and Local Repository**
* **Step 01:** Add multiple files through single command on central repository with the help of following commands.

git add file name1 file name2

git commit -m “Message”

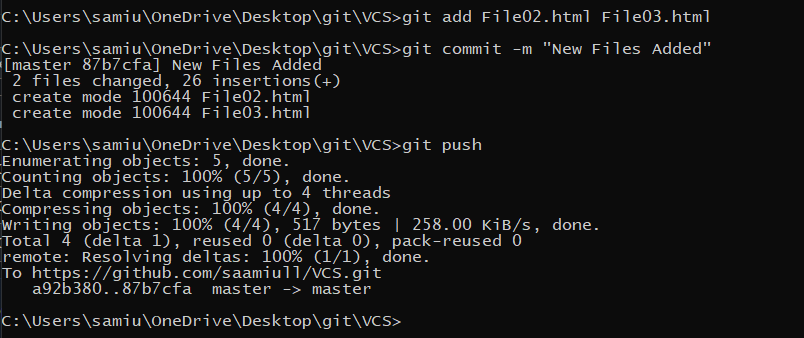
git push

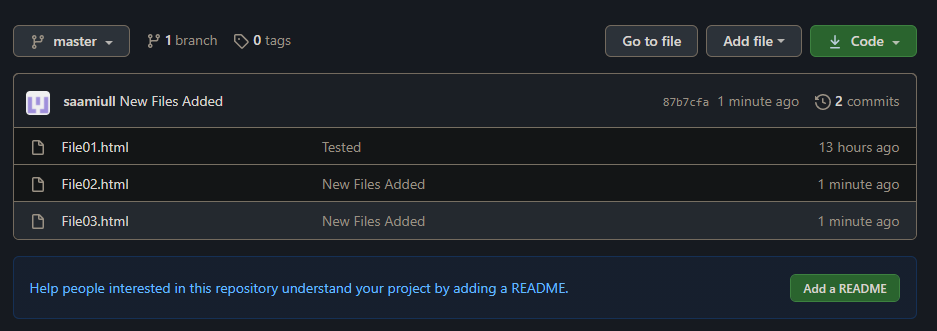
in my case it is

git add File02.html File03.html

git commit -m “New Files Added”

git push

****

****

* **Step 02:** To delete the file on central repository as well as on local repository we use following commands:

git rm File name

git add .

git commit -m “Message”

git push

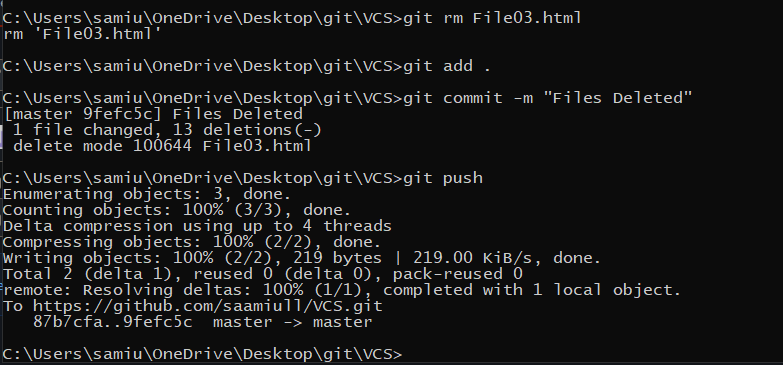
in my case it is

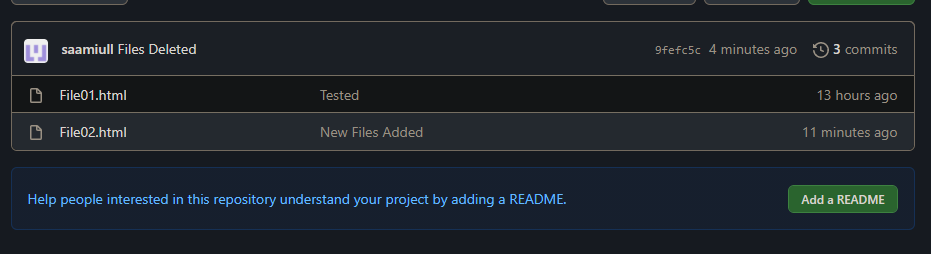
git rm File03.html

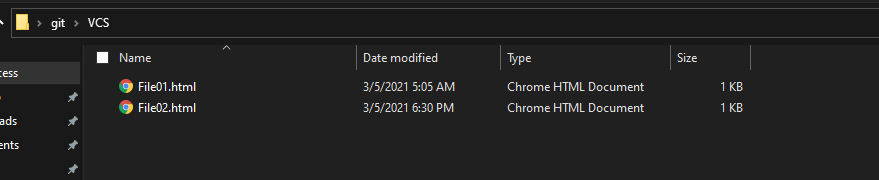
git add

git commit -m “Files Deleted”

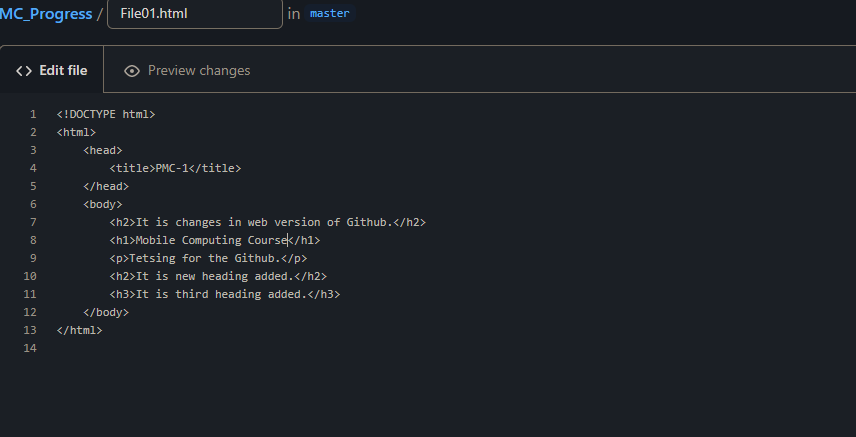
git push

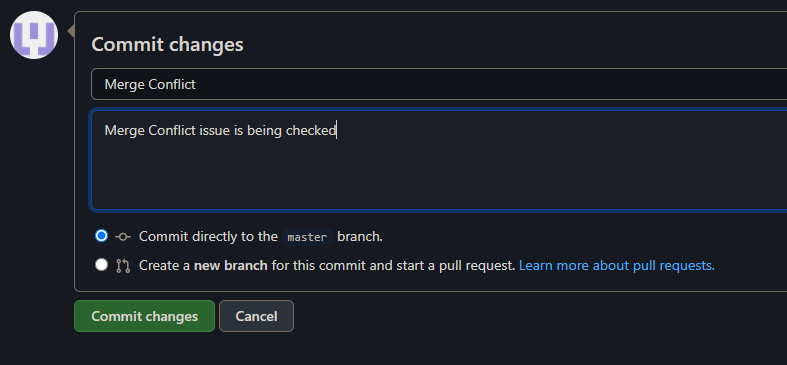
****

****

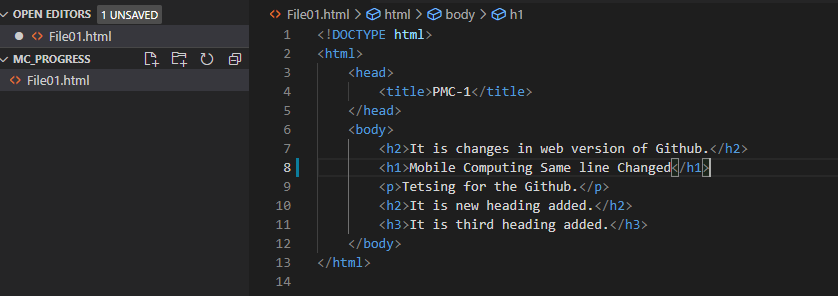
****

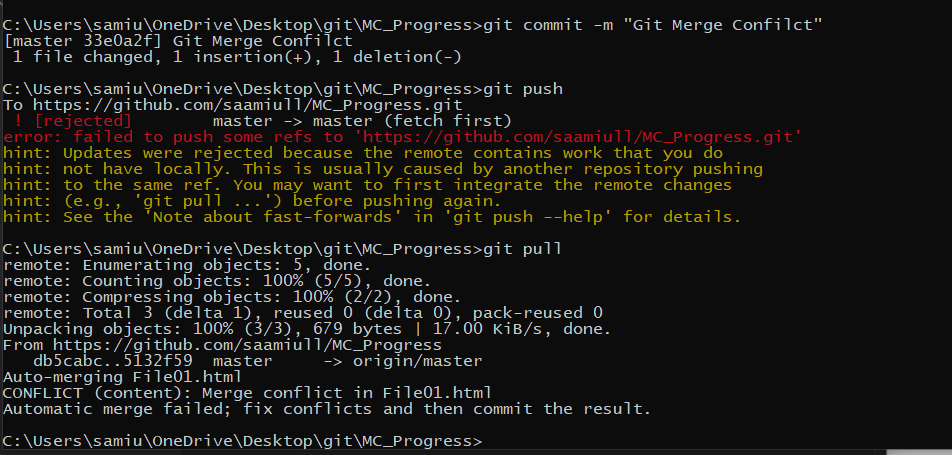
* **Merge Conflicts**
* **Step 01:** File changes at GitHub and add new commit.

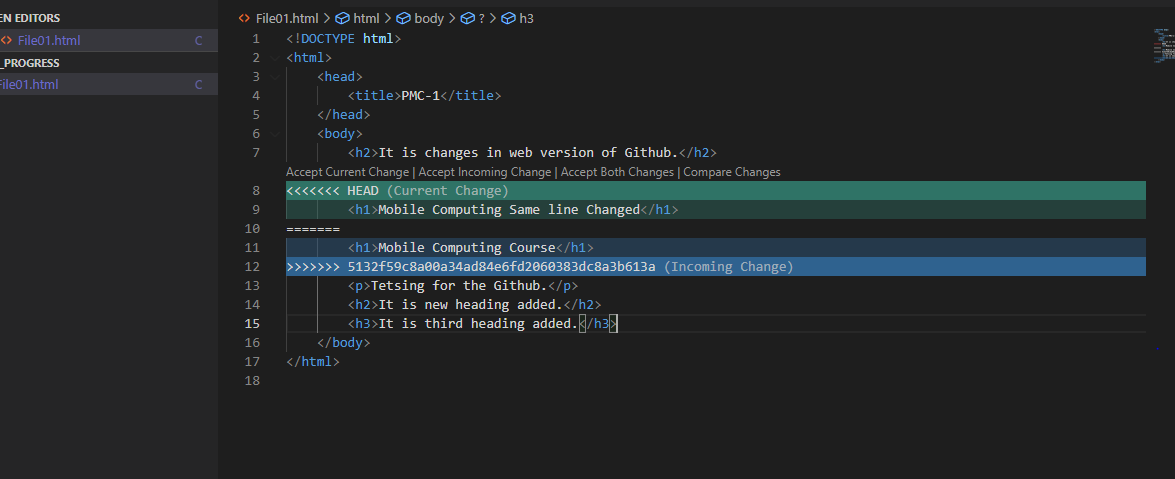




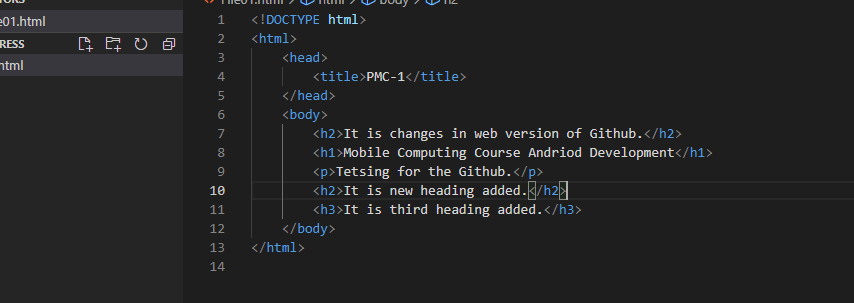
* **Step 02:** Same line changed at local repository, then add, commit and push through cmd.



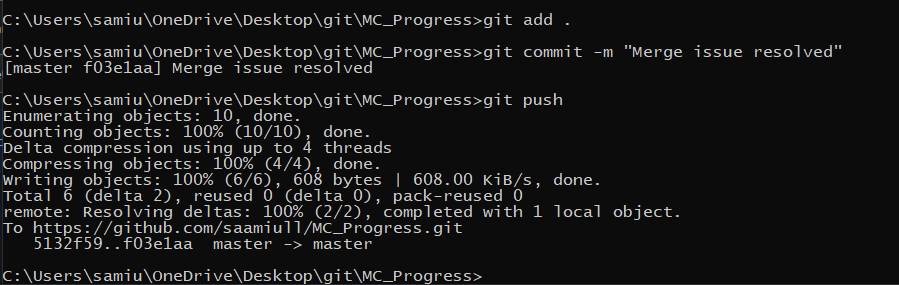




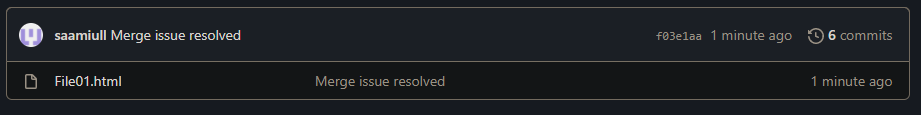
* **Step 03:** Select central repository changes or local repository changes to resolve merge issue. In my case, I am adding some extra text.



* **Step 04:** now add, commit and push again.



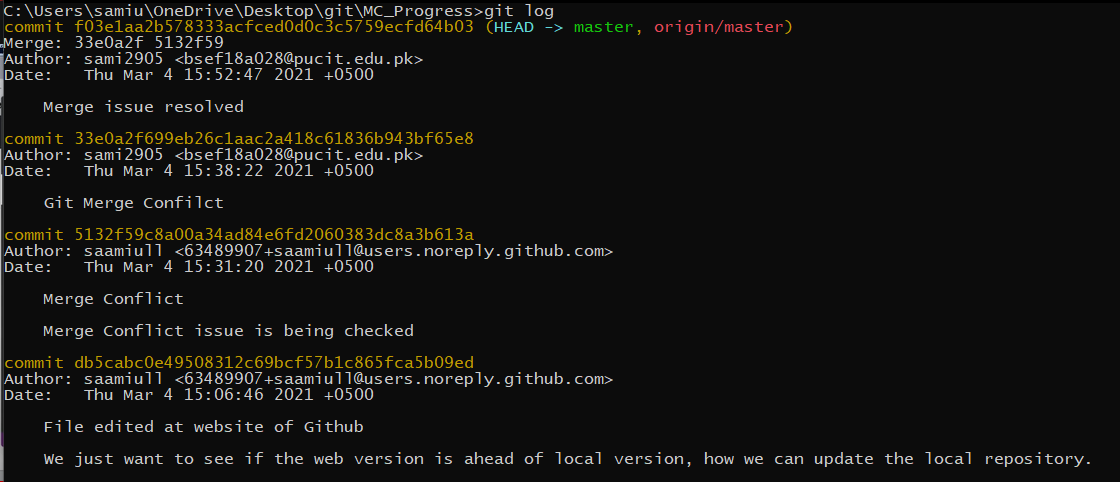
* **Step 05:** Issue resolved you can see on central repository.





* **Details of commits**
* **Step 01:** Write the following command on cmd:

git log

****

* **Branching**
* **Step 01:** To check your branch write the following command on cmd:

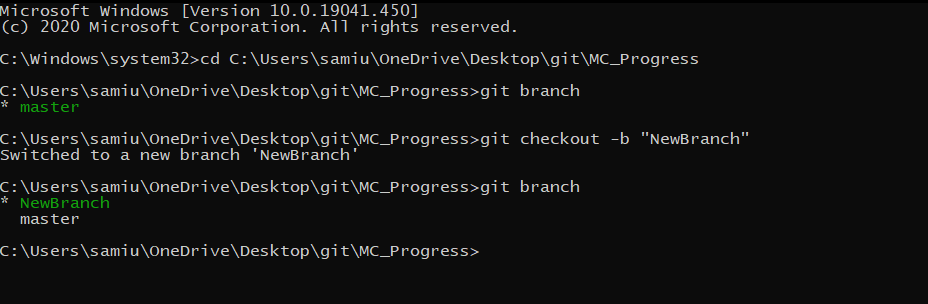
git branch

* **Step 02:** To create a new branch write the following command on cmd:

git checkout -b “Branch name”

in my case it is git checkout -b “NewBranch”.

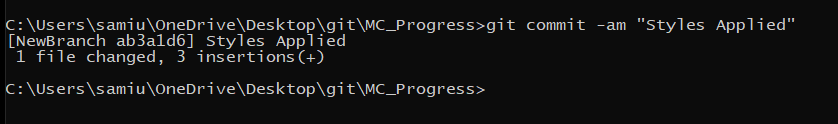
* **Step 03:** We are in new branch, to check it write the git branch command.



* **Step 04:** Add some extra code in new branch to test it.
* **Step 05:** Now add and commit the file by following command at a same time:

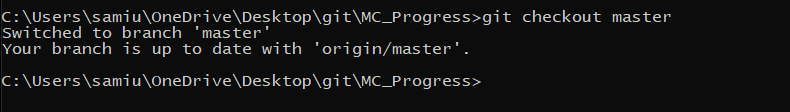
git commit -am “Commit Name”

in my case it is git commit -am “Styles Applied”.



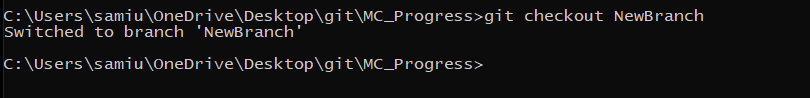
* **Step 06:** Switch to Master Branch with the help of following command.

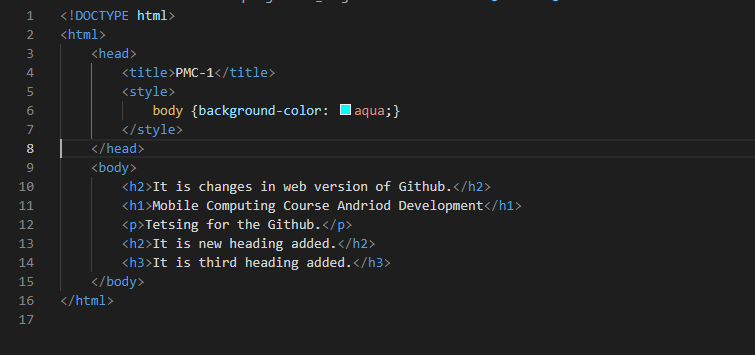
git checkout master





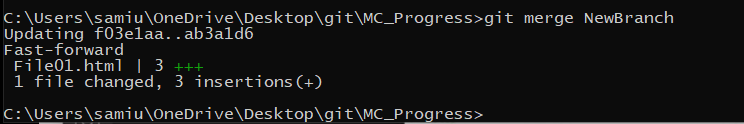
* **Step 07:** Switch to NewBranch with the help of following command.
* git checkout NewBranch





* **Merge Branches**
* **Step 01:** Add the new branch features or Merge the two Branches into master branch firstly switch to master branch then write the following command in cmd:

git merge NewBranch



* **Step 02:** See the master branch code.



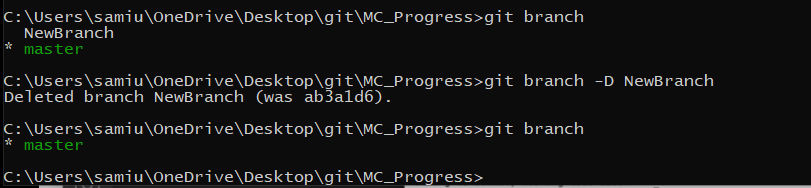
* **Delete Branch**
* **Step 01:** Firstly, check the all branches by following command:

git branch

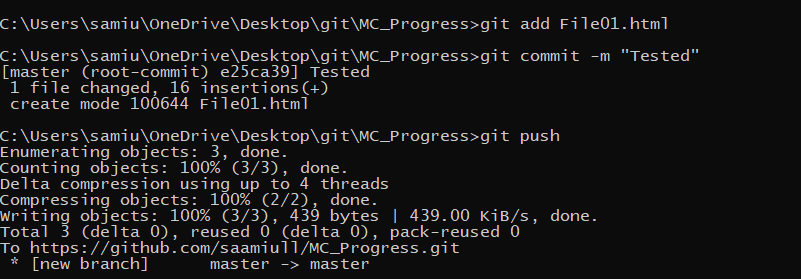
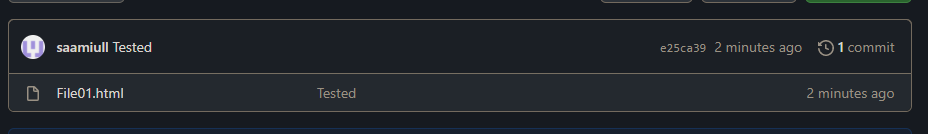
* **Step 02:** To delete the branch write the following command on cmd:

git branch -D Branch Name

in my case it is git branch -D NewBranch



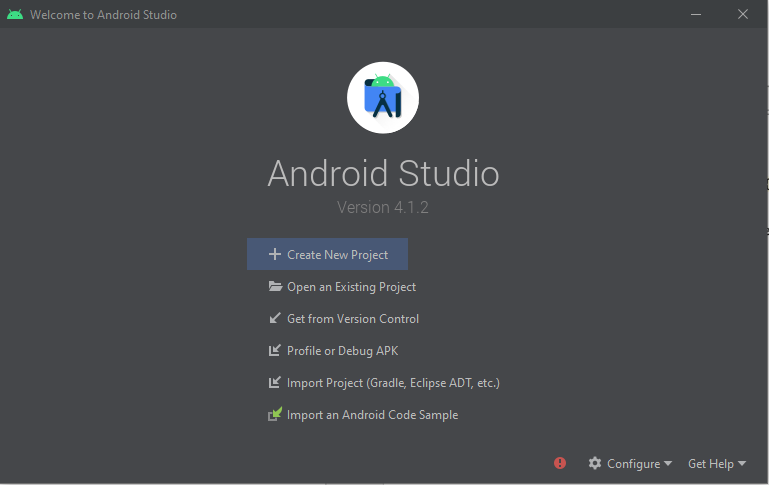
* **Step 03:** Now add, commit and push this new master branch

 .

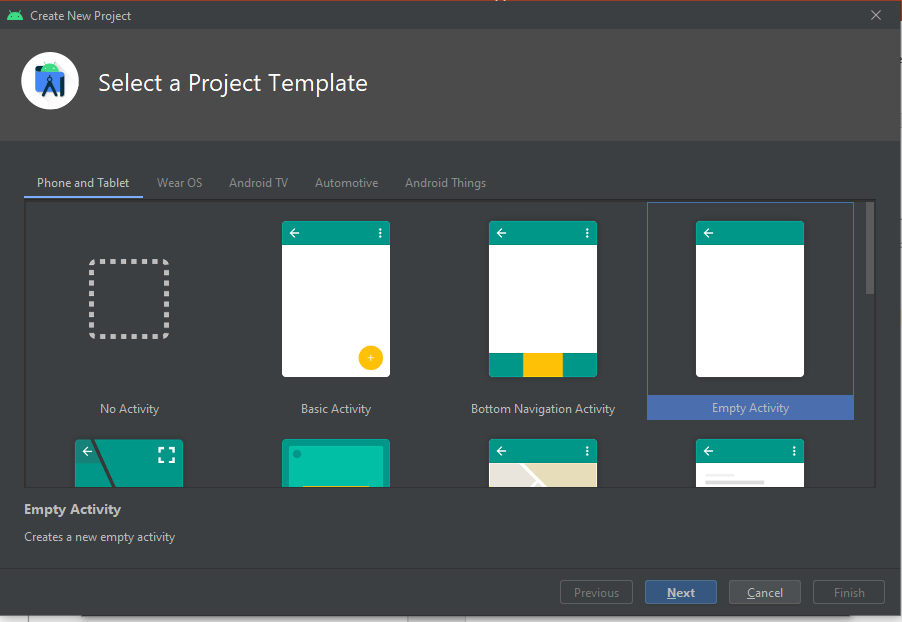
**Lecture – 04**

**Android Studio Development**

* **Making a new project**
* **Step 01:** To create a project on Android Studio, Open the Android Studio and select the Start a new Android Studio Project.

