**Name : Khadija Naveed**

**Roll No. : Mcsf19a031**

**Before Mids Syllabus**

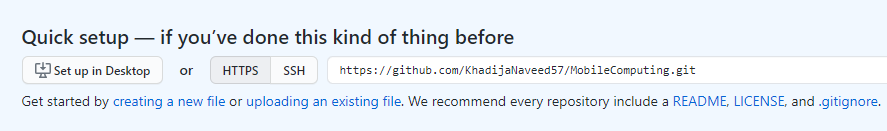
**Github:**

1. Install **“Git”.**
2. I work on “Git Bash” instead of **“Git Command”.**
3. First of all I create an account on github and then a repository to check that how it works.

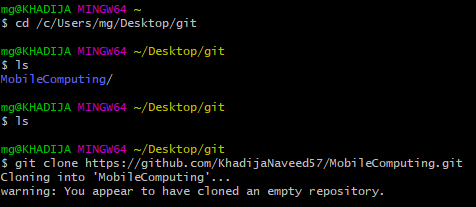
Here is the link to my repository on github:

[**https://github.com/KhadijaNaveed57/MobileComputing.git**](https://github.com/KhadijaNaveed57/MobileComputing.git)

Here is the screenshot of the link:

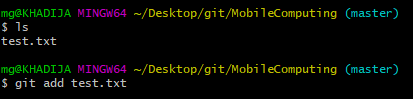


1. Now I create a folder in my computer with name git and create a clone of my central repository in this folder using the command.

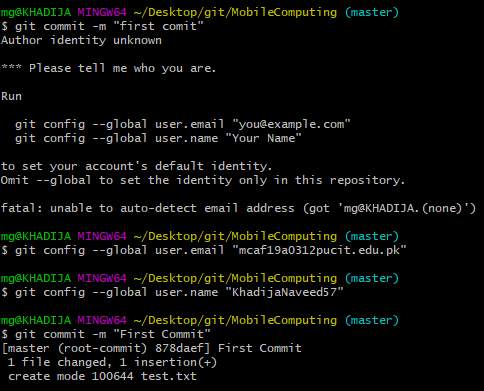


Cloned repository has been created.

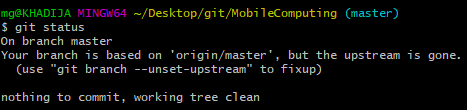
1. Now I save a random file in this folder and add using add command.



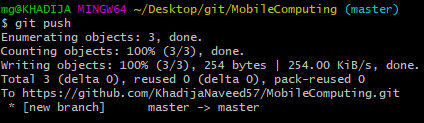
1. I commit this file after some configuration.



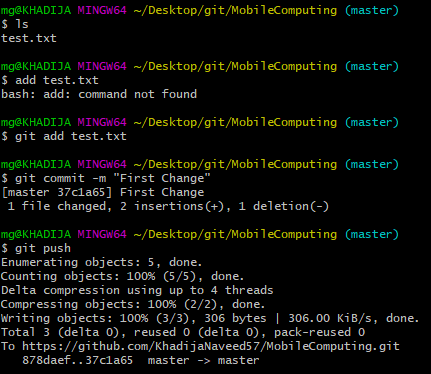
1. We can also check the status of the repository using the command.



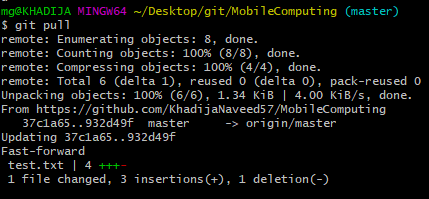
1. Now push the file in the central repository.



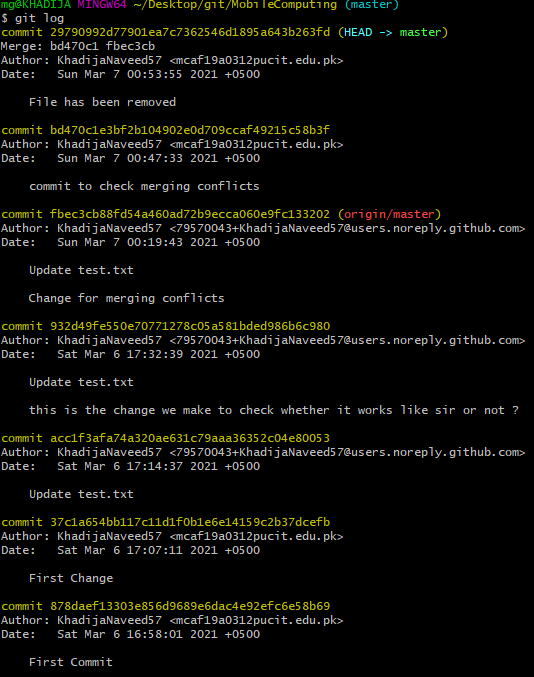
1. After applying some changes in the file we apply the same procedure to update the file in the central repository.



