CSE2216
Application Development Lab

Lab 2 : Designing Layouts

Md. Fahim Arefin

Contents

- Activity vs Layout

- Different types of Layouts

- XML Syntax

- Views - TextView, EditText, ImageView, Button

- Exporting APK

- FAQ

- Common Issues

Android App

How do you talk to your computer/ phone

- IDE – Integrated Development Environment

- XML – Xtensible Markup Language

- Java/Kotlin - Logics

ACTIVITIES

Activities define what each screen does

LAYOUTS

Layouts define what each screen looks like

Activity and Layouts

1. Device launches app and creates an **activity** object.

2. The **activity** object specifies a **layout**.

3. The **activity** tells Android to display the **layout** onscreen.

4. The user interacts with the **layout** that's displayed on the device.

5. The **activity** responds to these interactions by running application code.

6. The **activity** updates the display which the user sees on the device.

Creating Layouts

After creating new project,
your **activity_main.xml** should look like this.

- Click on **Design t**o get to the drag & drop interface
- Click on **Split** to view preview while coding
- Click on **Code** to code without any distractions

When we go to Design tab,
we can change attributes for each element and there's a
search button on top right to quickly search attributes
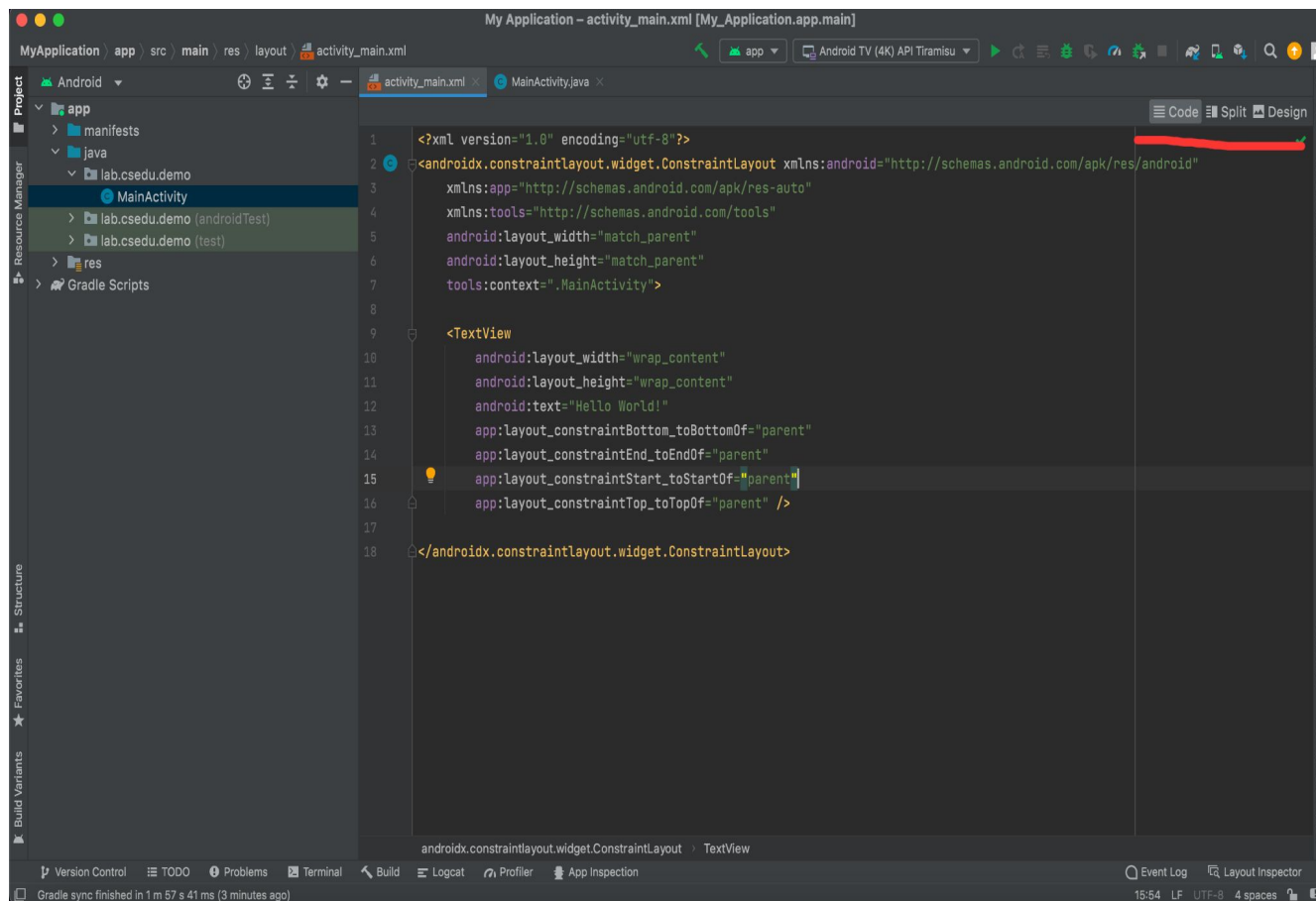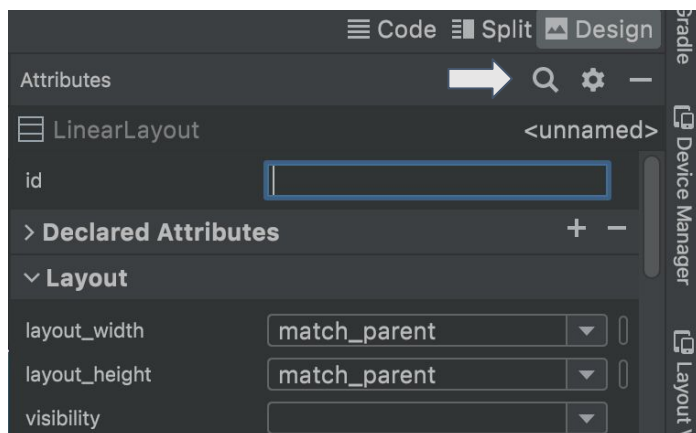
Layout Variations

