

// Andriod //

- 1: coding in xml file
- 2: Here Resources is connected to R.java file
- 3 like ex in MainActivity file its shown below

Setcontentview(R.layout.activity_main) -> that means R is connected to layout of activity main

4: Where is R file located?

- 1: Go to upward Andriod Select option Project file.
- 2: Go to App & Inside App Go to Build
- 3: Inside Build Go to Genrated/source
- 4: Inside source Go to buildconfig
- 5: Inside buildconfig Go to r
- 6: Inside r Go to Andriod Test/debug
- 7: Inside A test Go to andriod Then Go to App
- 8: Inside app there is R file.

5: We want to create new layout?

- 1: Go to Layout right click and open new file
- 2: Name file new_layout.xml
- 3: Apply some design
- 4: To run the File in App
- 5: Got to MainActivity
- 6: Change the Path
- 7: Setcontentview(R.layout.activity_main) <--> Setcontentview(R.layout.new_layout)
- 8: its Working

6: Set the reference of Button?

- 1: Create a object in MainActivity.java

```
private Button hibutton;
```

- 2: Go to design and Set button id hibutton

- 2: Reference the button inside the protected class

```
hibutton =(Button) findViewById(R.id.hibutton); -> Set the unique id of button  
hibutton.setText("Hello Button"); -> change the name or set the name of Text of Button  
hibutton.setTextColor(Color.BLACK); -> Change the color
```

7: where is the String file What is the use of it?

- 1: String file is located inside value
- 2: View is located inside Resources
- 3: open the String file

- 4: Change your button reference with permanent id in here like ex

```
<string name="button_name">Hey Button </string>
```

```
/* Here we reference the string by id
```

```
"button_name" jene the tame a ek j button badha ma use kre sako without changing
```

every

button name*/

5: Go to MainActivity.java

6: wrote hbutton.setText(R.string.button_name) -> button_name is referenced here & Your button String now showing in the app

instead of hbutton.setText("Hello Button")

7: Change also in layout/Activity.xml file

like ex:

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="#E91E63"
    android:text="@string/button_name" -----> Here the change is occurred,
Referenced your String inside it.
    app:layout_constraintBottom_toTopOf="@+id/textView"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.814"
    tools:visibility="visible" />
```

8: Mainfiest.xml file?

1: it define our App Structure

2: located inside App inside Manifiest file

3: Inside Manifiest

```
<application
    android:allowBackup="true"-----> Allow Backup When we live our
app
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity"> -----> Here Our Activity Screen
register
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>
```

for ex: We use according to needs

```
<uses-permission  
android:name="android.permission.ACCESS_MEDIA_LOCATION"></uses-permission>    -->  
Here We tell the user to accept the permission request
```

--> So we access your Media Permission

4: This Permission & Security Stuff are located inside Manifest file....

9: How Button Work dynamically when we click onclick function?

(1st Way is Slightly tuff)

- 1: Go to Design And Select TextView Change Their Visibility to --> Invisible
- 2: Go to Design Again Select button Change Onlick to according to your need like ex--> ShowMe
- 3: Go to MainActivity
- 4: Wrote code for ShowMe class object like ex:

```
public void ShowMe(View view){    -->    view parameter , Import view Parameter  
on Upward side like ex-->    import android.view.View;
```

```
    MTextView.setText(R.string.Show_Text); /* For onclick Button, Text is Show  
According to are need Change Text Resources To Show_text and name the Resources*/  
    MTextView.setVisibility(View.VISIBLE); /* For Visible the Text That we invisible  
earlier in Design */  
}
```

Changes are occurred in String.Xml file is

```
-->    <string name="Show_Text">Show Name</string>
```

Now go to design And Run the app And click the button it will show you the text "Show Name"

(2nd Way is Easy)

- 1: Go to Design Again Select button Change Onlick to blank
- 2: Go to MainActivity
- 3: Import the view

```
--> import android.view.View;
```

- 4: Write Code

Add listener to your class like ex:

```
Mbutton.setOnClickListener(new View.OnClickListener() {    --> just click Onlick  
following code will remain out automatically  
    @Override  
    public void onClick(View view) {
```

```

        MTextView.setText(R.string.Show_Text); /* For onclick Button,
Text is Show According to are need Change Text Resources To Show_text and name the
Resources*/

```

```

        MTextView.setVisibility(View.VISIBLE); /* For Visible the Text That
we invisible earlier in Design */

```

```

    }
});

```

Changes are occurred in String.Xml file is

```

-> <string name="Show_Text">Show Name</string>

```

Now go to design And Run the app And click the button it will show you the text "Show Name"

10: How We EditText and How to apply Plain text dynamically?

- 1: Go to Design choose Plain text from palette
- 2: Apply constraint and Go to Attribute
- 3: empty the text & Change the Hint according to your need like ex: Enter your name
- 4: Go to Main Activity
- 5: Reference the EditText like

```

private EditText MEditText;

```

- 6: Declare the id of EditText

```

MEditText = (EditText) findViewById(R.id.editTextTextPersonName);

```

- 7: Apply the String inside button listner event

```

Mbutton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

```

```

        String enteredtext;                                --> write the var enteredtext

```

```

        enteredtext = MEditText.getText().toString(); --> add at last for
declare it is toString() is String type

```

```

        MTextView.setText(enteredtext); --> Change the
MTextView.setText(R.string.Show_Text); To MTextView.setText(enteredtext);
        MTextView.setVisibility(View.VISIBLE);/* For Visible the Text That
we invisible earlier in Design */

```

```

    }
});

```

8: _____
Project _____



(Meter To Inches)

- 1: Drag and Drop the Button
- 2: Drag and Drop the Textview
- 3: Drag and Drop the EditText

: Now Reference them into Main Activity

```
private EditText EnterMeters;  
private Button ConvertButton;  
private TextView resultTextview;
```

: Find Id

```
EnterMeters = (EditText) findViewById(R.id.Metersid);  
resultTextview = (TextView) findViewById(R.id.Resultid);  
ConvertButton = (Button) findViewById(R.id.convertbuttonid);
```

```
ConvertButton.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
        public void onClick(View view) {  
            /* 1 meter = 39.37in */
```

```
            //Conversion Logic
```

```
            double Multiplier = 39.27;  
            double result = 0.0;
```

```
            double Metervalue = Double.parseDouble(EnterMeters.getText().toString());  
            result = Metervalue * Multiplier;
```

resultTextview.setText(Double.toString(result)); --> It will Show the result in large decimal value if u enter max number

resultTextview.setText(String.format("%.2f", result + " inches")); --> %.2f logic for show upto only 2 decimal point in result

it will show in String.format
 }

-->

3: Go to Design

4: Go to PlainText and click it and go to attribute and search input type

5: Inside Input type click check number and decimal only cuz result only accept in number and decimal

6: Now We Apply If condition for Check the Enter meter is valid number or not for this check downwards

```
        if (Entermeters.getText().equals("")){

            Entermeters.setText(R.string.error_message);
            Entermeters.setTextColor(Color.RED);

        }else{

            double Metersvalue =
            Double.parseDouble(Entermeters.getText().toString());

            result = Metersvalue*multiplier;

            resultTextView.setText(String.format("%.2f" ,result +" inches"));

        }
```

error_message --> Add to String like ex:

```
<string name="error_message">Please Enter Valid number </string>
```

7: //////////////////// Try Me App
////////////////////

1: Go to code of design

2: For Background do this