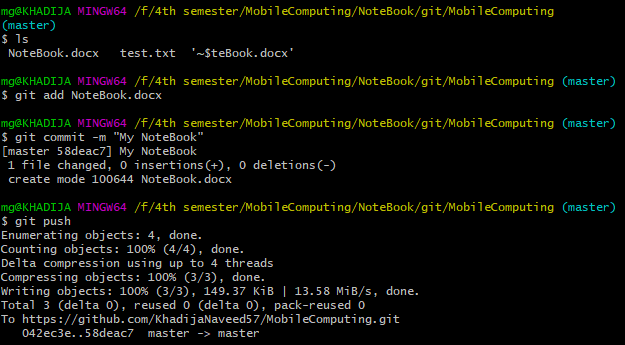
1. If we want to change in the central repository, we apply changings and use pull command to pull that changed file in our local repository.



1. If we apply changings in the local repository and the central repository at the same time there may occur the merging conflict. We should remove that conflict first and then save the changings according to our need.
2. We can use the git log commands to get the detail about the commits.



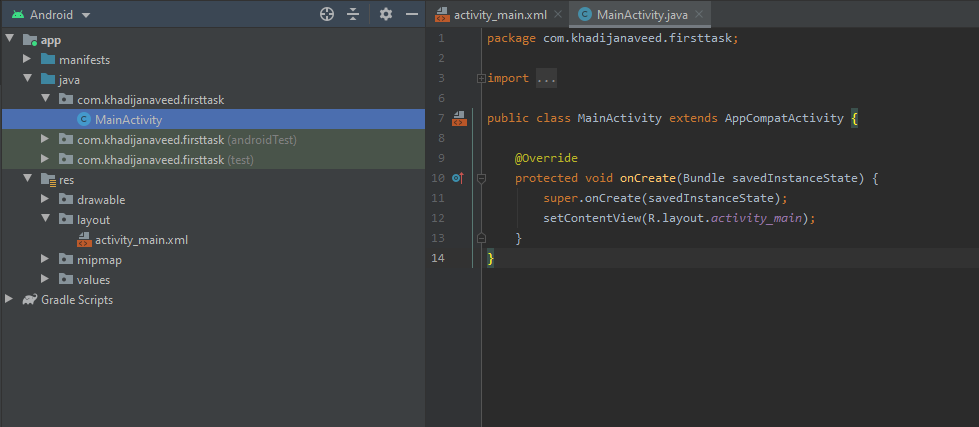
1. Now I’m applying the same procedure to save this file in the folder and then upload that file in the central repository and this fill will work like my notebook.