- ViewGroups : LinearLayout, RelativeLayout, GridLayout, ConstraintLayout, TableLayout etc.

- **Width/height :**

  - Fixed width/height : 200dp

  - Only take space which is required : wrap_content

  - Take all available space in the parent layout : match_parent

  - Evenly spacing out children : layout_weight =**1** and height/width = **0dp** and in the parent layout, **weightSum = 2**

LinearLayout

- android:orientation
  - "vertical" : All contents will be placed one **below** another
  - "horizontal" : All contents will be placed one **beside** another

- Nested LinearLayout
  - If we want to place two buttons side by side, we can add a LinearLayout(A) inside the main LinearLayout(B) and add the buttons in this nested LinearLayout

Changing to LinearLayout

When a new activity is created, the layout file looks like this
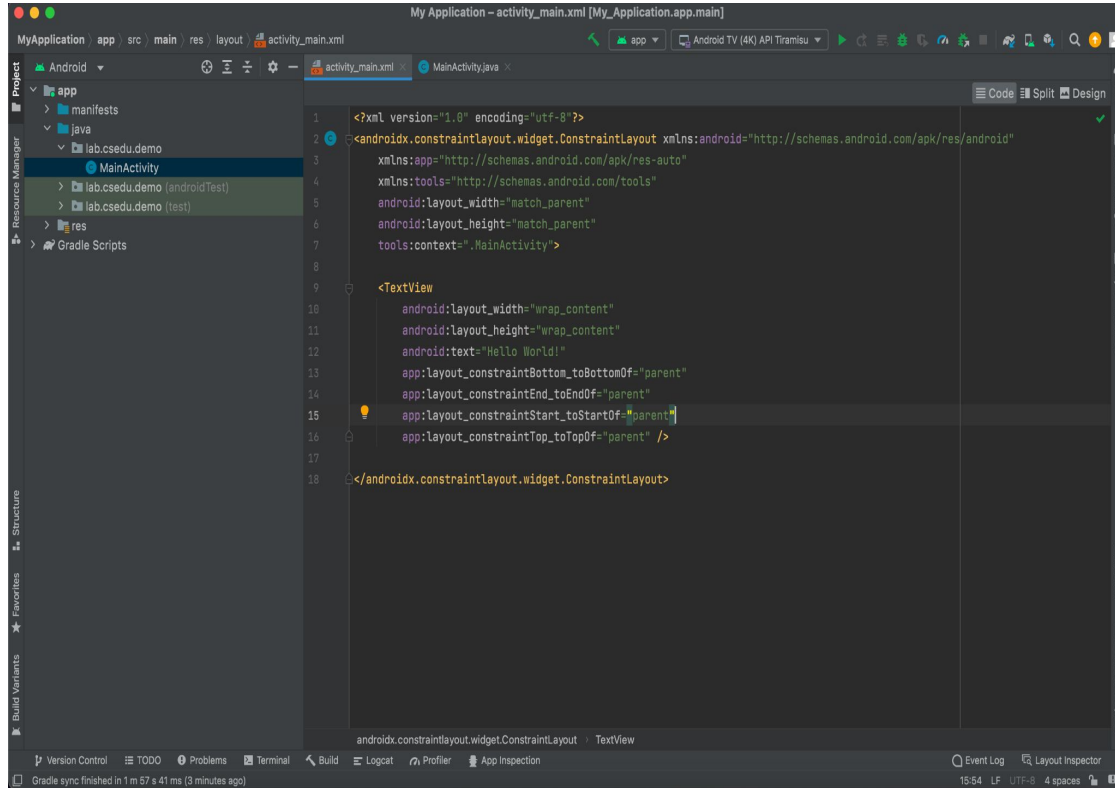
For simplicity, we will replace **ConstraintLayout** to **LinearLayout** today. The code should be like this :


```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        />

</LinearLayout>
```