```
android:id="@+id/windowviewId" --> id for Background
```

```
3: String File <string name="app_name">Try Me!</string>
          <string name="try_me">Try me</string>
```

4: MainActivity

```
import android.util.Log;
import android.view.View;
import android.widget.Button;
```

```
import java.util.Random;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private View windowview; --> For BG view
    private Button trymebutton;
    private int[] colors; --> For Multiple Colors
```

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

```

        colors = new
int[]{Color.YELLOW,Color.GRAY,Color.RED,Color.MAGENTA,Color.BLUE,Color.CYAN};

        windowview = findViewById(R.id.windowviewId);

        trymebutton = (Button) findViewById(R.id.buttonid);

        trymebutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                //logic

                4    int ColorArraylength = colors.length; --> For Check Colors Length

                1    Random random = new Random(); --> Create Random object

                2    int randomnum = random.nextInt(ColorArraylength); --> Apply length class
here inside randomnum class

                3    windowview.setBackgroundColor(colors[randomnum]); --> Apply
randomnum inside the Windowview class

                5    Log.d("Random",String.valueOf(randomnum)); --> For Debugging code go
to Andriod monitor and check it
            }
        });
    }
}

```

8: Radio Button Workflow

- 1: Go to Design and Drag the RadioGroup
- 2: One by one Add Raddio Button inside the Radiogroup
- 3: give them name and ids

4: Coding

```

private RadioGroup radioGroup;
private RadioButton radioButton;

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

    radioGroup = (RadioGroup) findViewById(R.id.RadioGroupId);
    radioGroup.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {

```

```

@Override
public void onCheckedChanged(RadioGroup radioGroup, @IdRes int checkedID)
{ --> checkedId is combine id of button

```

```

        radioButton = (RadioButton) findViewById(checkedID);
        switch (radioButton.getId()){
            case R.id.YesId:{
                if (radioButton.isChecked()){
                    Log.d("RD", "YES!!");
                }
            }
            break;

            case R.id.NoId:{
                if (radioButton.isChecked()){
                    Log.d("RD", "NO!!");
                }
            }
            break;

            case R.id.maybeld:{
                if (radioButton.isChecked()){
                    Log.d("RD", "MAYBE!! ");
                }
            }
            break;
        }
    }
}

```

```

    });
}
}

```

9: SeekBar Work Flow

1: Go to Design Drop the Seek Bar Set the MAX size According to You like: 10

2: Go to the Activity_main.xml code and set the width of seek bar

```

Ex:<SeekBar
    android:id="@+id/SeekBarId"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:max="10"
    android:scrollbarSize="6dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

```

3: Go to Activity Main and Code

```

private SeekBar seekBar;
private TextView resulttext;

```




```

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

    seekBar = (SeekBar) findViewById(R.id.SeekBarId);

    resulttext = (TextView) findViewById(R.id.ResultId);
    seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
        @Override
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
            resulttext.setText("Hottness Level: " + seekBar.getProgress() + "/" +
seekBar.getMax()); --> Set text for result
        }
    });
--> get progress And Get Max Size

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) { --> Set Tracking Touch

        Log.d("SB","OnStartTrackingTouch"); --> Check into log its only for log
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) { --> Stop Tracking Touch
        Log.d("SB","OnStopTrackingTouch"); --> Check into log its only for log
    }
});
}

```

10: Toogle Button WorkFlow

--> Toogle Button Allows User to Change a setting between Two State On And Off.

```

private ToggleButton Toogle;
private TextView Resultview;

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

    Toogle = (ToggleButton) findViewById(R.id.toggleButtonid);
    Resultview = (TextView) findViewById(R.id.resultId);

    Toogle.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {

            if (isChecked){

                Resultview.setVisibility(View.VISIBLE);
            }else{

```

```

        Resultview.setVisibility(View.INVISIBLE);
    }

    }

});
}

```

11: Checkbox Work Flow

```

private CheckBox Mom;
private CheckBox Dad;

private Button Submit;
private TextView Result;

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

    Mom = (CheckBox) findViewById(R.id.Momid);
    Dad = (CheckBox) findViewById(R.id.Dadid);

    Result = (TextView) findViewById(R.id.Resultid);
    Submit = (Button) findViewById(R.id.buttonid);

    Submit.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            StringBuilder StringBuilder = new StringBuilder();

            Mom.isChecked() + "\n";
            Dad.isChecked() + "\n";

            StringBuilder.append(Mom.getText().toString() + "Status is:" +
            StringBuilder.append(Dad.getText().toString() + "Status is:" +
            Dad.isChecked() + "\n");

            Result.setText(StringBuilder);

        }
    });
}

```

12: Alert Dialog Workflow

1: Drag and Drop the button inside the design

2: set the Strings

```

<string name="app_name">Alert box</string>
<string name="showbtn">Show Button</string>
<string name="title">Alert Dialog</string>
<string name="message">Are you sure?</string>
<string name="yes">Yes</string>
<string name="no">No</string>

```

3: Go to main Activity

```
private AlertDialog.Builder alertDialog; --> Alert Builder inisitate
private Button Showdialog;

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

    Showdialog = (Button) findViewById(R.id.ShowBtnId);
    Showdialog.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Show the actual dialog box

            alertDialog = new AlertDialog.Builder(MainActivity.this); --> Set the context
of MainActivity

            //Set up the title

            alertDialog.setTitle(getResources().getString(R.string.title));

            //Set up the message

            alertDialog.setMessage(getResources().getString(R.string.message));

            //set the Cancelable -----> Set the alertDialog
cancel to false so that it remains in the page.

            alertDialog.setCancelable(false);

            // Construct the actual dialog box by set up the positive and negative
button

            alertDialog.setPositiveButton(getResources().getString(R.string.yes), new
DialogInterface.OnClickListener() { --> With dialogInterface
                @Override
                public void onClick(DialogInterface dialog, int which) {

                    //Exit out of the window screen

                    MainActivity.this.finish();
                }
            });

            alertDialog.setNegativeButton(getResources().getString(R.string.no), new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {

                    dialog.cancel();
                }
            });
        }
    });
}
```

```

        // Create actual dialog

        AlertDialog dialog = alertDialog.create();

        // Show the actual dialog

        dialog.show();
    }
});
}
}

```

13: Tip Calculator project Workflow

```

import android.app.Application;
import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;

import org.w3c.dom.Text;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private EditText enteredAmount;
    private SeekBar seekBar;
    private Button calculateButton;
    private TextView totalResultTextView;
    private TextView textViewSeekBar;
    private int seekbarPercentage;
    private float enteredBillFloat;
    private TextView totalBillTv;

    public static final String TAG = "MainActivity";

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

        enteredAmount = (EditText) findViewById(R.id.billAmountID);
        seekBar = (SeekBar) findViewById(R.id.percentageSeekBar);
        calculateButton = (Button) findViewById(R.id.calculateButton);
        totalResultTextView = (TextView) findViewById(R.id.resultID);
        textViewSeekBar = (TextView) findViewById(R.id.textViewSeekBar);
        totalBillTv = (TextView) findViewById(R.id.totalBillTextView);

        Toast.makeText(getApplicationContext(), "Hello", Toast.LENGTH_LONG).show();
    }
}

```

```

calculateButton.setOnClickListener(this);

seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {

        textViewSeekBar.setText(String.valueOf(seekBar.getProgress()) + "%");