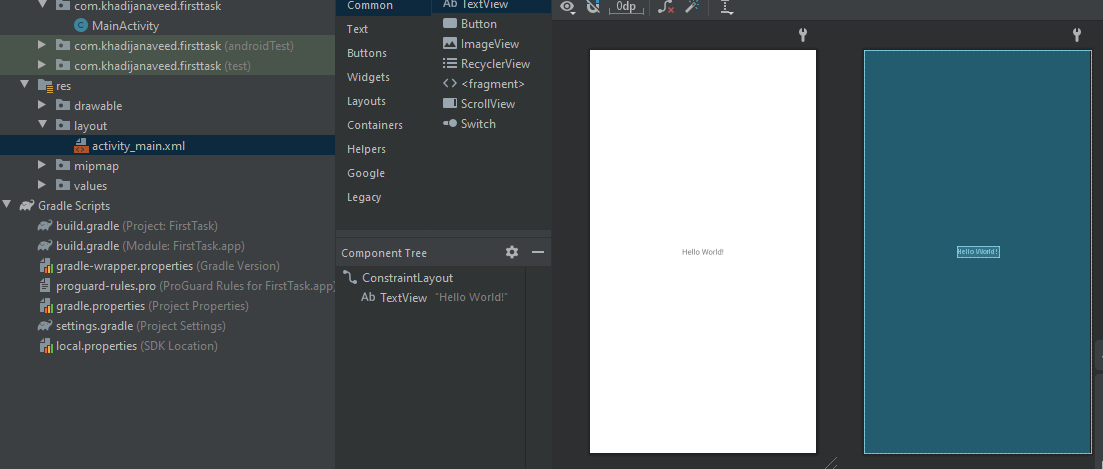
**Android Studio:**

**Lecture no. 4:**

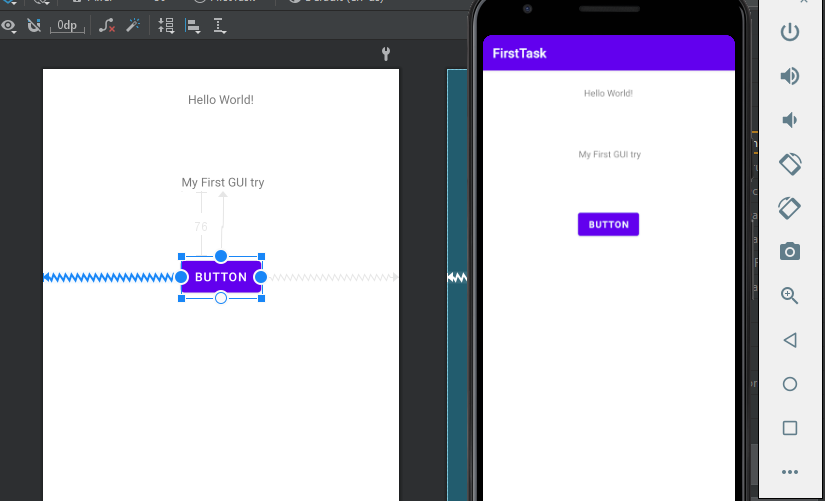
1. Install “Android Studio”.
2. Start a project in android studio.
3. Blank project like as follows:



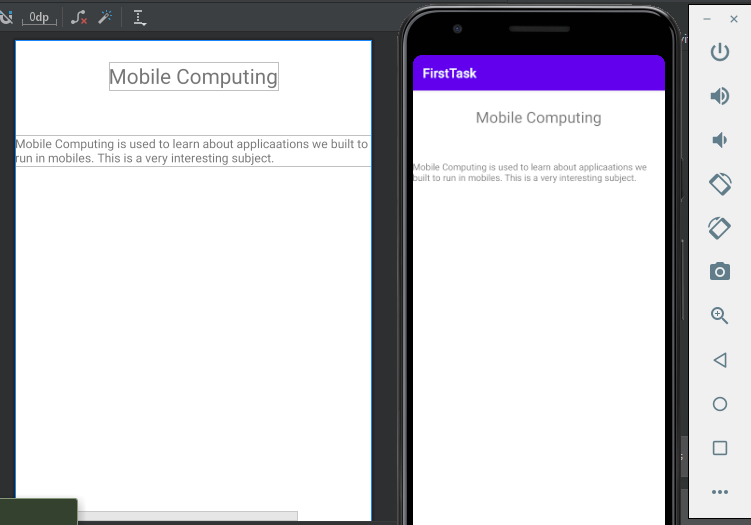
1. We can also write TODO. These are just like comments but TODO’s of the complete project shown in the file where ever we are.
2. Gradle Scripts-> build.gradle … This is the place where we can find all the information related to our project or application.
3. We can access Graphical user interface res->layout->activity\_main.xml



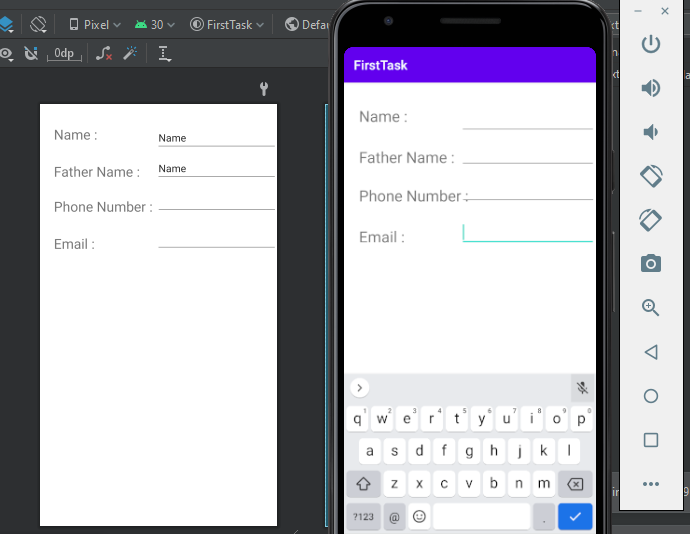
1. My First try to create a GUI. Here is the GUI interface and I’m gonna write code in another file in the same repository.



