MOBILE COMPUTING-PROGRESS FILE

Submitted to:

Sir Haq Nawaz

Submitted by:

Amber Shafique (Bsef18a019)

PUCIT, University of the Punjab, Lahore.

LECTURE#1

❖ VCS (Version Control System):

A **version control system** is a kind of software that helps the developer team to efficiently communicate and manage track of all the changes that have been made to the source code along with the information like who made and what change has been made.

- Version control software keeps track of every modification to the code in a special kind of database.
- Using a VCS also generally means that if you screw things up or lose files, you can easily recover

Benefits:-

1. **Code Synchronizing:** Same code available to all members on different repositories connected to a central repository. All the changes in the files are tracked under the central repository. The central repository includes all the information of versioned files, and list of users that check out files from that central place using **VCS.**

2. **Code Testing :VCS** helps to test changes without losing the original version of the application.

3. **Revert: VCS** allows us to revert back to previous versions of the file because it separately maintains each version of file.

❖ Git :

Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development.

Step-0

1.1 GitHub Account:

Sign up to ***github.com*** and create a new account.

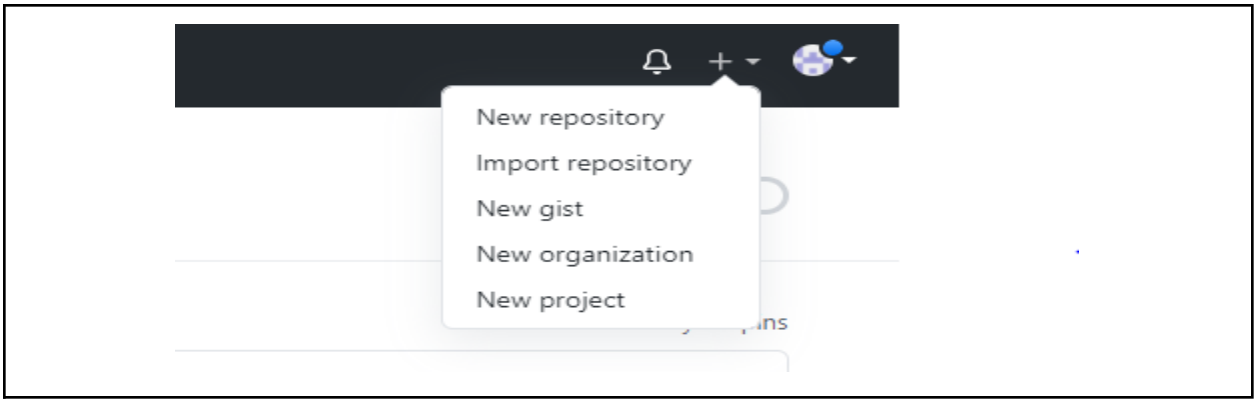
1.2 Git CMD:

Install git cmd and configure it on your PC.

Step-1

1.1 Create a new Repository:

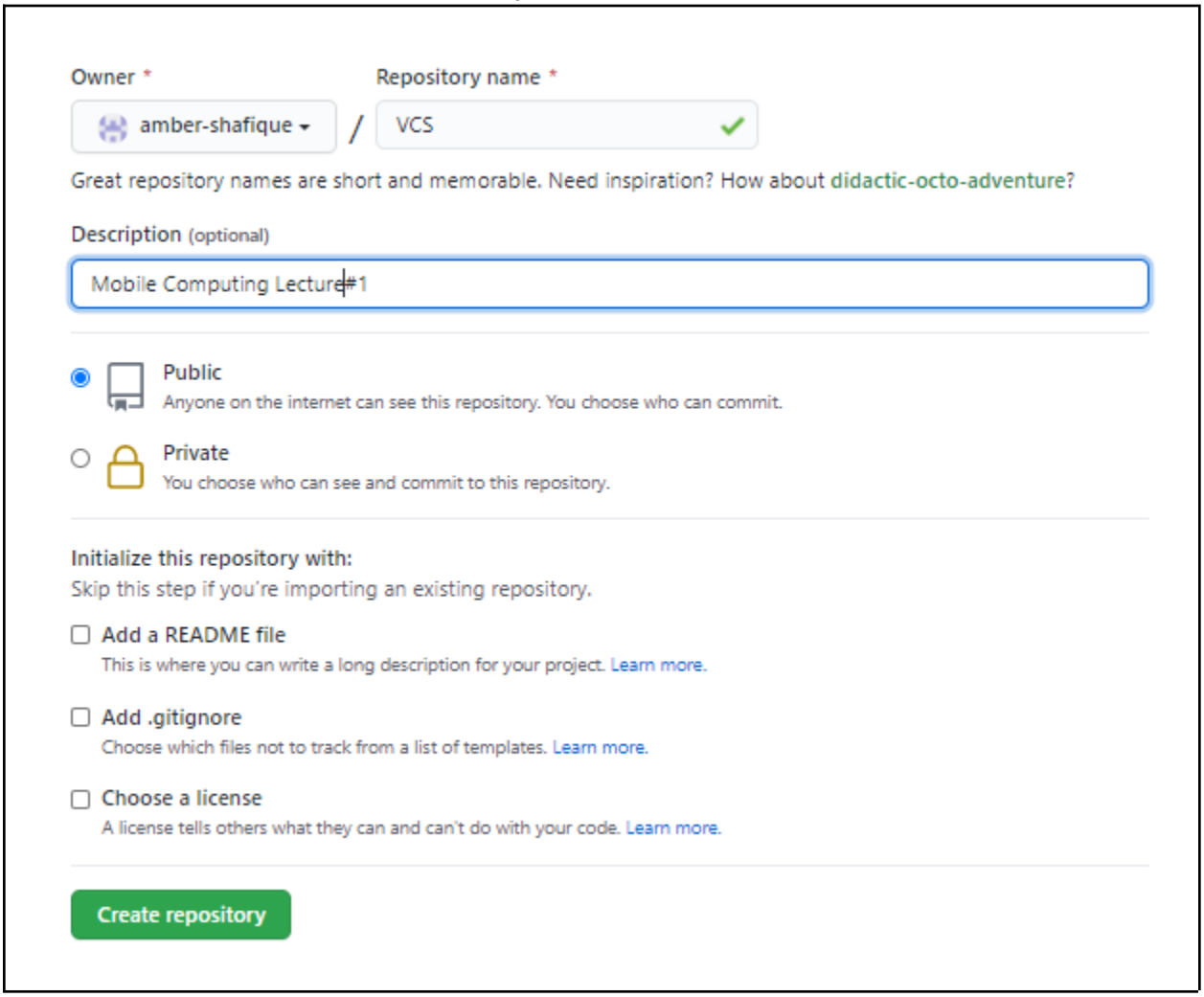
1.1.1 Click on add new repository button on your GitHub account.



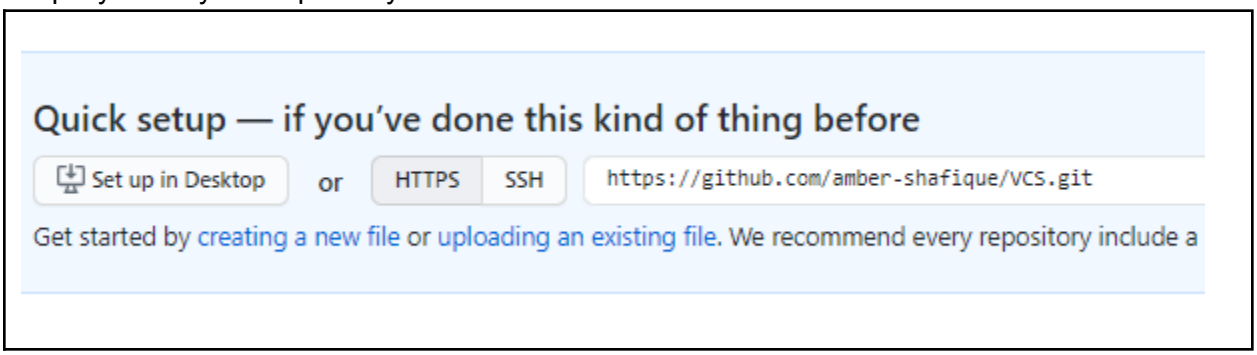
1.1.2 Then provide details of your repository => Name, Description,Type.

Public Repository: It is visible to every user on your GitHub.
Private Repository: It is only available to the repository owner.

1.1.3 Click on the create repository button.



1.2 Repository URL : On creating a new repository GitHub will provide its url that is used to uniquely identify our repository on GitHub.



LECTURE#2

❖ Git Clone:

- It is primarily used to point to an existing repository and make a clone or copy of that repository in a new directory, at another location.
- Makes connection between **local repository(**on PC) and **central repository** (on GitHub).

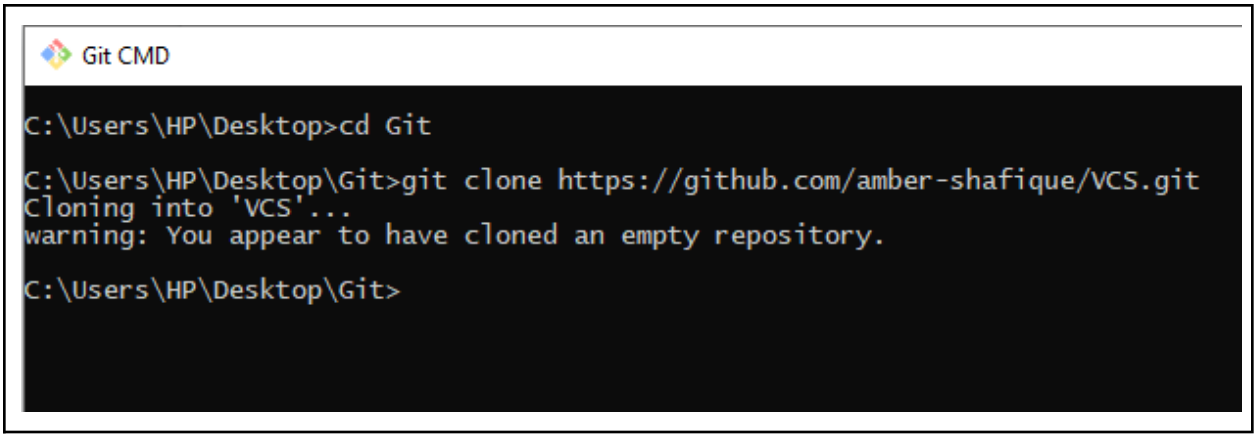
Step-1

Open **git CMD** and go to that folder path (using **cd**) where you want to clone the repository.

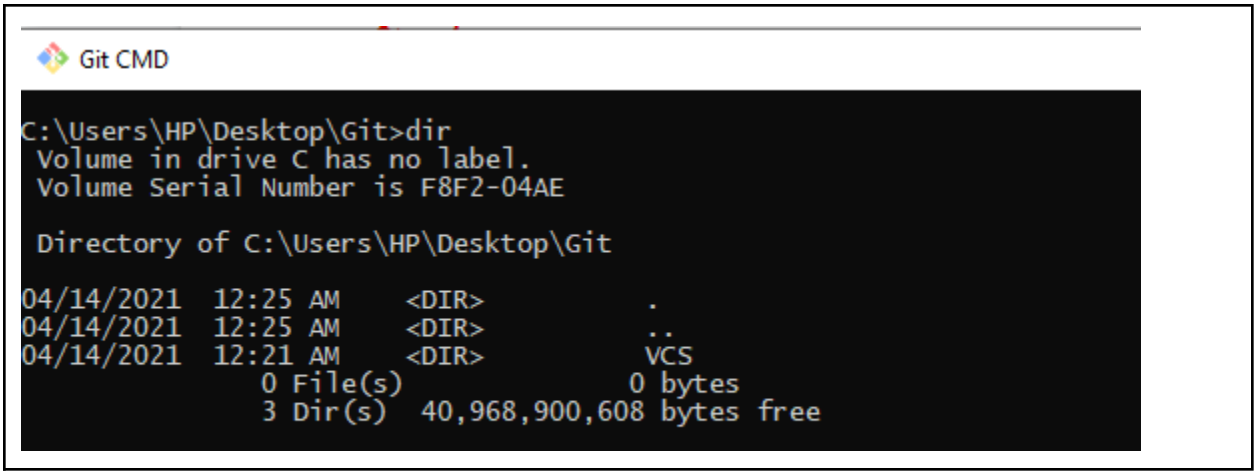
Step-2

Run the following command and give the **git url** of the repository you want to clone locally.

```
git clone URL
```



Repository cloned Locally (see using **dir**)



❖ Add File:

Add locally created file to the central(online) repository.

Step-1

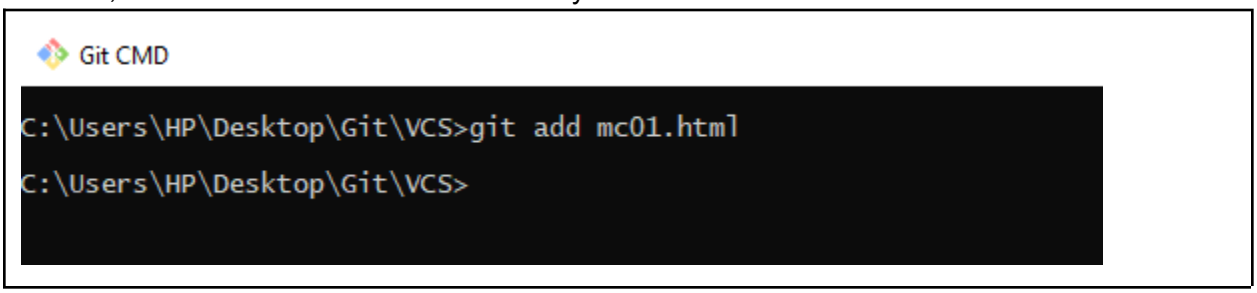
Creating file in local repository.

Step-2

Run the following command and give the **File Name** you want to add on git repository.

```
git add FileName
```

No error, so file has been added successfully.



❖ Add all files :

- **git add --all**
- **git add .**

❖ Add multiple files at same time:

git add FileName FileName FileName ...

❖ Git Commit

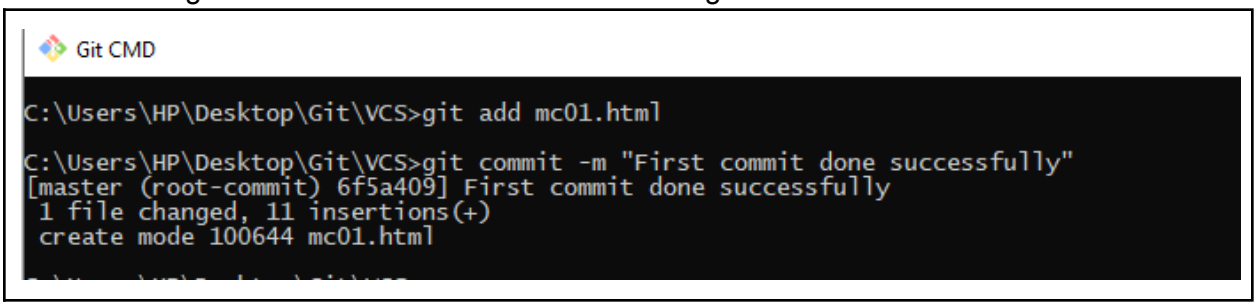
The "commit" command is used to save your changes and track messages.

Step-3

Run the following command and give the **Message** you want to get displayed with added file.

```
git commit -m “message”
```

Message=> to inform the users about the change.



❖ Add and commit simultaneously

```
git commit -am “message”
```

❖ Git Push:

To push the changes made in file locally to server (online repository).

Step-4

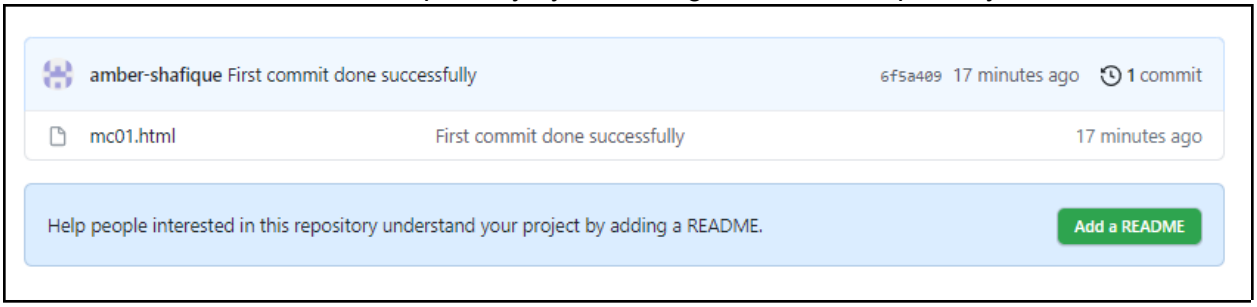
Run the following command to complete the synchronization.

```
git push
```

Now we are synchronized with data available on local disk and online repository.

```
Git CMD

C:\Users\HP\Desktop\Git\VCS>git push
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 345 bytes | 69.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amber-shafique/VCS.git
 * [new branch]      master -> master
```

Check the status of the online repository by refreshing the GitHub Repository URL.



Content in online and local repository is synchronized.

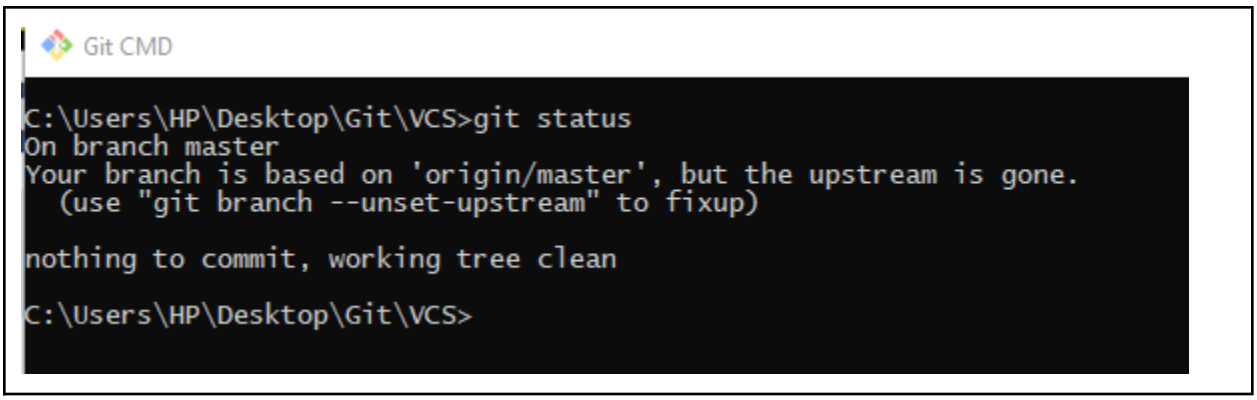
11 lines (10 sloc) | 146 Bytes

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4          <title>Mobile Computing</title>
5  </head>
6  <body>
7
8      <h1>Lectur-2</h1>
9      <p>Testing for GitHub</p>
10 </body>
11 </html>
```

- **Git Status:**

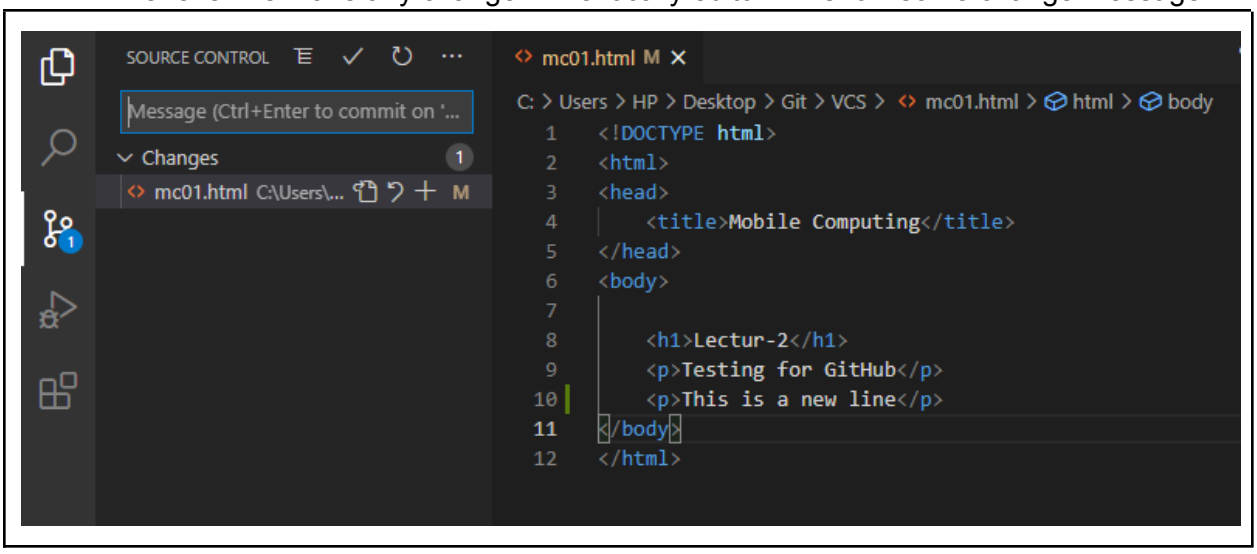
  Gives the status of repository.

git status



Making change in file locally using IDE:

Whenever we make any change in file locally editor will show some change message



❖ Adding New Commit (To save locally made changes online) :

Run these previously discussed commands on git CMD in sequence.

Step-1

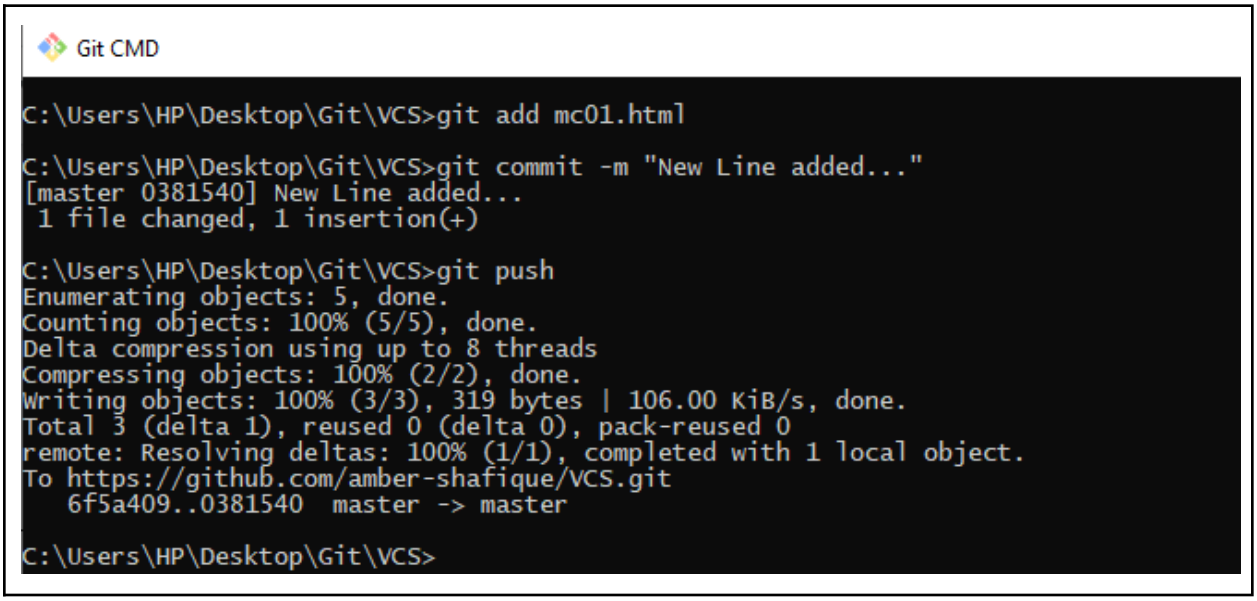
git add FileName

Step-2

git commit -m "message"

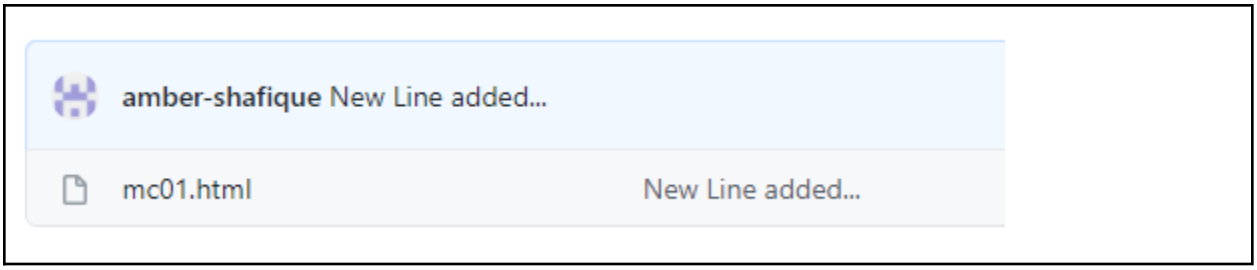
Step-3

git push



Track of Changes:

To preview change online refresh the repository url and see the changes have been made.



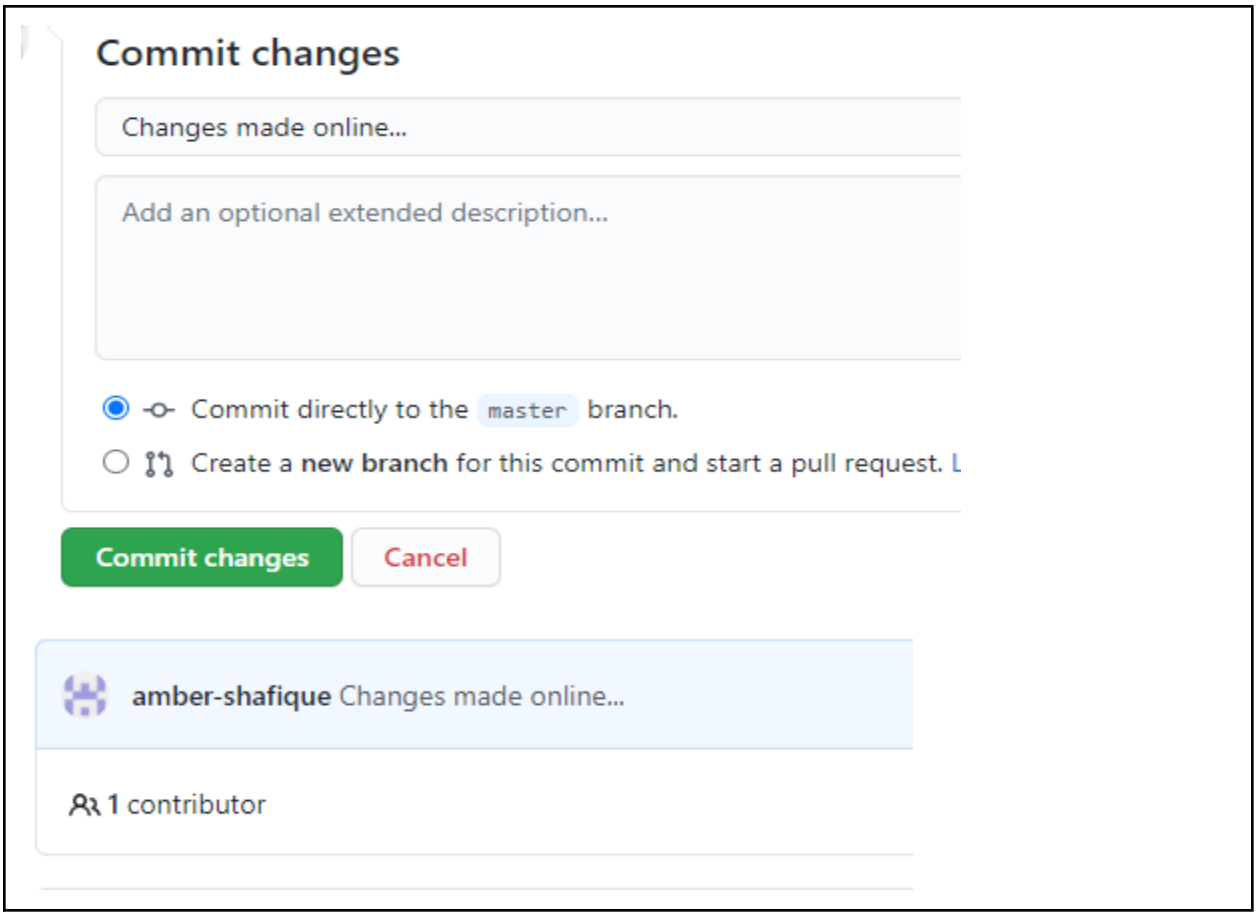
❖ Editing a file online in github repository:

Step-1

Edit the file online using the edit button on file.

Step-2

Save commit changes.



❖ Git Pull:

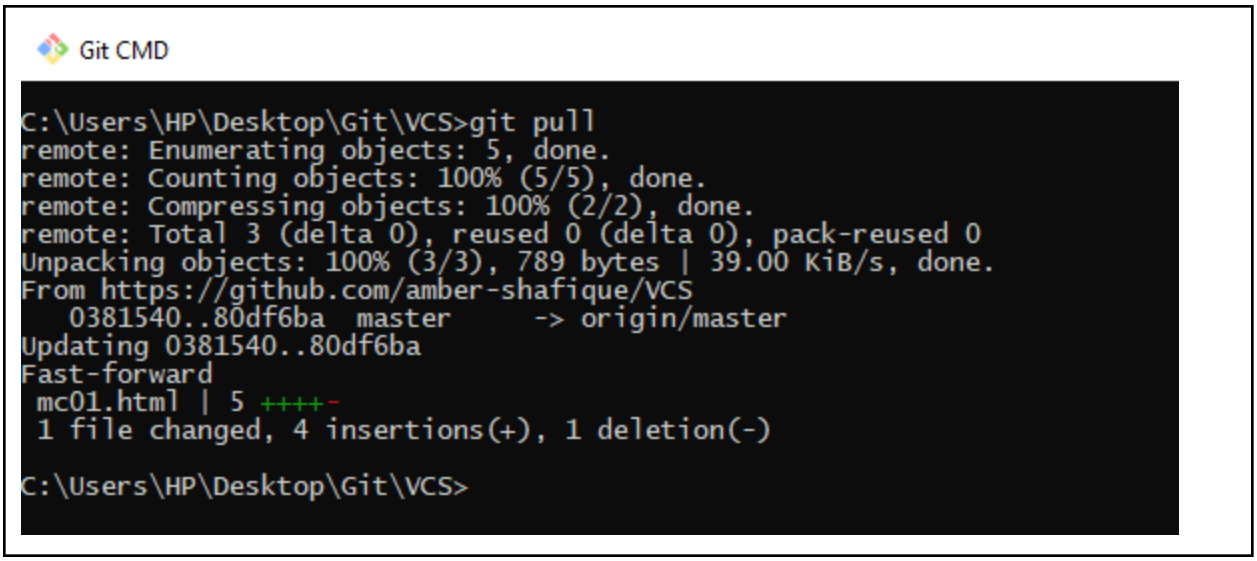
To get Changes from GitHub to local repository.

Step-1

After making changes online, run the following command on git CMD.

```
git push
```

Changes have been made to local repository.



❖ Version Hierarchy:

File showing with latest commit.

- **Red Highlighted lines:** The lines that have been deleted.
- **Green Highlighted lines:** The lines that have been changed.

Parents=> previous versions of file.

1 parent 80df6ba commit

LECTURE#3

❖ Merge Conflicts:

A merge conflict is an event that takes place when Git is unable to automatically resolve differences in code between two commits. Git can merge the changes automatically only if the commits are on different lines or branches.

Step-1

Changes in lin-8 online.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4          <title>Mobile Computing</title>
5  </head>
6  <body>
7
8      <h1>Lectur-2 changed by git</h1>
```

Step-2

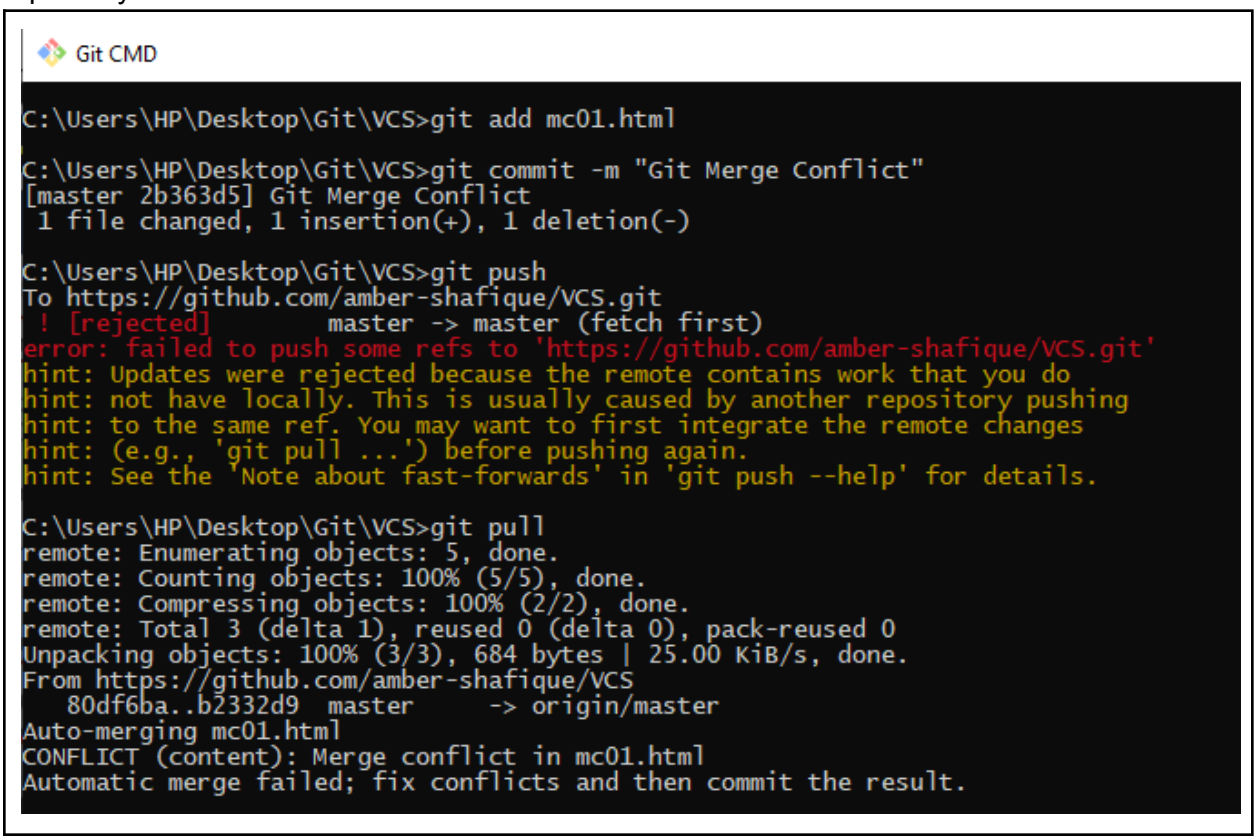
Changes in same line-8 locally.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Mobile Computing</title>
5  </head>
6  <body>
7
8      <h1>Lectur-2 chaged locally</h1>
```

Step-3

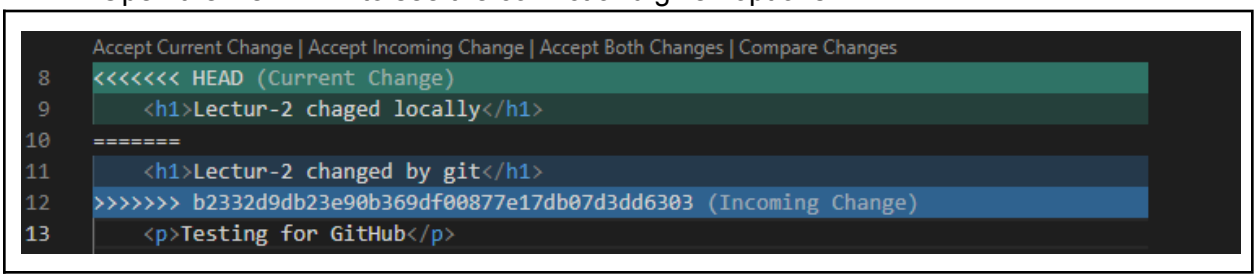
Run commands step by step to save file changes in git CMD.

Merge Conflict has occurred because we are trying to change the same line locally and in online repository.



Step-4

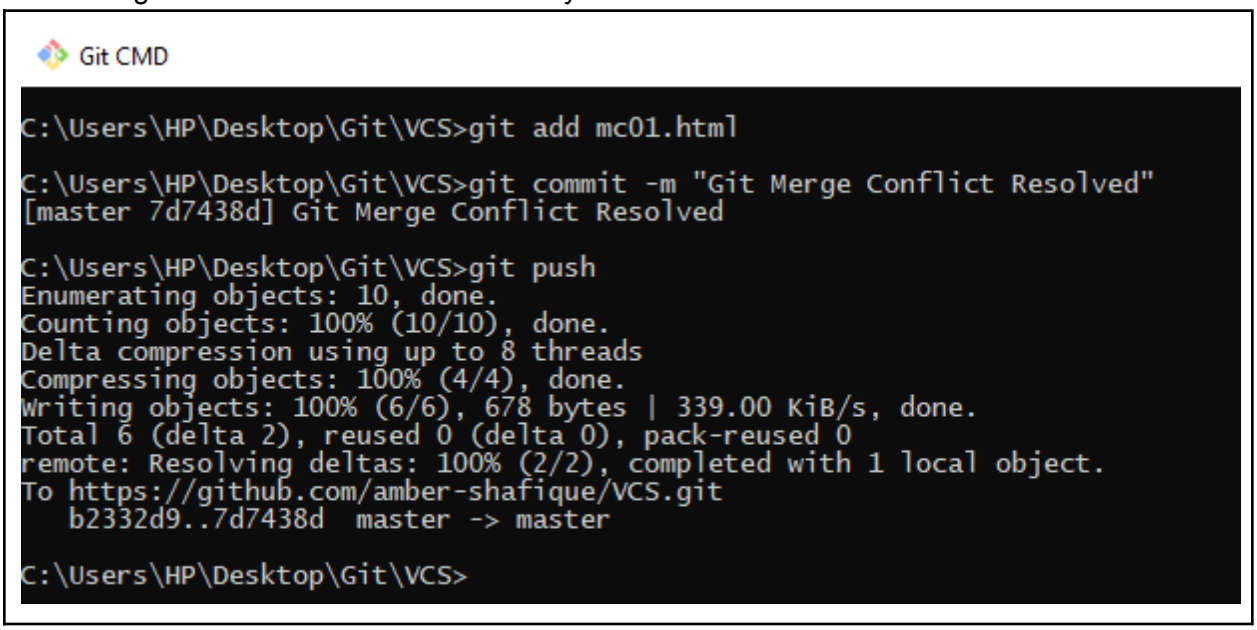
Open the file in IDE to see the conflict and given options.



Step-5

Accept Change and resolve the merge conflict and **again** Run commands step by step to save file changes in git CMD.

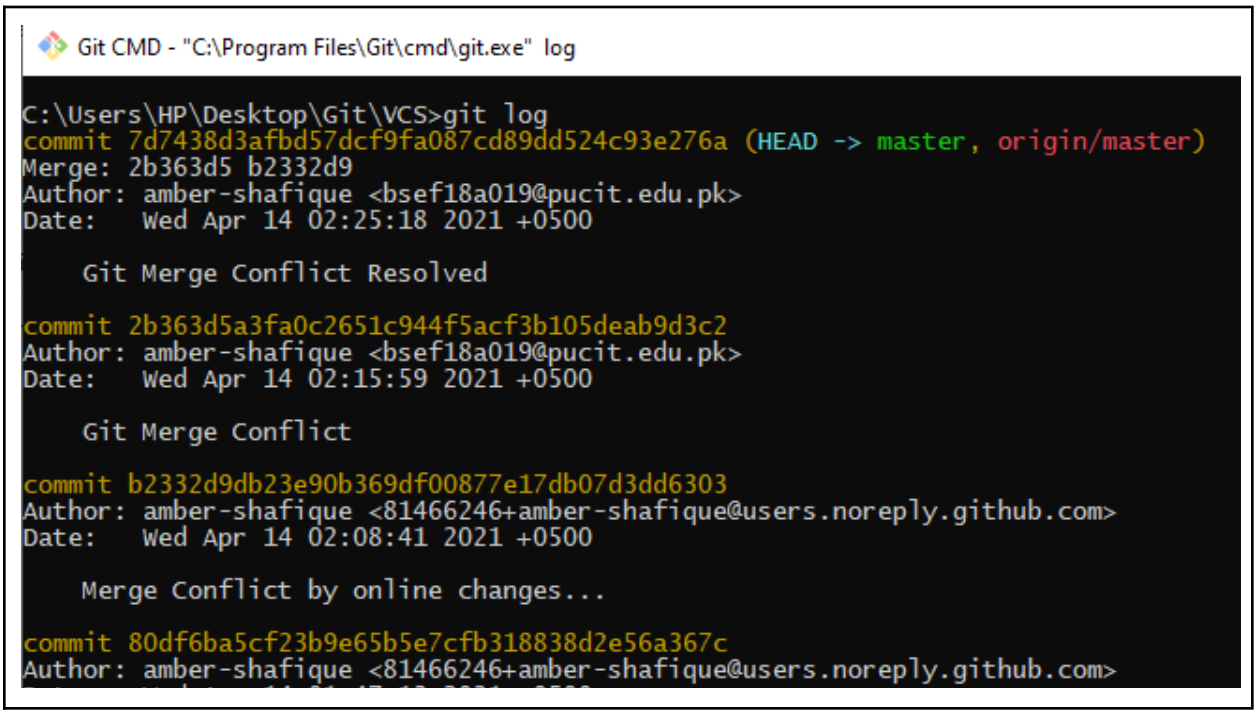
Now changes have been saved successfully as conflict is resolved.



❖ Git Logs:

To see the details of all commits.

```
git log
```



❖ Remove a File:

Files already present

```
C:\Users\HP\Desktop\Git\MC-1>dir
 Volume in drive C has no label.
 Volume Serial Number is F8F2-04AE

 Directory of C:\Users\HP\Desktop\Git\MC-1

04/14/2021  03:44 PM    <DIR>          .
04/14/2021  03:44 PM    <DIR>          ..
04/14/2021  02:42 AM                 6 file1.txt
04/14/2021  03:44 PM                 6 file2.txt
04/14/2021  03:06 AM                 7 file3.txt
               3 File(s)             19 bytes
               2 Dir(s)  41,993,297,920 bytes free
```

Step-1

To remove file type the following command on git CMD, and give the name of file you want to delete.

git rm FileName

File deleted locally

Git CMD

```
C:\Users\HP\Desktop\Git\MC-1>git rm file1.txt
rm 'file1.txt'

C:\Users\HP\Desktop\Git\MC-1>
```

Now to update delete changes to git:

Step-2

Git add .

Step-3

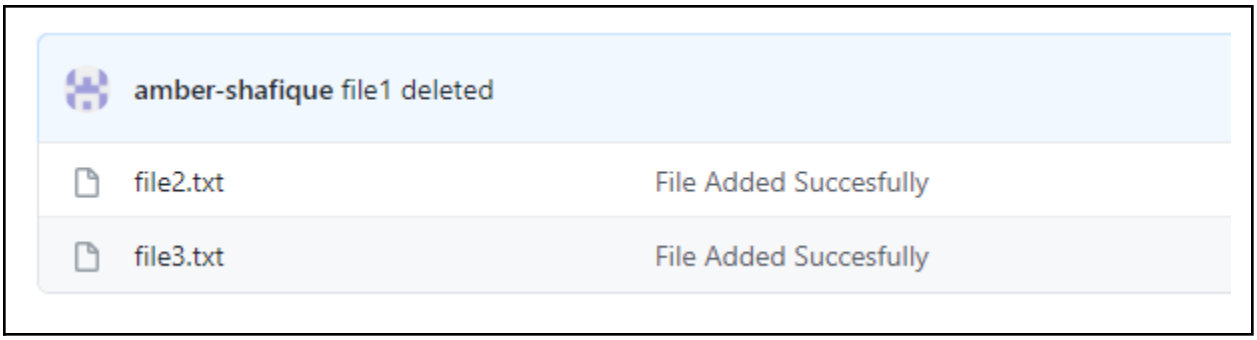
git commit -m “message”

Step-4

git push

```
C:\Users\HP\Desktop\Git\MC-1>git add .

C:\Users\HP\Desktop\Git\MC-1>git commit -m "file1 deleted"
[master 37a39ee] file1 deleted
 1 file changed, 1 deletion(-)
 delete mode 100644 file1.txt

C:\Users\HP\Desktop\Git\MC-1>git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 264 bytes | 88.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/amber-shafique/MC-1.git
   f837866..37a39ee  master -> master

C:\Users\HP\Desktop\Git\MC-1>
```

Preview delete changes on GitHub



❖ Branching:

Branching is the practice of creating copies of programs or objects in development to work in parallel versions, retaining the original and working on the branch or making different changes to each.

Step-1

Git Branch:

Run the following command on cmd to check available branches.

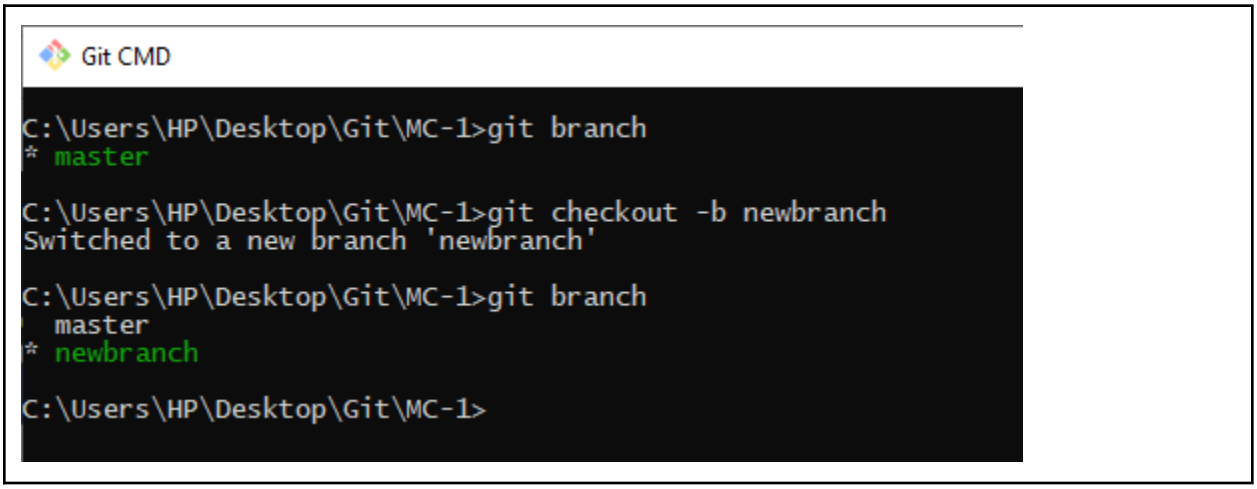
```
git branch
```

Step-2

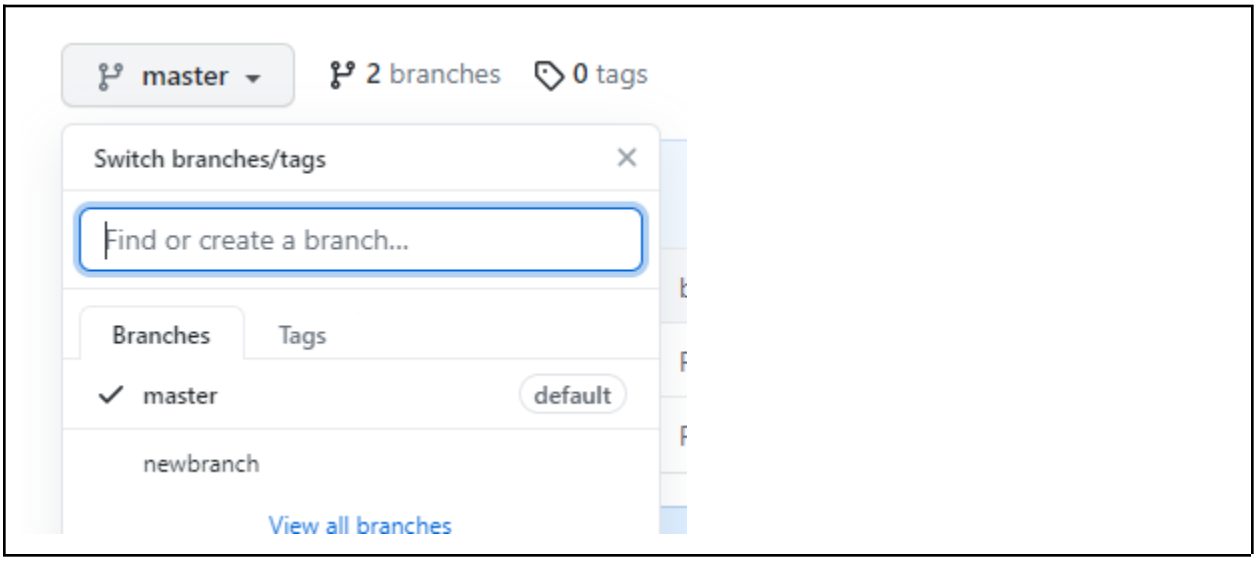
Git Checkout:

To create a new branch, type the following command on git CMD and give the name of branch.

```
git checkout -b newbranch
```



Branch Added on GitHub

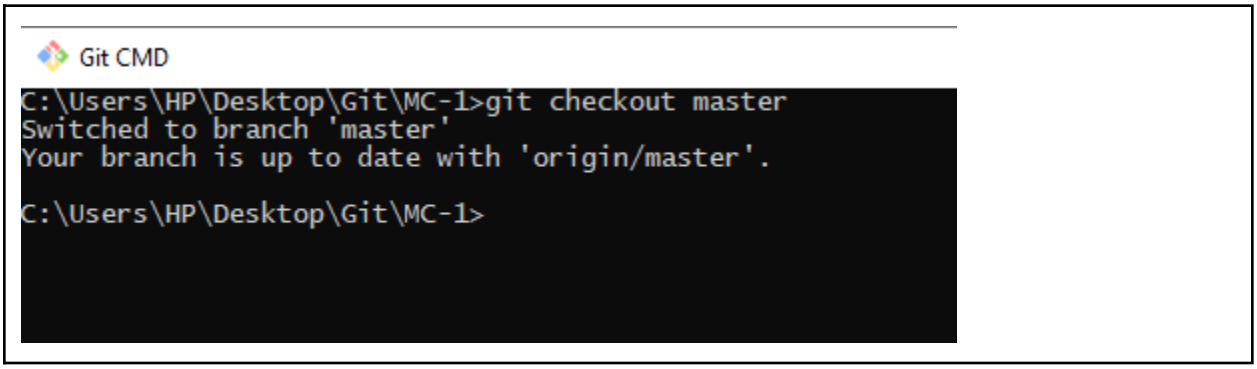


Step-3

Git Checkout Master:

To navigate from one branch to another. (master=>branch name)

git checkout master



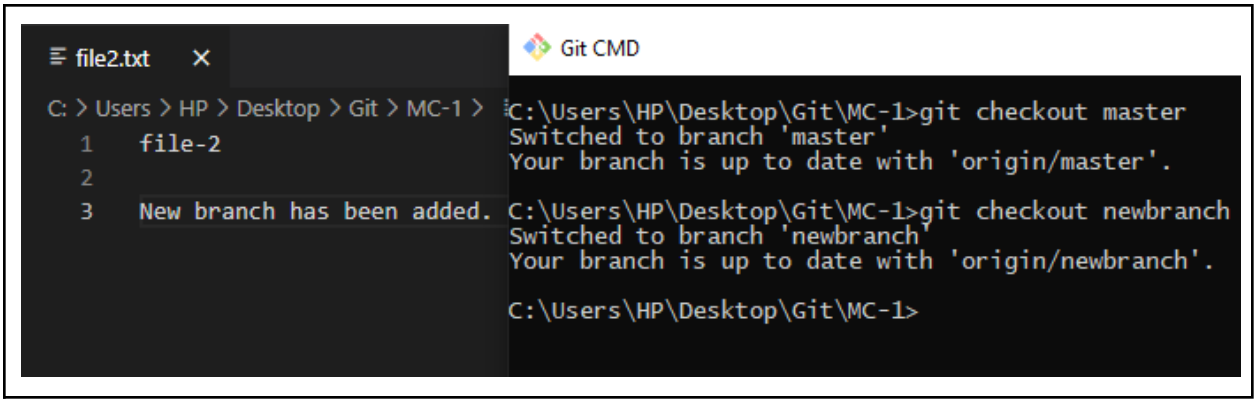
To Check Branching:

Step-1 Add changes to a file on master branch.

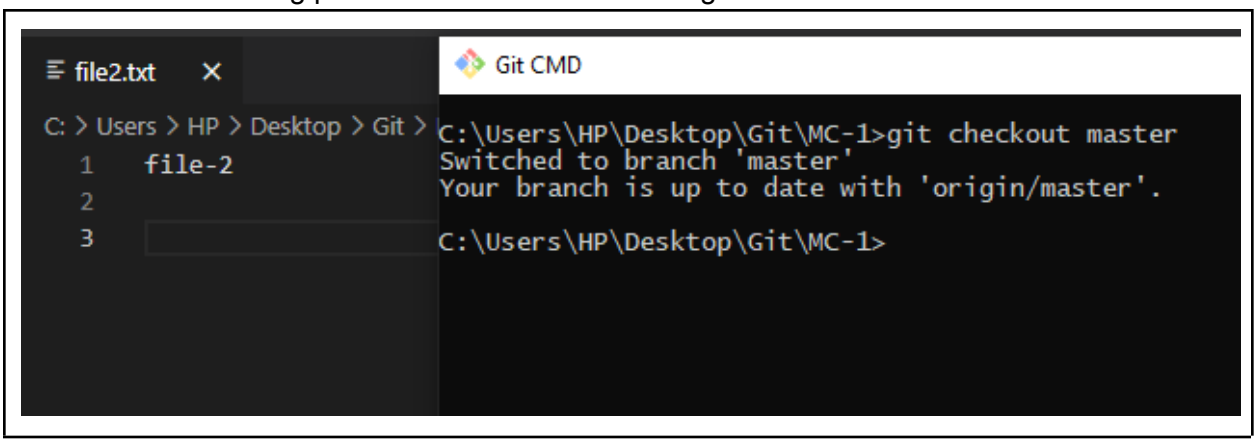
Step-2 Move to new branch and add file on the new branch.

Step-3 Checkout file on both branches and see the changes accordingly on IDE.

New branch showing changes...



Master branch showing previous version without changes...



❖ Merging:

Merging is Git's way of putting a forked history back together again. The git merge command lets you take the independent lines of development created by git branch and integrate them into a single branch.

Merge Branch:

To merge the changes in branches.

Git merge branchname

```
C:\Users\HP\Desktop\Git\MC-1>git merge newbranch
Removing file3.txt
Auto-merging file2.txt
CONFLICT (content): Merge conflict in file2.txt
CONFLICT (add/add): Merge conflict in BranchingPractice.html
Auto-merging BranchingPractice.html
Automatic merge failed; fix conflicts and then commit the result.

C:\Users\HP\Desktop\Git\MC-1>
```

❖ Delete Branch:

To delete a branch.NewBranch=>you want to delete.

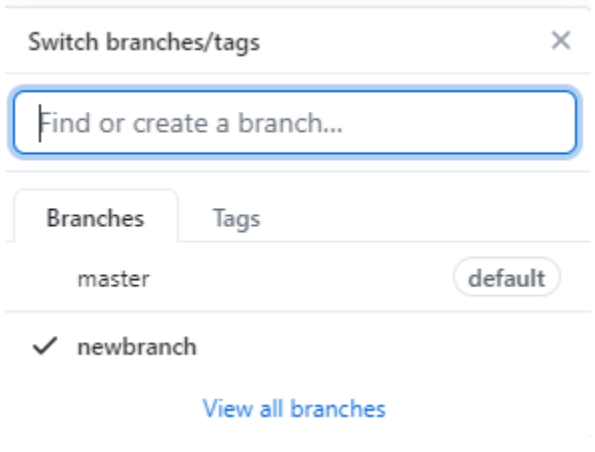
Git branch -D NewBranch

Checking out branches before and after deletion...

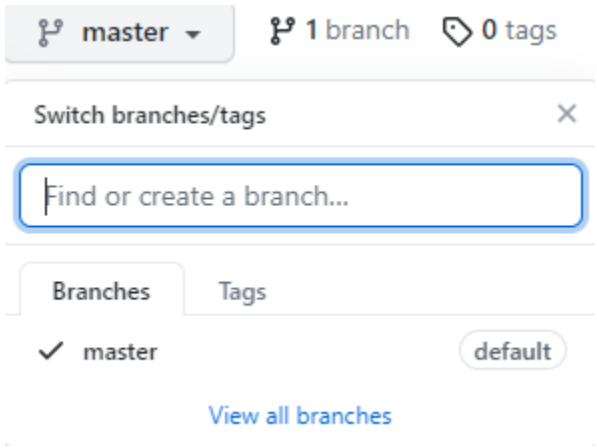
Git CMD

```
C:\Users\HP\Desktop\Git\MC-1>git branch
* master
  newbranch

C:\Users\HP\Desktop\Git\MC-1>git branch -D newbranch
Deleted branch newbranch (was ba2149a).