    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

        seekbarPercentage = seekBar.getProgress();

    }
});

}

@Override
public void onClick(View v) {
    calculate();

    //TODO: Fix this code
    //TODO: Build amazing application with this example

}

public void calculate() {

    float result = 0.0f;

    if (!enteredAmount.getText().toString().equals("")) {

        enteredBillFloat = Float.parseFloat(enteredAmount.getText().toString());
        result = enteredBillFloat * seekbarPercentage / 100;
        totalResultTextView.setText("You tip will be " + " $" + String.valueOf(result) );
        totalBillTv.setText("Total bill: " + " $" + String.valueOf(enteredBillFloat + result));

        Log.v(TAG, String.valueOf(result));
    }
}

```

```

        }else {
            Toast.makeText(MainActivity.this, "Please enter a bill amount.",
Toast.LENGTH_LONG).show();
        }

    }
}

```

Note: Here Context is main thing

Q:what is Context class?

It's an abstract class whose implementation is provided by the Android system. Context allows access to application-specific resources and classes, as well as calls for application-level operations such as launching activities, broadcasting and receiving intents, etc.

Q:What is Abstract class?

Abstract is Model class That allow use the functionality of other class!

Q: What is Toast class in andriod?

```

Toast.makeText(MainActivity.this, "Please enter a bill amount.",
Toast.LENGTH_LONG).show();

```

-->Here this declare the whole activity of project

```

Toast.makeText(getApplicationContext(), "Hello", Toast.LENGTH_LONG).show();

```

-->Here We Get and access entire Application of our project

```

Toast.makeText(getBaseContext(), "Hello", Toast.LENGTH_LONG).show();

```

-->Here We Get and access entire Source file of our project

Q: What is log, How it is work?

log is used to show our result in log in Andriod monitor

v() method is used to log verbose messages. --> Verbose
d() method is used to log debug messages. --> Debug
i() method is used to log informational messages. --> Info
w() method is used to log warnings.--> Warn

e() method is used to log errors. --> error

```
Log.v(TAG, String.valueOf(result));
```

--> see the result in the log

```
Log.v(TAG, MainActivity.this.String.valueOf(result));
```

--> see the String result of MainActivity

```
log.d(TAG,"Hello From Main Activity");
```

Note: Why we use TAG?

Ans: if we want to print or debugging our result of MainActivity in console we use TAG

1: For using TAG we first initiate our TAG Method class

2: private static final string TAG = "Main Activity";

Q: What is activity?

Activity is something that we shown to users

like example of Activity class.

1:

```
import android.app.Activity;
```

```
public class MainActivity extends Activity{ --> Here we extends our Activitiy from our sources
```

@Override -----> Here we done override method in which we create method oncreate to initiate out MainActivity

```
protected void onCreate(Bundle savedInstanceState) { --> here savedInstanceState that save the all instances
```

```
super.onCreate(savedInstanceState); --> we call the method by passing instance state
```

```
setContentView(R.layout.activity_main); --> We set our XML files resources here
```

2:

```
import androidx.appcompat.app.AppCompatActivity; --> Here we import
```

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{ --> Here we extend ou AppCompatActivity and implements on View.onclicklistner
```

Q: what are the type of override methods?

1: right click and go to generate

2: inside generate go to override methods and choose one

14: Activity life Cycle

1:

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toast.makeText(getApplicationContext(),"OnCreate
called",Toast.LENGTH_LONG).show();

        -----> on create with Toast method with having context
        -----> Having message of oncreate called
        -----> Toast.LENGTH_LONG is used for message duration
        -----> and >show() is used for showing the message

    }
```

```
    @Override
    protected void onStart() { -----> Onstart override method used
        super.onStart();
        Toast.makeText(getApplicationContext(),"OnStart
called",Toast.LENGTH_LONG).show();
    }
```

```
    @Override
    protected void onResume() { -----> On postResume method used
        super.onResume();
        Toast.makeText(MainActivity.this,"OnPostResume
called",Toast.LENGTH_LONG).show();
    }
```

```
    @Override
    protected void onStop() { -----> onStop method used
        super.onStop();
        Toast.makeText(MainActivity.this,"OnStop called",Toast.LENGTH_LONG).show();
    }
```

```
    @Override
    protected void onDestroy() { -----> onDestroy method used
        super.onDestroy();
        Toast.makeText(MainActivity.this,"OnDestroy called",Toast.LENGTH_LONG).show();
    }
```



```

@Override
protected void onPause() { -----> onPause Method used
    super.onPause();
    Toast.makeText(MainActivity.this,"OnPause called",Toast.LENGTH_LONG).show();
}
}

```

2: (Activity Naviagte to another Activity)

1: create first activity project

2: create button

3: go inside java file go to project Activity flow

4: click right click go to new

5: go to Activity and select empty activity

```

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

```

```

public class MainActivity extends AppCompatActivity {

```

```

    private Button ShowActivitybutton;

```

```

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

```

```

    ShowActivitybutton = (Button) findViewById(R.id.ShowButtonId);
    ShowActivitybutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            --> code here to move to next activity
            --> Firstly we create an intent class constructor
            --> intent class is used to move from one activity to another

```

```

//First way
Intent intent = new Intent(MainActivity.this, MainActivity2.class);

```

```

----->Here we are at the first activity
-----> And our goal is to go to next activity show we pass

```

MainActivity@.class

```

startActivity(intent);---->Here we Start our activity

```

```

//Second Way

```

```

startActivity(new Intent(MainActivity.this,MainActivity2.class));

```

```

    });
}
}
}

```

3; (Passing the Message from First Activity to second Activity by Key and Value

1: MainActivity

```

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private Button ShowActivitybutton;
    private final int REQUEST_CODE = 2; -----> Request code for Second Activity

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

        ShowActivitybutton = (Button) findViewById(R.id.ShowButtonId);
        ShowActivitybutton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                // code here to move to next activity
                // Firstly we create an intent class constructor
                // intent class is used to move from one activity to another

                Intent intent = new Intent(MainActivity.this, MainActivity2.class); // Here we are at the first
                activity
                // // And our goal is to go to next activity show we pass
                MainActivity@.class

                intent.putExtra("Message","Hello");-->passing key & value inside the Extra /
                Put Extra is method of passing the message
                intent.putExtra("value",123); // passing int value
                // startActivity(intent); //Here we Start our activity

                /// Forget the Start Activity(intent)
                /// And Write
            }
        });
    }
}

```

startActivityForResult(intent, REQUEST_CODE);-->For set the intent as well
get the intent from activity via Request code

```
// To return the result of second activity to here
// First we override the parent method on create with
// On Activity result
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    //Match the parameters by If condition

    if (requestCode == REQUEST_CODE){

        if (resultCode == RESULT_OK){

            String result = data.getStringExtra("returnData"); --> Here We passing the
            key of second activity

            Toast.makeText(MainActivity.this, result,Toast.LENGTH_LONG ).show();
        }
    }
}
```

2: MainActivity 2

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
```

```
public class MainActivity2 extends AppCompatActivity {
```

```
    private TextView showMessage;
    private Button ResultBackBtn;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);

    showMessage = (TextView) findViewById(R.id.ShowActID);
```