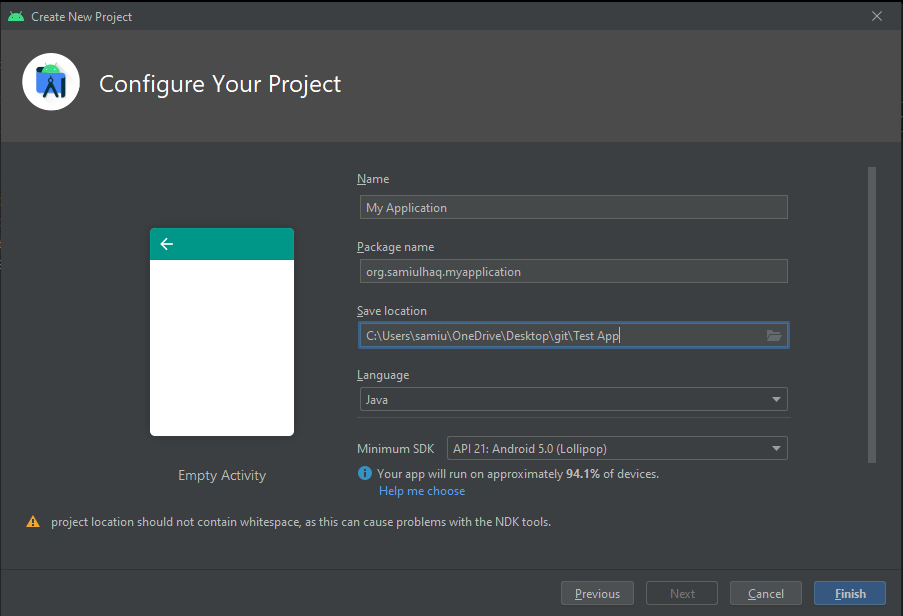


* **Step 02:** New Window is open Choose your activity Template; in my case I choose the Empty Activity and press the next button.



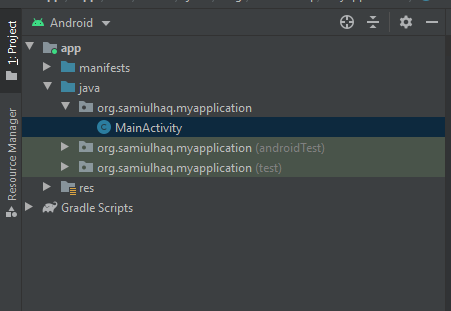
* + **Step 03:** New Window is open, write your project name, package name and set the location where you can save your project. Select language and API level. At last press the finish button.

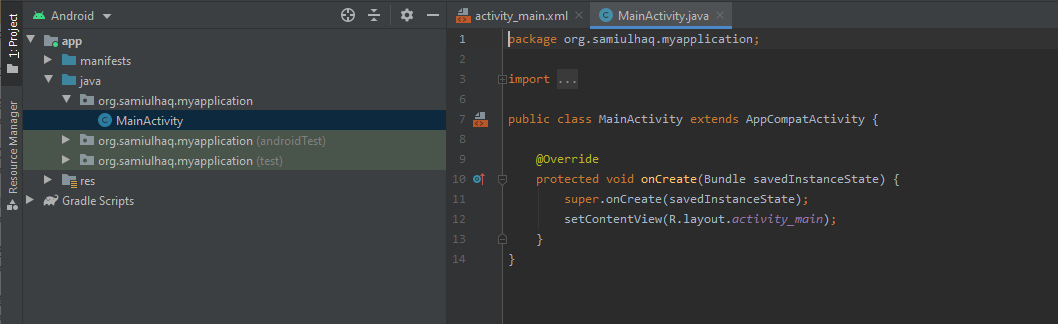
In my case, project name is My Application, my location is on Desktop folder git, package name is org.samiulhaq.myapplication, language is Java and API level is API 21:Android 5.0 (Lollipop). I choose this API level because of this my app run on 94.1% of devices.

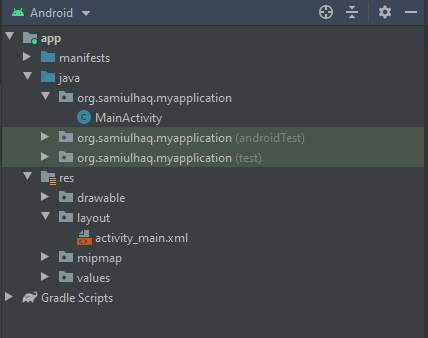


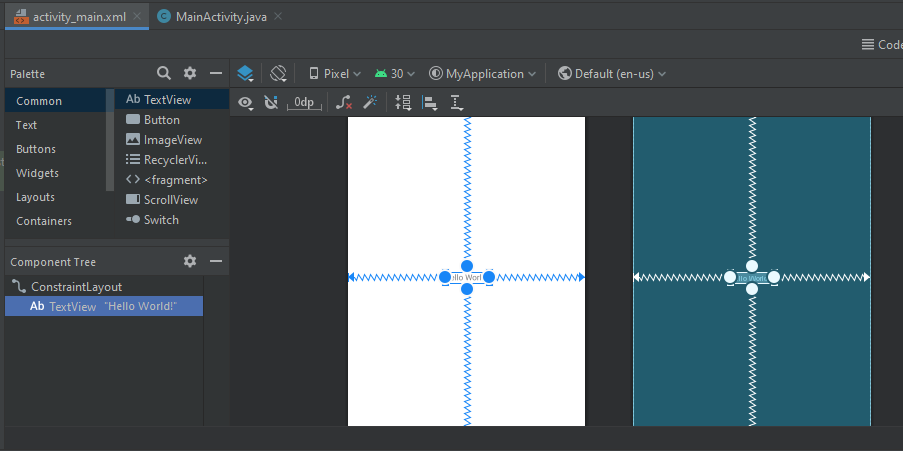
* **Step 04:** Now project is created.

* **Information about Android Studio and Project**
* **Step 01:** On left side you see the Android project folder hierarchy, all java code written in MainActivity.java file and practical user interface can be written in activity\_main.xml file which is present in layout folder under resource folder.

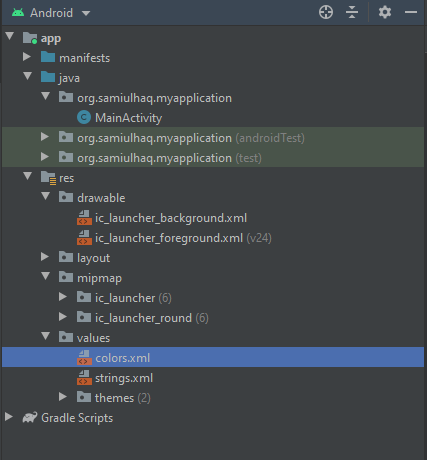




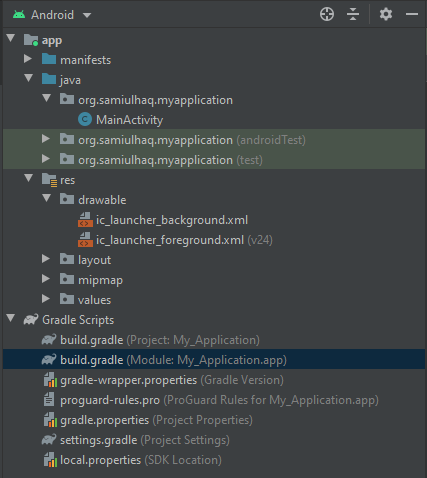


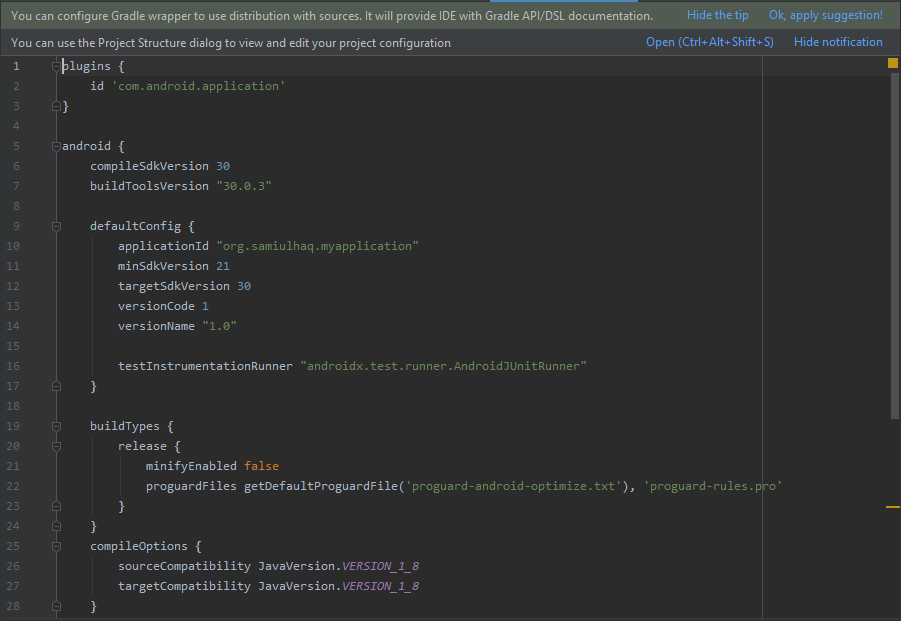


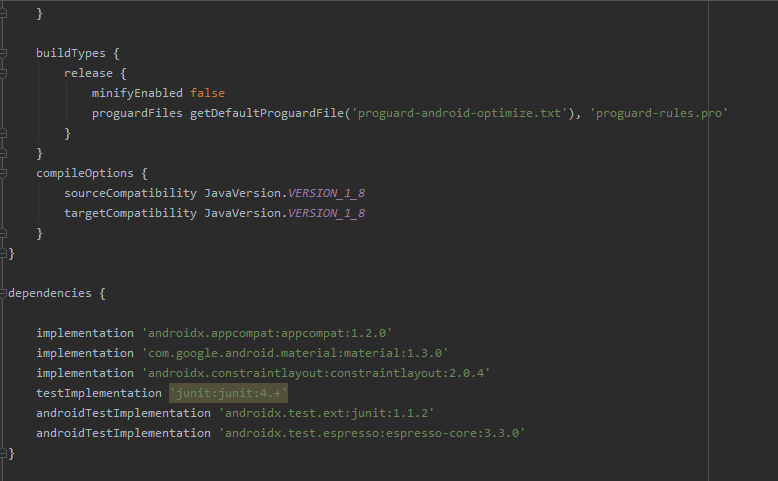
* **Step 02:** All images that can be used in our project present in drawable folder which is in under resources folder. All icons of project present in mipmap folder under resources folder. All values that can be used mostly are store in values folder under resources folder.



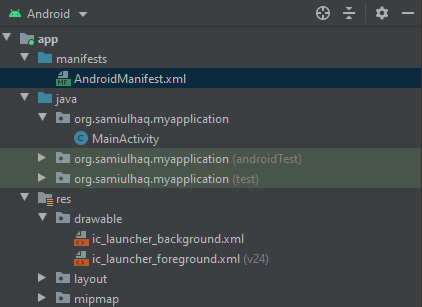
* **Step 03:** All information about your App can be present in Gradle Scripts.







* **Step 04:** All information about your project can be present in manifests folder, like icons can be present in mipmap folder.

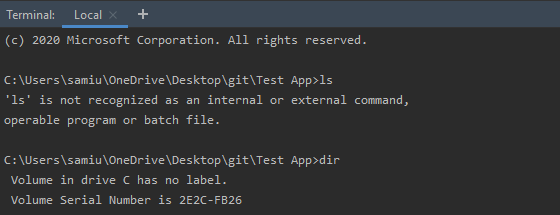
****

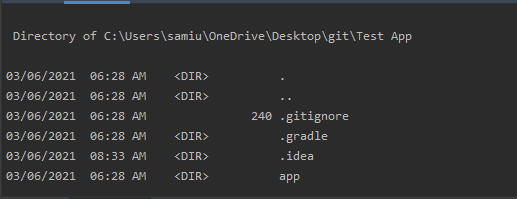
****

* **Step 05:** To Do Task is used for resume or pending work. For this we write //TODO: and after this write message about task. It is like a comment. At a glance we check our pending tasks.

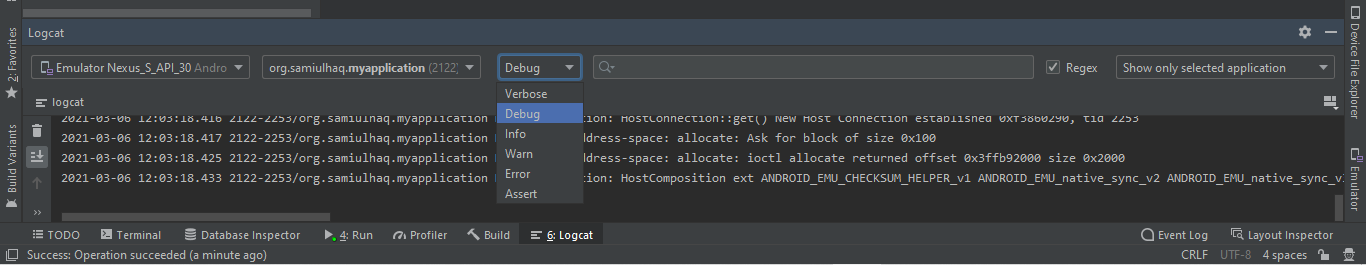
****

* **Step 06:** Terminal is present for running the commands. Like I run the dir command to see all directories because I am using windows OS.

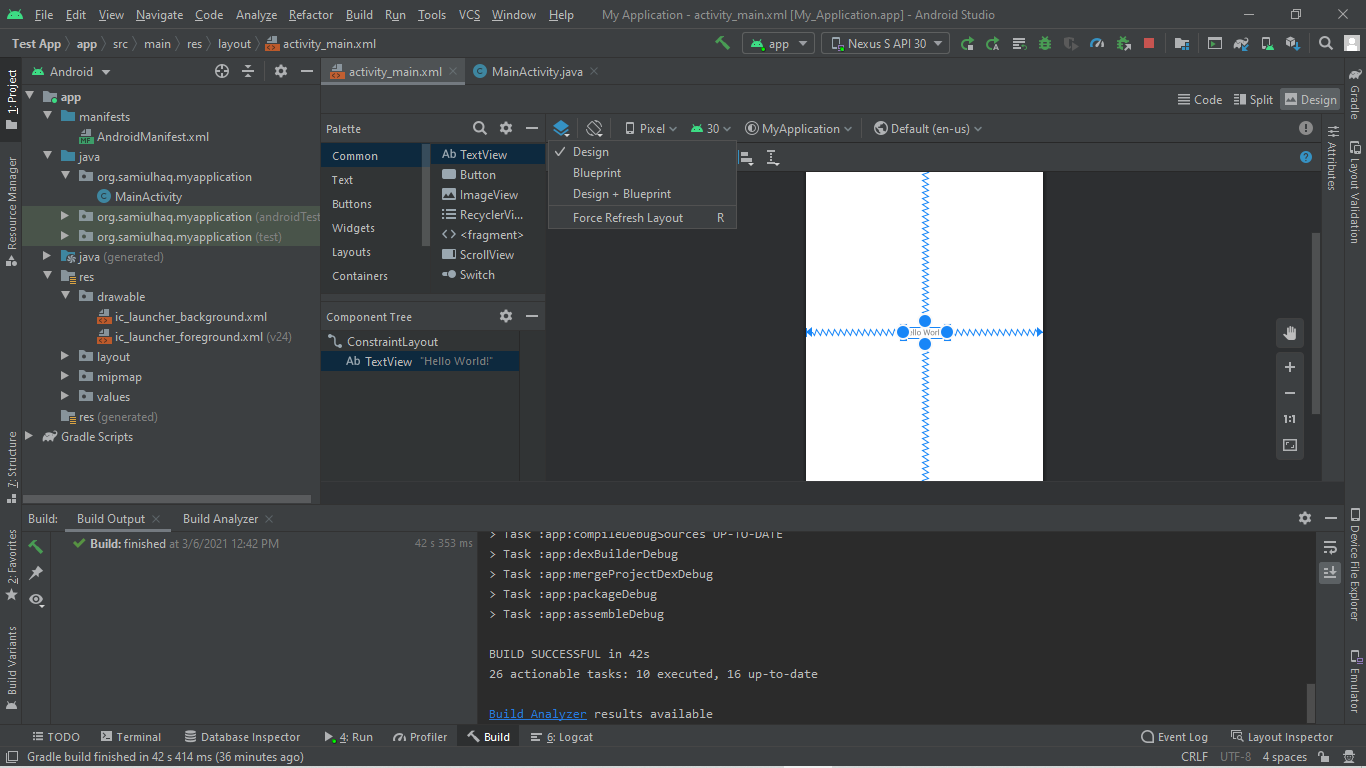


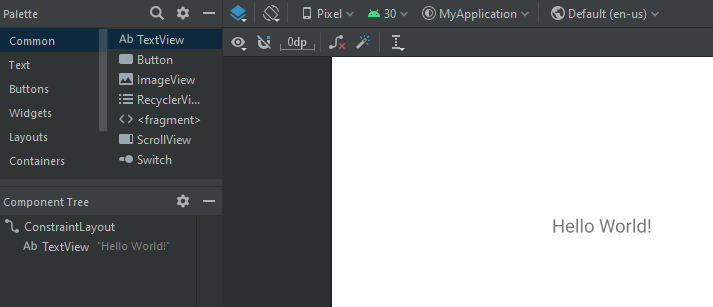


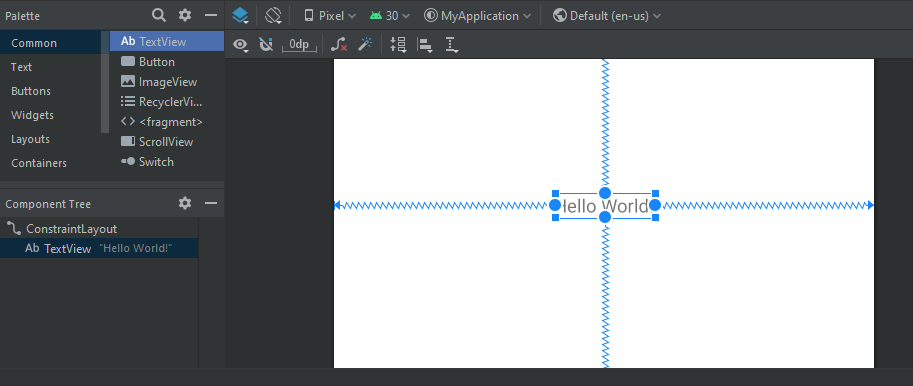
* **Step 07:** Through Logcat you can debug the program, check error etc. Logcat is a command-line tool that dumps a log of system messages.

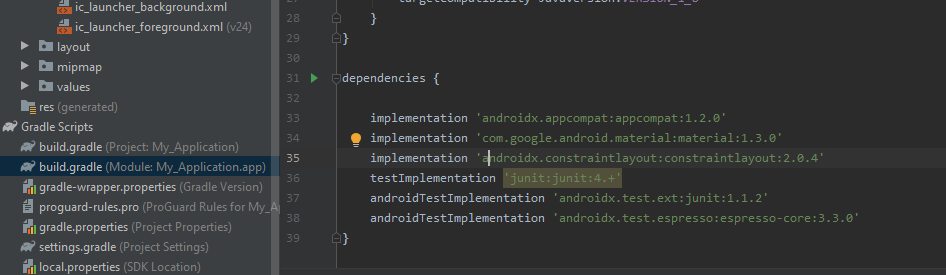


* **Step 08:** Design view, Blueprint view and Design view+Blueprint can be changed easily through view button which is available on bottom menu bar. Constraint’s layout can be seen through ConstarintLayout button. Text can be connected with four-sided spring it can be changeable. Its dependency adds in Gradle folder.

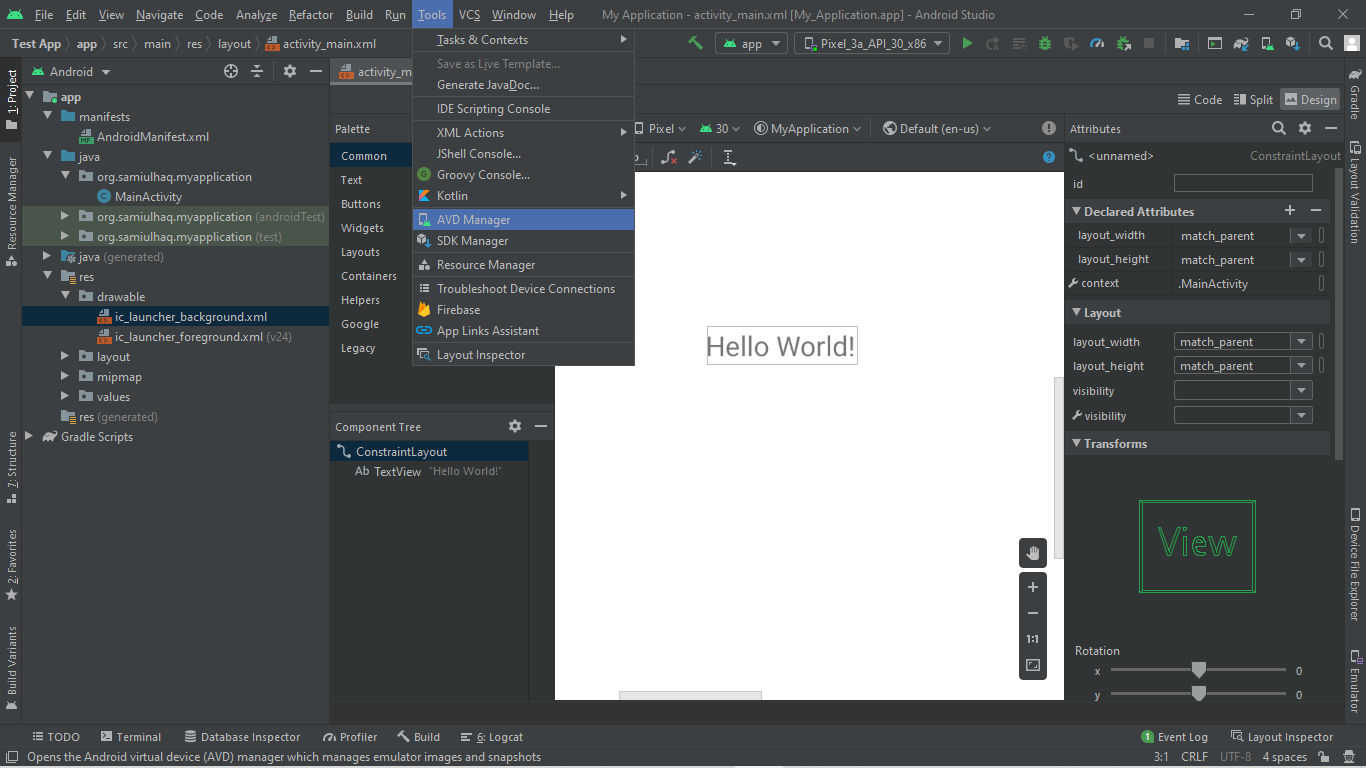






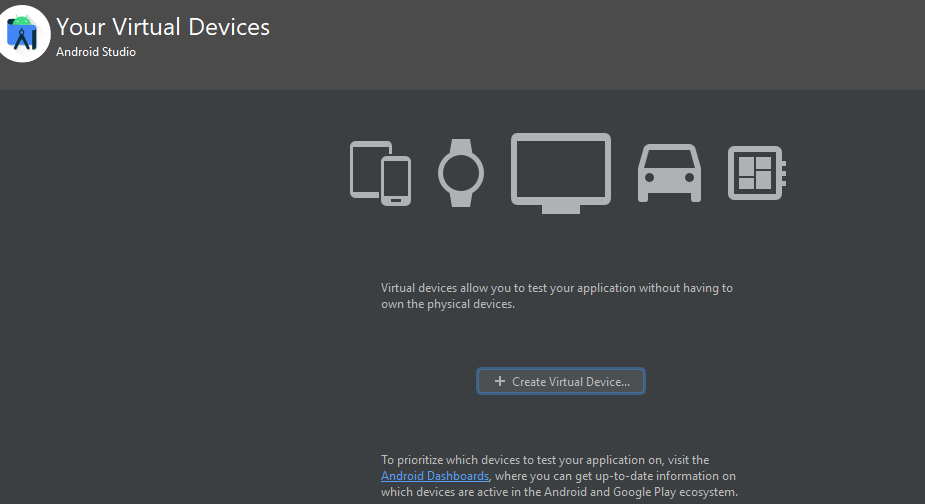


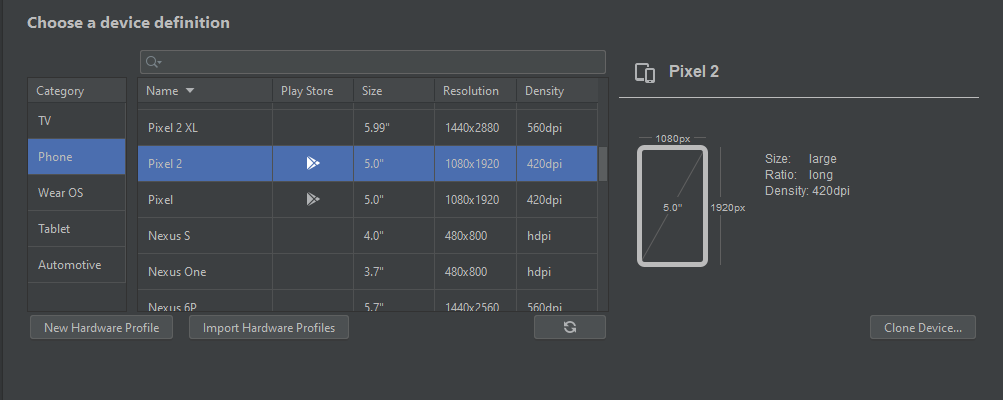
* **Install Virtual Device**
* **Step 01:** To run the app firstly we need to install the virtual device on which our output will be show.
* **Step 02:** On menu bar select the Tools and then select AVD Manager.