1. Here is the interface of my Second GUI and I’m gonna write code in another file in the same repository.



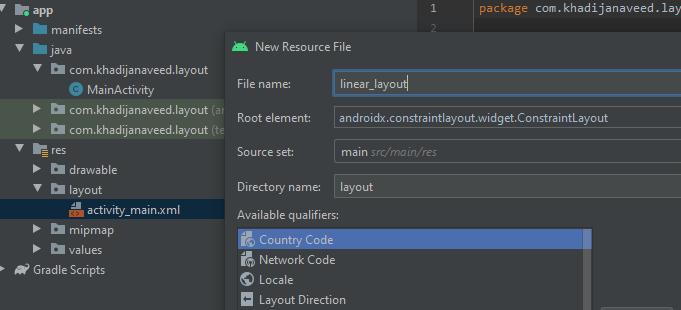
1. Here is the interface of my Third GUI and I’m gonna write code in another file in the same repository.



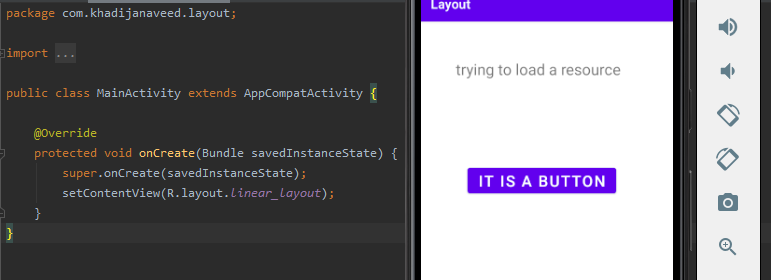
**Lecture no. 5:**

**Layout:**

1. Here we are going to learn about Layout. I create a new project and we’ll add a new layout recsource file from res->layout->new->layout resource file and name it as linear\_layout

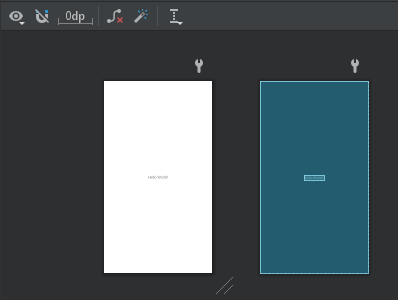


1. Now add some button or textview in the design view and will load this resource (We inform in main\_activity.java that what or where is our main point, setContentView will declare that which view we are going to set.). So to change the view we do the code as shown below. Here is the result too.



**Layout Parameters:**

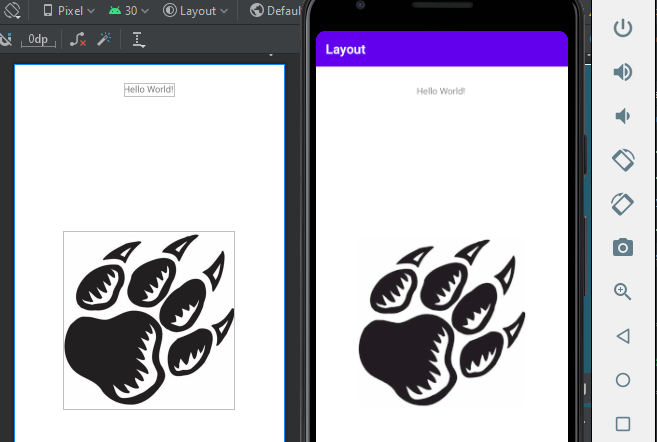
1. **Implementation ‘androidx.constraintlayout:constraintlayout:2.0.4’** this dependency would be changed in our project.
2. We can clear all constraints using the following cross symbol.



**Image:**

Now I’ll set an image in the design view.

1. First of all copy any image and paste in the drawable and then drag imageview in the GUI view and then click on project and then desired picture. Here I tried once.