Bundle extras = getIntent().getExtras();--> Here We get our Extras that we set on first
activity and Get our extras here

```

// Checks

if (extras != null){
    String message = extras.getString("Message");---->Here we pass our first
activity key inside string
    ---->Now we Set the get the integer value cuz our value of 2 extra is integer
    ---->create a int class

    int myInt = extras.getInt("value"); ---> here pass key inside the parameter

    showMessage.setText("Messsage is: " + message + "value is:" +
String.valueOf(myInt));

}

// Getting Back Result

ResultBackBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent returnIntent = getIntent();
        returnIntent.putExtra("returnData", "From Second Activity"); --> Key Value
        setResult(RESULT_OK,returnIntent);
        finish();
    }
});
}
}
}

```

15: (Pet Bio App)

- 1: Set the 2 icon in drawble
- 2: Set 2 Image view in design
- 3: MainActivity

```

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.media.Image;
import android.os.Bundle;
import android.view.View;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

    private ImageView dogview;
    private ImageView catview;

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

```

```

        dogview = (ImageView) findViewById(R.id.dogId);
        catview = (ImageView) findViewById(R.id.CatId);

        dogview.setOnClickListener(this);
        catview.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        switch (view.getId()){

            case R.id.CatId:
                //Go to second activity
                Intent catintent = new Intent(MainActivity.this, bioActivity.class);
                catintent.putExtra("name","Jarvis");
                catintent.putExtra("bio","She loves everyone. She Meow Also");
                startActivity(catintent);
                //Toast.makeText(MainActivity.this,"Cat",Toast.LENGTH_LONG).show();
                break;

            case R.id.dogId:
                // Go to second activity

                Intent dogintent = new Intent(MainActivity.this, bioActivity.class);
                dogintent.putExtra("name","Gafur");
                dogintent.putExtra("bio","He loves everyone. He Bark Also");
                startActivity(dogintent);

                Toast.makeText(MainActivity.this,"Dog",Toast.LENGTH_LONG).show();
                break;

        }
    }
}

```

1: Bioactivity

- 1: set the imageview and set the background transparent
- 2: set 2 textviw one for name and another for bio

3: Bio Activity

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompatExtras;

import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;

public class bioActivity extends AppCompatActivity {

    private ImageView petimage;

```

```

private TextView nameview;
private TextView bioview;
private Bundle extras;

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

    petimage = (ImageView) findViewById(R.id.petimageid);
    nameview = (TextView) findViewById(R.id.nameid);
    bioview = (TextView) findViewById(R.id.bioid);

    extras = getIntent().getExtras();

    if (extras != null){
        String name = extras.getString("name");
        String bio = extras.getString("bio");

        setUp(name,bio);
    }
}

public void setUp(String name, String bio){

    if (name.equals("Gafur")){
        // show dog result

        petimage.setImageDrawable(getResources().getDrawable(R.drawable.icon_lg_dog));
        nameview.setText(name);
        bioview.setText(bio);

    }else if (name.equals("Jarvis")){
        //show cat result

        petimage.setImageDrawable(getResources().getDrawable(R.drawable.icon_lg_cat));
        nameview.setText(name);
        bioview.setText(bio);

    }

}
}

```

16: Recyler Views (List view)

- 1: Go to Design Select the Recylerview
- 2: libraries are set in Gradle Scripts after allow Reclview
- 3: create a sperate list_row xml file
- 4: inside sperate list xml file go to code and set the relative layout instead of linear
- 5: Go to design of MainAcitivity xml allow the cardview button
- 6: set the card view button inside the Relative layout in component tree
- 7: Now search the LinearLayout vertically
- 8: put it inside the component tree inside cardview
- 9: Go to app inside java add the 3 packages



10: Adapter,Model,Util
11: inside the Adapter
12: Add the java class name MyAdapter

```
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.ListAdapter;
import androidx.recyclerview.widget.RecyclerView;

import java.util.List;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {

    private Context context;
    private List<ListItem> listItems;

    private MyAdapter(Context context, List listitem){

    }

    @NonNull
    @Override
    public MyAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(@NonNull MyAdapter.ViewHolder holder, int position) {

    }

    @Override
    public int getItemCount() {
        return 0;
    }
}
```

13: Inside the Model set the java class name ListItem.java to show the list

```
package Model;

public class ListItem {

    private String name;
    private String description;

    // Add Constructor just click generate and add constructo

    public ListItem( String name, String description) { --> pass the both the String parameter
        this.name = name;
        this.description = description;
    }

    // Getter Setter Method of name description just click and generate and inside generate
getter setter option

    public String getName() {
```

```

        return name;
    }

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

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }
}

```

14: Go to Adapter and import the ListItem model

15: Now add the viewholder class

16: extends the View RecyclerView.ViewHolder

17: Add the Their Matching constructor

18: Fix the ViewHolder By Setting the Ids

```

public class ViewHolder extends RecyclerView.ViewHolder{
    private TextView name;
    private TextView description;

    public ViewHolder(@NonNull View itemView) {
        super(itemView);

        name = (TextView) itemView.findViewById(R.id.title);    --> itemView is
reference from ViewHolder
        description =(TextView) itemView.findViewById(R.id.description);
    }
}

```

16: Add the constructor filed in upside

```

private MyAdapter(Context context, List listitem){
    this.context = context;
    this.listItems = listitem;
}

```

15: Q What is inflating?

--> convert the xml file into something that is visible the xml content

16: inside the ViewHolder

Here Our Inflater class is used for Layout

@NonNull

@Override

```

public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
    View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.list_row,
parent,false) --> We are Outside our Main Activity So first we import our R
    return new ViewHolder(view);
}

```



import---->import com.example.recyclerview.R;

17: Now we bind our ViewHolder to Adapter for Showing inside Recycler view\

```
@Override
    public void onBindViewHolder(@NonNull MyAdapter.ViewHolder holder, int position) {

        --> We already Instantiate our ViewHolder class at bottom, we access ViewHolder
        widget here

        holder.name.setText("Hello");
        holder.description.setText("Description");

    }
```

```
18:  @Override
    public int getItemCount() {
        return listItems.size(); -->return the size of ListItems
    }
```

19: Now We initiate Our Adapter in MainActivity

```
package com.example.recyclerview;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import java.util.ArrayList;
import java.util.List;

import Adapter.MyAdapter;
import Model.ListItem;

public class MainActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    private RecyclerView.Adapter adapter;
    private List<ListItem> listItems;

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

        recyclerView = (RecyclerView) findViewById(R.id.recyclerview);
        recyclerView.setHasFixedSize(true);
        //every item has a fixed size
        recyclerView.setLayoutManager(new
            LinearLayoutManager(this));

        listItems = new ArrayList<>();
```

```

        for (int i = 0; i<10; i++) {
            ListItem listItem = new ListItem(
                "Item " + (i+1),
                "Description"
            );
            listItems.add(listItem);
        }

        adapter = new MyAdapter(this, listItems);

        recyclerView.setAdapter(adapter);
    }
}

```

20: Adapter

```

package Adapter;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import androidx.recyclerview.widget.RecyclerView;

import com.example.recyclerview.R;

import java.util.List;

import Model.ListItem;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder> {

    private Context context;
    private List<ListItem> listItems;

    public MyAdapter(Context context, List listItem){
        this.context = context;
        this.listItems = listItem;
    }

    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View v = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.list_row, parent, false);

        return new ViewHolder(v);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        ListItem listItem = listItems.get(position);
        holder.name.setText(listItem.getName());
        holder.description.setText(listItem.getDescription());
    }
}

```