XML Syntax

- Starts with : **<**
- Name of XML Element
- Attribute names should have "android:name"
- Then = attribute value inside double quote
- Ends with : **/>** : Self closing tag

- Separate closing tag is useful for LinearLayout, when it contains more views
- Generally uses default values,
- if you want to change, then you can set it in XML

Notes about XML

- XML Tags are **case sensitive** and can not contain space:
    - Correct : TextView
    - Incorrect : ~~Textview, Text View, txtView~~

- Each Element should have a unique id, it is defined as

    android:id = "@+id/text_view_1"

    Here text_view_1 is the id.

How to Design contents of a Layout
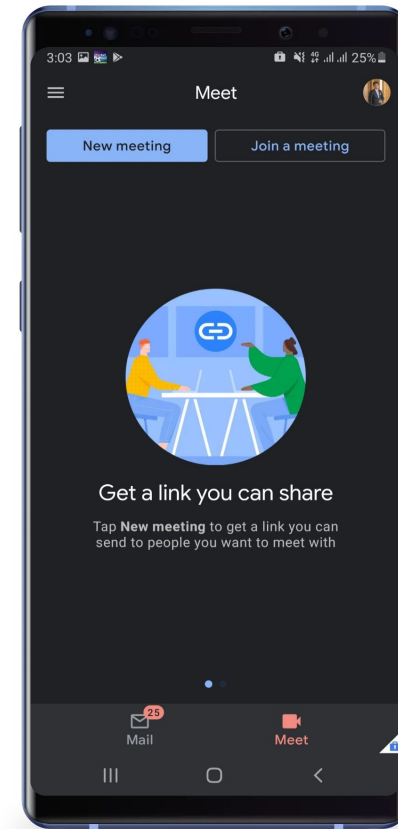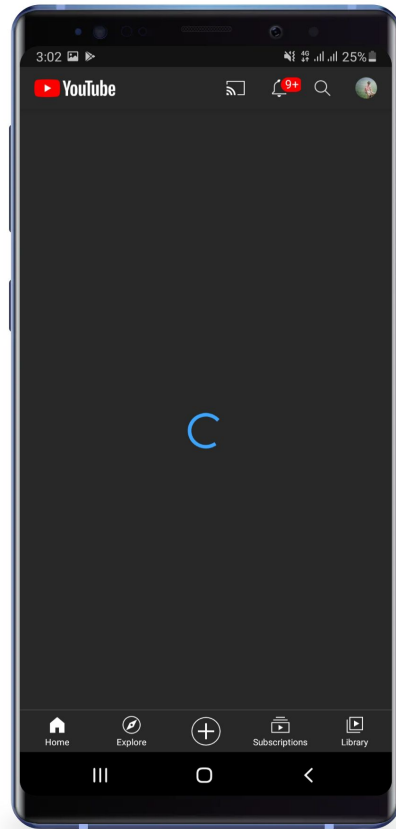
Sign Up/ Log In/ Forgot Password