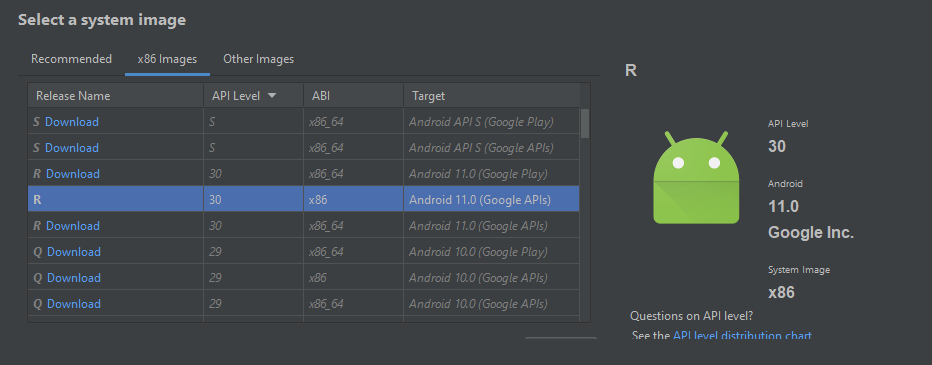


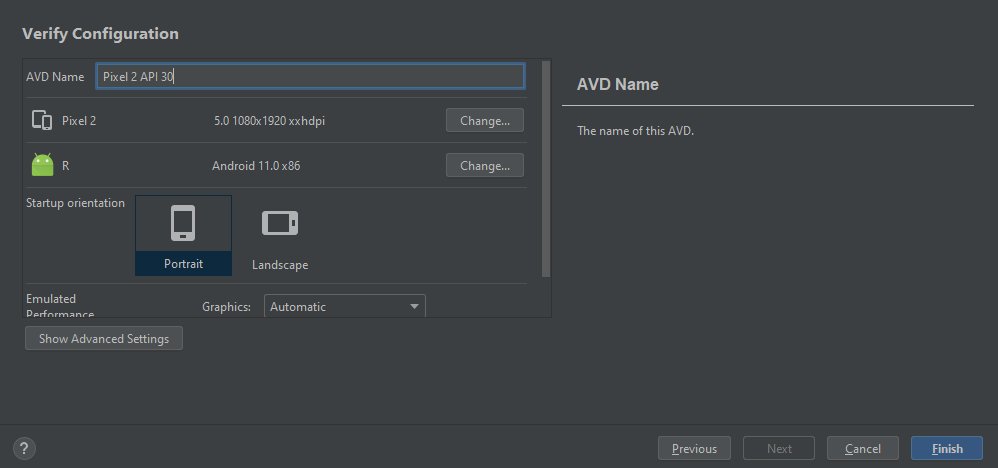
* **Step 03:** New window is open to install virtual device, press the Create Virtual Devices button, new window is open, select Phone and device name which you need to be install, press next button new window is open, select API level which you need to be install and press next button, new window is open write device name, select orientation and press the finish button.

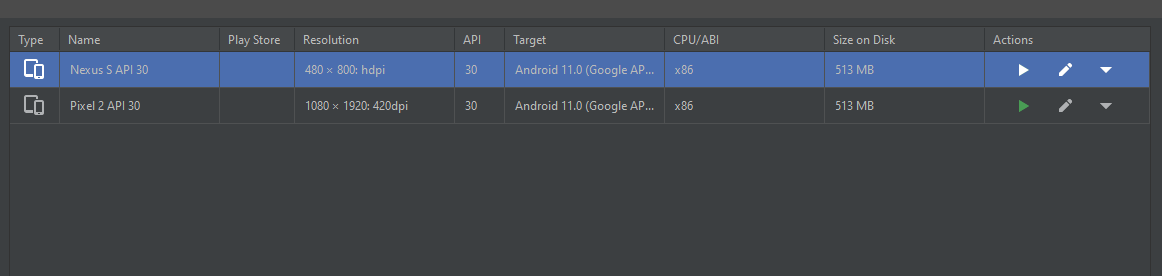
In my case, I installed two virtual devices Pixel 2 API 30 and Nexus S API 30 by default name which is given by Android Studio.



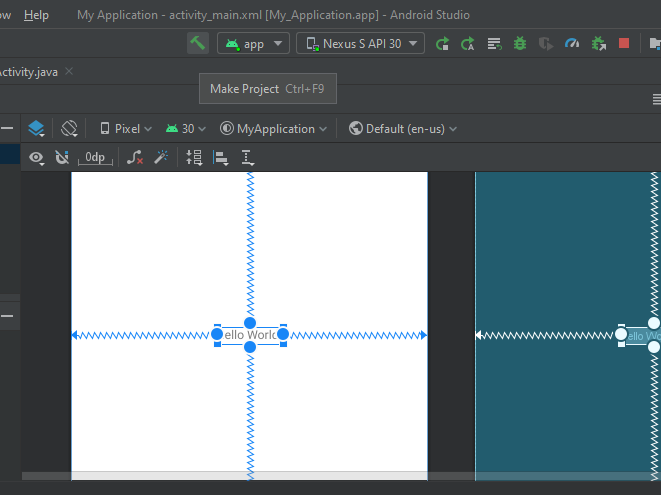


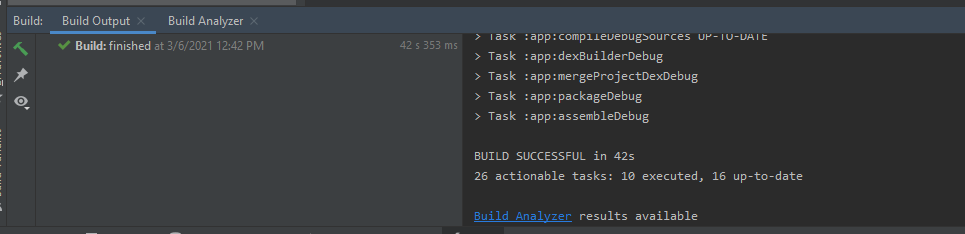


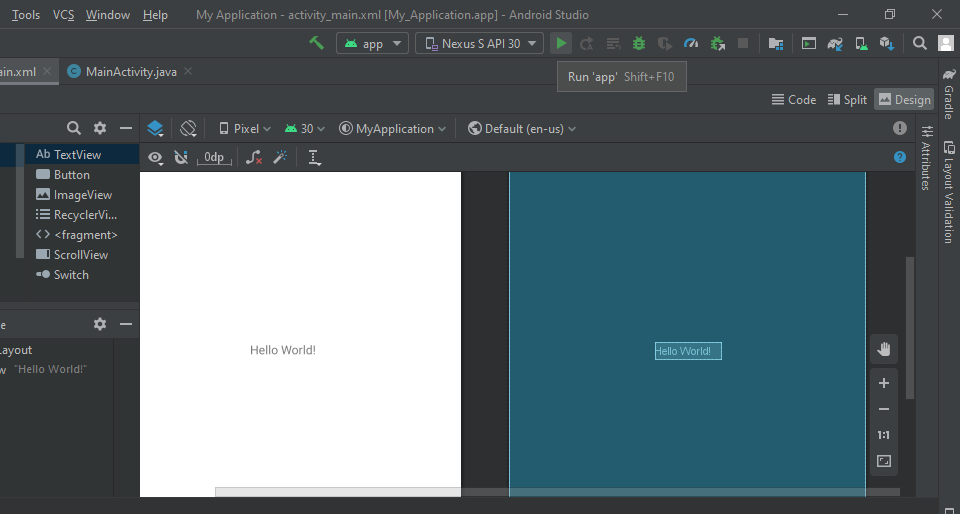


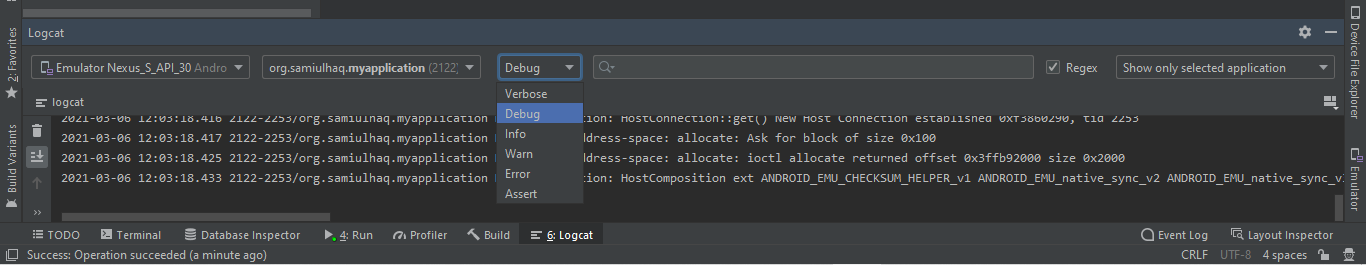


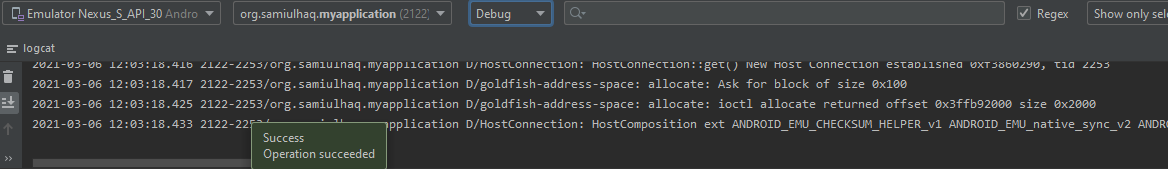
* **Run Default Program**
* **Step 01:** To run the default Hello World program click the build button then click run button, now you can see program is running. And you see the virtual device is one of them you can install and select for this program. You can see after some time Hello World is shown on Nexus S API 30 Virtual Device.

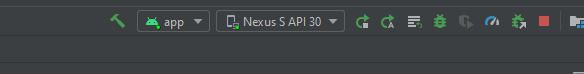


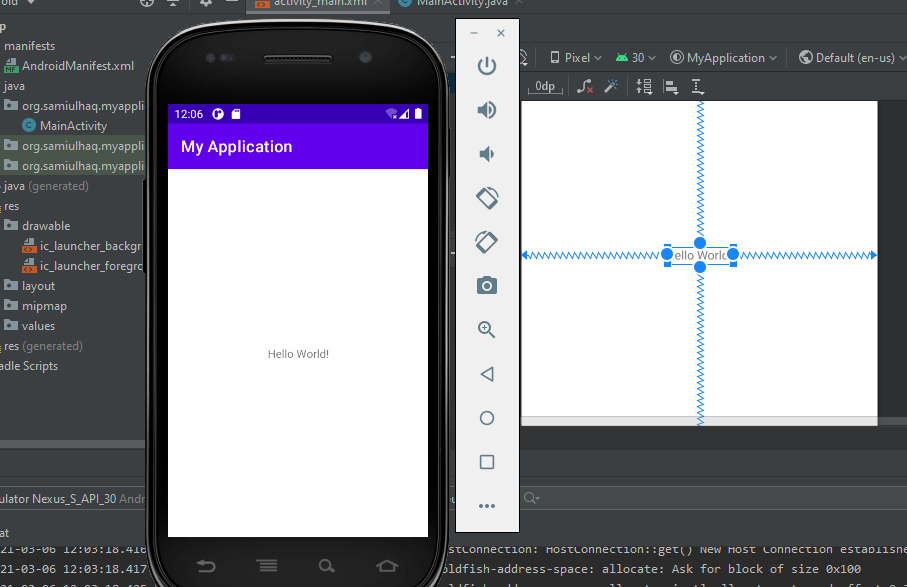








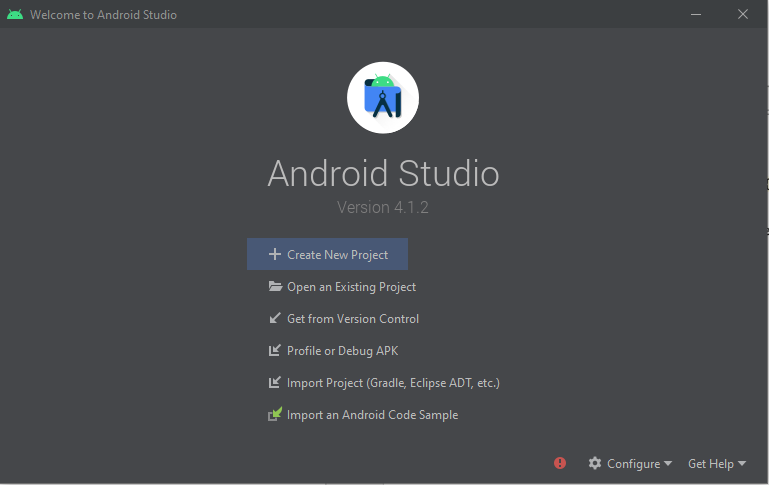


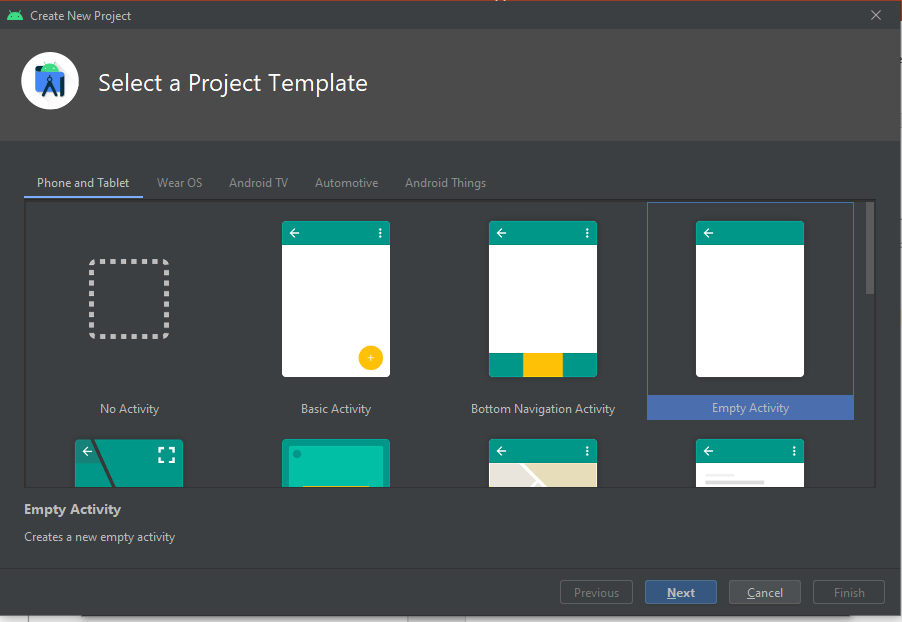


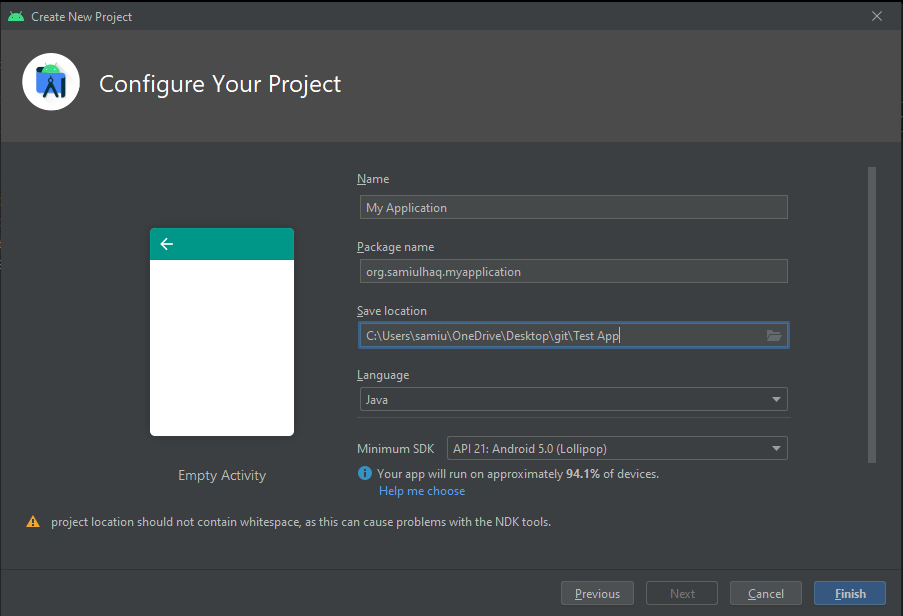
* **Multiple Screens Task**

**Task-01**

* **Step 01:** Open the Android Studio a project with name My Application in Test App on Desktop under git folder.

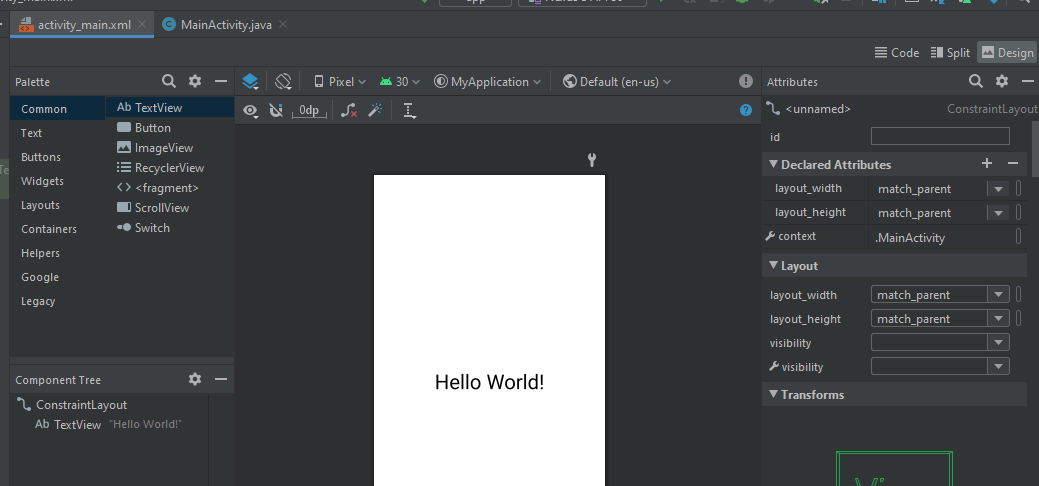






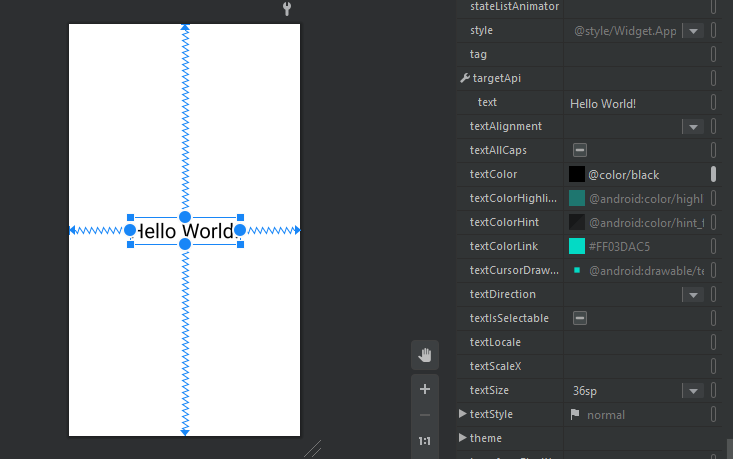
* **Step 02:** Now project is created, go into activity\_main.xml file, on right side go in Attributes, then in All Attributes, select the Hello World text, you see the option text then remove the text hello world and type the

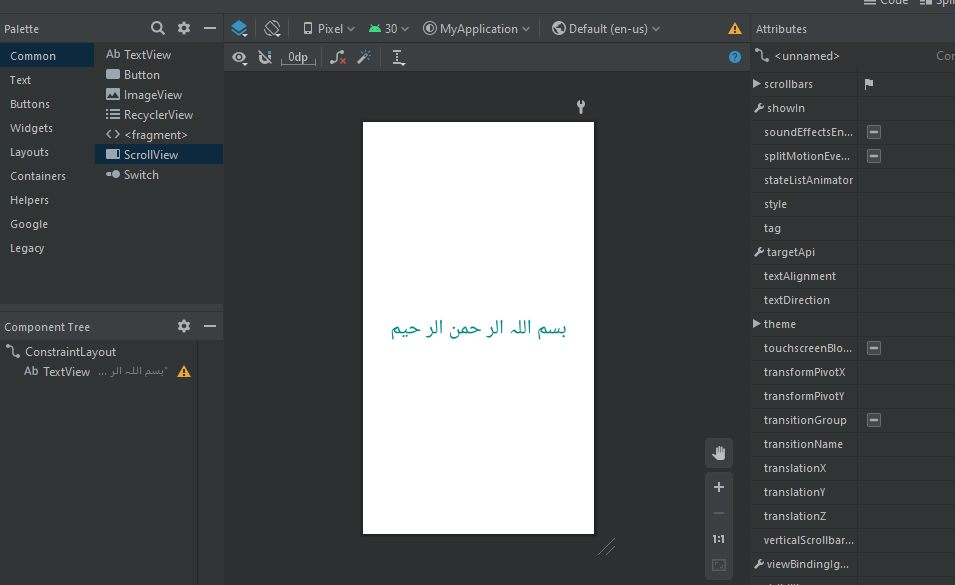
﷽ .



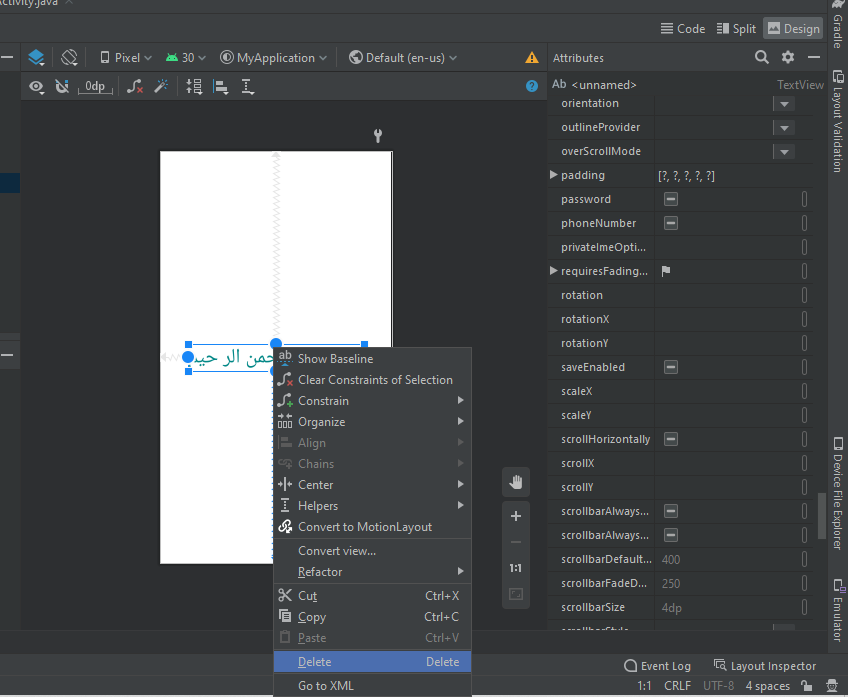


* **Step 03:** Then change the text color into teal-700 and set the size of text 36sp.