```

    }

    @Override
    public int getItemCount() {
        return listItems.size(); // return the size of ListItems
    }

    // Create the viewHolder class
    // Able to fetch textView name textView Description
    public class ViewHolder extends RecyclerView.ViewHolder{
        private TextView name;
        private TextView description;

        public ViewHolder(View itemView) {
            super(itemView);

            name = (TextView) itemView.findViewById(R.id.title);
            description =(TextView) itemView.findViewById(R.id.description);

        }
    }
}

```

21 ListItem

```

package Model;

public class ListItem {

    private String name;
    private String description;

    // Add Constructor

    public ListItem( String name, String description) {
        this.name = name;
        this.description = description;
    }

    // Getter Setter Method

    public String getName() {
        return name;
    }

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

    public String getDescription() {
        return description;
    }
}

```

```

public void setDescription(String description) {
    this.description = description;
}

```

22: go to list_row.xml file and set

```

<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"> -----> set the Wrap_content

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_margin="@dimen/cardview_compat_inset_shadow" --> set the Layout
margin
        android:layout_height="wrap_content" >

```

23: Add Another Textview into the list_row

```

<TextView
    android:id="@+id/rating"
    android:text="Great"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"/>

```

24: Go to the Listitem.java and inistiate the Textview

Add the Textview into Constructor also
Set the Getter and Setter method

```

private String rating;

// Add Constructor

public ListItem(String name, String description, String rating) {
    this.name = name;
    this.description = description;
    this.rating = rating;
}

// Getter Setter Method

public String getRating() {
    return rating;
}

public void setRating(String rating) {
    this.rating = rating;
}

```

25: Go to My adapter.java

---> Intitiate the Same Listview inside the Viewholder class
---> Get them id

```

public class ViewHolder extends RecyclerView.ViewHolder{
    private TextView name;
    private TextView description;
}

```

```

private TextView rating;    --> This one is the new ListView

public ViewHolder(View itemView) {
    super(itemView);

    name = (TextView) itemView.findViewById(R.id.title);
    description = (TextView) itemView.findViewById(R.id.description);
    rating = (TextView) itemView.findViewById(R.id.rating);
}

}

```

--> Bind the TextView into the OnBindViewHolder

```

@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    ListItem listItem = listItems.get(position);
    holder.name.setText(listItem.getName());
    holder.description.setText(listItem.getDescription());
    holder.rating.setText(listItem.getRating());
}

```

26: Go to the MainActivity.java

--> Add the View into the loop so that it will show on App

```

for (int i = 0; i < 10; i++) {
    ListItem listItem = new ListItem(
        "Item " + (i+1),
        "Description",
        "Excellent" --> Here it is We change our text name Excellent instead of
Great!
    );
    listItems.add(listItem);
}

```

27: Suppose We dont want many listitem

--> We Delete the loop and one by one add the Listitem

--> Firstly we create the Listmethod inside the MainActivity.java

```
listItems = new ArrayList<>();
```

ListItem item1 = new ListItem("Movie1","About Love Story","Awesome!"); --> Add the Parameters that you want to Show

```

ListItem item2 = new ListItem("Movie2","About Action","Not So Good!");
ListItem item3 = new ListItem("Movie3","About Horror","Amazing!");

```

--> Now Add the listItems

```

listItems.add(item1); // Show the list Without Loop
listItems.add(item2);
listItems.add(item3);

```

28: Now Adding Event Listener to each row

--> Go to Adapter.java

--> Implements Event ViewHolder by Adding Event Listener and Implements the method
 --> After that Set on the clicklistener to the itemView and pass the parameter this
 --> Inside the OnClick method
 --> Add the position
 --> Add your position into your listItem method
 --> Select on position and set the output by Toast

```

public class ViewHolder extends RecyclerView.ViewHolder implements
View.OnClickListener{
    private TextView name;
    private TextView description;
    private TextView rating;

    public ViewHolder(View itemView) {
        super(itemView);

        itemView.setOnClickListener(this); //Set the onclick listener by adding the
context this

        name = (TextView) itemView.findViewById(R.id.title);
        description =(TextView) itemView.findViewById(R.id.description);
        rating = (TextView) itemView.findViewById(R.id.rating);
    }

    @Override
    public void onClick(View view) {

        // Get the position of the each view or clicked position

        int position = getAdapterPosition();

        ListItem item = listItems.get(position);

        Toast.makeText(context,item.getName(),Toast.LENGTH_LONG).show();
    }
}

```

29: Now We want to Show our list result to the another activity

----> Create the empty Activity inside the java com file named as Detailsactivity.java
 ----> Set the 3 Textview
 ----> go to the adapter.java
 ----> Add the Intent class inside the viewholder
 ----> set the key and value
 ----> go to the DetailsActivity.java
 ----> initialize the class and get the id
 ----> set the bundle
 ----> set the list inside the if condition

Adapter

```

@Override
public void onClick(View view) {

    // Get the position of the each view or clicked position

```

```

        int position = getAdapterPosition();

        ListItem item = listItems.get(position);

        // Show Our List to another activity, We Create the Intent Class

        Intent intent = new Intent(context, DetailActivity.class);    // Here We pass
context cuz we already initialized upwards

        intent.putExtra("name",item.getName());
        intent.putExtra("description",item.getDescription());
        intent.putExtra("rating",item.getRating());

        context.startActivity(intent);    // here we use context

```

DetailActivity

```

import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class DetailActivity extends AppCompatActivity {

    private TextView name,Description,rating;
    private Bundle extras;

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

        name = (TextView) findViewById(R.id.DnameId);
        Description = (TextView) findViewById(R.id.DdescriptionId);
        rating = (TextView) findViewById(R.id.DratingsId);

        extras = getIntent().getExtras();

        if (extras != null){

            name.setText(extras.getString("name"));
            Description.setText(extras.getString("description"));
            rating.setText(extras.getString("rating"));
        }
    }
}

```

17: (Working of Relative Layout)

- 1: Suppose We have two Button name login and enter
- 2: Thier id is Button 5 and Button 4
- 3: As we add Relative layout two button are bind together that means showing only one



button instead of two

4: To sperate the both button and set side by side we look below

```
<RelativeLayout xmlns:android=""
    xmlns:tools=""
    android:layout_width="match_parent"
    android:layout_height="match_parent"

    <Button
        android:id="@+id/button5"
        android:layout_centerVertical="True" ---> first we set our button to vertical
        android:layout_centerHorizontal="True"---> Then we set our button to Horizontal
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter" />

    <Button
        android:id="@+id/button4"
        android:layout_toLeftof="@+id/button5" ---> that means our login button is sperate
and set left to Enter button
        android:layout_alignBottom="@+id/button5"---> Now our button is set to alignBottom
to the Enterbutton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login" />