C:\Users\HP\Desktop\Git\MC-1>git branch
* master

C:\Users\HP\Desktop\Git\MC-1>
```

Before Deletion



After Deletion



LECTURE#4

❖ Android Studio:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. A unified environment where you can develop for all Android devices. Apply Changes to push code and resource changes to your running app without restarting your app.

Step-1

Installation:

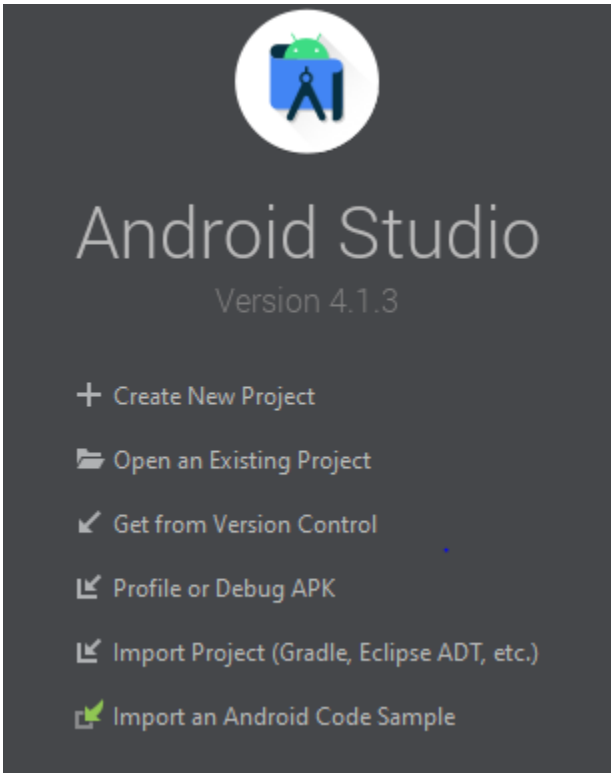
Download any latest version of **Android Studio** and configure it on your PC.

(https://developer.android.com/studio)

Step-2

Create a New Project:

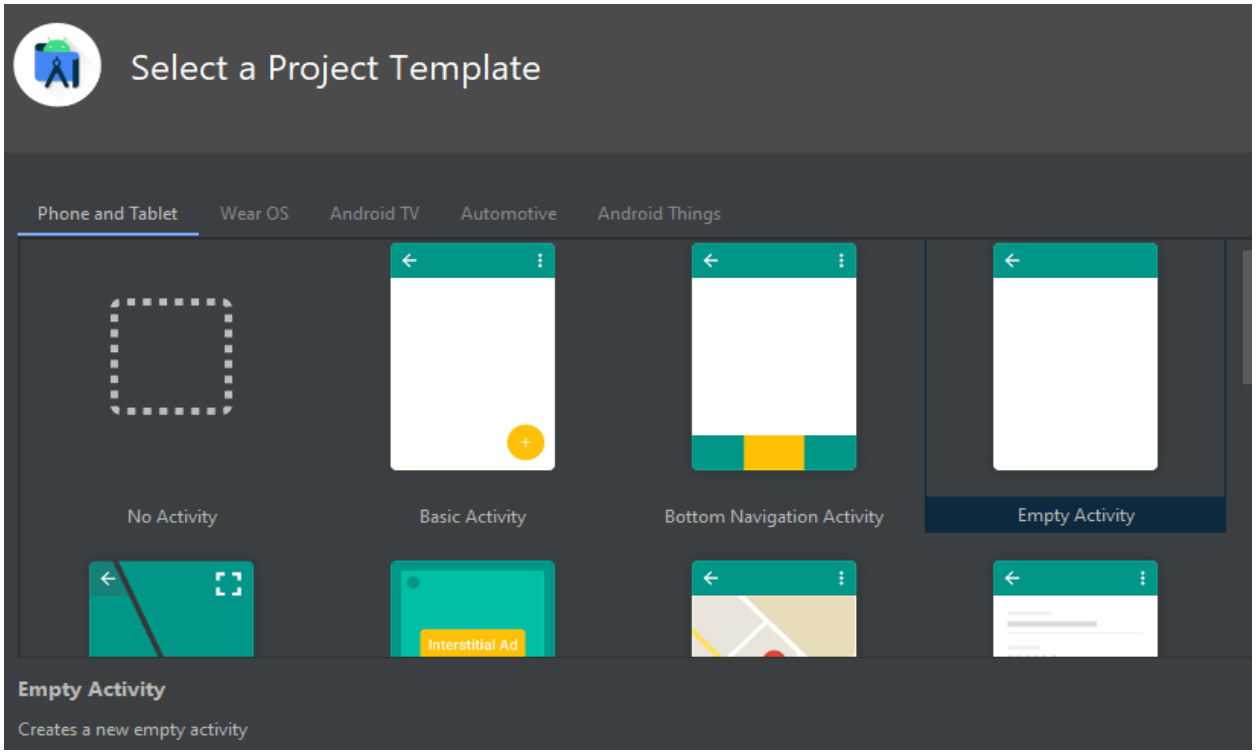
Open Android Studio and select the option to create a new project.



Step-3

Template Selection:

From templates choose Empty Activity.



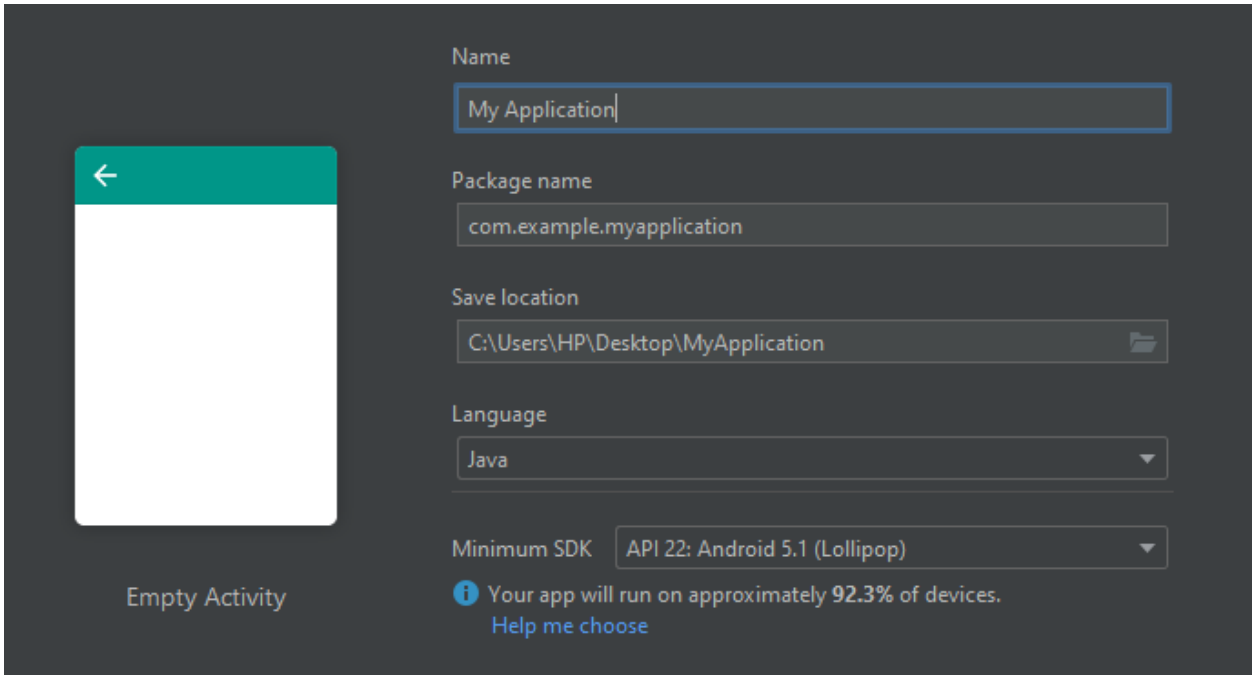
Step-4

Project Name:

Select a name and location for project for project.

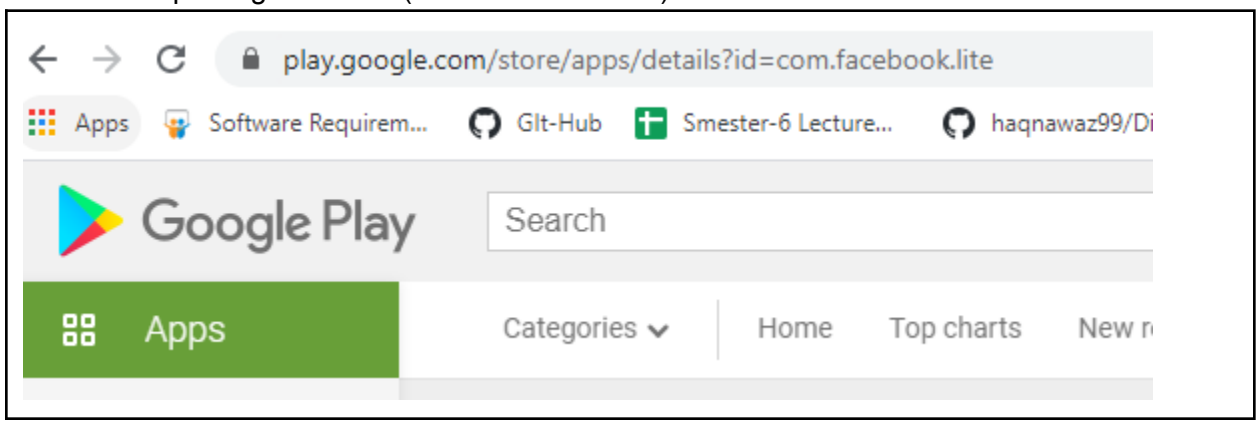
Package Name:

It is used to uniquely identify the APK file of our application and it should must be unique to publish the App on play store.

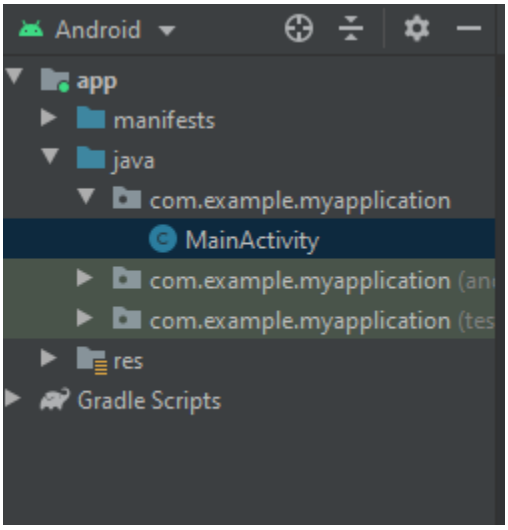
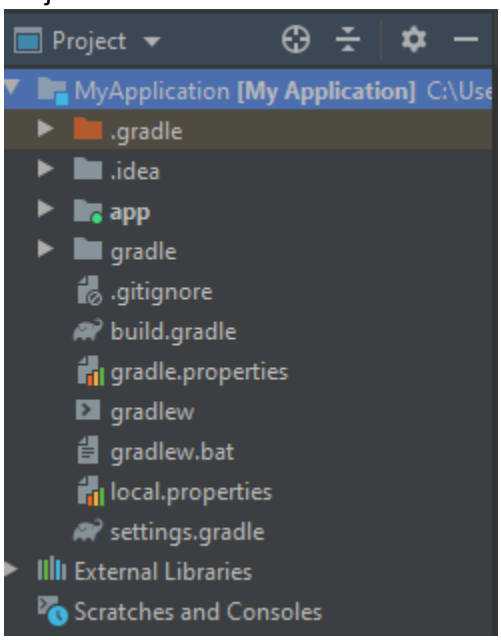


Package Name Example:

Here package name is (**com.facebook.lite**)

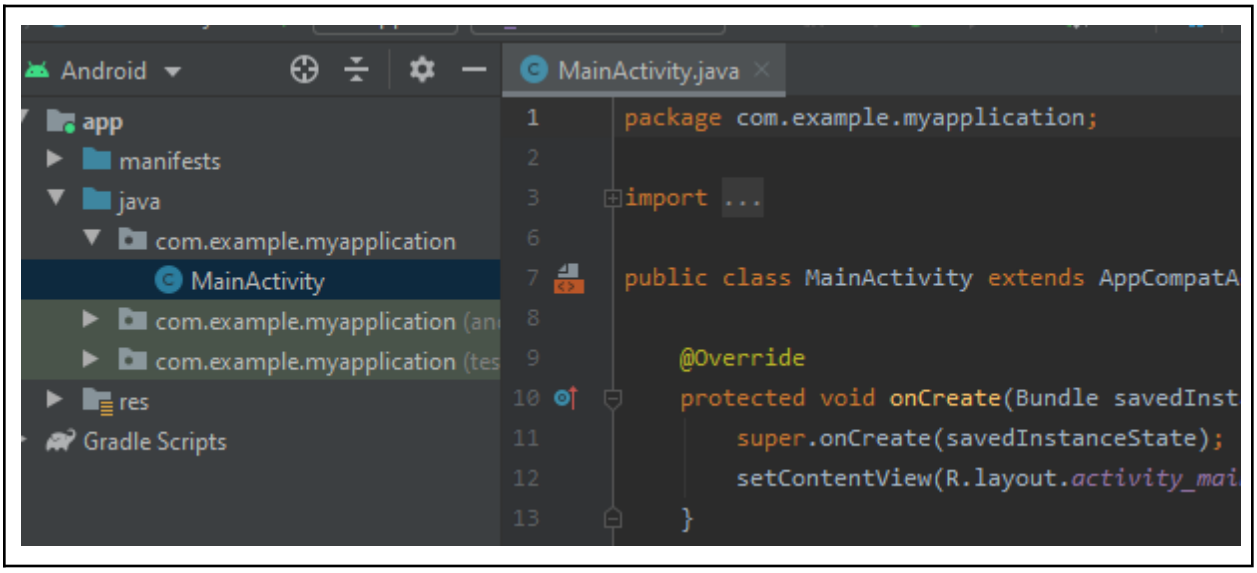


❖ Android Studio File Structure:

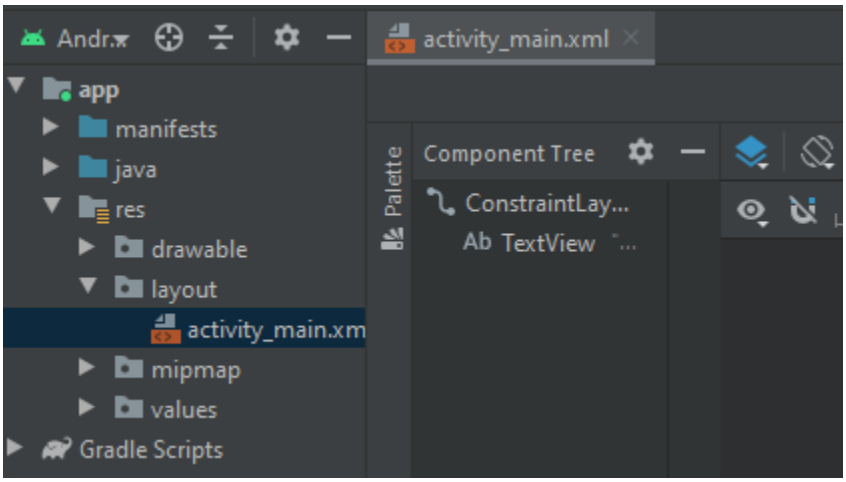
Android view (used by default)	Project view
	

Files for coding:

- **MainActivity.java** file for coding.



Activity_main.xml file for Graphical user interface(**GUI**).



Step-4

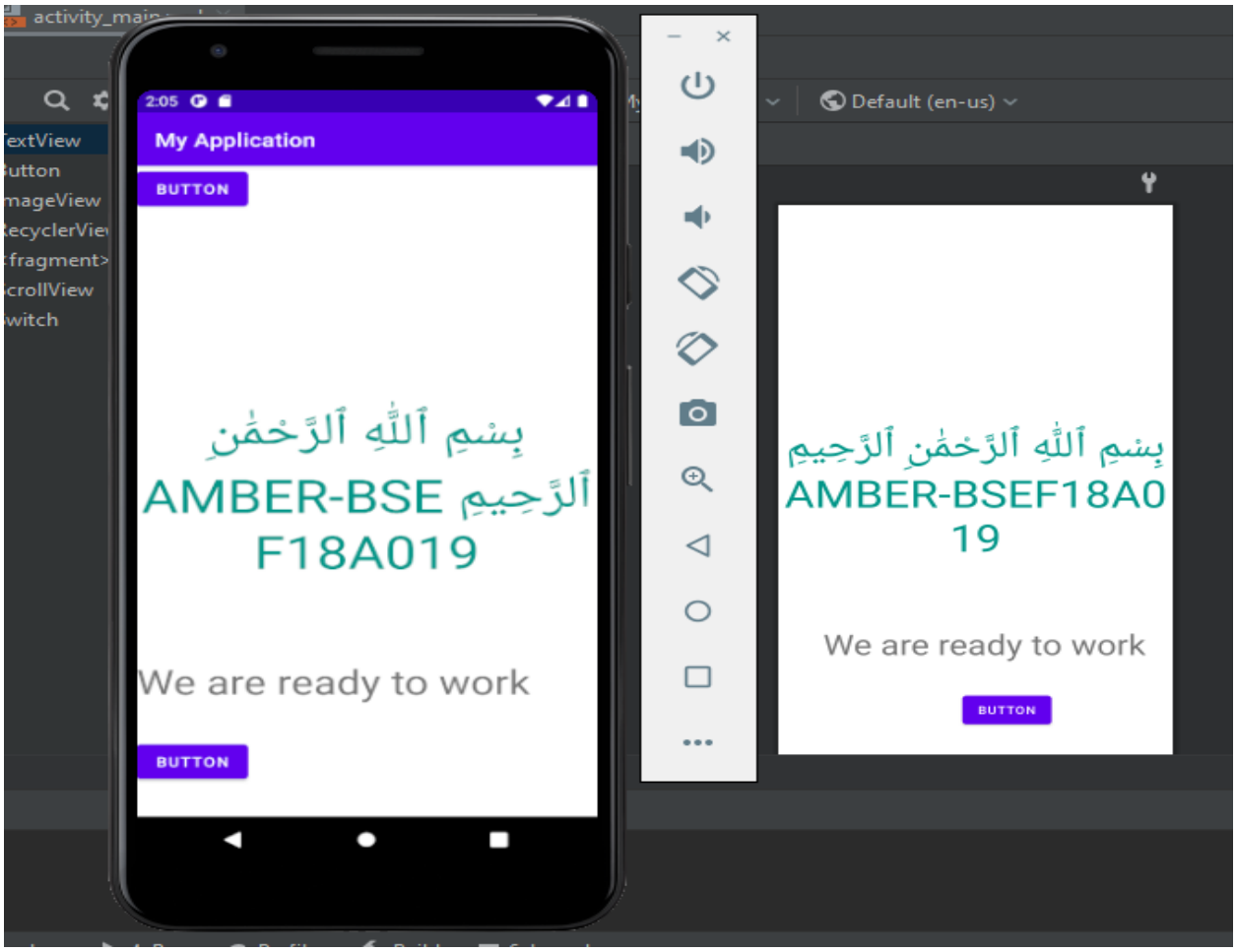
Select Virtual Device:

Select **Virtual Devices** from AVD(Android Virtual Device) Manager to view output.

Step-4

To view output:

Run the **Emulator** choosing any virtual device and see the output.


2:05
My Application
BUTTON
بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ
AMBER-BSE ٱلرَّحِيمِ
F18A019
We are ready to work
BUTTON
Default (en-us)
بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ
AMBER-BSEF18A019
We are ready to work
BUTTON

To Do list:

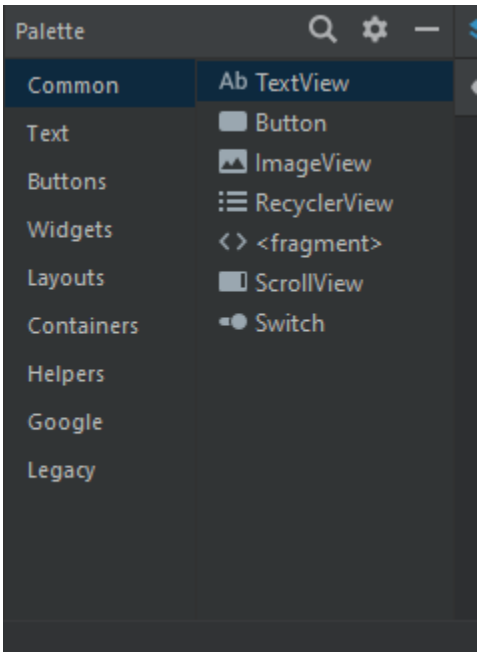
```
6
7   public class MainActivity extends AppCompatActivity {
8
9       @Override
10      protected void onCreate(Bundle savedInstanceState) {
11          super.onCreate(savedInstanceState);
12          //TODO: it is testing one
13          setContentView(R.layout.activity_main);
14
15          String name="Amber";
16          //It will be a comment
17          //TODO: it is testing two
18      }
19  }
```

TODO: Project Current File Scope Based Default Changelist

Found 2 TODO items in 1 file
com.example.myapplication 2 items
MainActivity.java 2 items
(12, 11) //TODO: it is testing one
(17, 11) //TODO: it is testing two

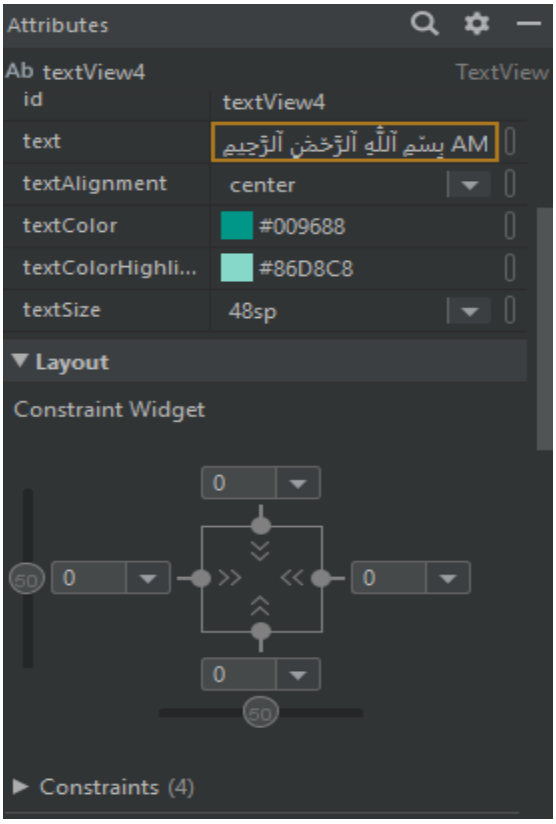
Design Palette:

We can add text,Button,image and other controls by using drag and drop from this palette.



Attributes:

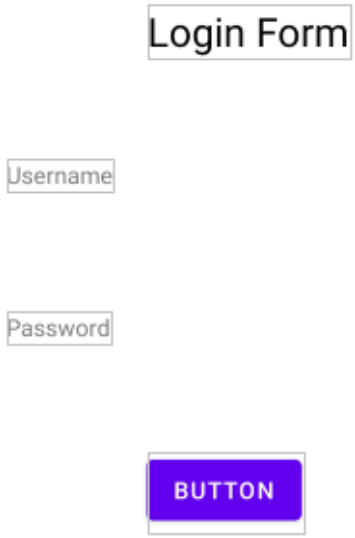
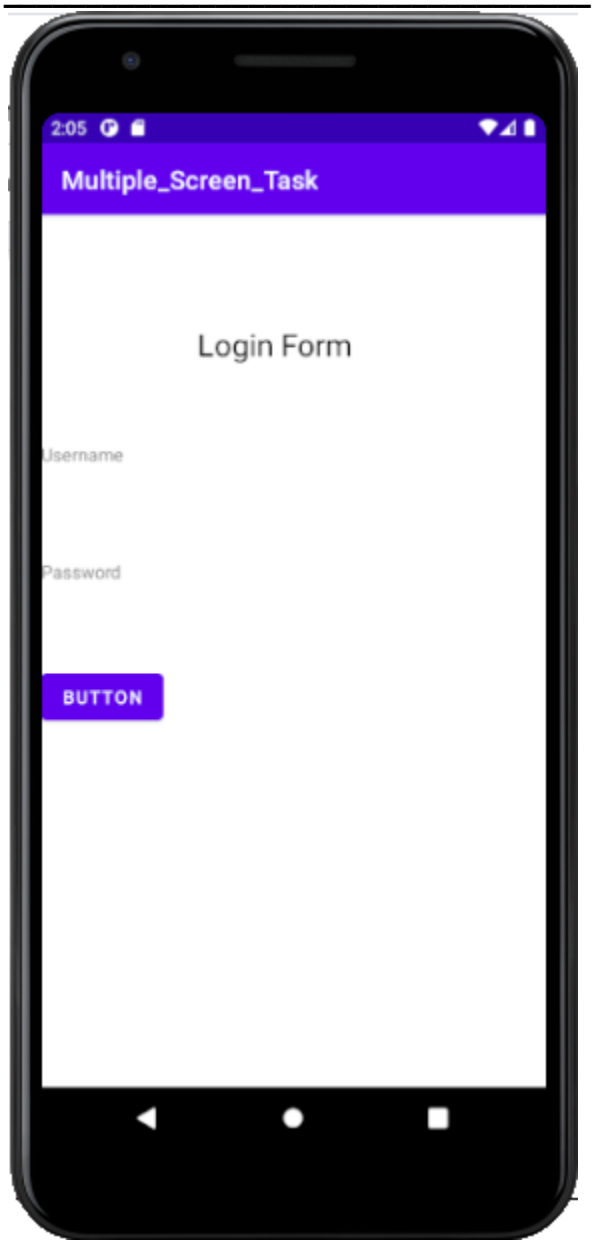
To set the values of elements added in xml.



❖ Multiple Screen Task:

Screen-1

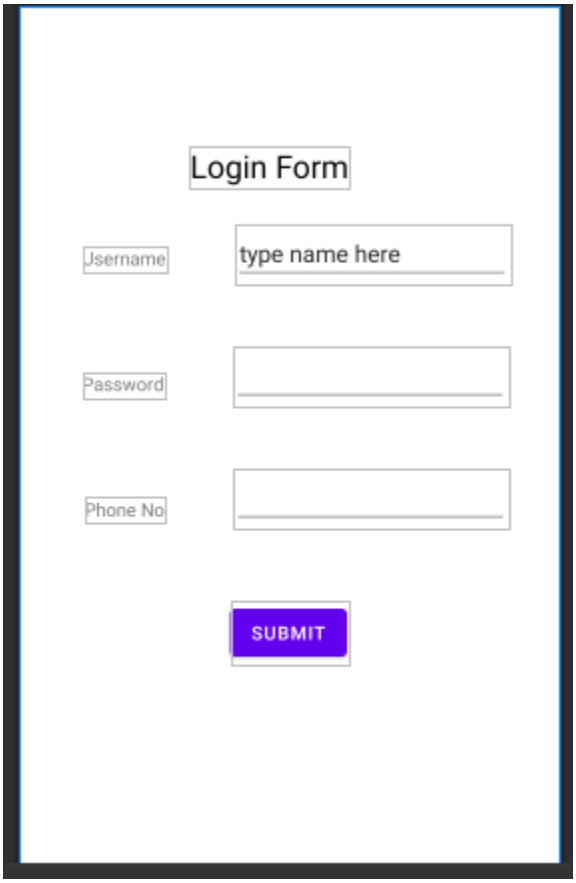
Step-1 Add elements of text box and buttons on the screen from palette by drag and drop.

XML View	Emulator View
	

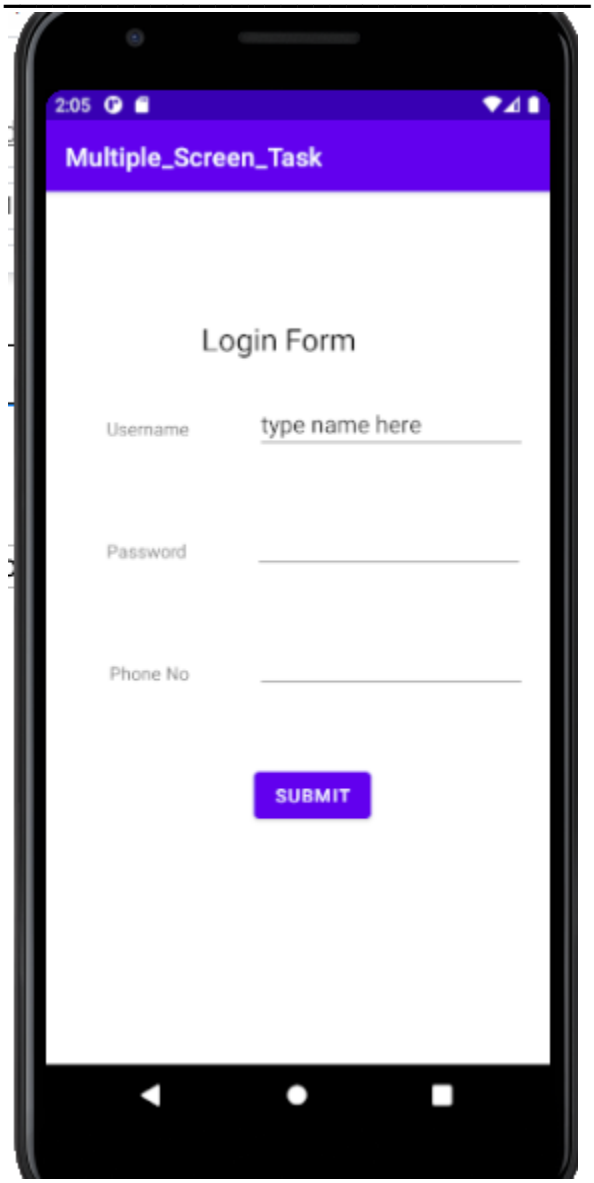
Screen-2

Step-2 Adding more elements of plain text and password to take input.

XML View

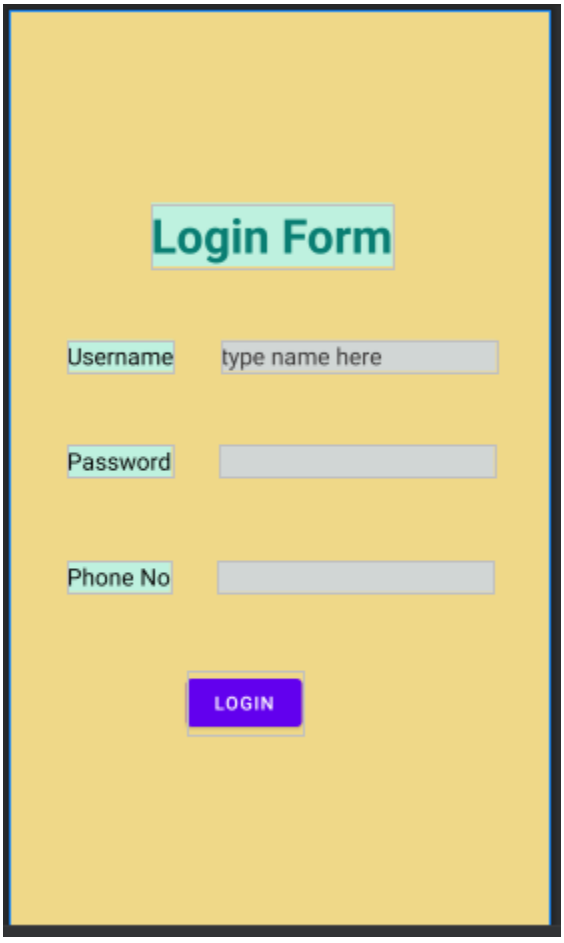
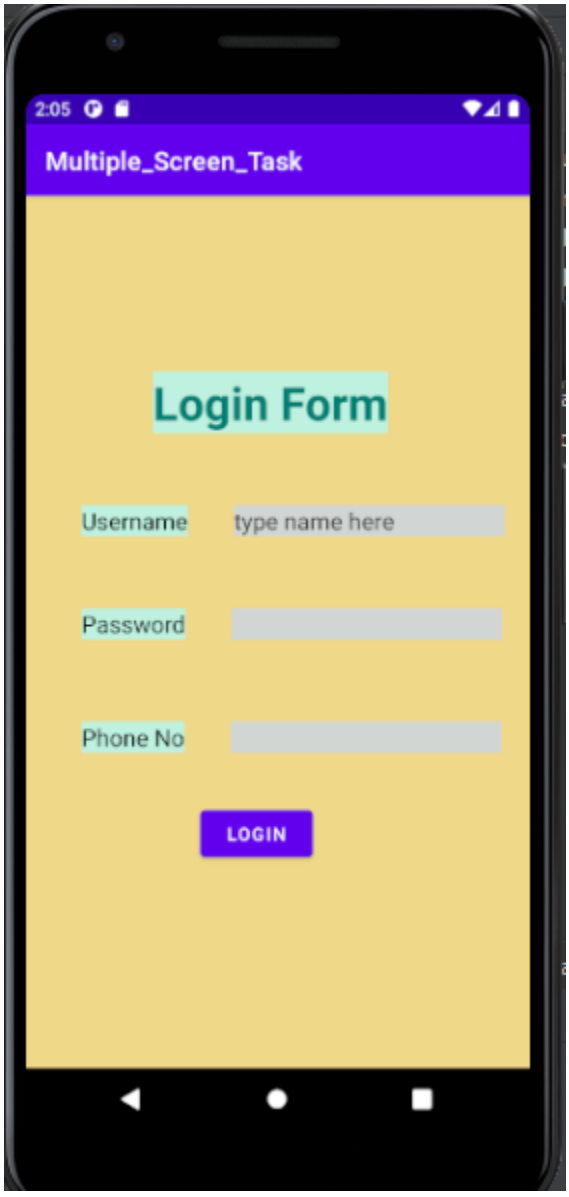


Emulator View



Screen-3

Step-3 Setting constraints and others attributes to each element to set the screen presentation.

XML View	Emulator View
	

Final Screen view taking input:

LECTURE#5

❖ View Group:

A ViewGroup is an **invisible container**. It is a special view that can contain other views (called children.)

- **Layouts:**

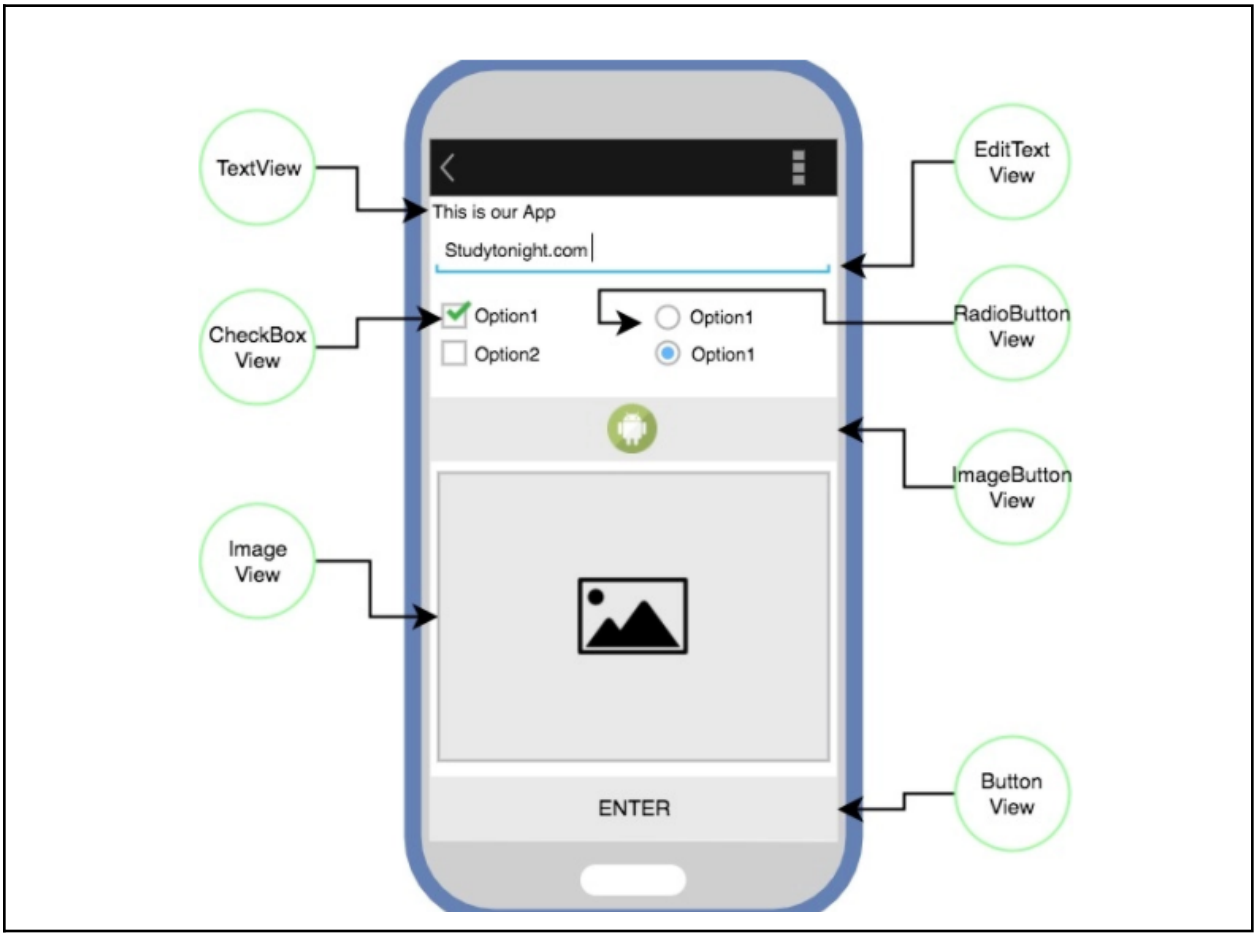
  View group objects are called **“layouts”** can be one of many types that provide a different layout structure, such as LinearLayout or ConstraintLayout .

❖ View:

View is the basic building block of **UI**(User Interface) in android. View refers to the android. view. View class, which is the super class for all the GUI components like TextView, ImageView, Button etc. It draws something that user can see and interact with.

- **Widgets:**

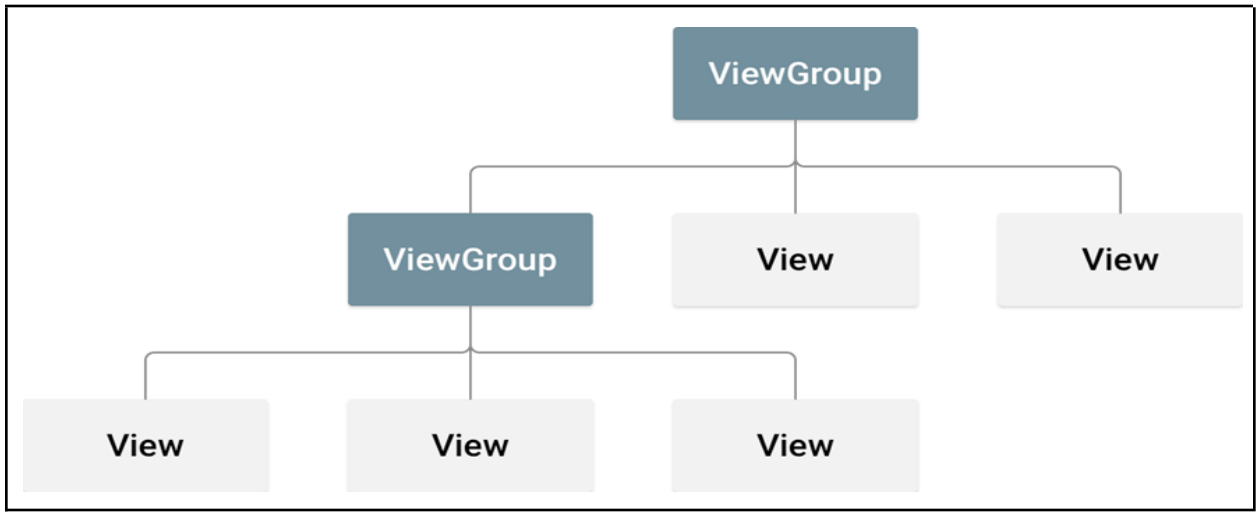
  The View objects are usually called **"widgets"** and can be one of many subclasses, such as Button or TextView.



Root Element:

Each layout file must contain exactly one **root element**, which must be a **View or ViewGroup** object. Once you've defined the root element, you can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout. For example, here's an XML layout that uses a vertical LinearLayout to hold a TextView and a Button.

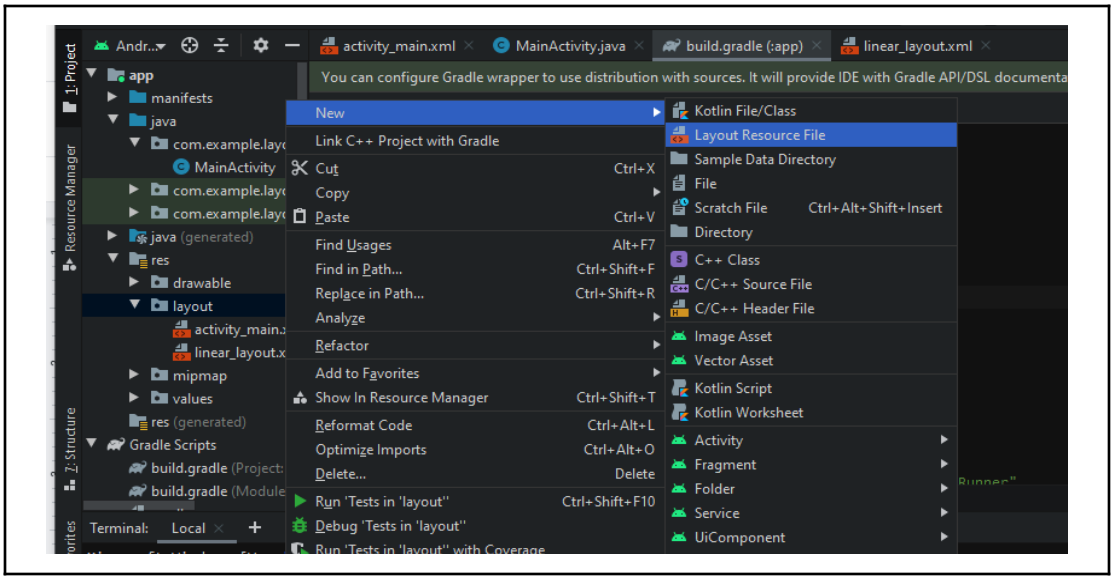
Constraint Layout is the root element by default.



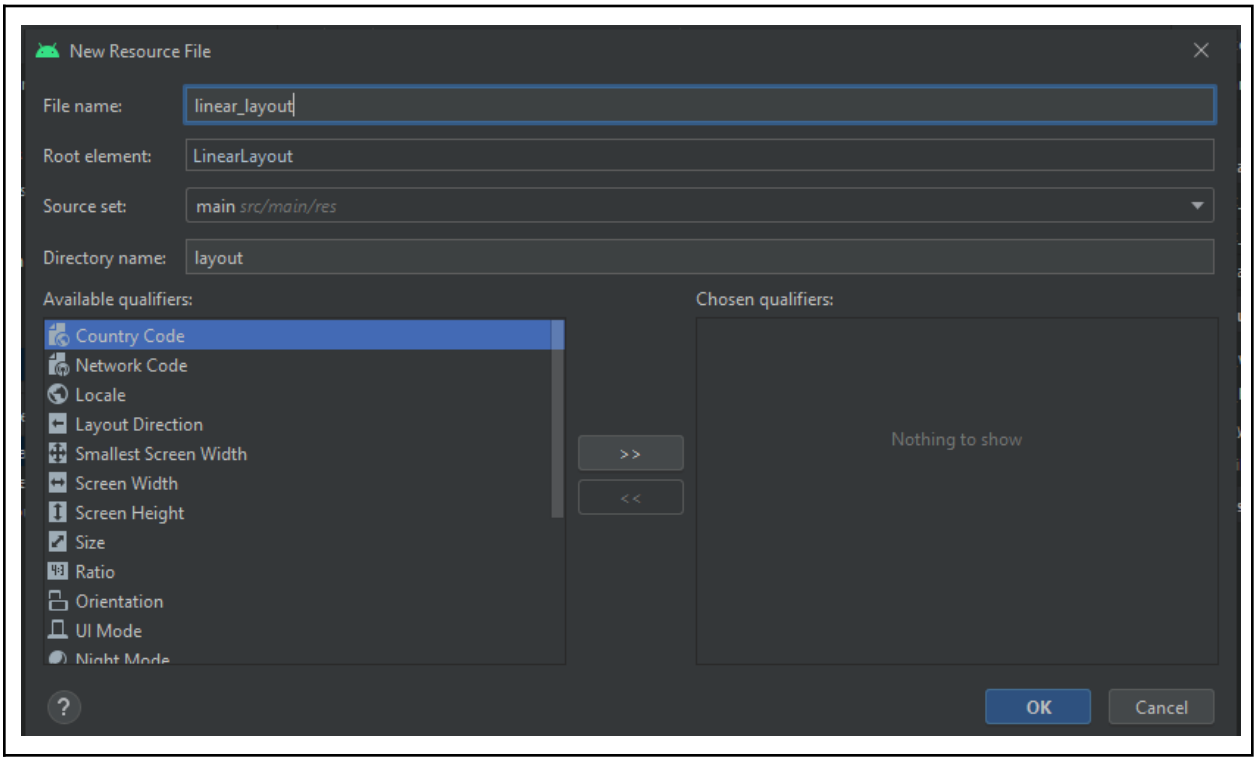
❖ Layouts Practice:

Step-1 Create a new project with empty template.

Step-2 Add a new resource file.



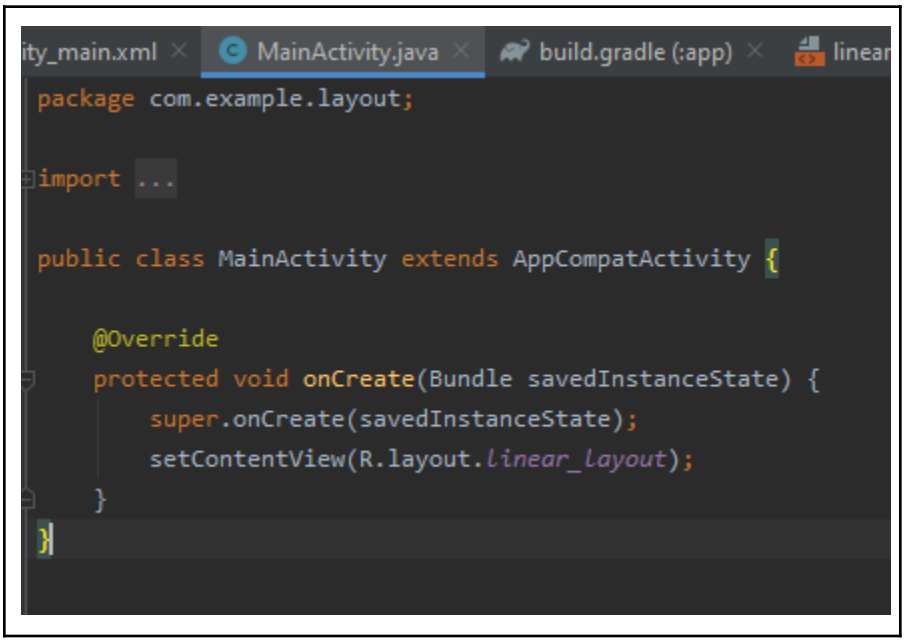
Step-3 Select the root element of **LinearLayout**.



Step-4 Add a text view and button, and you will see that these elements are displayed in a linear layout on xml design view.

❖ Load XML Resources:

Step-5 In MainActivity.java file change the **setContentView(R.layout .___)**, to the layout file you want to run on emulator.



Layout Parametres:

They are used by views to tell their parents how they want to be laid out.

❖ Constraint Layout:

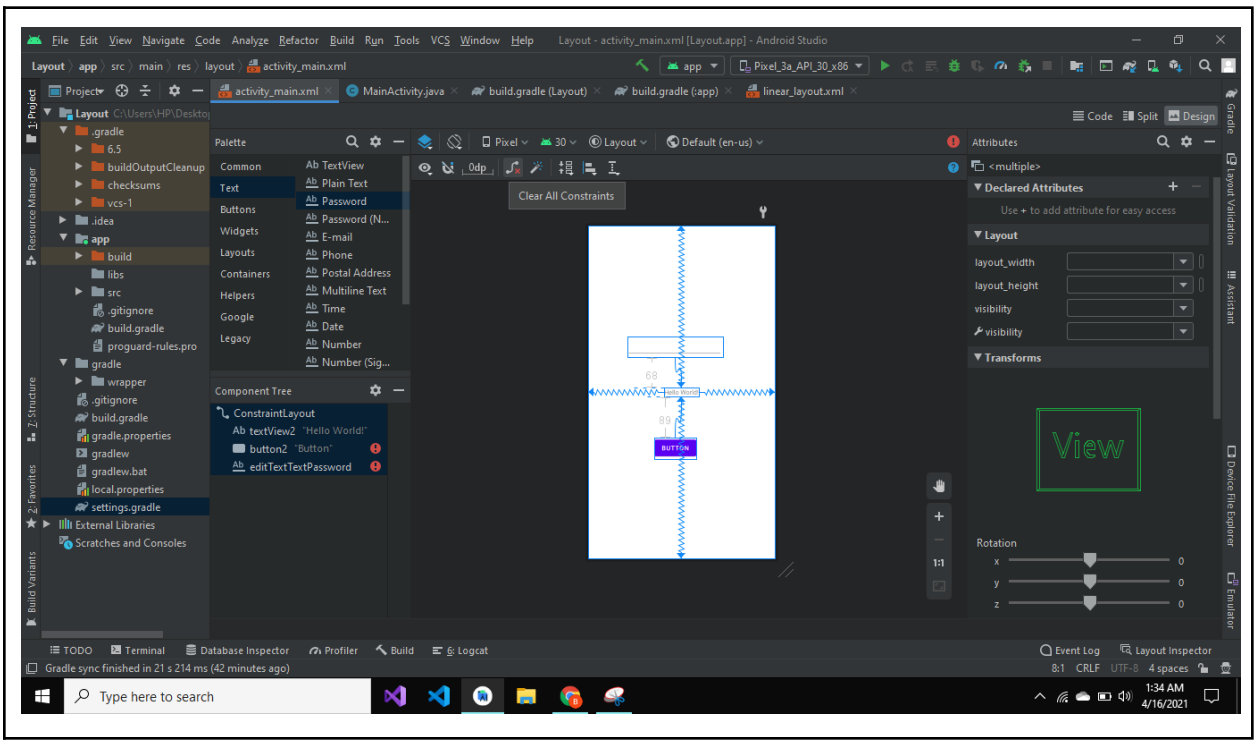
In Constraint Layout, you can connect all the view components to each other and place them on screen.

- To define a **view's position** in ConstraintLayout , you must add at least one horizontal and one vertical constraint for the view. Each constraint represents a connection or alignment to another view, the parent layout, or an invisible guideline.

Clear All constraints:

Step-1 Select all on xml design layout by **ctrl+a**

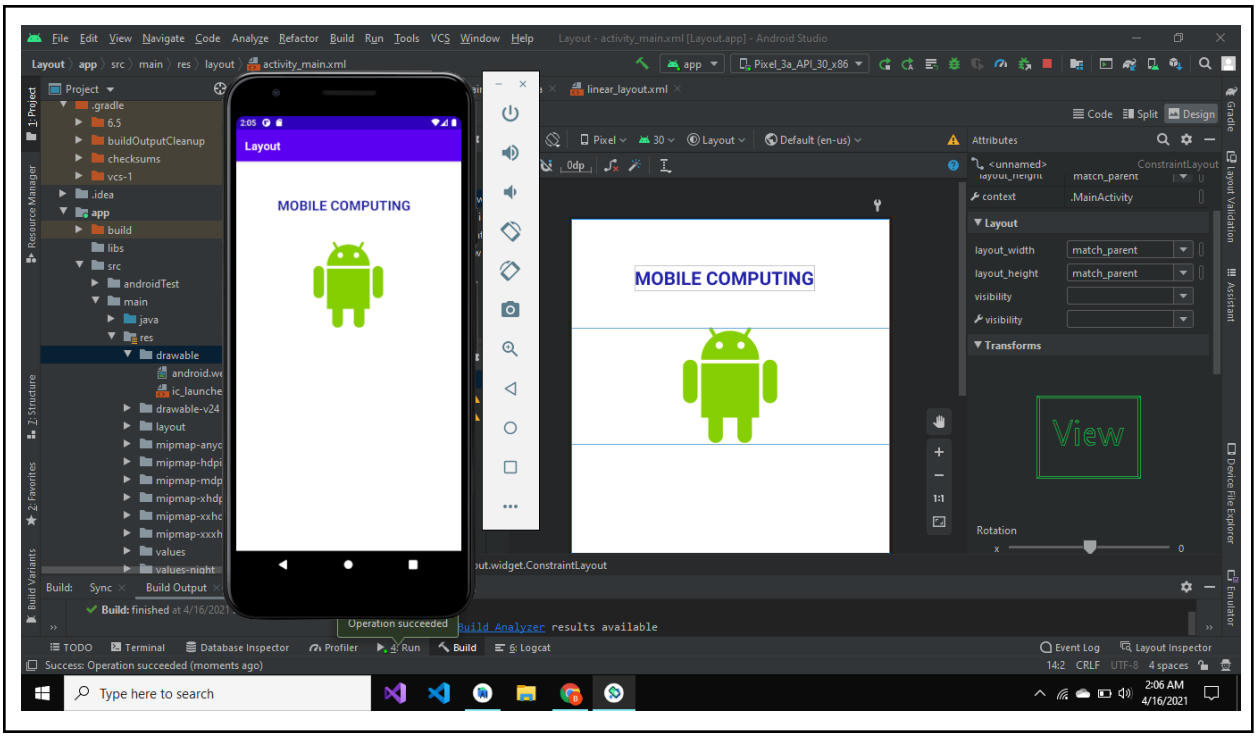
Step-2 Select the option Clear All constraints.

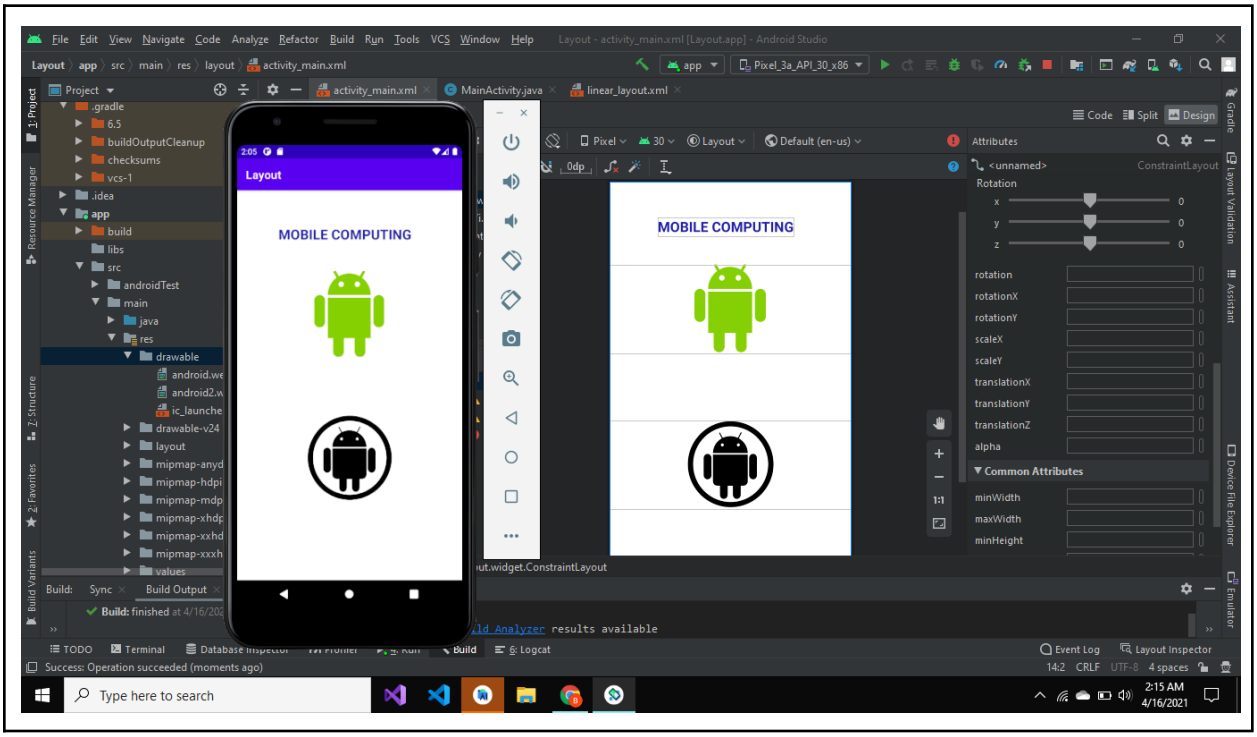


❖ Adding Image in Android Layout:

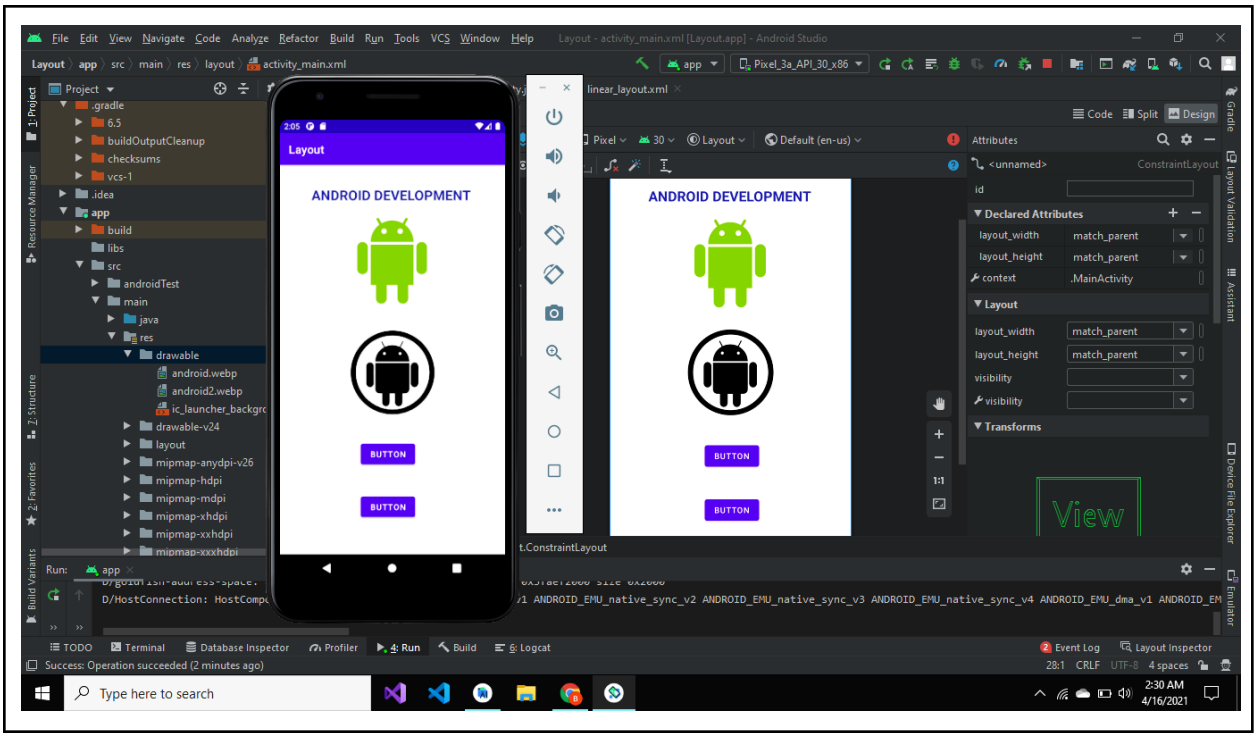
Step-1 In drawable paste any image from your system

Step-2 Add an image view from palette and select the image you want to add.



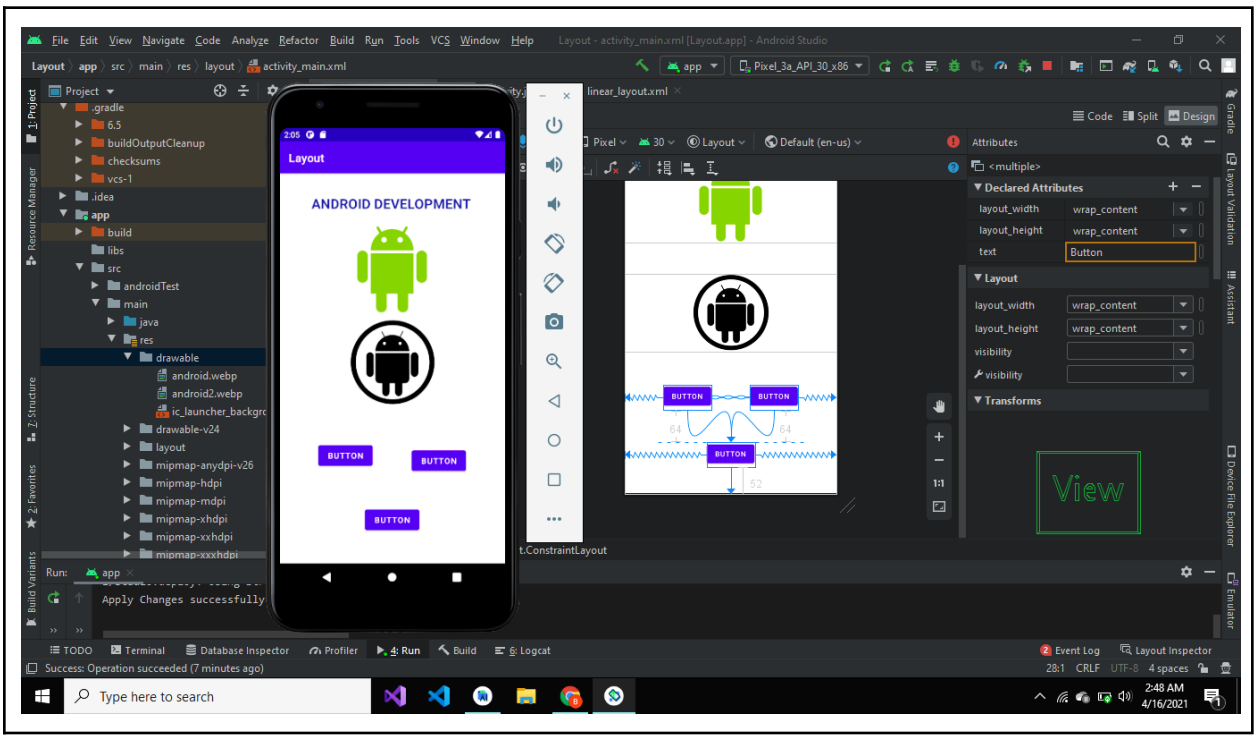


❖ **Adding buttons with alignment:**



❖ Buttons using Layout Constraint Center-Horizontally:

Reminder place details




2:05
Layout
ANDROID DEVELOPMENT
BUTTON
BUTTON
Someone is There
BUTTON


2:05
Layout
ANDROID DEVELOPMENT
BUTTON
BUTTON
Some one new is There
BUTTON

❖ Making Instances for buttons:

Reminder place details

LECTURE#6

❖ Activity:

An Activity is an application component that represents one window, one hierarchy of views.

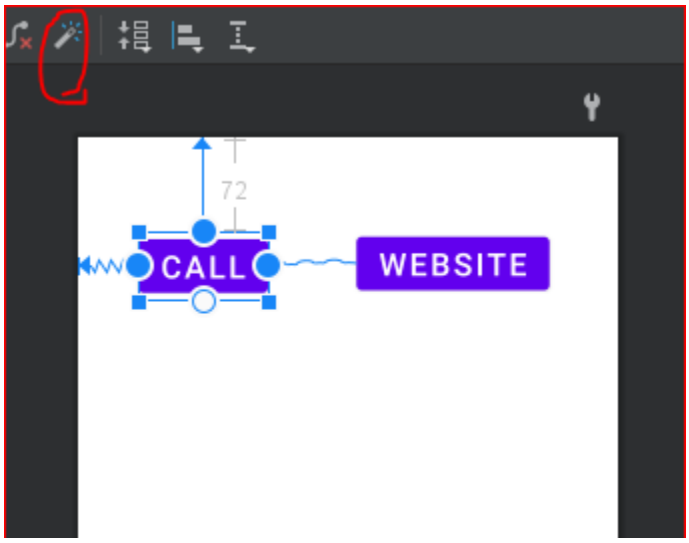
- Typically fills the screen, but can be embedded in other Activity or appear as floating window.
- Java class, typically one Activity in one file.

❖ Intent:

An intent is to perform an action on the screen. It is mostly used to start activity, send broadcast receiver,start services and send message between two activities.

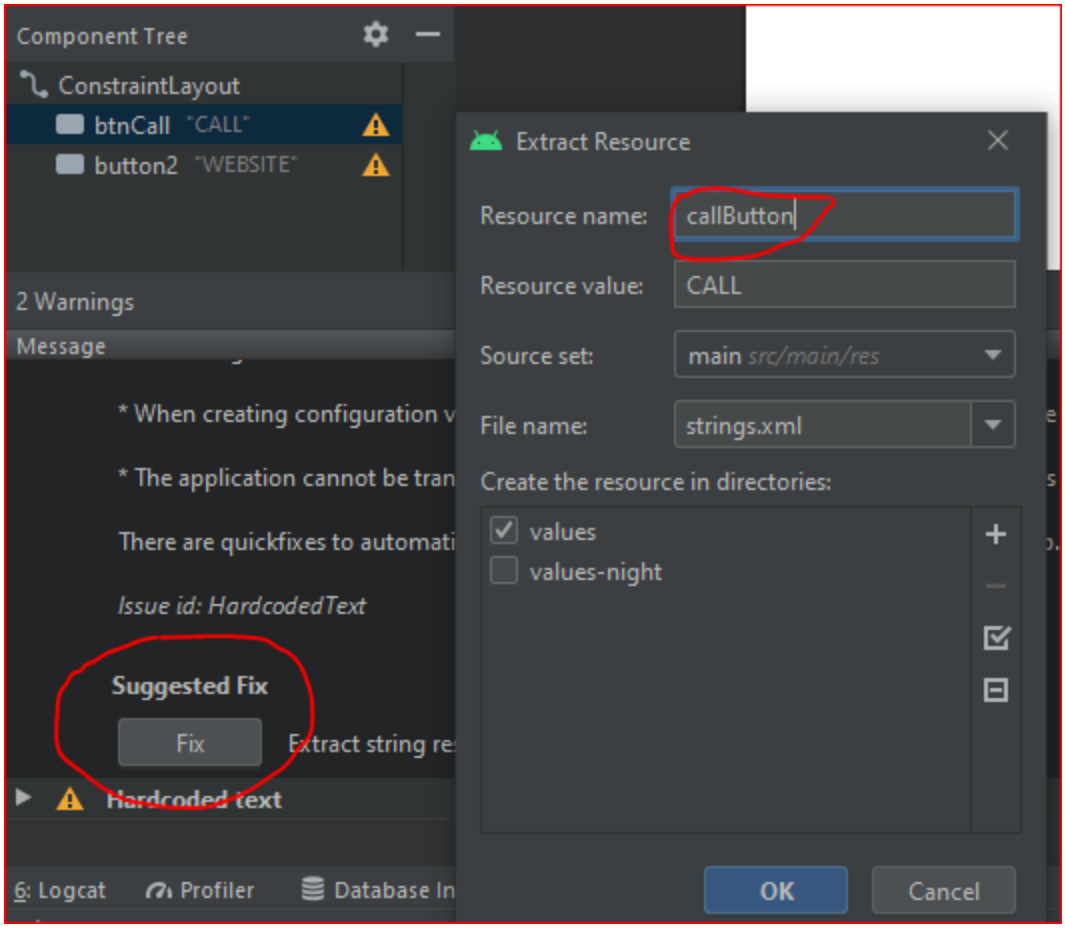
To Set Constraints Automatically:

Step-1 Use the highlighted tool to set constraints automatically.

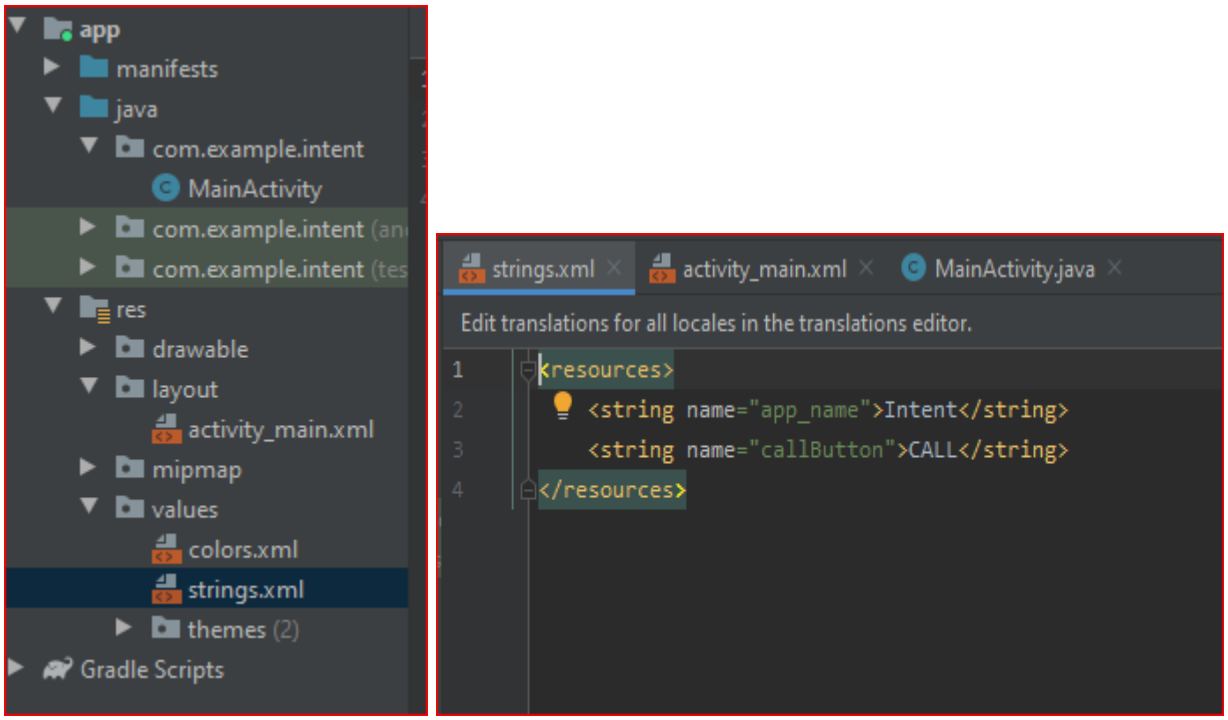


To define String Resources:

Step-2 Click on the fixes button and set a string name for the element.



- **The string name will be defined in string.xml file that is**



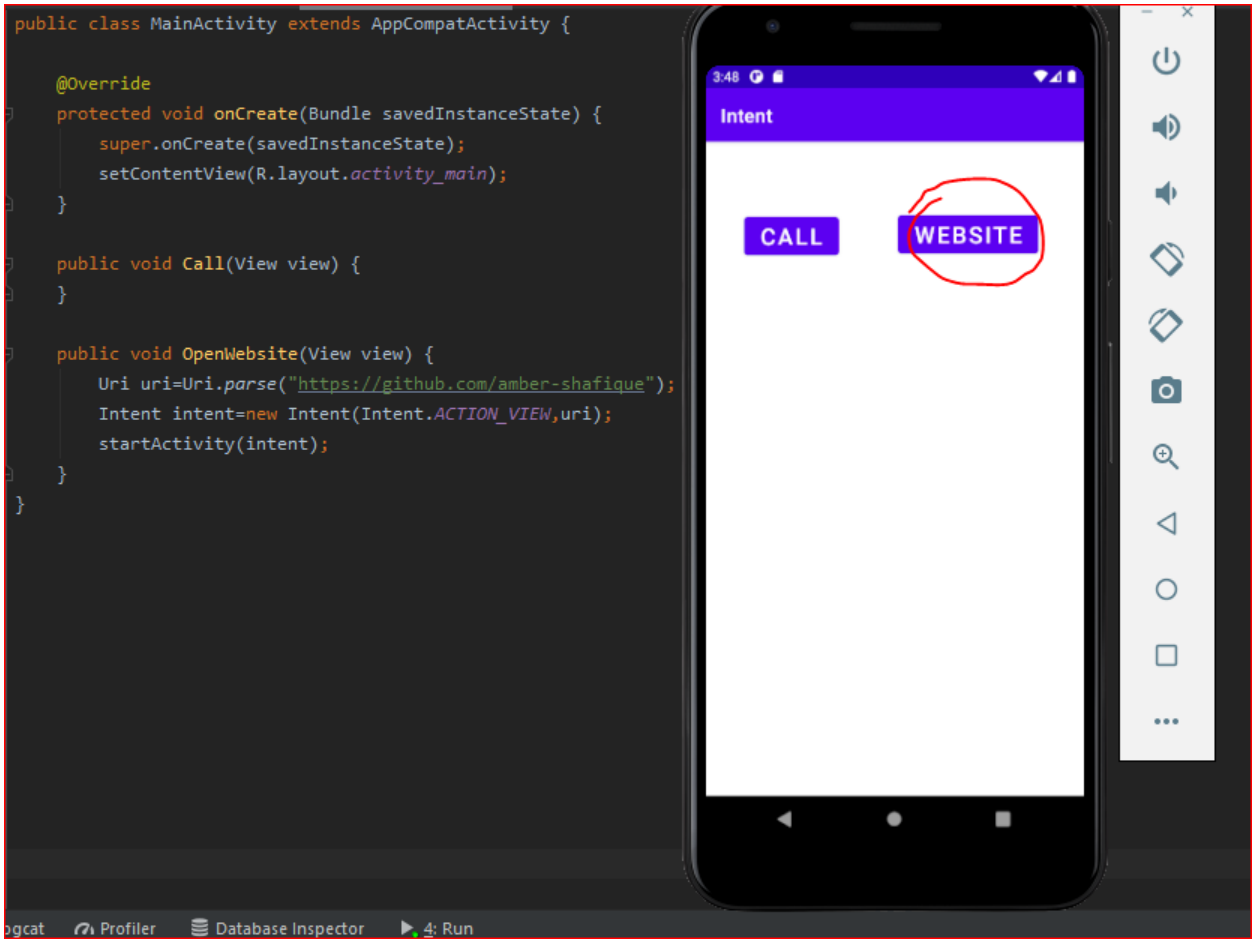
❖ Intent Uses

❖

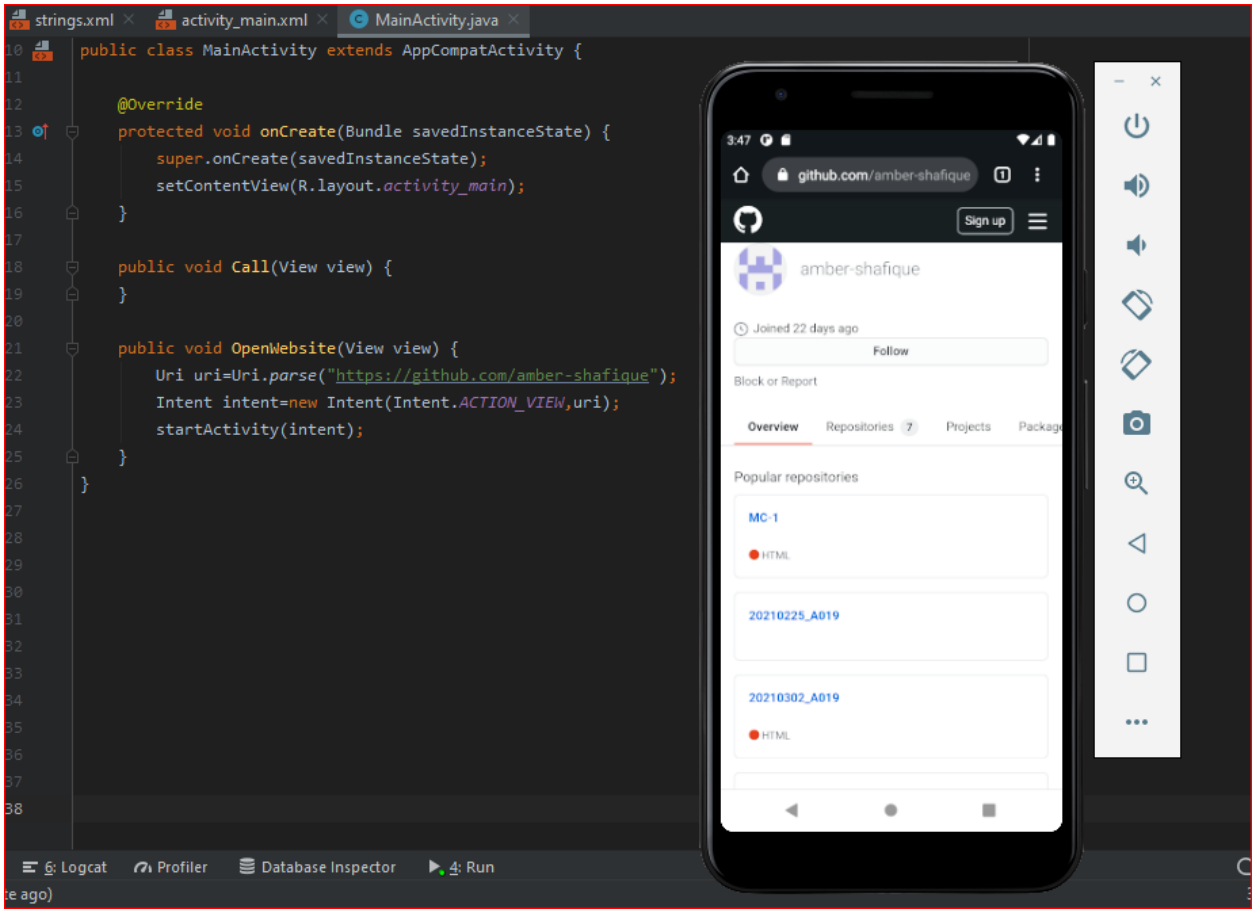
1. To open Website using Intent:

```
Uri uri=Uri.parse("https://github.com/amber-shafique");
Intent intent=new Intent(Intent.ACTION_VIEW,uri);
startActivity(intent);
```

Step-1 Add the above code onClick instance of button.
Step-2 Click on the website button on click of which you have added intent to open website.



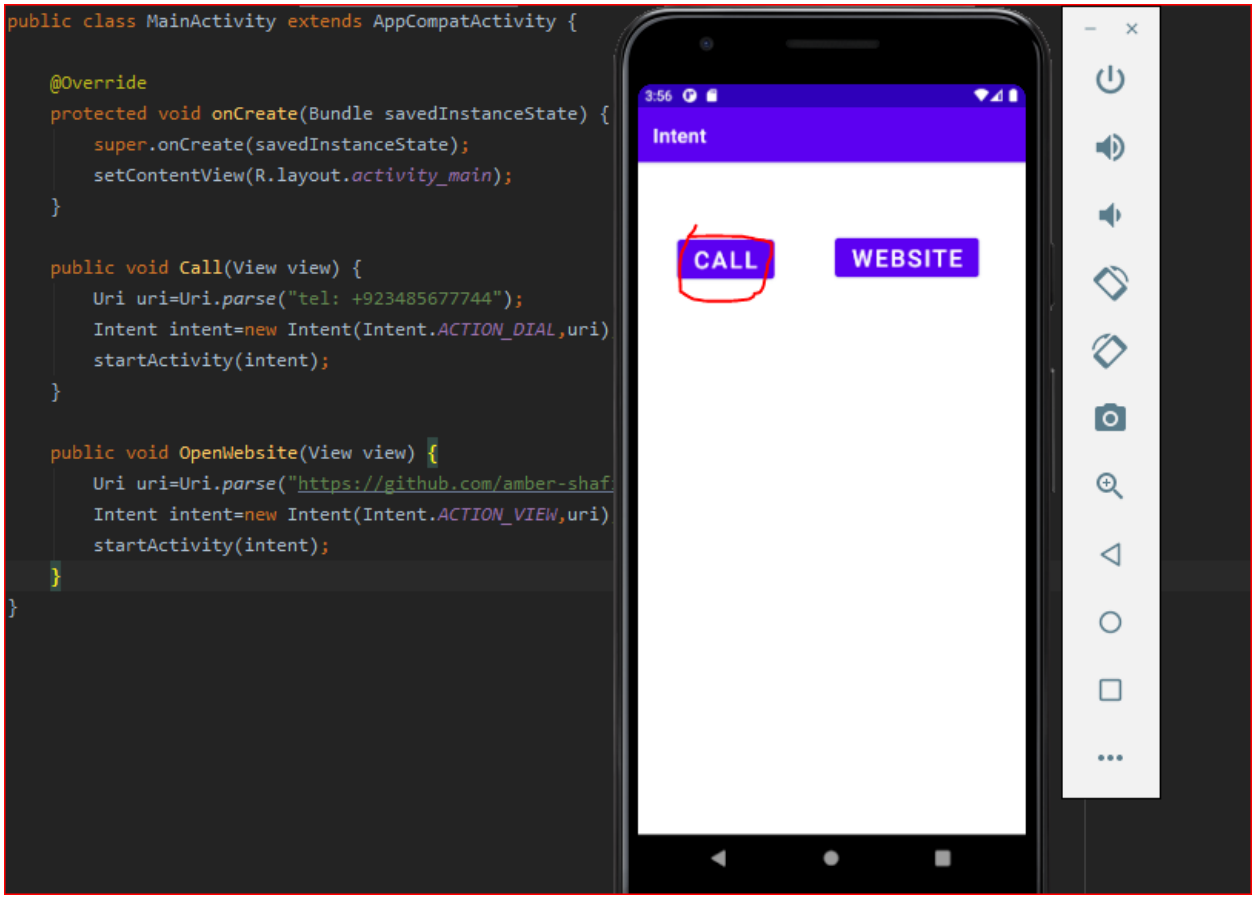
- Just after Clicking the **website** button , the url will open



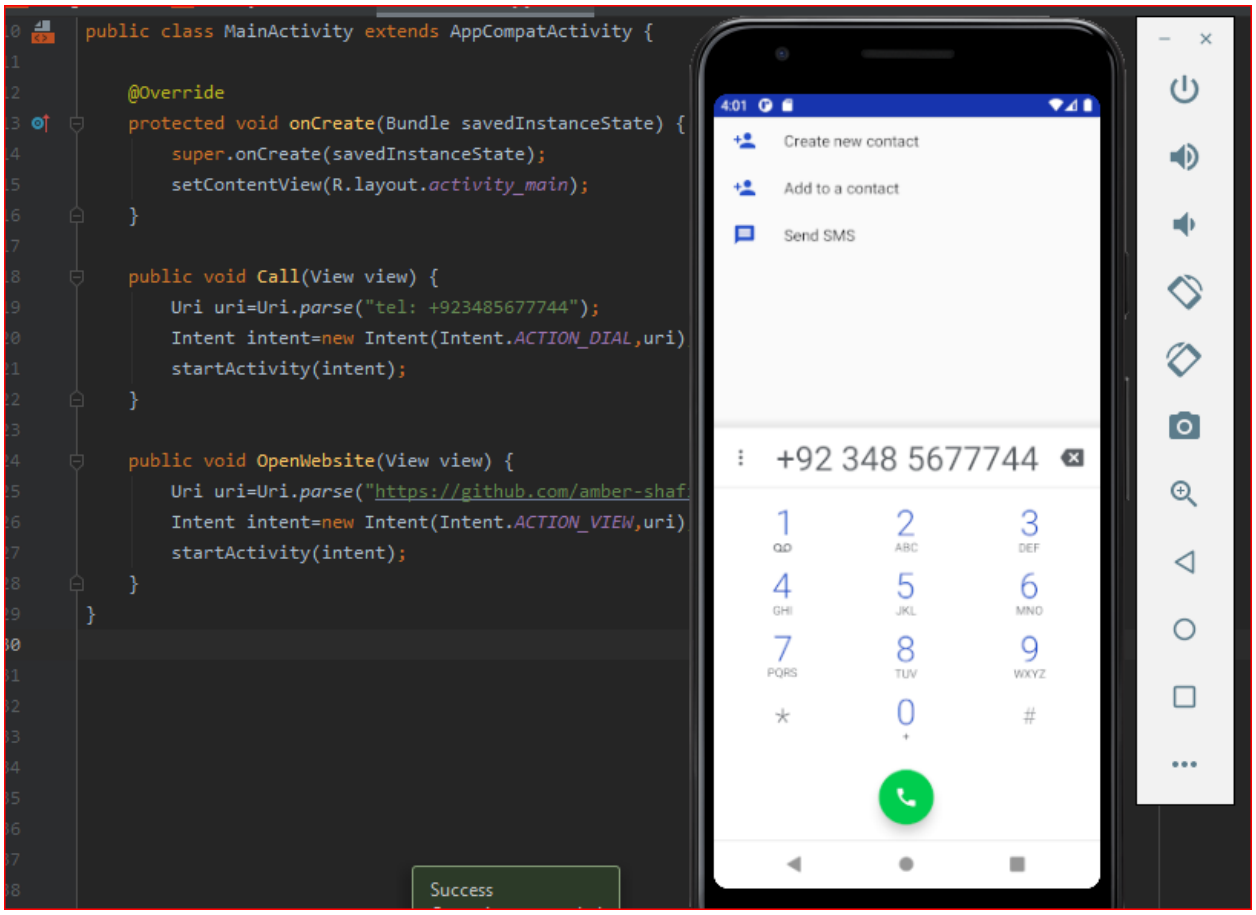
2. To Dial a Call using Intent:

```
Uri uri=Uri.parse("tel: +923485677744");
Intent intent=new Intent(Intent.ACTION_DIAL,uri);
startActivity(intent);
```

Step-1 Add the above code onClick instance of button.
Step-2 Click on the website button on click of which you have added intent to open call dialer.

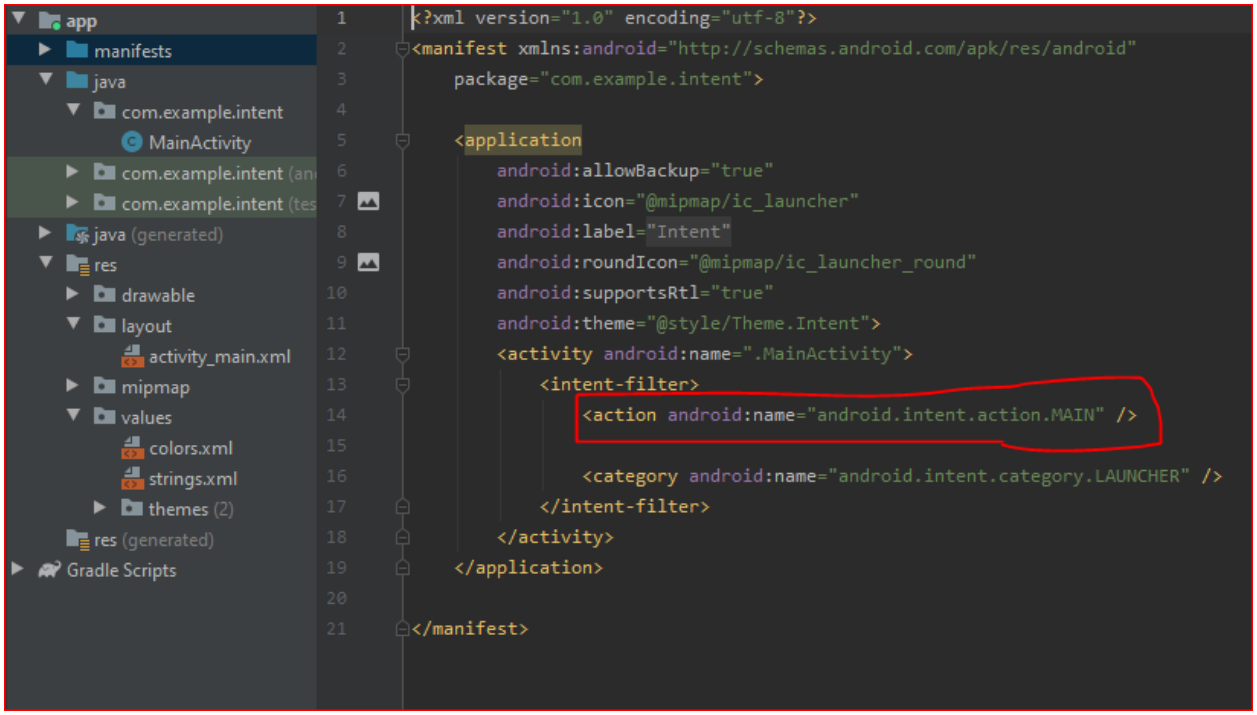


- Just after Clicking the **call** button , the call dialer will open

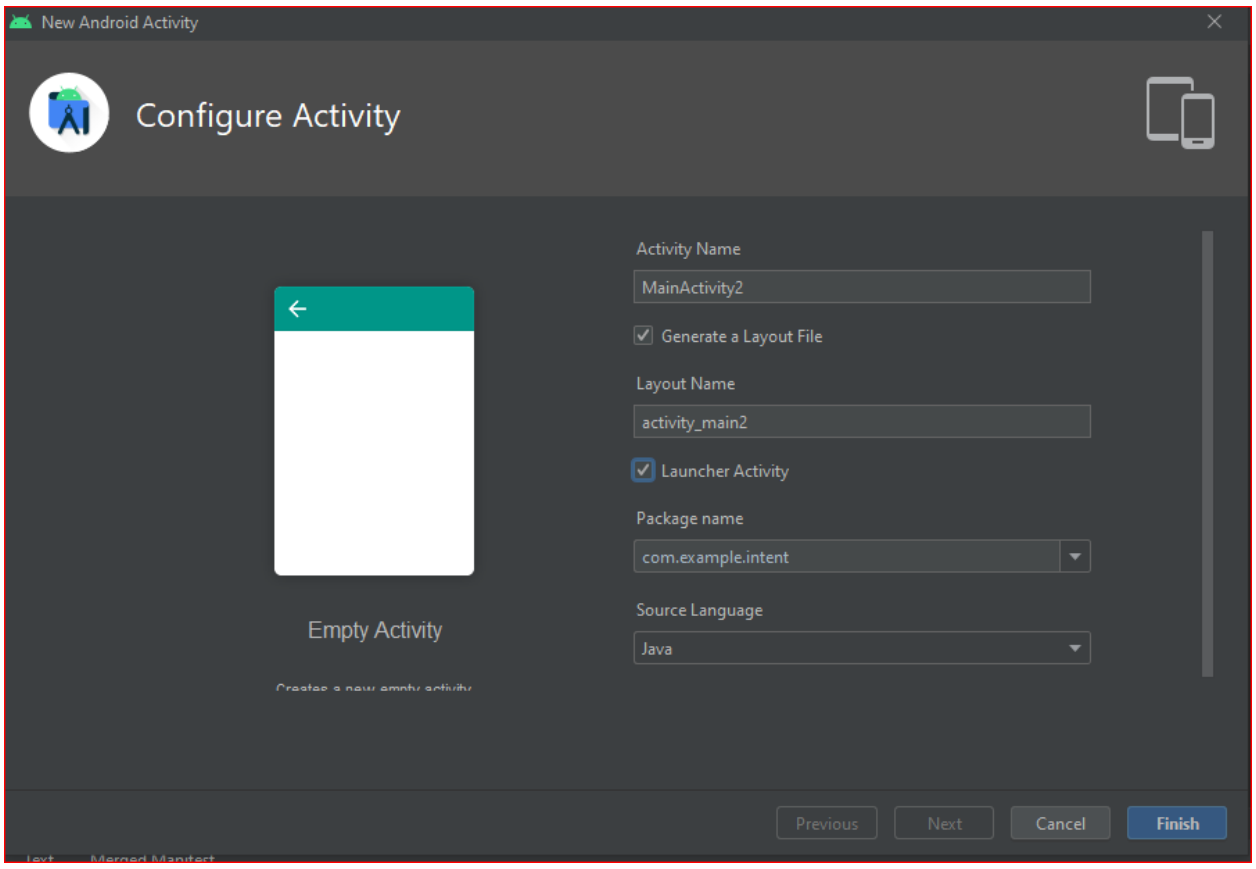


- **Setting another Activity as Launcher Activity:**

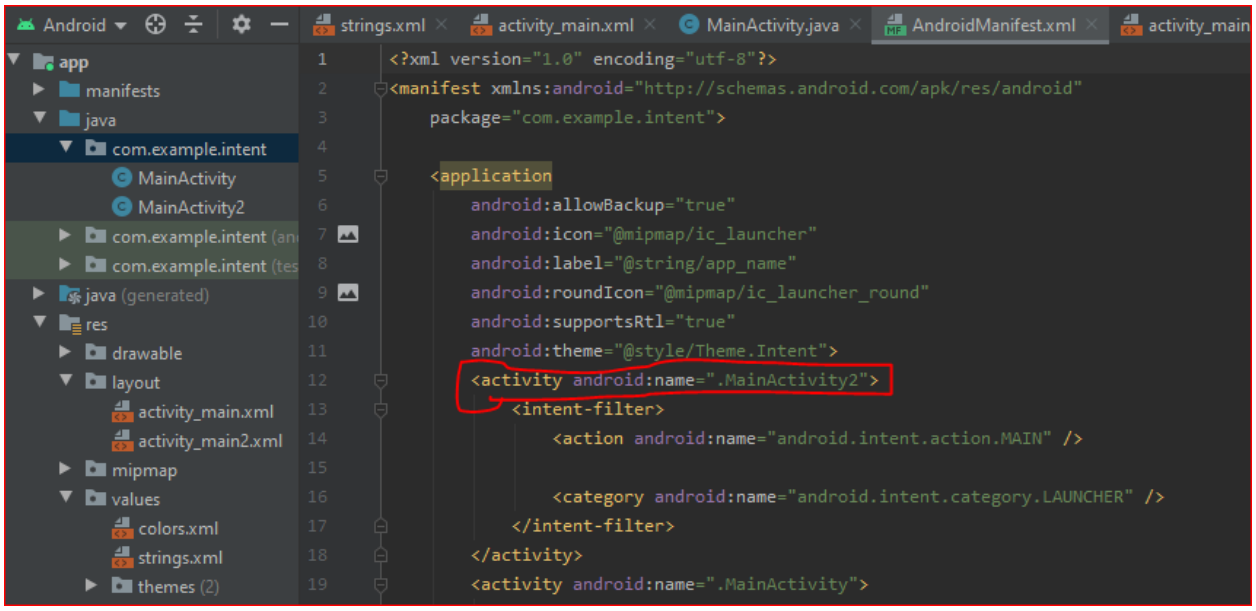
  By default launching activity in **AndoidManifest.xml** file.



Create a new activity and check its attribute of **Launcher Activity.**



Launcher Activity changes in **AndoidManifest.xml** file



3. To move from one activity to another:

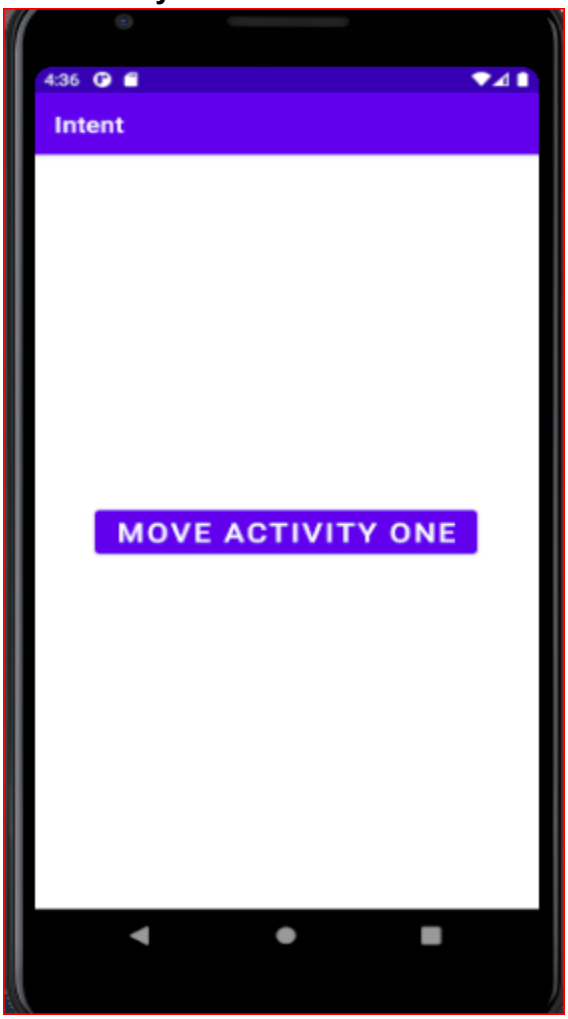
this=> present activity
ActivitName.class=> activity where to move

```
Intent intent=new Intent(this, MainActivity.class);
startActivity(intent);
```

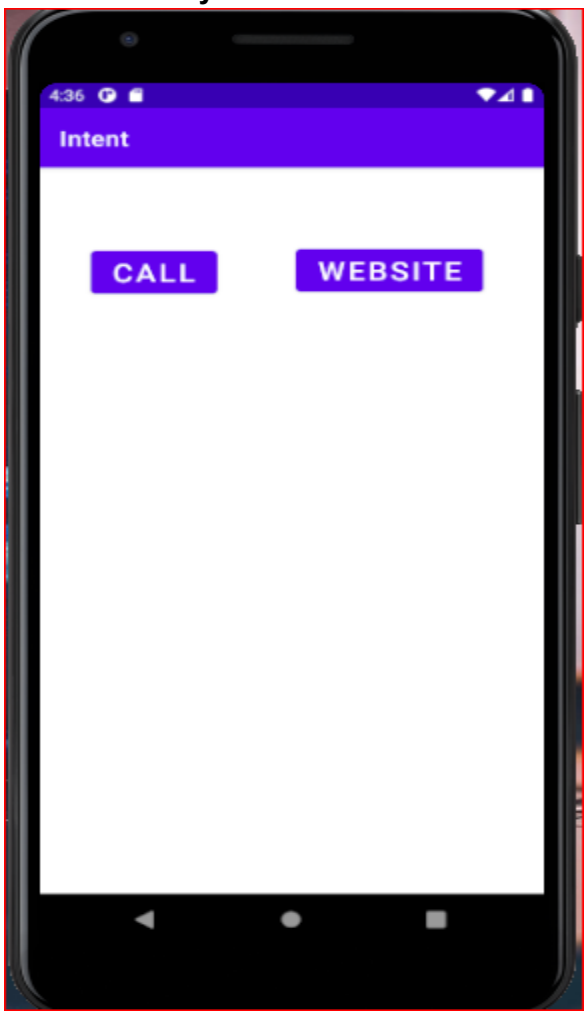
Step-1 Add the above code onClick instance of button.
Step-2 Click on the website button on click of which you have added intent to open another activity..

- Just after Clicking the **Move Activity One** button , the second activity will open.

First activity



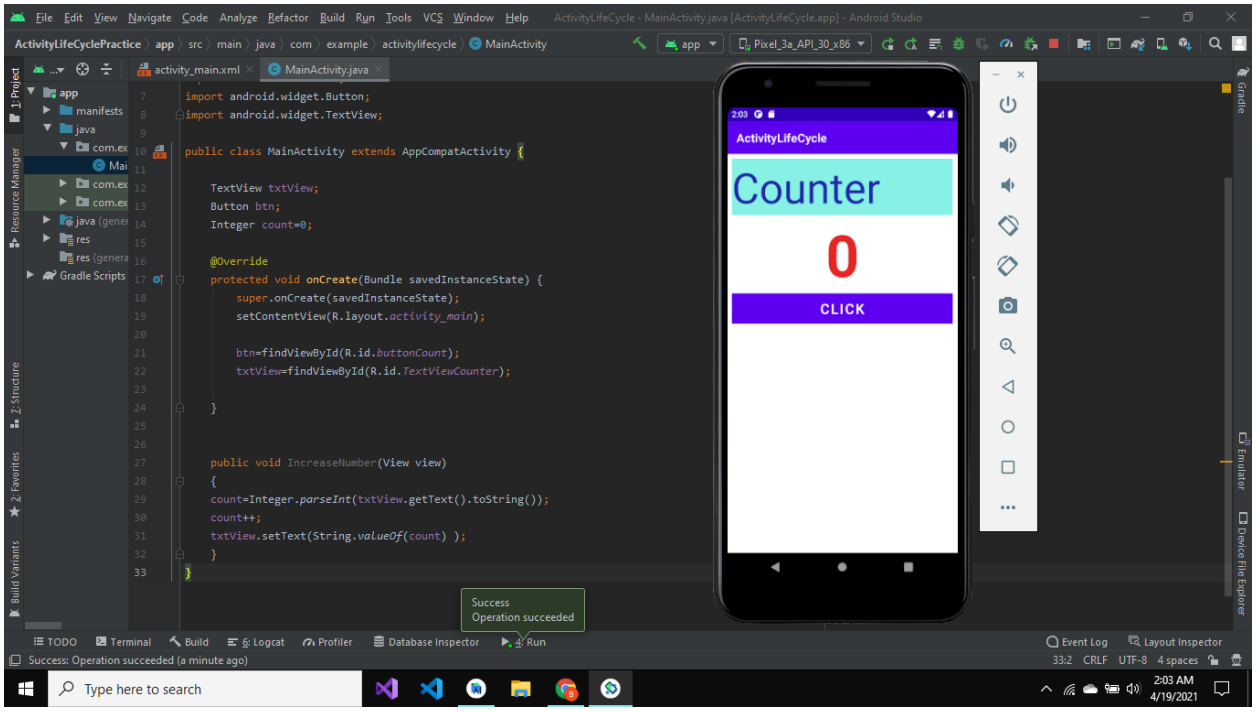
Second activity



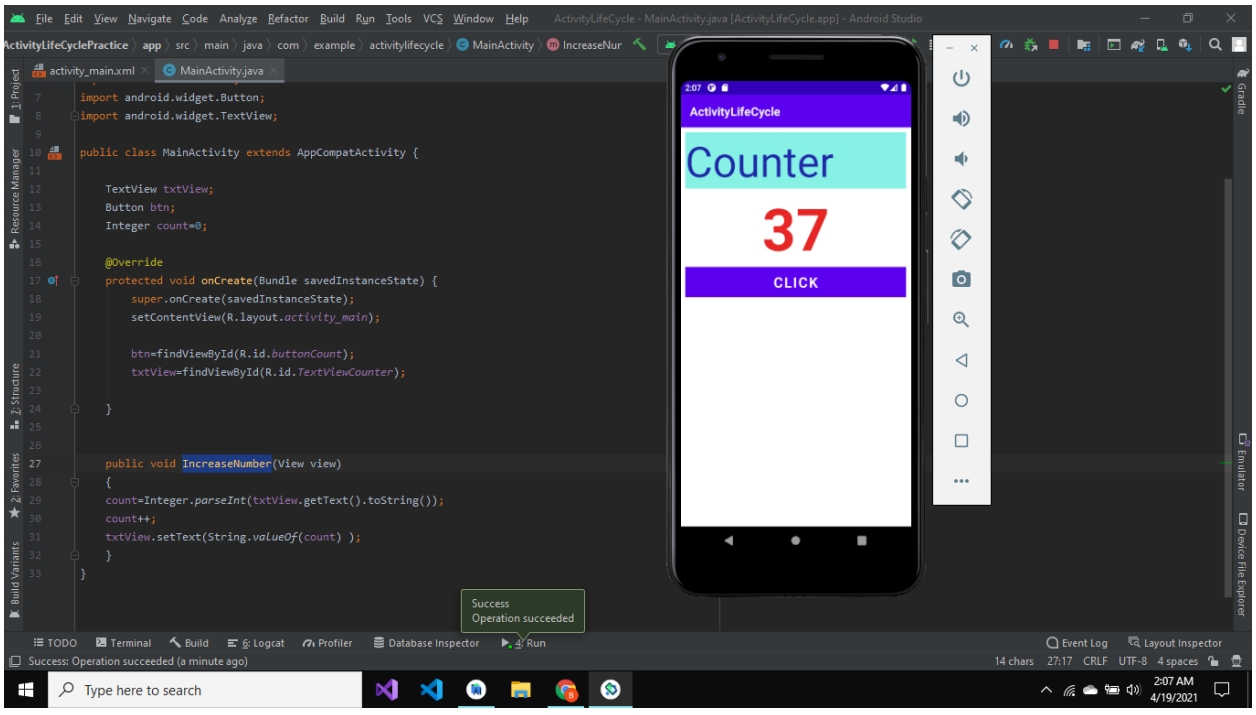
LECTURE#7

❖ Counter:

Adding code to display a counter on screen.

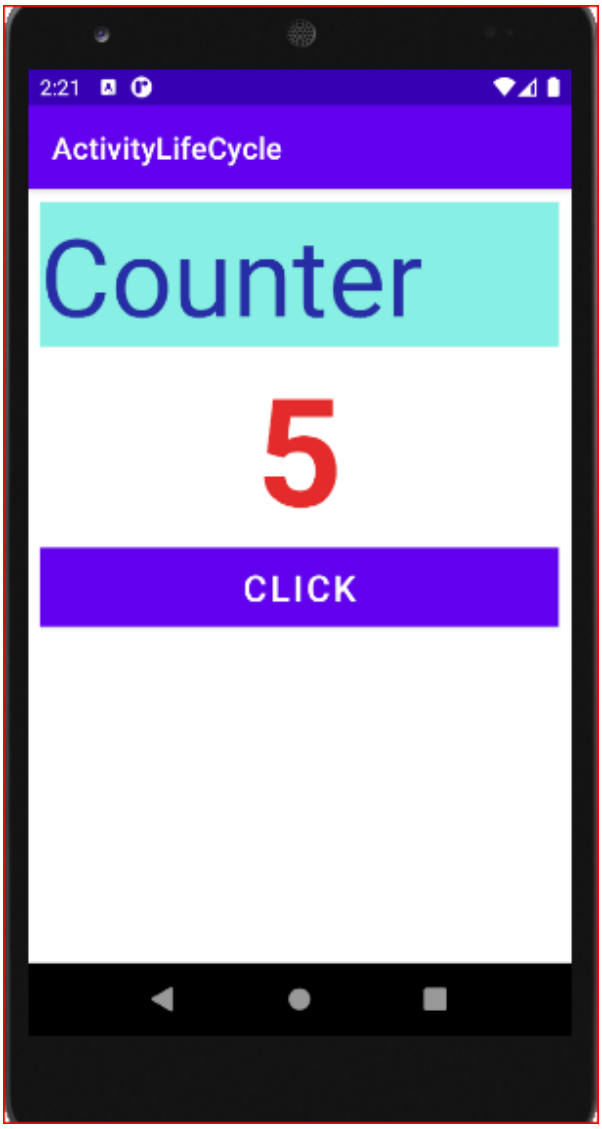


Click on the button to increase value of counter and see the result.



Rotation-Action Problem:

When we rotate the screen counter values goes back to zero

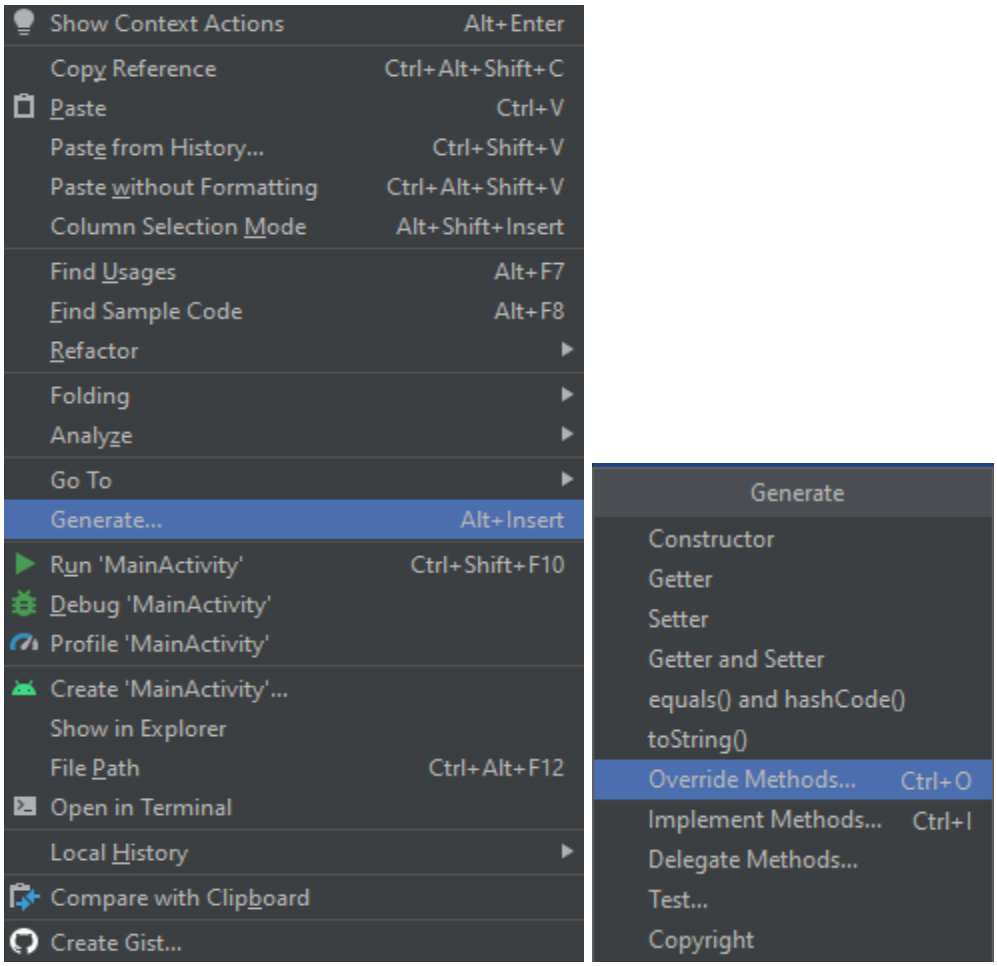
Before Rotation:	**After Rotation:**
	

❖ Activity LifeCycle

- **Create a LOG**

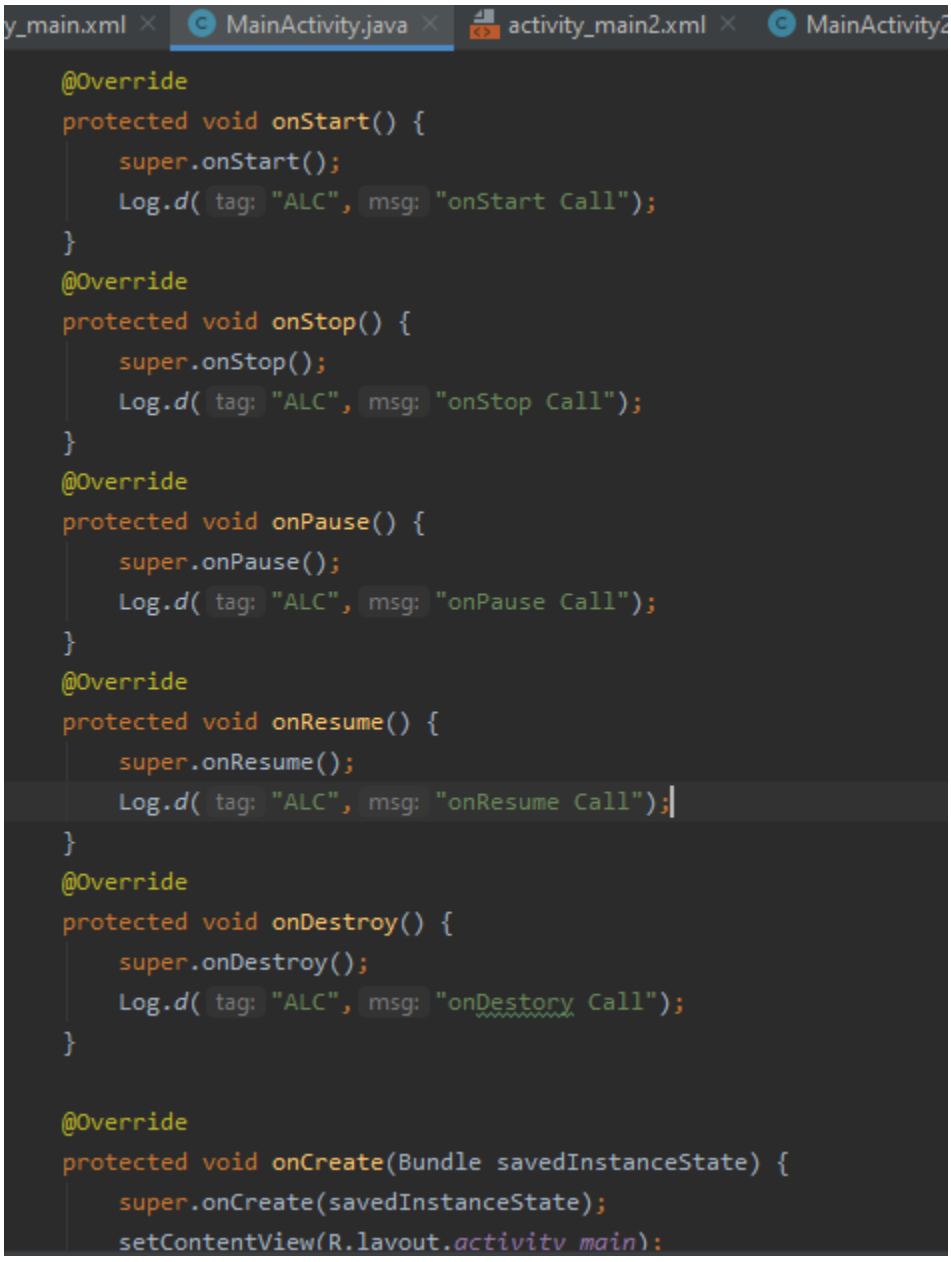
```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    btn=findViewById(R.id.buttonCount);
    txtView=findViewById(R.id.TextViewCounter);

    Log.d( tag: "ALC", msg: "onCreate Call");
}
```

- **To Override methods:**



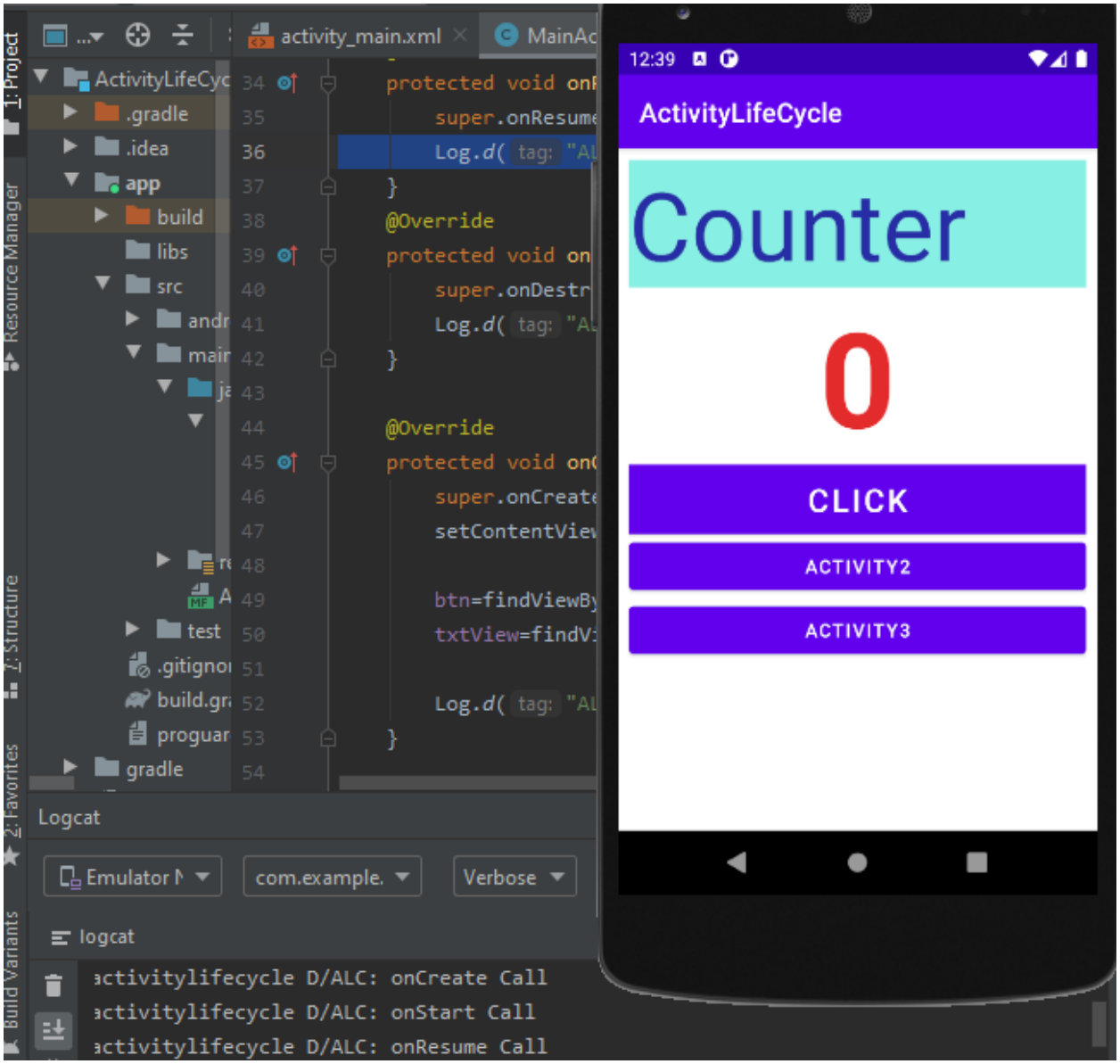
Override the 6 methods in main activity to check their logs in logcat.

- **onCreate()**
- **onStart()**
- **onResume()**
- **on pause()**
- **on Stop()**
- **onDestroy()**

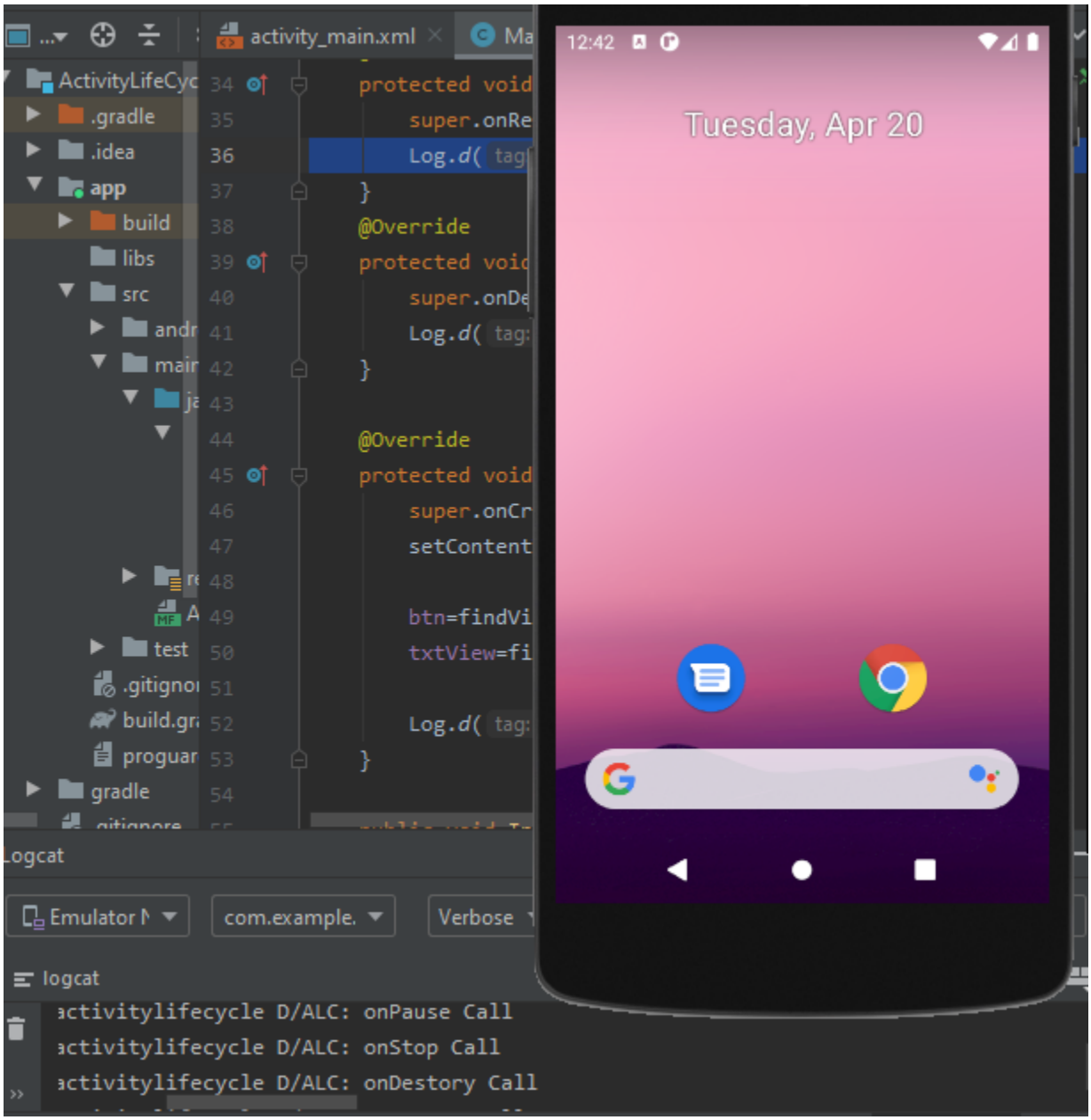


❖ ALC of One Activity:

- Run the emulator and check the logcat.
- On running the application following three funcion will be called as shown in logcat
    1. onCreate()
    2. onStart()
    3. onResume()

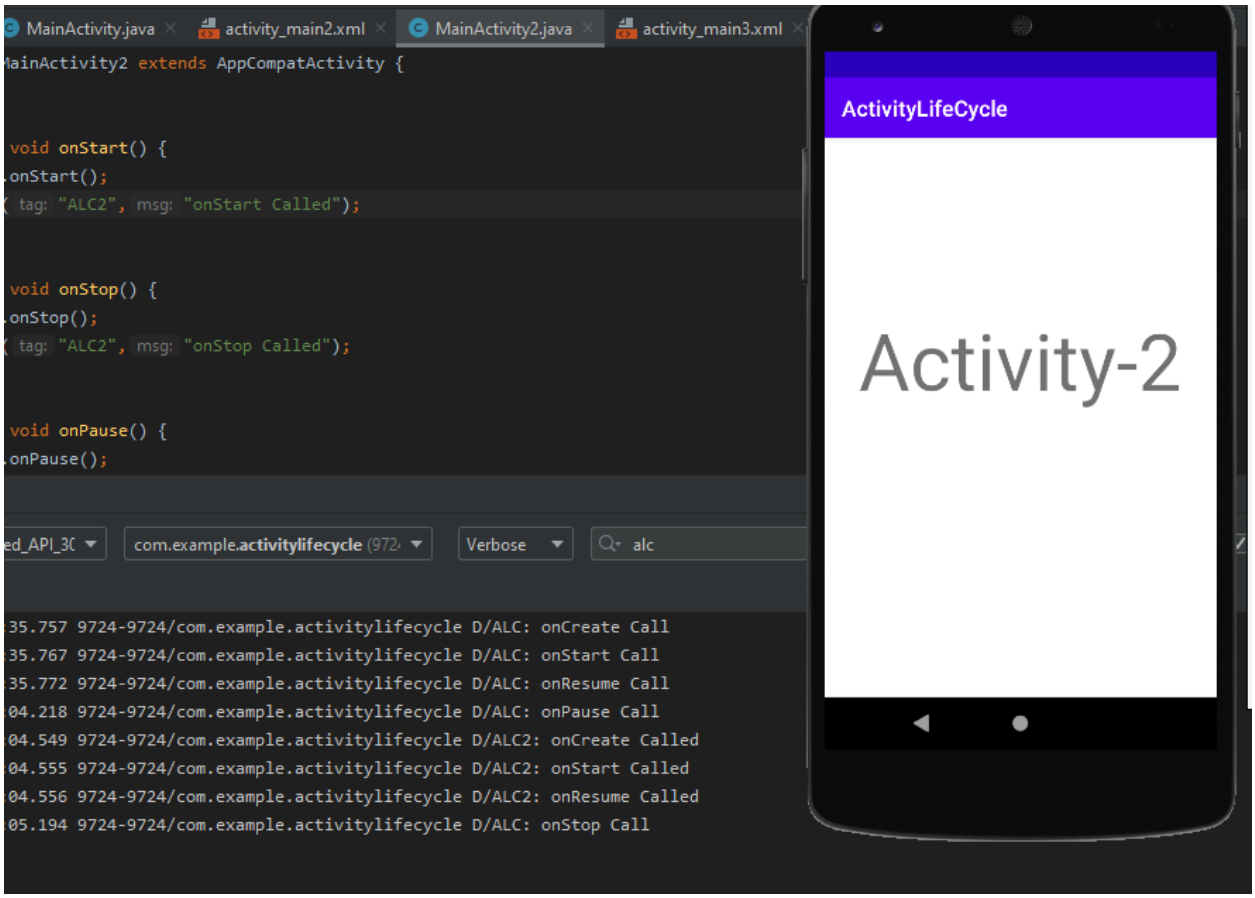


- On just clicking back button on emulator screen following three functions will be called.
    1. on pause()
    2. on Stop()
    3. onDestroy()

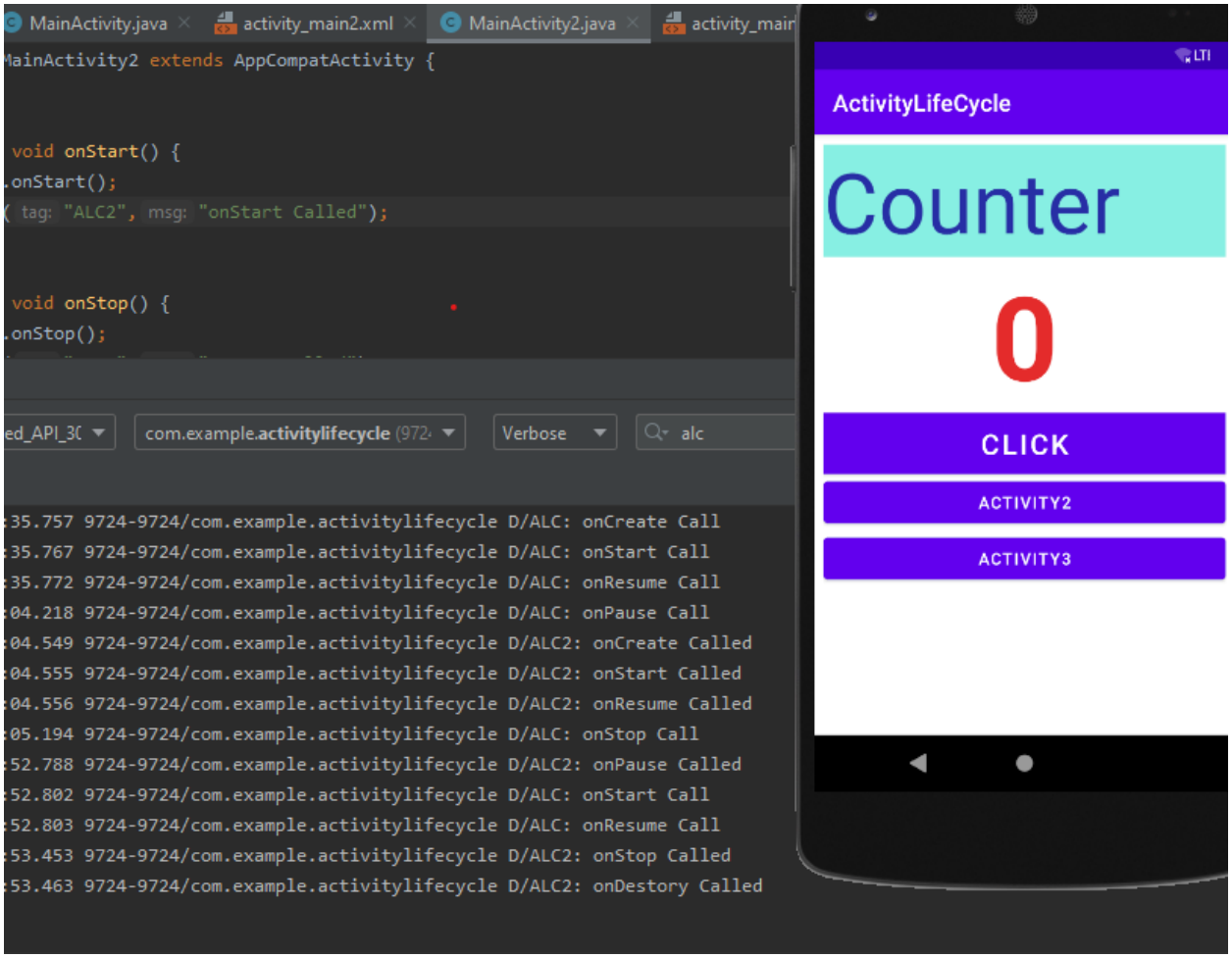


❖ ALC of Main activity to any other activity:

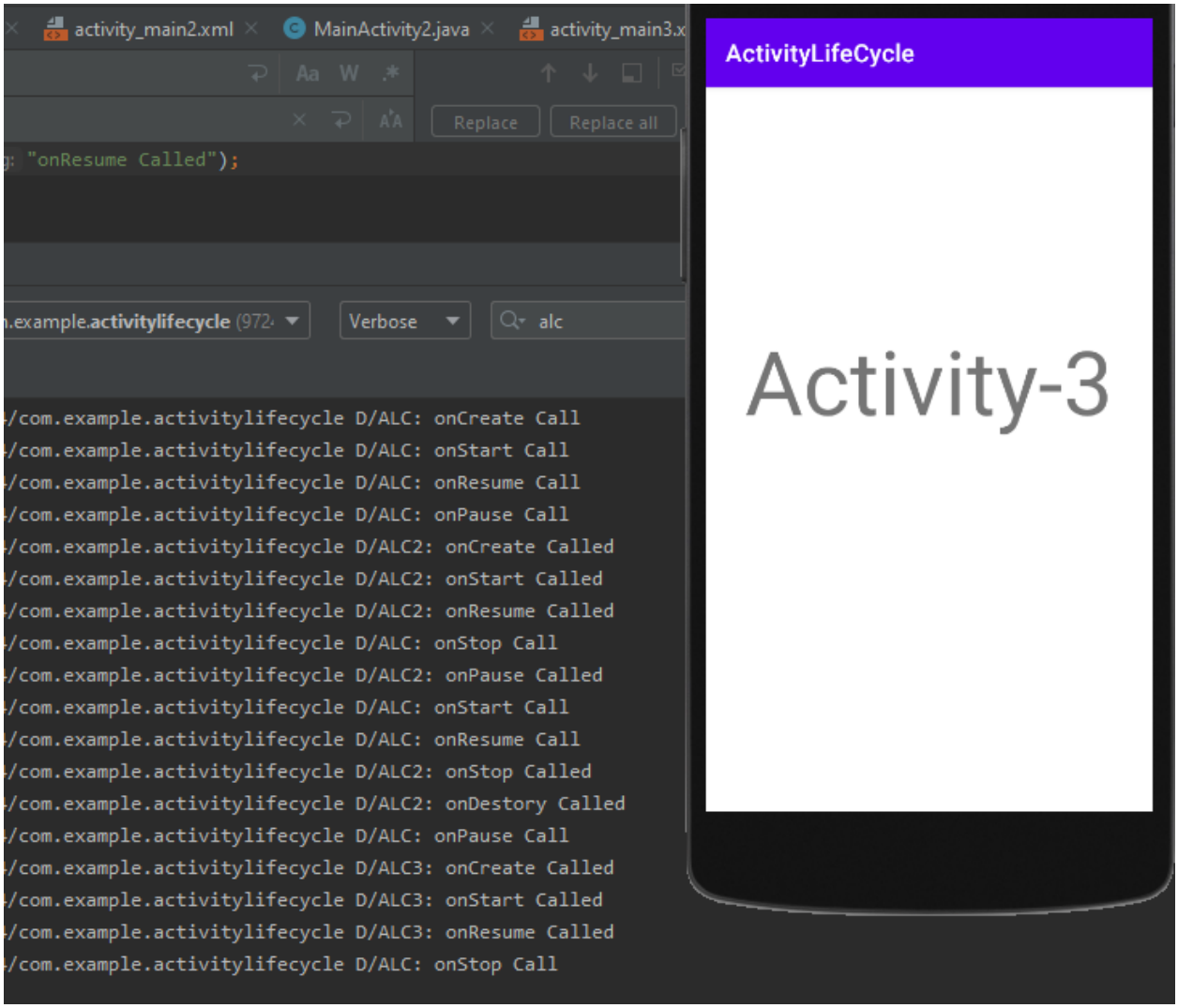
- Override the same 6 functions in activity2.
- Rebuild the application and this time after opening the main activity navigate to second activity and see the logcat.
- Logcat is displaying functions calls by tag names as:
  - ALC => for Main Activity
  - ALC2 => for second activity when clicked



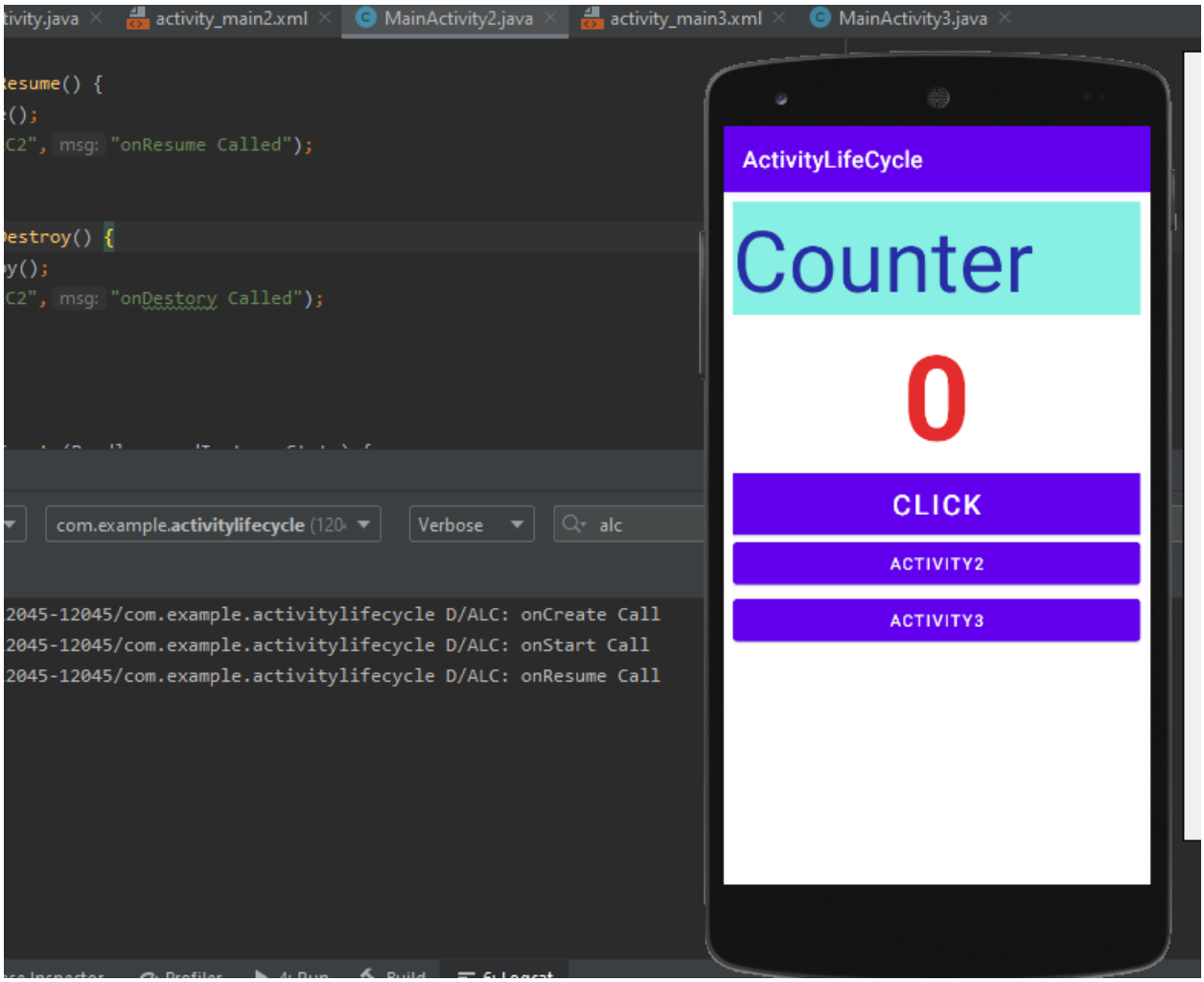
- Now from second activity click back to move back to main activity and see the logcat.

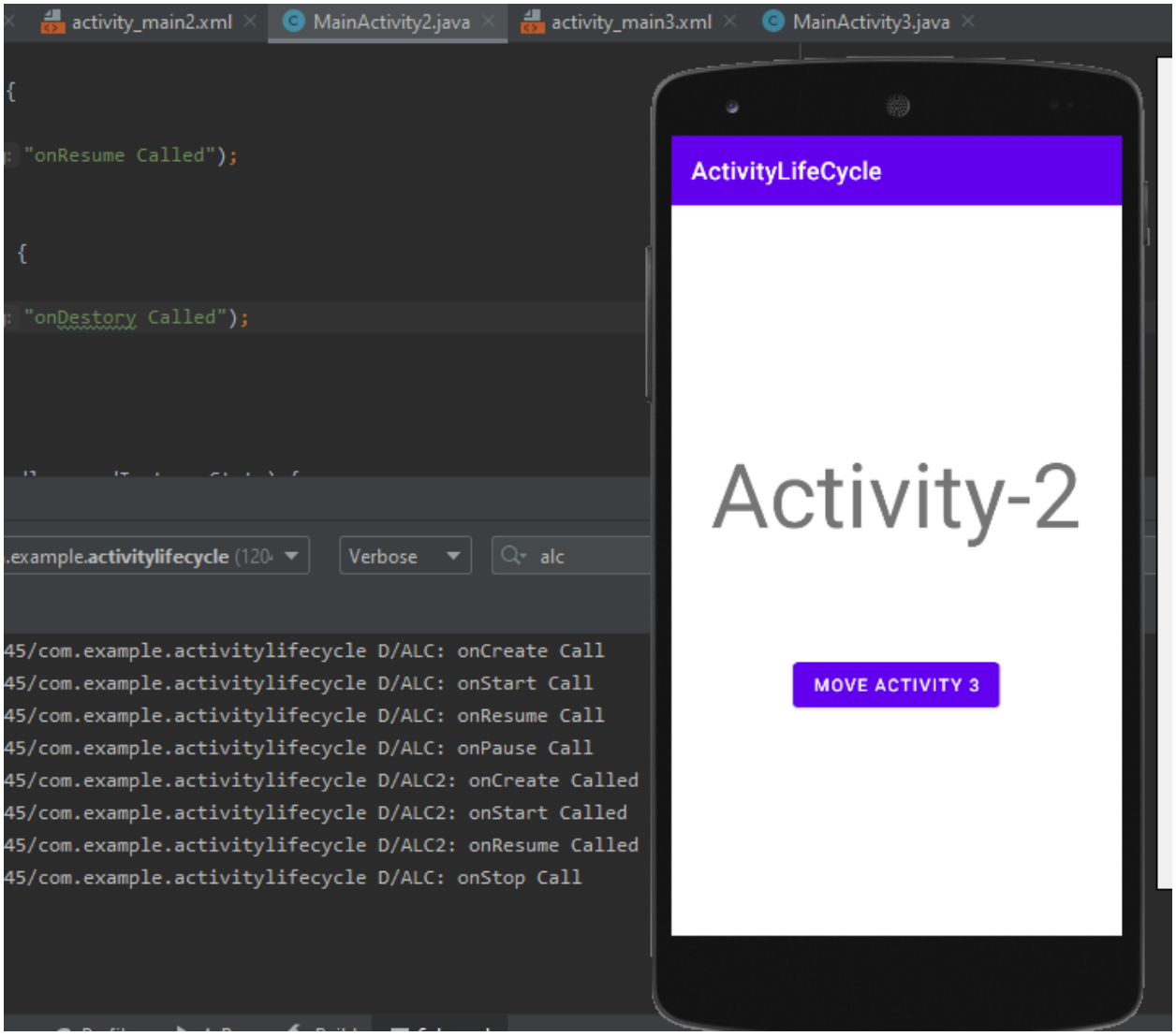


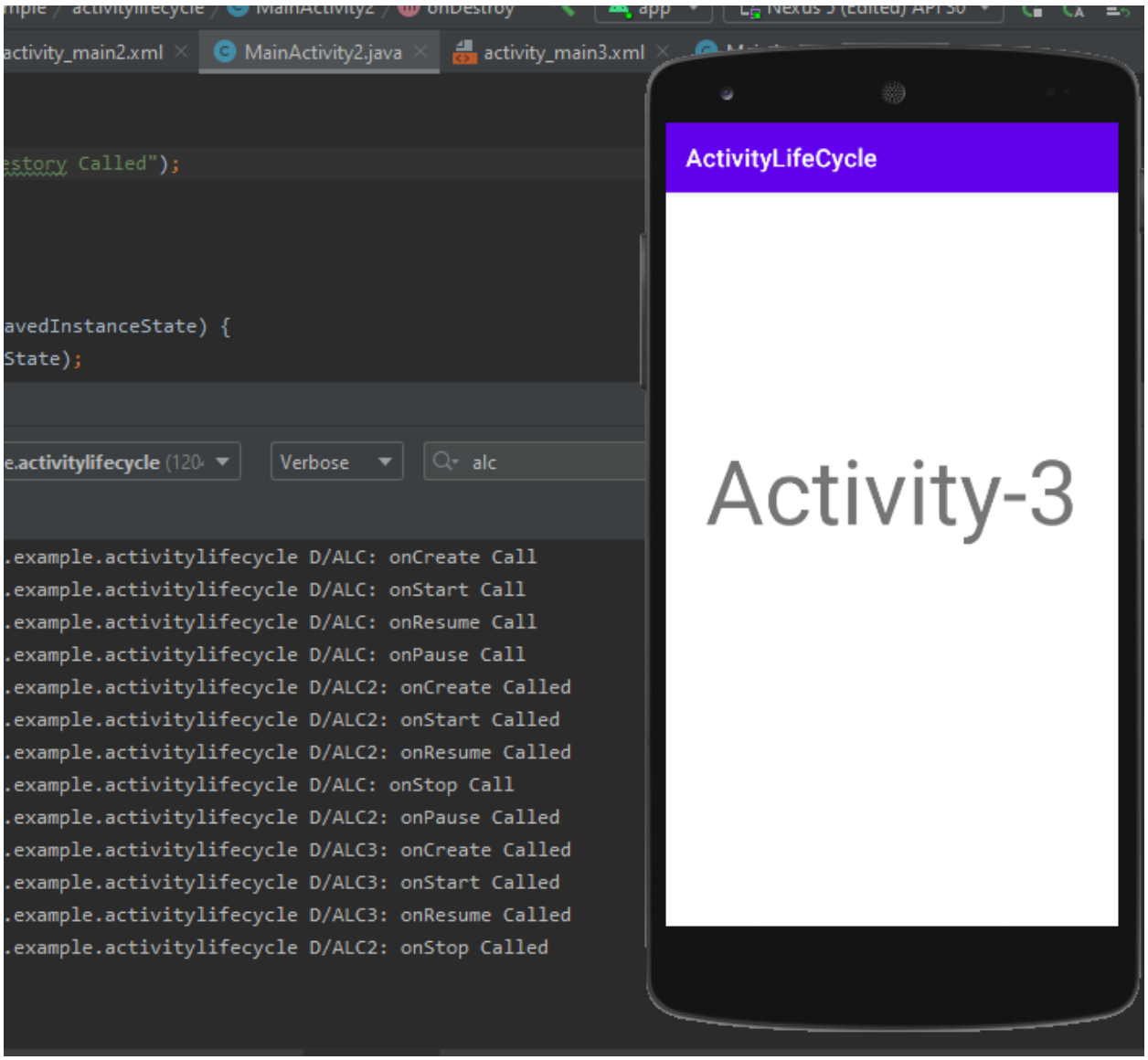
Repeating same steps for activity 3,navigate from main to 2nd activity then back to main and move to Activity-3. Then see the logcat.

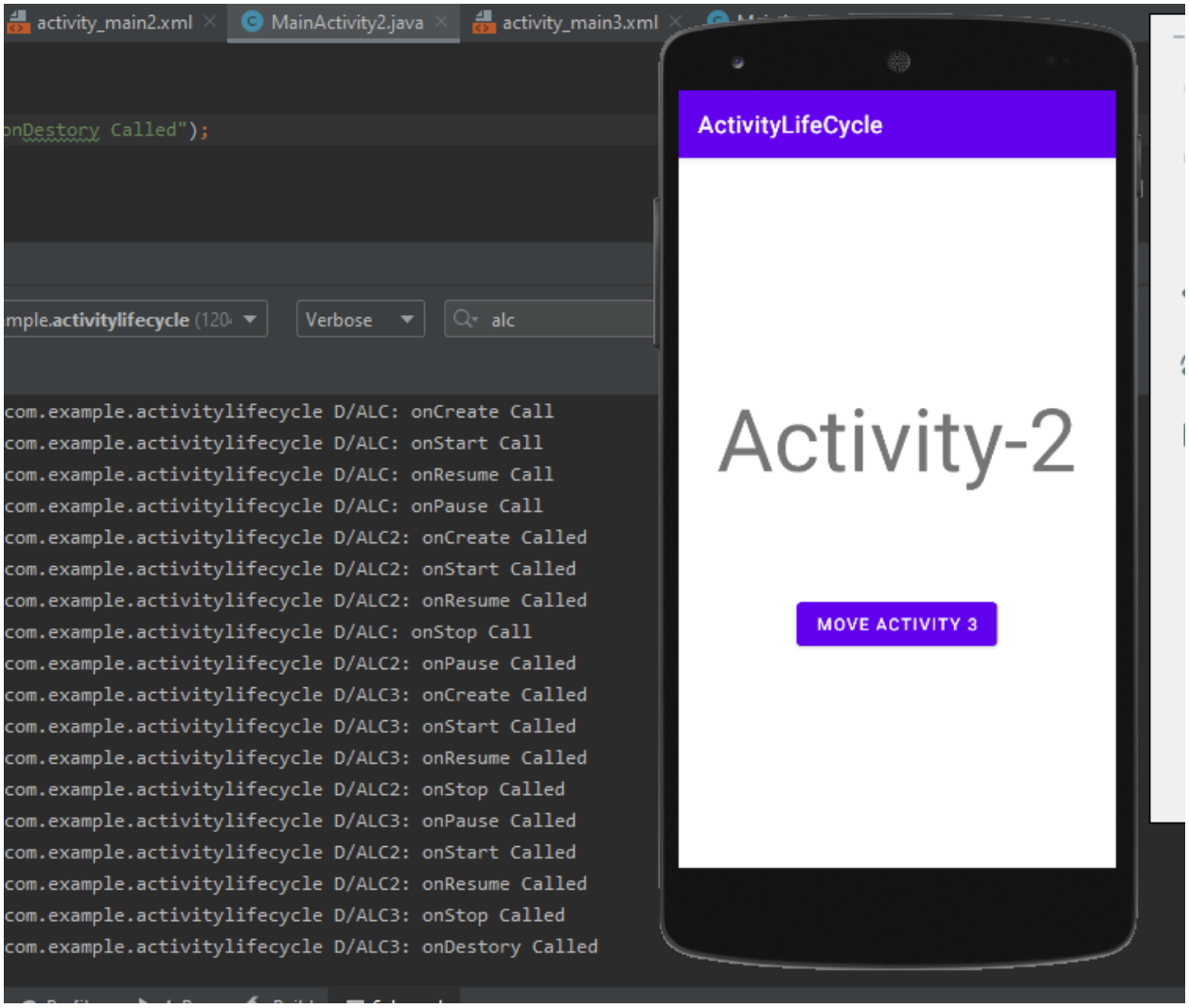


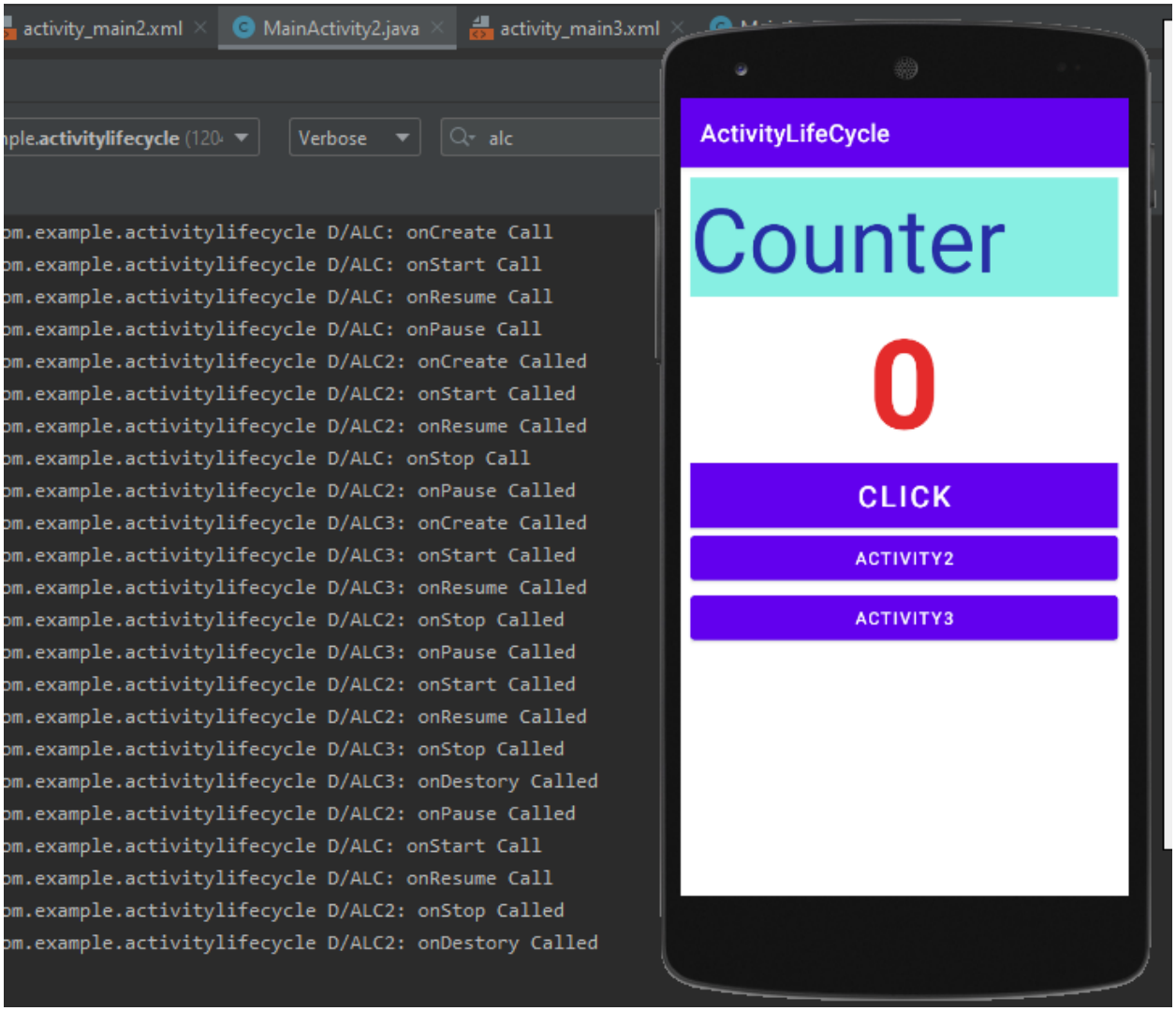
❖ **ALC of Main activity to other activity from other to another activity:**




activity_main2.xml
MainActivity2.java
activity_main3.xml
MainActivity3.java
"onResume Called");
"onDestory Called");
.example.activitylifecycle (120
Verbose
alc
45/com.example.activitylifecycle D/ALC: onCreate Call
45/com.example.activitylifecycle D/ALC: onStart Call
45/com.example.activitylifecycle D/ALC: onResume Call
45/com.example.activitylifecycle D/ALC: onPause Call
45/com.example.activitylifecycle D/ALC2: onCreate Called
45/com.example.activitylifecycle D/ALC2: onStart Called
45/com.example.activitylifecycle D/ALC2: onResume Called
45/com.example.activitylifecycle D/ALC: onStop Call
ActivityLifeCycle
Activity-2
MOVE ACTIVITY 3

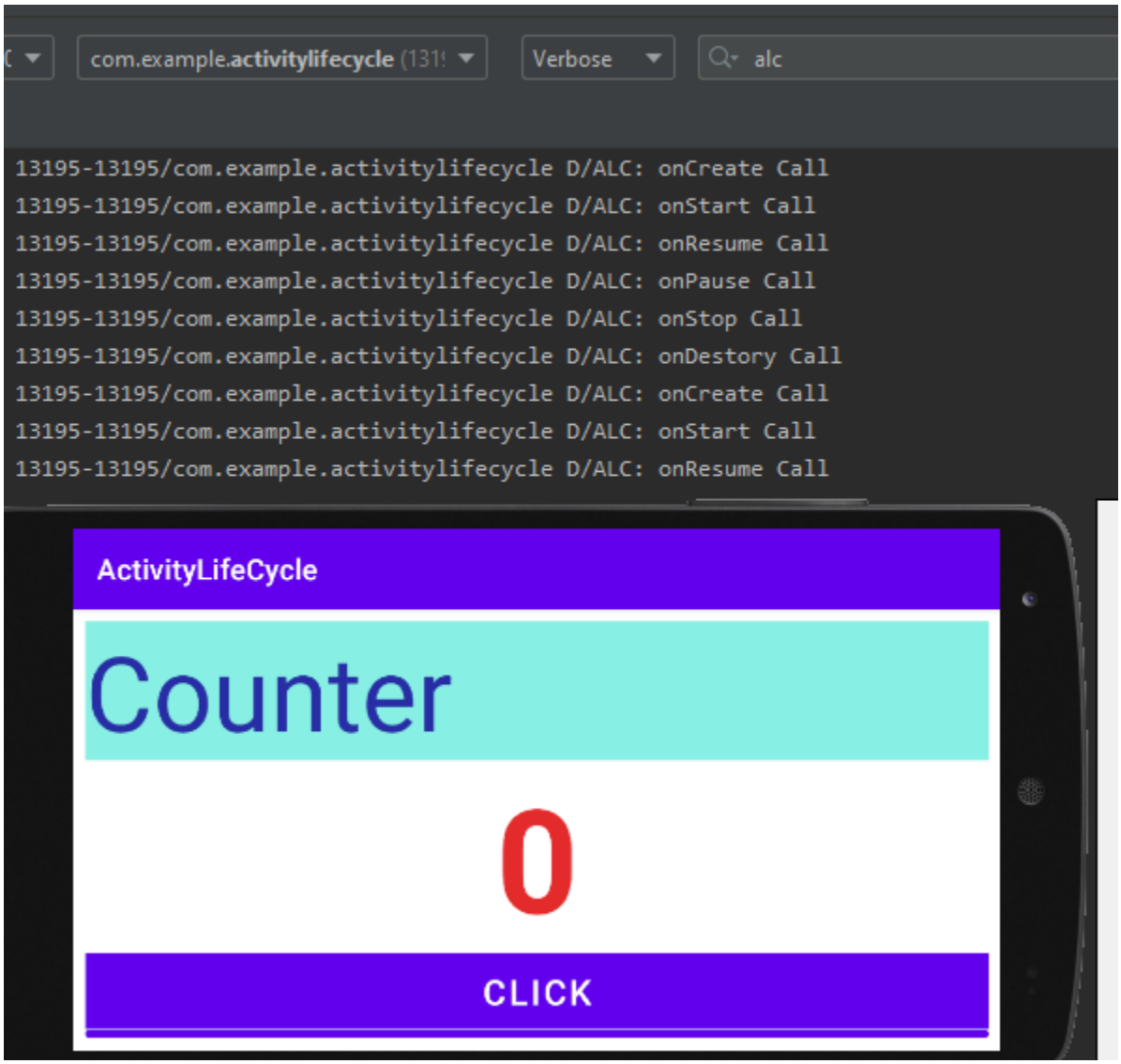

activity_main2.xml
MainActivity2.java
activity_main3.xml
estory Called");
avedInstanceState) {
State);
e.activitylifecycle (120
Verbose
alc
.example.activitylifecycle D/ALC: onCreate Call
.example.activitylifecycle D/ALC: onStart Call
.example.activitylifecycle D/ALC: onResume Call
.example.activitylifecycle D/ALC: onPause Call
.example.activitylifecycle D/ALC2: onCreate Called
.example.activitylifecycle D/ALC2: onStart Called
.example.activitylifecycle D/ALC2: onResume Called
.example.activitylifecycle D/ALC: onStop Call
.example.activitylifecycle D/ALC2: onPause Called
.example.activitylifecycle D/ALC3: onCreate Called
.example.activitylifecycle D/ALC3: onStart Called
.example.activitylifecycle D/ALC3: onResume Called
.example.activitylifecycle D/ALC2: onStop Called
ActivityLifeCycle
Activity-3


activity_main2.xml
MainActivity2.java
activity_main3.xml
onDestory Called");
mple.activitylifecycle (120
Verbose
alc
com.example.activitylifecycle D/ALC: onCreate Call
com.example.activitylifecycle D/ALC: onStart Call
com.example.activitylifecycle D/ALC: onResume Call
com.example.activitylifecycle D/ALC: onPause Call
com.example.activitylifecycle D/ALC2: onCreate Called
com.example.activitylifecycle D/ALC2: onStart Called
com.example.activitylifecycle D/ALC2: onResume Called
com.example.activitylifecycle D/ALC: onStop Call
com.example.activitylifecycle D/ALC2: onPause Called
com.example.activitylifecycle D/ALC3: onCreate Called
com.example.activitylifecycle D/ALC3: onStart Called
com.example.activitylifecycle D/ALC3: onResume Called
com.example.activitylifecycle D/ALC2: onStop Called
com.example.activitylifecycle D/ALC3: onPause Called
com.example.activitylifecycle D/ALC2: onStart Called
com.example.activitylifecycle D/ALC2: onResume Called
com.example.activitylifecycle D/ALC3: onStop Called
com.example.activitylifecycle D/ALC3: onDestory Called
ActivityLifeCycle
Activity-2
MOVE ACTIVITY 3


activity_main2.xml
MainActivity2.java
activity_main3.xml
ple.activitylifecycle (120
Verbose
alc
om.example.activitylifecycle D/ALC: onCreate Call
om.example.activitylifecycle D/ALC: onStart Call
om.example.activitylifecycle D/ALC: onResume Call
om.example.activitylifecycle D/ALC: onPause Call
om.example.activitylifecycle D/ALC2: onCreate Called
om.example.activitylifecycle D/ALC2: onStart Called
om.example.activitylifecycle D/ALC2: onResume Called
om.example.activitylifecycle D/ALC: onStop Call
om.example.activitylifecycle D/ALC2: onPause Called
om.example.activitylifecycle D/ALC3: onCreate Called
om.example.activitylifecycle D/ALC3: onStart Called
om.example.activitylifecycle D/ALC3: onResume Called
om.example.activitylifecycle D/ALC2: onStop Called
om.example.activitylifecycle D/ALC3: onPause Called
om.example.activitylifecycle D/ALC2: onStart Called
om.example.activitylifecycle D/ALC2: onResume Called
om.example.activitylifecycle D/ALC3: onStop Called
om.example.activitylifecycle D/ALC3: onDestory Called
om.example.activitylifecycle D/ALC2: onPause Called
om.example.activitylifecycle D/ALC: onStart Call
om.example.activitylifecycle D/ALC: onResume Call
om.example.activitylifecycle D/ALC2: onStop Called
om.example.activitylifecycle D/ALC2: onDestory Called
ActivityLifeCycle
Counter
0
CLICK
ACTIVITY2
ACTIVITY3

Reason =>Rotation-Action Problem:

- **On screen rotation onDestroy() is called so value of count doesn't get saved and after rotation activity starts again .**



Solution =>Rotation-Action Problem:

- **Save values before destroy**

Add the following method in the activity and save value of count in the bundle.So that value of count can be stored before call of onDestory() and reused later on.