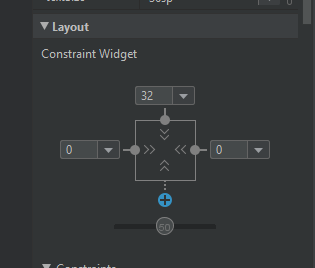


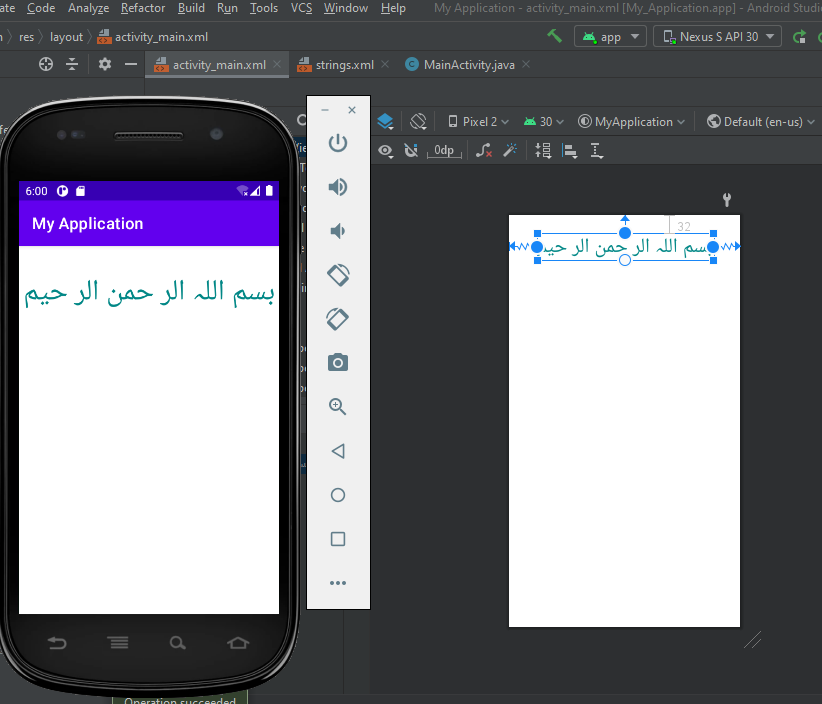


* **Step 04:** Now we can move the text on the top, for this delete the lower side constraint layout spring.



* **Step 05:** Add Space on top side by 32 through layout**,** Now selects the virtual device, then Build and Run this project.



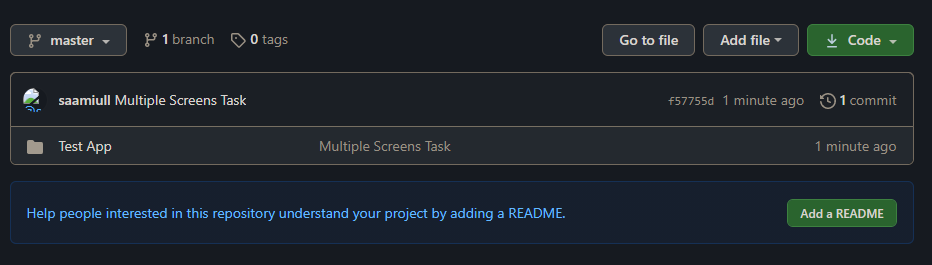


* **Step 06:** Now make a repository on GitHub, the add this project on GitHub by following commands on cmd:

git clone <https://github.com/saamiull/Android-Studio-Introduction.git>

git add “Test App”

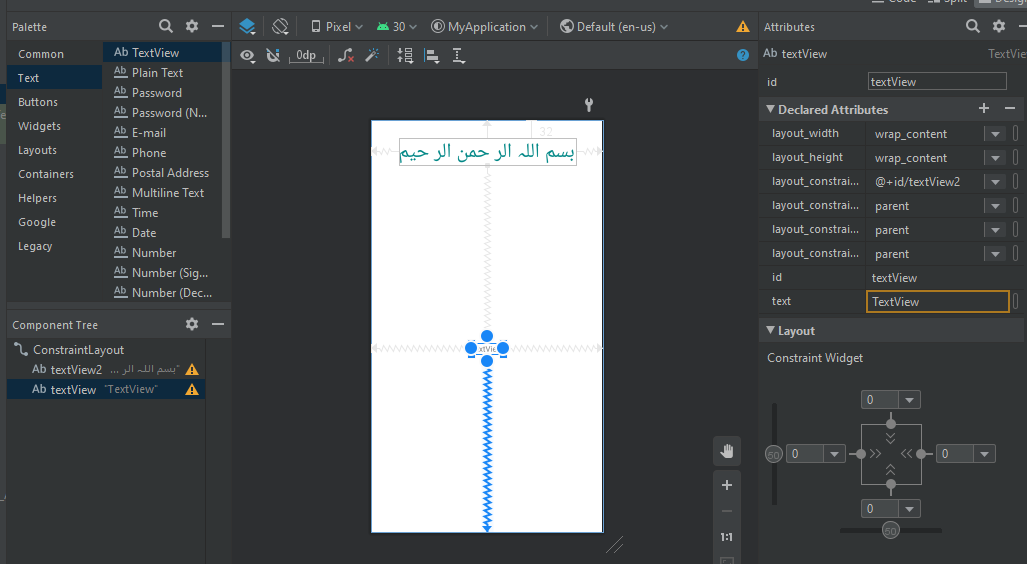
git commit -m “Multiple Screens Task”

git push 

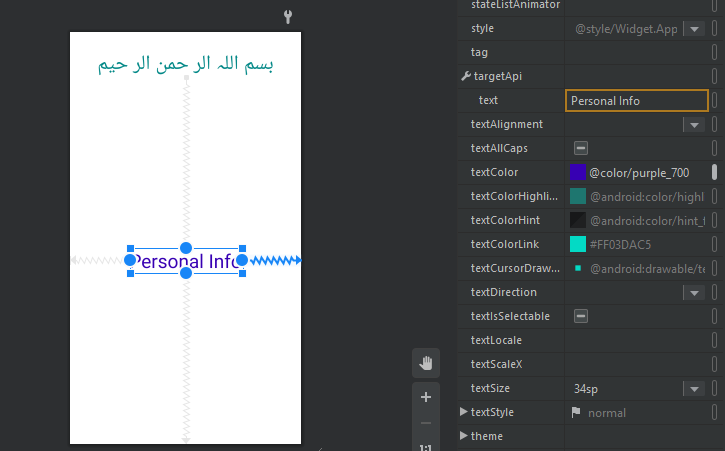
* **Step 07:** Now Project added on GitHub.

**Task-02**

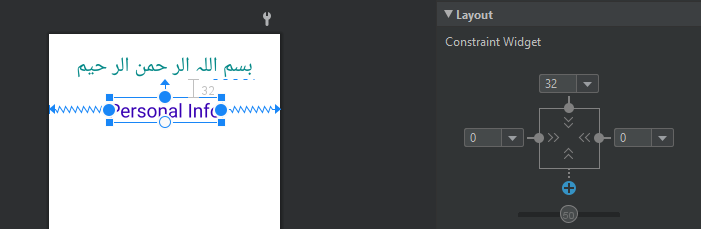
* **Step 01:** In My Application project, add new Text View under Text folder. Then add the springs of new Text view to all borders, but upper spring can be attached to the lower spring corner of previous Text View.



* **Step 02:** Now select the new Text View, rename it by Personal Info, change the text color to purple\_700 and change the text size to 34sp.



* **Step 03:** Now Add 32 Space in upper spring, and delete the lower spring.



* **Step 04:** Again, Build and Run the project. Now Output can be shown on same virtual device Nexus S API 30.

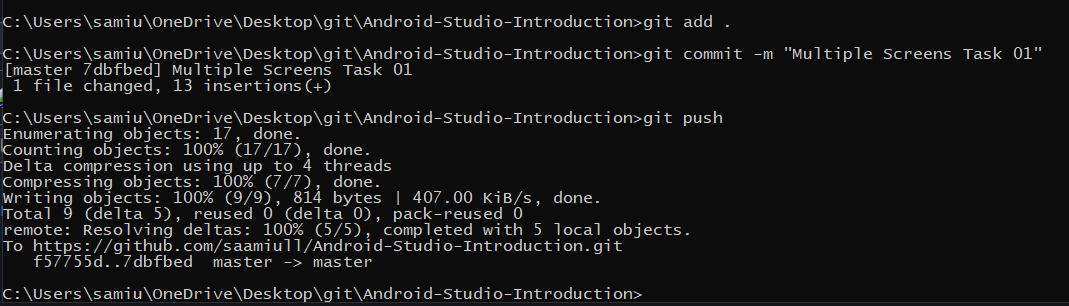


* **Step 06:** Now again add this project on GitHub by following commands on cmd:

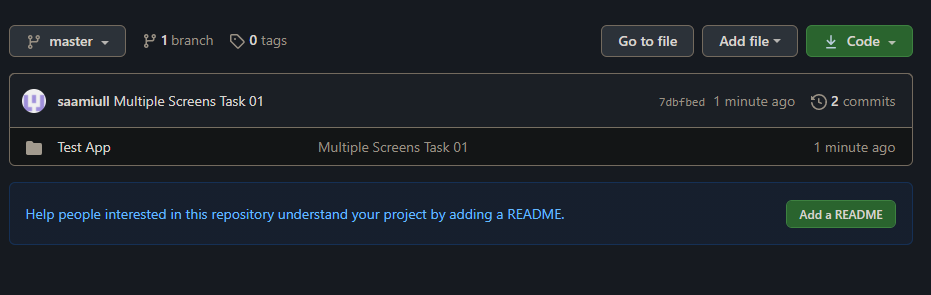
git add .

git commit -m “Multiple Screens Task 01”

git push

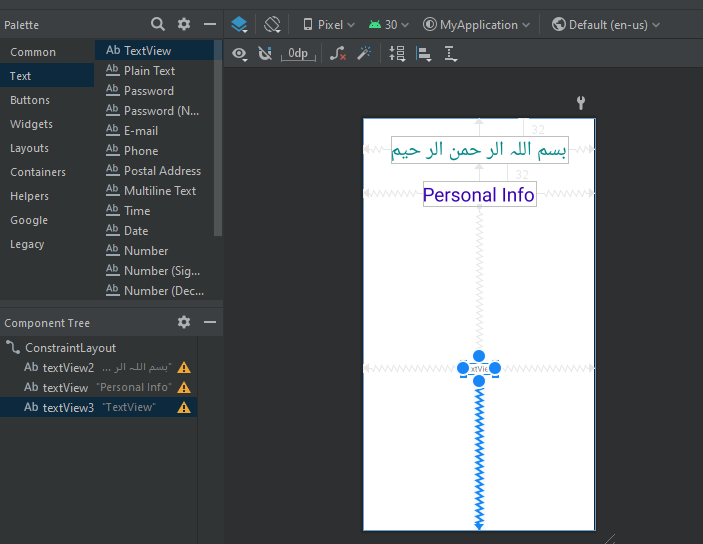


* **Step 07:** Now Project added on GitHub.

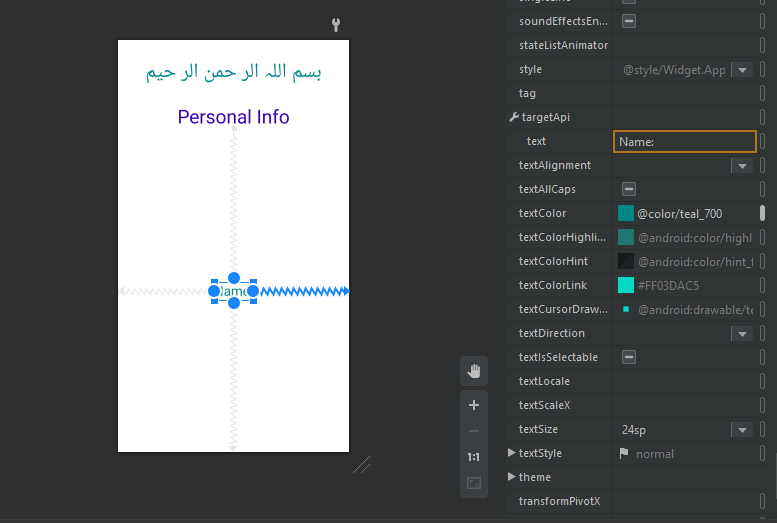


**Task-03**

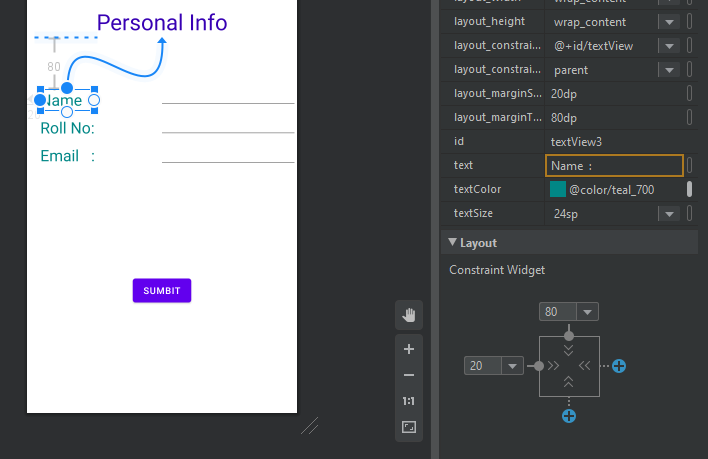
* **Step 01:** In My Application project, add new Text View under Text folder. Then attach the springs of new Text view to all borders, but upper spring can be attached to the lower spring corner of previous Text View.



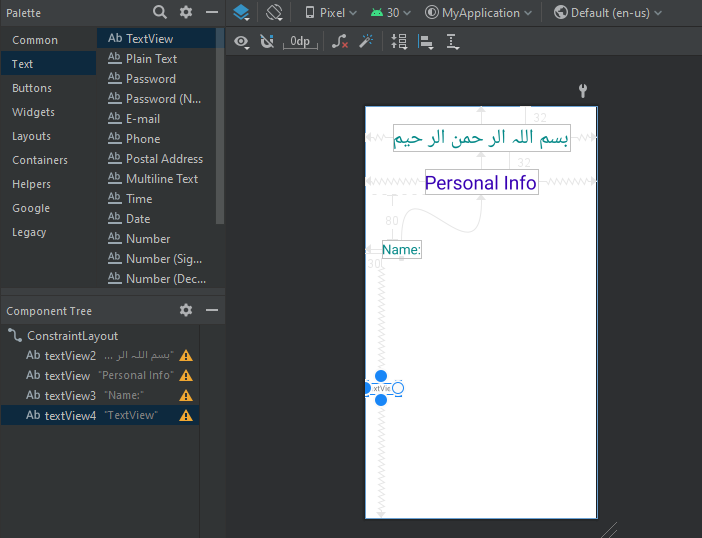
* **Step 02:** Now select the new Text View, rename it by Name:, change the text color to teal\_700 and change the text size to 24sp.



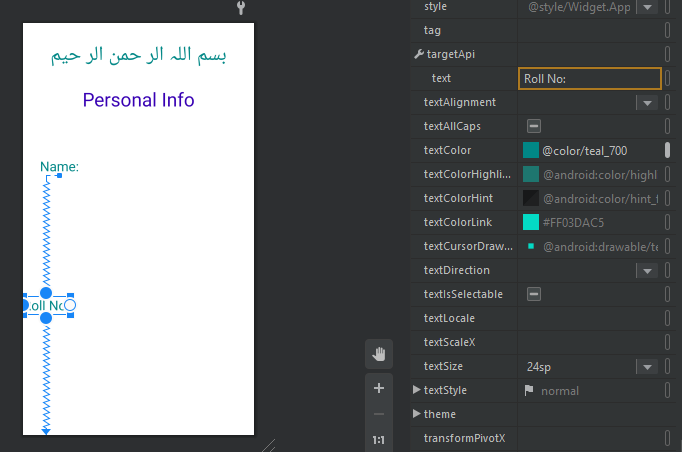
* **Step 03:** Now delete the lower spring and right spring, add space 80 in upper spring and 20 space in left spring.



* **Step 04:** Add new Text View under Text folder. Then attach the lower springs of new Text view to lower side, left spring to left side but upper spring can be attached to the lower spring corner of previous Text View (Name:).



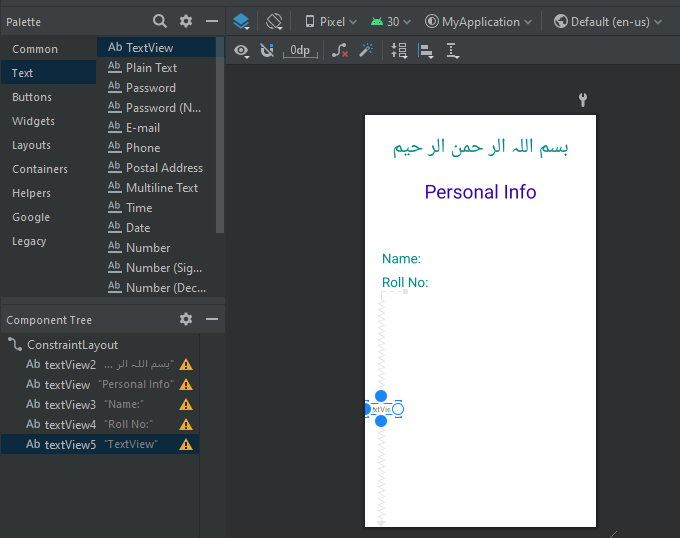
* **Step 05:** Now select the new Text View, rename it by Roll No:, change the text color to teal\_700 and change the text size to 24sp.



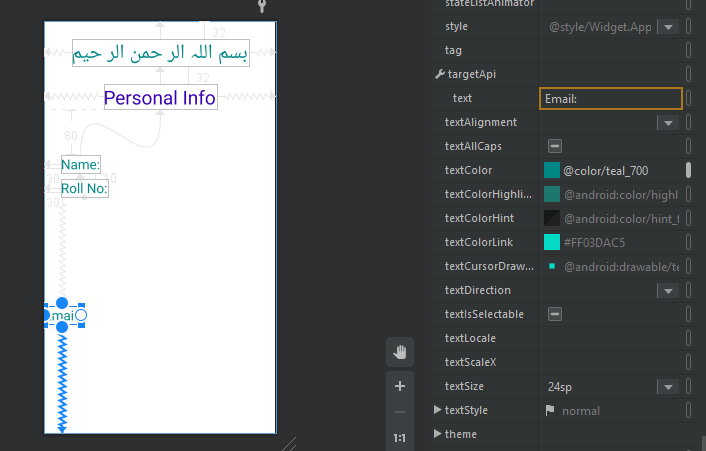
* **Step 06:** Now delete the lower spring, add space 10 in upper spring and 20 space in left spring.



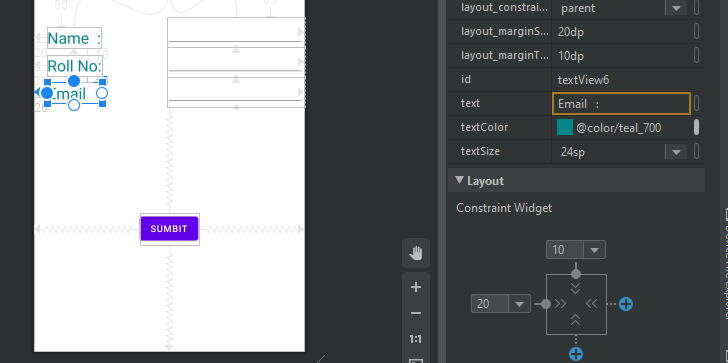
* **Step 07:** Add new Text View under Text folder. Then attach the lower springs of new Text view to lower side, left spring to left side but upper spring can be attached to the lower spring of previous Text View (Roll No:).



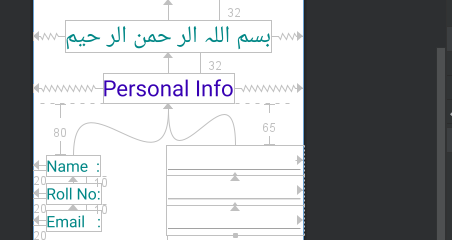
* **Step 08:** Now select the new Text View, rename it by Email:, change the text color to teal\_700 and change the text size to 24sp.



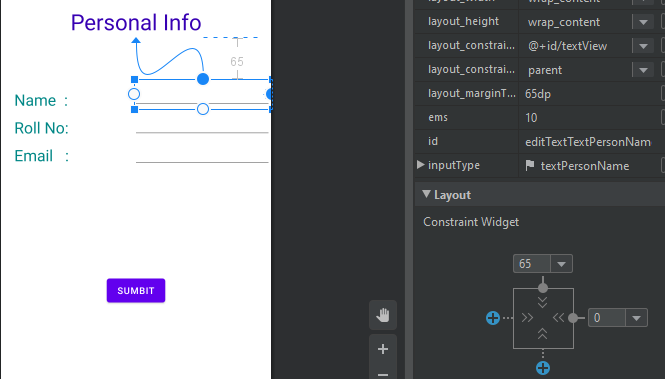
* **Step 09:** Now delete the lower spring, add space 10 in upper spring and 20 space in left spring.



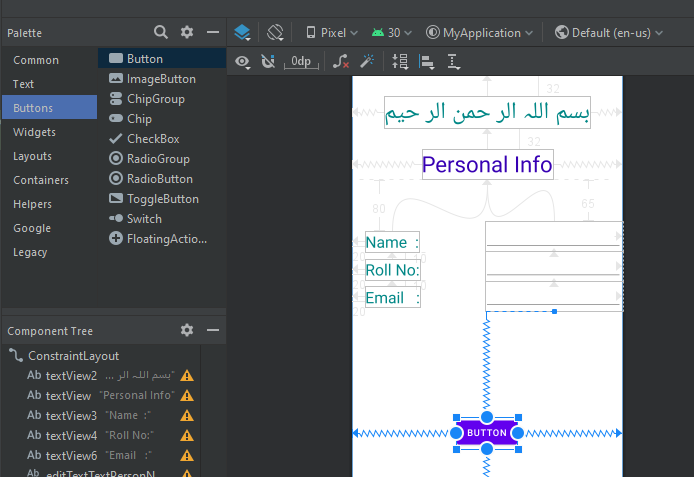
* **Step 10:** Add new Text View under Text folder. Then attach the right springs of new Text view to right side, upper spring to lower side of Personal Info. Again, Add new Text View under Text folder. Then attach the right springs of new Text view to right side, upper spring to lower spring of previous Text View. Then again, Add new Text View under Text folder. Then attach the right springs of new Text view to right side, upper spring to lower spring of previous Text View.



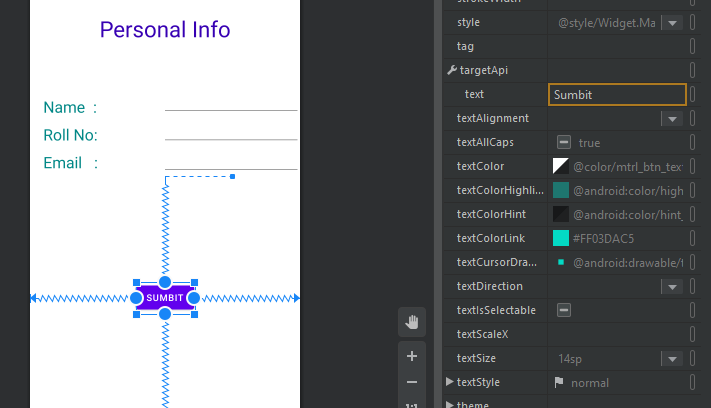
* **Step 11:** Now add 65 spacing in upper Text View among last three.



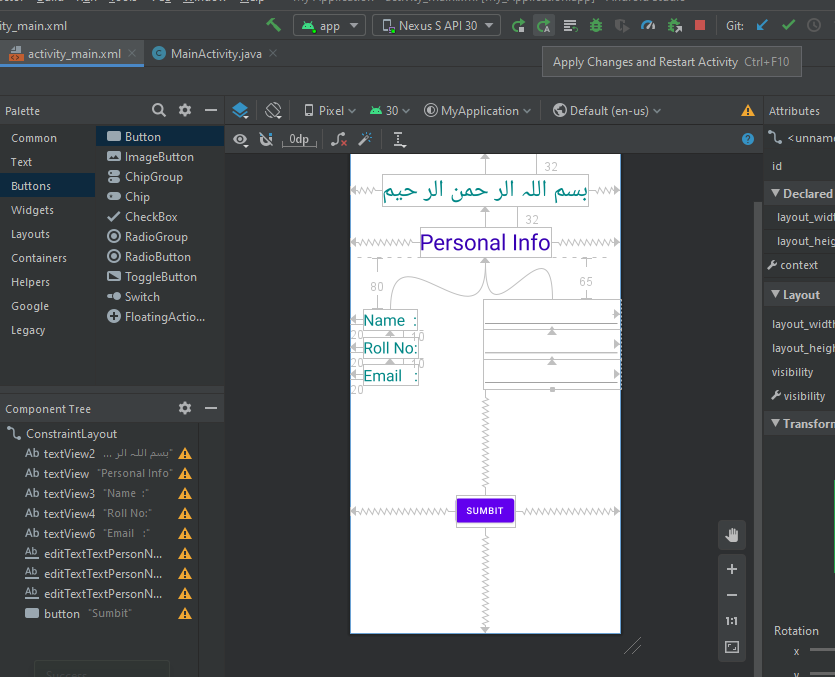
* **Step 12:** Now add Button and attach all spring to borders but upper spring attach to the lower spring of last Text Viewer precious three Text Viewer.