screen of your project

Views

Views - Youtube & gmail

Views

- Any rectangular area in your screen is a view

- Basic building block to design the layout

- The borders are generally invisible

- View names are generally camelCase – beginning of each word is capital

TextView

TextView Attributes

- layout_margin - Add margin(blank space) around the Textview

- padding - Add space inside TextView

- textSize - Font Size, in sp

- textColor - Change color of text

- fontFamily - Change font (Select from dropdown)

- gravity - sets the gravity(alignment) of the contents of this View.

- layout_gravity - sets the gravity of this View relative to its parent.

- alpha : Adding transparency (Example : 0.8)

Customizing TextView

- Changing background color : background = "@color/black"
- Changing background image : background = "@drawable/bg"
- Choosing Colors :

  https://material.io/design/color/the-color-system.html#tools-for-picking-colors

- Changing TextSize : 24**sp**
- **width/height :** wrap_content/match_parent/200dp/0**dp**+weight
- What are dp and sp ? Explained next.

DP vs SP

- Dp = density independent pixels, used to define width, height, padding, margin
- Medium, High and Ultra High Resolution Devices – 2x2 pixels

- Using dp, it's the same physical size on all devices
- All touch targets should be **48dp at least**

- SP = Scale Independent Pixels [Only for font size]
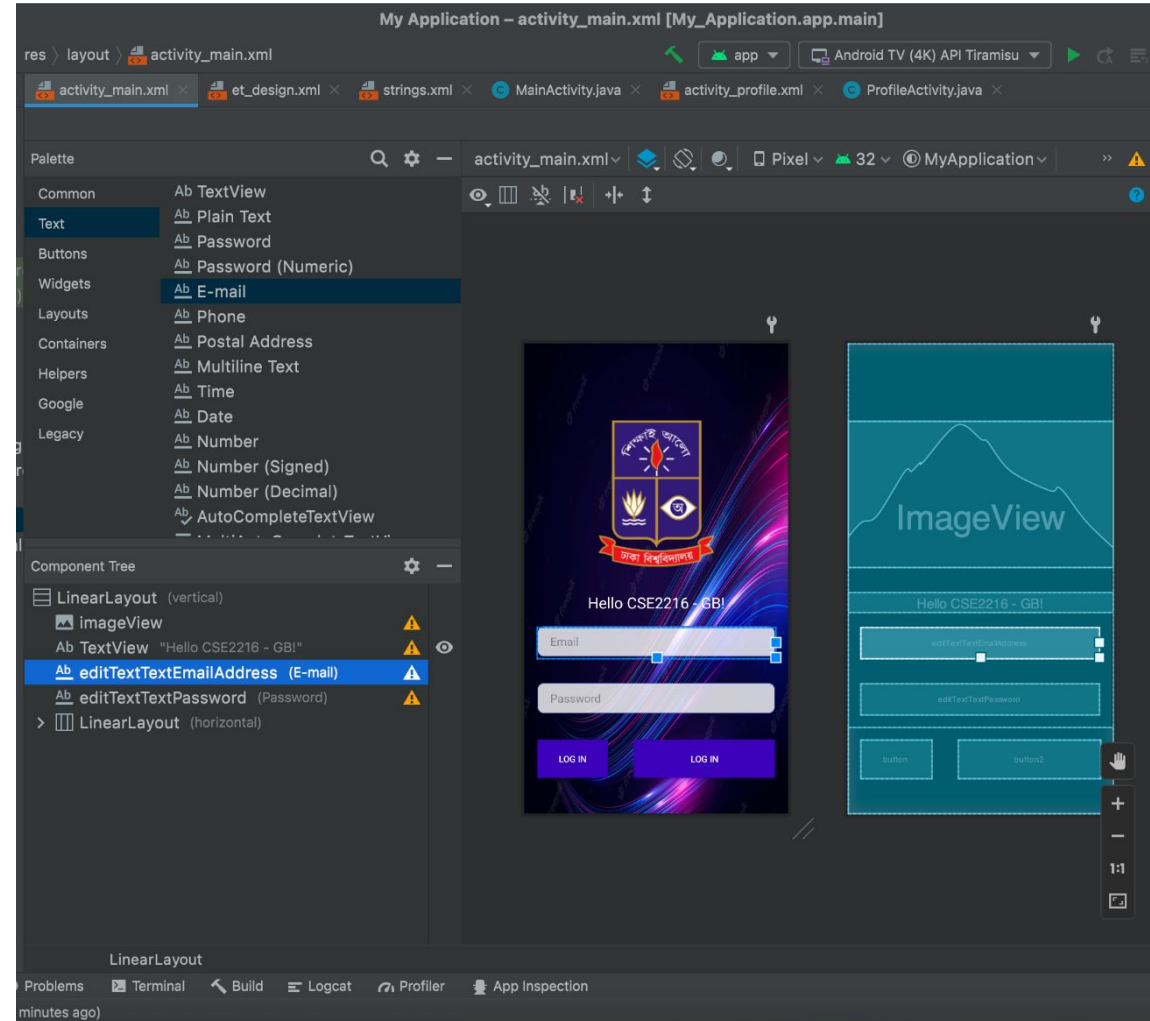
How do I take input ?