```
@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putInt("value",count);
```

```
}
```

Put an if condition to check instance state and display value stored by bundle.

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

   if (savedInstanceState!=null)
{
    count=savedInstanceState.getInt("value");
    txtView.setText(String.valueOf(count) );
}

}
```

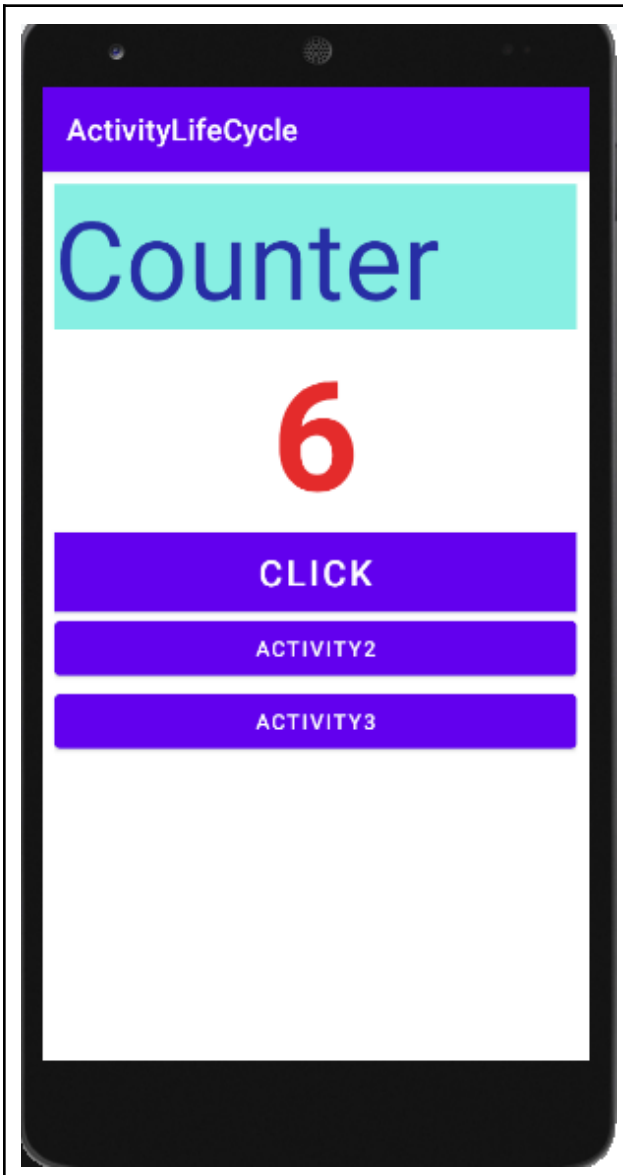
Now check the screen after rotation it will display the count exactly that was before rotation.

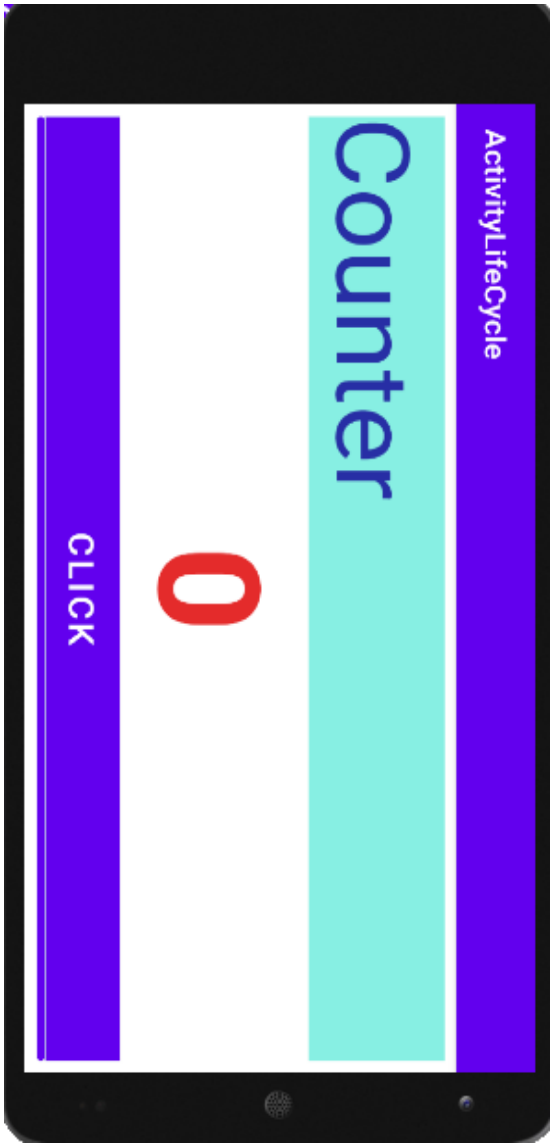
Checking Counter on Activity Navigation:

Now comment the display code in if condition.

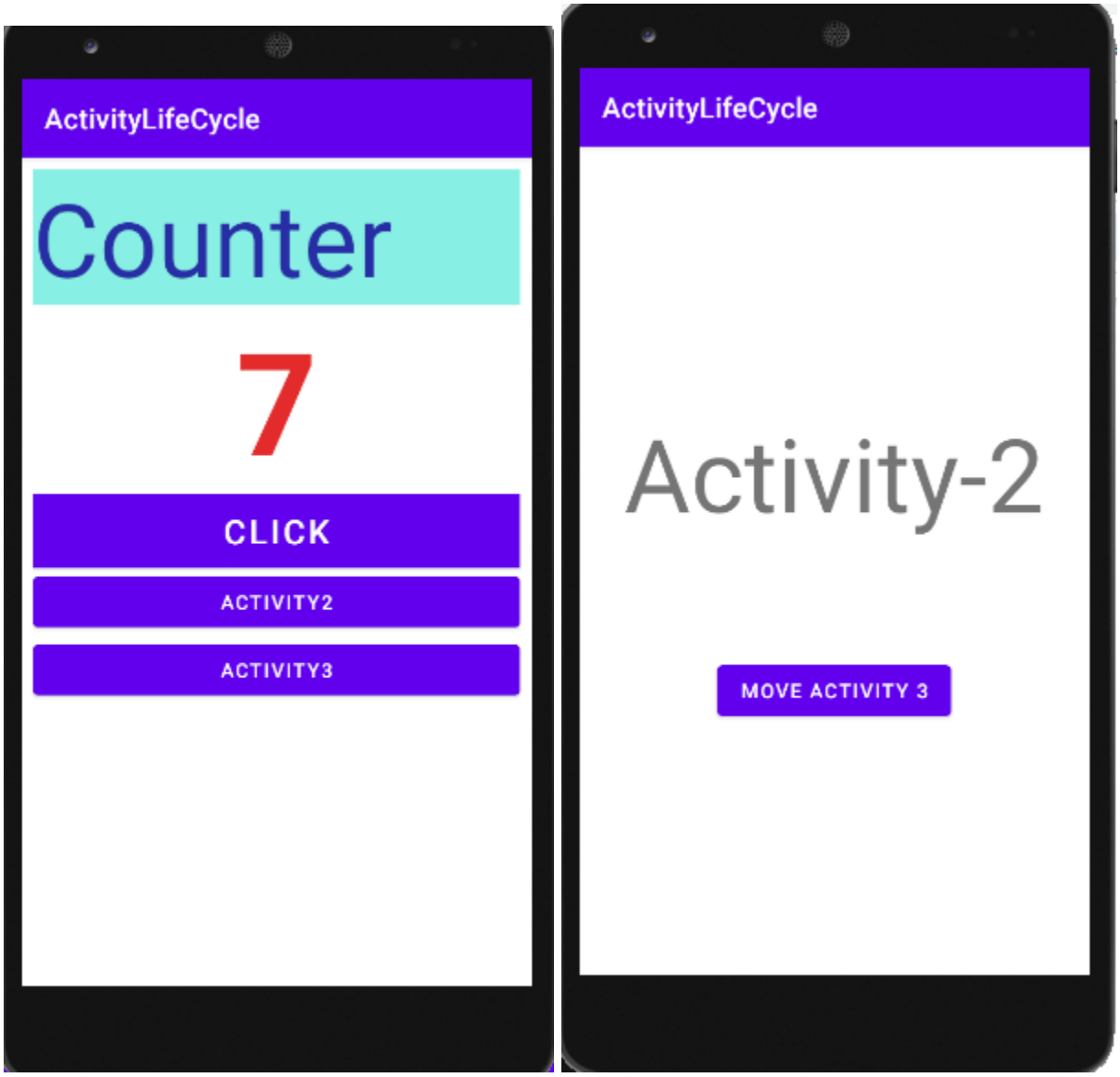
```
if (savedInstanceState!=null)
{
    count=savedInstanceState.getInt( key: "value");
    //txtView.setText(String.valueOf(count) );
}
```

Run the emulator and see add come counts,then rotate and see behavoiur.

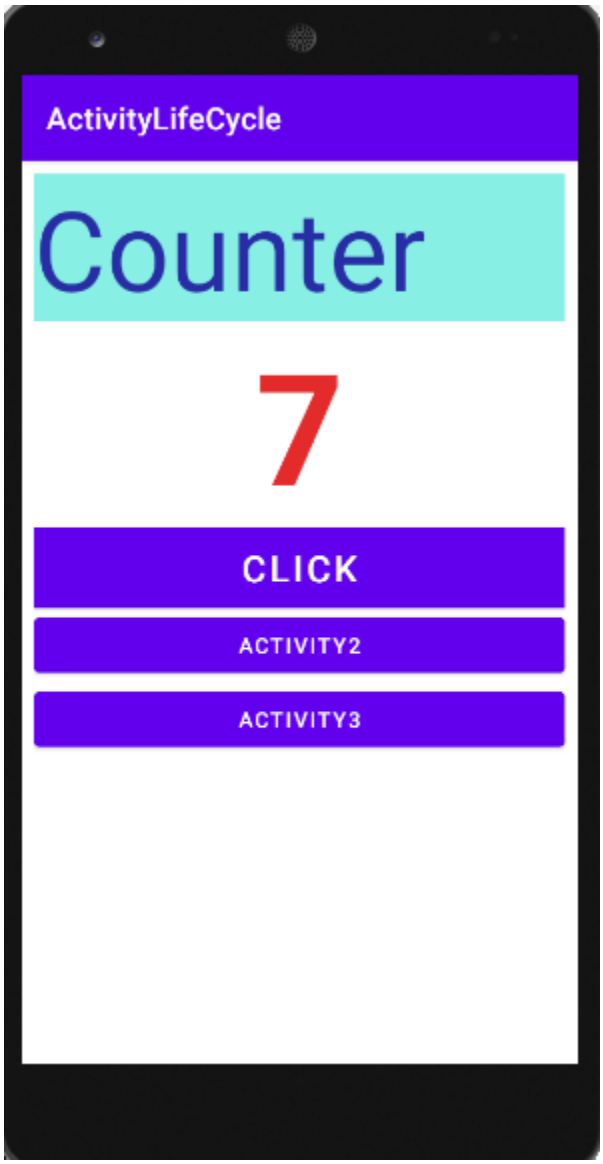




Now get back to original rotation format and then move to other activity.



Move back from second activity and see the behaviour. The count value will be the same as you saved.



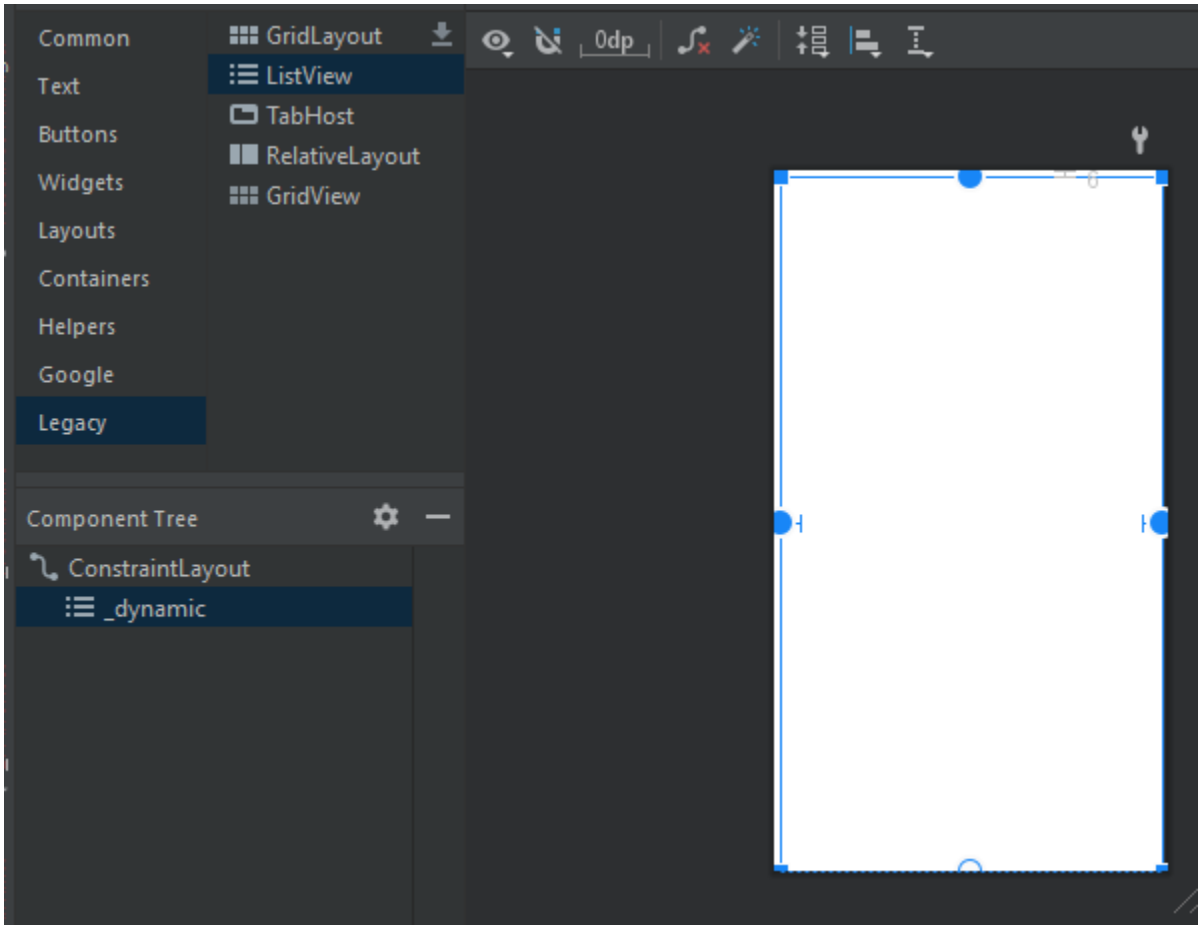
LECTURE#8

❖ Android Adapter:

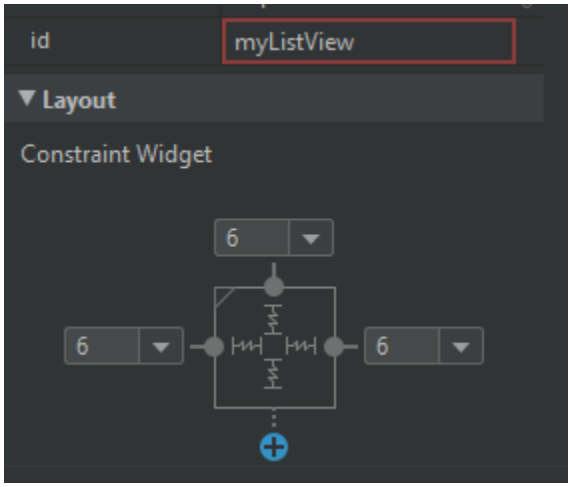
In Android, Adapter is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to an Adapter view then view can takes the data from the adapter view and shows the data on different views like as ListView, GridView, Spinner etc.

❖ List View:

- Add a list view from pallet on xml page.

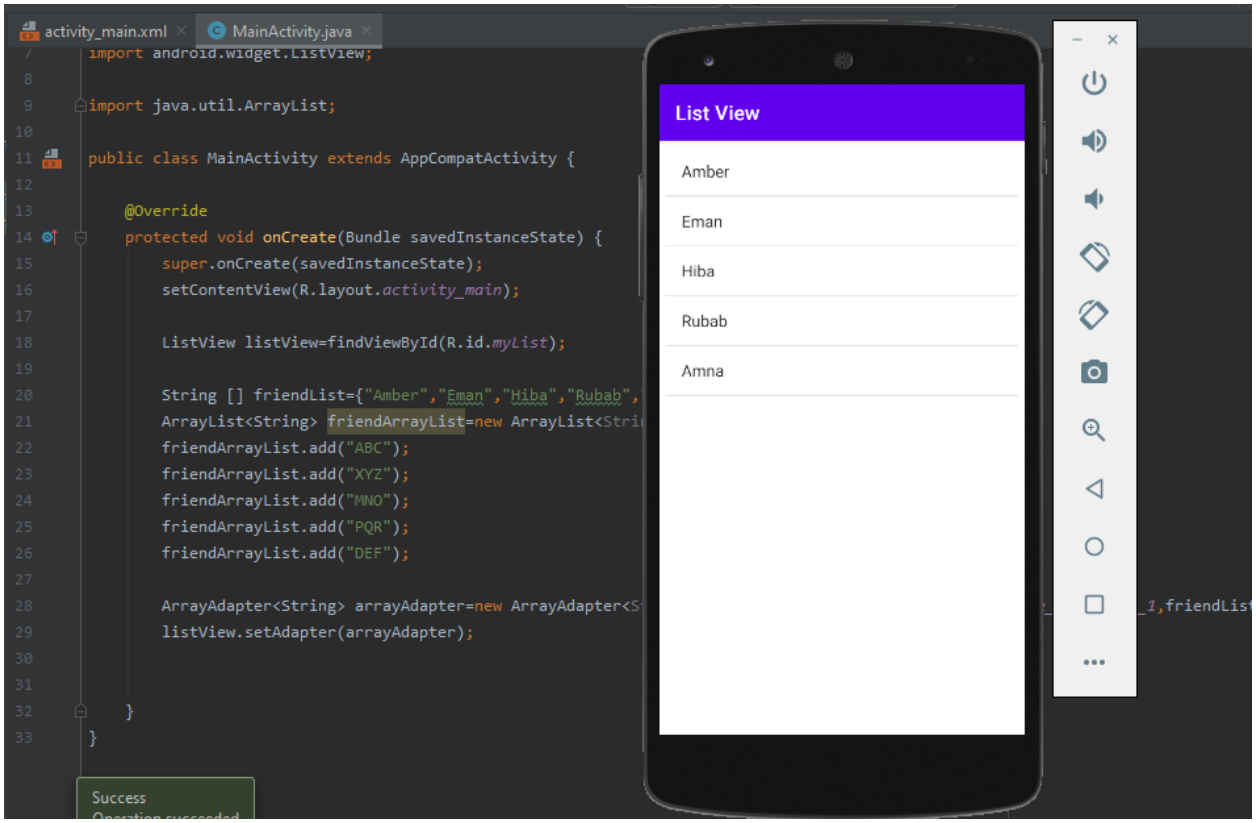


- Set Constraints.



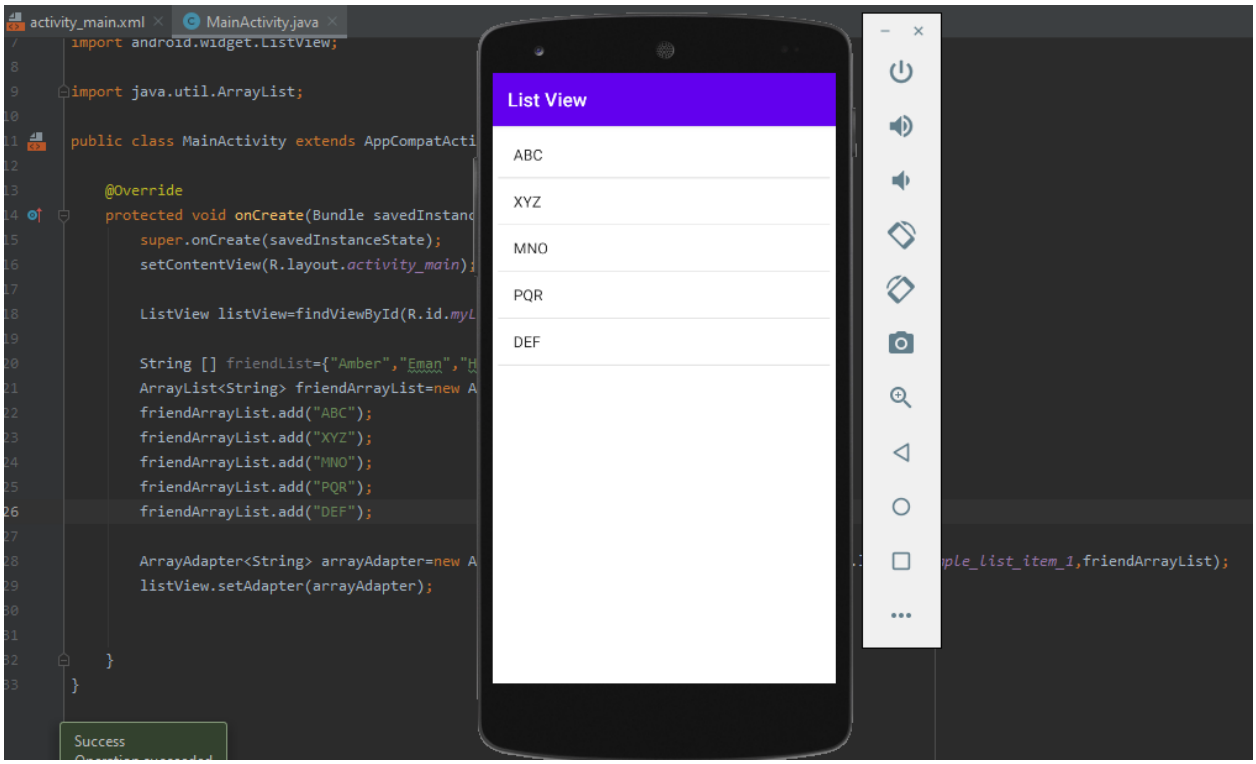
❖ Array :

- Make an array of strings, and display it using **array adapter.**



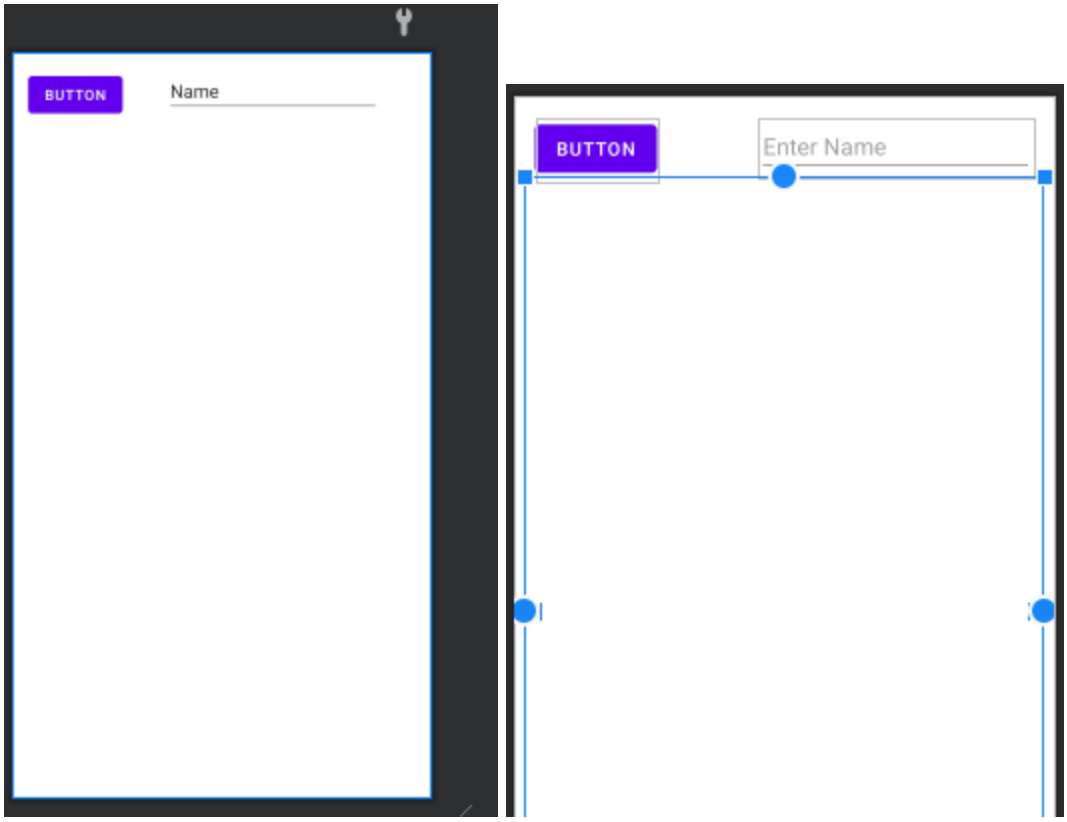
❖ Array List:

Make an ArrayList of strings, and display it using **array adapter.**

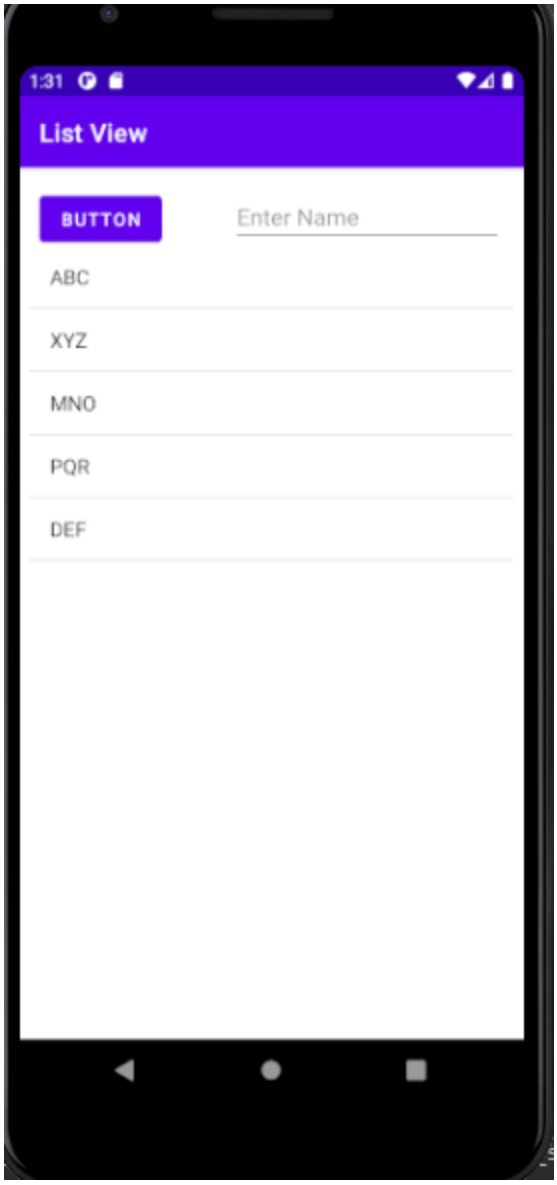


❖ Add new Record

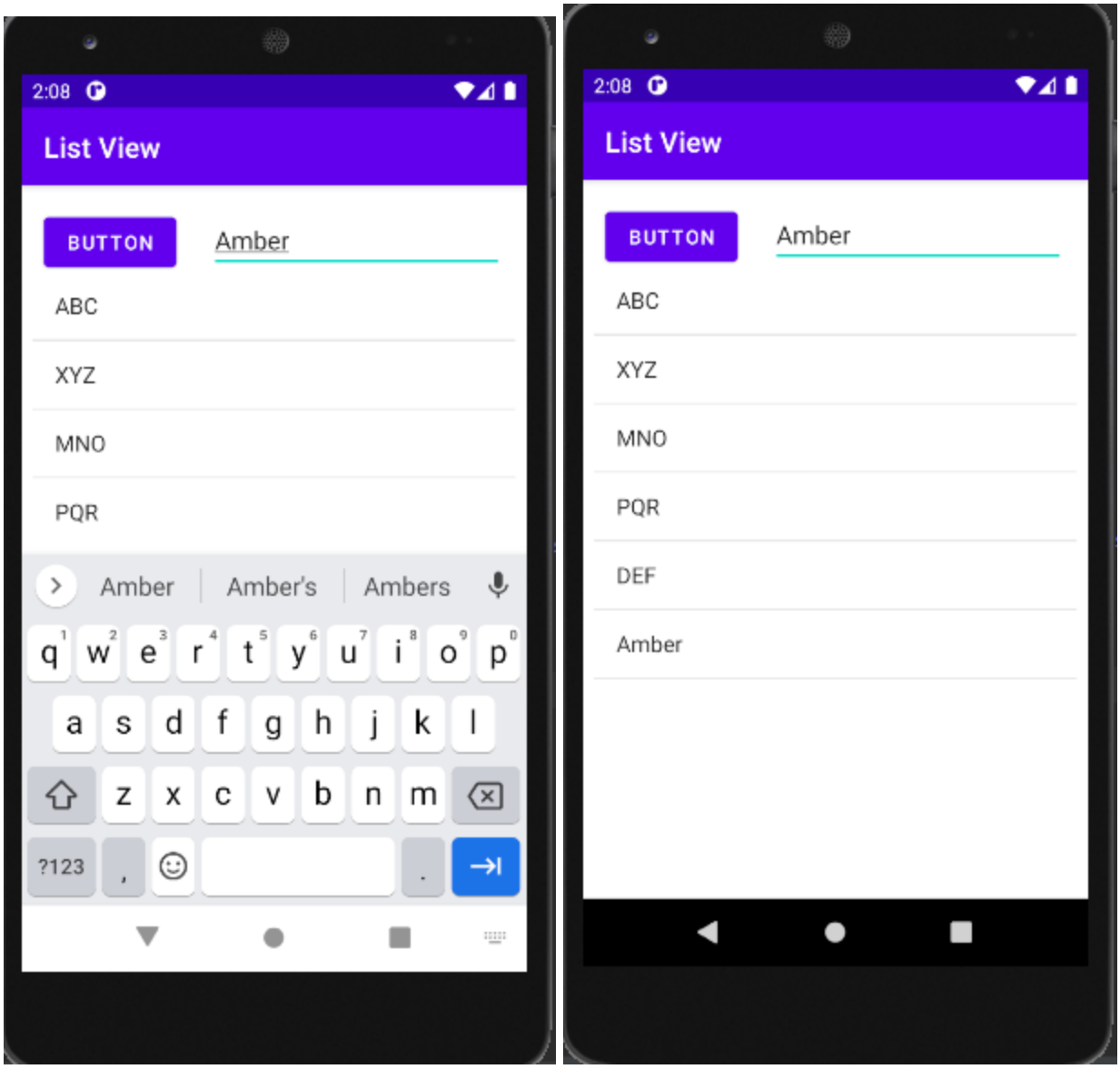
- In the list view, Add a button and plain text.
- Then set their constraints in such a way that list is displayed under them.



- The layout will be like the image below.

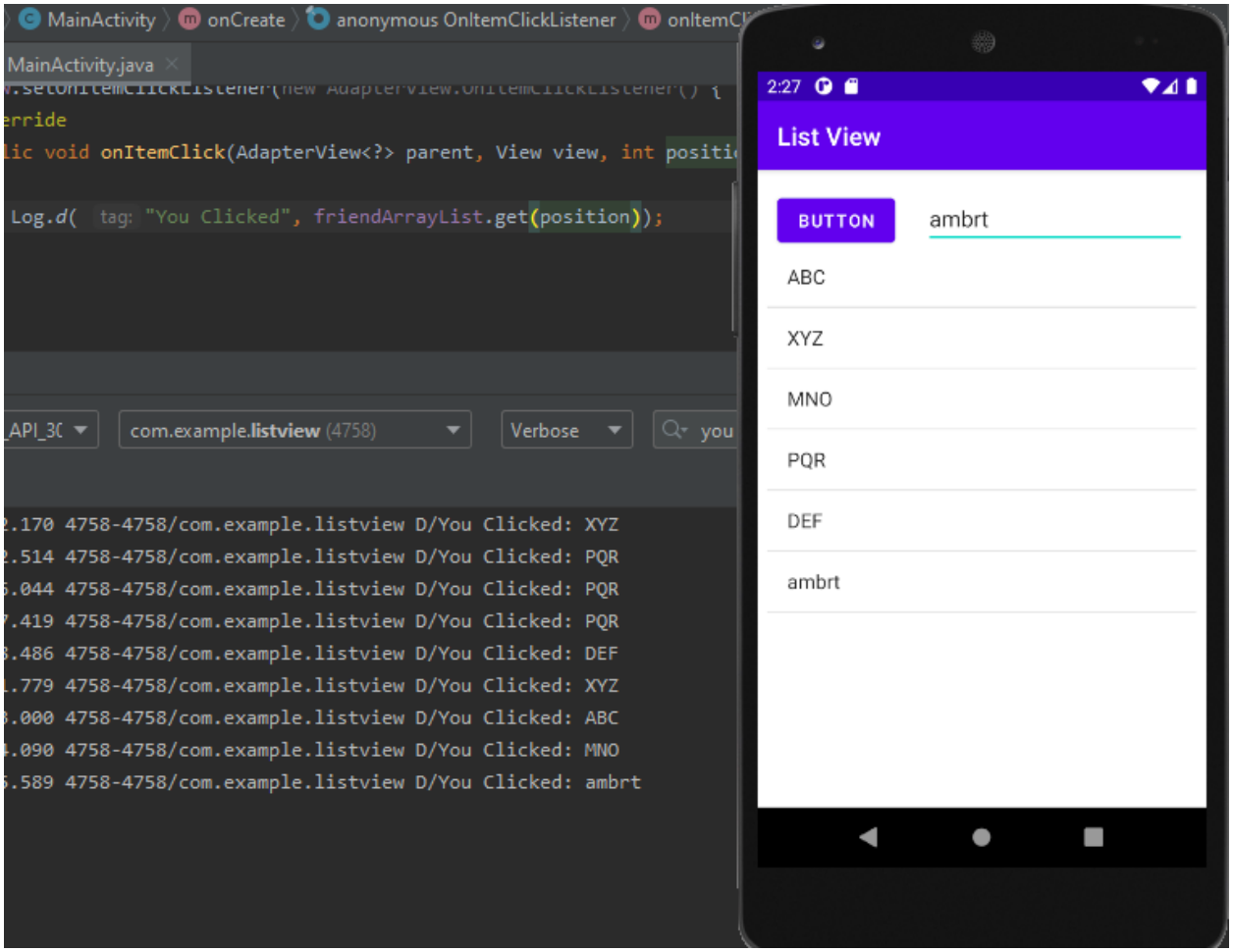


- On click event of listener add text to list using function.



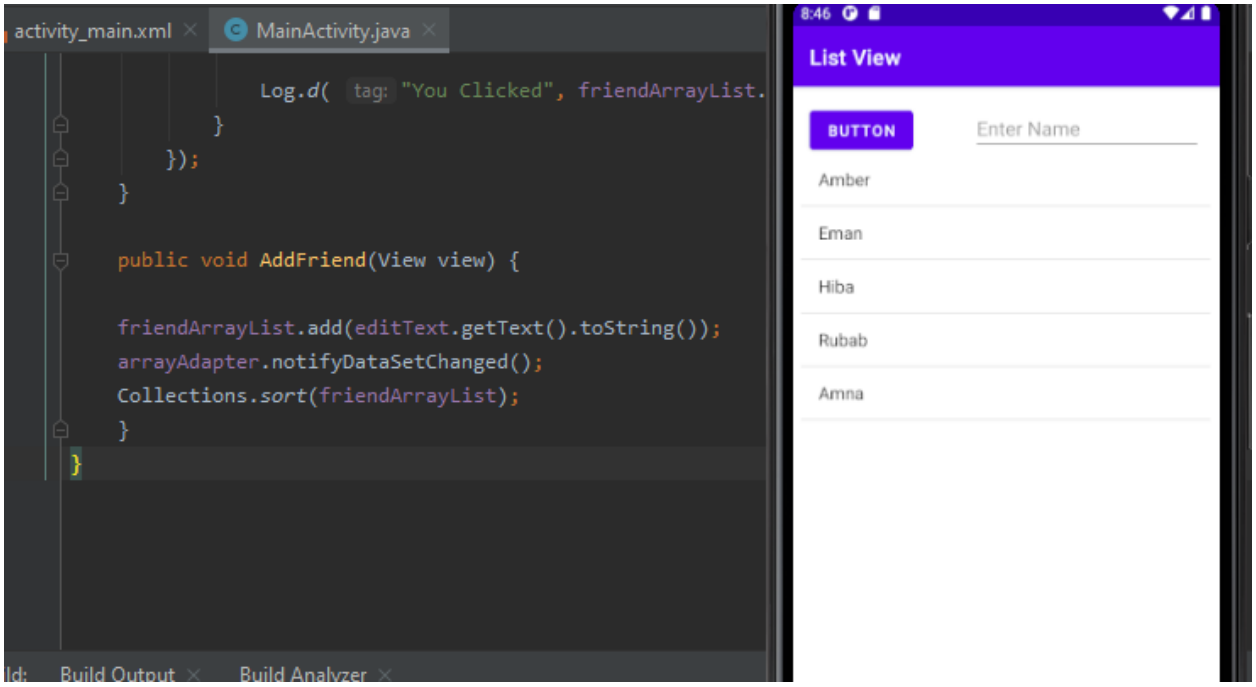
❖ Log Checking List Items :

- Check the clicked item in log using the log function in activity.java file.

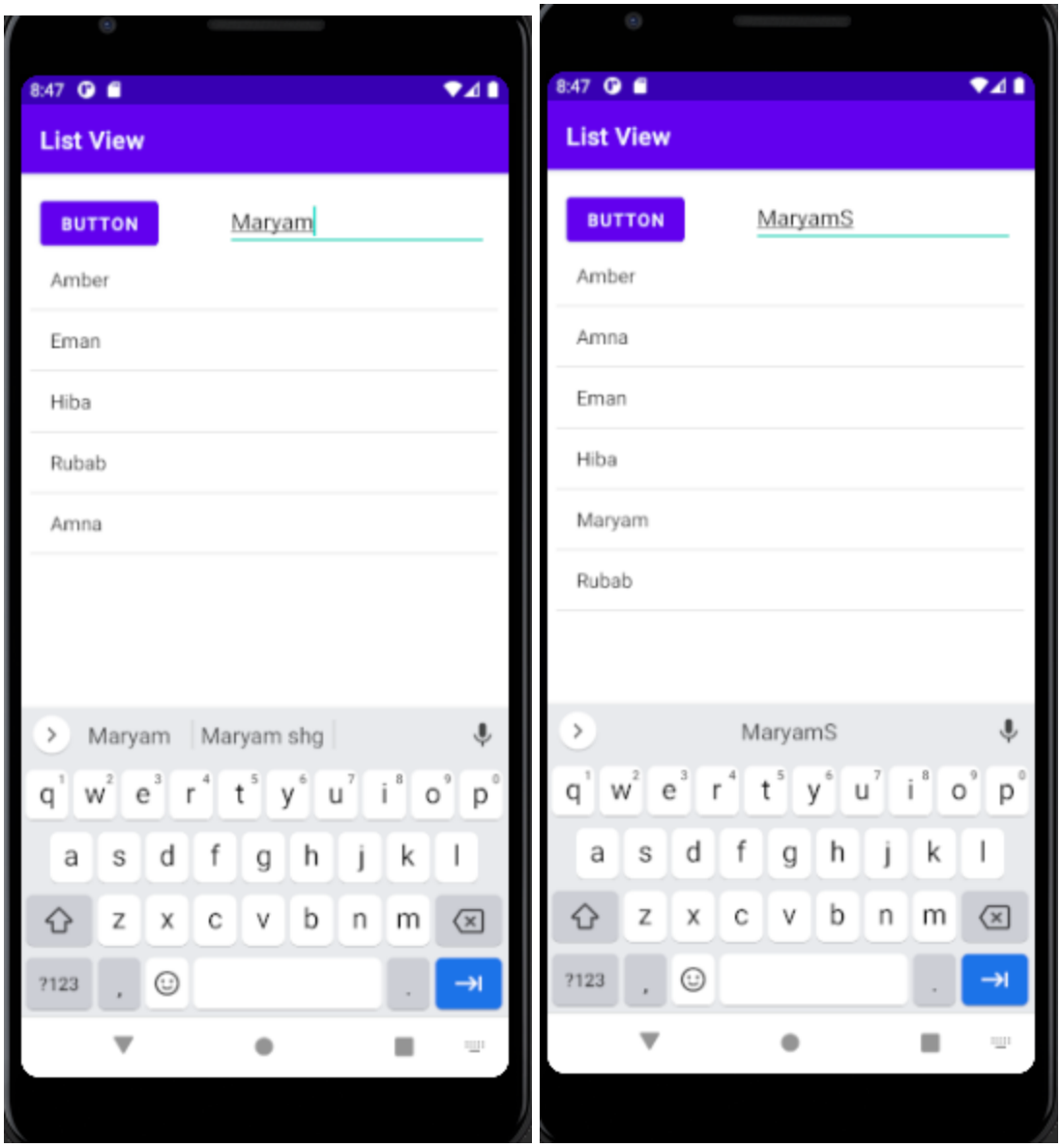


❖ Sorting:

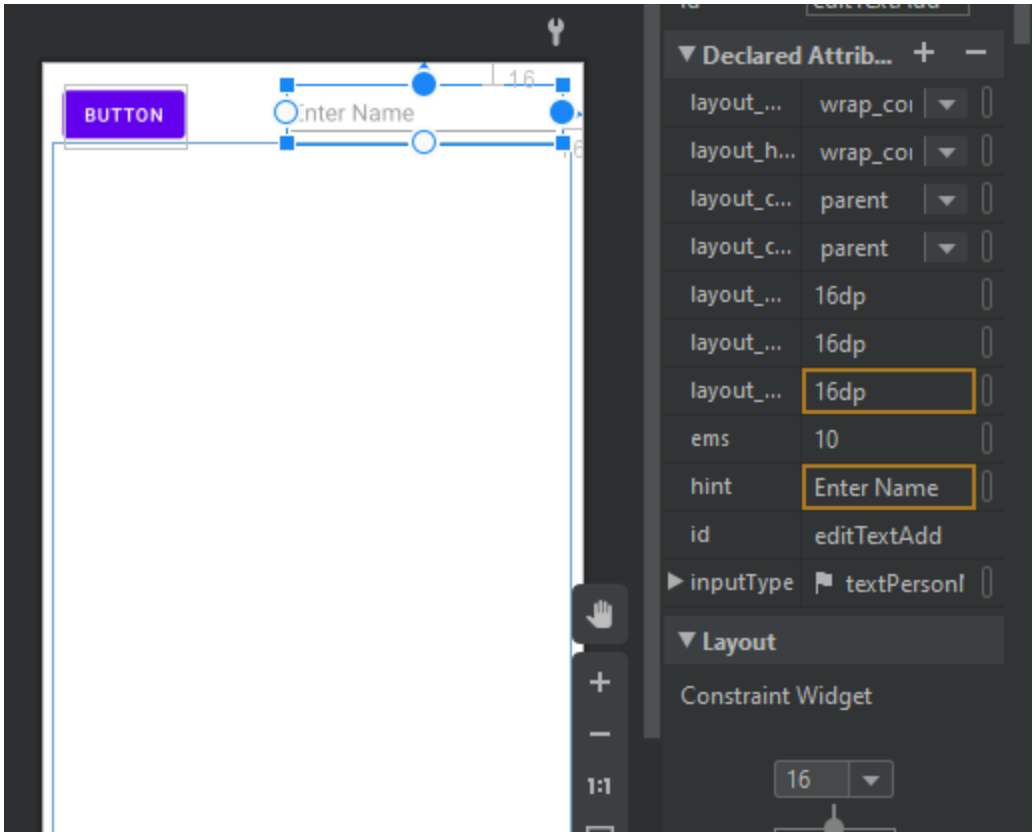
- Array list items can be sorted using the functions as shown below.



- After using the sort function add new items to list and check the their sorting accordingly.

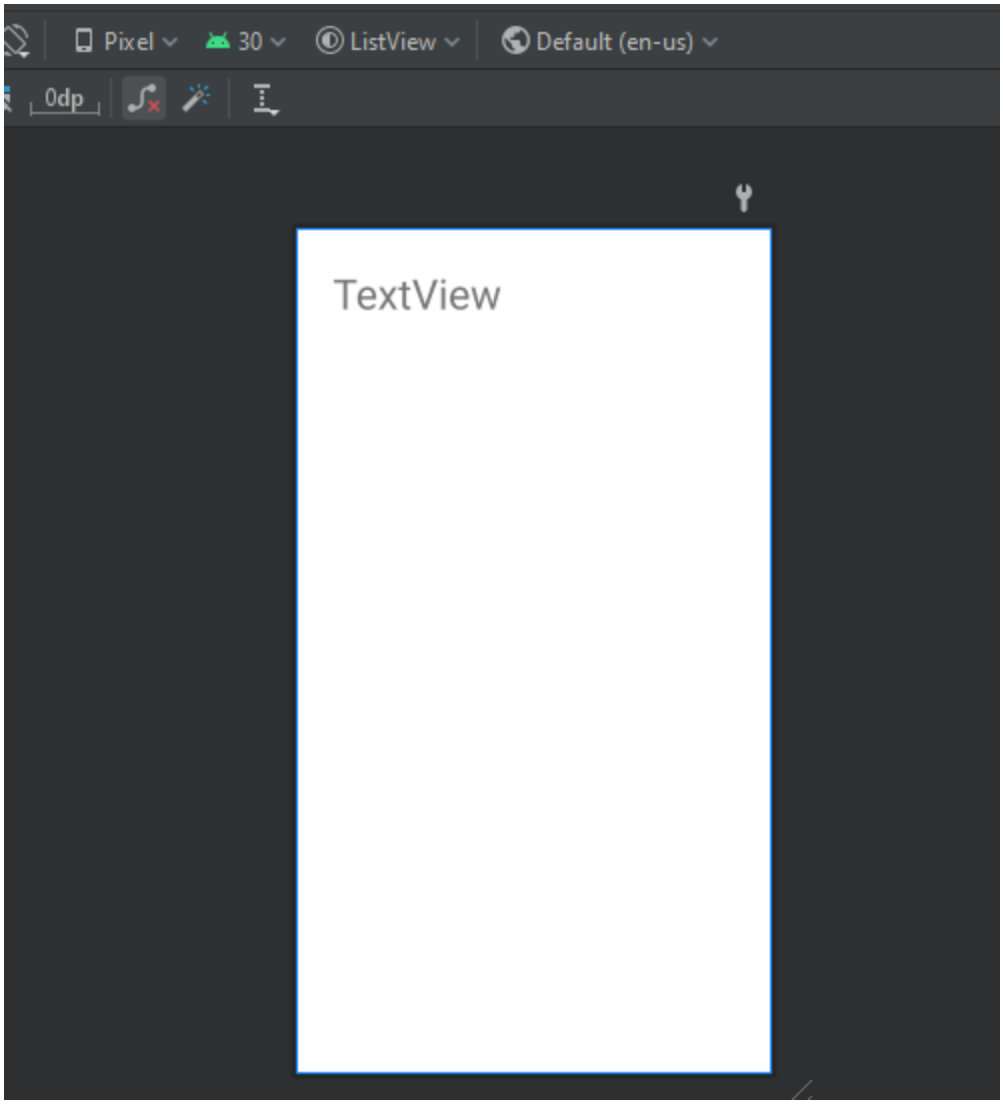


- **Add Hint in Text View:**
    - Instead of using some text add hint to text view to facilitate user to add thierown text while clicking on the view.



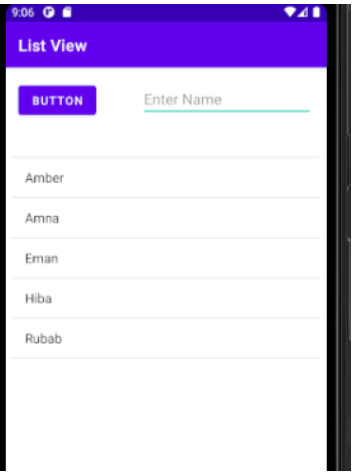
❖ Pass Data from One Activity to Another:

- To pass the data from one activity to another, add new activity and then add a text view in it.

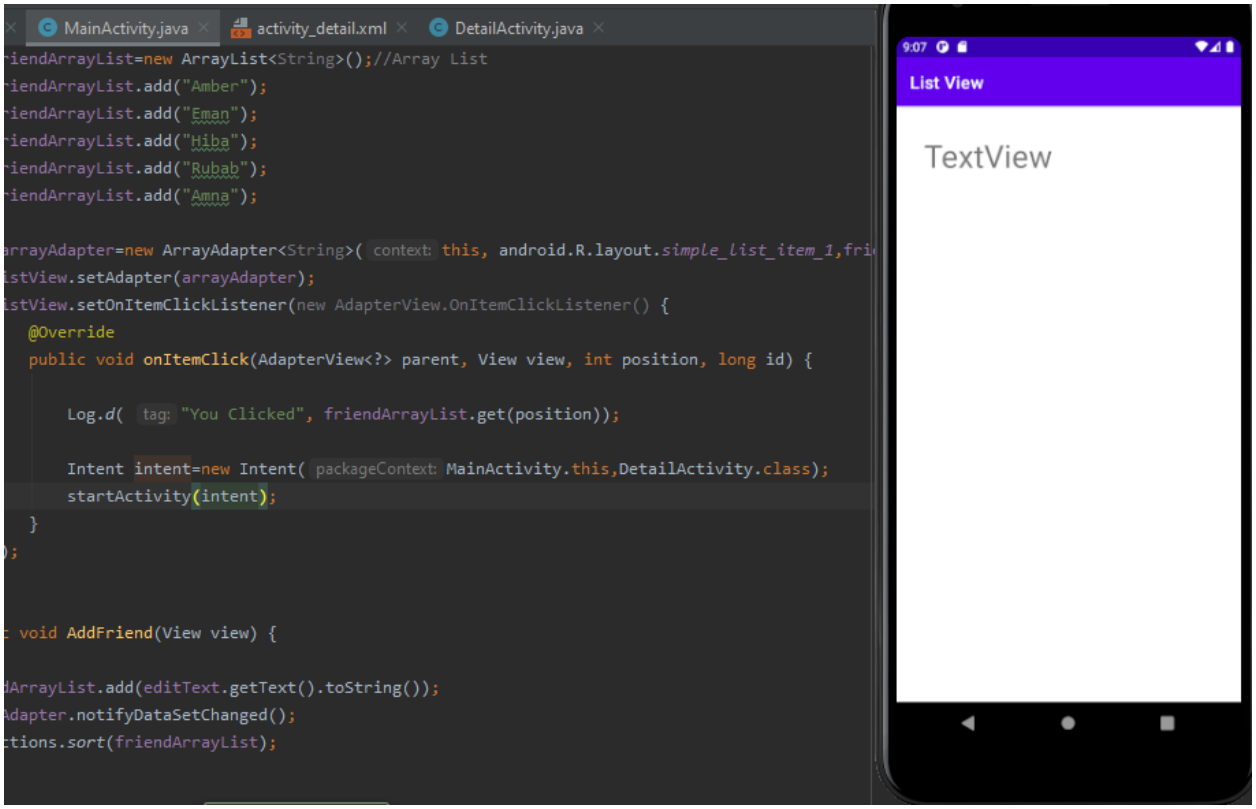


- Then make an intent as below to move from 1st activity to second.

```
friendArrayList=new ArrayList<String>();//Array List
friendArrayList.add("Amber");
friendArrayList.add("Eman");
friendArrayList.add("Hiba");
friendArrayList.add("Rubab");
friendArrayList.add("Amna");

 arrayAdapter=new ArrayAdapter<String>( context: this, android.R.layout.simple_list_item_1,fri
listView.setAdapter(arrayAdapter);
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

        Log.d( tag: "You Clicked", friendArrayList.get(position));

        Intent intent=new Intent( packageContext: MainActivity.this,DetailActivity.class);
        startActivity(intent);
    }
```

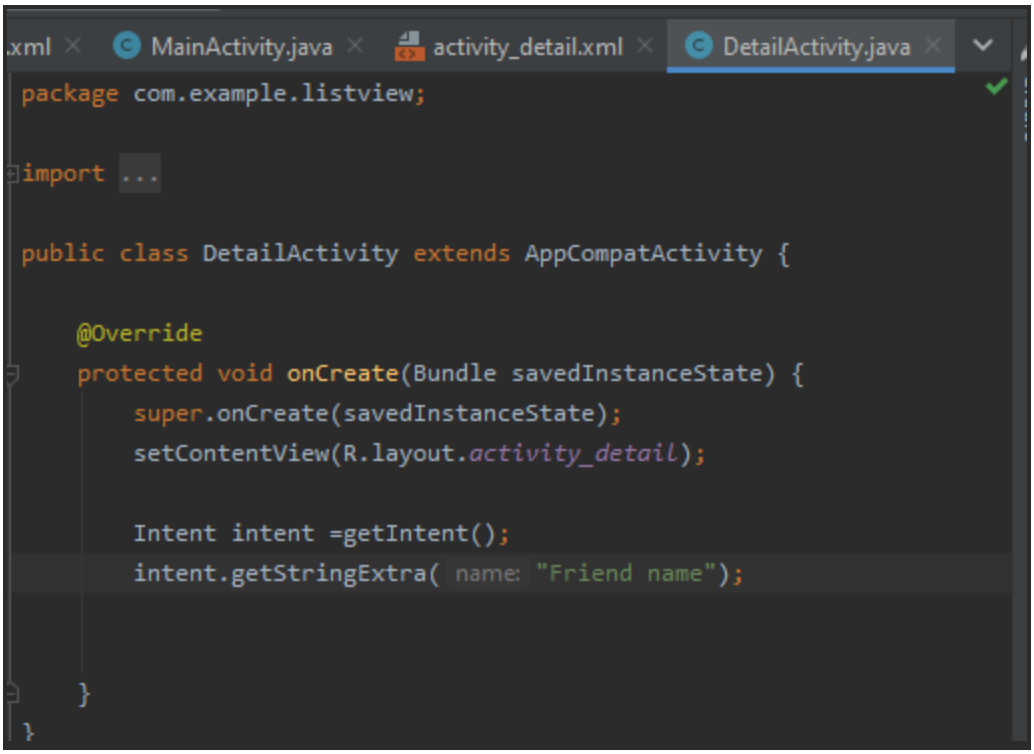


- Check the activity navigation on emulator.

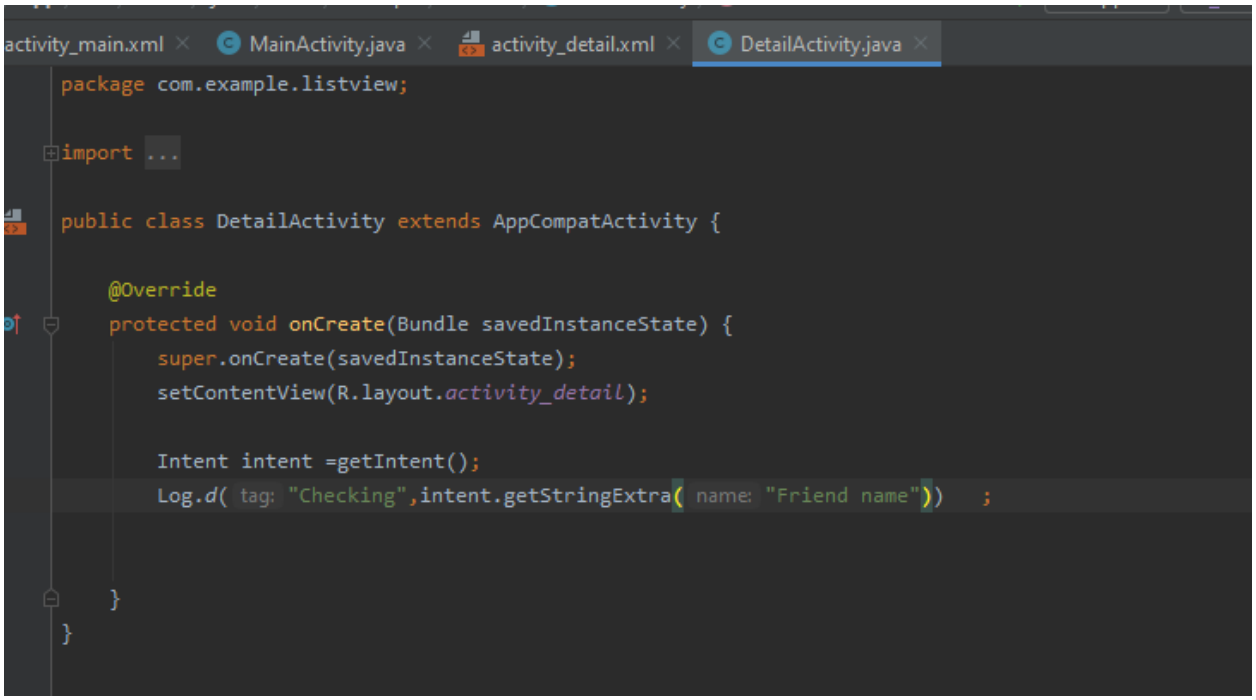


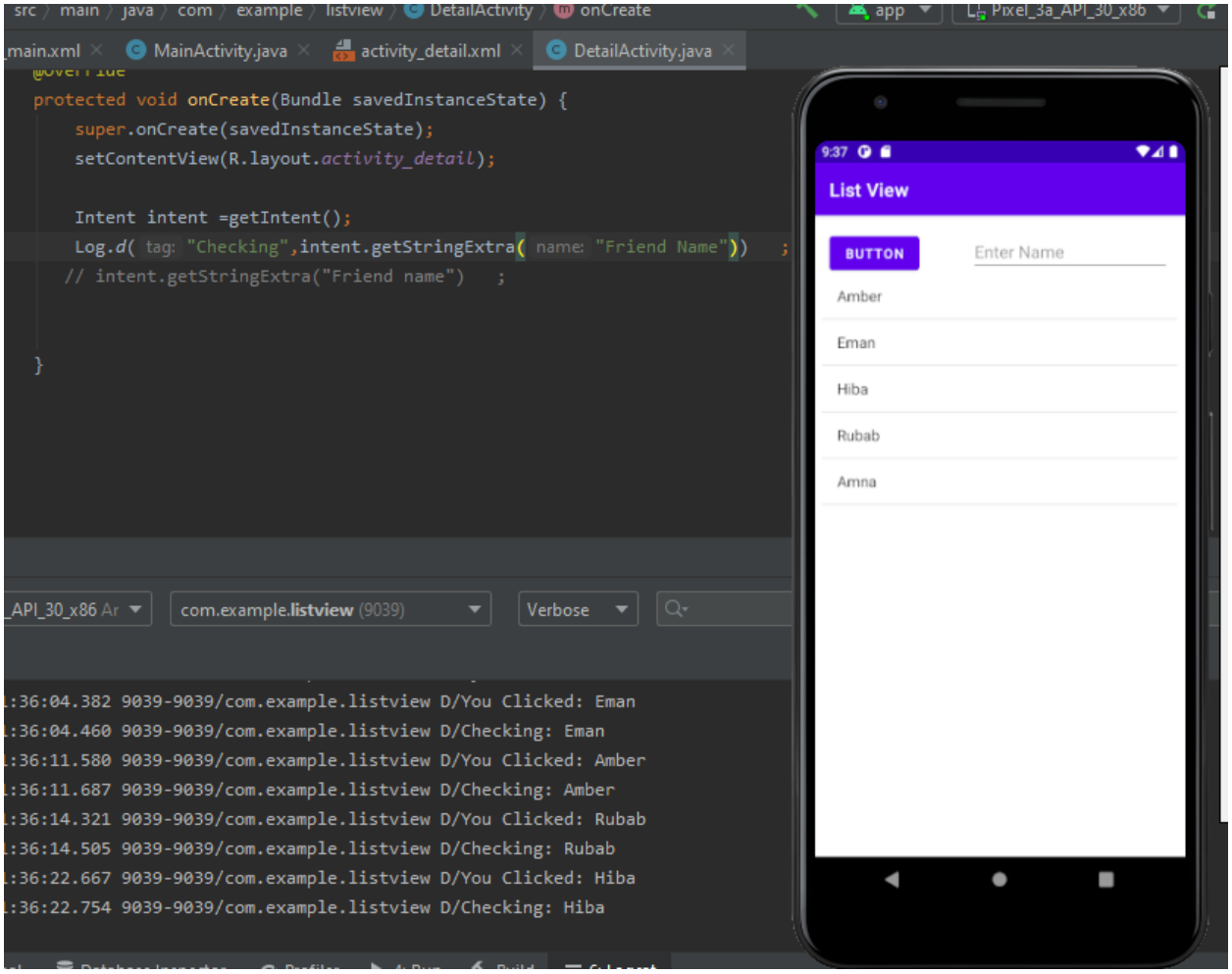
- Now set the clicked item of list of first activity as text of second activity.

```
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

    Log.d( tag: "You Clicked", friendArrayList.get(position));

    Intent intent=new Intent( packageContext: MainActivity.this,DetailActivity.class);
    intent.putExtra( name: "Friend Name",friendArrayList.get(position));
    startActivity(intent);
}
```

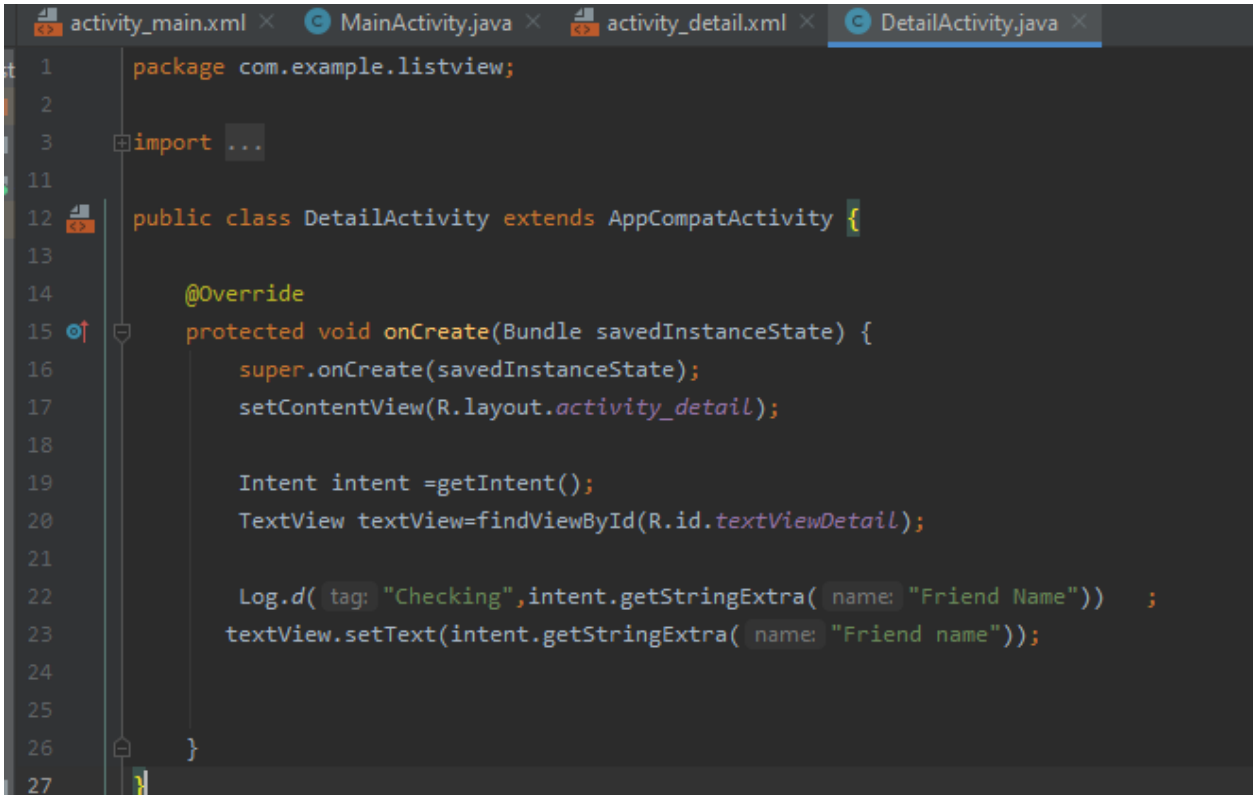


- Place a log function to see the clicked items in one activity and checked in second activity.