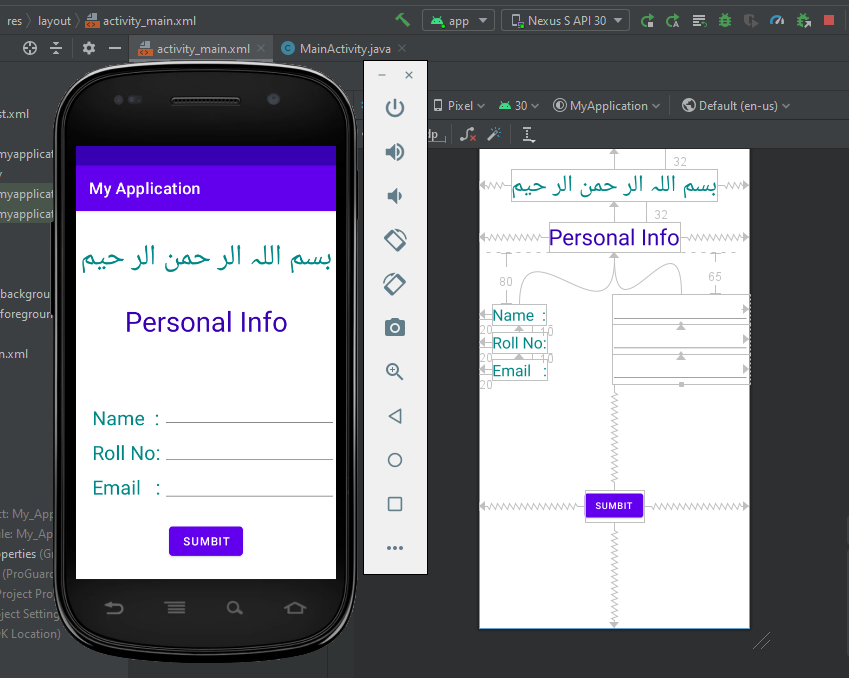


* **Step 13:** Reset the name of button to SUBMIT.



* **Step 04:** Now Apply Changes and restart the activity then output will be shown on same virtual device Nexus S API 30.





* **Step 06:** Now again add this project on GitHub by following commands on cmd:

git add .

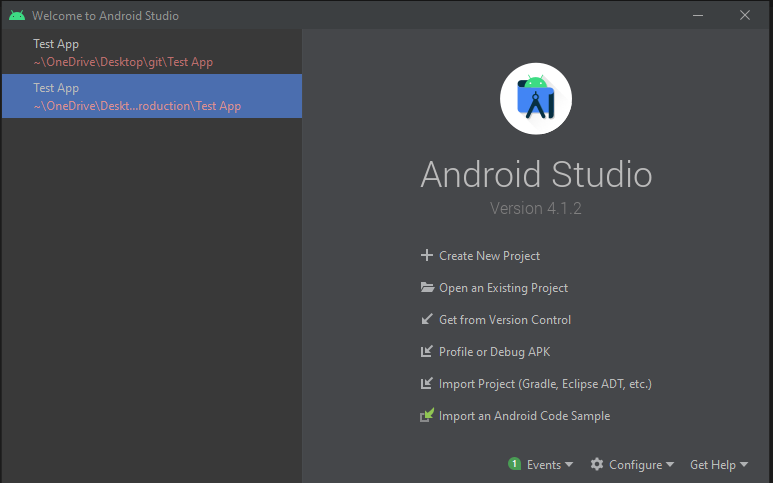
git commit -m “Multiple Screens Task 02”

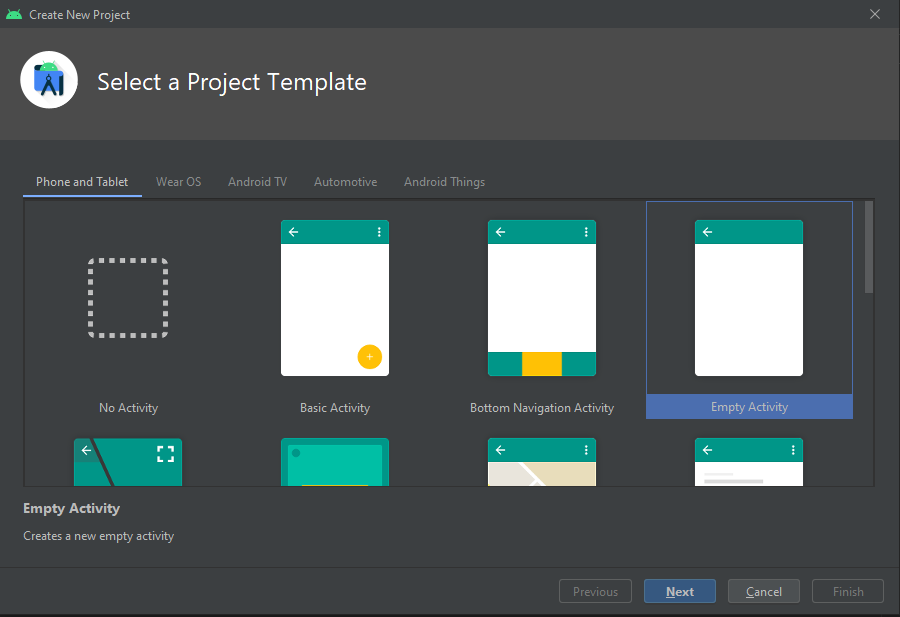
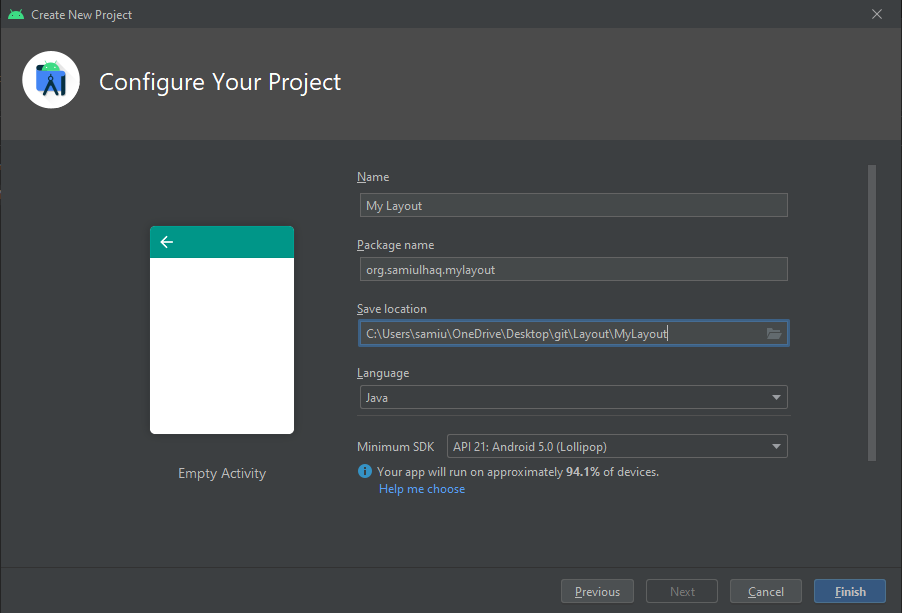
git push

* **Step 07:** Now Project added on GitHub.

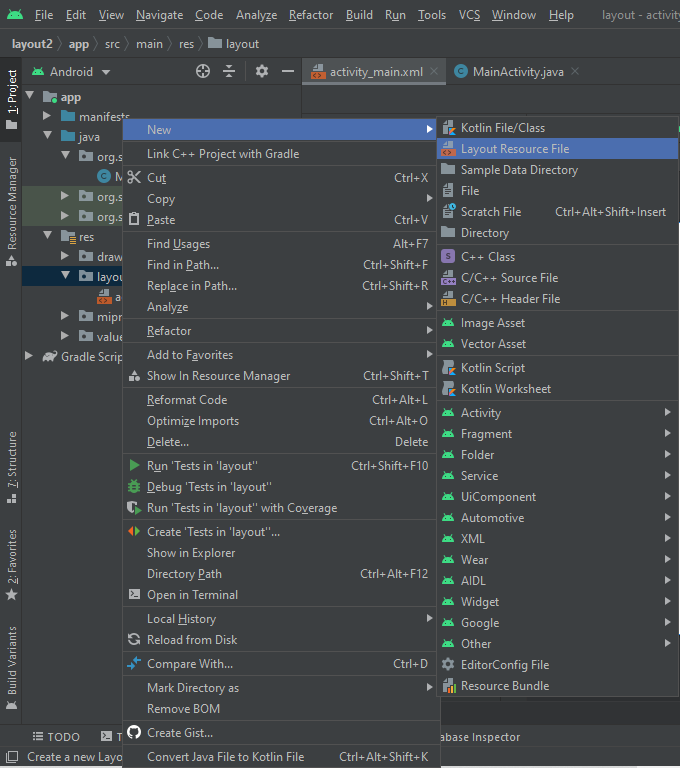
**Lecture – 05**

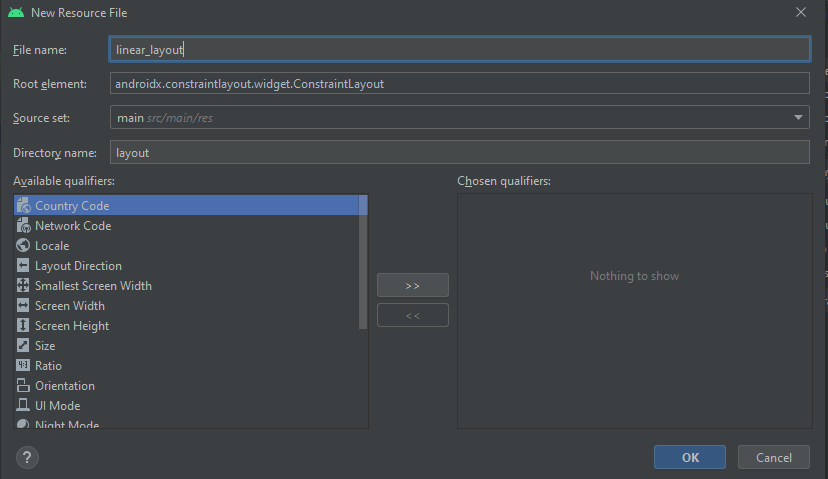
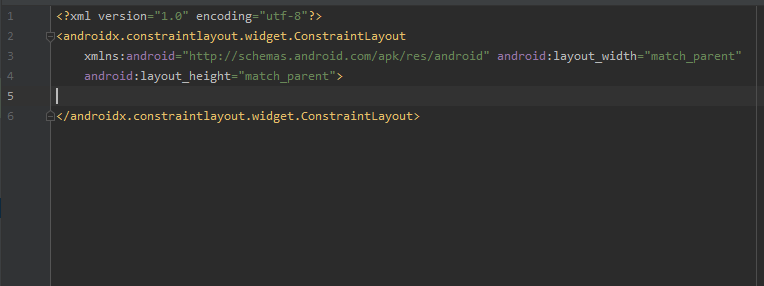
* **Make another layout File**
* **Step 01:** Firstly, we can make a project with the name of My Layout.



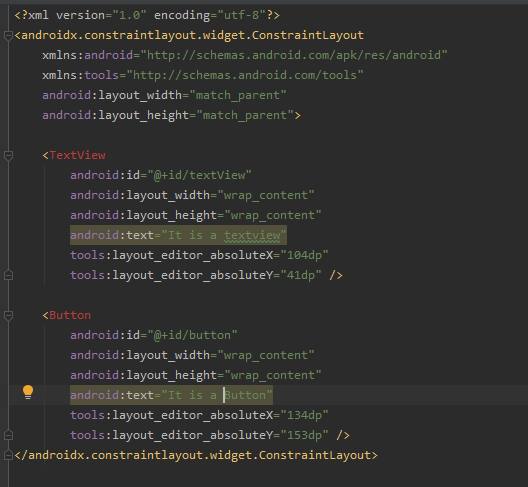
* **Step 02:** On the left side right click on layout folder under resource folder, select New, then Layout Resource File, new window is open, type the name of new xml file. In my case I select linear\_layout. New xml file is created.

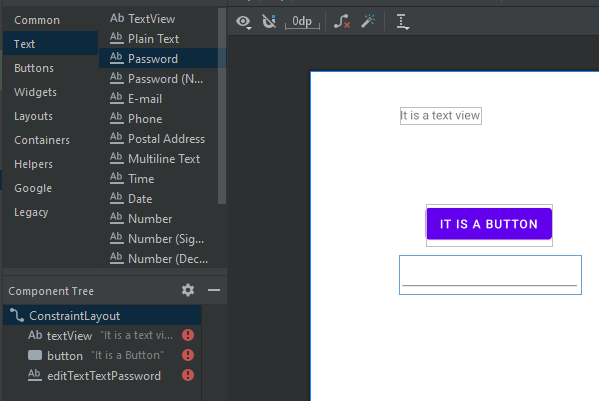


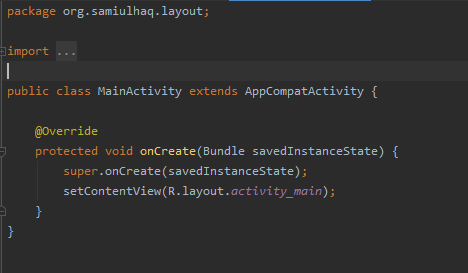
* **Step 03:** Then we make a button, text View and a password.

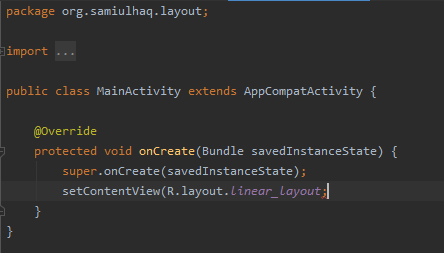


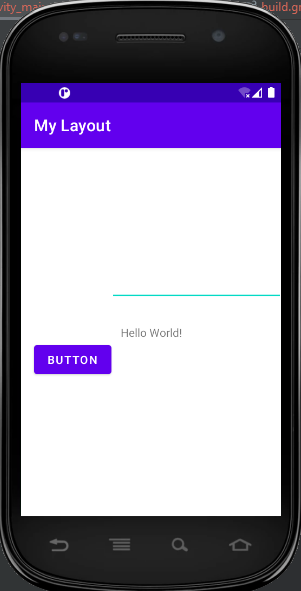




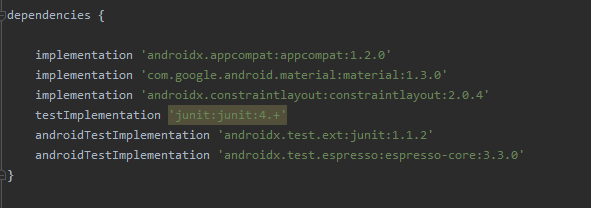
* **Step 04:** By default, in MainActivity.java contentView is set on activity\_main. To run the linear\_layout file changes the contentView activity\_main to linear\_layout.



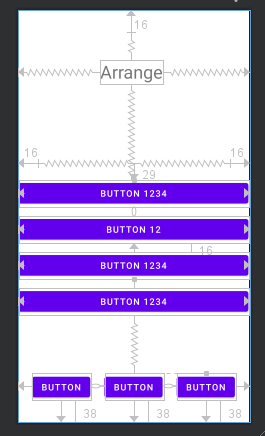


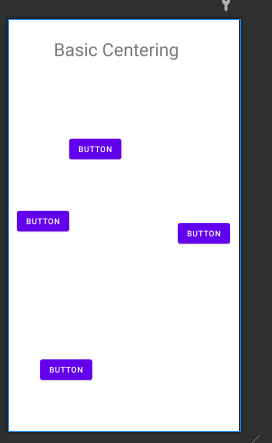
****

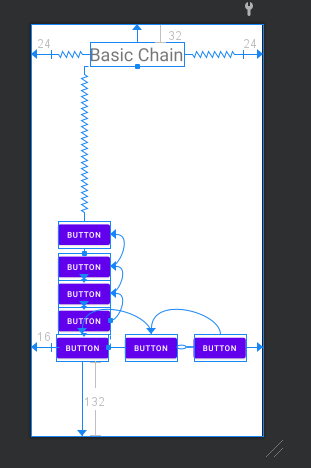
* **Required Dependency**
* **Step 01:** All dependencies are present in build.gradle folder under Gradle Scripts.
* **Step 02:** Add the new dependencies, in this folder.

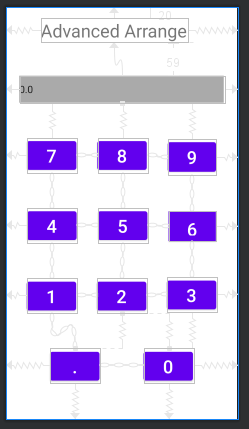


* **Constraint layout**
* **Step 01:** I made the following file for constraint layout.

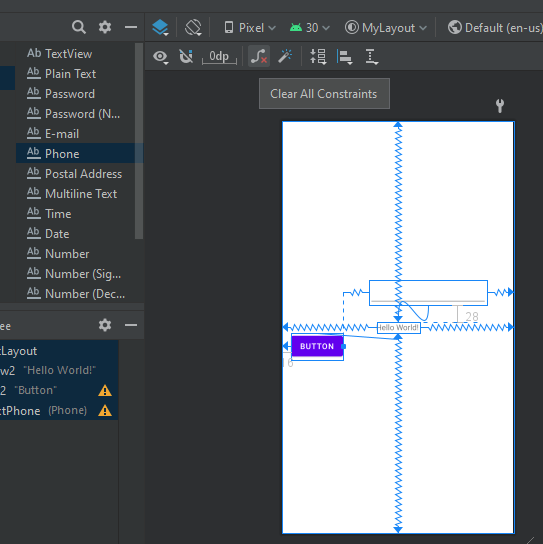


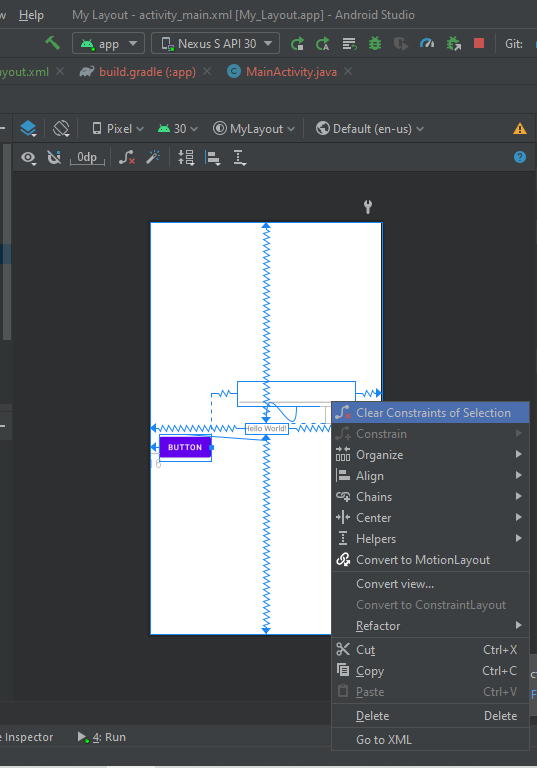




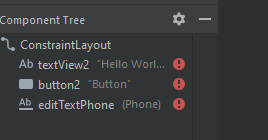


* **Step 02:** Manage constraint, horizontal layout, vertical layout, right align and left align etc.
* **Clear all Constraints**
* **Step 01:** clear all constraints by upper toolbar icon or right click on selected widgets and select the option of clear all constraints.



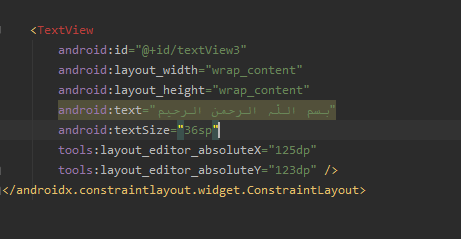


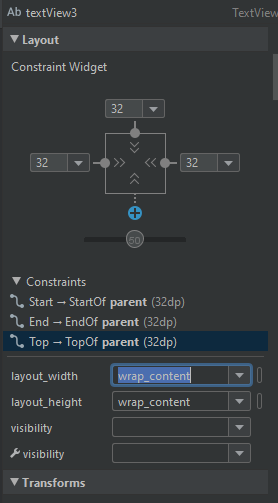
* **Step 02:** Then we see all constraints are clear and message shown in Constraint’s layout.





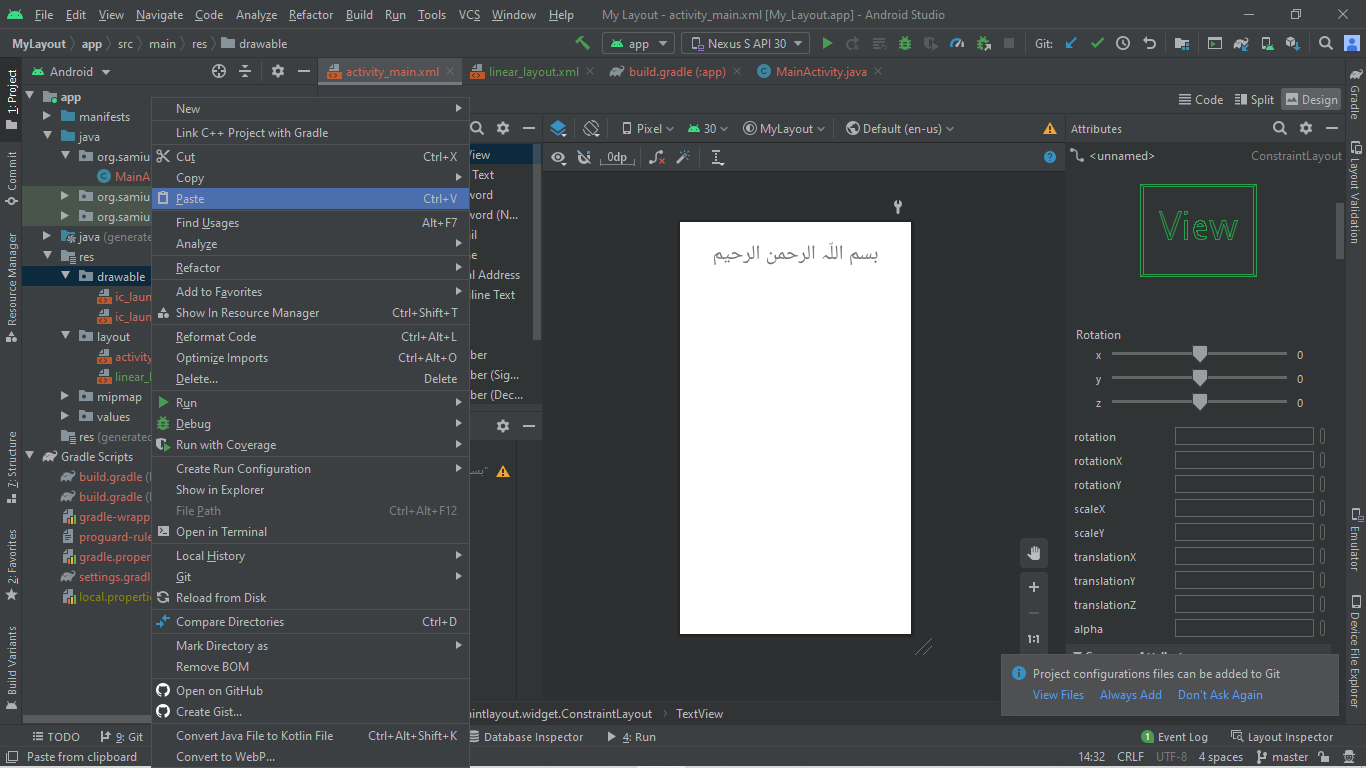
* **Layout Constraints with image**
* **Step 01:** We draw the text view and write the text بسم اللّہ الرحمن الرحیم Then, adjust the layout width and layout height. By default, it is selected wrap\_content.