</RelativeLayout>
```

5: If we add textview and want to set it on bottom of the parent element Button

---> android:layout_alignBottom="@+id/button5" ----> Use this

18: (Table Layout)

- > Create a Project Table Layout
- > give root to Relative Layout
- > go to design
- > Drag TableLayout into Component Tree inside the Relative Layout
- > Drag a Table Row inside the TableLayout into the Component Tree

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableLayout
        android:padding="15dp" ---> Set the local padding
        android:layout_width="match_parent"
        android:layout_height="match_parent">

        <TableRow
            android:background="@color/colorAccent"
            android:layout_width="match_parent"
            android:layout_height="match_parent">
```



```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorPrimaryDark"
    android:text="FirstName" /> ----> Set the Header name
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="20dp" ----> Set the Header Gap
    android:textColor="@color/colorPrimaryDark"
    android:text="Lastname" /> ----> Set the Header name
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="20dp"
    android:textColor="@color/colorPrimaryDark"
    android:text="Age" /> ----> Set the Header name
```

```
</TableRow>
```

```
<!--Next Table Row -->
```

```
<TableRow
    android:background="@color/colorAccent"
    android:layout_marginTop="20dp" ----> Set the padding from the uper row
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textColor="@color/colorPrimaryDark"
    android:text="Hrithik" /> ----> Set the value name
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingLeft="20dp"
    android:textColor="@color/colorPrimaryDark"
    android:text="Patel" /> ---.Set the value name
```

```
<TextView
    android:layout_width="wrap_content"
```


--> Go to Syles.xml
--> Add Style class

```
<style name="CustomeStyle">

    <item name="android:textSize">14dp</item>
    <item name="fontFamily">sans-serif-condensed-medium</item>
    <item name="android:textColor">#E91E63</item>
</style>
```

--> put the style class into textview Style so that u can get your style result

2: Style the Appptheme

--> Appthema has a allready three primary colour
--> Declared at the Minifiestfile

----> Base App theme in styles.xml file

```
<!-- Base application theme. -->
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>
```

----> Theme color is allready declared in Colors.xml

```
<resources>
    <color name="colorPrimary">#6200EE</color>
    <color name="colorPrimaryDark">#3700B3</color>
    <color name="colorAccent">#03DAC5</color>
</resources>
```

--> These by default theme Style is declared in Minifiest file

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme"> ----> Here Style Theme is Declared
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

3: You can change your App theme Acording to your choice

---> Add your style in Style.xml

```
<style name="AppTheme2" parent="Theme.AppCompat.DayNight.DarkActionBar">
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
</style>
```

---> Change the color in colors.xml

```
<color name="colorPrimary">#1E1D1E</color>
<color name="colorPrimaryDark">#302B3A</color>
<color name="colorAccent">#707373</color>
```

---> Change the Theme name in Manifest.xml file

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme2"> ---> Style Theme is Change
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

Note: Go to developer.android.com for more theme and colors and layout

21: (Animations Frame)

--> Here We doing Animation Frames

--> Animation Frame of Bat

--> There Are 3 images and one animations Xml put in the drawable

bat_Animation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" ----> We use
animation-list for animations frame
    android:oneshot="false"> -----> We set the oneshot false
    <item android:drawable="@drawable/bat_03" android:duration="80" />-----> Set the
```

duration in milisecond for each image

```
<item android:drawable="@drawable/bat_05" android:duration="80" />-----> same
<item android:drawable="@drawable/bat_07" android:duration="80" />-----> same
<!-- <item android:drawable="@drawable/bat_09" android:duration="80" />-->
<!-->
</animation-list>
```

MainActiviy.xml

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.graphics.drawable.AnimationDrawable;
```

```
import android.os.Bundle;
```

```
import android.view.MotionEvent;
```

```
import android.widget.ImageView;
```

```
public class MainActivity extends AppCompatActivity {
    private AnimationDrawable batAnimation; -----> set the reference of AnimationDrawable
    private ImageView batImage; ----> Set the Imageview
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
        batImage = (ImageView) findViewById(R.id.imageID);
```

```
        batImage.setBackgroundResource(R.drawable.bat_anim); ----> set the
bat_animation.xml resorce in Imageview as background
```

```
        batAnimation = (AnimationDrawable) batImage.getBackground(); --> Now get the
background in AnimationDrawable
```

```
    }
```

```
    @Override
```

```
    public boolean onTouchEvent(MotionEvent event) { ----> set the override method
onTouchEvent for this Animation frame
```

```
        batAnimation.start(); ----> now start the animation
```

```
        return super.onTouchEvent(event);
```

```
    }
```

```
}
```

-----> Save and see the output itworks but image 1 is still stuck

-----> For solution

-----> go to mainActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<androidx.constraintlayout.widget.ConstraintLayout
```

```
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">

<ImageView
    android:id="@+id/imageID"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:srcCompat="@drawable/bat_03" />-----> Erase
this line srcCompat
</androidx.constraintlayout.widget.ConstraintLayout>

```

2: Now we set the animation for particular second and stop it

----> set the handler class

```

@Override
public boolean onTouchEvent(MotionEvent event) {

    batAnimation.start();

    Handler mHandler = new Handler();
    mHandler.postDelayed(new Runnable() { ----> Set the parameter Runnable
        @Override
        public void run() {
            //stop the animation
            batAnimation.stop();
        }
    }, 5000); // 5seconds ----> set the animation at 5000milisecond
    return super.onTouchEvent(event);
}
}

```

3: For Fading Animation

----> go to resources and right click add the new resources directory
 ----> name anim
 ----> inside anim set the fadein_animation.xml

show the udemy lesson 105 section 16

23:(Andriod Media Player)

----> go to resources and right click add the new resources directory
 ----> name raw value raw
 ----> inside raw set the Add tunpocket.mp3 file 10 sec music mp3

----> Add the button to desing give to name play and id u like it

MainActivity.java

```
package com.example.andriodmediaplayer;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
import android.content.DialogInterface;
```

```
import android.media.MediaPlayer;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
public class MainActivity extends AppCompatActivity {  
    private MediaPlayer mediaPlayer; ----> Instanticate the Mediaplayer  
    private Button playButton;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
        mediaPlayer = new MediaPlayer(); ----> create mediaplyaer Object
```

```
        mediaPlayer = MediaPlayer.create(getApplicationContext(), R.raw.tunepocket); ---->  
set the mp3 file
```

```
        playButton = (Button) findViewById(R.id.btnId);
```

```
        playButton.setOnClickListener(new View.OnClickListener() { ----> create onclick  
listener
```

```
            @Override
```

```
            public void onClick(View view) {
```

```
                if(mediaPlayer.isPlaying()){ ----> media plyar is playing ----> for start  
and stop the music we create methods for it
```

```
                // Stop and give option to start play
```

```
                pauseMusic(); ----> set the void method here
```

```
            }else{
```

```
                startMusic(); ----> set the void method here
```

```
            }
```

```
        }
```

```
    });
```

```
}
```

```
    public void pauseMusic(){
```

```
-----> methods for pause the music
```

```
        if (mediaPlayer != null){
```

```
            mediaPlayer.pause();
```

```
            playButton.setText("Play"); ----> set the button text when music is  
pause to play text
```

```
        }
```

```
    }
```

```
    public void startMusic(){
```

```
-----> methods for start the music
```

```
        if(mediaPlayer != null){
```

```
            mediaPlayer.start();
```

```
            playButton.setText("Stop"); ----> set the button text when music is  
start to play stop
```

```
        }
```

```
    }
```

```
}
```



-----> if music is not working download the plugins

2: (Set the duration of music)

--> set the onclick duration method
--> create int
--> set the string
--> apply toast method

```
mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {  
    @Override  
    public void onCompletion(MediaPlayer mediaPlayer) {  
        int duration = mediaPlayer.getDuration(); --> set the int  
        String mDuration = String.valueOf(duration/1000); --> Set the duration in  
string // it is logic duration/1000 gives Ans in sec  
  