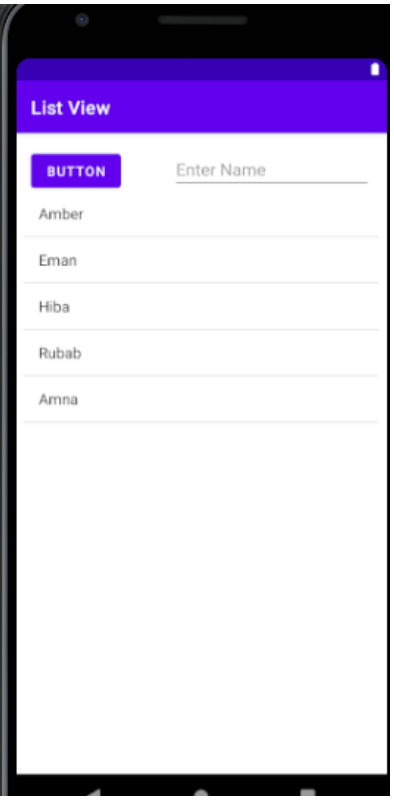

src main java com example listview DetailActivity onCreate
app
Pixel_3a_API_30_x86
main.xml
MainActivity.java
activity_detail.xml
DetailActivity.java
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_detail);
Intent intent =getIntent();
Log.d(tag: "Checking",intent.getStringExtra(name: "Friend Name"))
// intent.getStringExtra("Friend name")
}
com.example.listview (9039)
Verbose
:36:04.382 9039-9039/com.example.listview D/You Clicked: Eman
:36:04.460 9039-9039/com.example.listview D/Checking: Eman
:36:11.580 9039-9039/com.example.listview D/You Clicked: Amber
:36:11.687 9039-9039/com.example.listview D/Checking: Amber
:36:14.321 9039-9039/com.example.listview D/You Clicked: Rubab
:36:14.505 9039-9039/com.example.listview D/Checking: Rubab
:36:22.667 9039-9039/com.example.listview D/You Clicked: Hiba
:36:22.754 9039-9039/com.example.listview D/Checking: Hiba
9:37
List View
BUTTON
Enter Name
Amber
Eman
Hiba
Rubab
Amna

- Get the clicked item in second activity to display when clicked.



- Now see the output on emulator, as we click any item in activity one it will be displayed in second activity.

```java
package com.example.listview;

import ...

public class DetailActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        TextView txtView=findViewById(R.id.textViewDetail);
        Intent intent =getIntent();

        Log.d( tag: "Checking",intent.getStringExtra( name: "Friend Name"))    ;
        txtView.setText(intent.getStringExtra( name: "Friend Name"));


    }
}
```



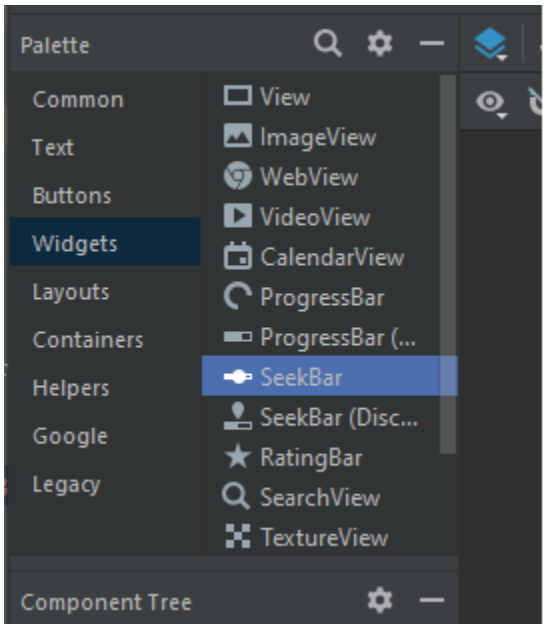
activity_main.xml | activity_detail.xml | DetailActivity.java

```java
package com.example.listview;

import ...

public class DetailActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_detail);

        TextView txtView=findViewById(R.id.textViewDetail);
        Intent intent =getIntent();

        Log.d( tag: "Checking",intent.getStringExtra( name: "Friend Name"
        txtView.setText(intent.getStringExtra( name: "Friend Name"));


    }
}
```

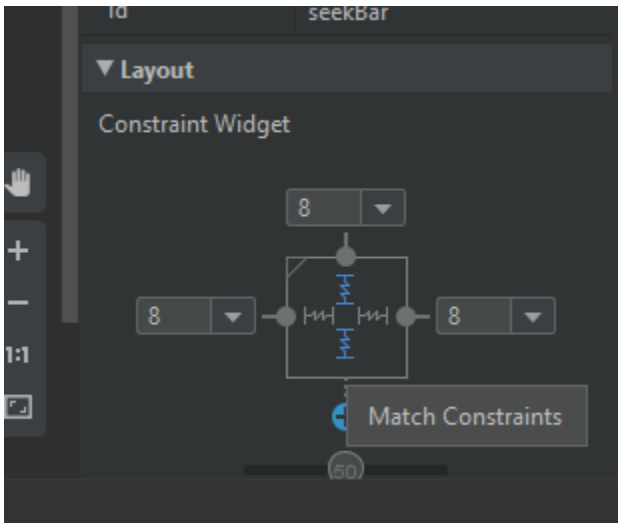
LECTURE#9

❖ Seek bar:

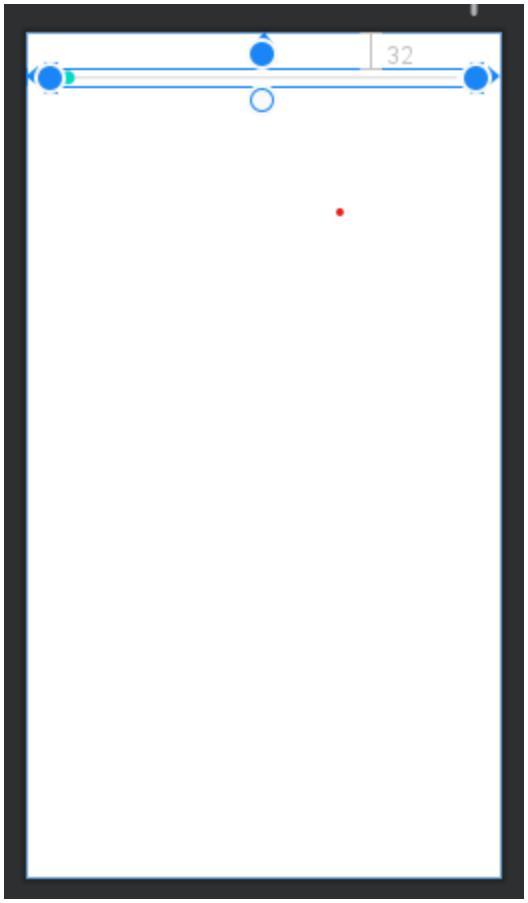
- Add seek bar from widgets on xml file.



- Set constraints of seek bar.

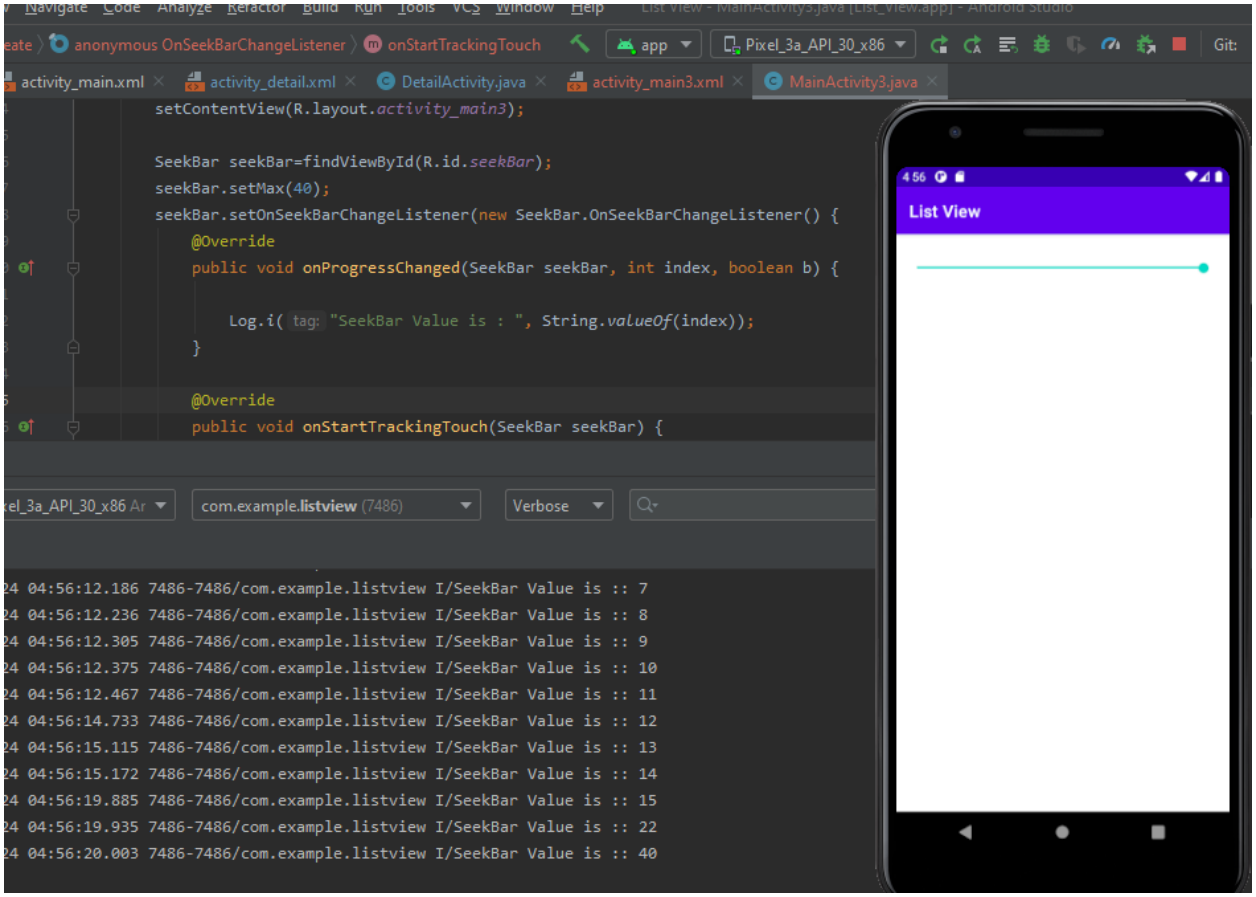


- After setting the position of seek bar , run it ti see its view on emulator.



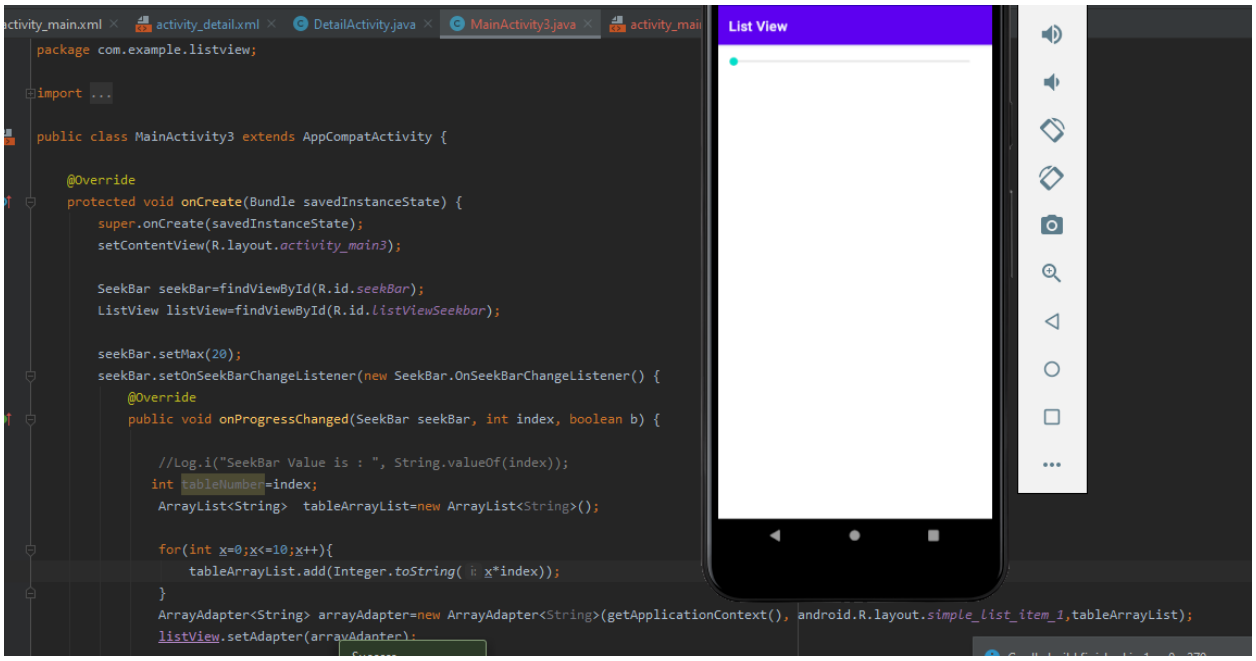
❖ Setting Max value and read it in log:

- Set maximum value of seek bar using function as below and check its output values by using log.
- As we run the emulator and move the seek bar the values will be displayed in log accordingly.

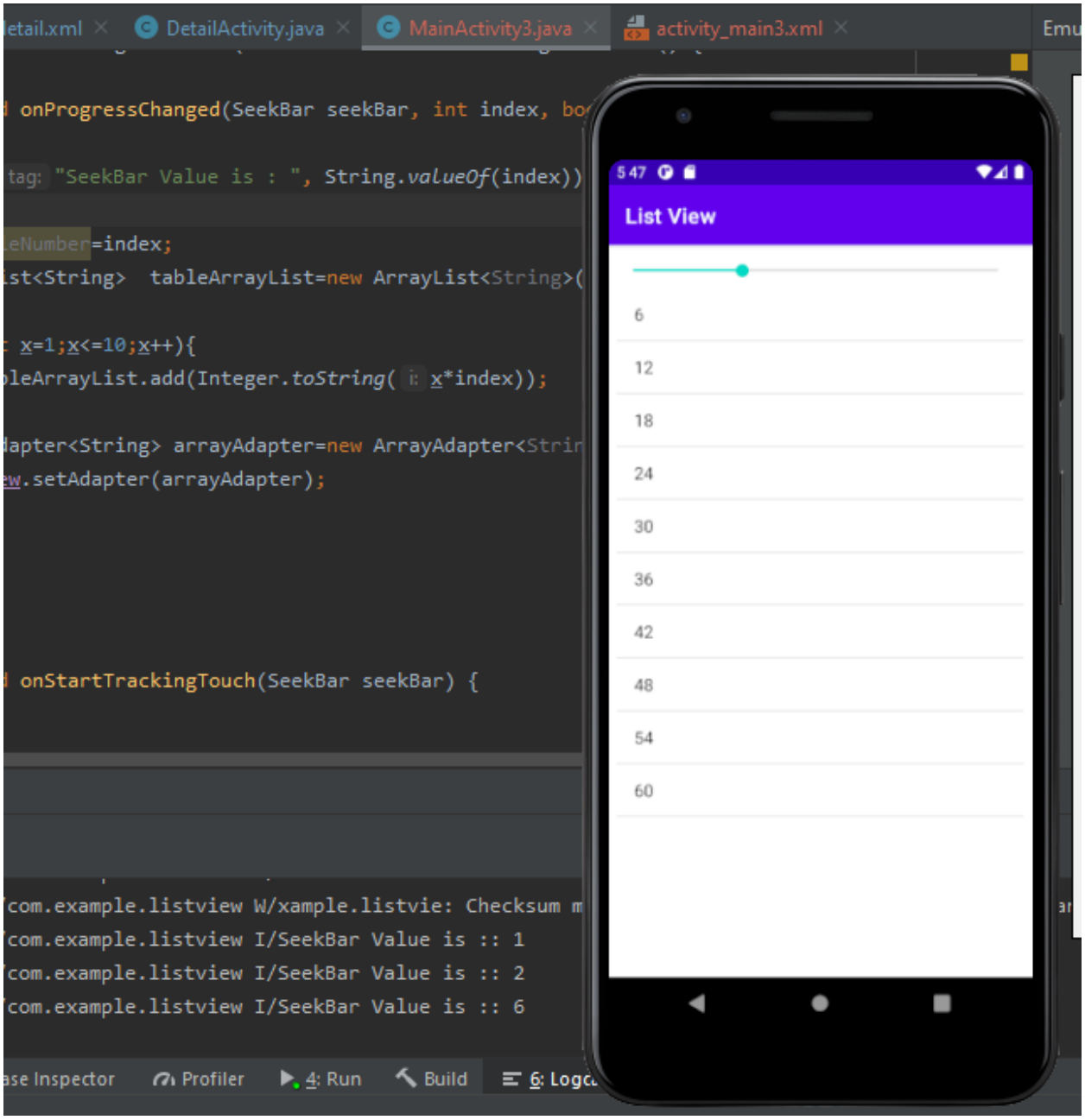


❖ To print table of number on seekbar:

- Use seek bar function to get its value and store it in table.

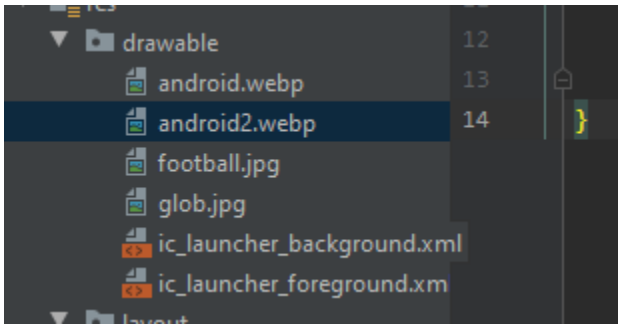


- Add a list view under seek bar in xml file .
- Write a function to display the table of value selected on seek bar.
- Run the emulator to see the output on screen.

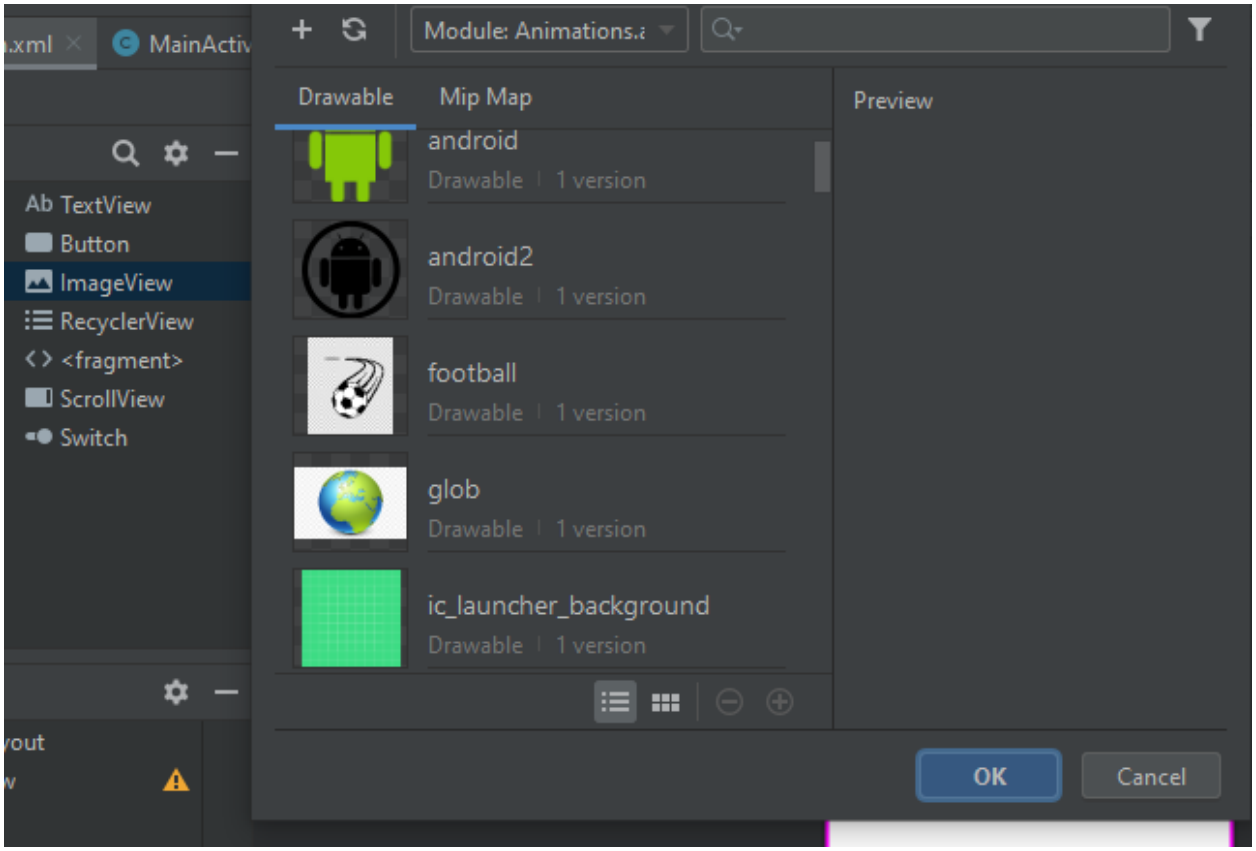

etail.xml
DetailActivity.java
MainActivity3.java
activity_main3.xml
onProgressChanged(SeekBar seekBar, int index, bo
tag: "SeekBar Value is : ", String.valueOf(index))
eNumber=index;
ist<String> tableArrayList=new ArrayList<String>(
x=1;x<=10;x++){
leArrayList.add(Integer.toString(i: x*index));
dapter<String> arrayAdapter=new ArrayAdapter<Strin
w.setAdapter(arrayAdapter);
onStartTrackingTouch(SeekBar seekBar) {
com.example.listview W/xample.listvie: Checksum m
com.example.listview I/SeekBar Value is :: 1
com.example.listview I/SeekBar Value is :: 2
com.example.listview I/SeekBar Value is :: 6
ase Inspector
Profiler
4: Run
Build
6: Logc
Emu
5 47
List View
6
12
18
24
30
36
42
48
54
60

❖ Widgets and Animations:

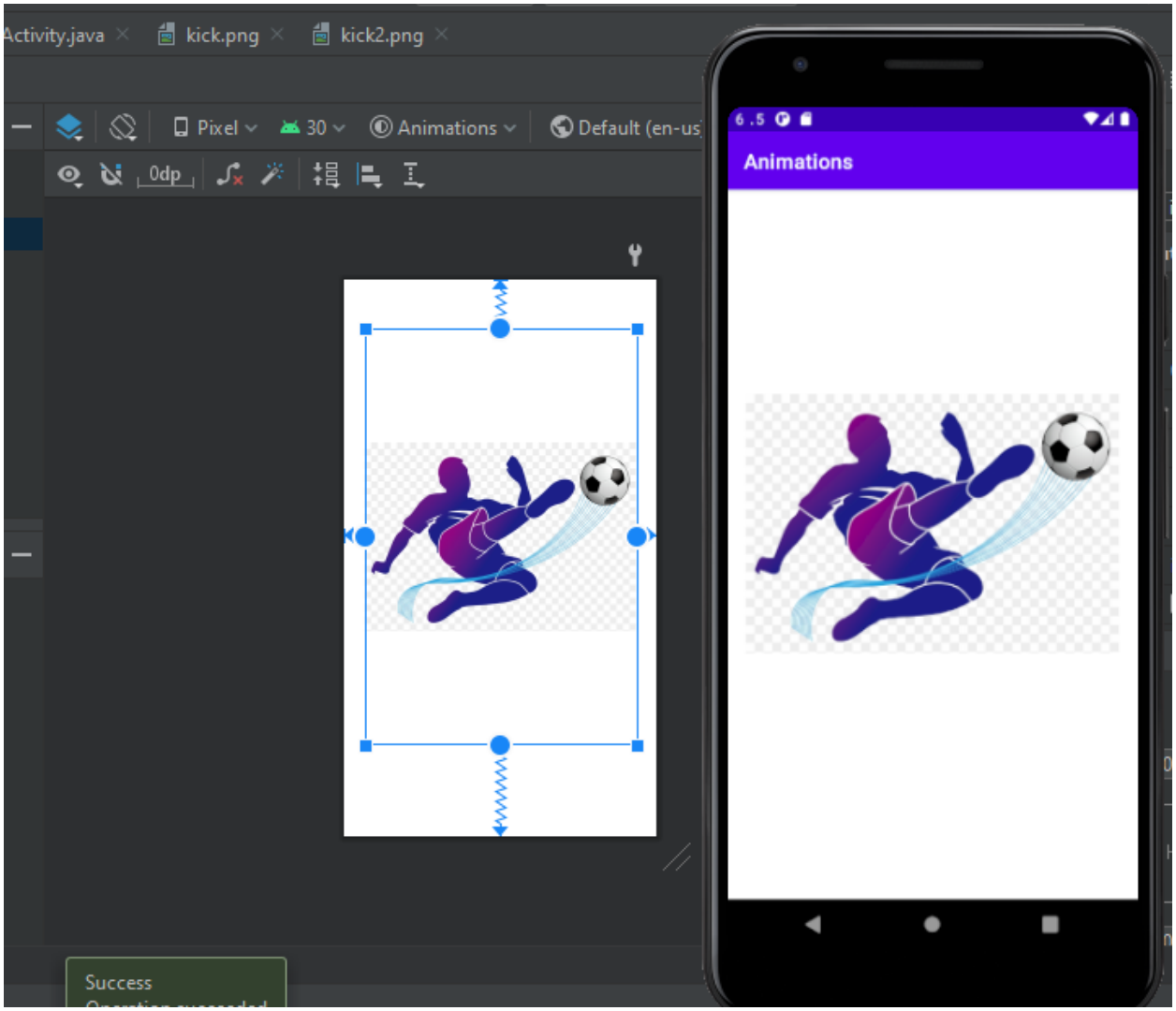
- Copy come images from your PC and paste then in **drawable** folder in your project.



- Add an image view on xml file and select any image you want to add.



- Set image constraints and position.
- Run the emulator to see the output of image.



❖ On click for image:

- Write an onclick function for image to perform actions on image click.

```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="357dp"
    android:layout_height="546dp"
    android:onClick="Animation"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.351"
```

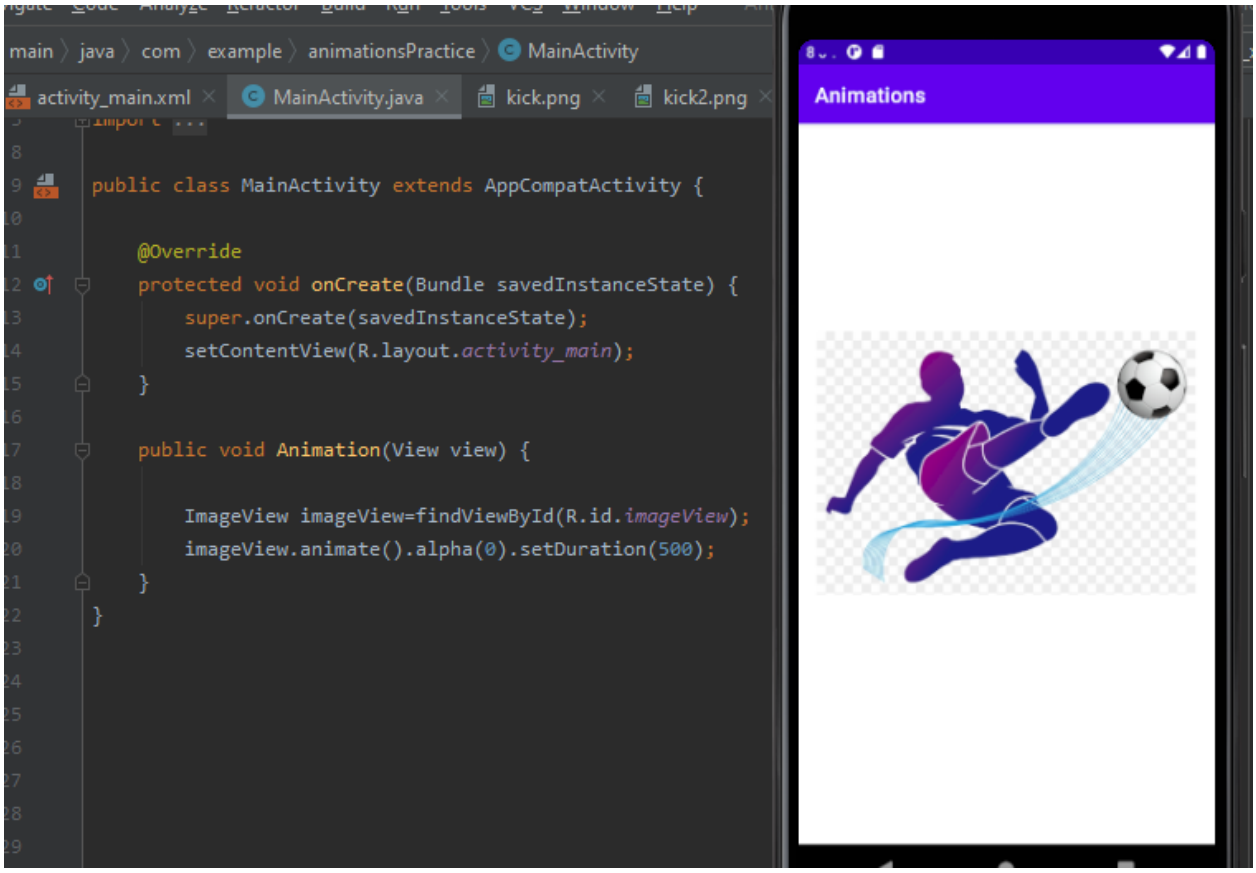
Basic functions of Animation:

❖ Image Disappear Animation :

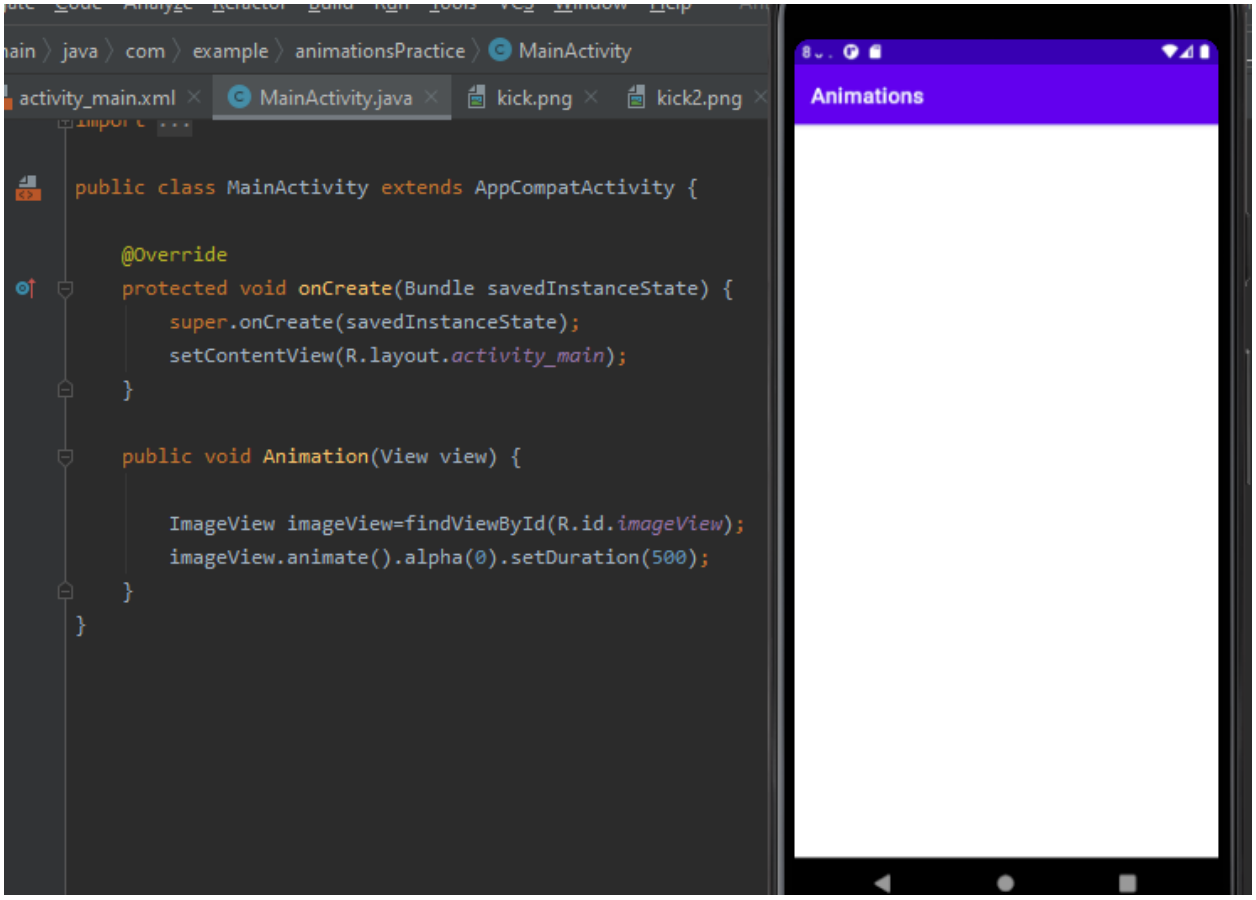
- Write the function as below to get the disappearing effect on image click.

```
tivity_main.xml | MainActivity.java | kick.png | kick2.png

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void Animation(View view) {

        ImageView imageView=findViewById(R.id.imageView);
        imageView.animate().alpha(0).setDuration(500);
    }
```

- Image view before click.



- Image disappeared just after click.

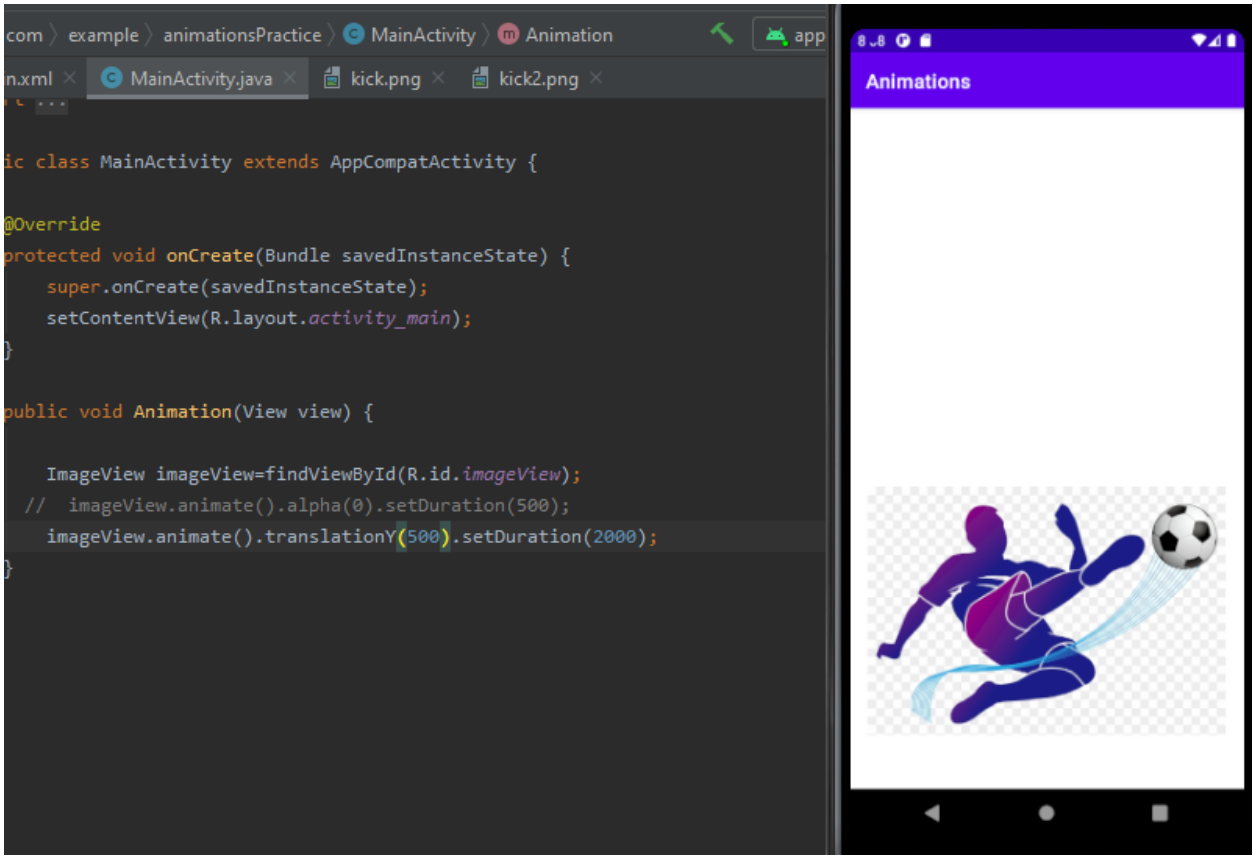


❖ Image TranslationY Animation:

- Write the function as below to get Translation along Y effect on image click.

```
public void Animation(View view) {

    ImageView imageView=findViewById(R.id.imageView);
  //  imageView.animate().alpha(0).setDuration(500);
    imageView.animate().translationY(500).setDuration(2000);
}
```

- Image moved along Y axis just after click.

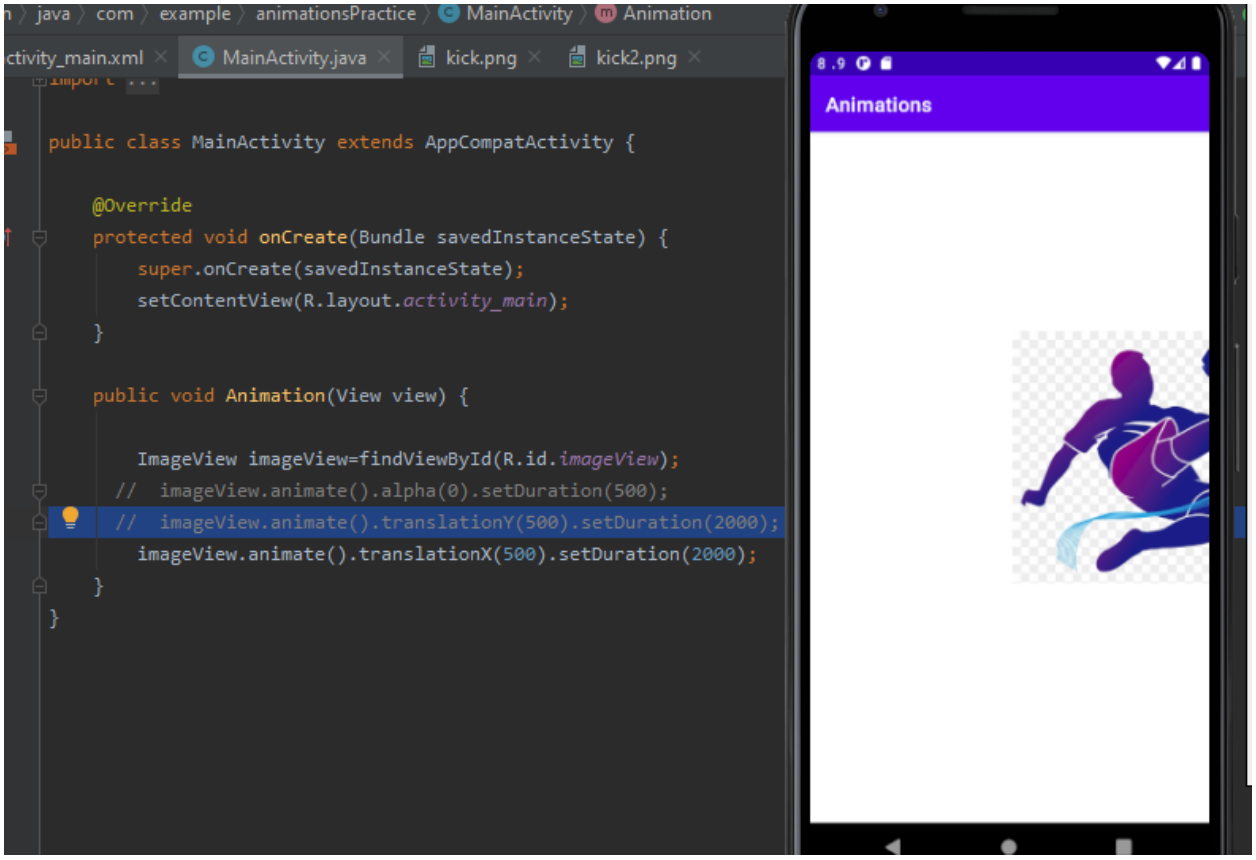


❖ Image TranslationX Animation:

- Write the function as below to get Translation along X effect on image click.

```
public void Animation(View view) {

    ImageView imageView=findViewById(R.id.imageView);
//  imageView.animate().alpha(0).setDuration(500);
//  imageView.animate().translationY(500).setDuration(2000);
    imageView.animate().translationX(500).setDuration(2000);
}
```

- Image moved along X axis just after click.

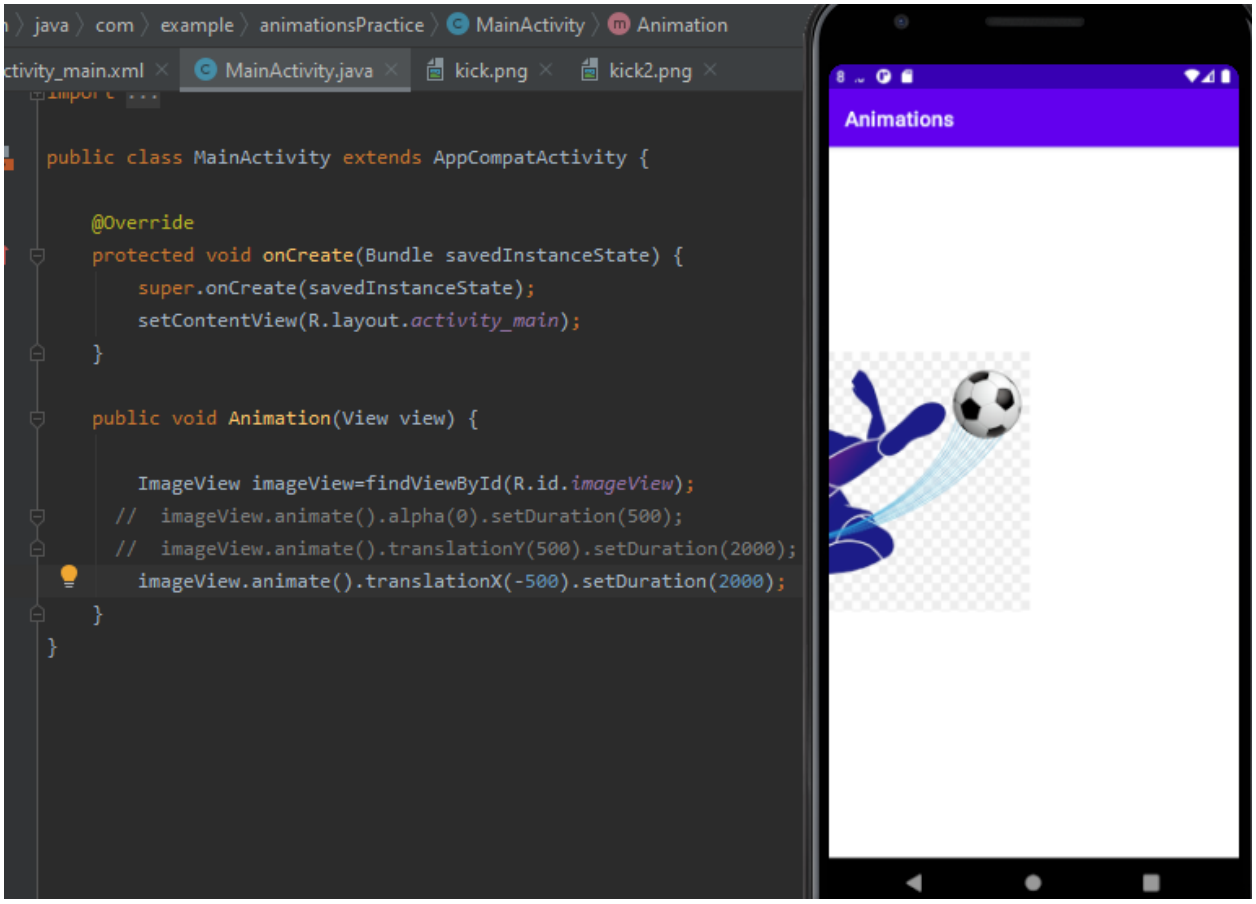


❖ Image -Ve Translation Animation:

Negative value => move to opposite side

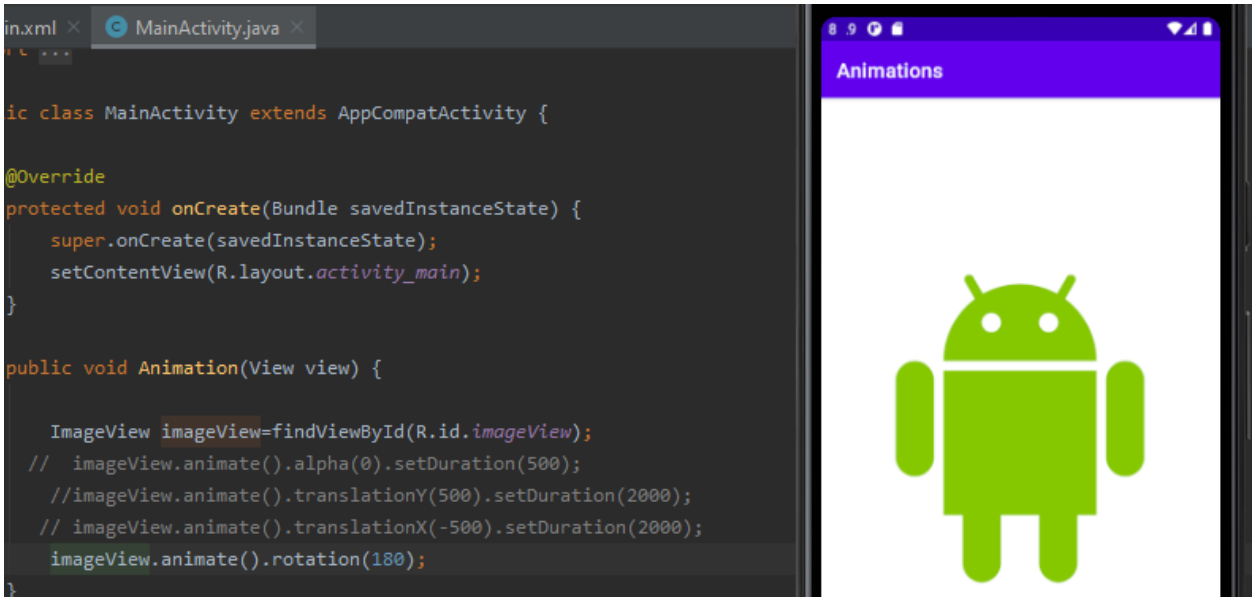
- Write the function as below to get-Ve Translation Animation effect on image click.

- Image moved along opposite to X axis just after click.

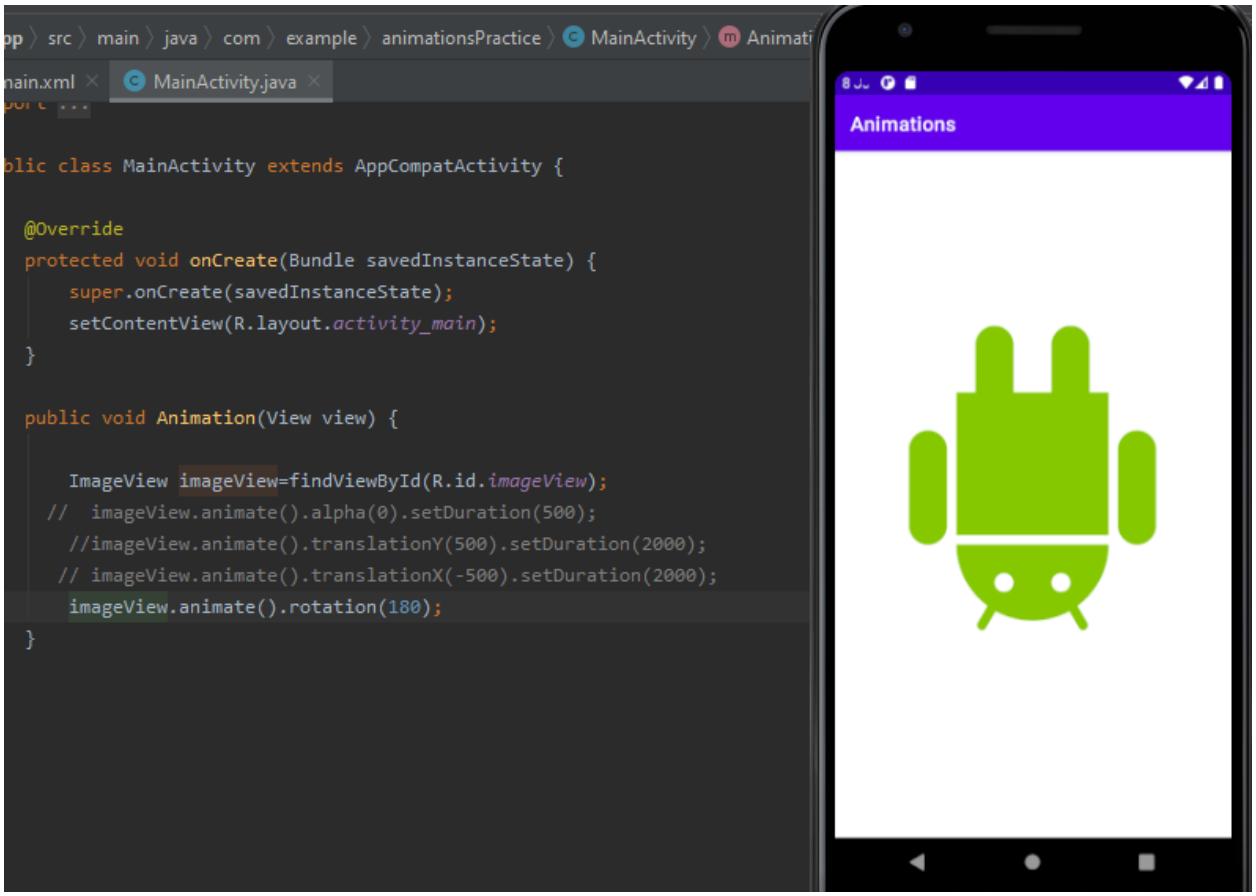


❖ Image Rotation Animation:

- Write the function as below to get rotation effect on image click.



- Image rotated to 180 degree just after click.

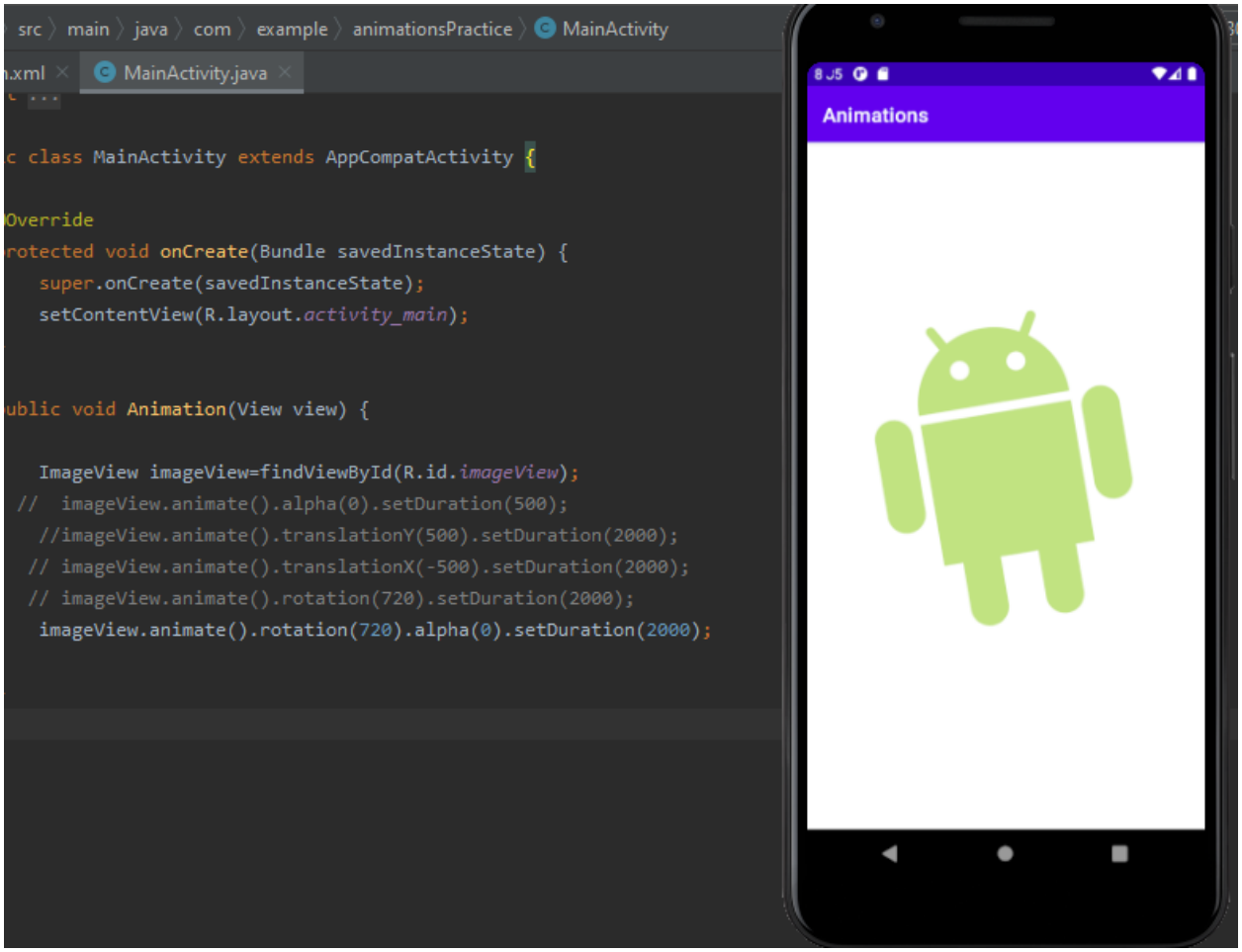


For Double Rotations:

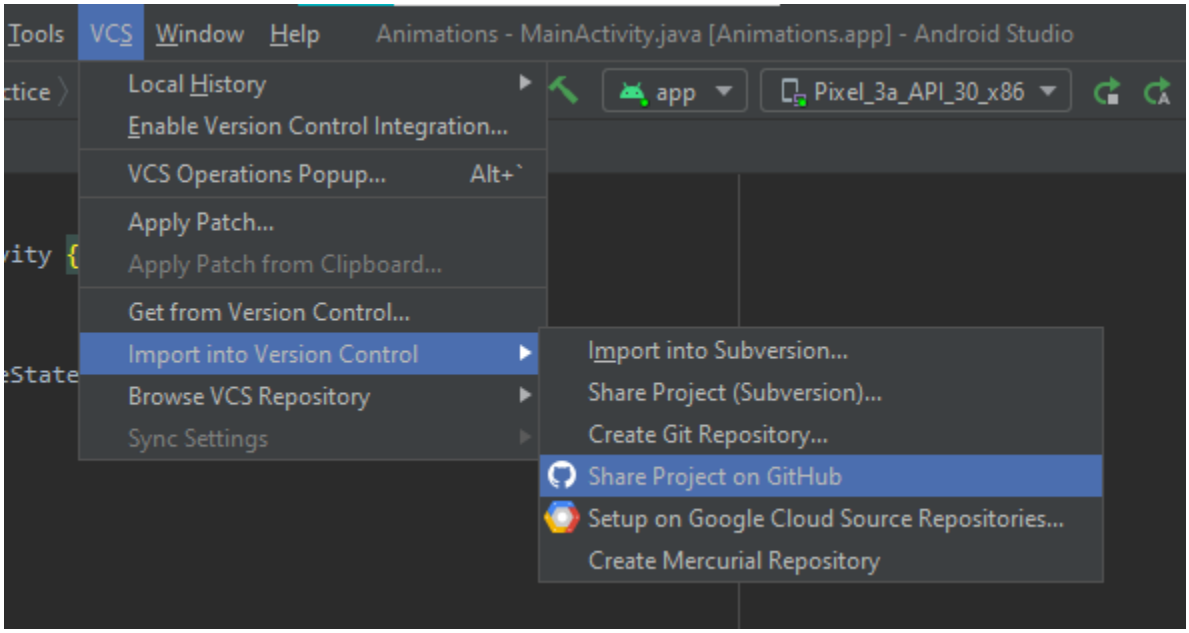
```
public void Animation(View view) {

    ImageView imageView=findViewById(R.id.imageView);
//  imageView.animate().alpha(0).setDuration(500);
  //imageView.animate().translationY(500).setDuration(2000);
 // imageView.animate().translationX(-500).setDuration(2000);

    imageView.animate().rotation(720).setDuration(2000);
```

❖ Image Rotation,Disappear and Duration at same time:

- Write the function as below to get rotation, disappear effect with some duration on image click.
- Image rotated and disappeared in given duration.



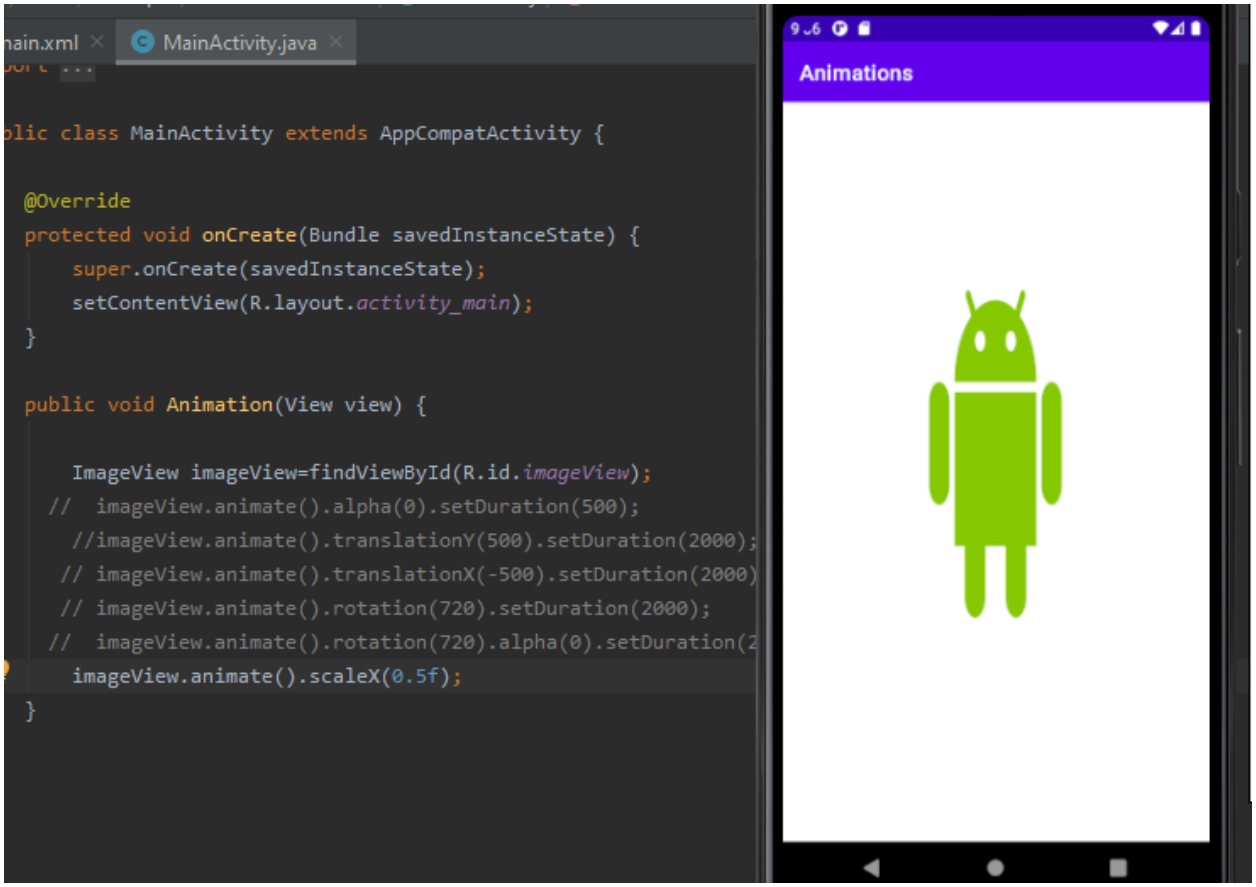
❖ Share to GitHub Using Android IDE:



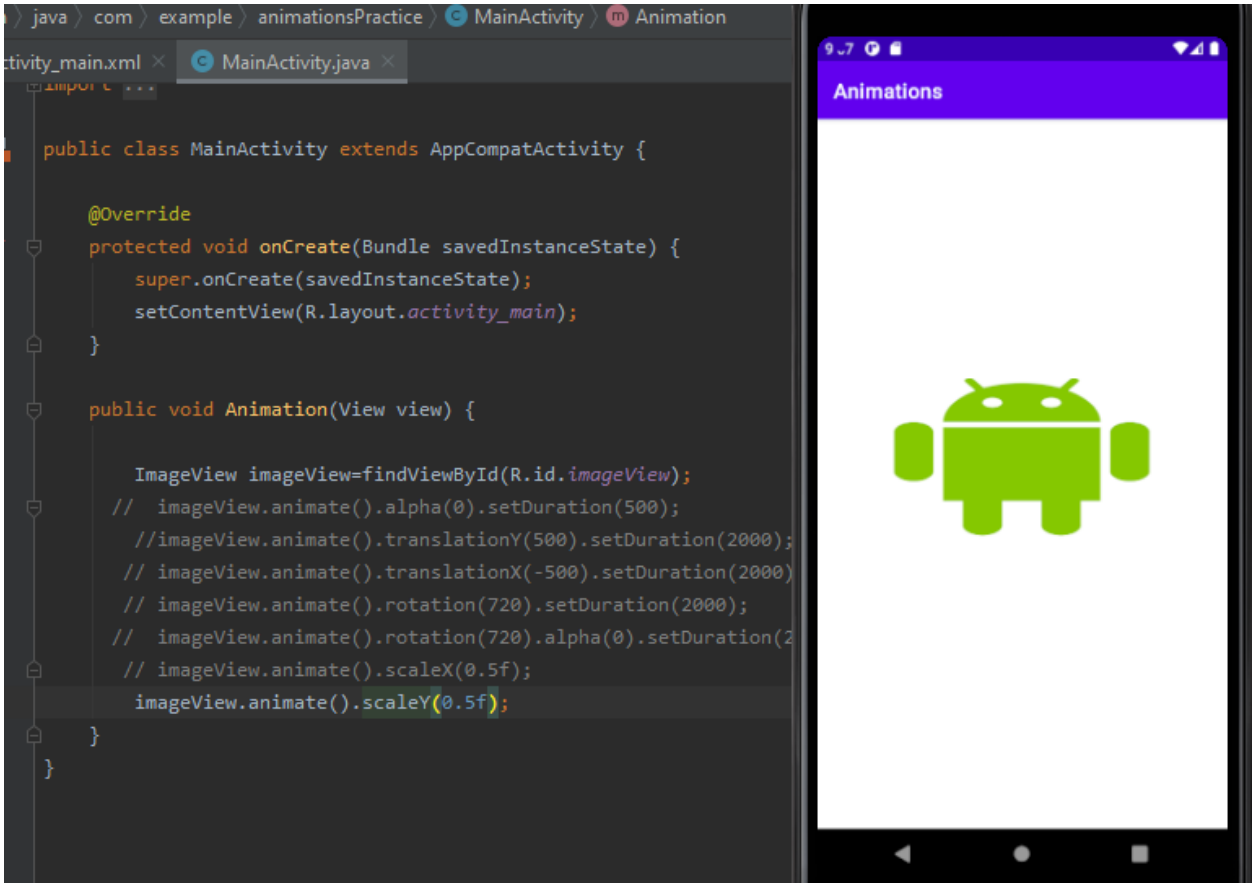
❖ Image Scaling Animation:

- Write the function as below to get scaling effect on image click.

- Image scaled to x-axis just after click.



- Image scaled to y-axis just after click.