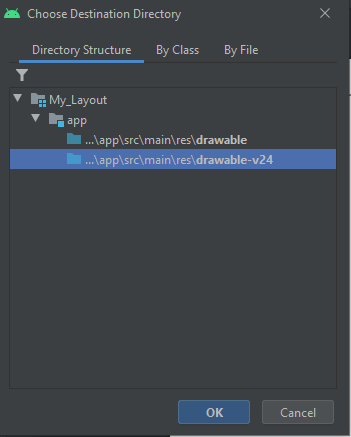


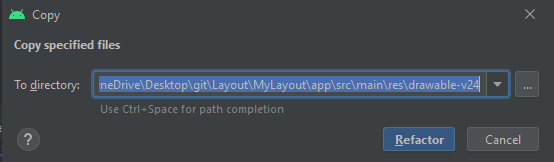


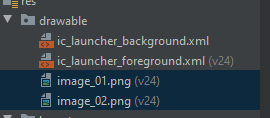
* **Step 02:** To add the images we copy the images and paste in drawable folder. Then add image view, new window is open you can see your images under project folder. Select it and your image is shown on screen.

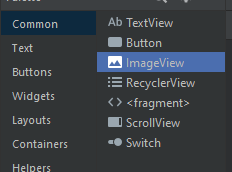


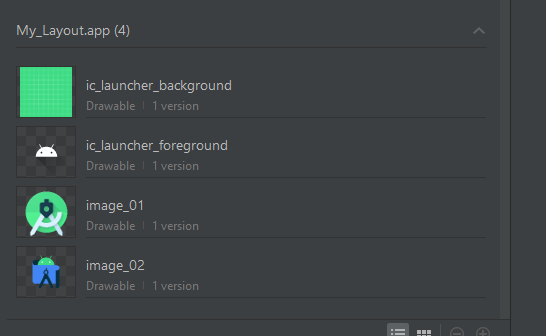






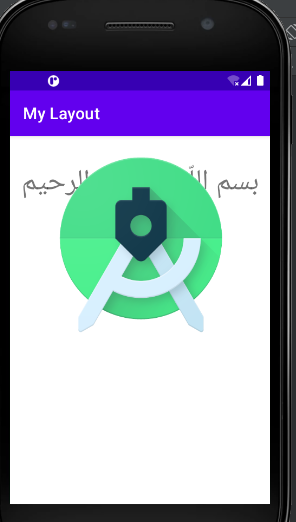




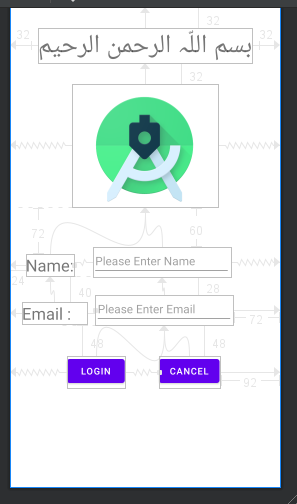




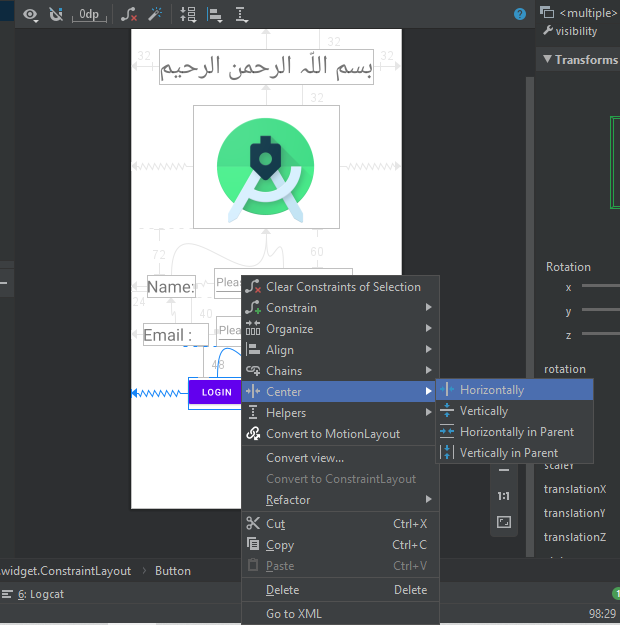
* **Step 03:** When we run the application, it moves on (0,0) axis because we cannot set the constraints of this pitcher.

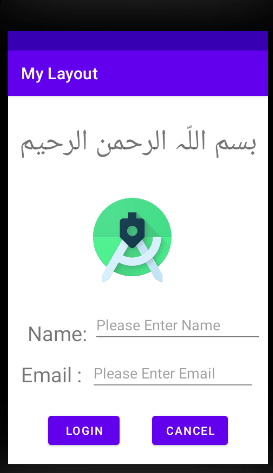


* **Step 04:** We set the constraint of this pitcher then we draw two text View of Name and Email, two buttons login and cancel and Email text View and Name Text View for enter the name and email.



* **Step 05:** We set the login and cancel button horizontal alignment. Then select the show baseline option for in same line widgets.

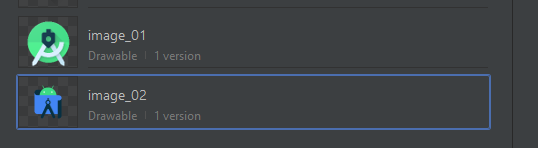




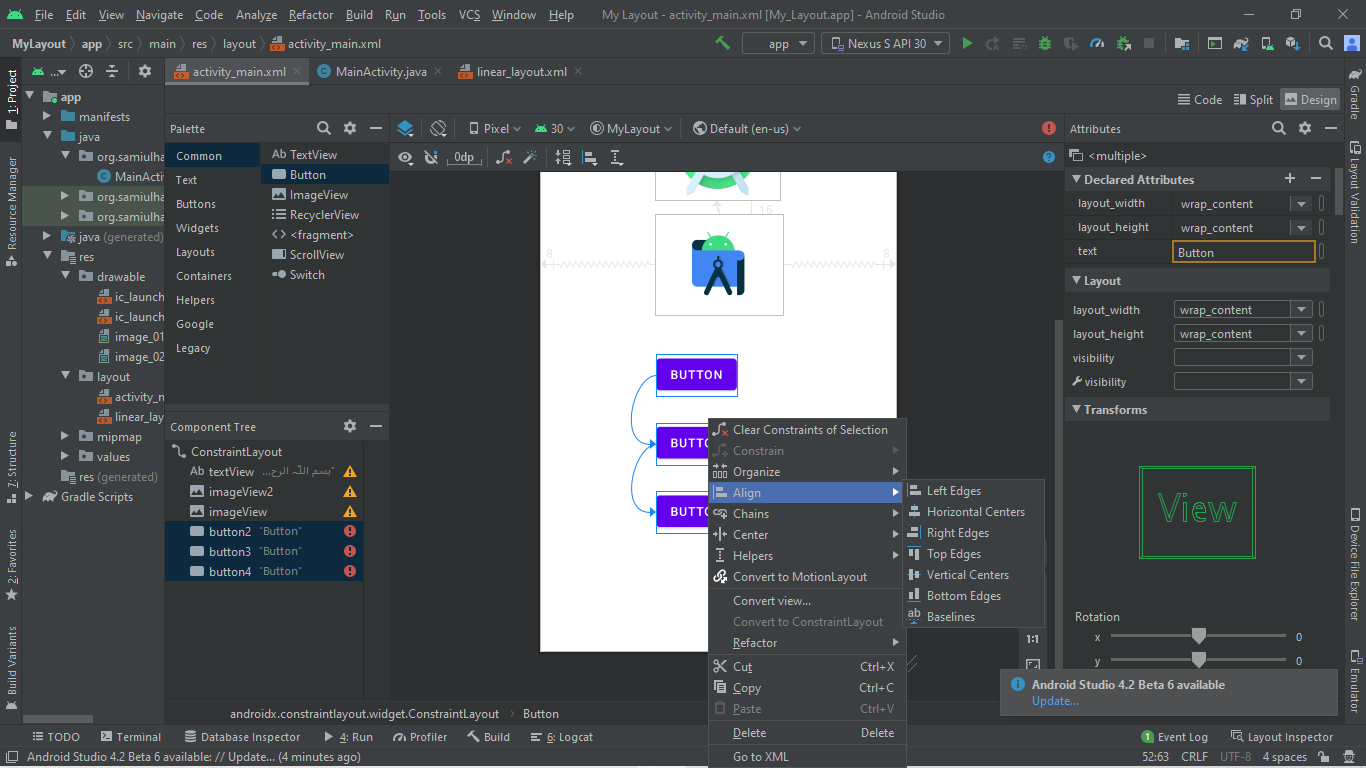




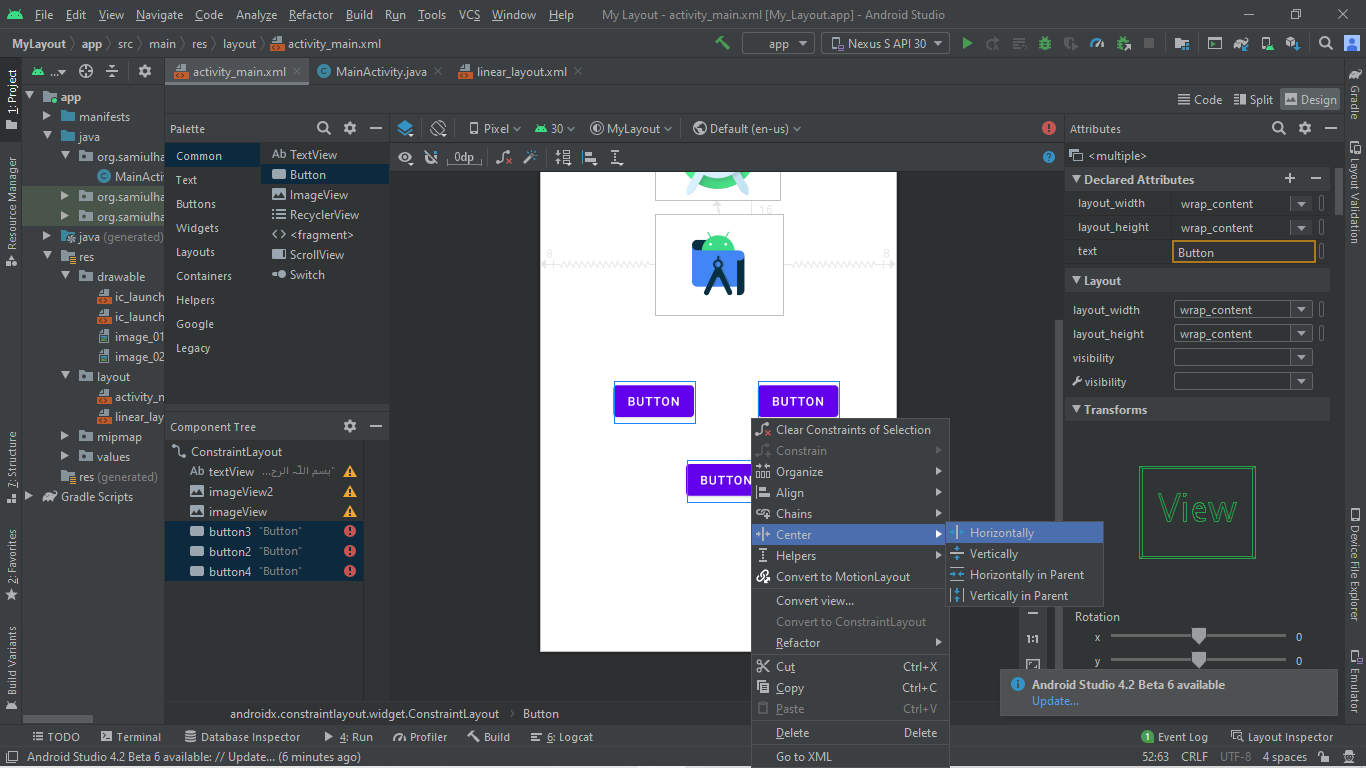
* **Step 06:** We can add the 2nd image with same procedure discuss above. And we make three buttons, then we apply constraints level operations on it.

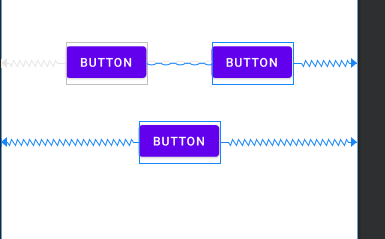


* **Step 07:** We Align these buttons by right click on selected buttons and select the option of left align or right align, in my case I select the left alignment.

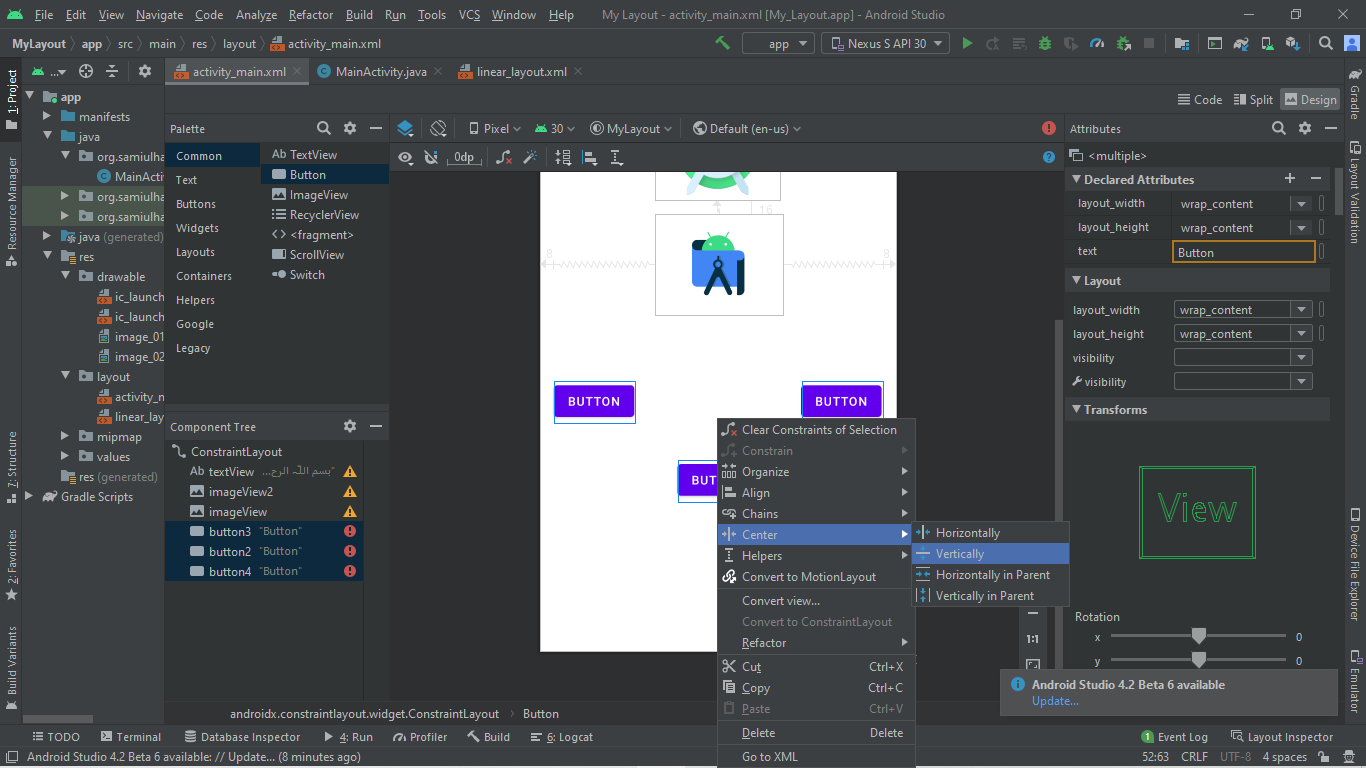


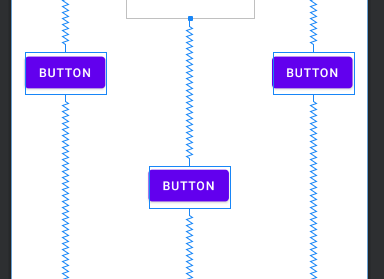
* **Step 08:** We set the buttons center Horizontally.



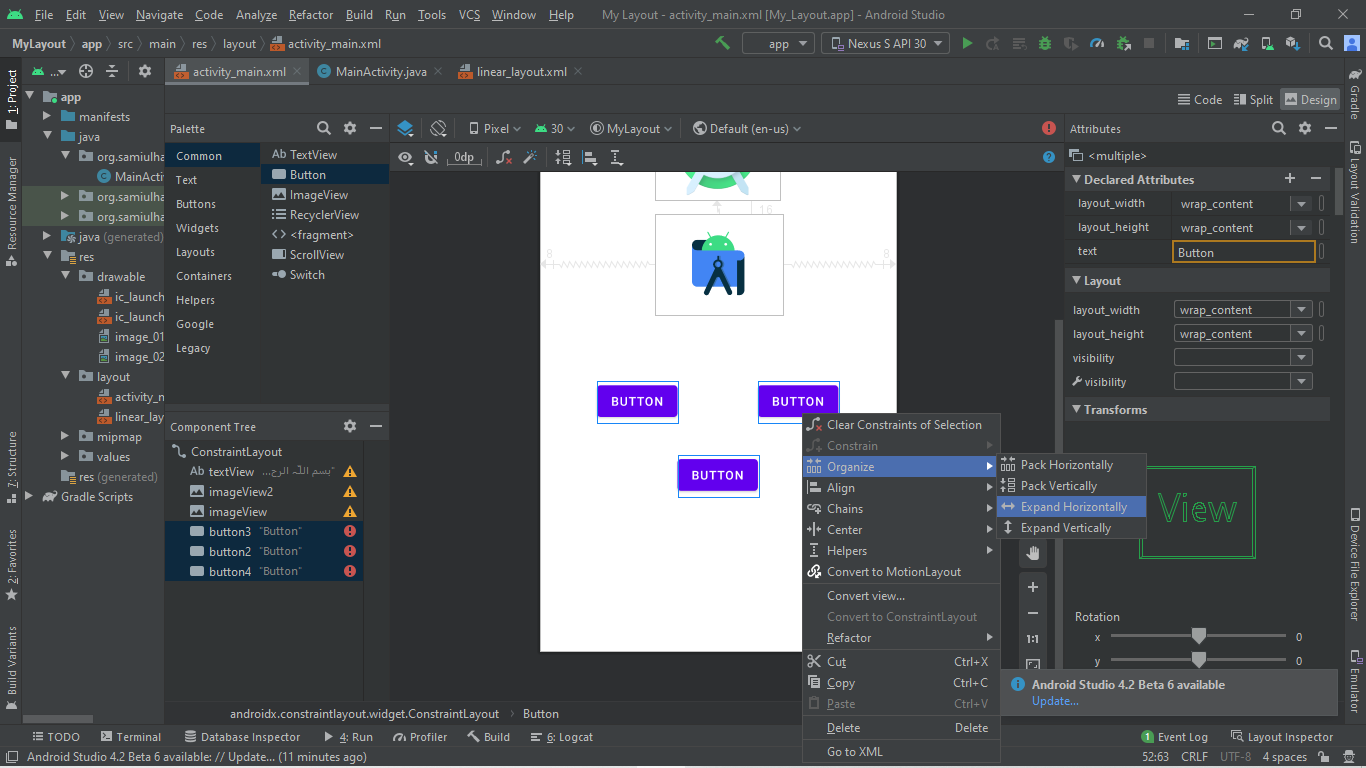


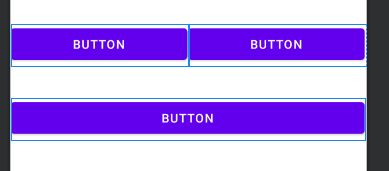
* **Step 09:** We set the buttons center Vertically.



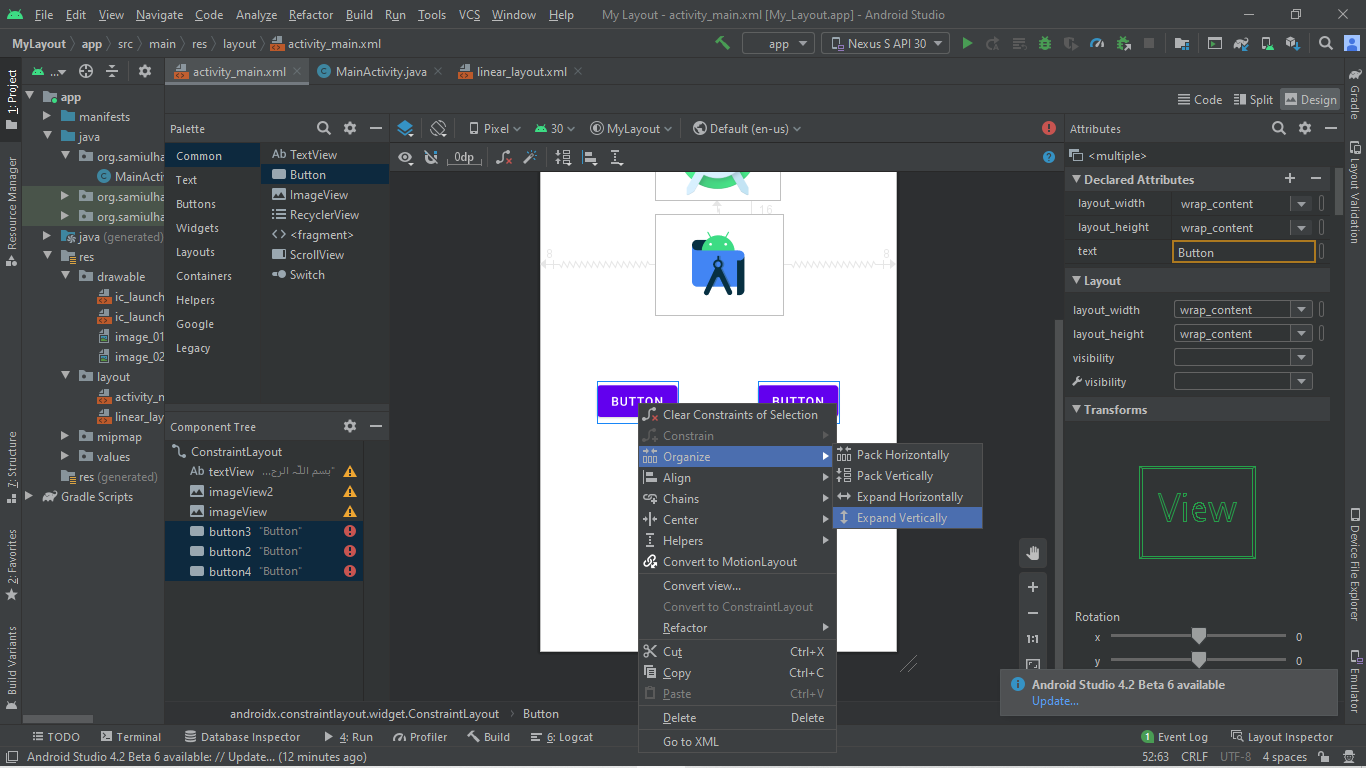


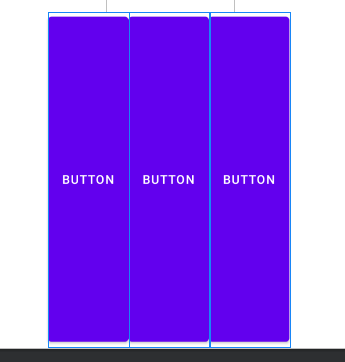
* **Step 10:** We set the buttons Expand Horizontally.



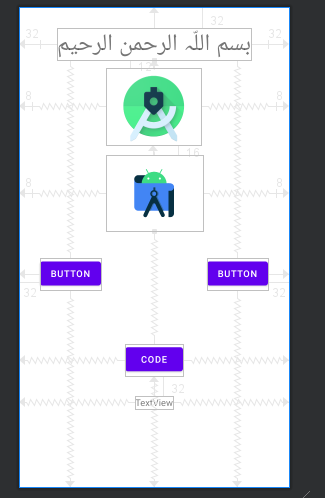


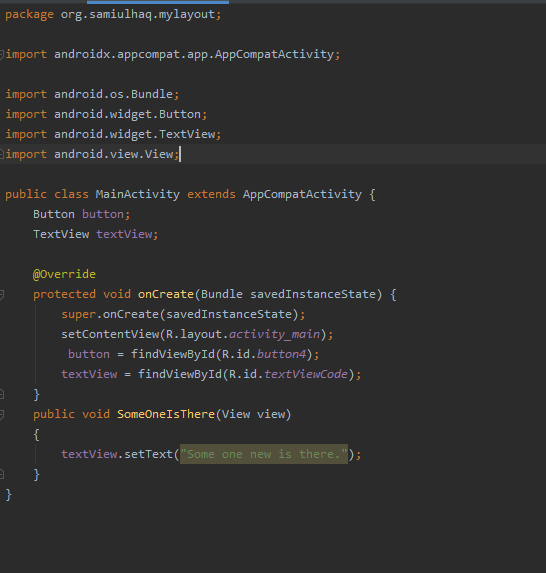
* **Step 11:** We set the buttons Expand Vertically.