1. We can convert a linear\_layout into a constraint\_layout
2. Till now we have done everything using design view but now we’ll write a code against our design.
3. We can create an instance of button by writing the following code

**Button button = findViewById(R.id.button4);**

**TextView textview = findViewById(R.id.textview3);**

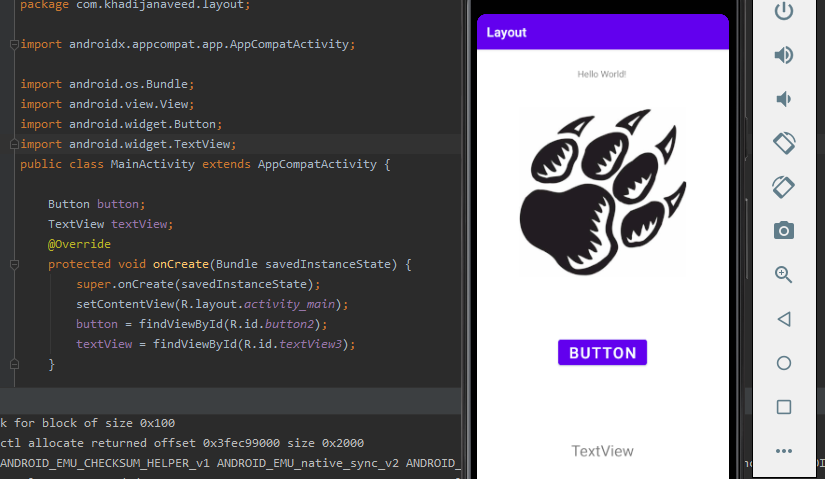
1. For such coding purpose we need to import ;

android.widget.Button;

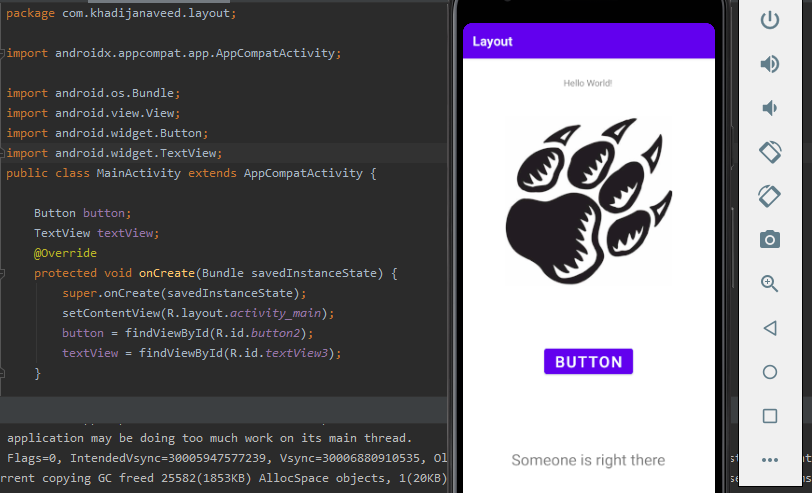
android.widget.TextView;

1. Here below we create a button and and change the textview after performing an action against the button.

Before I click the button:



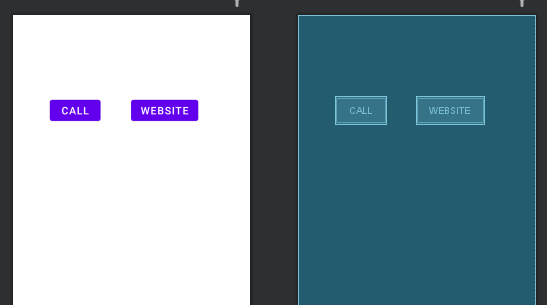
After I click the button:



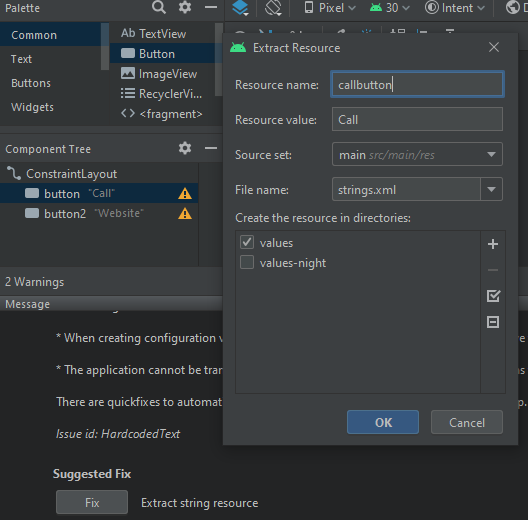
**Lecture No. 6:**

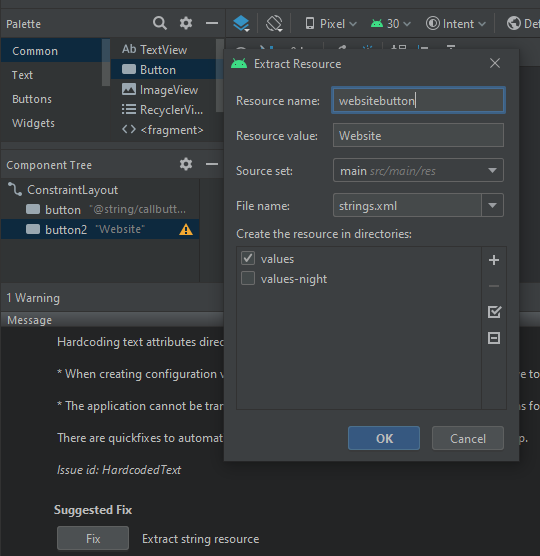
**Intents:**

1. Here we are going to learn about Intents.
2. I create an empty project, drag and drop 2 buttons name 1 as Call and 2nd as Website

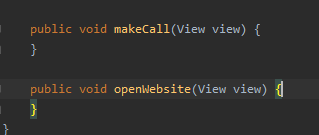


1. After I change there text, there an error arose to convert those hard coded text to string resource.

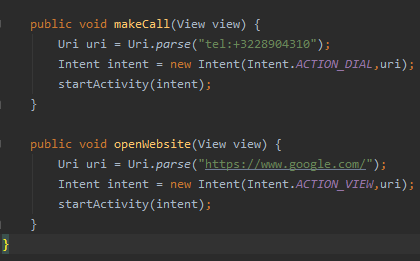




1. Then I make 2 function to perform actions against these two buttons.



1. I write code to perform actions and I use intents and Uri:



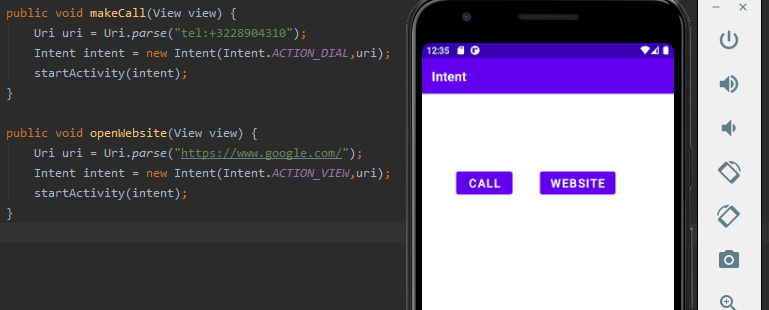
1. What is Uri?

The most common form of Uri is the web page address. Android uses Uri string as the basis for requesting data in a content provider(i.e. to retrieve a list of contacts) and for requesting actions(i.e. opening a webpage in a browser.)

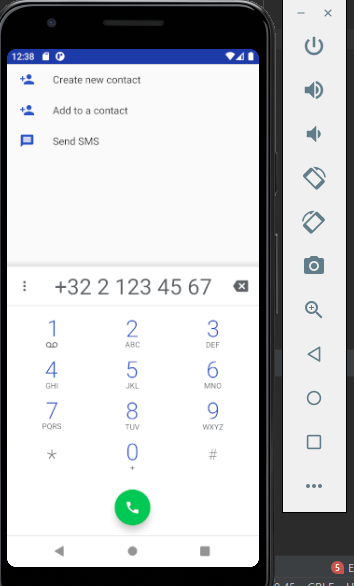
1. What is Intent?

Intent are the objects which is used in android for passing the information among Activities in an application and from one application to another also. Intents are used for communicating between the application components and it also provides the connectivity between two apps.

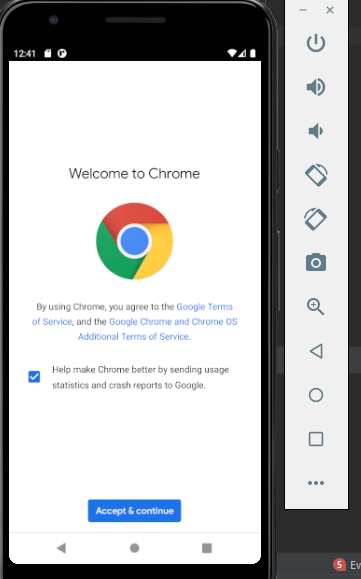
1. After I run the project following steps are done:



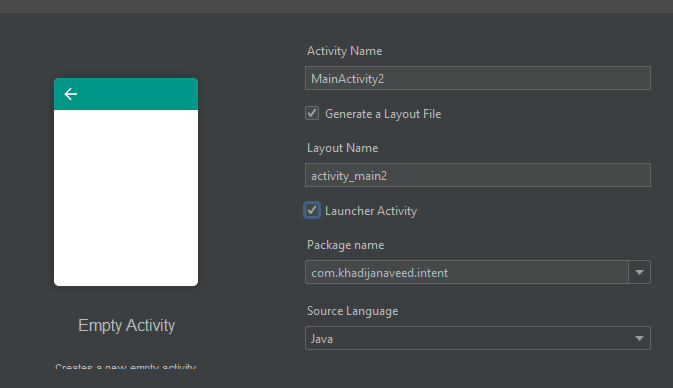
1. After I press CALL Button screen appears as shown below:



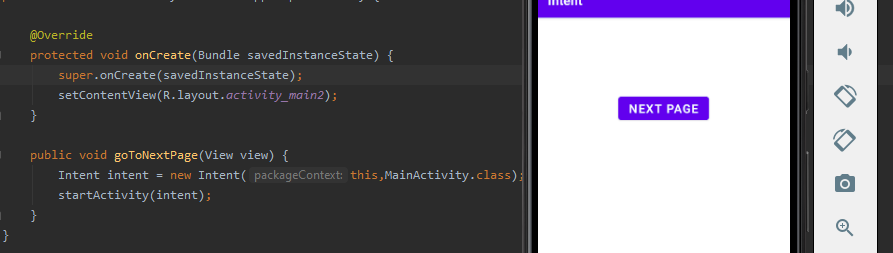
1. When I press website button, screen appears as shown below:



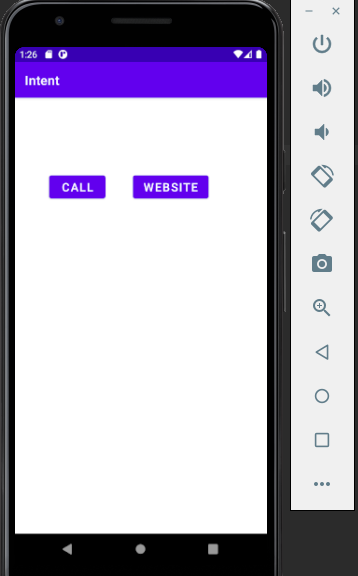
1. We can also add new activity while creating new activity, to select launcher activity means that do we want to add this activity as by-default page.



1. Now I have add a button to the mainactivity page 2 to go to the page 1 and create a function in it’s main to perform task against this button.



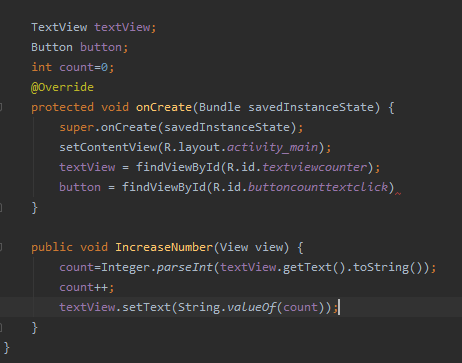
1. After pressing this NEXT PAGE button we go to the next page as you can seen below:



**Lecture No. 7:**

**Activity Life Cycle:**

1. Firstly, I have created a project and name it as activity\_life\_cycle.
2. Then I create a counter to check.
3. I have used vertical linear layout in this project and then drag and drop 2 textviews and a button.
4. I create a click event for a button.



1. If we write the above code, our counter will work properly but when we rotate screen our counter will again jump to 0. This is all because of activity\_life\_cycle.
2. Activity\_lifr\_cycle is a combination of such steps:

onCreate();

onStart();

onResume();

onPause();