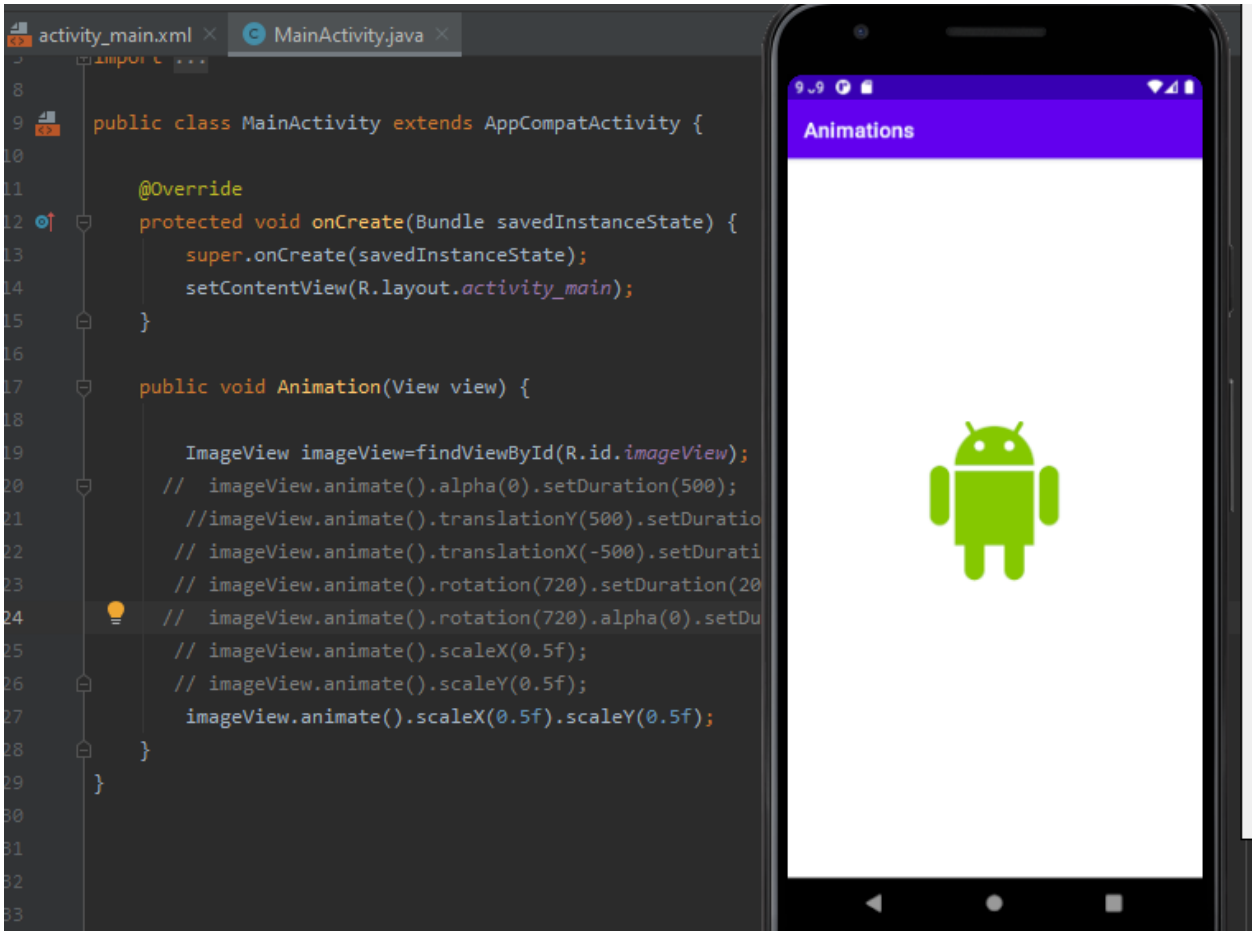
❖ Image Scaling ALong Both XY axis:

- Write the function as below to get scaling effect on image click.

y_main.xml | MainActivity.java

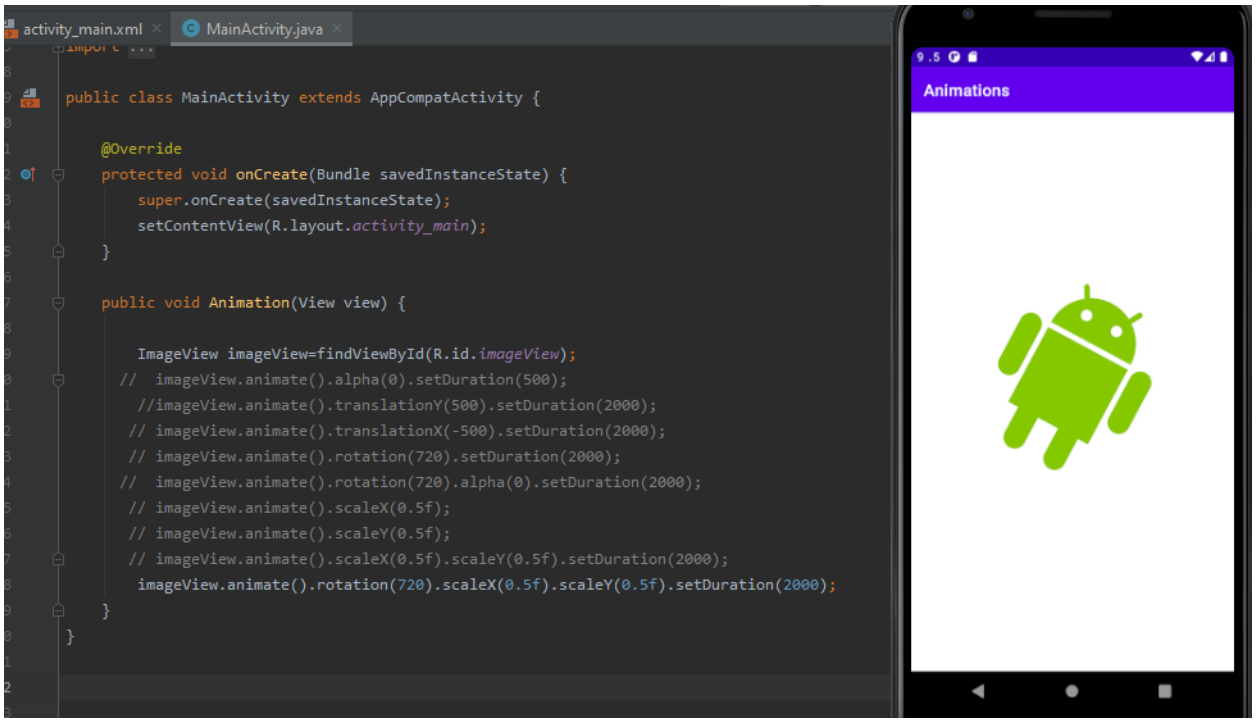
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void Animation(View view) {

        ImageView imageView=findViewById(R.id.imageView);
      //  imageView.animate().alpha(0).setDuration(500);
       //imageView.animate().translationY(500).setDuratio
      // imageView.animate().translationX(-500).setDurati
       // imageView.animate().rotation(720).setDuration(20
      //  imageView.animate().rotation(720).alpha(0).setDu
       // imageView.animate().scaleX(0.5f);
       // imageView.animate().scaleY(0.5f);
        imageView.animate().scaleX(0.5f).scaleY(0.5f);
    }
}
```

- Image scaled to both xy-axis just after click so squeezed.



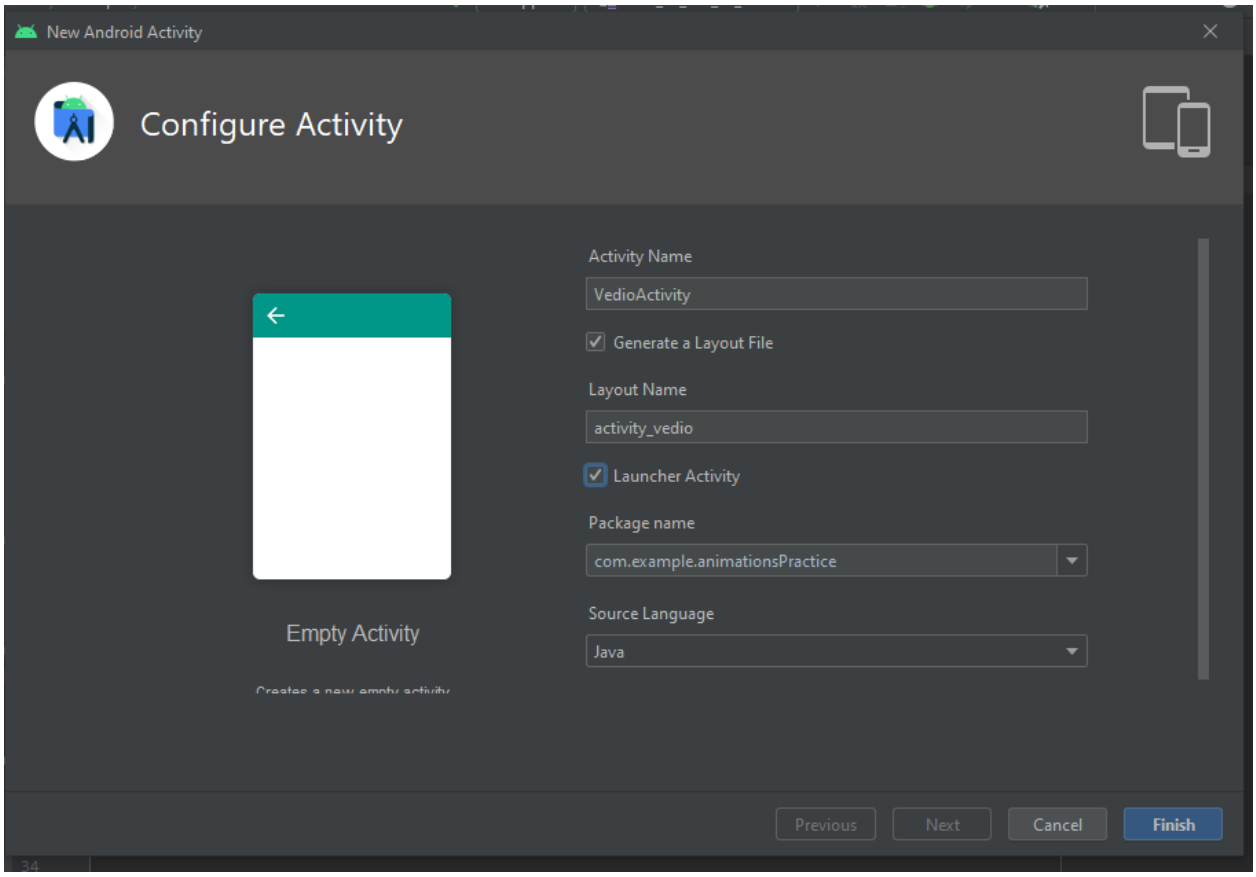
❖ Image Scaling + Rotation at same time:

- Write the function as below to get scaling and rotation effect on image click.
- Image rotated and scaled to both xy-axis just after click.

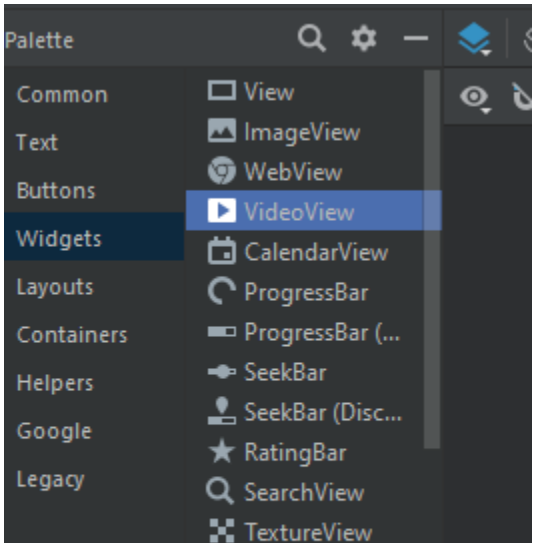


❖ Videos:

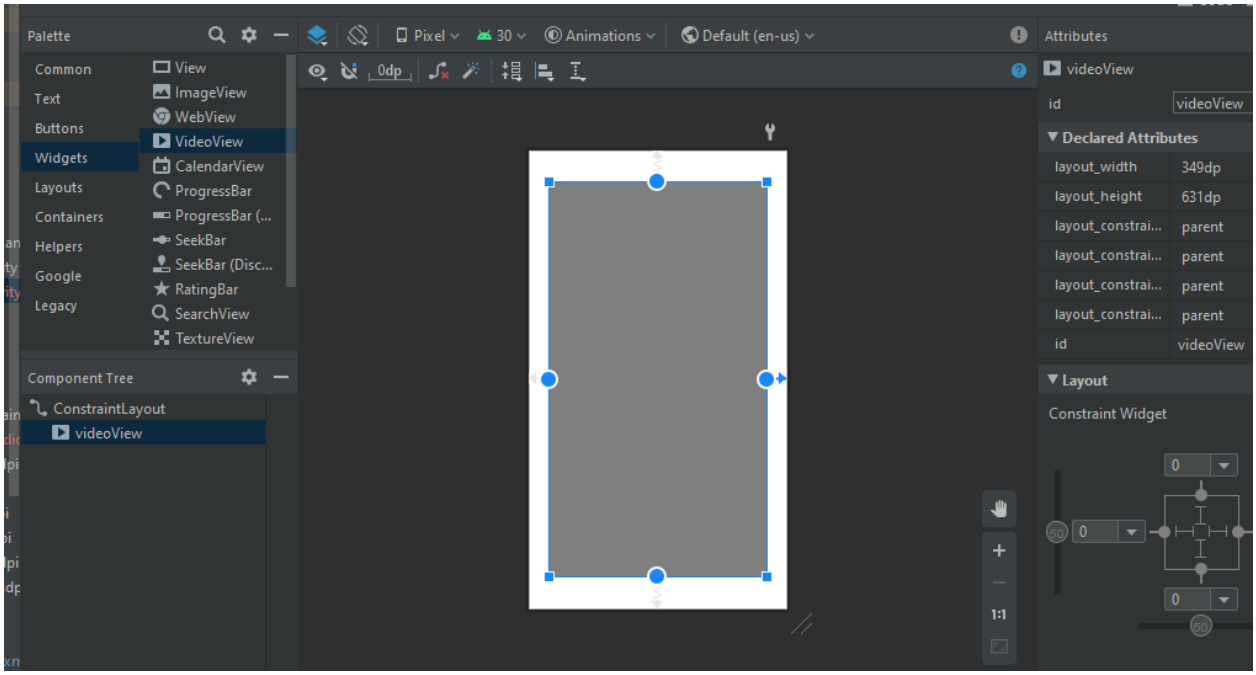
- Create a new activity and set it as launching activity.

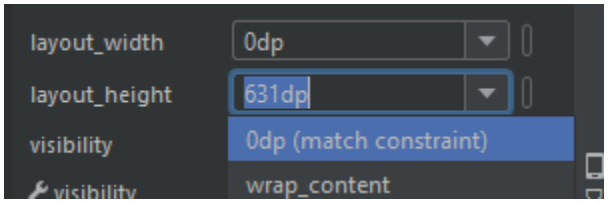


- Add an video view from widgets.

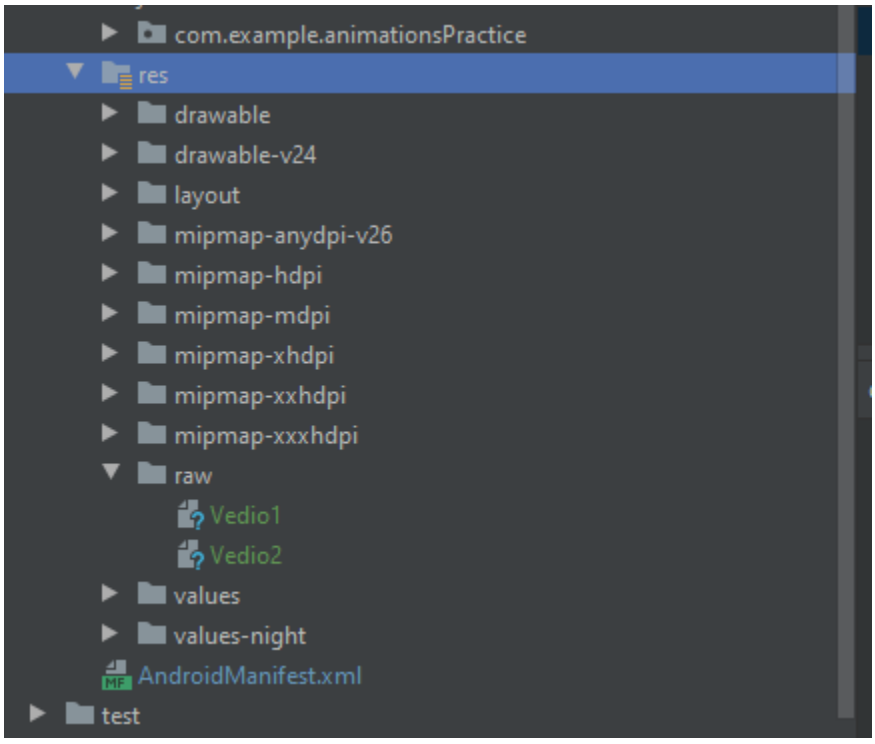


- Set the constraints of view and fix its position.

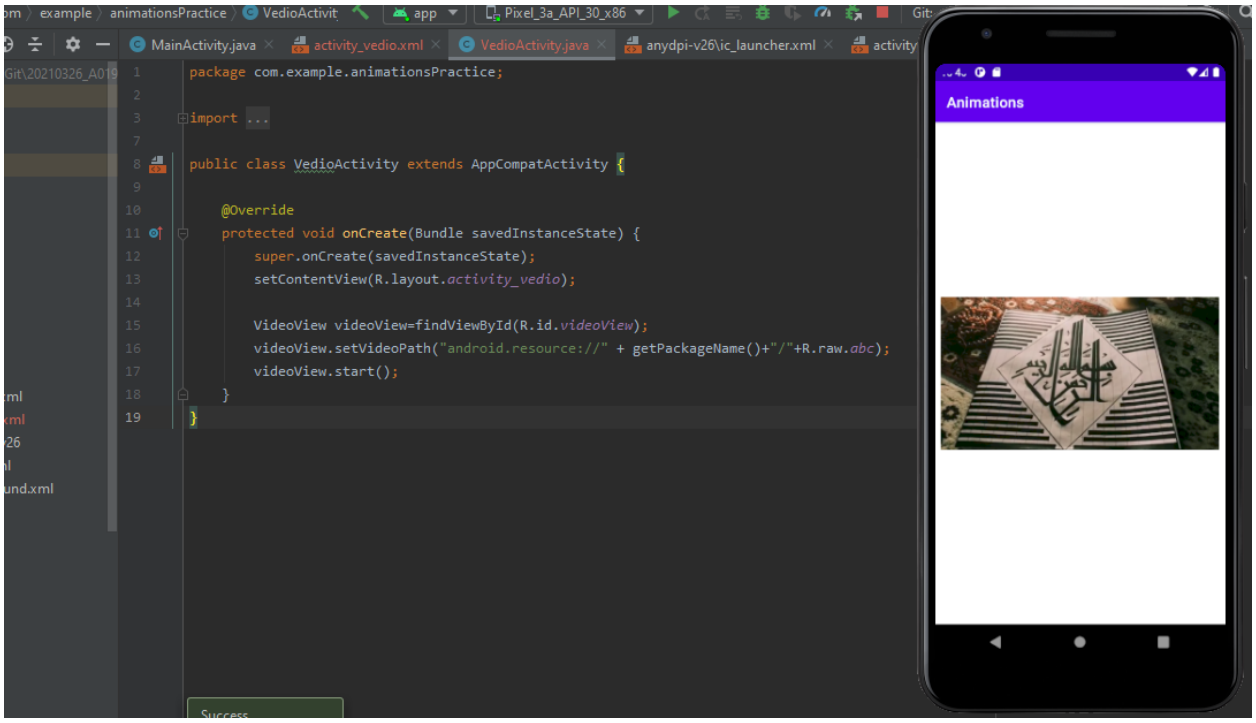




- In **res** folder make a folder named **raw.** Then copy videos you want to add from PC and paste in this folder.



- Add the code below to make a video view on screen. Run it and see the video output on emulator.

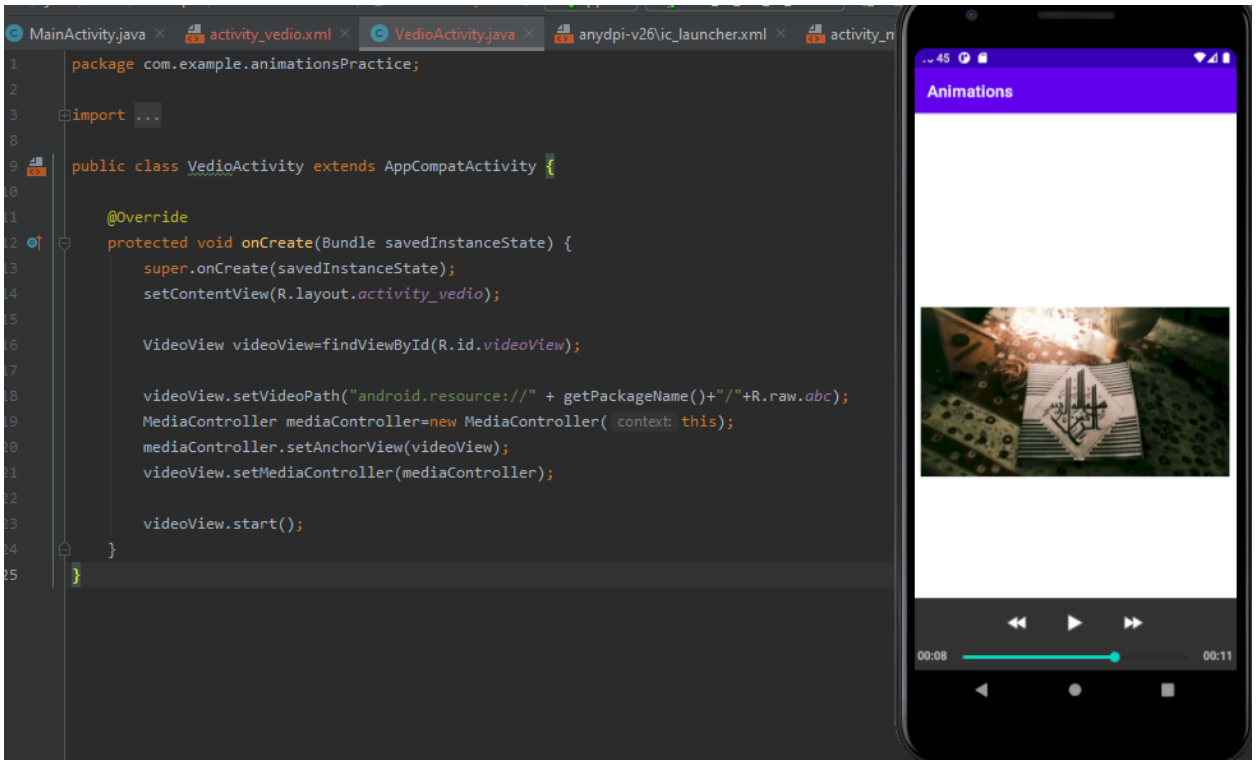


❖ Set Video Controls:

- To set video controls add the code below.

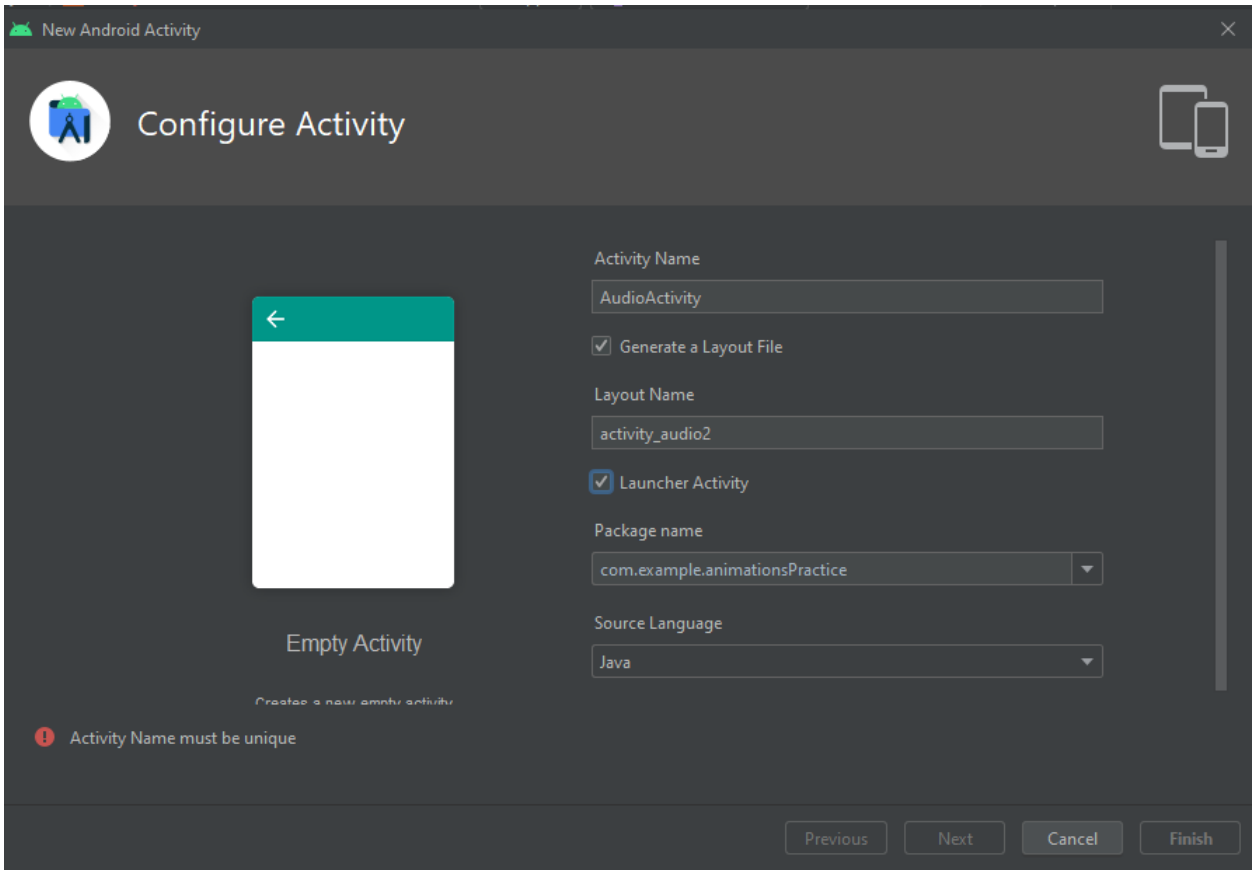
```
public class VedioActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_vedio);

        VideoView videoView=findViewById(R.id.videoView);

        videoView.setVideoPath("android.resource://" + getPackageName()+"/"+R.raw.abc);
        MediaController mediaController=new MediaController( context: this);
        mediaController.setAnchorView(videoView);
        videoView.setMediaController(mediaController);

        videoView.start();
    }
```

- Run it and see the output on emulator, the video with controls.

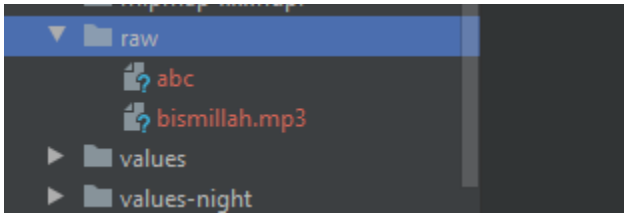


❖ Audio Files:

- Create a new activity and set it as launching activity.



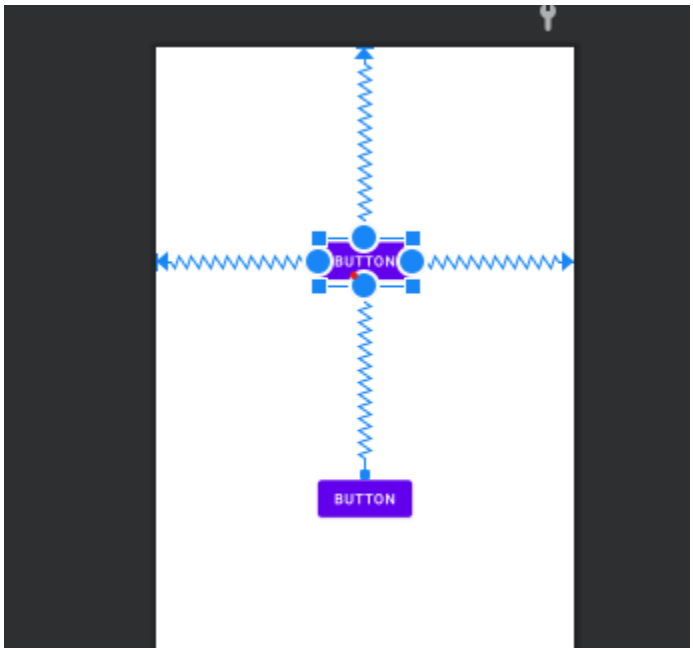
- In **res** folder make a folder named **raw.** Then copy audios you want to add from PC and paste in this folder.



❖ Set Audio Controls:

Android don't have built in functionality so wee need to add buttons to set audio controls.

- Add 2 Buttons on xml file and set their constraints with position.

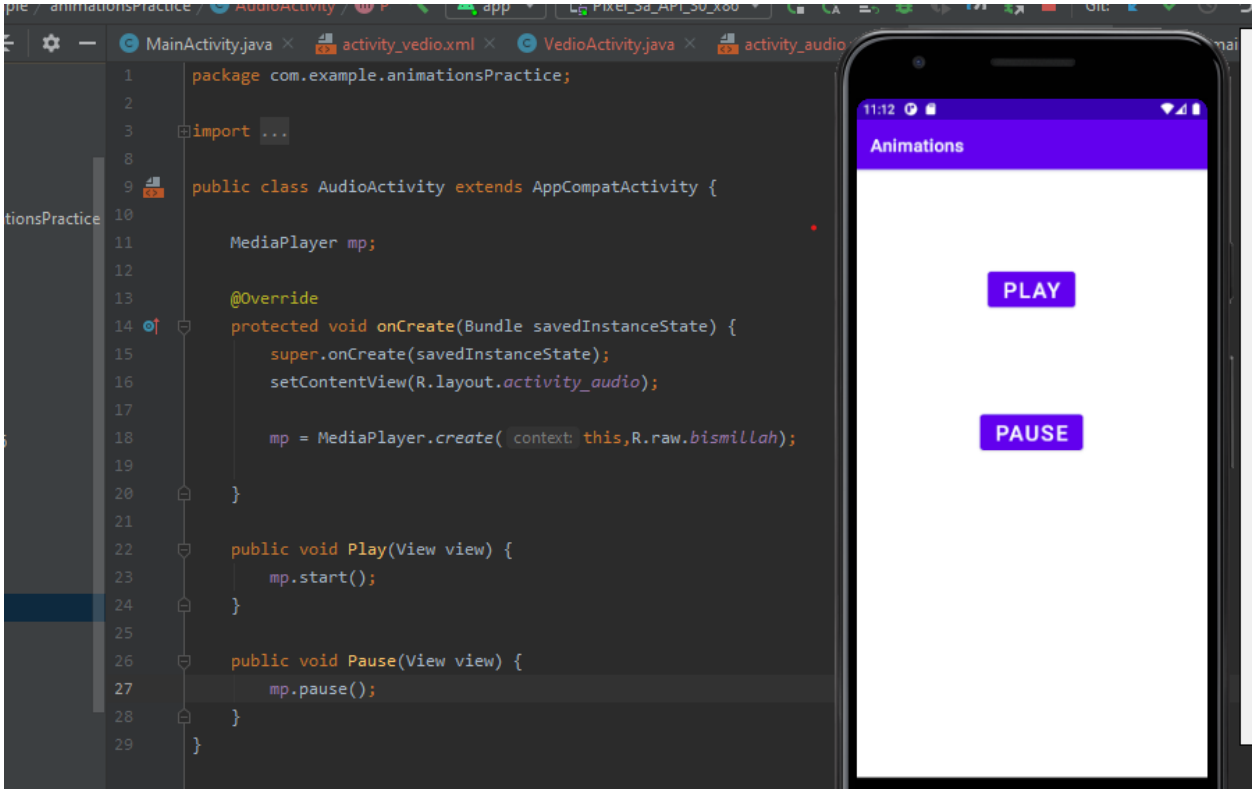


- Now write click event for one button as Play and other button as Pause.

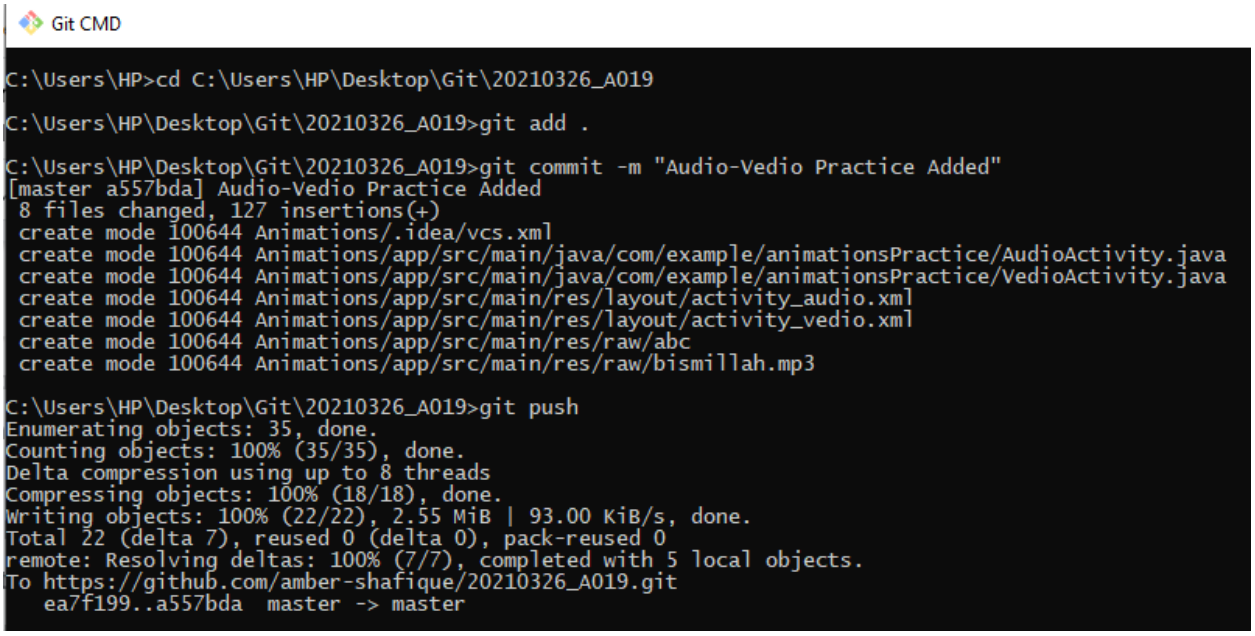
```
<Button
    android:id="@+id/btnPlay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Button"
    android:onClick="Play"
    app:layout_constraintBottom_toTopOf="@+id/BtnPause"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/BtnPause"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="360dp"
    android:text="Button"
    android:onClick="Pause"
```

```
    public void Play(View view) {
        mp.start();
    }

    public void Pause(View view) {
        mp.pause();
    }
}
```

- Write the code to add audio on screen and set controls in functions on clicked by buttons.
- Run the file and see output on emulator. Play button will play the audio and pause will stop the audio.



❖ Uploading the Project on GitHub Repository:

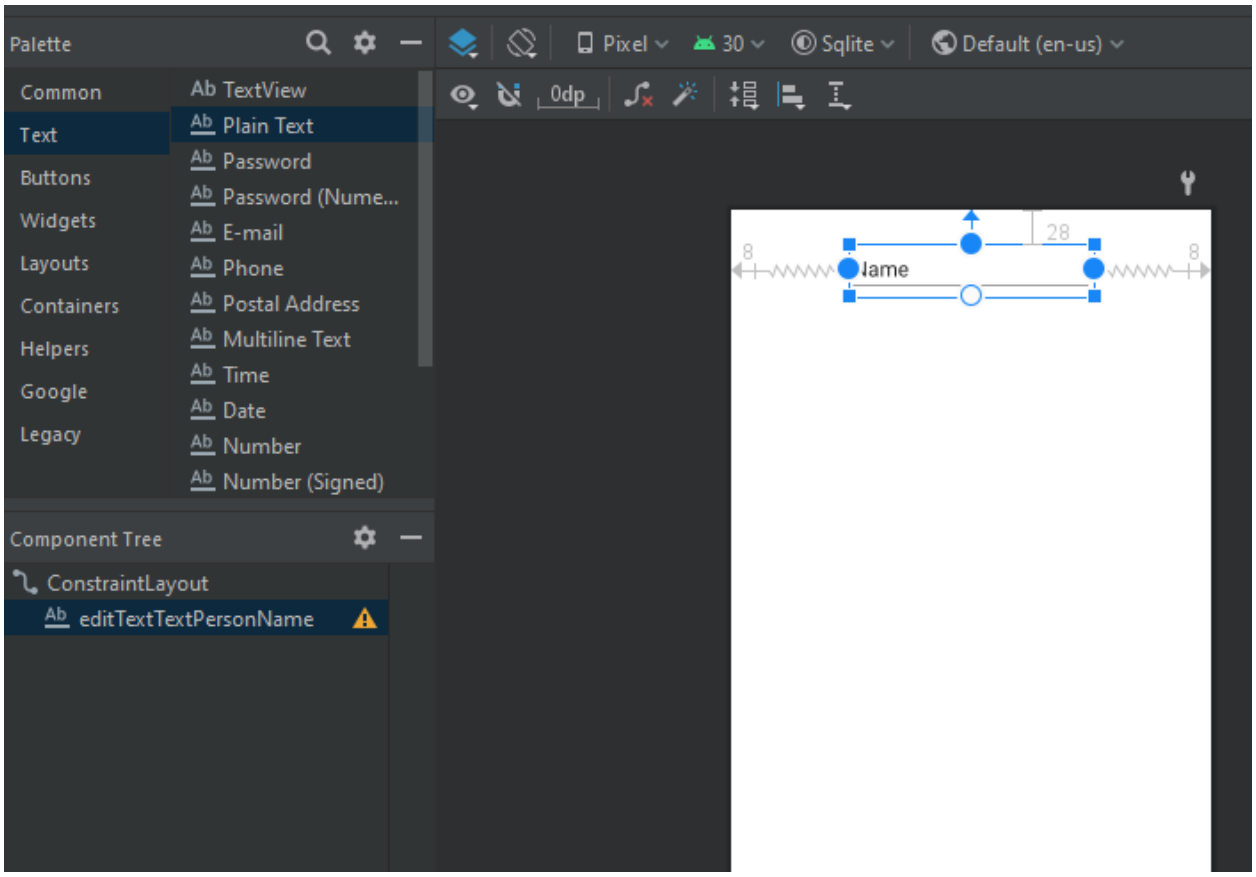


LECTURE#10

❖ SQlite-Lec1:

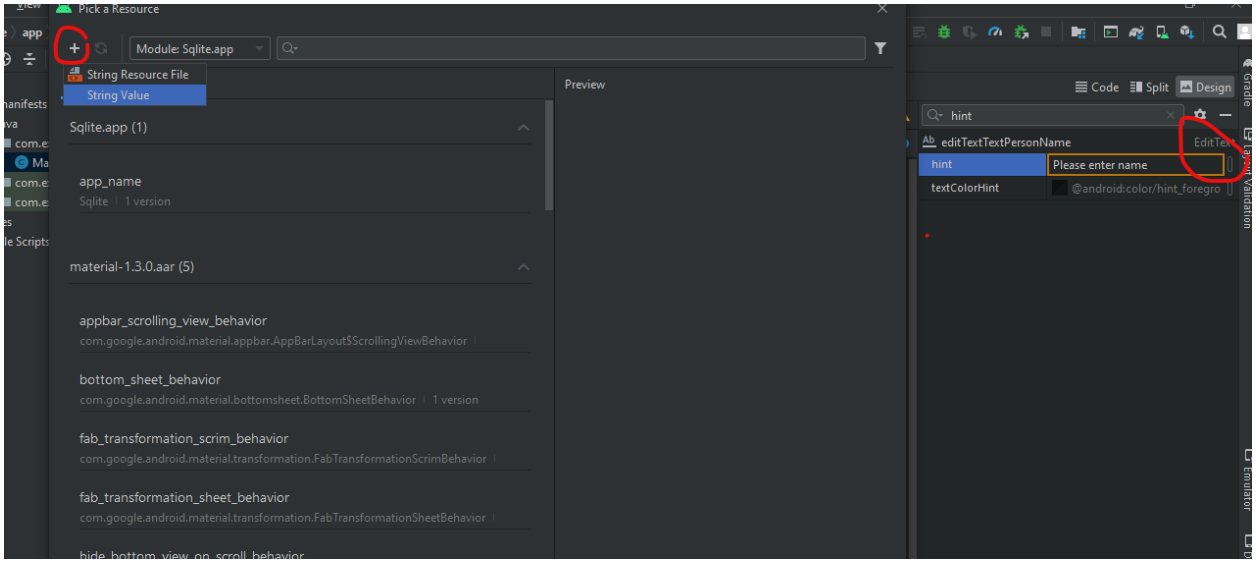
To make a view for entering and retrieving data from database:

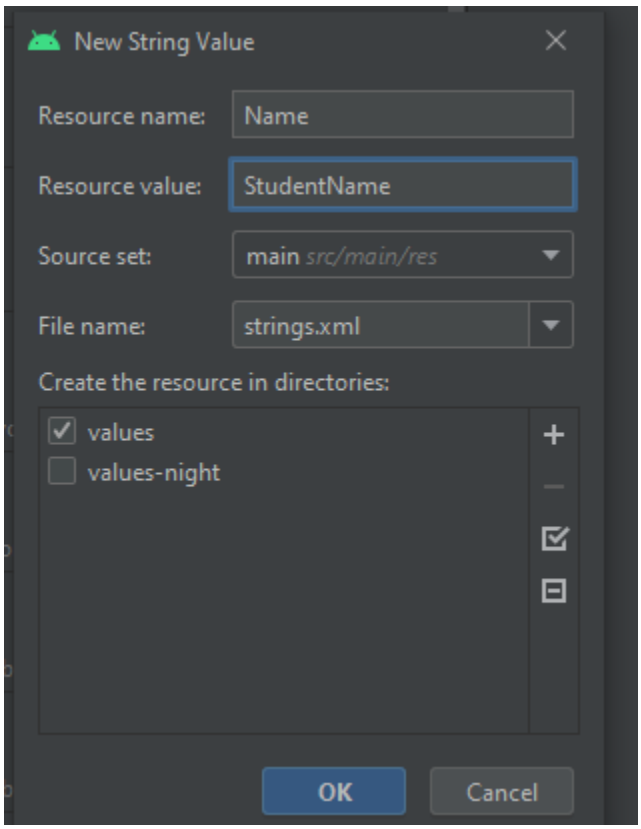
- Add a plain text field and set its constraints.



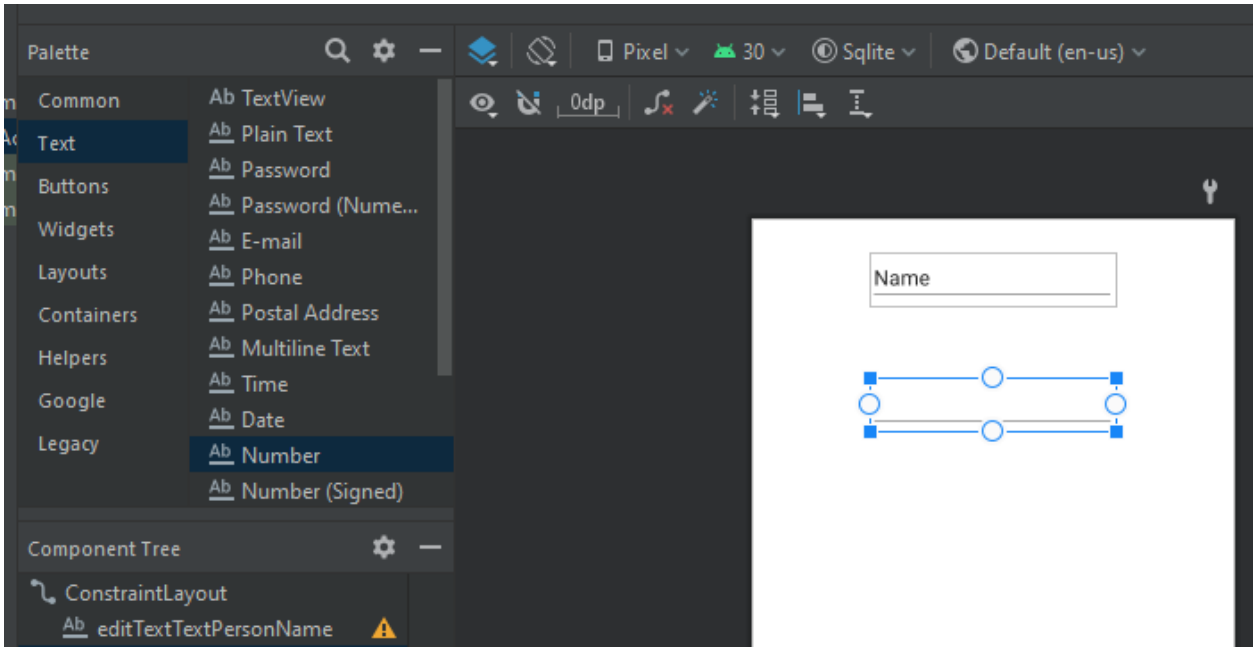
- **Add new string resource:**

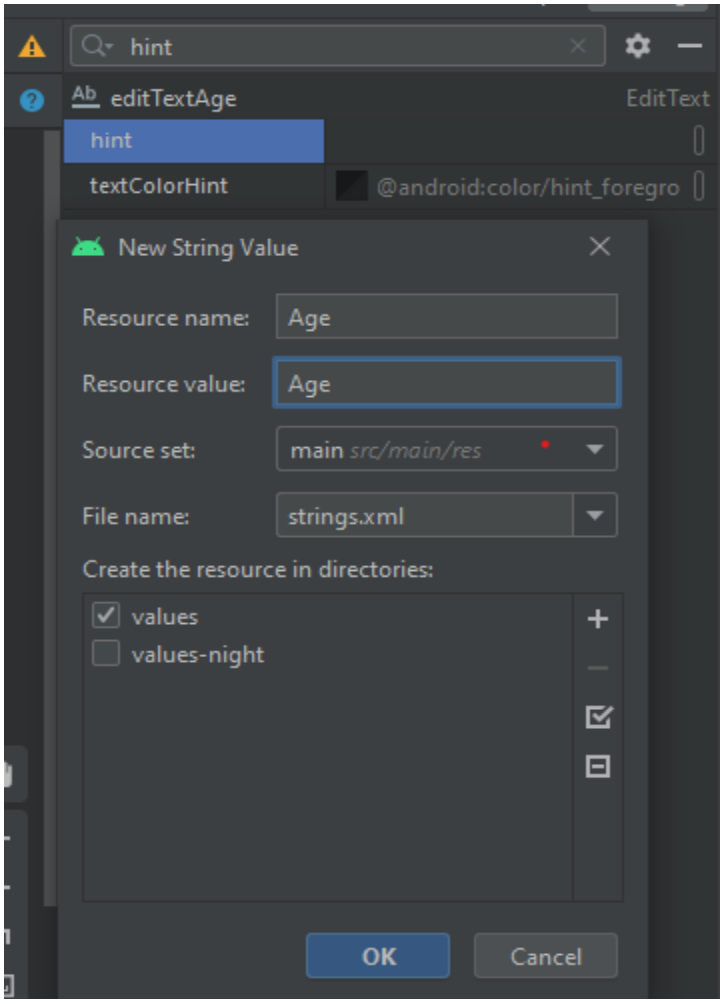
  To save the id of the text field create a new string resource.





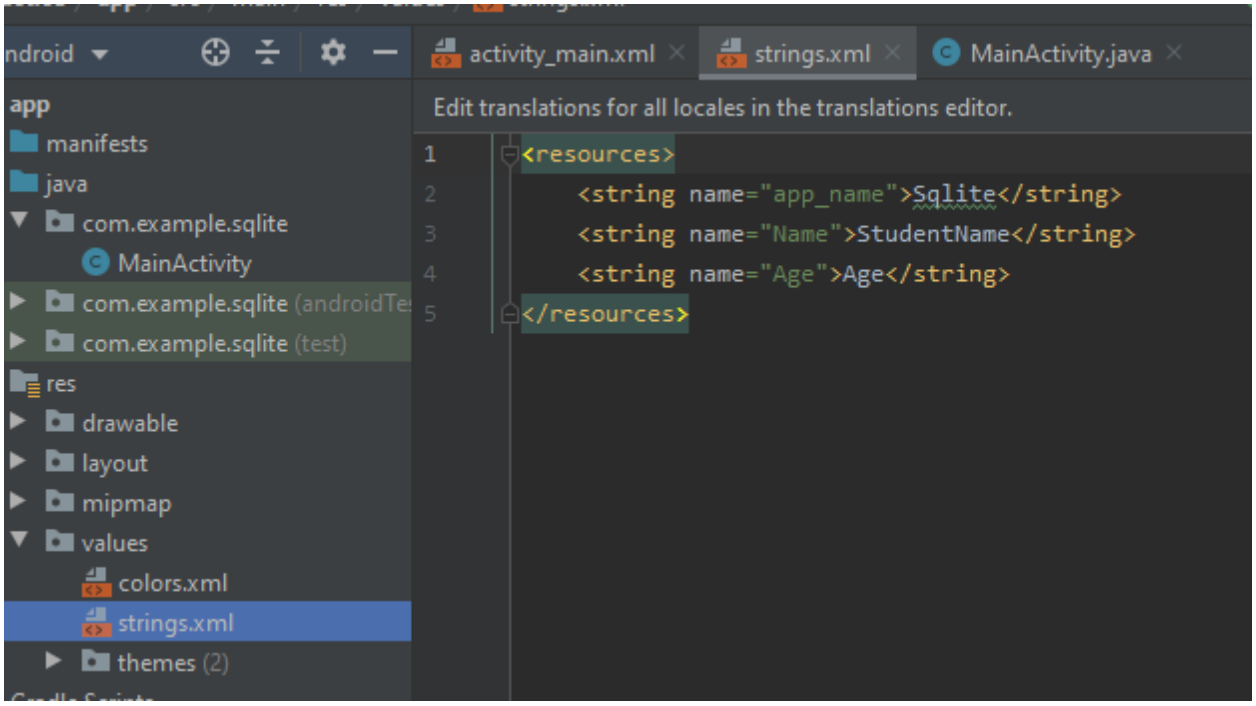
- Add another field of number & set its constraints too.Also add new string resource for it.



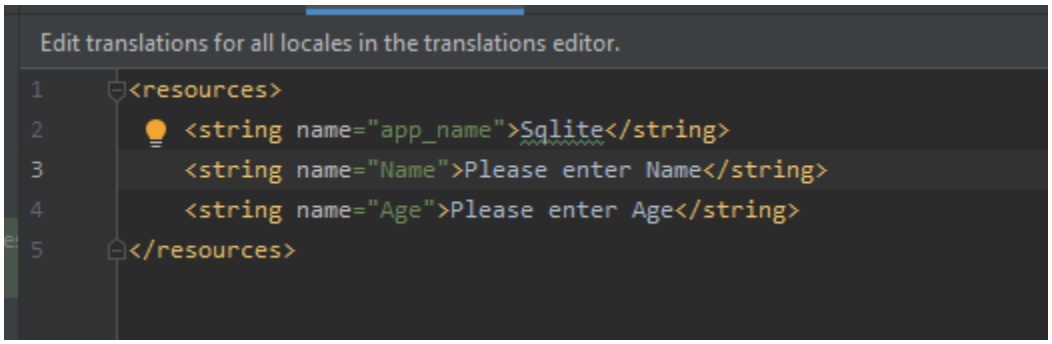


- **Viewing String Resources:**

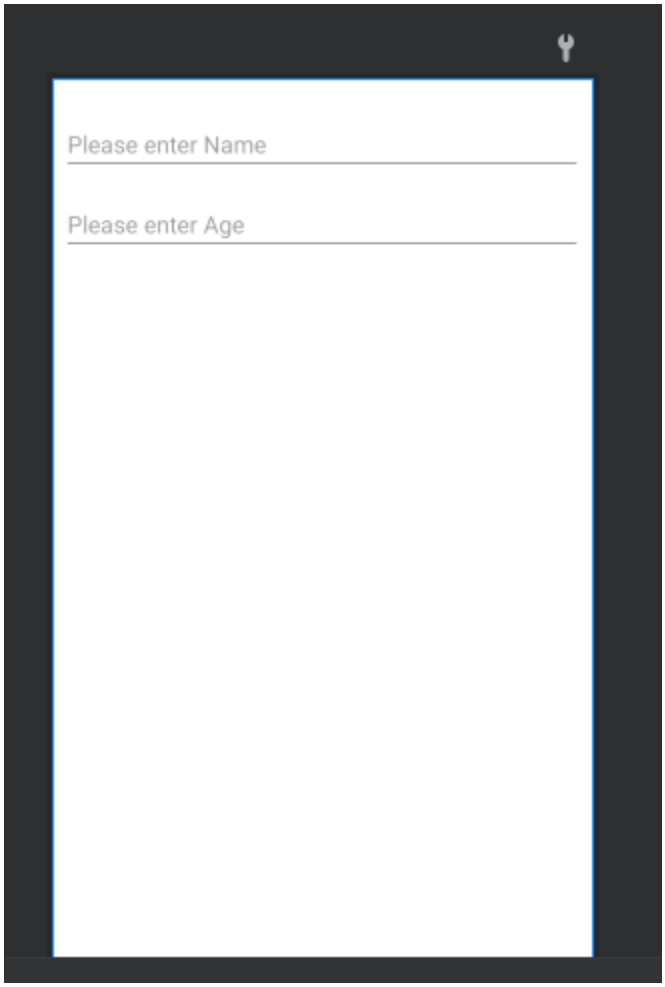
In resources=> values, string.xml file will be saving string resources.



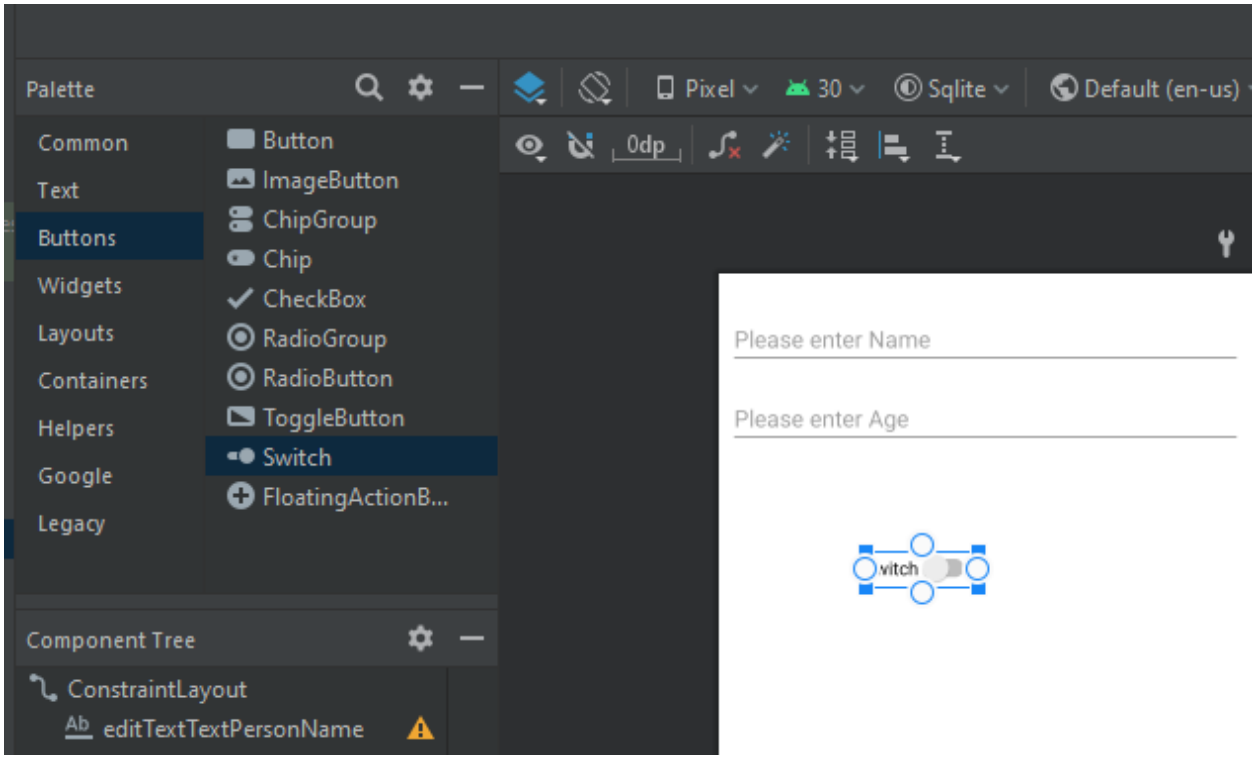
Set the default text for these strings and save.



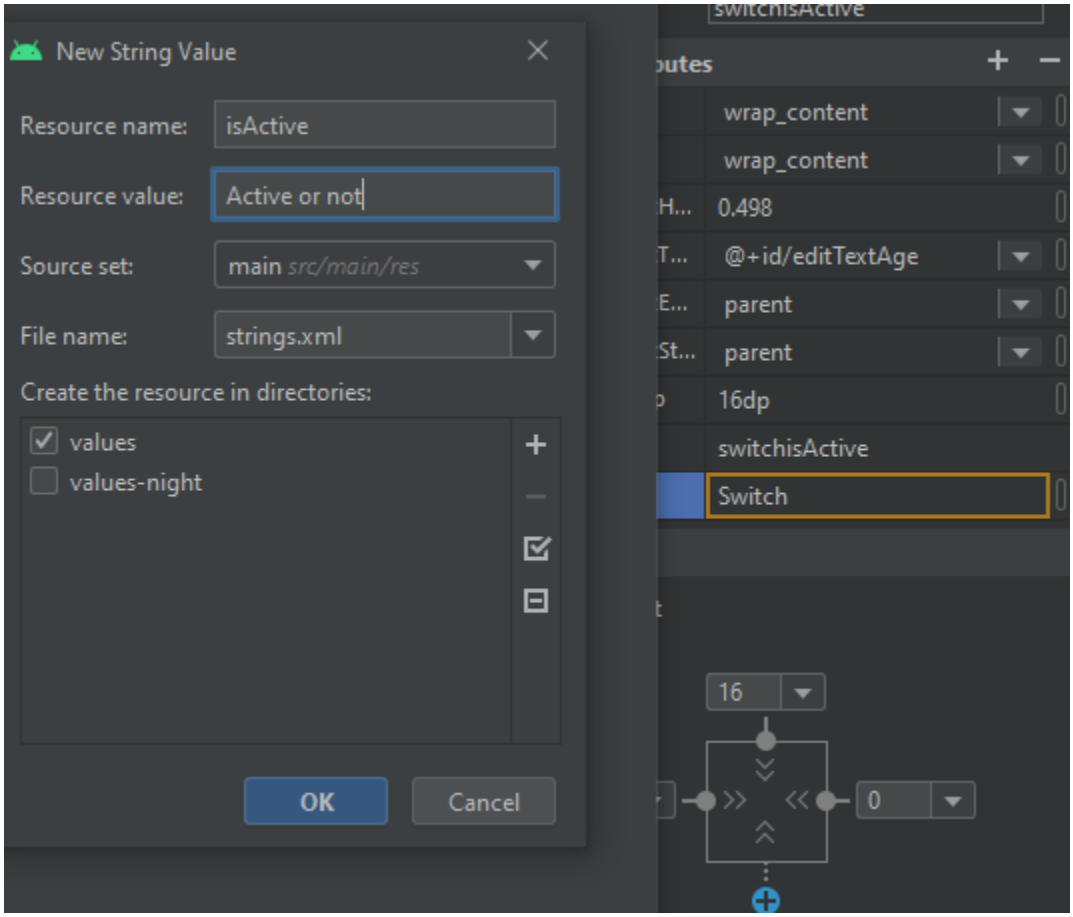
- After adding default text for strings run the emulator and see the output on screen.



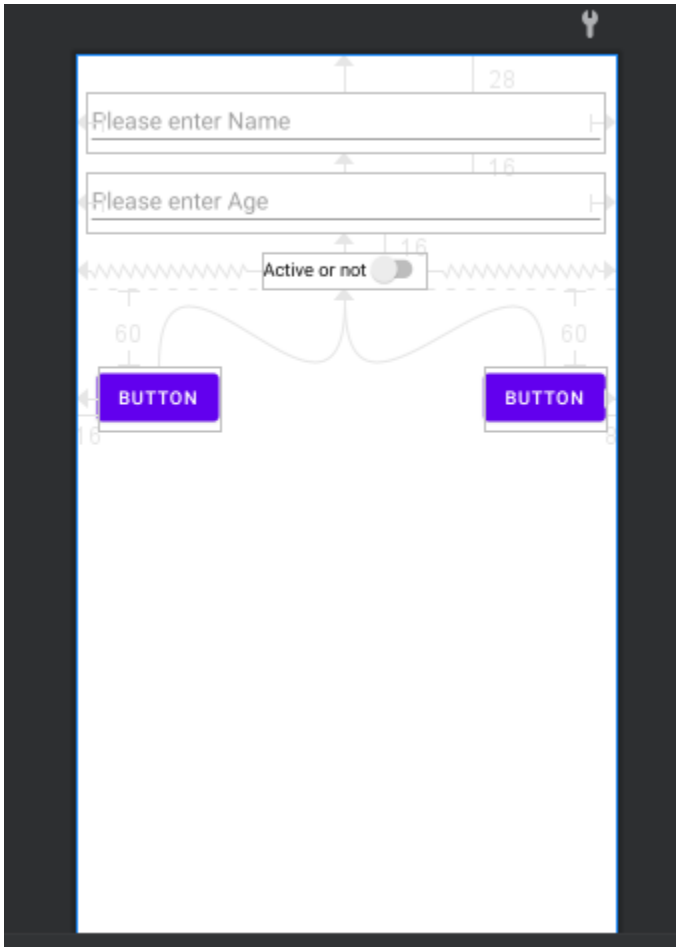
- Now add another field of switch on screen from the pallet.



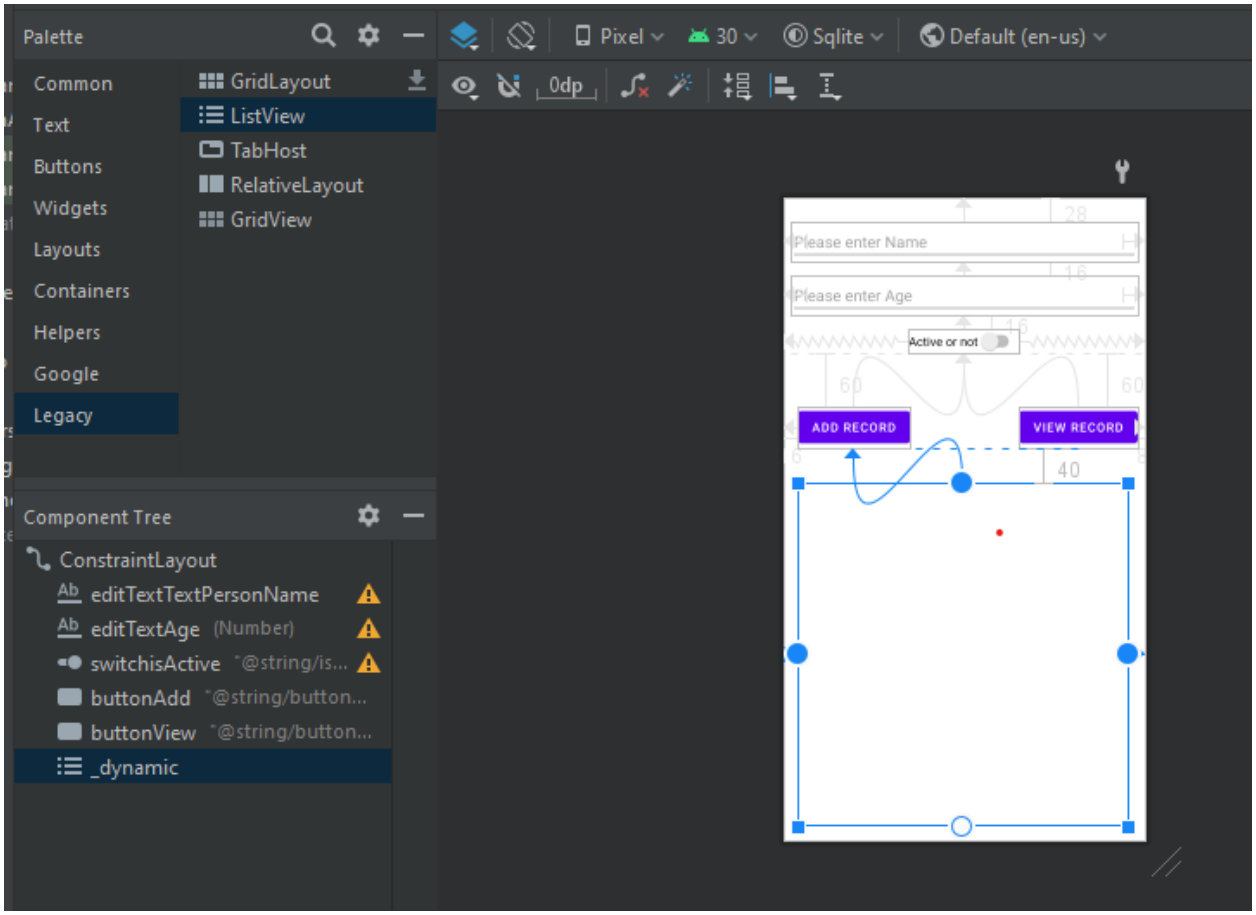
- Add a string resource for switch also.



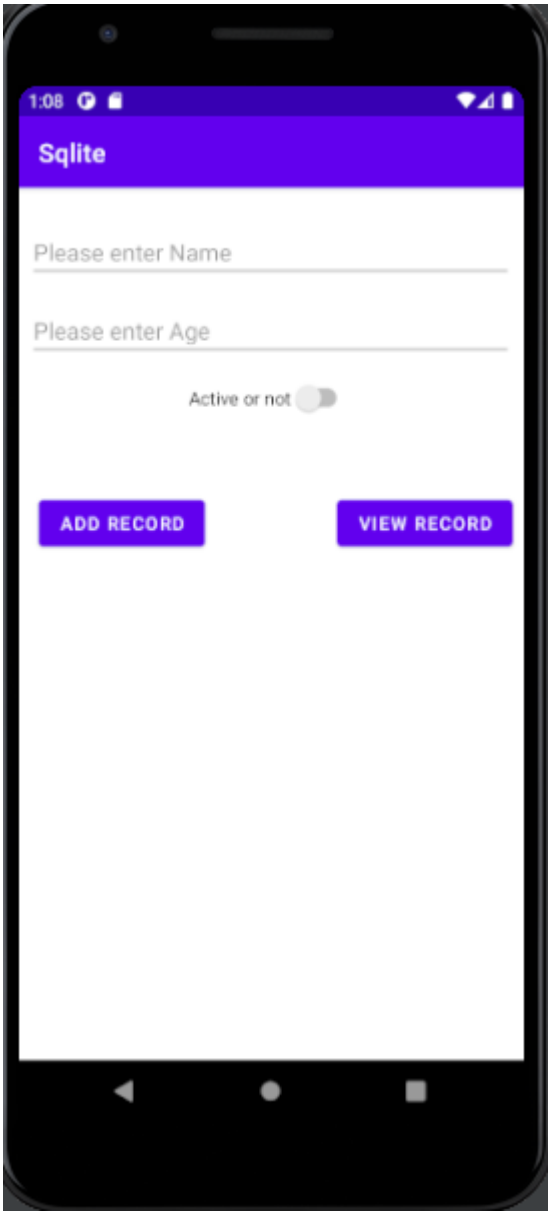
- Now add two buttons on the screen and set their constraints.



- Add a list view and set its constraints with buttons.



- Now run the emulator to see the whole srcreen view.

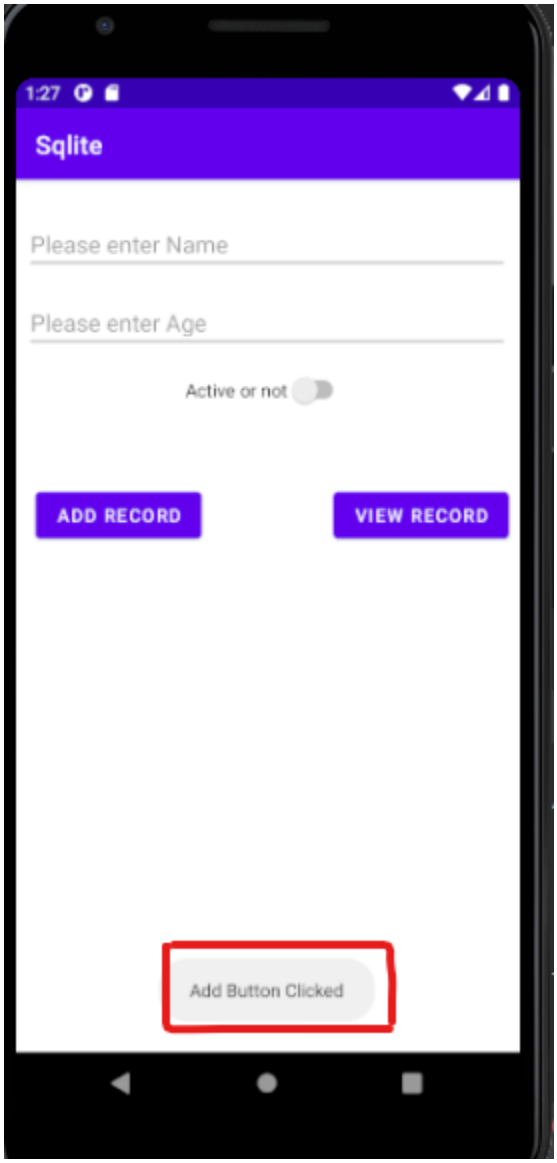


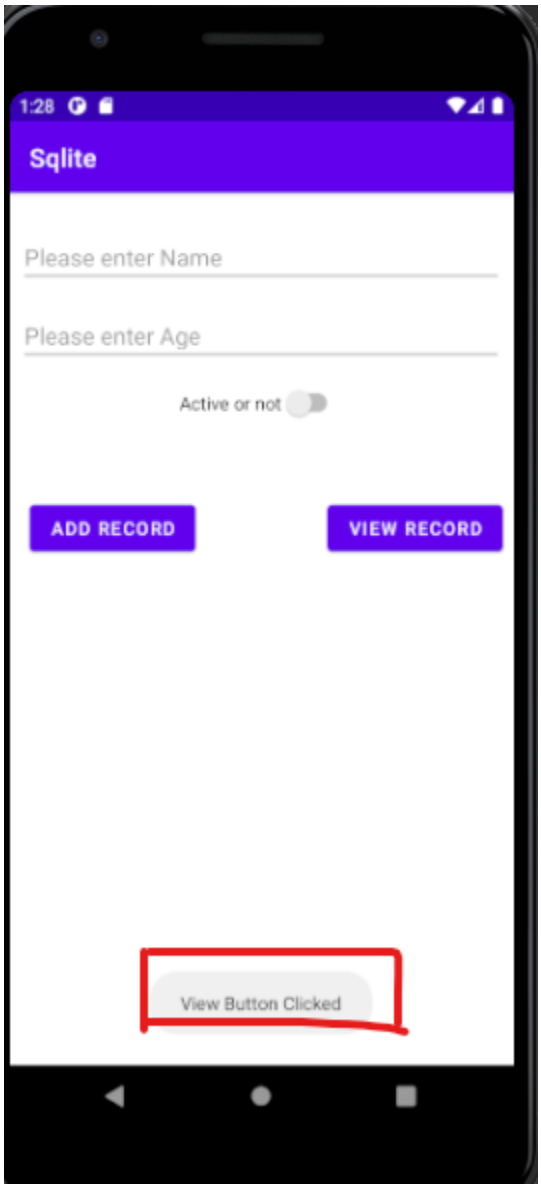
- **Using toast instead of log:**

  To see the click events happening on screen add toast function instead of log.