EditText

- Go to the "Design" tab
- Click on Text > Email
- Drag it down to the component tree below.

This will create an input field which can take emails as input

You can drag other types to take other types of input.

An EditText has all properties of a textview

and

android:hint : For placeholder texts.

How do I add my logo !

Drag and Drop ImageView

- Drag and drop an image in drawable folder
- Then drag and drop an image view in the component tree. A code snippet like this should be added to your code.

```
<ImageView
    app:srcCompat = "@drawable/logo"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:scaleType = "center"/>
```

- If you face issues, check slides 43 and 44.

Placing Resources

- Layout files in **res > layout > activity_name.xml**

- Drawable xml/images inside **res> drawable** folder

- Strings in **res > values> strings.xml**

- Colors in **res>values>colors.xml**
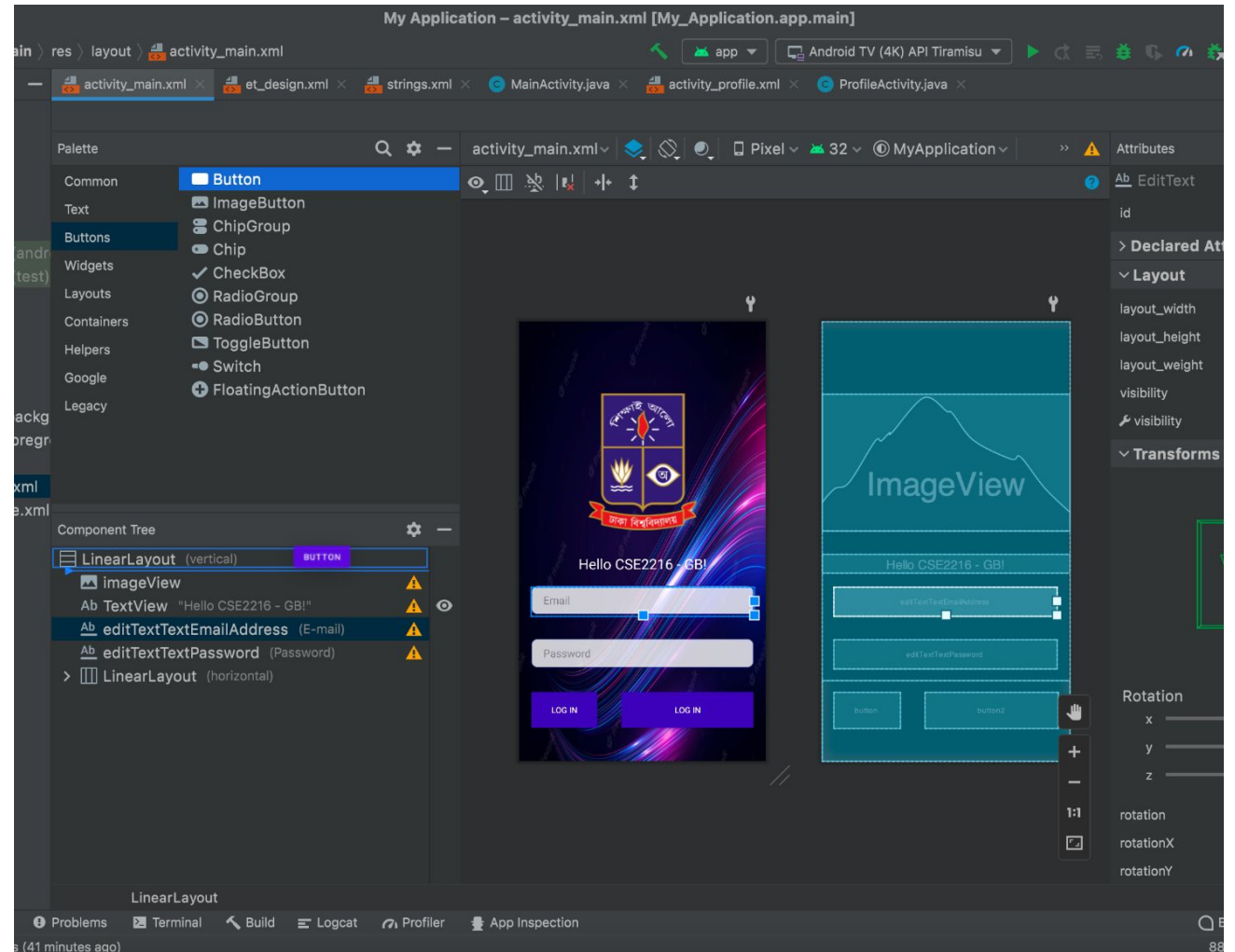
How do I add buttons !