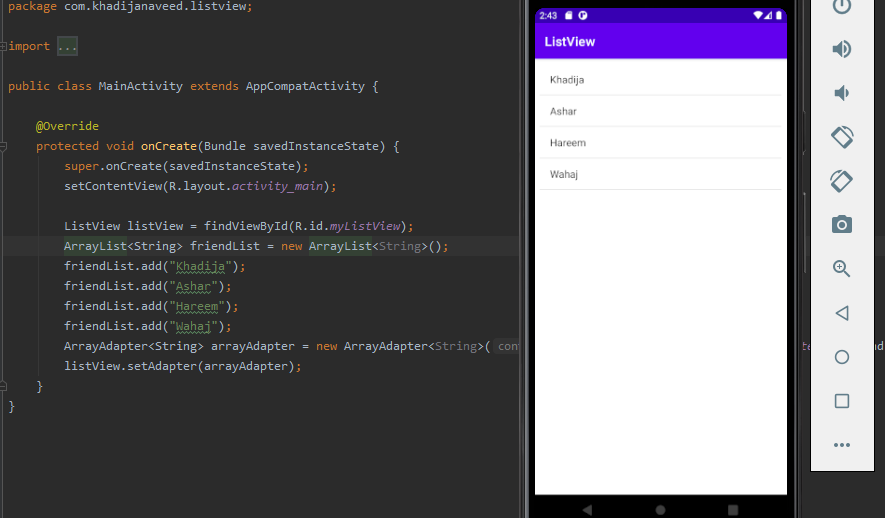
onStop();

onDestroy();

**Lecture No. 8:**

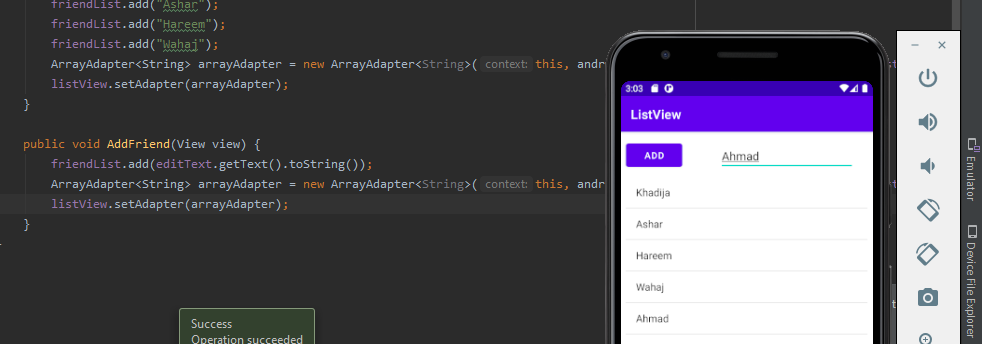
**ListView:**

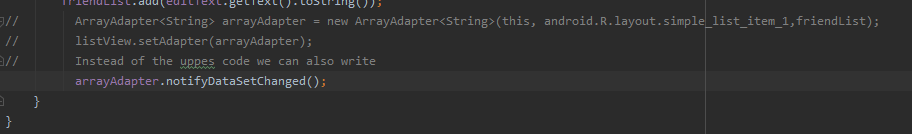
1. In this lecture, we learnt about adapters. In short adapters are used to get data from database or any dataset and show them to the UI Components.
2. How to use adapters? I have drag and drop a layout from the legacy, apply constraints and then write code to show the hardcoded data in to that listView as shown below:

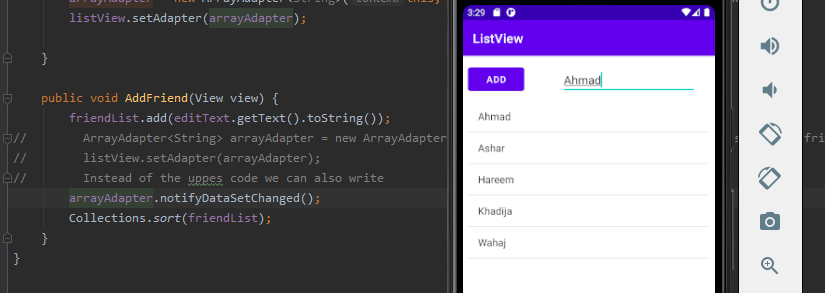


**ListView binding:**

1. We learnt is this topic that how to add an entry by the user.
2. Drag and drop a button and a edittext before the listview and then create a function against the button to add a new friend.

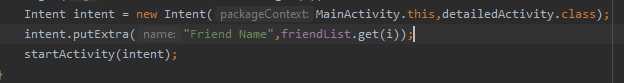


1. In a plain text it always show text but we can remove the text and can put hint there.
2. 
3. We can also sort data.

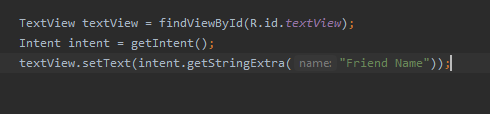


**How to move data from 1 activity to another:**

1. I create an activity and drap and drop a textview in it.
2. I write the code below to push data from 1 activity to anonther:



1. I write the code below to get data from the other activity:



1. Now, when I click on any name it will be shown on another activity.