```
btnAdd= findViewById(R.id.buttonAdd);
btnView=findViewById(R.id.buttonView);

btnAdd.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        Toast.makeText( context: MainActivity.this, text: "Add Button Clicked",Toast.LENGTH_SHORT).show();
    }
});
btnView.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {

        Toast.makeText( context: MainActivity.this, text: "View Button Clicked",Toast.LENGTH_SHORT).show();
    }
});
```

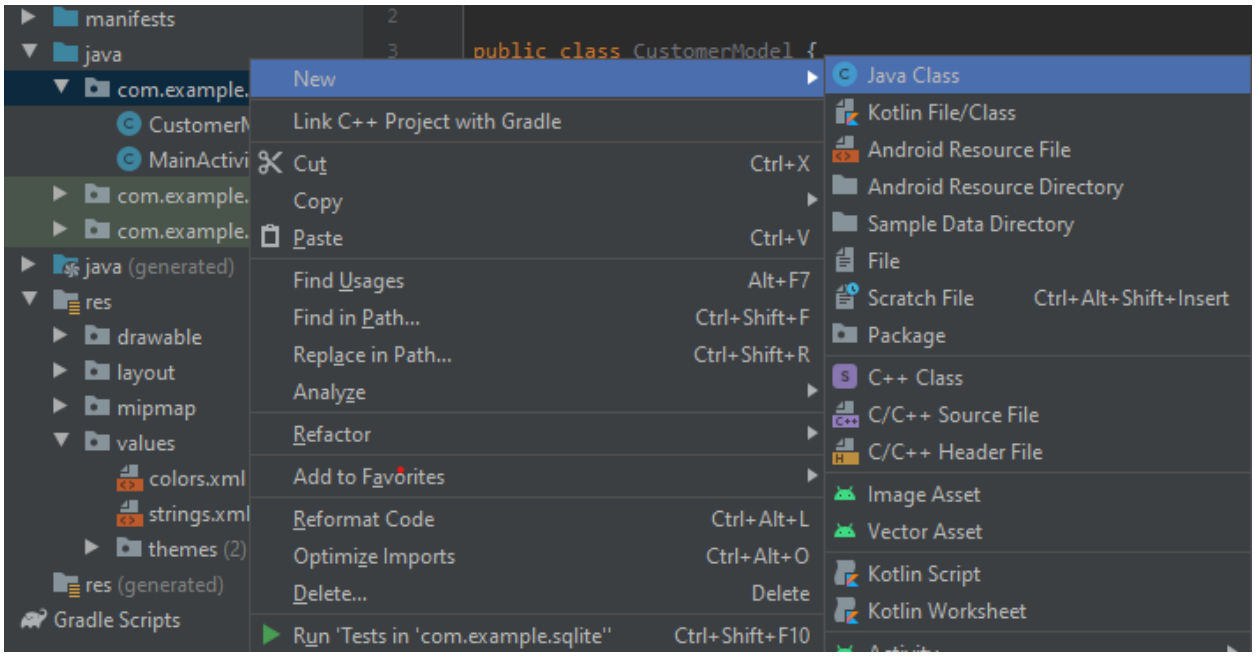
- After adding toast on click events of button run the emulator to see output.
- Just when you click the button the toast function will display a text notification in the bottom to let us know that which button is being clicked.

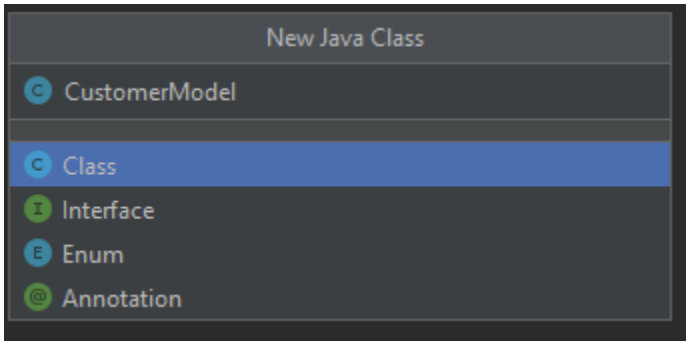




❖ SQlite-Lec2:

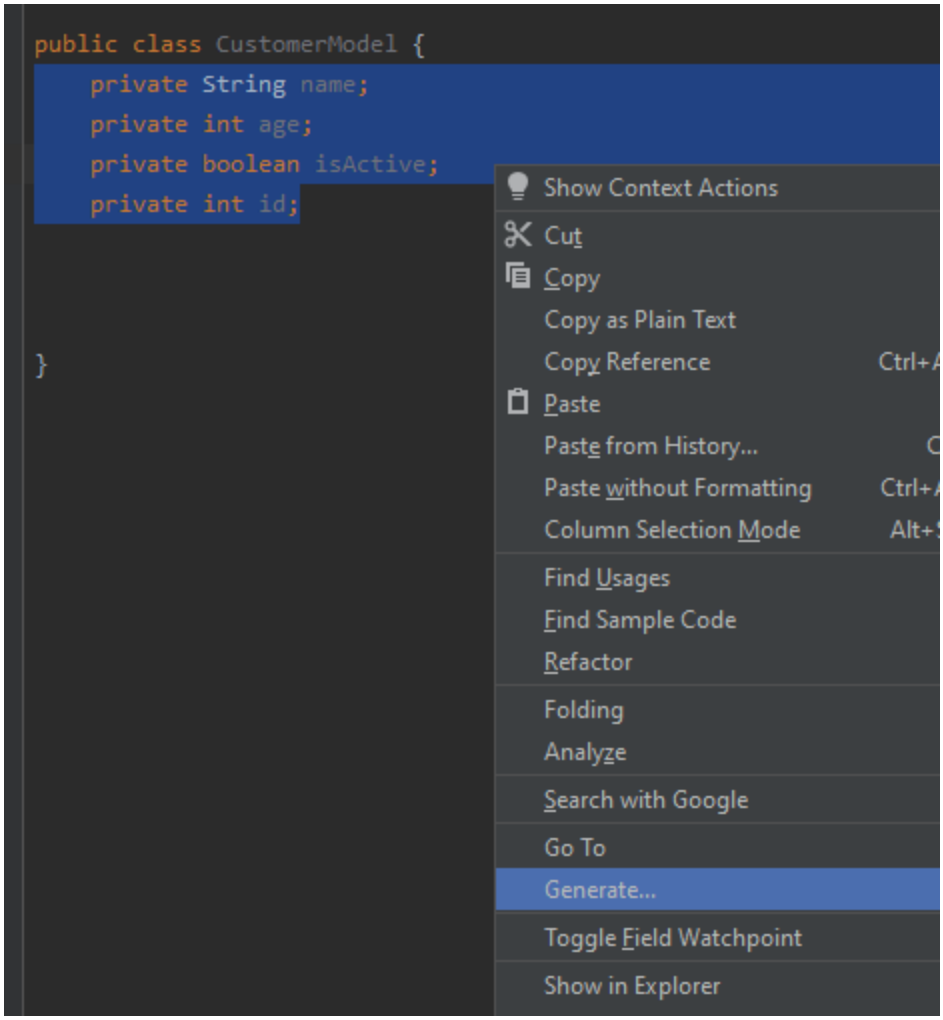
- **Retrieving Values from the model:**
  - Add a new Java class named as customerModel.

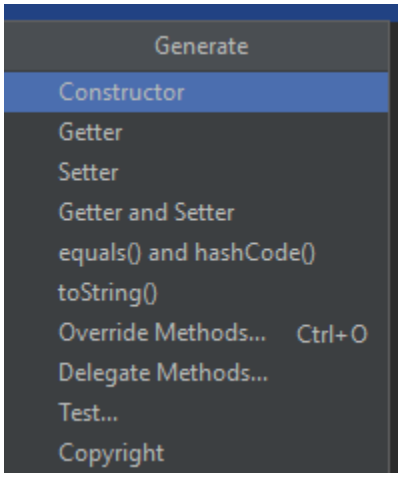


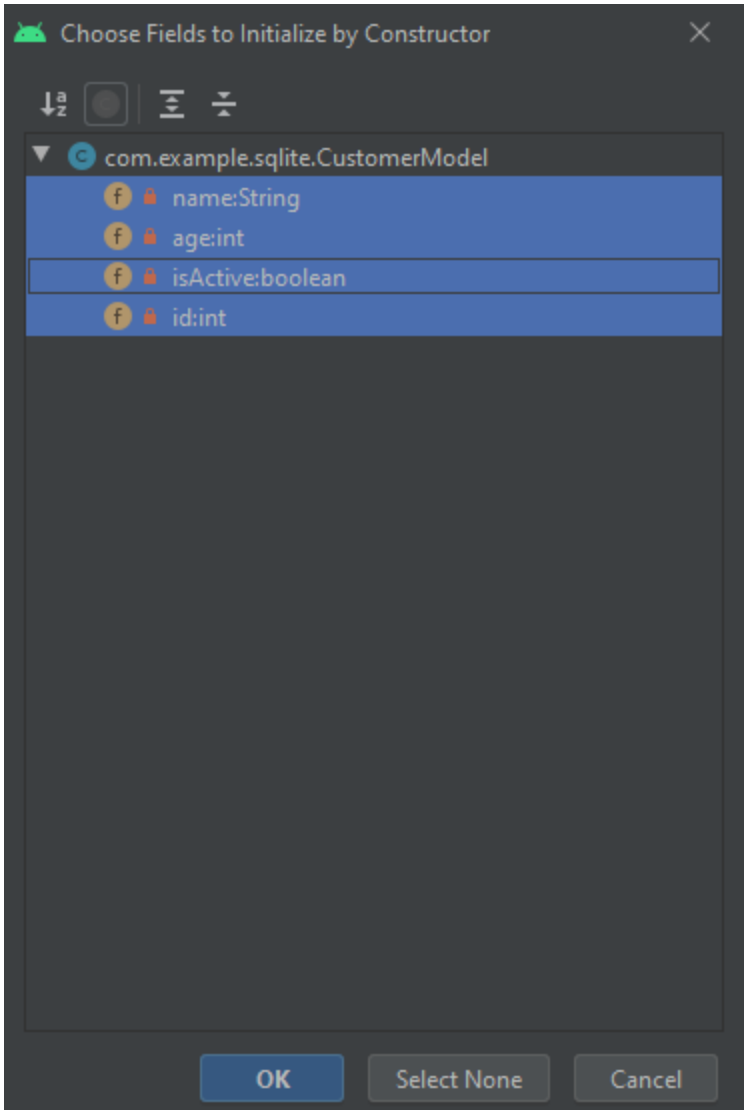


❖ Adding Constructor:

- After making a new java class add constructor using following steps.


public class CustomerModel {
private String name;
private int age;
private boolean isActive;
private int id;
}
Show Context Actions
Cut
Copy
Copy as Plain Text
Copy Reference
Paste
Paste from History...
Paste without Formatting
Column Selection Mode
Find Usages
Find Sample Code
Refactor
Folding
Analyze
Search with Google
Go To
Generate...
Toggle Field Watchpoint
Show in Explorer


Generate
Constructor
Getter
Setter
Getter and Setter
equals() and hashCode()
toString()
Override Methods... Ctrl+O
Delegate Methods...
Test...
Copyright

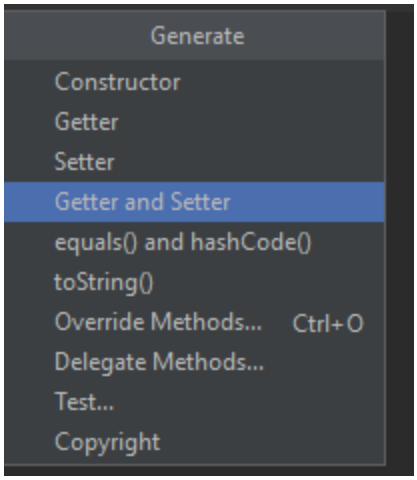


- After following above steps, following is the constructor that has been added to the class.

```
public CustomerModel(String name, int age, boolean isActive, int id) {
    this.name = name;
    this.age = age;
    this.isActive = isActive;
    this.id = id;
}
```

❖ Adding Getter Setter:

- To add getter setter in tha java class following are the steps to follow.

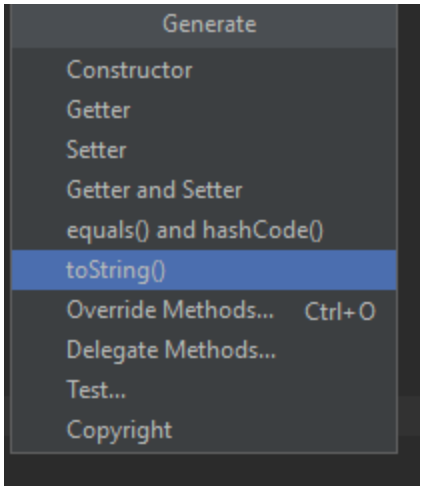


- After following the above steps, following is the view of getter setters that have been added to java class.

```
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}
```

Override toString method:

Now override the toString() method in class using following steps.



- After following the above steps, following is the view of toString() method that have been added to java class.

```
@Override
public String toString() {
    return "CustomerModel{" +
            "name='" + name + '\'' +
            ", age=" + age +
            ", isActive=" + isActive +
            ", id=" + id +
            '}';
}
}
```

❖ Adding class object in main activity:

- Add instance of the above class in main activity so we can get its values.

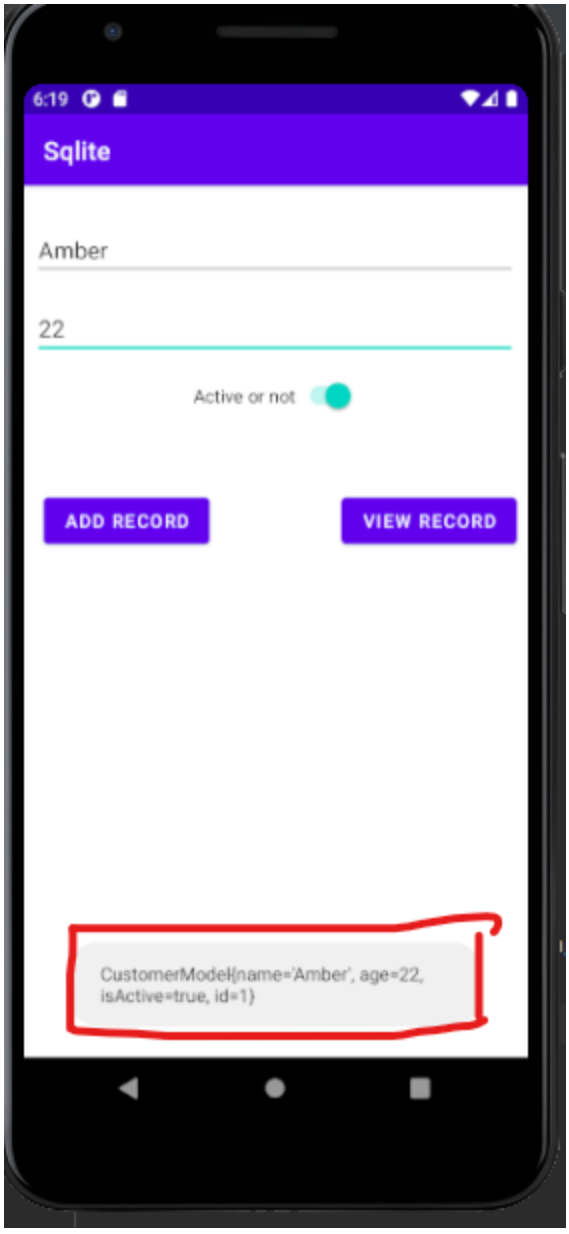
```
setContentView(R.layout.activity_main);

btnAdd= findViewById(R.id.buttonAdd);
btnView=findViewById(R.id.buttonView);
editName=findViewById(R.id.editTextName);
editAge=findViewById(R.id.editTextAge);
switchIsActive=findViewById(R.id.switchisActive);

btnAdd.setOnClickListener(new View.OnClickListener()
{
    CustomerModel customerModel;

    @Override
    public void onClick(View view) {

        CustomerModel customerModel=new CustomerModel(editName.getText().toString(), Integer.parseInt(editAge.getText().toString()),switchIsActive.isChecked(), id: 1);

      //  Toast.makeText(MainActivity.this,"Add Button Clicked",Toast.LENGTH_SHORT).show();
    }
});
```

❖ Add new toast to see data added:

- Now to see the data added on click of button we add a new toast to get string values that have been passed to customer's object.

```
Toast.makeText( context: MainActivity.this,customerModel.toString(),Toast.LENGTH_LONG).show();
```

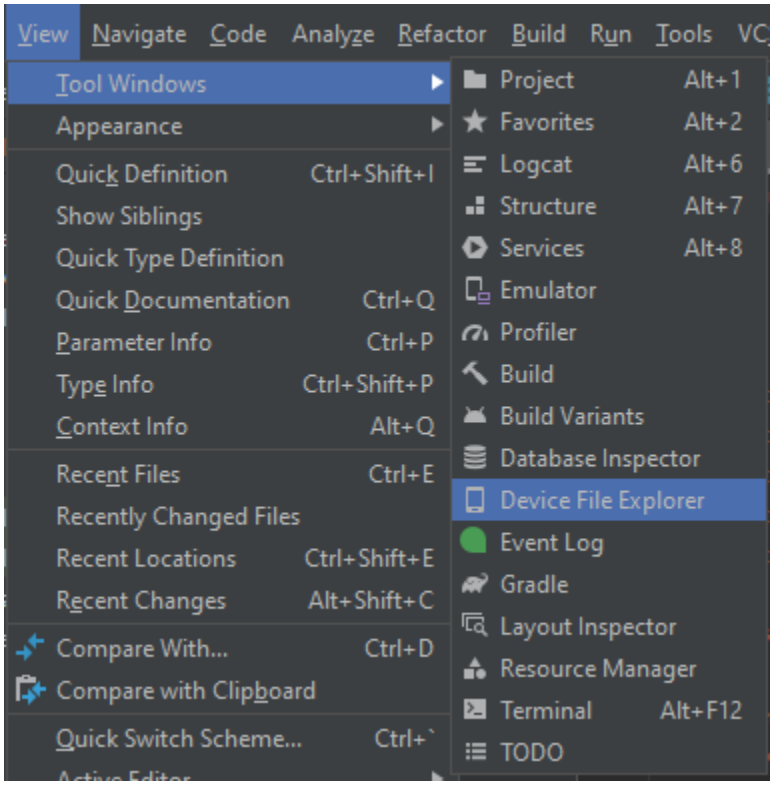
- Run the emulator to see the data added on click of add record button.
- As we enter the data in text fields and click the add button, we will get its values represented in bottom of screen that are displayed by toast.



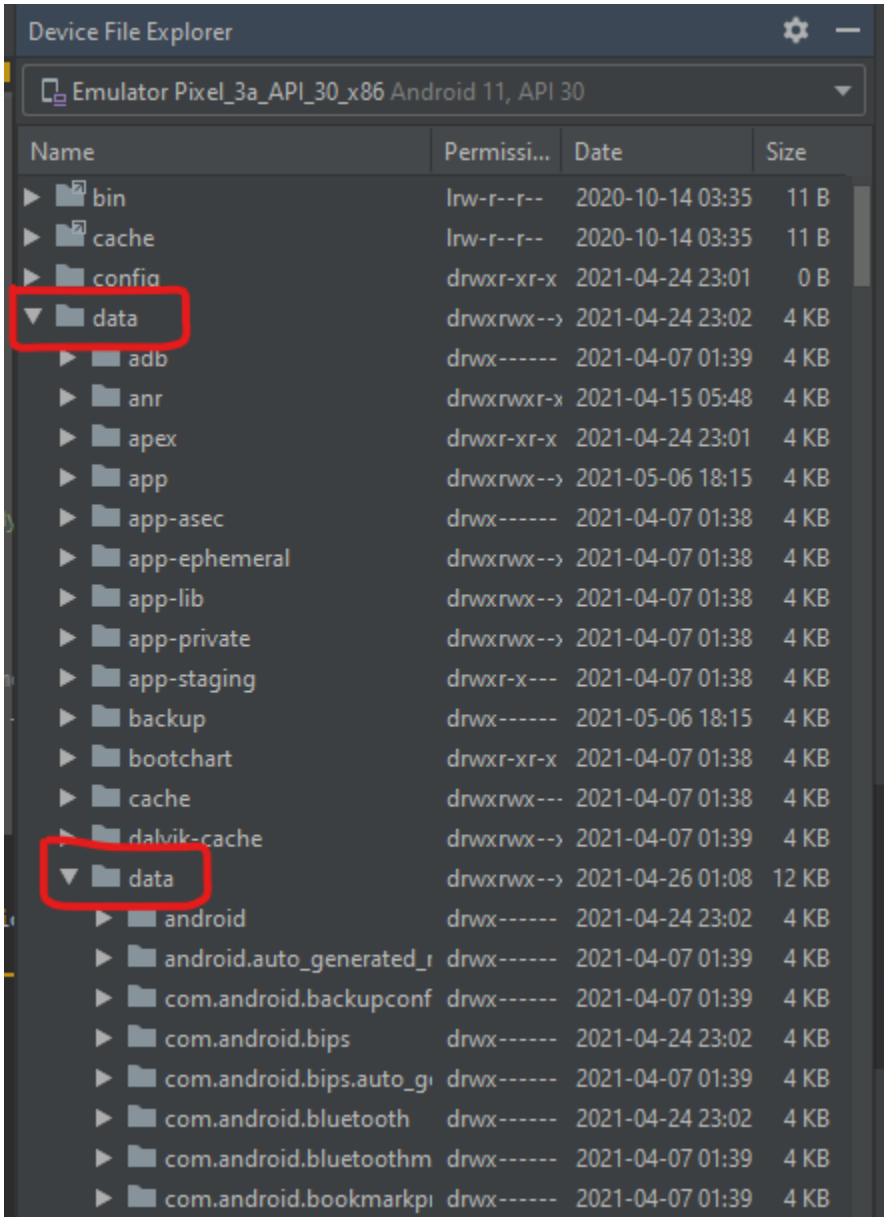
To Open Created Database:

❖ To view files of the device:

- Run the emulator first, and then follow the steps bellow.

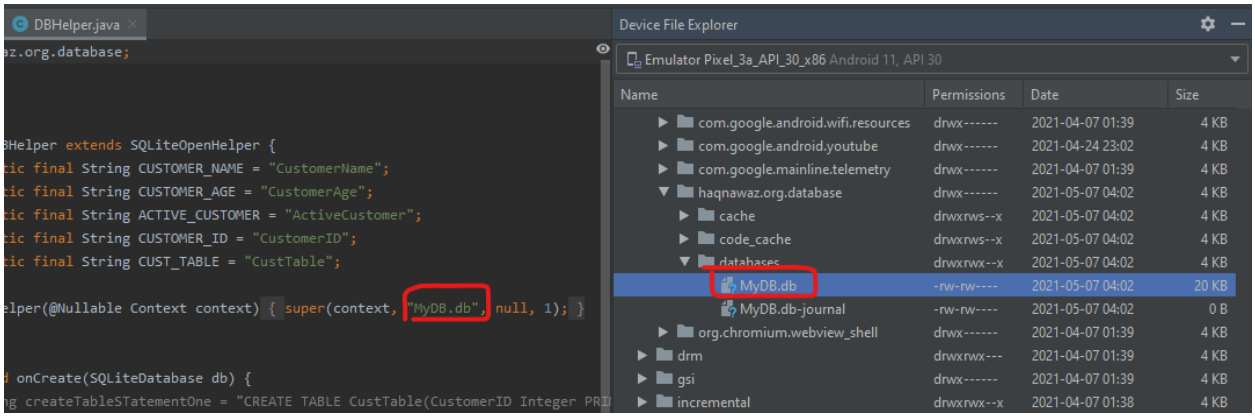


- Here in **data folder** are all applications that are available that we run on the emulator.

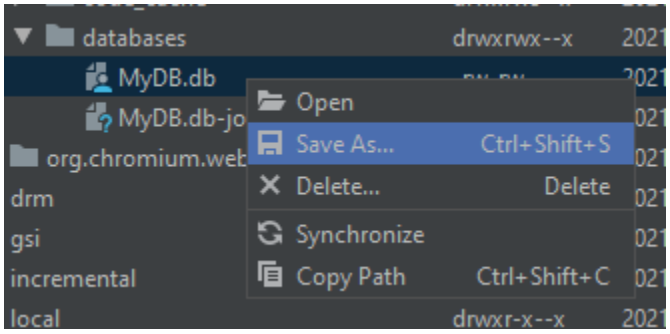


❖ To View Database file:

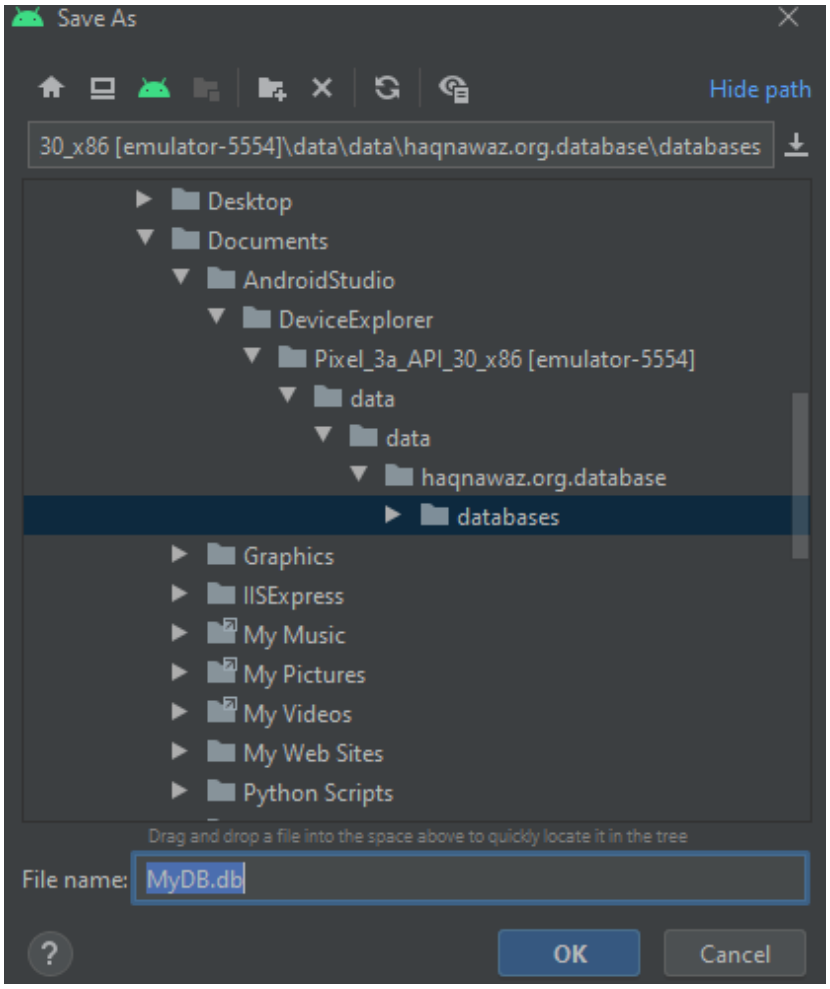
- To see the database file open the project's package and click on its database as named in the DB helper class.



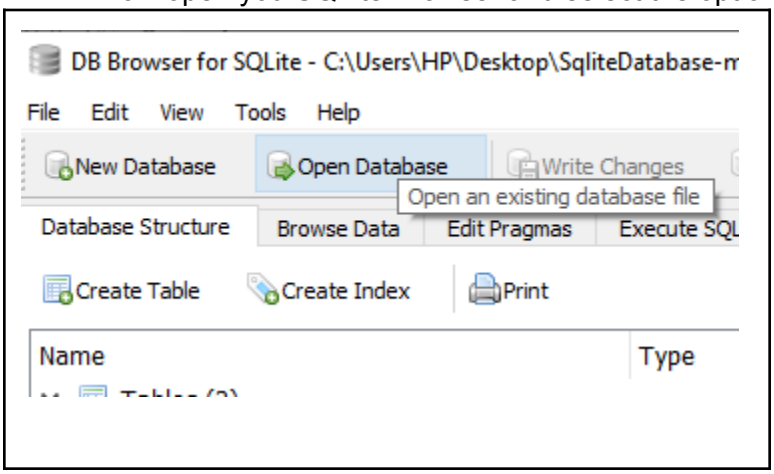
- Export the database on your system using save as.



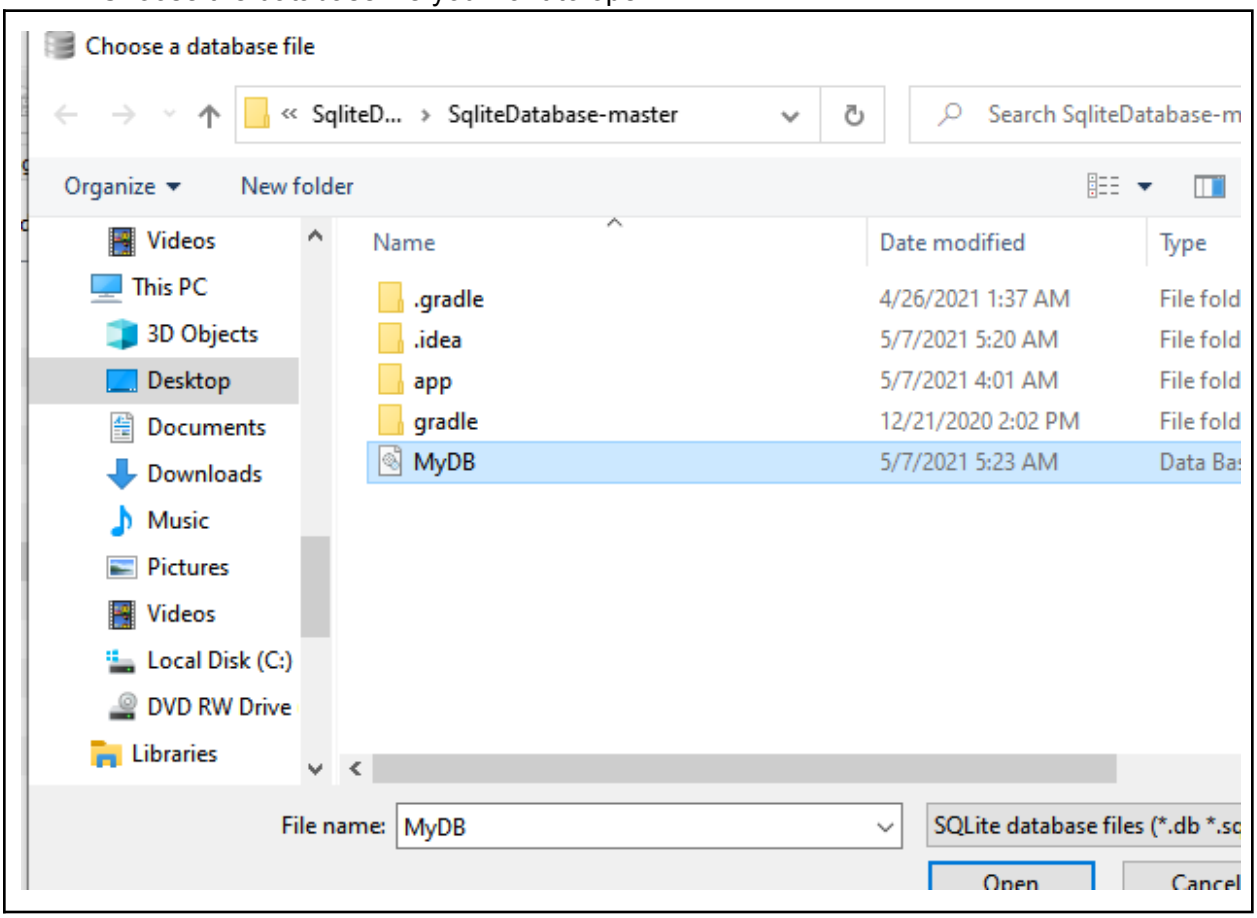
- Select a location to save your database, so you can view it later on in **SQLite Browser.**



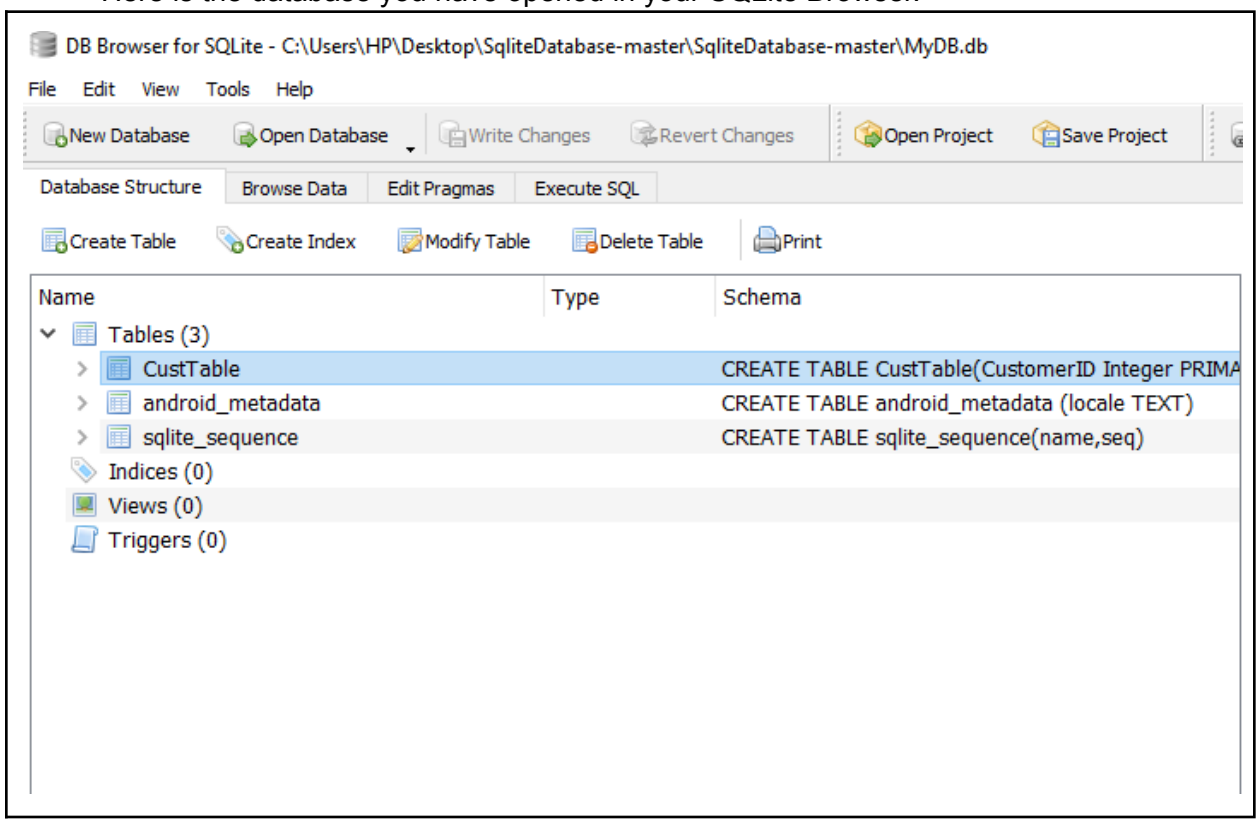
- Now open you SQLite Browser and select the option to open database.



- Choose the database file you want to open.



- Here is the database you have opened in your SQLite Browser.



LECTURE#11

SQLite Lec#3

❖ Retrieving Data from Database:

- Make a function **getAllRecords()** will return a list of type customer model.

```
}
public List<CustomerModel>getAllRecords(){

    List<CustomerModel> myList =new ArrayList<>();


    return myList;
}
```

Define Query:

- Define the following query to retrieve data from database.

```
public List<CustomerModel>getAllRecords(){

    List<CustomerModel> myList =new ArrayList<>();
    String query="SELECT *FROM"+ CUST_TABLE;


    return myList;
}
```

❖ Make an object of SQLite Database:

- To get record make an object of database.

```
}
public List<CustomerModel>getAllRecords(){

    List<CustomerModel> myList =new ArrayList<>();
    String query="SELECT *FROM"+ CUST_TABLE;
    SQLiteDatabase DB=this.getReadableDatabase();


    return myList;
```

❖ **Curser:** used to return dataSet.(kind of a table)

- Define **rawQuery()** that will return us a cursor as dataset.

```
public List<CustomerModel>getAllRecords(){

    List<CustomerModel> myList =new ArrayList<>();
    String query="SELECT *FROM"+ CUST_TABLE;
    SQLiteDatabase DB=this.getReadableDatabase();
    Cursor cursor=DB.rawQuery(query, selectionArgs: null);

    return myList;
}
```

❖ Checks and place records:

- Place a check in if condition to check if records are available, then get them.

```
    if(cursor.moveToFirst()){
        do{
            String custName=cursor.getString( columnIndex: 1);
            int custAge=cursor.getInt( columnIndex: 2);
            Boolean isActive=cursor.getInt( columnIndex: 3)==1?true:false;

        }while(cursor.moveToNext());
    }
```

❖ Using Model:

- Now define a customer class model to get its values to database.

```
if(cursor.moveToFirst()){
    do{
        String custName=cursor.getString( columnIndex: 1);
        int custAge=cursor.getInt( columnIndex: 2);
        Boolean isActive=cursor.getInt( columnIndex: 3)==1?true:false;

        CustomerModel newCustomerModel=new CustomerModel(custName,custAge,isActive, id: 1);
    }while(cursor.moveToNext());
}
```

❖ Adding to list:

- Add values of model to list in order to display them.

```
if(cursor.moveToFirst()){
    do{
        String custName=cursor.getString( columnIndex: 1);
        int custAge=cursor.getInt( columnIndex: 2);
        Boolean isActive=cursor.getInt( columnIndex: 3)==1?true:false

        CustomerModel newCustomerModel=new CustomerModel(custName,cu
        myList.add(newCustomerModel);
    }while(cursor.moveToNext());
}
```

❖ Closing Resources after use:

- Close the used resources after getting values from them.

```
if(cursor.moveToFirst()){
    do{
        String custName=curs
        int custAge=cursor.ge
        Boolean isActive=curs

        CustomerModel newCust
        myList.add(newCustome
    }while(cursor.moveToNext
}
cursor.close();
DB.close();
return myList;
```

❖ Add DB Helper & List:

- OnClick() Add DBHelper and list so we can get values from databse on click.

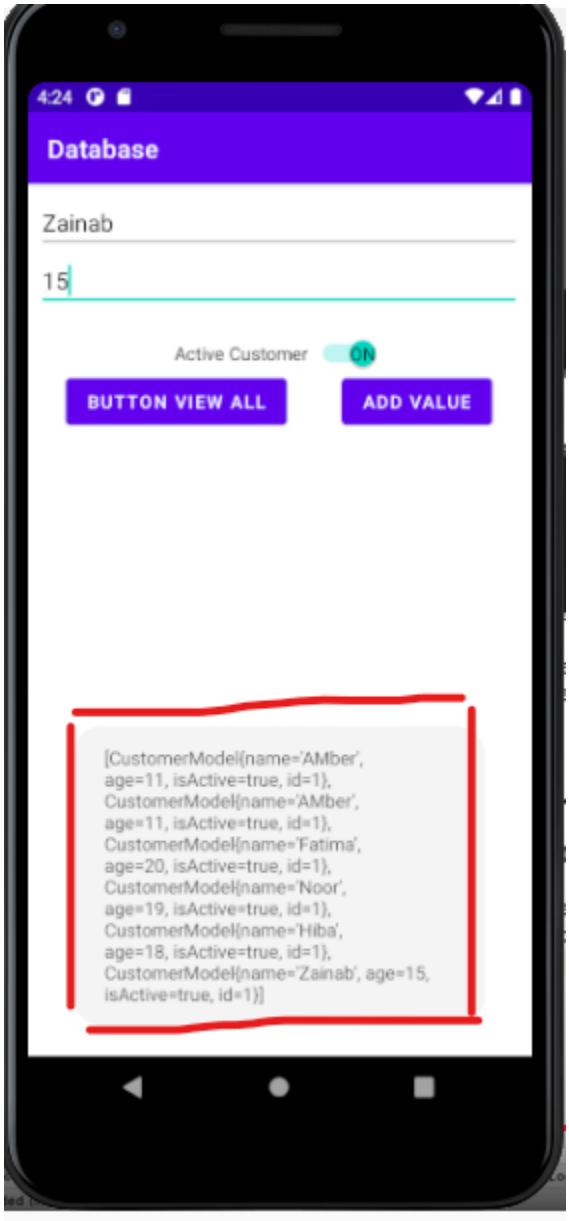
```
buttonViewAll.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DBHelper dbHelper=new DBHelper( context: MainActivity.this);
        List<CustomerModel> allCustomers=dbHelper.getAllRecords();
```

❖ Displaying in toast:

- Use a toast to check if values are being added on click of add button.

```
    public void onClick(View v) {
        DBHelper dbHelper=new DBHelper( context: MainActivity.this);
        List<CustomerModel> allCustomers=dbHelper.getAllRecords();

        Toast.makeText( context: MainActivity.this, allCustomers.toString(), Toast.LENGTH_SHORT).show();
    }
});
```

- Click on add value then view record



❖ To Display in ListView:

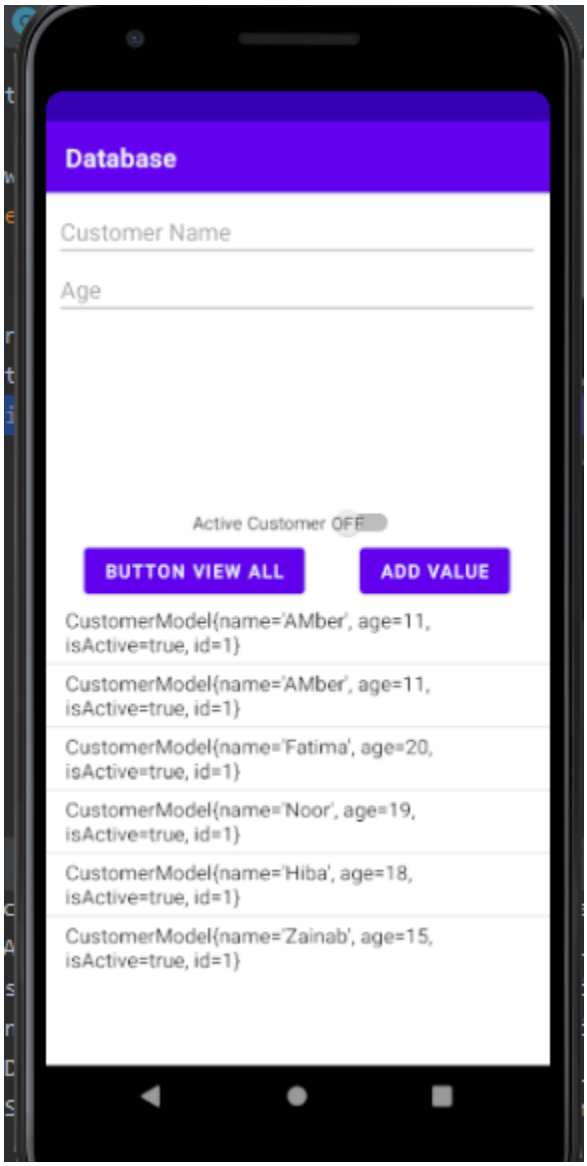
- List is made so make a new array adapter to show it.

```
ListView listViewCustomer;
ArrayAdapter<CustomerModel> arrayAdapter;
@Override
```

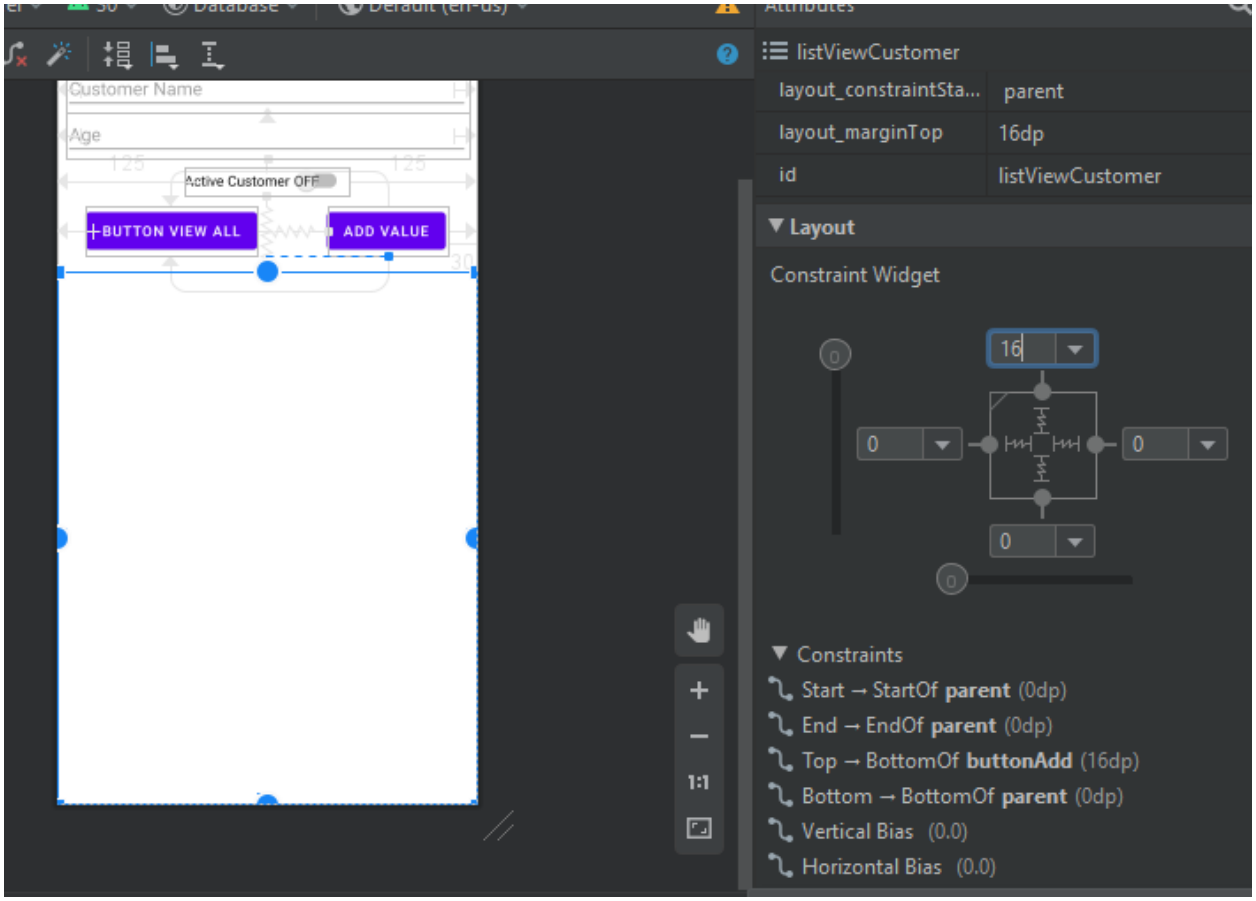
- Pass customer class values to array adapter then pass adapter to list to get displayed.

```
arrayAdapter=new ArrayAdapter<CustomerModel>( context: MainActivity.this, android.R.layout.simple_list_item
listViewCustomer.setAdapter(arrayAdapter);
```

- Now as we run the emulator we can see values being displayed in list on clicking add value.

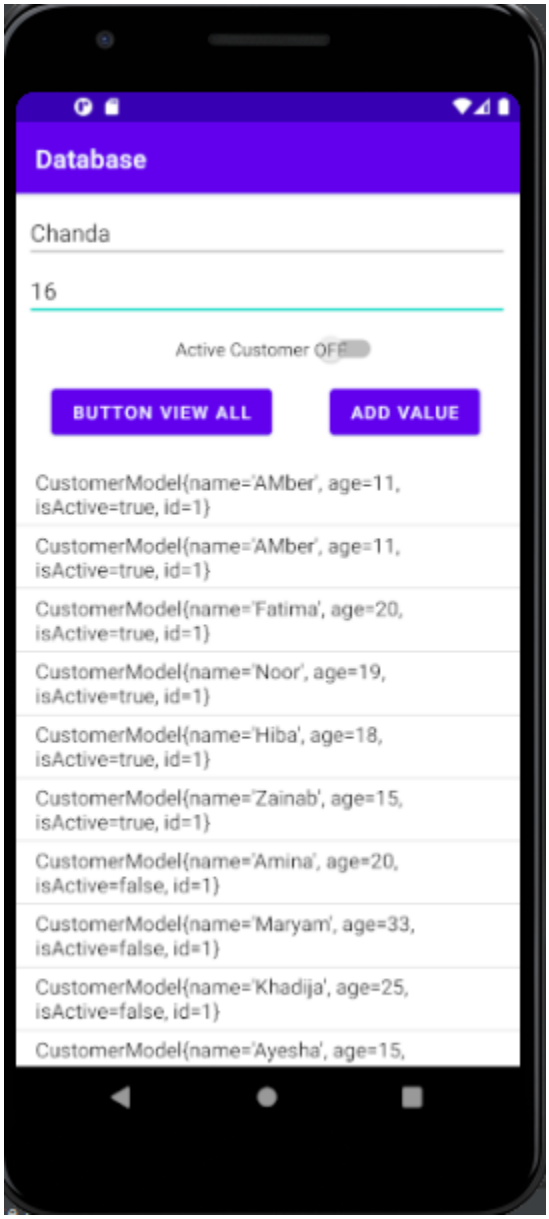


- List is not displayed in scroll view properly so reset its constraints.



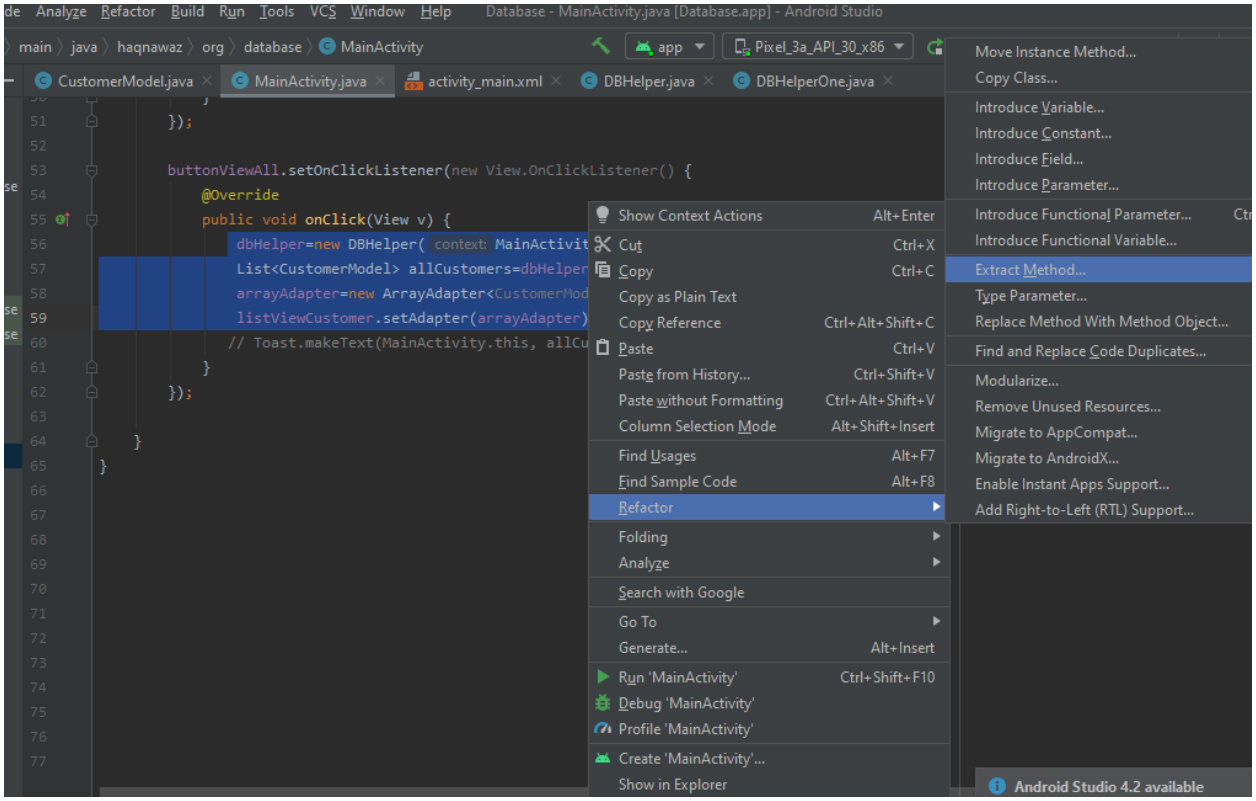
.

- After setting constraints rerun the emulator and see a scrolling view of the records being displayed in list.

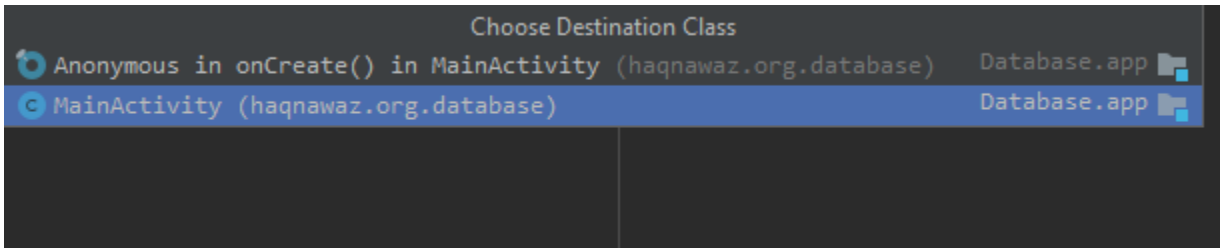


❖ Refactor Method:

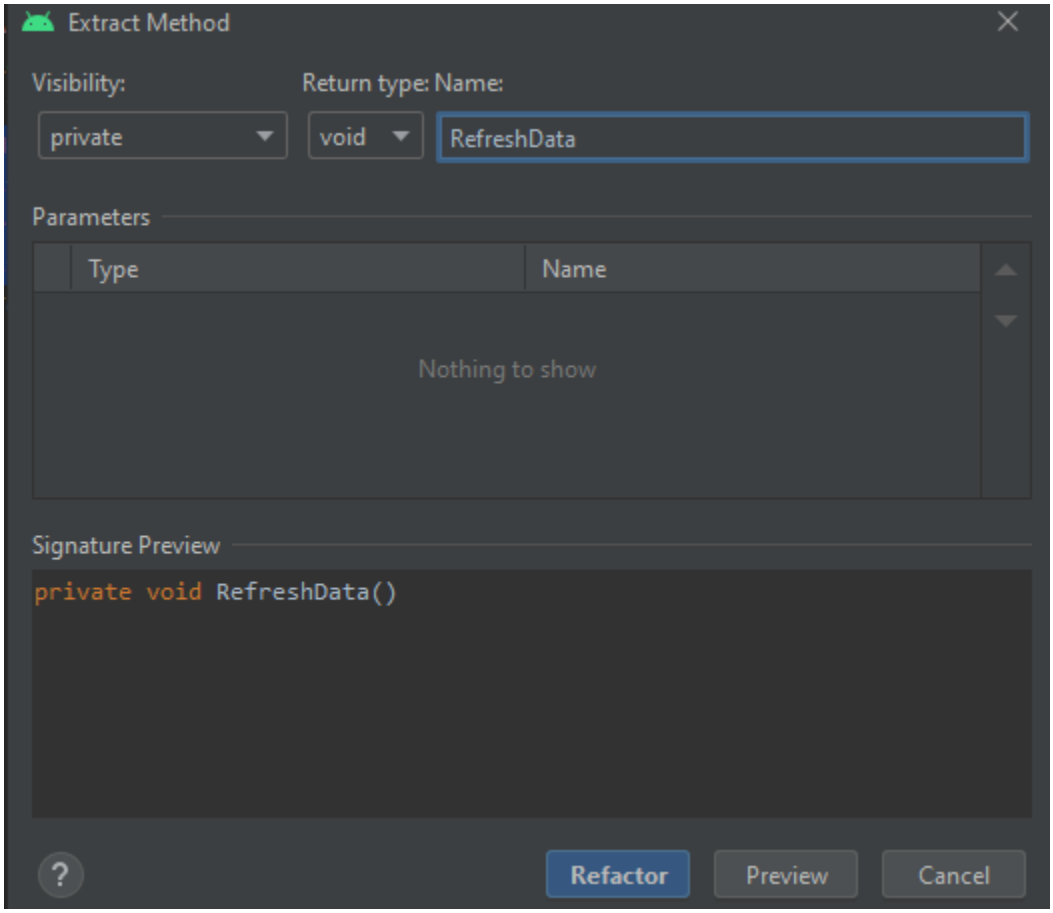
- To refactor method follow the steps bellow.



- Choose the activity where you want to refactor.



- Select a return type name for the method to be refactored.

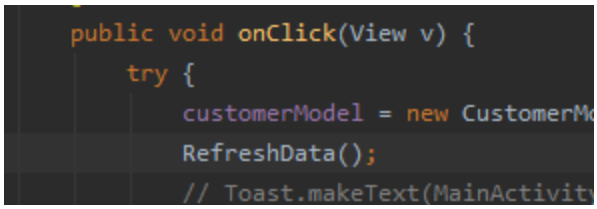


- Now instead of using all the statements just call the refactored method in click function.