        Toast.makeText(getApplicationContext(),"duration" + mDuration,  
Toast.LENGTH_LONG).show();  
    }  
});
```

3: (With OnDestroy method)

```
package com.example.andriodmediaplayer;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.BroadcastReceiver;  
import android.content.Intent;  
import android.media.MediaPlayer;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
public class MainActivity extends AppCompatActivity {  
    private MediaPlayer mediaPlayer;  
    private Button playButton;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        mediaPlayer = new MediaPlayer();  
        mediaPlayer = MediaPlayer.create(getApplicationContext(), R.raw.tunepocket);  
  
        mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {  
            @Override  
            public void onCompletion(MediaPlayer mediaPlayer) {  
                int duration = mediaPlayer.getDuration();  
                String mDuration = String.valueOf(duration/1000);  
  
                Toast.makeText(getApplicationContext(),"duration" + mDuration,  
Toast.LENGTH_LONG).show();  
            }  
        });  
    }  
}
```



```

    });

    playButton = (Button) findViewById(R.id.btnId);
    playButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(mediaPlayer.isPlaying()){
                // Stop and give option to start play
                pauseMusic();
            }else{
                startMusic();
            }
        }
    });
}

public void pauseMusic(){
    if (mediaPlayer != null){
        mediaPlayer.pause();
        playButton.setText("Play");
    }
}

public void startMusic(){
    if(mediaPlayer != null){
        mediaPlayer.start();
        playButton.setText("Stop");
    }
}

@Override
protected void onDestroy() { ---> Create on Destroy method

    if (mediaPlayer != null && mediaPlayer.isPlaying()){ ---> check the condition
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
    super.onDestroy();
}
}

```

4: (Set the Seek Bar)

```

package com.example.andriodmediaplayer;

import androidx.appcompat.app.AppCompatActivity;

import android.content.DialogInterface;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.SeekBar;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private MediaPlayer mediaPlayer;
    private Button playButton;

```

```

private SeekBar mseekbar; -----> instantiate Seekbar
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    mediaPlayer = new MediaPlayer();
    mediaPlayer = MediaPlayer.create(getApplicationContext(), R.raw.tunepocket);

    mseekbar = (SeekBar) findViewById(R.id.mseekbar); --> get id
    mseekbar.setMax(mediaPlayer.getDuration()); --> Set the Max duration

    mseekbar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener()
{ -----> Set onseekBarChangeListener method
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
        if(fromUser){
            mediaPlayer.seekTo(progress);    -- > fromUser parameter , Click by
User and passs the pass progress in seekTo method

        }
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});

mediaPlayer.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {
    @Override
    public void onCompletion(MediaPlayer mediaPlayer) {
        int duration = mediaPlayer.getDuration();
        String mDuration = String.valueOf(duration/1000);    // it is logic
duration/1000 gives Ans in sec

        Toast.makeText(getApplicationContext(),"duration" + mDuration,
Toast.LENGTH_LONG).show();
    }
});

playButton = (Button) findViewById(R.id.btnId);
playButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(mediaPlayer.isPlaying()){
            // Stop and give option to start play
            pauseMusic();
        }else{
            startMusic();
        }
    }
}

```

```

    });
}

public void pauseMusic(){
    if (mediaPlayer != null){
        mediaPlayer.pause();
        playButton.setText("Play");
    }
}

public void startMusic(){
    if(mediaPlayer != null){
        mediaPlayer.start();
        playButton.setText("Stop");
    }
}

@Override
protected void onDestroy() {

    if (mediaPlayer != null && mediaPlayer.isPlaying()){
        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
    super.onDestroy();
}
}

```

24: (Music Box App)

- > Set the raw file set the mp3 inside it
- > Set the imageview
- > Set the oval.xml file inside the drawable resources

oval.xml

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android" --> Set the shape
    android:shape="oval">
    <solid
        android:color="@color/dark" --> set the color
    />
</shape>

```

- > change the color in colors.xml

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#FF9800</color>
    <color name="colorPrimaryDark">#ECCD73</color>
    <color name="colorAccent">#FB6598</color>
    <color name="dark">#3C3A38</color>
</resources>

```

----> Go to the code of MainActivity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.138"
        app:srcCompat="@drawable/oval" /> -----> Change is here
set the drawble
</androidx.constraintlayout.widget.ConstraintLayout>
```

----> Change the drawble according to your need

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="oval">
    <stroke -----> i use stroke for border

        android:dashWidth="2dp" ----> use for Border Width
        android:color="@color/dark"
        android:Width="3dp" ----> set the width

    />
</shape>
```

-----> Set the Divider imageview

-----> set the new Drawble resources name divider

(divider.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">

    <solid
        android:color="@color/colorPrimary"
    />

</shape>
```

----> set the textview named Song name
 ----> set the textview named artist name
 ----> Set the SeekBar
 ----> set the play time 0.00
 ----> set the play time 4.00

----> set the tablerow
 ----> inside the table add 3 button

```

<TableRow
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="@color/colorPrimary" -----> set color
    android:padding="14dp" -----> set padding
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/seekBarId"
    app:layout_constraintVertical_bias="0.219">

    <Button
        android:id="@+id/prevbutton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@android:drawable/ic_media_previous" /> ----> set the
icon previous

    <Button
        android:id="@+id/playbutton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@android:drawable/ic_media_play" /> ---> set the icon
play

    <Button
        android:id="@+id/nextbutton"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:background="@android:drawable/ic_media_next" /> --> set the icon
next

</TableRow>

```

----> Note: Change the Sdk min max according to your application
 ----> for change the sdk go to gradle script build.module app
 ----> android {
 compileSdkVersion 30
 buildToolsVersion "30.0.0"

 defaultConfig {
 applicationId "com.example.musicboxapp"
 minSdkVersion 21 -----> Change acc to your app need

```

        targetSdkVersion 30 -----> Max
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

```

(MainActivity.java)

```

package com.example.musicboxapp;

import android.media.MediaPlayer;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import java.text.SimpleDateFormat;
import java.util.Date;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {
    private MediaPlayer mediaPlayer;
    private ImageView artistImage;
    private TextView leftTime;
    private TextView rightTime;
    private SeekBar seekBar;
    private Button prevButton;
    private Button playButton;
    private Button nextButton;
    private Thread thread;

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

        setUpUI(); // here we declare the class setUpUI that we created in downwords

        seekBar.setMax(mediaPlayer.getDuration());
        seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean
fromUser) {
                if (fromUser) {
                    mediaPlayer.seekTo(progress);
                }

                SimpleDateFormat dateFormat = new SimpleDateFormat("mm:ss"); // data
class is created
                int currentPos = mediaPlayer.getCurrentPosition(); // CurrentPos int to get
the position
            }
        });
    }
}

```

```

        int duration = mediaPlayer.getDuration(); //duration int to get the duration

        leftTime.setText(dateFormat.format(new Date(currentPos))); // new obj
        Date which convert the position into mm:ss formate
        rightTime.setText(dateFormat.format(new Date(duration - currentPos))); //
        same here but duration is minus form currentPos
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }

    });
}

```

```

public void setUpUI() {

```

```

    mediaPlayer = new MediaPlayer();
    mediaPlayer = MediaPlayer.create(getApplicationContext(),R.raw.tunepocket);