Buttons

We can add a button by dragging the Button element to the component tree below.

This will generate code for the button.

You can go to the "Code" tab to view that.

If you can't change the background color of the button, check slide 41.

How do I generate APK

Signed Release APK - https://youtu.be/db7xor5ba4w

FAQ

Creating a new Project

Click on **File > New > New Project**

Scroll down and choose **"Empty Views Activity"**

In the next screen :

- Give your activity a name - **"MainActivity"**
- Define a 3 part package name, example - "lab.csedu.demo"
- Note the Save Location - Your project will be saved there
- Select Language as **Java** (By default Kotlin might be selected)
- Click on Finish

You should see two files open - MainActivity.java and activity_main.xml

A New Project looks like this
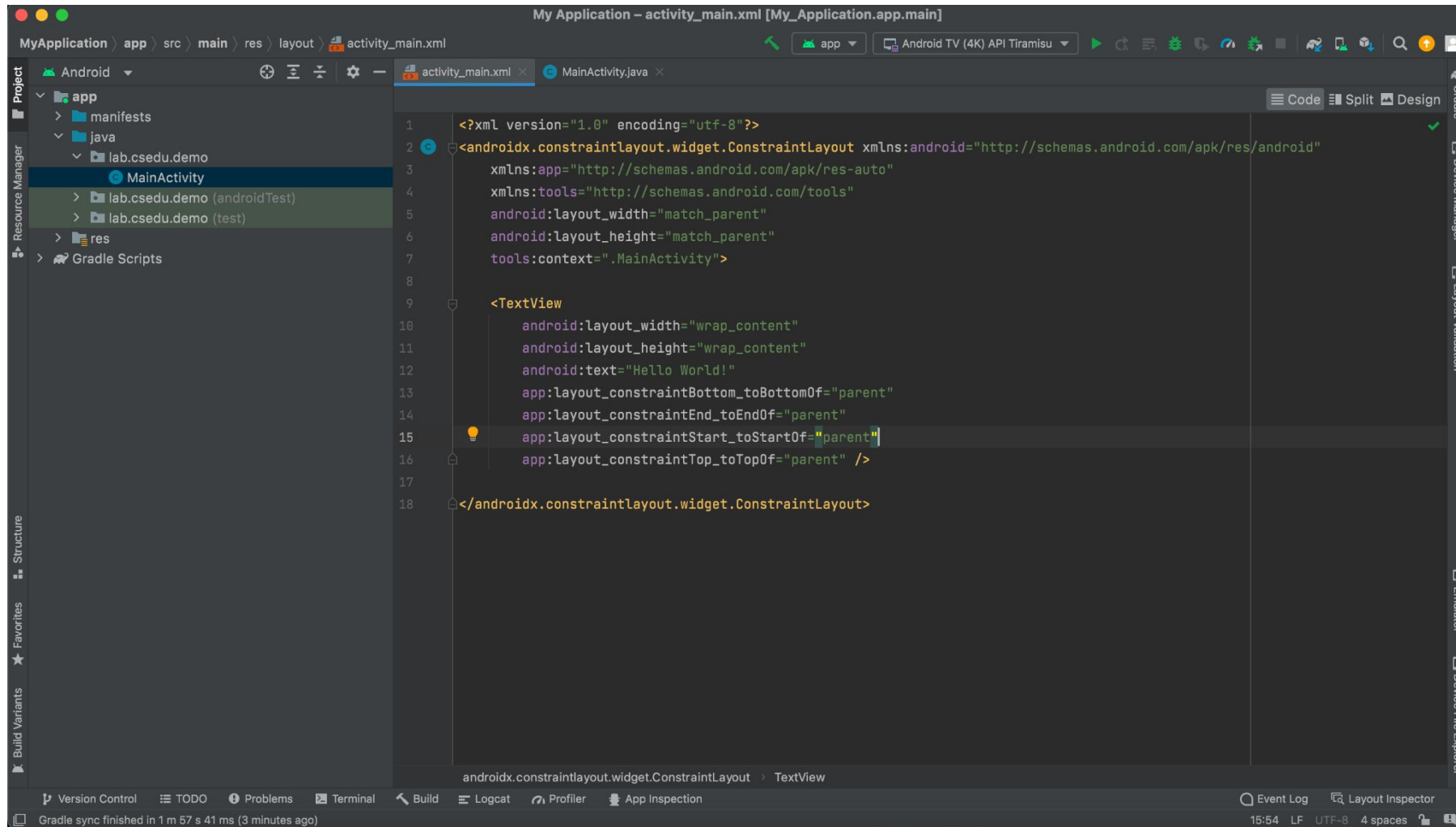
I want to make button/text fields have rounded borders.

Make Corners Round

Create an xml file named "et_design" in drawable folder, paste this:

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/white"/>
    <stroke android:width="1dp" android:color="#3F51B5" />
    <corners android:radius="10dp" />
</shape>
```

Then in the EditText/ Button, set background = "@drawable/et_design".

I added two items, only one can be seen!

Go to the code tab

Your LinearLayout has **"orientation : horizontal"** and
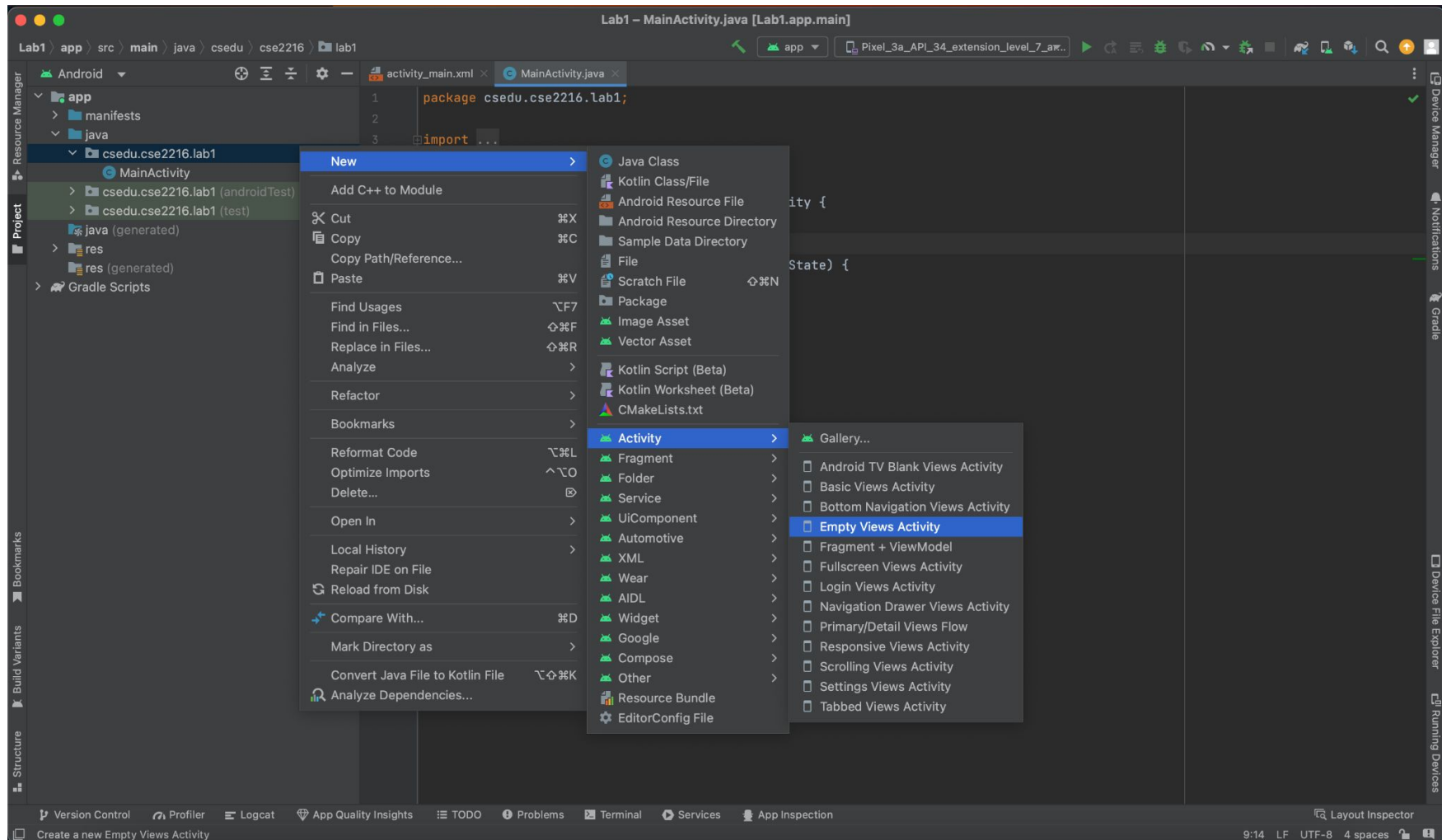your first textView has **layout_width = "match_parent"**.

Or

Your LinearLayout has **"orientation : vertical"** and
your first textView has **layout_height = "match_parent"**.

Therefore, other elements are placed outside the screen.

How to create a new Activity ?

Like this

Creating a new Activity

To create the second activity, follow these steps:

- In the Project window, right-click the **app** folder or **package** name and select **New > Activity > Empty VIews Activity.**

- In the Configure Activity window, enter "ProfileActivity" for Activity Name. Leave all other properties set to their defaults and click Finish.

Android Studio automatically does three things:

- Creates the ProfileActivity.java file.

- Creates the layout file activity_profile.xml, which corresponds with the ProfileActivity file.

- Adds the required <activity> element in AndroidManifest.xml.

How to start the new activity ?

Sending to a new Activity

Add this code to MainActivity, below "setContentView(R.layout.activity_main);"

```
findViewById(R.id.button).setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(MainActivity.this,ProfileActivity.class);
        startActivity(i);
        finish();
    }
});
```

Documentation on how to create new activity and pass data between activities

https://developer.android.com/training/basics/firstapp/starting-activity#java

Common Issues

Button background color is not changing!

In the code tab, Replace "**<Button**" with "**<androidx.appcompat.widget.AppCompatButton**".

You can then add image or color as "background" of the button.

How to position elements. LinearLayout places them one below another!

Use Padding and Margins

I added image in drawable. But my image is showing error ?

Your image name should be lowercase with underscores only. No digits or upper case letter or spaces.

After adding an image, my app is crashing

Your image might be too large. Keep it <1MB.

I can not run your given code !

You can't copy and paste the whole folder. You should see it and make the changes to your project accordingly. That's the most important part in Android Development.

Can't install app on my xiaomi phone - "Failed to install via usb - Device temporarily restricted 3-3"

This may help :

https://stackoverflow.com/questions/460202 37/install-app-via-usb-the-device-is-temporari ly-restricted

Thanks!

Any questions?

You can find me at

- fahim@cse.du.ac.bd