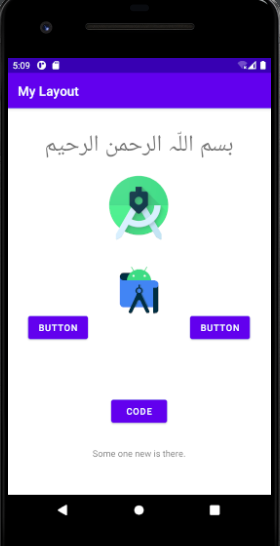
* **Load XML Resources**
* **Step 01:** We give the specific id to all constraints. In my case I select the code Button id buuton4 and TextView is textViewCode. Then we write the code in xml file and we make a function on button click SomeOneIsThere() and write something in it. In my case I write some one is there. And import all required libraries in xml file which can be added by compiler intelligence.





* **Step 02:** Firstly, we click on Code Button, then new Text is shown is case of present text.





* **Step 03:** Now add this project on GitHub by following commands on cmd:

git add “MyLayout”

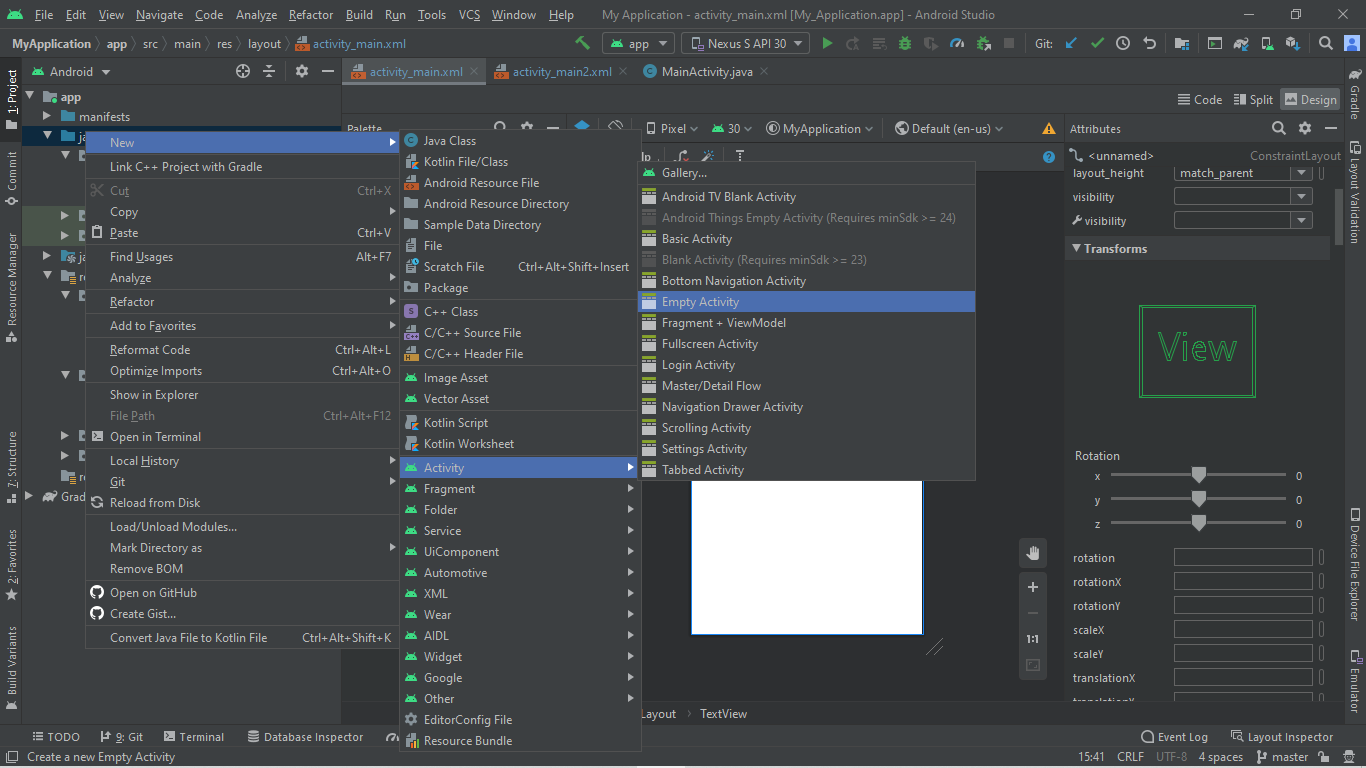
git commit -m “BSEF18A028”

git push

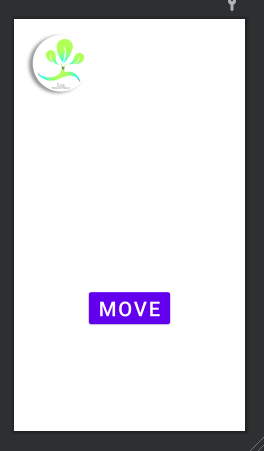
* **Step 04:** Now Project added on GitHub.

**Lecture – 06**

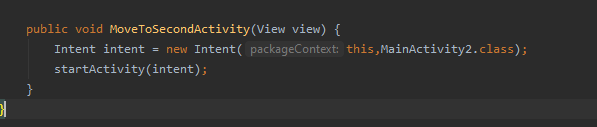
* **Move to next Activity and Phone bar and Website is open**
* **Step 01:** Firstly, we make a project, then we make a new activity with name of MainActivity2.



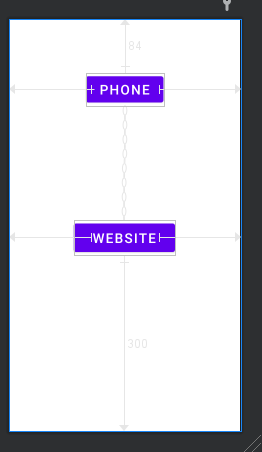
* **Step 02:** We draw the image icon and a button with name Move.

****

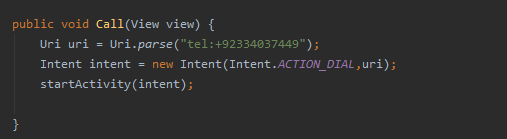
* **Step 03:** On click of Move button we call the function of MoveToSecondActivity.

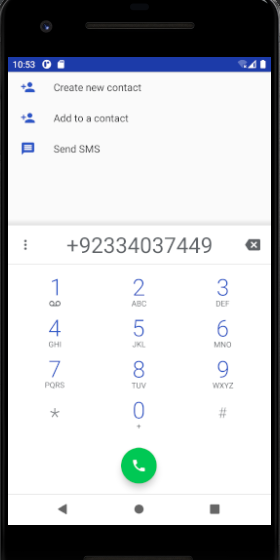
****

* **Step 04:** On MainActivity2 we make two Button with the name of Phone and Website.

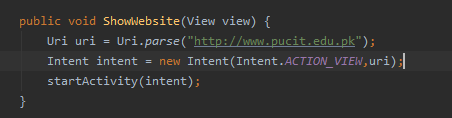
****

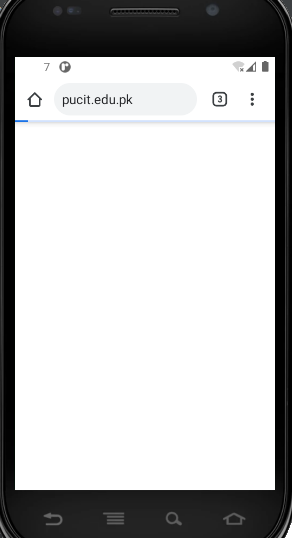
* **Step 05:** On click of Phone Button we call the function of Call, in which phone dial bar is open.

****

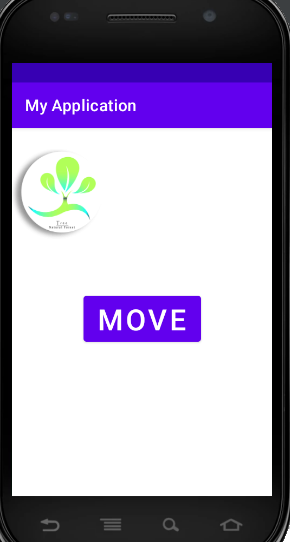
****

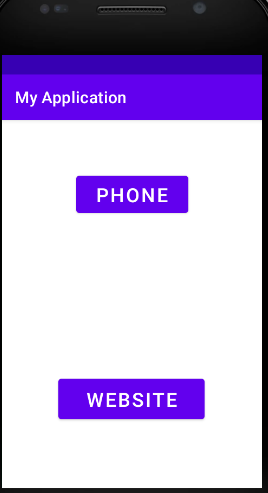
* **Step 06:** On click of Website Button we call the function of ShowWebsite, in which website is open.





* **Step 07:** We run the project and output is shown on mobile.

****

****

* **Step 08:** Now add this project on GitHub by following commands on cmd:

git add “MyApplication”

git commit -m “BSEF18A028\_Sami-Ul-Haq”

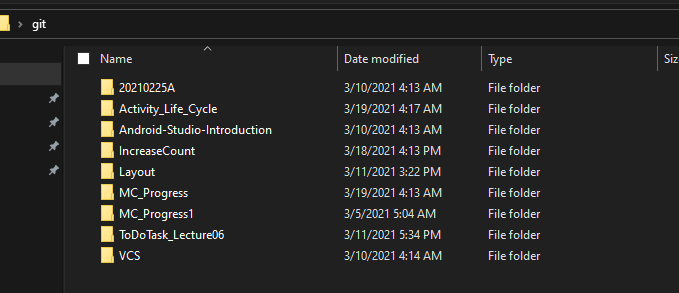
git push

* **Step 09:** Now Project added on GitHub.

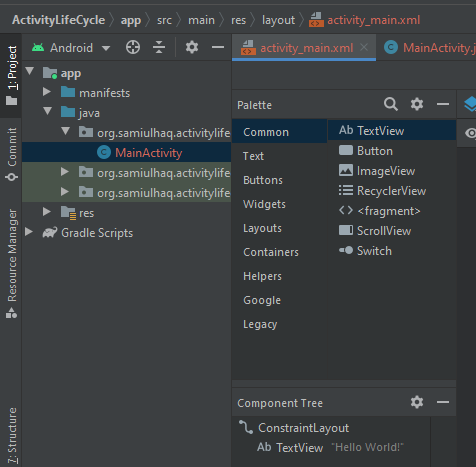


**Lecture – 07**

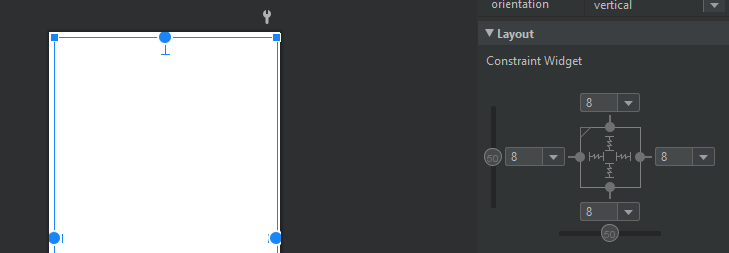
* **Activity Life Cycle**
* **Step 01:** Firstly, we make the repository on git hub with name of Activity\_Life\_Cycle then makes a clone of this repository in git folder which is on Desktop. Then in this clone we make a project with name of ActivityLifeCycle.

****

* **Step 02:** Our Project is created.

****

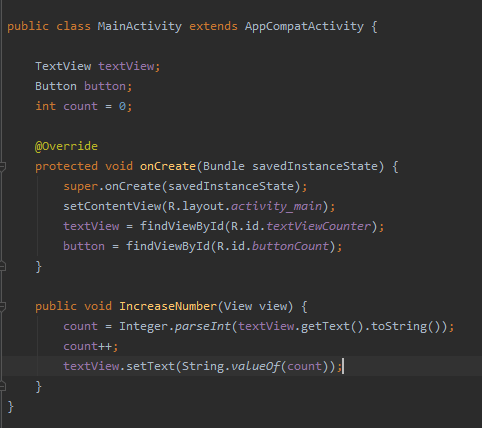
* **Step 03:** We delete the default TextView and Create a new vertical linear layout the padding of 8 from all sides.

****

* **Step 04:** We makes the two TextView and one button in this layout.

****

* **Step 05:** We adjust the size, text color and background color of both TextView and button.
* **Step 06:** Then we make increaseNumber in mainActivity.java function that increase the counter value after clicking the button.

****

* **Step 07:** Then we run the project to check the increase count.

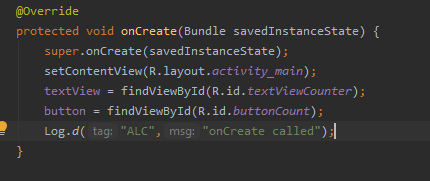
****

* **Step 08:** When we rotate the screen then counter value set to zero.

****

* **Stages of Activity Life Cycle**
* **Step 01:** To see the stages of Activity Life Cycle in logcat we write the following command in onCreate function.

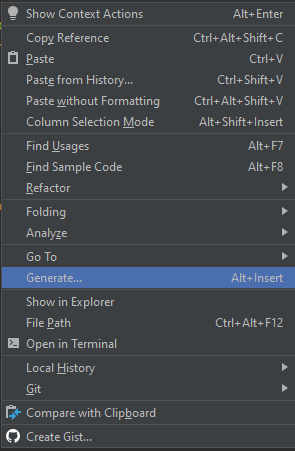
Log.d("ALC", "onCreate called");

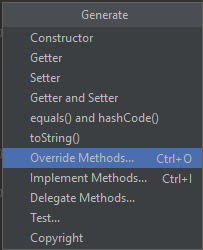
****

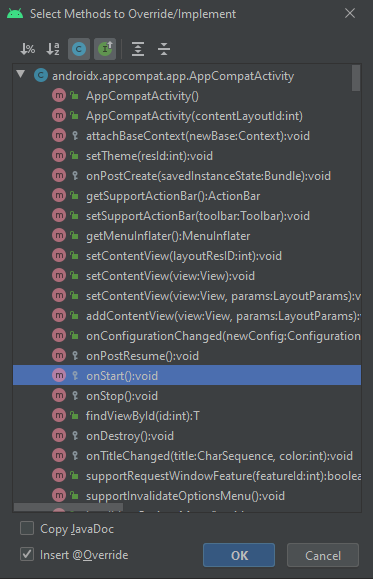
* **Step 02:** Then we can see that on create stage is executed in logout.

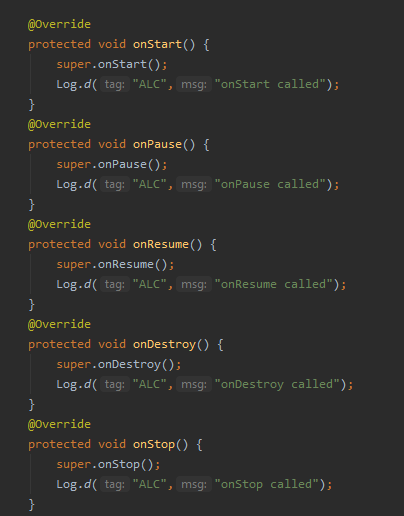
****

* **Step 03:** Now we can write all stages in main\_activity.java. There are two ways to write the stages. Firstly, copy paste of create and then change the name for all others, second is right click then go to generate then Override Methods then new window is open select one of them which you need to be add.

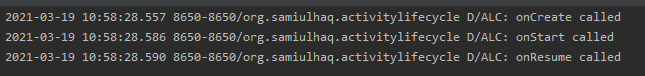
****

****

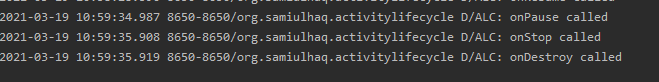
****

****

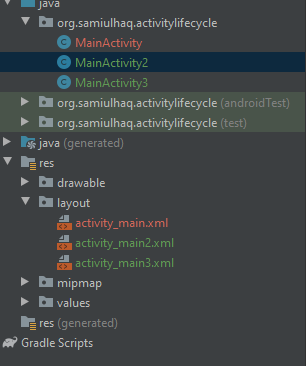
* **Step 04:** Now we run the project and check the logout status.
* **Step 05:** When app is start running onCreate, onStart and onResume message occurs.

****

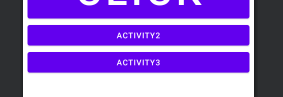
* **Step 06:** When we press back button then onPause, onStop and onDestroy message occurs.

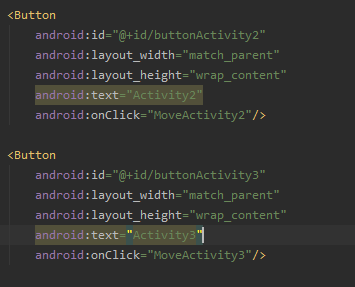
****

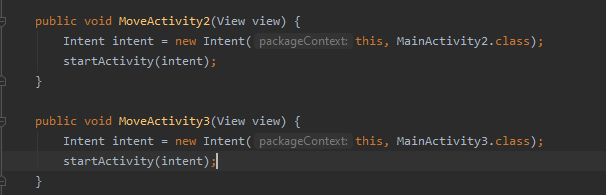
* **Step 07:** Now we create two more Activities with name of mainActivity2 and mainActivity3 then check this behavior.

****

* **Step 08:** We make two buttons to move to mainActivity2 and mainActivity3 then write their function in java file.

****

****

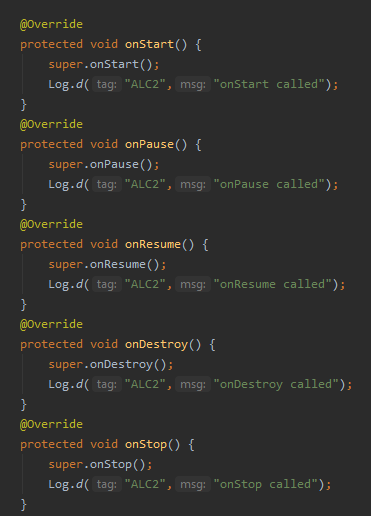
****

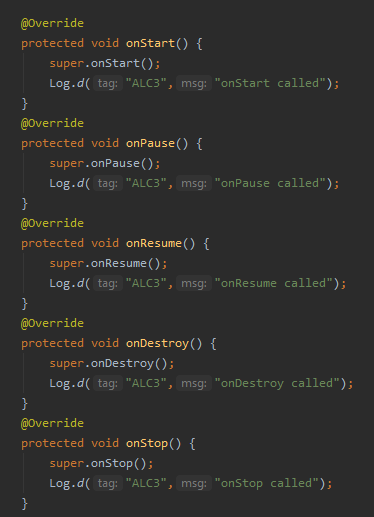
* **Step 09:** Then we add the two textView, one in mainActivity2 and other one in mainActivity3.

****

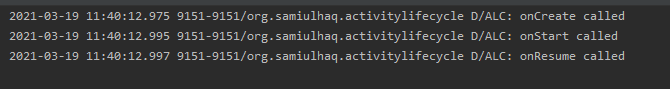
****

* **Step 10:** Then add all override methods in both activities and set their names according to activity.

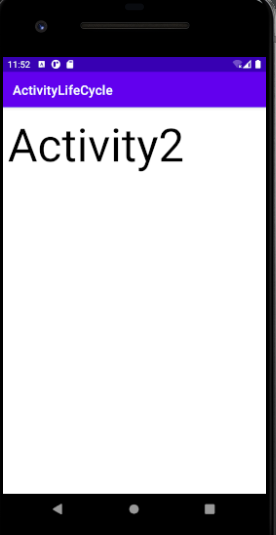
****

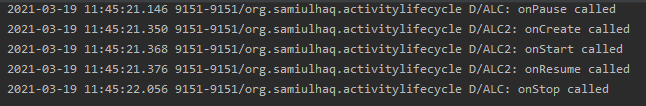
****

* **Step 11:** Now we run the project and check the logout status again.
* **Step 12:** When app start then logcat shows this behavior.

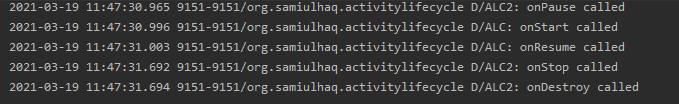
****

When we move to Activity2 then it can show this behavior.



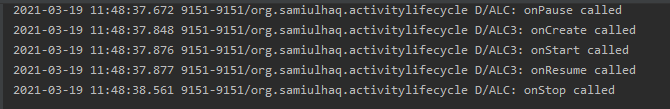


When we press back button then it can show this behavior.

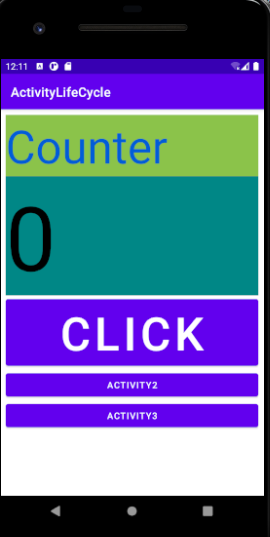


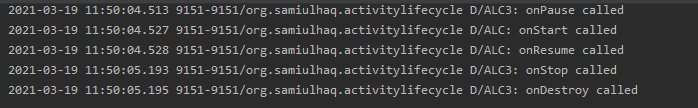
And when we move to activity3 then it can show this behavior.



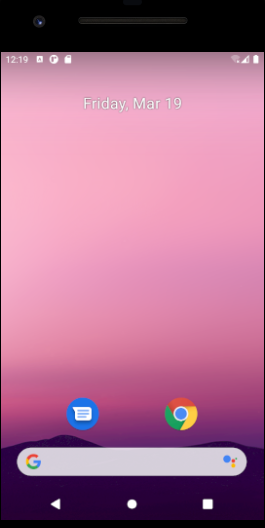


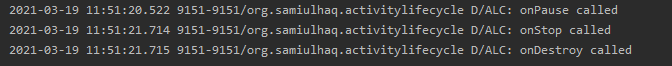
After this when we press back button it can show this behavior.



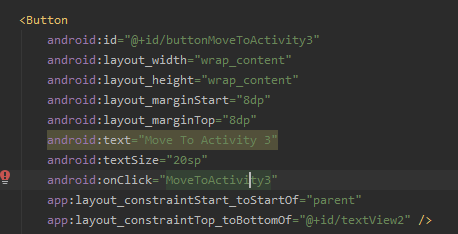


When we press again back button it can show this behavior.

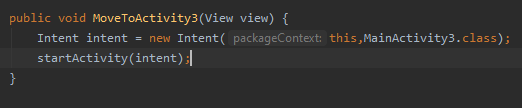


****

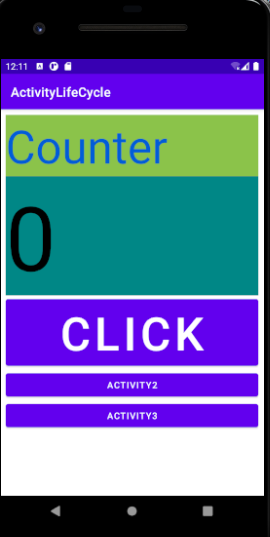
* **Step 13:** After this we add a new button in mainActivity2 and create a onClick function to move to mainActivity3.

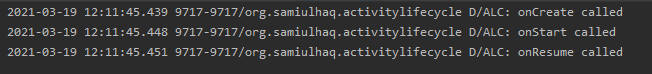
****

****

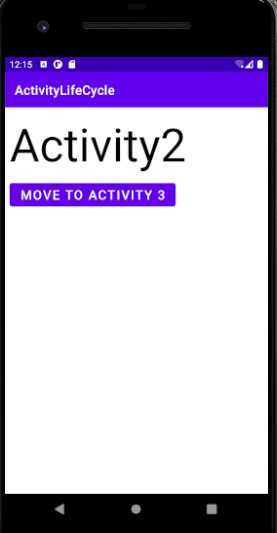
****

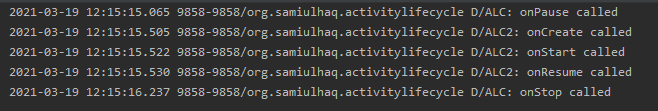
* **Step 14:** Now again run the project and check the logout status.
* **Step 15:** When app start then logcat shows this behavior.

****

****

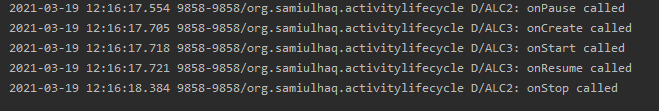
When we move to Activity2 then it can show this behavior.



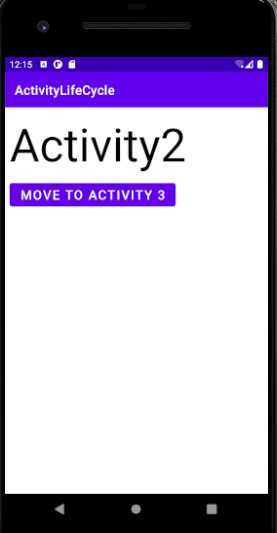


When we press button in Activity2 then it can move to Activity3 and show this behavior.