```
buttonViewAll.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        RefreshData();
        // Toast.makeText(MainActivity.this, allCustomers.toString(), Toast.LENGTH_LONG)
    }
});



ivate void RefreshData() {
dbHelper=new DBHelper( context: MainActivity.this);
List<CustomerModel> allCustomers=dbHelper.getAllRecords();
arrayAdapter=new ArrayAdapter<CustomerModel>( context: MainActivity.this, android.R.layout
listViewCustomer.setAdapter(arrayAdapter);
```

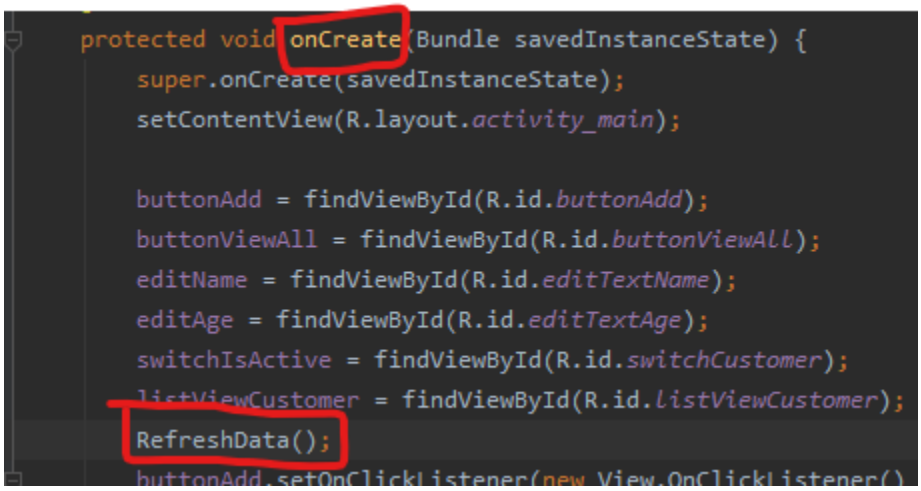
❖ Refresh on Add new record:

- Also add refactor method to on click try so record being added to list when clicked.

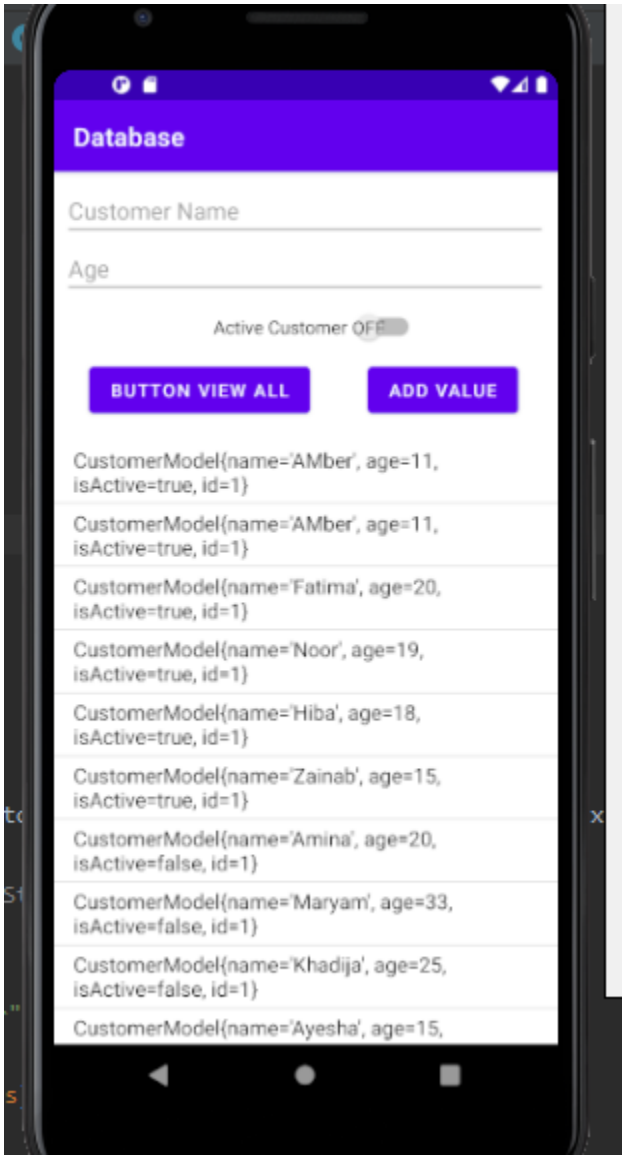


❖ Refresh on OnCreate:

- Also add refactor method to on create so record being displayed not just on adding but also on first view of screen too.

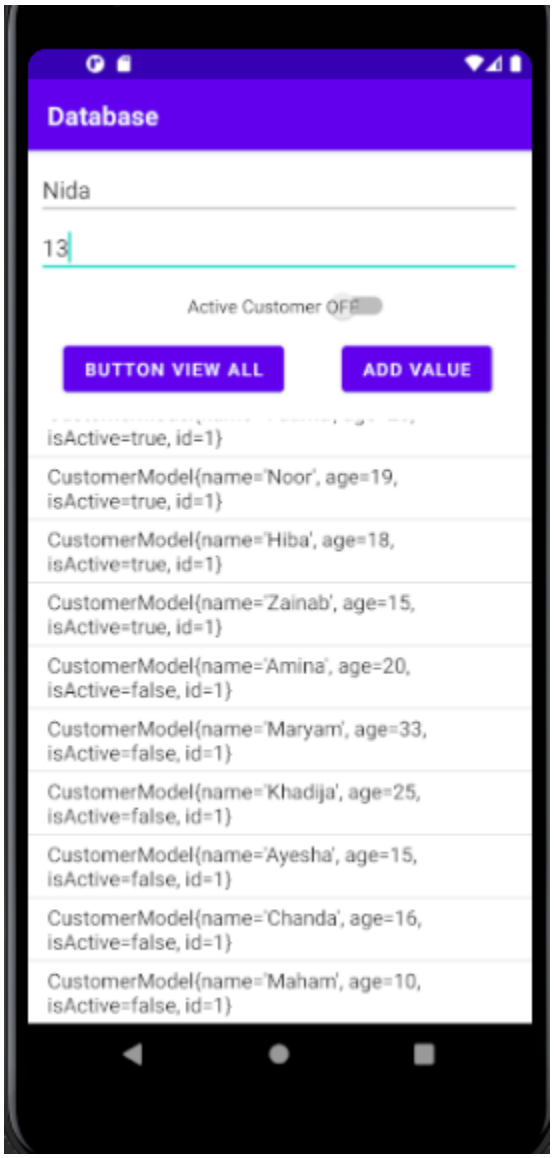


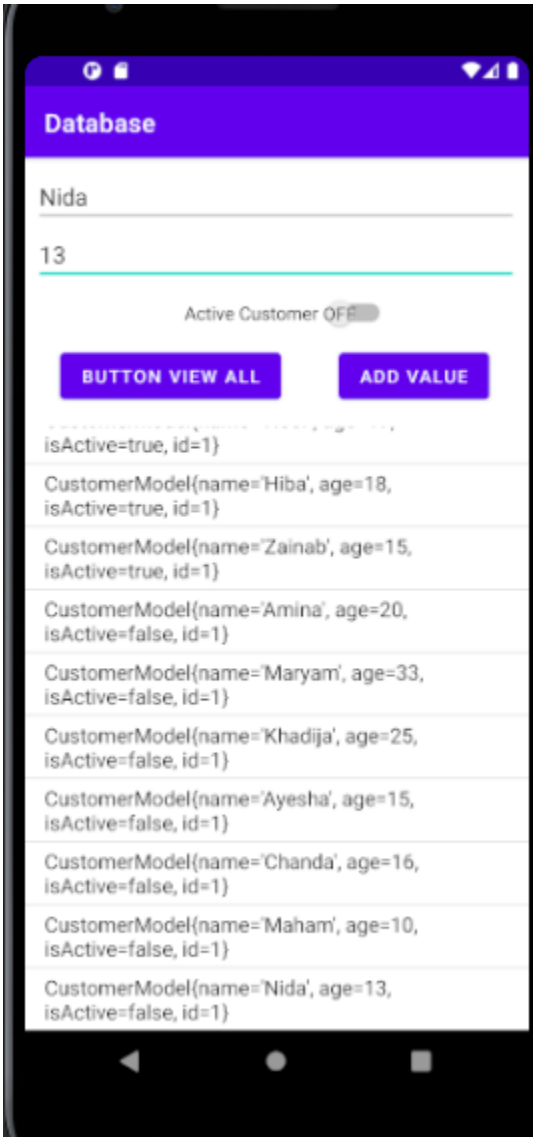
- Run the emulator and see the previous records on screen without even adding a new one.



- **Bug:**
    - New Record added is displayed in list not on clicking add value but after clicking view all records.

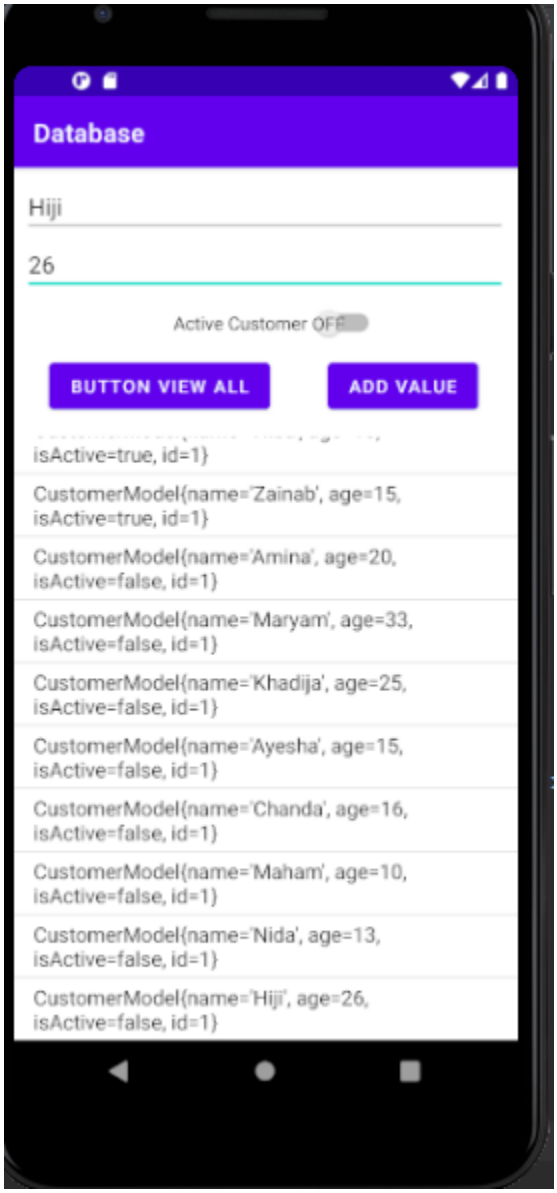
Add value=> no record show **…….. View all =>** record added show





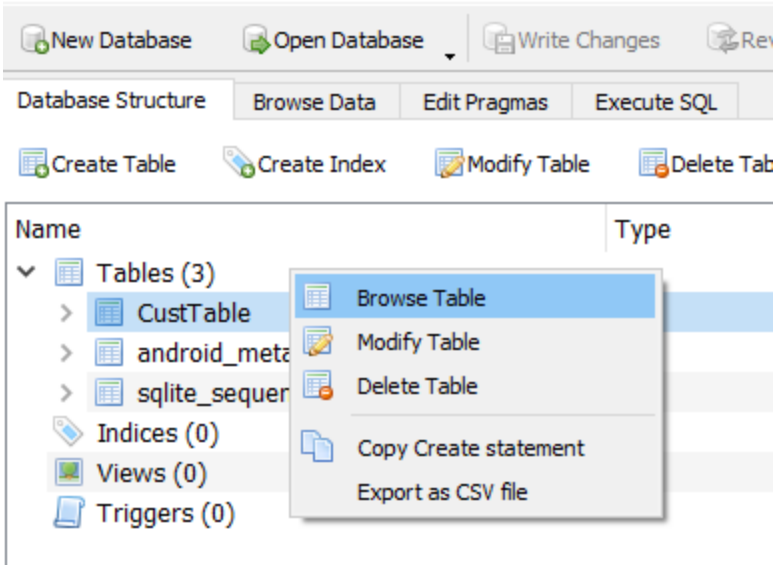
- **Remove Bug:**
    - Add refresh method just after the values are being added to list.

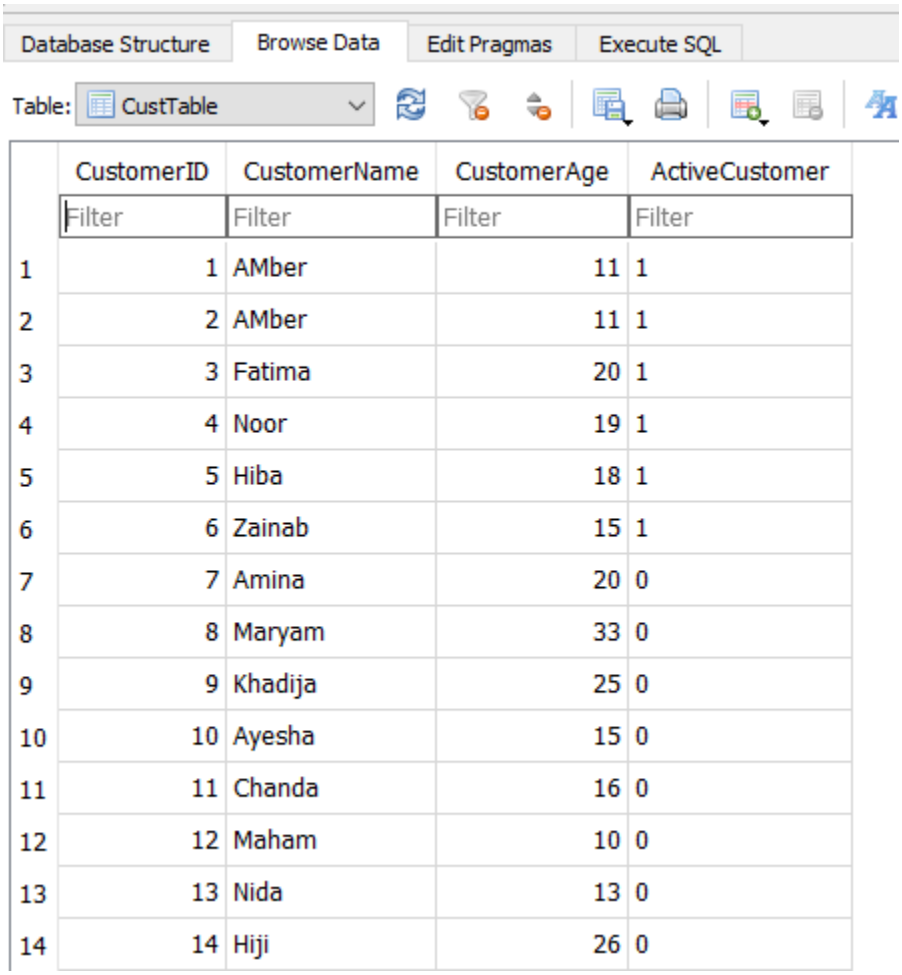
- Run th emulator and see ,Record displayed on clicking add value.



❖ To make changes in database:

- Open SQLite Browser, open your database and do the step bellow.

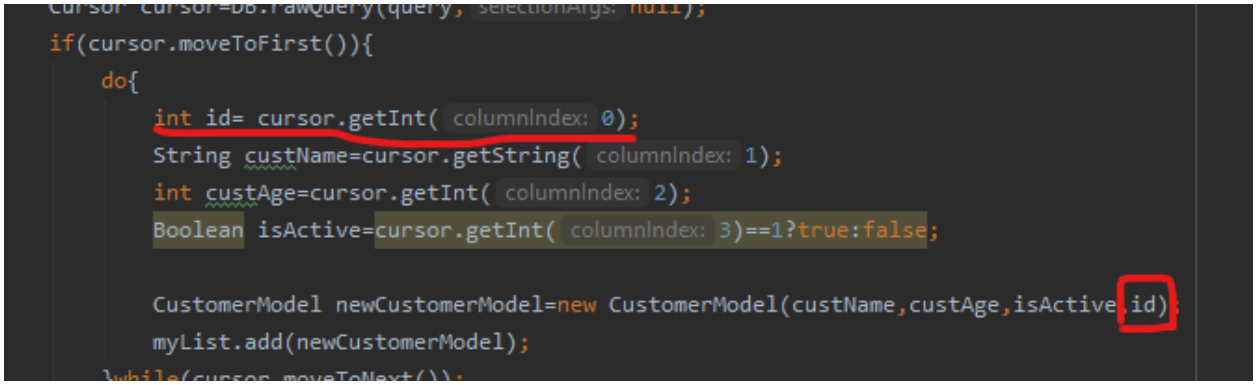


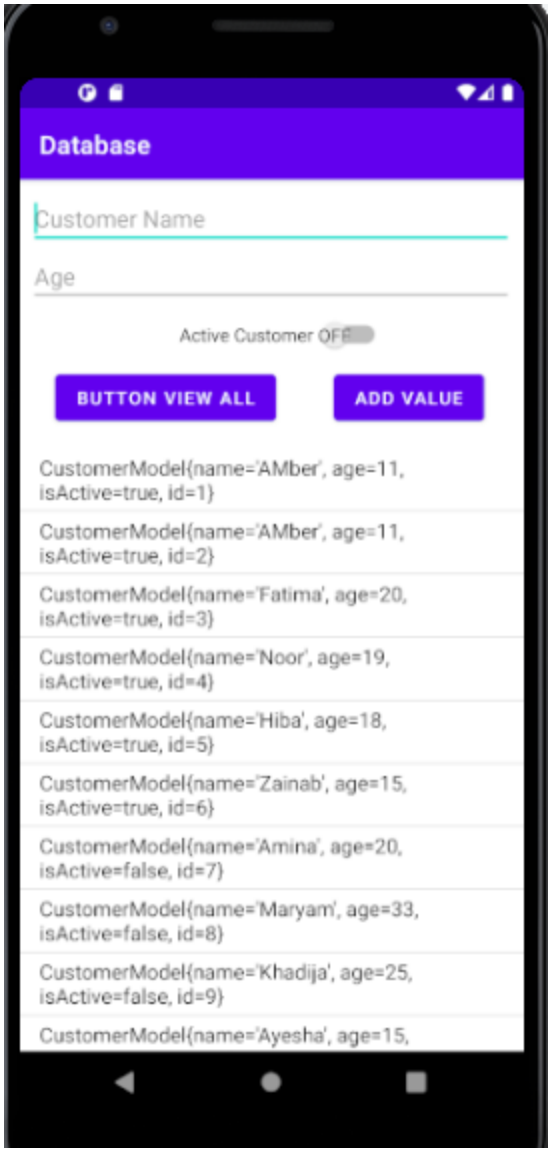


	CustomerID	CustomerName	CustomerAge	ActiveCustomer
	Filter	Filter	Filter	Filter
1	1	AMber	11	1
2	2	AMber	11	1
3	3	Fatima	20	1
4	4	Noor	19	1
5	5	Hiba	18	1
6	6	Zainab	15	1
7	7	Amina	20	0
8	8	Maryam	33	0
9	9	Khadija	25	0
10	10	Ayesha	15	0
11	11	Chanda	16	0
12	12	Maham	10	0
13	13	Nida	13	0
14	14	Hiji	26	0

Homework TASKS:

1. BUG=>Update id at data retrieval
2. On click delete record



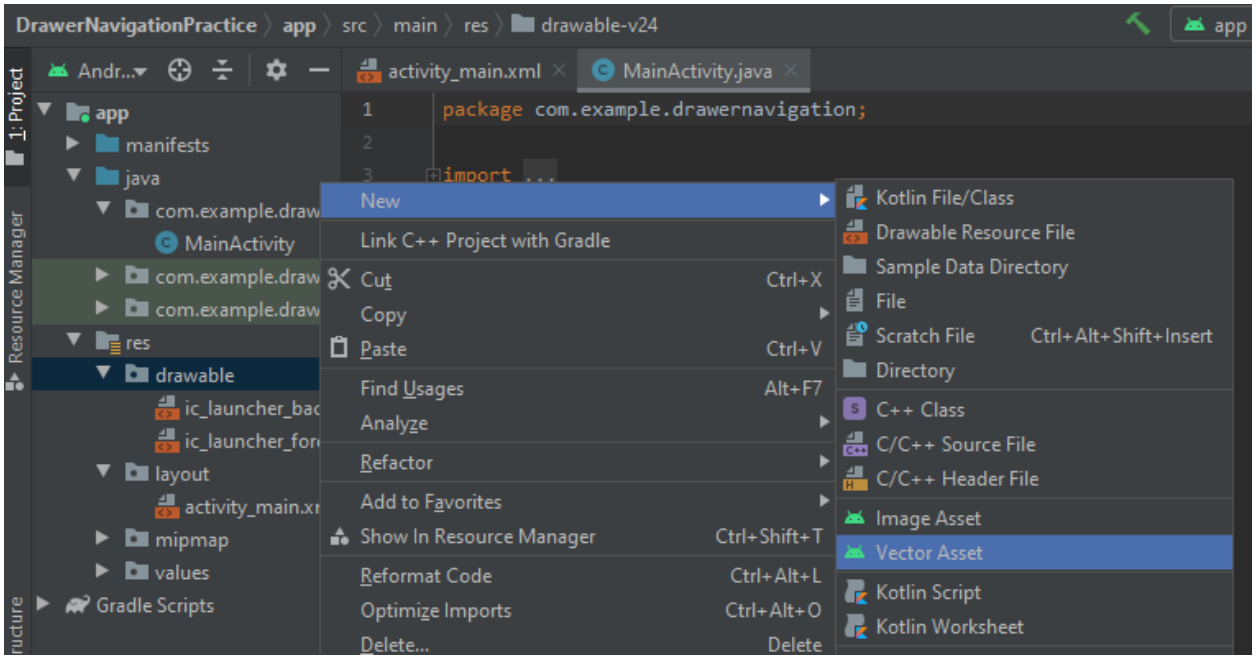


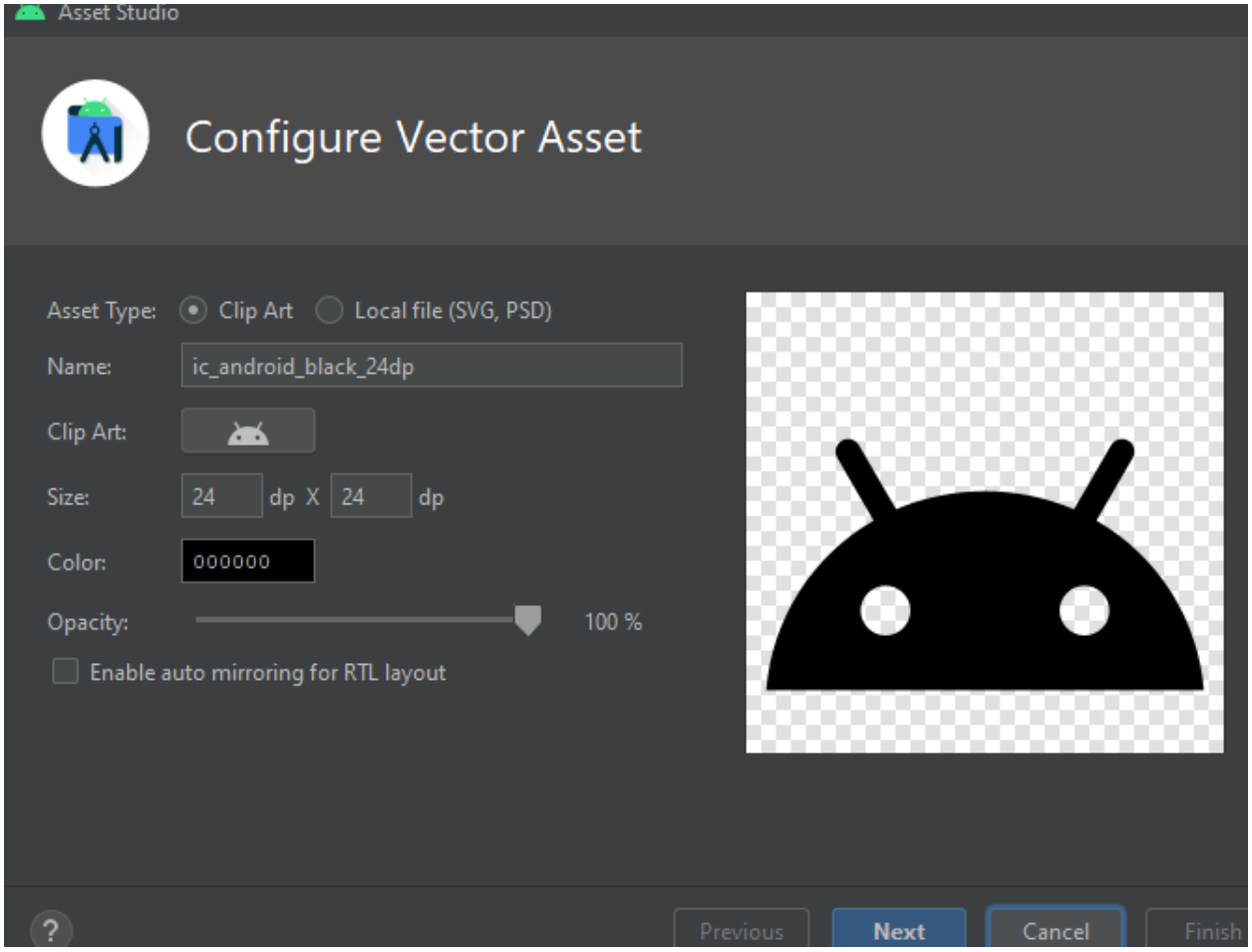
On click delete record

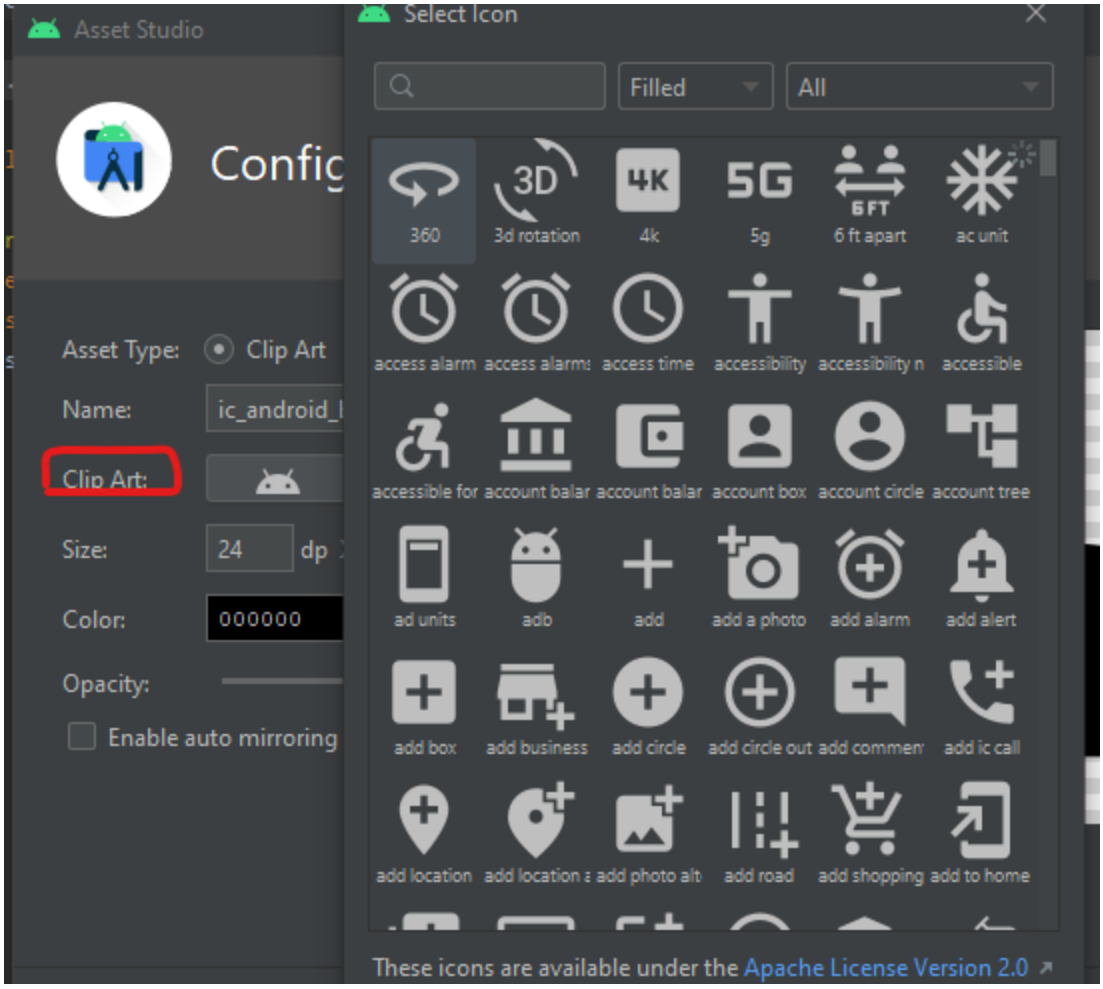
LECTURE#12

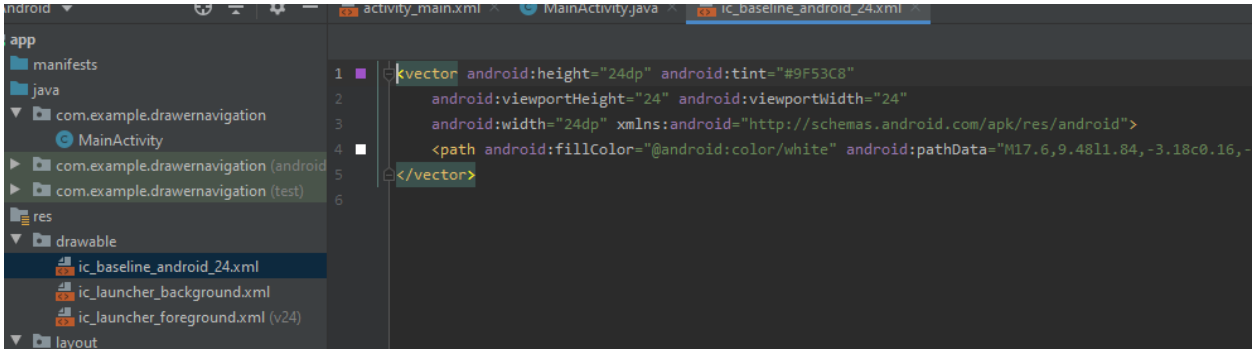
❖ Drawer Navigation:

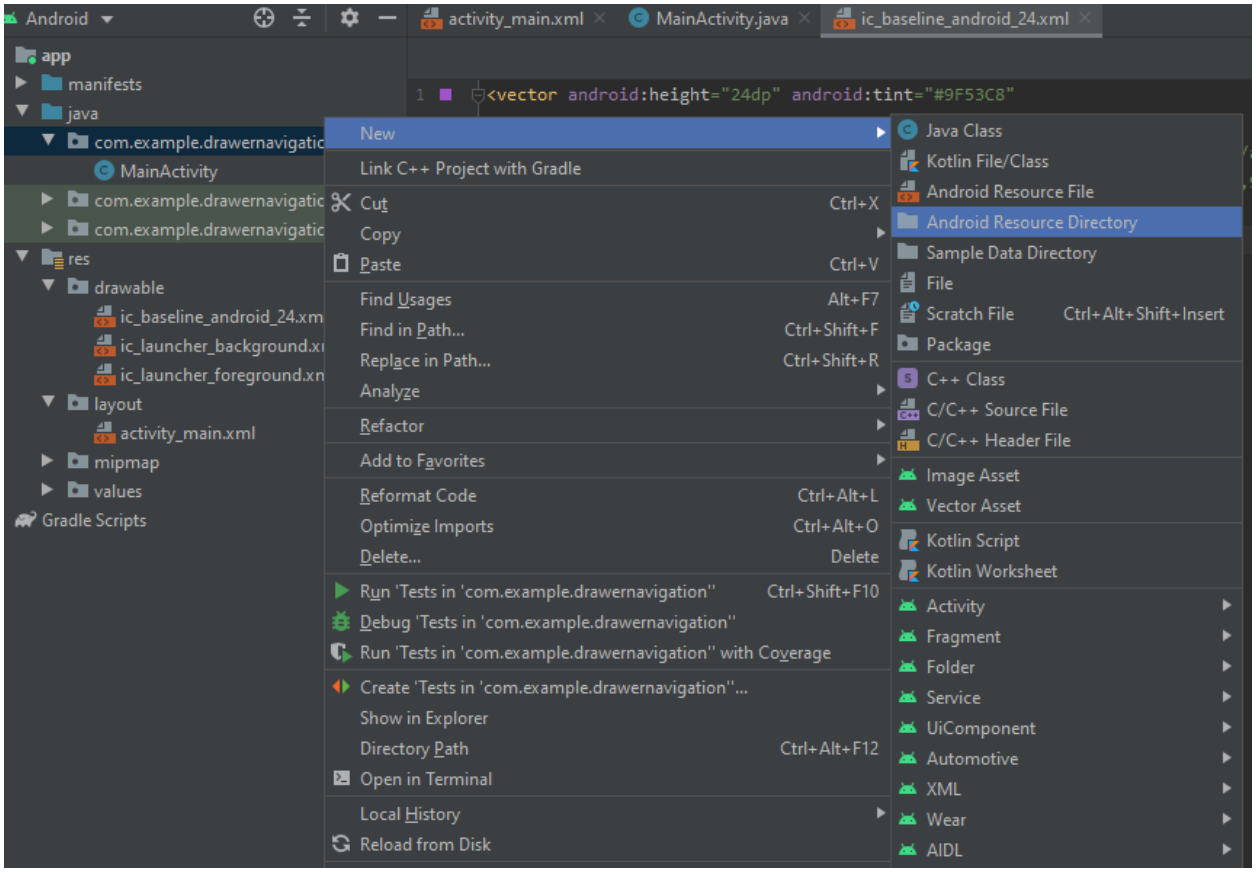
To add an icon=>Vector Asset:

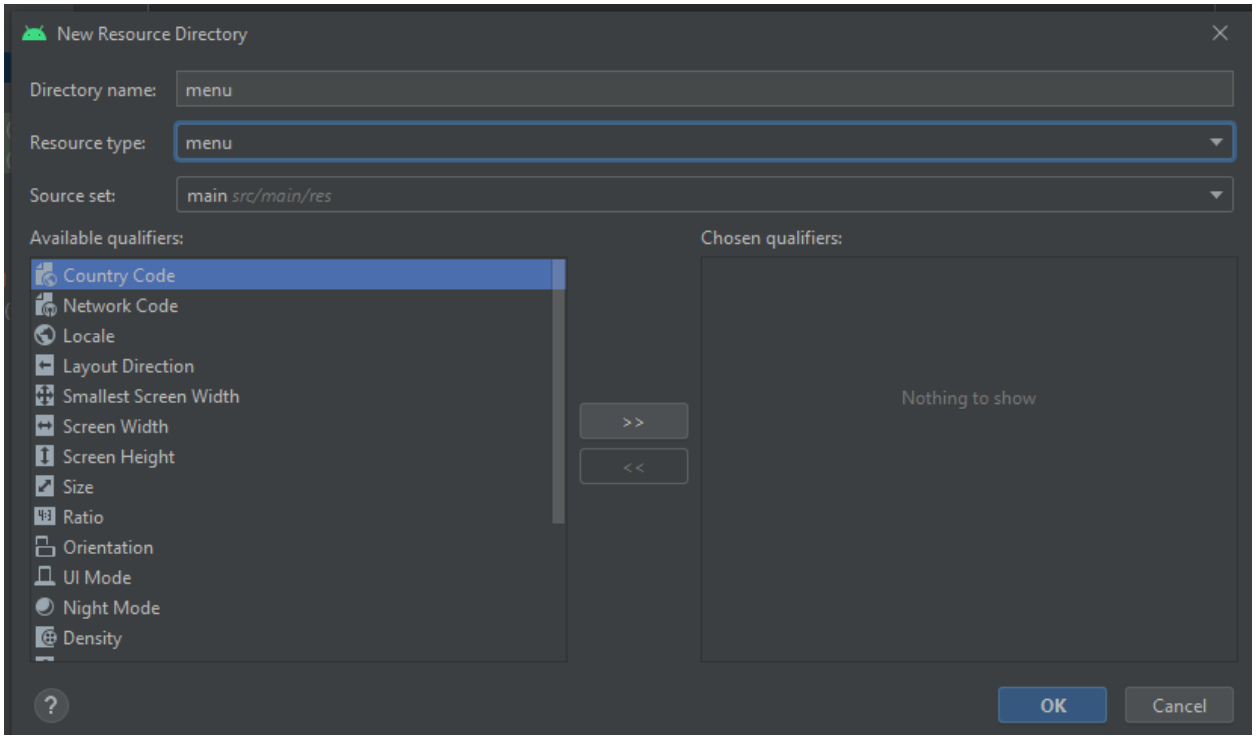


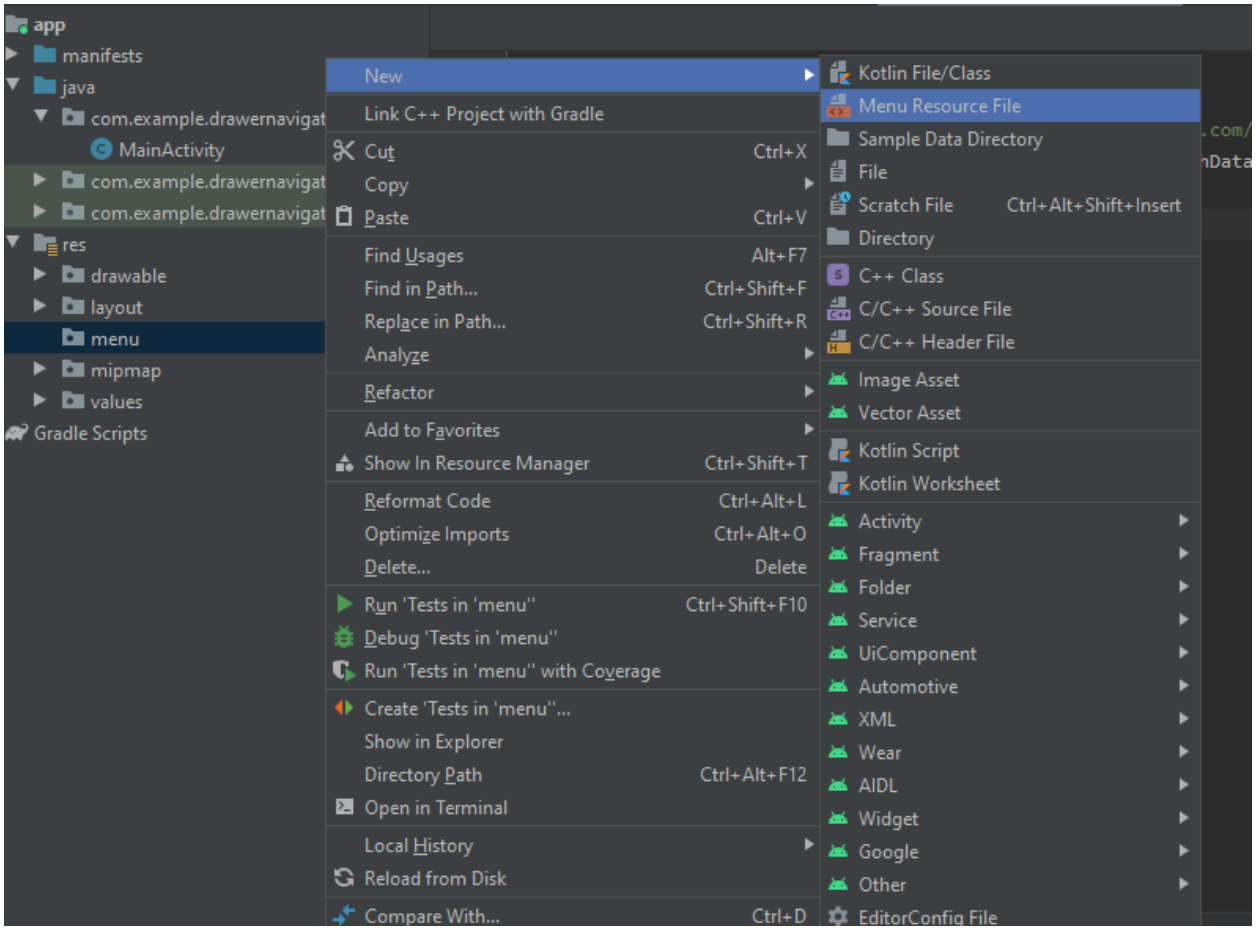

Asset Studio
Configure Vector Asset
Asset Type:
Clip Art
Local file (SVG, PSD)
Name:
ic_android_black_24dp
Clip Art:
Size:
24
dp X
24
dp
Color:
000000
Opacity:
100 %
Enable auto mirroring for RTL layout
Previous
Next
Cancel
Finish

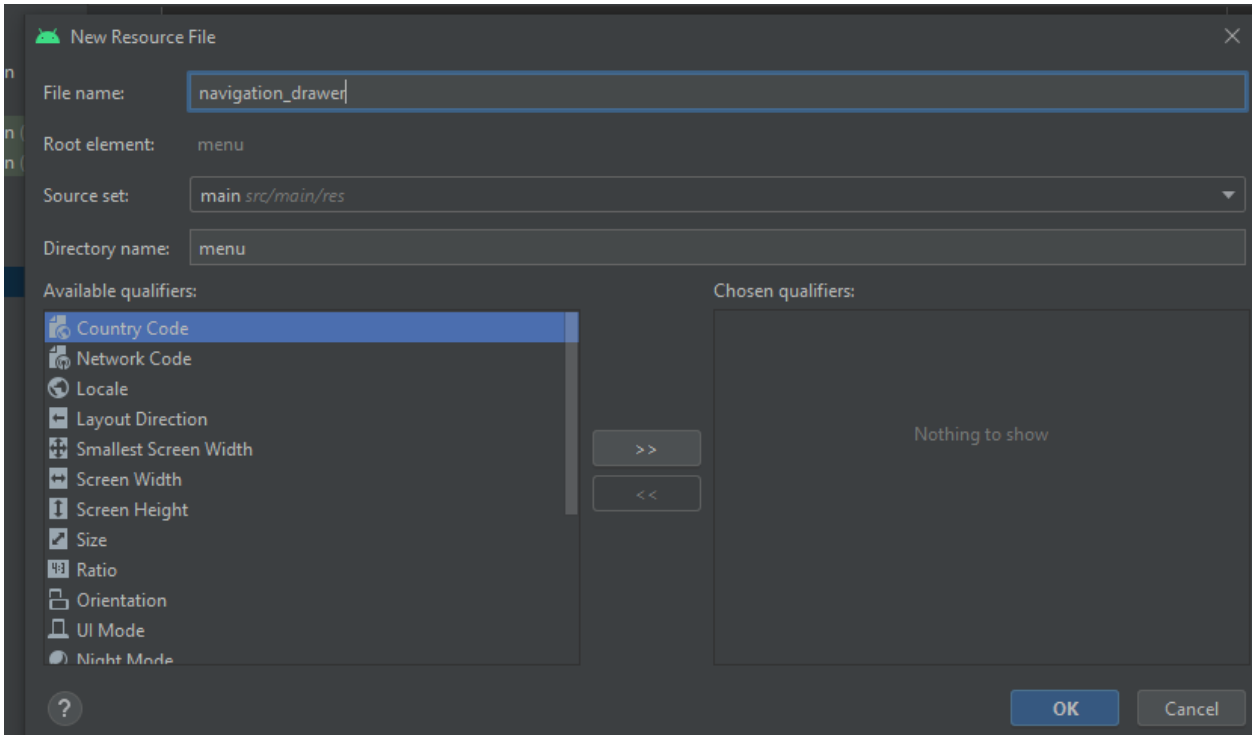

Asset Studio
Config
Asset Type:
Clip Art
Name:
ic_android_l
Clip Art:
Size:
24
dp
Color:
000000
Opacity:
Enable auto mirroring
Select Icon
Filled
All
360
3d rotation
4k
5g
6 ft apart
ac unit
access alarm
access alarms
access time
accessibility
accessibility n
accessible
accessible for
account balar
account balar
account box
account circle
account tree
ad units
adb
add
add a photo
add alarm
add alert
add box
add business
add circle
add circle out
add commen
add ic call
add location
add location a
add photo alt
add road
add shopping
add to home
These icons are available under the Apache License Version 2.0

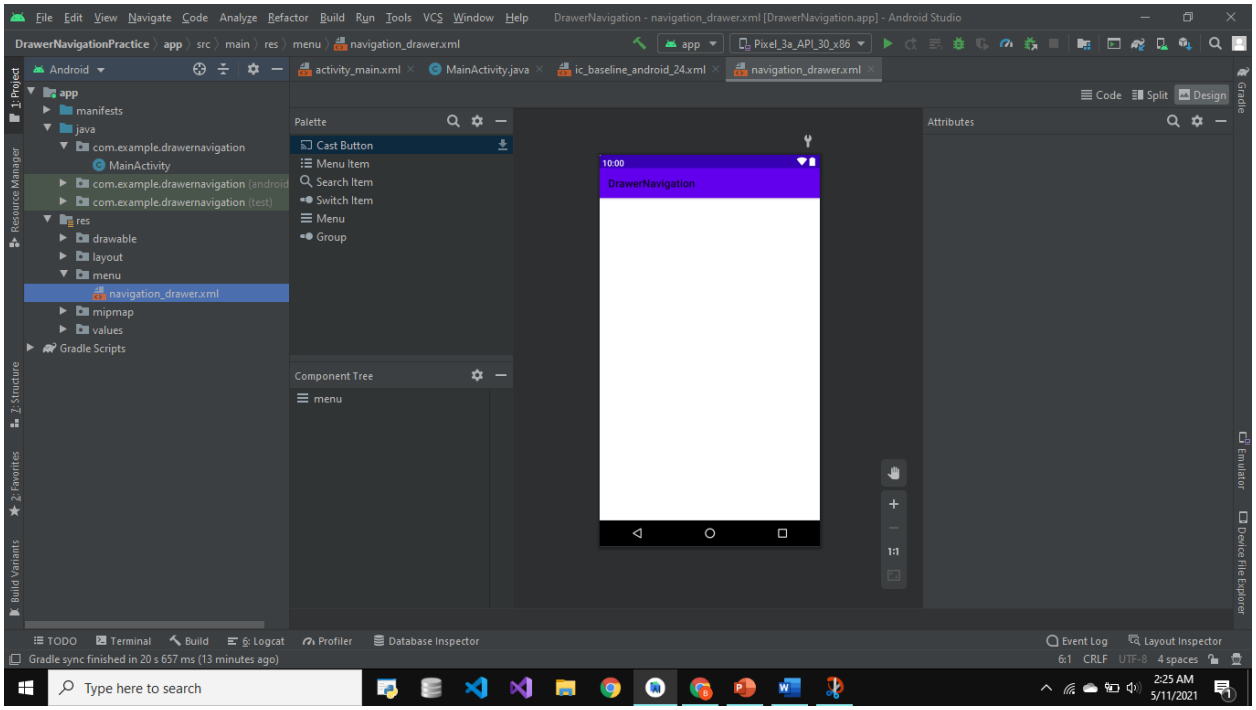

app
manifests
java
com.example.drawernavigation
MainActivity
com.example.drawernavigation (android
com.example.drawernavigation (test)
res
drawable
ic_baseline_android_24.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
layout
activity_main.xml
MainActivity.java
ic_baseline_android_24.xml
<vector android:height="24dp" android:tint="#9F53C8"
android:viewportHeight="24" android:viewportWidth="24"
android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">
<path android:fillColor="@android:color/white" android:pathData="M17.6,9.48l1.84,-3.18c0.16,-
</vector>

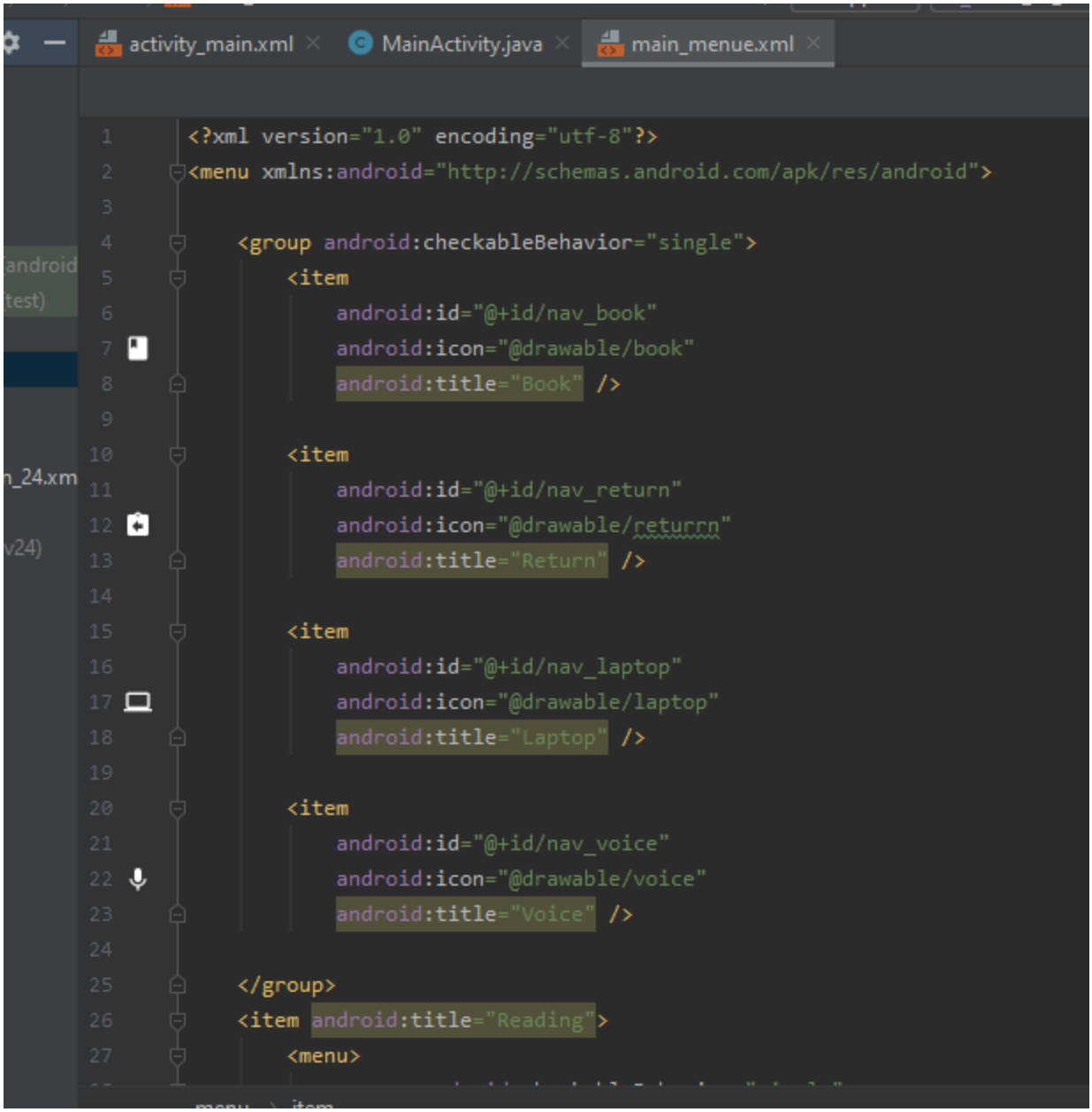

Android
activity_main.xml
MainActivity.java
ic_baseline_android_24.xml
app
manifests
java
com.example.drawernavigatic
MainActivity
com.example.drawernavigatic
com.example.drawernavigatic
res
drawable
ic_baseline_android_24.xm
ic_launcher_background.x
ic_launcher_foreground.xn
layout
activity_main.xml
mipmap
values
Gradle Scripts
1 <vector android:height="24dp" android:tint="#9F53C8"
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Run 'Tests in 'com.example.drawernavigation'' Ctrl+Shift+F10
Debug 'Tests in 'com.example.drawernavigation''
Run 'Tests in 'com.example.drawernavigation'' with Coverage
Create 'Tests in 'com.example.drawernavigation''...
Show in Explorer
Directory Path Ctrl+Alt+F12
Open in Terminal
Local History
Reload from Disk
Java Class
Kotlin File/Class
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL

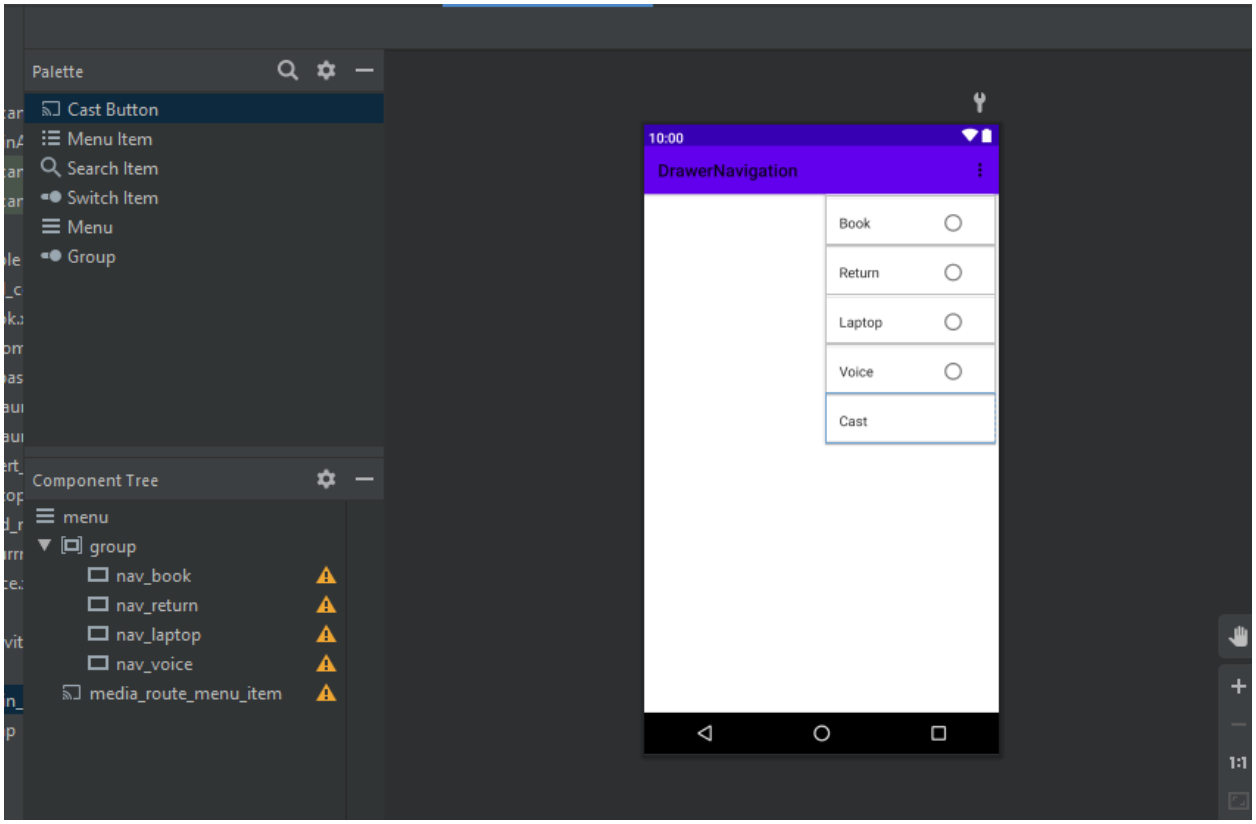

New Resource Directory
Directory name: menu
Resource type: menu
Source set: main src/main/res
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Density
>>
<<
Chosen qualifiers:
Nothing to show
OK
Cancel

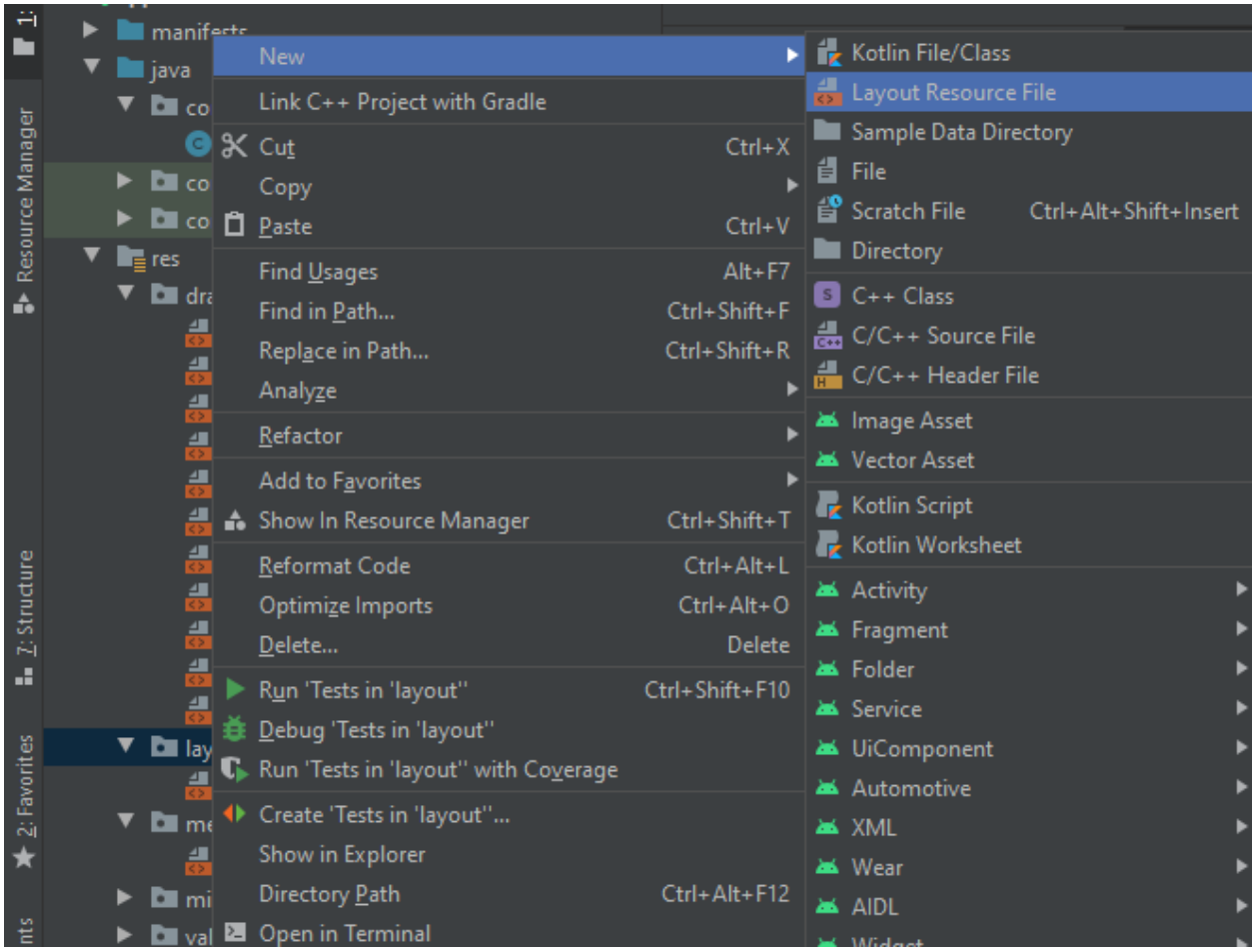

app
manifests
java
com.example.drawernavigat
MainActivity
com.example.drawernavigat
com.example.drawernavigat
res
drawable
layout
menu
mipmap
values
Gradle Scripts
New
Link C++ Project with Gradle
Cut Ctrl+X
Copy
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show In Resource Manager Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Run 'Tests in 'menu'' Ctrl+Shift+F10
Debug 'Tests in 'menu''
Run 'Tests in 'menu'' with Coverage
Create 'Tests in 'menu''...
Show in Explorer
Directory Path Ctrl+Alt+F12
Open in Terminal
Local History
Reload from Disk
Compare With... Ctrl+D
Kotlin File/Class
Menu Resource File
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Other
EditorConfig File

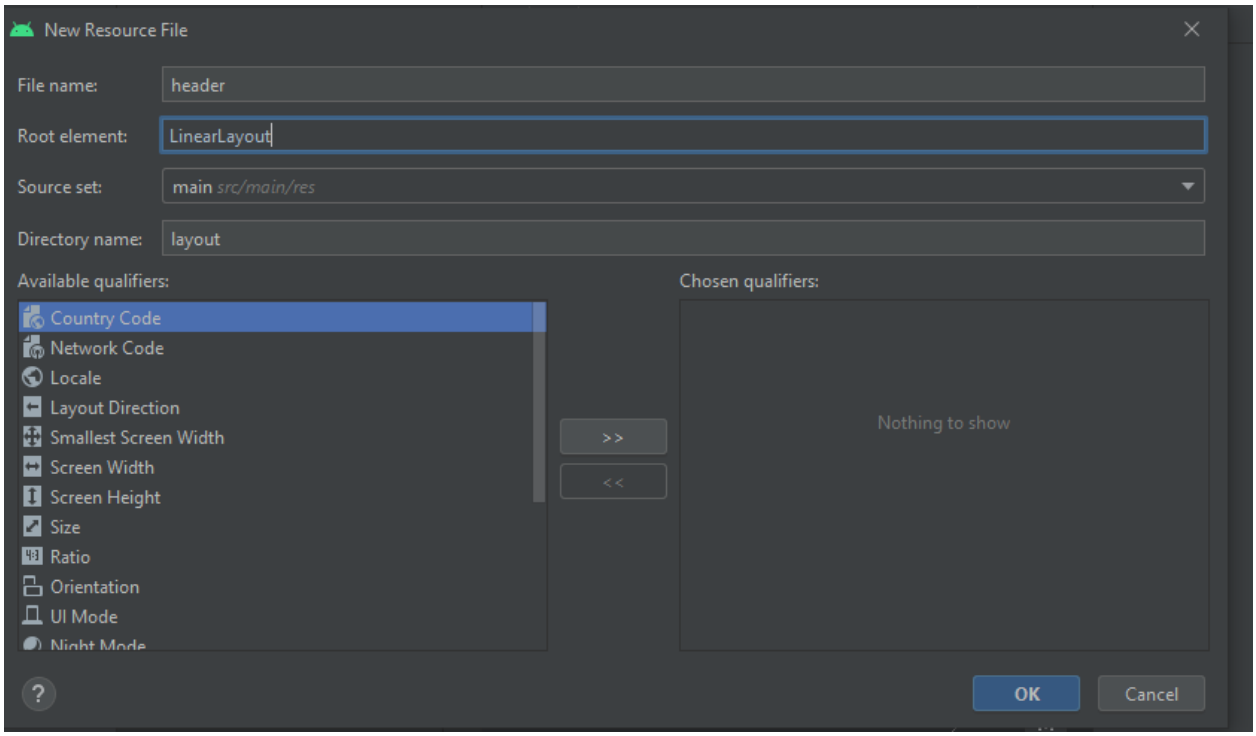

New Resource File
File name: navigation_drawer
Root element: menu
Source set: main src/main/res
Directory name: menu
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
>>
<<
Chosen qualifiers:
Nothing to show
OK
Cancel

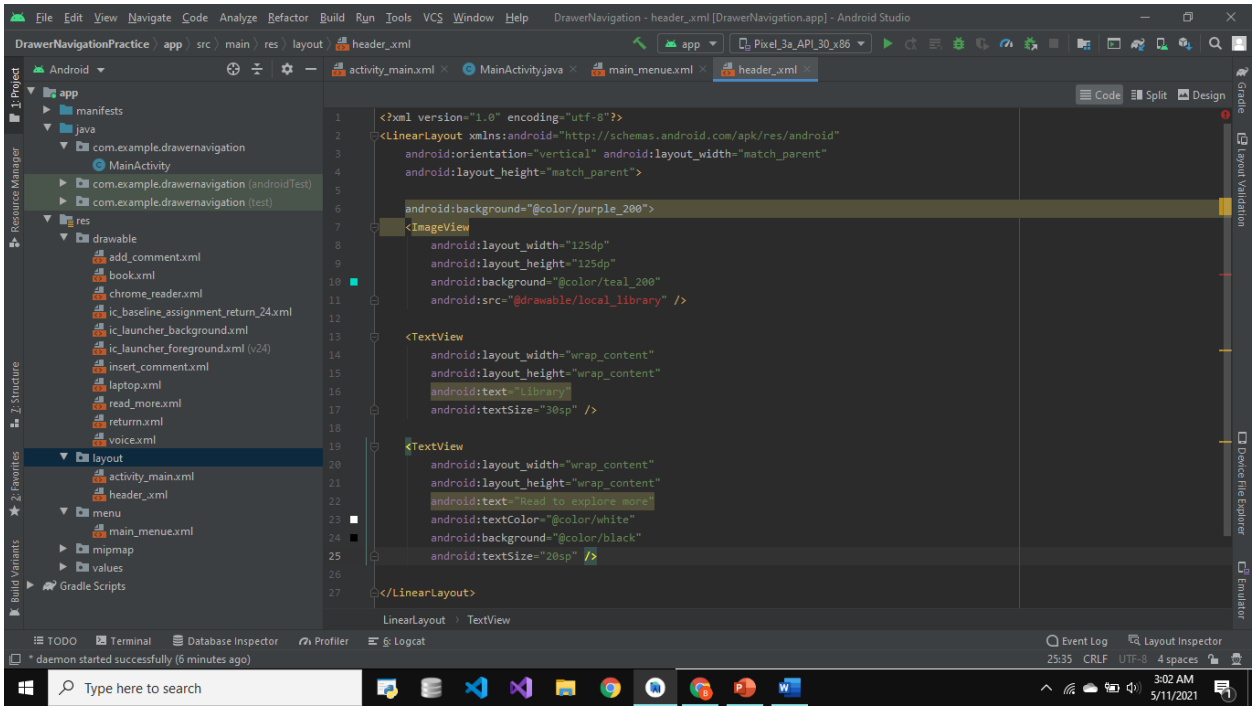

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DrawerNavigation - navigation_drawer.xml [DrawerNavigation.app] - Android Studio
DrawerNavigationPractice app src main res menu navigation_drawer.xml
app
Pixel_3a_API_30_x86
Android
activity_main.xml
MainActivity.java
ic_baseline_android_24.xml
navigation_drawer.xml
Code Split Design
app
manifests
java
com.example.drawernavigation
MainActivity
com.example.drawernavigation (android
com.example.drawernavigation (test)
res
drawable
layout
menu
navigation_drawer.xml
mipmap
values
Gradle Scripts
Palette
Cast Button
Menu Item
Search Item
Switch Item
Menu
Group
Component Tree
menu
Attributes
10:00
DrawerNavigation
1:1
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants
Gradle
Emulator
Device File Explorer
TODO
Terminal
Build
6: Logcat
Profiler
Database Inspector
Event Log
Layout Inspector
Gradle sync finished in 20 s 657 ms (13 minutes ago)
6:1 CRLF UTF-8 4 spaces
Type here to search
2:25 AM
5/11/2021


activity_main.xml
MainActivity.java
main_menue.xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
<group android:checkableBehavior="single">
<item
android:id="@+id/nav_book"
android:icon="@drawable/book"
android:title="Book" />
<item
android:id="@+id/nav_return"
android:icon="@drawable/returrn"
android:title="Return" />
<item
android:id="@+id/nav_laptop"
android:icon="@drawable/laptop"
android:title="Laptop" />
<item
android:id="@+id/nav_voice"
android:icon="@drawable/voice"
android:title="Voice" />
</group>
<item android:title="Reading">
<menu>


Palette
Cast Button
Menu Item
Search Item
Switch Item
Menu
Group
Component Tree
menu
group
nav_book
nav_return
nav_laptop
nav_voice
media_route_menu_item
10:00
DrawerNavigation
Book
Return
Laptop
Voice
Cast
1:1


New
Link C++ Project with Gradle
Cut Ctrl+X
Copy
Paste Ctrl+V
Find Usages Alt+F7
Find in Path... Ctrl+Shift+F
Replace in Path... Ctrl+Shift+R
Analyze
Refactor
Add to Favorites
Show In Resource Manager Ctrl+Shift+T
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Run 'Tests in 'layout'' Ctrl+Shift+F10
Debug 'Tests in 'layout''
Run 'Tests in 'layout'' with Coverage
Create 'Tests in 'layout''...
Show in Explorer
Directory Path Ctrl+Alt+F12
Open in Terminal
Kotlin File/Class
Layout Resource File
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Directory
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Resource Manager
7: Structure
2: Favorites

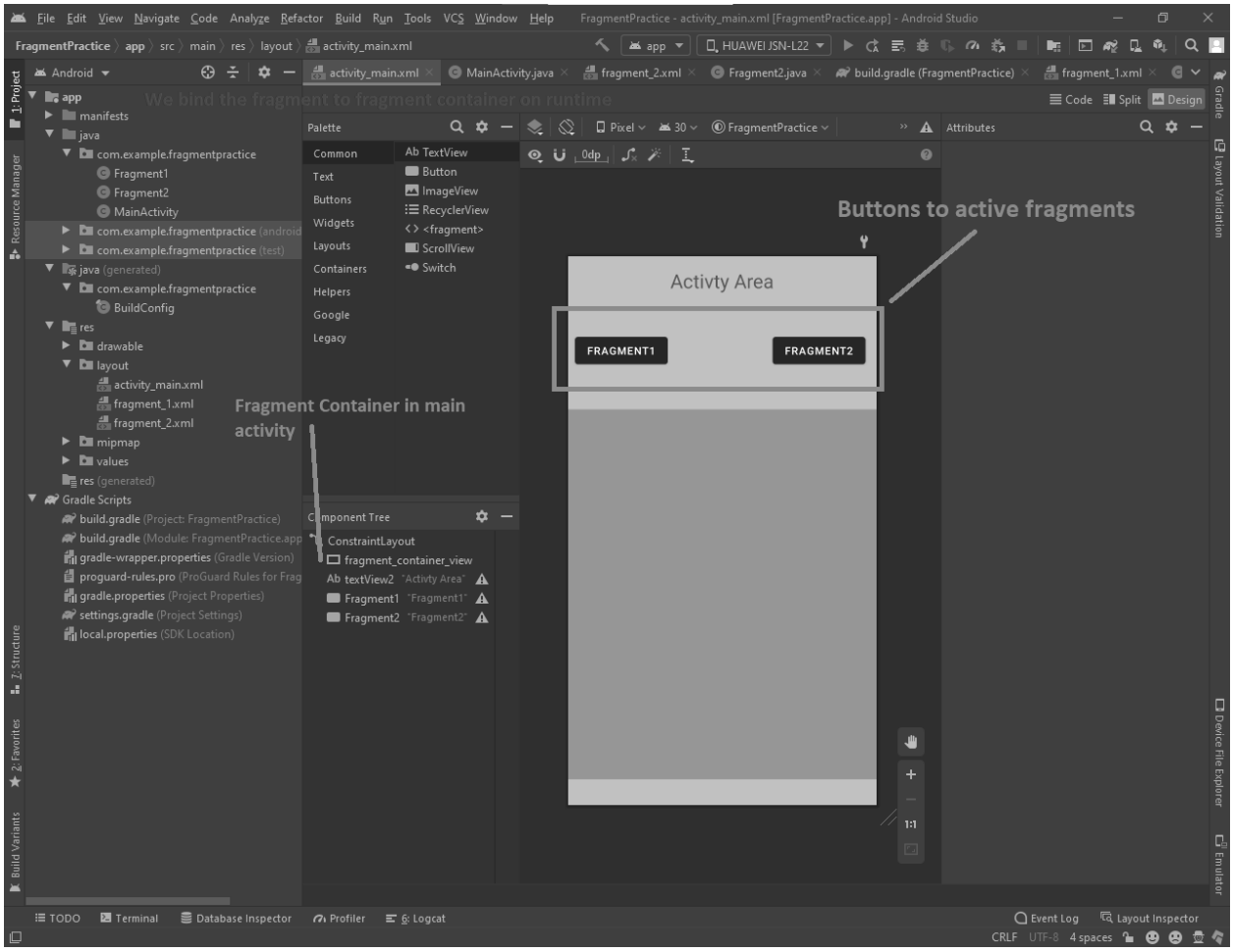

New Resource File
File name: header
Root element: LinearLayout
Source set: main src/main/res
Directory name: layout
Available qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Chosen qualifiers:
Nothing to show
>>
<<
OK
Cancel


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DrawerNavigation - header_xml [DrawerNavigation.app] - Android Studio
DrawerNavigationPractice app src main res layout header_xml
app
Pixel_3a_API_30_x86
Android
activity_main.xml
MainActivity.java
main_menue.xml
header_xml
Code
Split
Design
app
manifests
java
com.example.drawernavigation
MainActivity
com.example.drawernavigation (androidTest)
com.example.drawernavigation (test)
res
drawable
add_comment.xml
book.xml
chrome_reader.xml
ic_baseline_assignment_return_24.xml
ic_launcher_background.xml
ic_launcher_foreground.xml (v24)
insert_comment.xml
laptop.xml
read_more.xml
returrn.xml
voice.xml
layout
activity_main.xml
header_xml
menu
main_menue.xml
mipmap
values
Gradle Scripts
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="match_parent"
android:layout_height="match_parent">
android:background="@color/purple_200">
<ImageView
android:layout_width="125dp"
android:layout_height="125dp"
android:background="@color/teal_200"
android:src="@drawable/local_library" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Library"
android:textSize="30sp" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Read to explore more"
android:textColor="@color/white"
android:background="@color/black"
android:textSize="20sp" />
</LinearLayout>
LinearLayout TextView
TODO
Terminal
Database Inspector
Profiler
6: Logcat
Event Log
Layout Inspector
daemon started successfully (6 minutes ago)
25:35 CRLF UTF-8 4 spaces
Type here to search
3:02 AM
5/11/2021

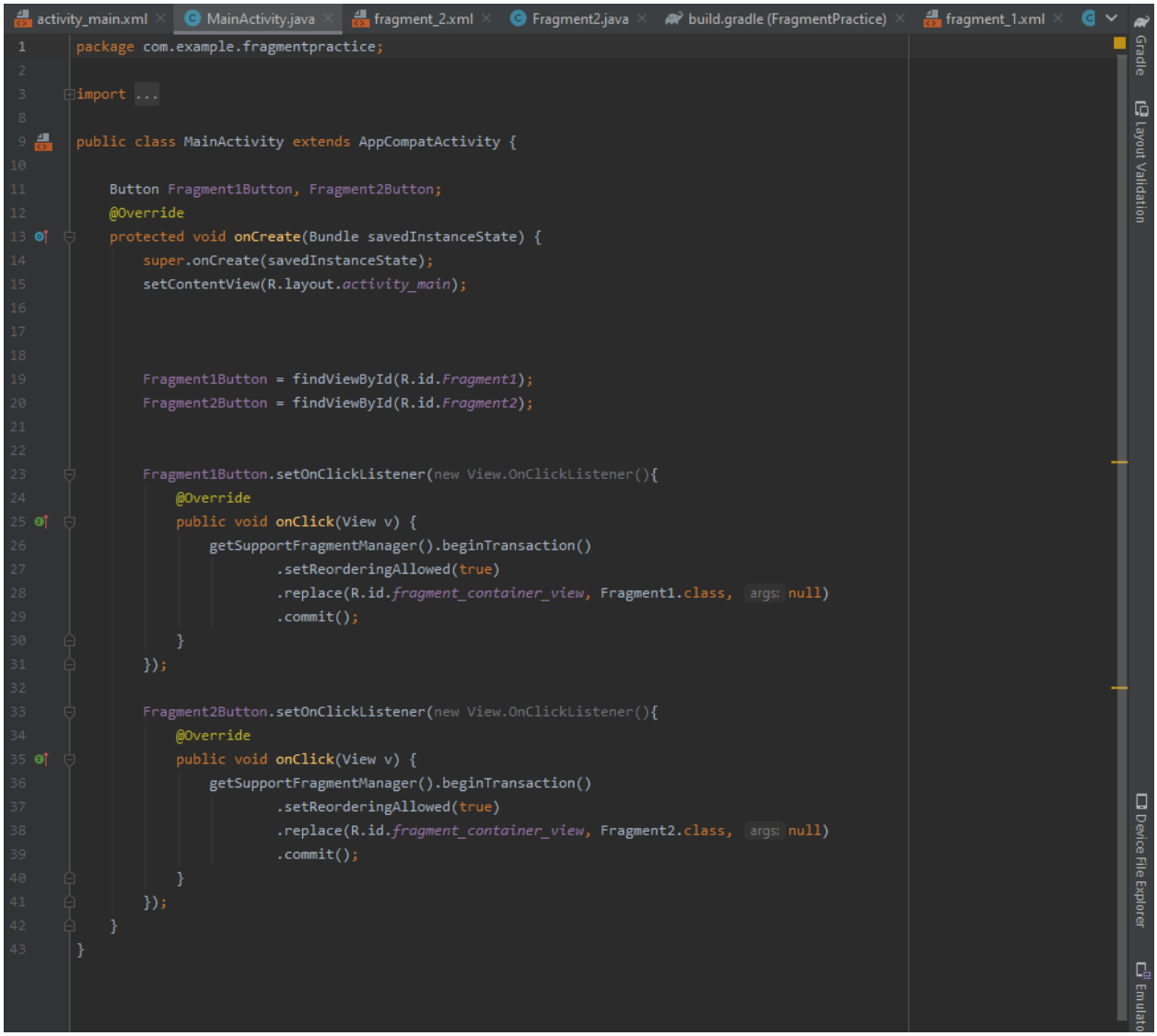
Fragments

A Fragment represents a reusable portion of your app's UI. A fragment defines and manages its own layout, has its own lifecycle, and can handle its own input events. Fragments cannot live on their own--they must be hosted by an activity or another fragment.

Application



MainActivty with a fragment container (replace on button click)




```java
package com.example.fragmentpractice;

import ...

public class MainActivity extends AppCompatActivity {

    Button Fragment1Button, Fragment2Button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);



        Fragment1Button = findViewById(R.id.Fragment1);
        Fragment2Button = findViewById(R.id.Fragment2);


        Fragment1Button.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                getSupportFragmentManager().beginTransaction()
                        .setReorderingAllowed(true)
                        .replace(R.id.fragment_container_view, Fragment1.class,  args: null)
                        .commit();
            }
        });

        Fragment2Button.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View v) {
                getSupportFragmentManager().beginTransaction()
                        .setReorderingAllowed(true)
                        .replace(R.id.fragment_container_view, Fragment2.class,  args: null)
                        .commit();
            }
        });
    }
}
```

Fragment – 1

Fragment - 2