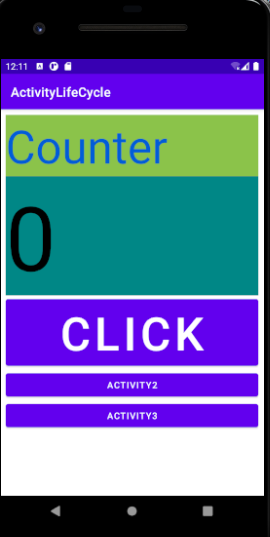


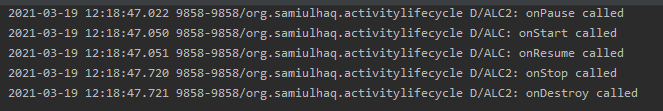
When we press back button, it can move back to Activity2 and show this behavior.



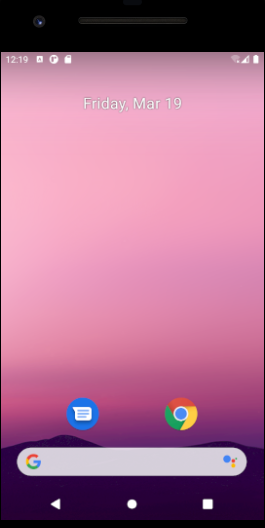


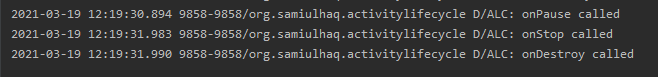
Again, when we press back button, it can move back to Activity1 and show this behavior.



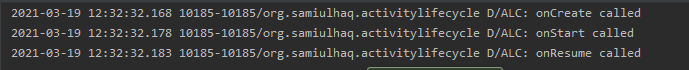


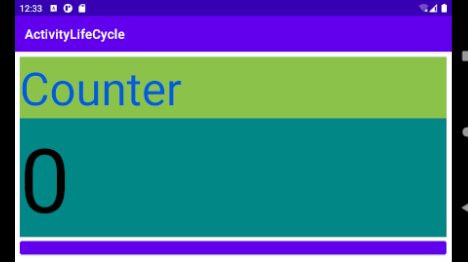
After this when we press back button it can show this behavior.

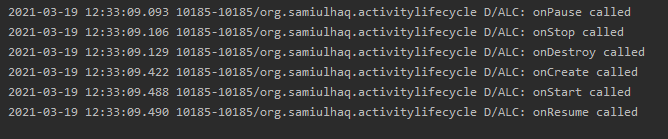




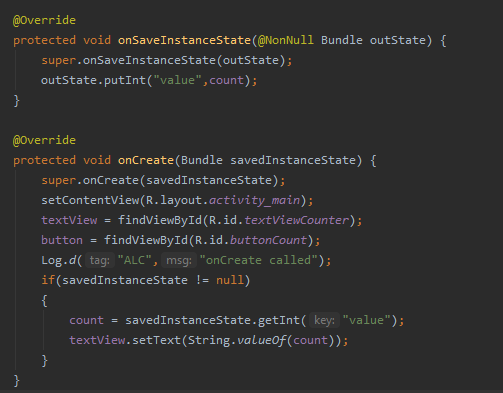
* **Resume Counter Value on Rotate Screen**
* **Step 01:** Firstly, we run the app and check the log status after rotating the screen.







* **Step 02:** Now we can see that on rotation it pause then stop then destroy. It means we need to save the counter value before destroy. For this we search a onSaveInstanceState function in generate override method. We save our counter value in this function which can store in our value variable then we use it in onCreate function and display it.



* **Step 03:** After this we can see that counter value remains same after rotating the screen.





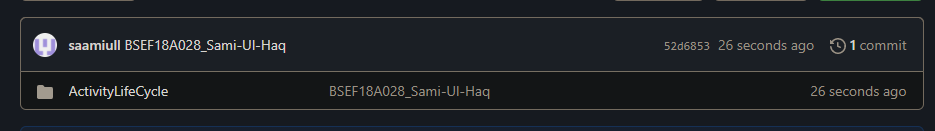
* **Step 04:** Now add this project on GitHub by following commands on cmd:

git add “ActivityLifeCycle”

git commit -m “BSEF18A028\_Sami-Ul-Haq”

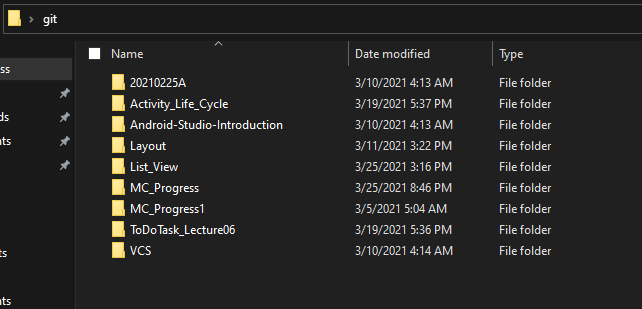
git push

* **Step 05:** Now Project added on GitHub.

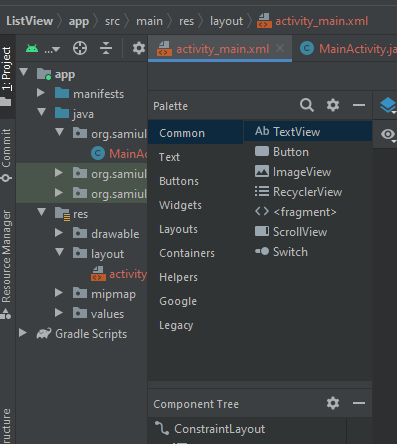


**Lecture – 08**

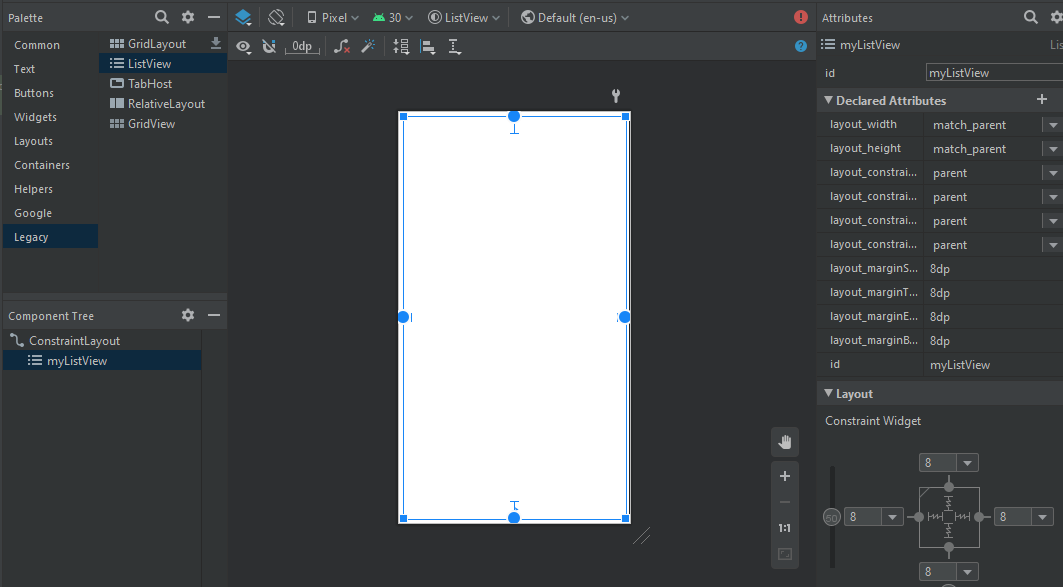
* **List View Binding**
* **Step 01:** Firstly, we make the repository on git hub with name of List\_View then makes a clone of this repository in git folder which is on Desktop. Then in this clone we make a project with name of ListView.

****

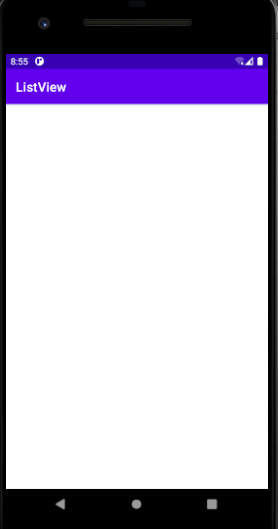
* **Step 02:** Our Project is created.

****

* **Step 03:** We delete the default TextView and Create a new List View under legacy, set its id myListView and attach with all borders with 8 padding.

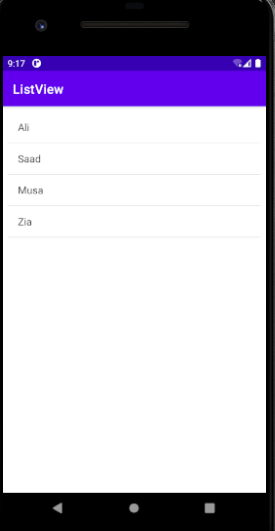
****

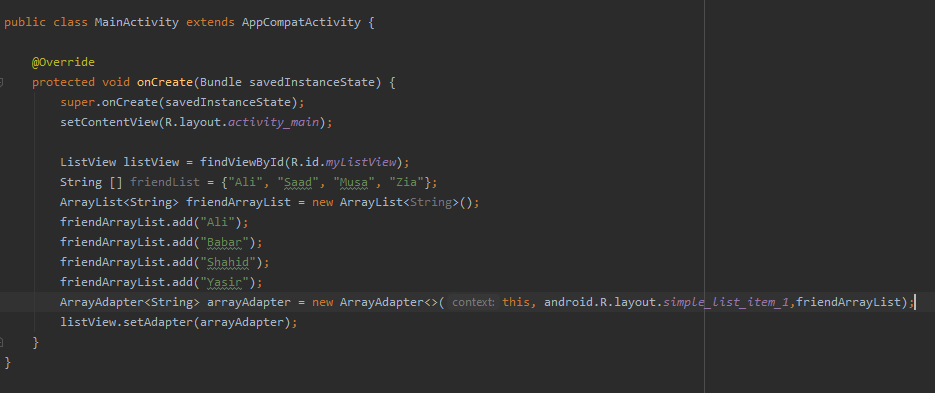
* **Step 04:** Now run the app to check the list view.

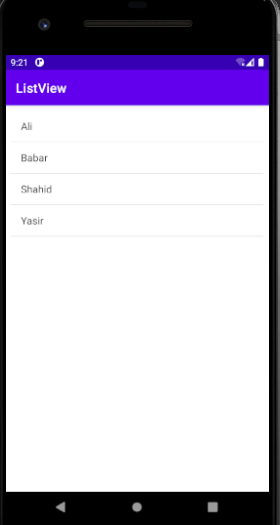
****

* **Step 05:** Then make the listView in MainActivity, make one array and one Array List. Now makes the Array adapter using both array and Array List. At last, set the adapter. Now see the results.

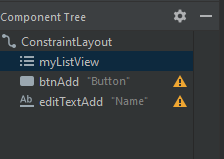
****

****

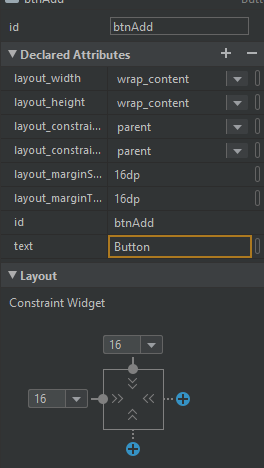
****

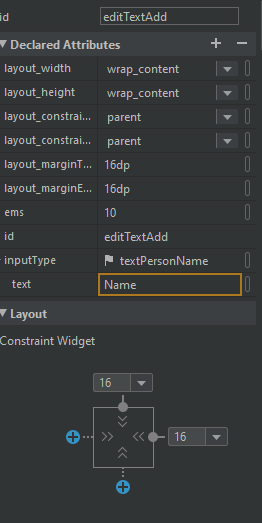
****

* **List View Add Record**
* **Step 01:** We add the new button with id btnAdd and text View with id editTextAdd.

****

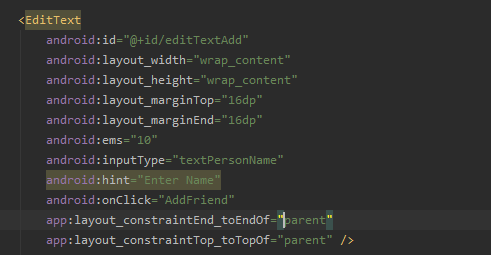
* **Step 02:** Set the left and upper constraints of button with padding of 16 and set the right and upper constraints of text View with padding of 16. Now delete the upper constraint of list view and again set the wrapMatch constraint.

****

****

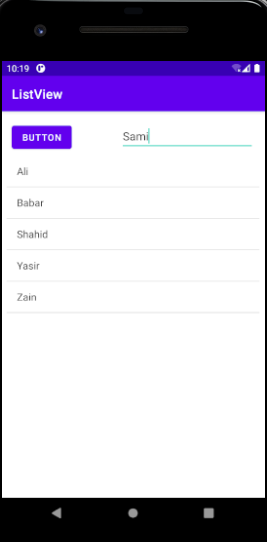
****

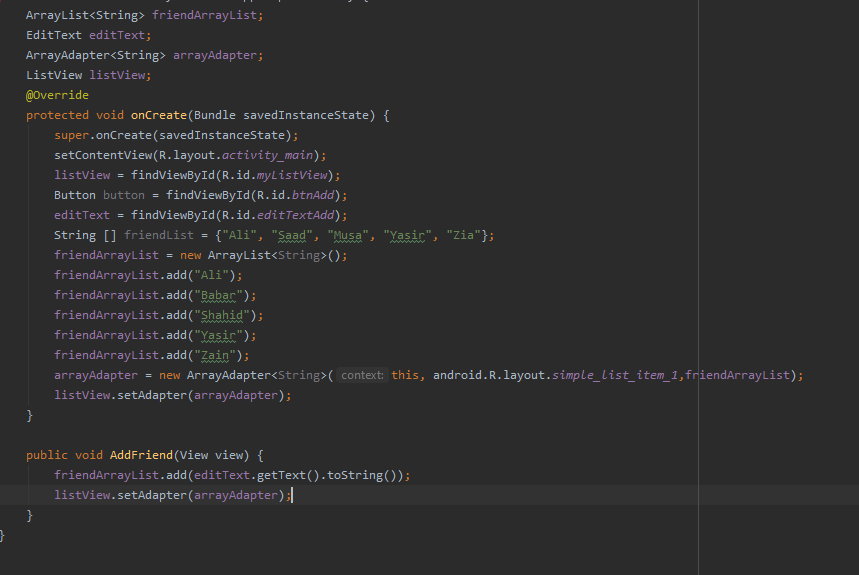
* **Step 03:** Add the Plain Text hint Enter Name and on click function with name of AddFriend and define this function in MainActivity.

****

****

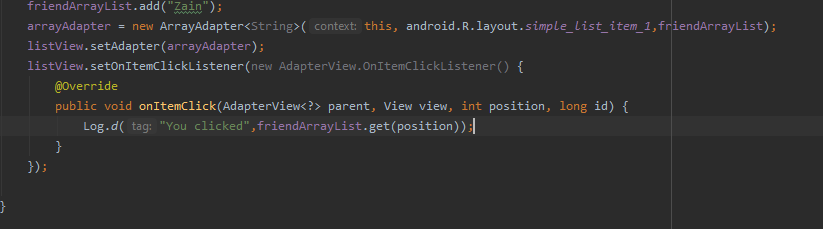
* **Step 04:** This is not working and name is not added in list so we can update the function. We can move listView and ArrayAdapter library definition outside the onCreate function. Now it is working and name is added.

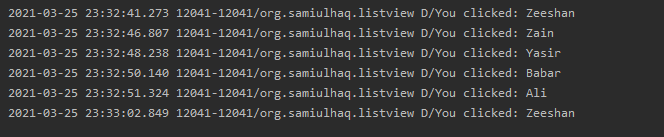
****

****

****

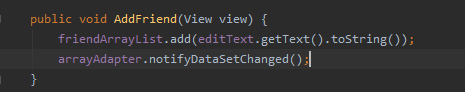
* **listView Item Click Listner**
* **Step 01:** To check the status of name on logcat we write the setOnItemClickListener function in MainActivity.

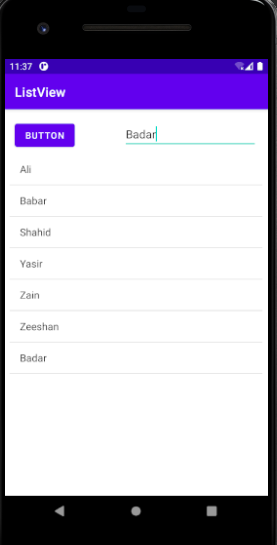
****

****

* **Update of Records**
* **Step 01:** To update the record we write the following command:

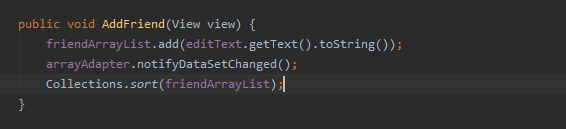
arrayAdapter.notifyDataSetChanged();

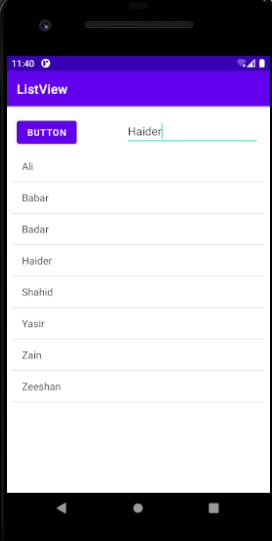
****

****

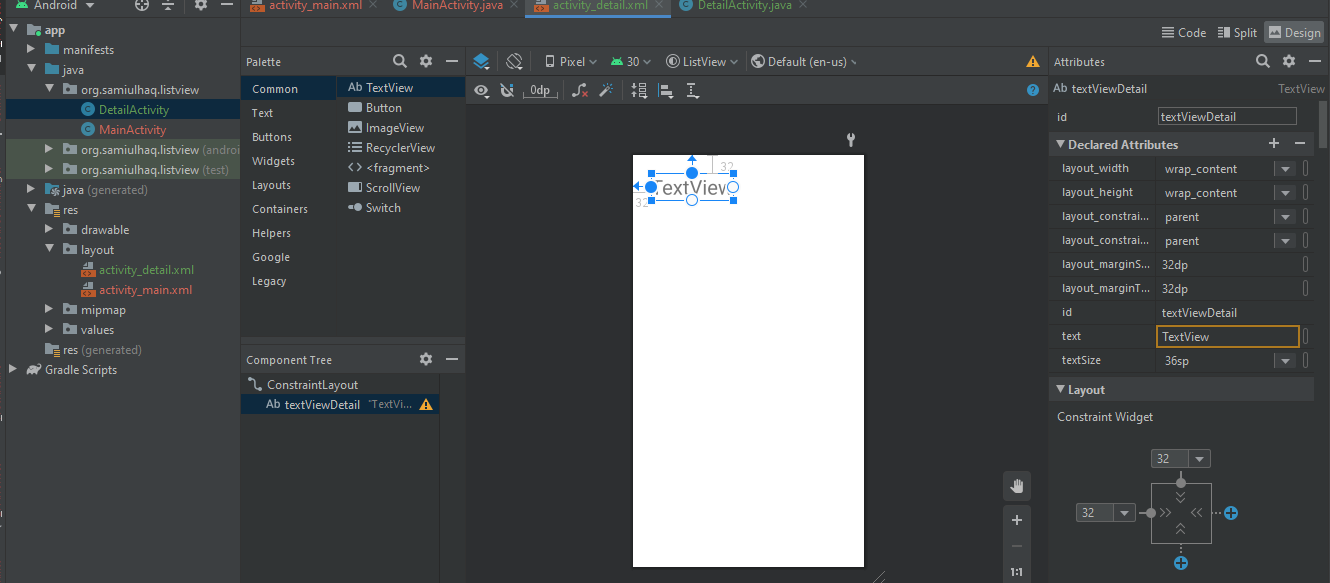
* **Sorting of Records**
* **Step 01:** To sort the record we write the following command:

Collections.sort(friendArrayList);

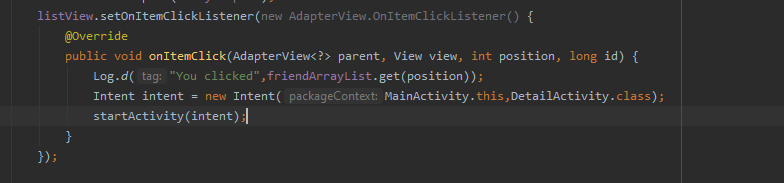
****

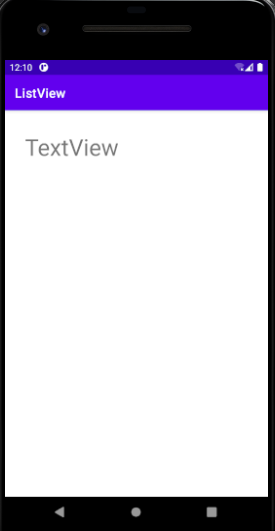
****

* **Move one Activity to another by click on Item**
* **Step 01:** We create a new Empty Activity with name of DetailActivity then add a textView in it, resize by 36sp and add left and upper constraints.

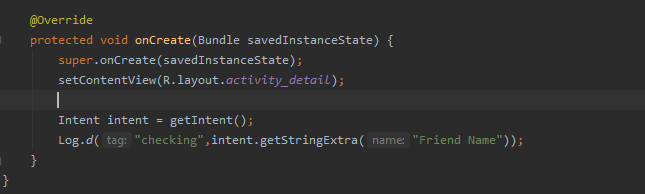
****

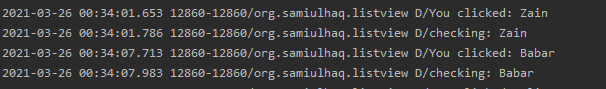
* **Step 02:** Write the intent code for moving one activity to another in setOnItemClickListener function. Then see the result by running the app. Now we move in DetailActivity.

****

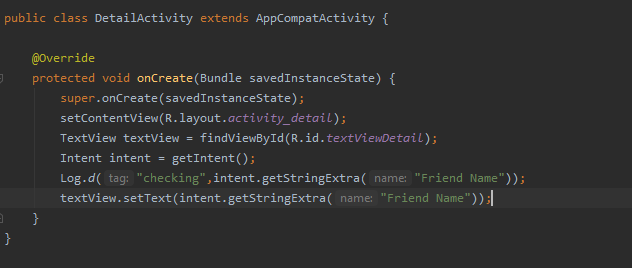
****

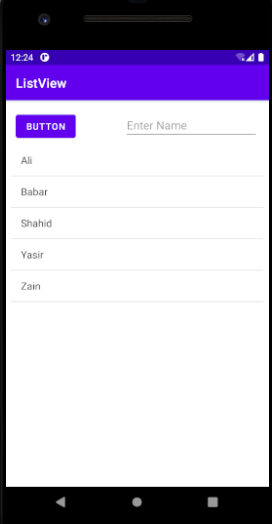
* **Step 03:** We get the intent in DetailActivity and check the status on logcat of clicking the item.

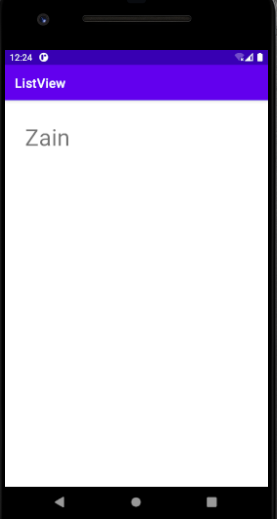
****

****

* **Step 04:** For getting the item name on DetailActivity we get the textView id of textViewDetail and set text add the extra string in DetailActivity.

****

****

****

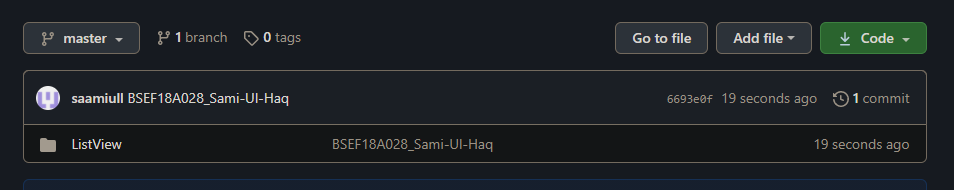
* **Step 05:** Now add this project on GitHub by following commands on cmd:

git add “DetailActivity”

git commit -m “BSEF18A028\_Sami-Ul-Haq”

git push

* **Step 06:** Now Project added on GitHub.

****