```

```

    artistImage = (ImageView) findViewById(R.id.imageView);
    leftTime = (TextView) findViewById(R.id.timeId);
    rightTime = (TextView) findViewById(R.id.time2Id);
    seekBar = (SeekBar) findViewById(R.id.seekBarId);
    prevButton = (Button) findViewById(R.id.prevbutton);
    playButton = (Button) findViewById(R.id.playbutton);
    nextButton = (Button) findViewById(R.id.nextbutton);

```

```

    prevButton.setOnClickListener(this);
    playButton.setOnClickListener(this);
    nextButton.setOnClickListener(this);

```

```

}

```

```

@Override

```

```

public void onClick(View view) {

```

```

    switch (view.getId()){

```

```

        case R.id.prevbutton:

```

```

            // code

```

```

            BackMusic(); // class that prev the music that we created downwards
            break;

```

```

        case R.id.playbutton:

```

```

            //code

```

```

            if (mediaPlayer.isPlaying()){

```

```

                pauseMusic();

```

```

            }else {

```

```

                startMusic();

```

```

            }

```

```

            break;

```

```

        case R.id.nextbutton:

```



```

        //code

        NextMusic(); // Class that next music that we created downwards
        break;
    }

}

//PauseMusic
public void pauseMusic(){
    if (mediaPlayer != null){
        mediaPlayer.pause();
        playButton.setBackgroundResource(R.drawable.);
    }
}

// startMusic
public void startMusic(){
    if (mediaPlayer != null){
        mediaPlayer.start();
        updateThread();
        playButton.setBackgroundResource(R.drawable.);
    }
}

// BackMusic

public void BackMusic(){
    if (mediaPlayer.isPlaying()){
        mediaPlayer.seekTo(0);
    }
}

// NextMusic

public void NextMusic(){
    if (mediaPlayer.isPlaying()){

        mediaPlayer.seekTo(mediaPlayer.getDuration() - 1000);
        // -----> As we prev Music it will back to 0:00
        // side 0:00 at last that why (-1000)
    }
}

```

```

// Update the music second as we move further automatically
public void updateThread(){
    thread = new Thread() {

        @Override
        public void run() {
            try {
                while (mediaPlayer != null && mediaPlayer.isPlaying()) {

                    Thread.sleep(50);
                    // -----> upto 50 milisecond
                    runOnUiThread(new Runnable() {
                        // ----> using Runnable

```

method


```

        @Override
        public void run() {

            int newPosition = mediaPlayer.getCurrentPosition();    -->
declare newpositon
            int newMax = mediaPlayer.getDuration();    --> declare
duration
            seekBar.setMax(newMax);
            seekBar.setProgress(newPosition);

            //update the text
            leftTime.setText(String.valueOf(new
SimpleDateFormat("mm:ss")
                .format(new Date(mediaPlayer.getCurrentPosition()))));

            rightTime.setText(String.valueOf(new
SimpleDateFormat("mm:ss")
                .format(new
Date(mediaPlayer.getDuration()))));

        }
    });
} catch (InterruptedException e) {
    e.printStackTrace();
}
}

};
thread.start(); -----> here we declare the thread Start()
}

@Override
protected void onDestroy() {

    if (mediaPlayer != null && mediaPlayer.isPlaying()){

        mediaPlayer.stop();
        mediaPlayer.release();
        mediaPlayer = null;
    }
    thread.interrupt();
    thread = null;

    super.onDestroy();
}
}

```

25:(Pasing and get the data back with Shared Preference)

----> used to data in file
 ----> Create plaintext
 ----> Create Button
 ----> Create textview
 ----> At last the search the SharedPrefs app and see the result

(MainActivity.java)

```
package com.example.sharedprefs;

import androidx.appcompat.app.AppCompatActivity;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    private EditText EnterMessage;
    private Button saveButton;
    private TextView result;
    private SharedPreferences myPrefs;           -----> instantiate myPrefs
    private static final String PREFS_NAME = "myPrefsFile";           -----> create myPrefsfile

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

        EnterMessage = (EditText) findViewById(R.id.Enternameld);
        result = (TextView) findViewById(R.id.resultViewId);
        saveButton = (Button) findViewById(R.id.saveBtnId);
        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                myPrefs = getSharedPreferences("PREFS_NAME",0); // mode level 0 //
                Name of File String final String4
                SharedPreferences.Editor editor = myPrefs.edit();e

                // pass the data using key and value

                editor.putString("message",EnterMessage.getText().toString());
                editor.commit(); // commit means completely save the data with key

            }
        });

        // get the data back

        SharedPreferences prefs = getSharedPreferences("message", 0);
        if (prefs.contains("message")){
            String message = prefs.getString("message","not found");    // get the message
            back with key and set the default value not found
            result.setText("Message: "+ message);
        }
    }
}
```

```
}  
}  
}
```

26: (Honey do list App)---> Working with file Structure

----> Create the Plain multi text
----> Create Button

```
package com.example.todolist;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Context;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
  
import java.io.BufferedReader;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.OutputStreamWriter;  
  
public class MainActivity extends AppCompatActivity {  
    private EditText EnterMessage;  
    private Button SaveButton;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        EnterMessage = (EditText) findViewById(R.id.EnterTextId);  
        SaveButton = (Button) findViewById(R.id.saveBtnId);  
  
        SaveButton.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
  
                // logic here  
  
                if (!EnterMessage.getText().toString().equals("")) {  
  
                    String message = EnterMessage.getText().toString();  
                    try {  
                        WriteToFile(message);  
                    } catch (FileNotFoundException e) {
```

```

        e.printStackTrace();
    }
    }else {
        // do nothing for show
    }
}

});

try {
    if (ReadFromFile() != null){
        EnterMessage.setText(ReadFromFile());
    }
} catch (IOException e) {
    e.printStackTrace();
}
}

private void WriteToFile(String message) throws FileNotFoundException {

    try{
        OutputStreamWriter outputStreamWriter = new
        OutputStreamWriter(openFileOutput("todoList.txt", Context.MODE_PRIVATE));
        outputStreamWriter.write(message);
        outputStreamWriter.close(); // always close your stream
    }catch (FileNotFoundException e){
        e.printStackTrace();
    }catch (IOException e){
        e.printStackTrace();
    }
}

private String ReadFromFile() throws IOException {

    String result = "";

    InputStream inputStream = openFileInput("todoList.txt");

    if (inputStream != null){

        InputStreamReader inputStreamReader = new InputStreamReader(inputStream);
        BufferedReader bufferedReader = new BufferedReader(inputStreamReader);

        String tempString = "";
        StringBuilder stringBuilder = new StringBuilder();

        while ((tempString = bufferedReader.readLine()) != null ){

            stringBuilder.append(tempString);
        }
        inputStream.close();
        result = stringBuilder.toString();
    }

    return result;
}
}

```

27: (Contact Manager App With SQLite DB)

----> create 3 Package in java Main
----> 1st Model
----> 2nd Data
----> 3rd Utils

In Model

--> create Java as Contact
--> Create their Constructor
--> Create their Getter & Setter Method

package Model;

public class contact {

private int id;
private String Name;
private String PhoneNumber;

// Set their Constructor

public contact() {
}

public contact(int id, String name, String phoneNumber) {
this.id = id;
Name = name;
PhoneNumber = phoneNumber;
}

public contact(String name, String phoneNumber) {
Name = name;
PhoneNumber = phoneNumber;
}

/// set the Getter & Setter Method

public int getId() {
return id;
}

public void setId(int id) {
this.id = id;
}

public String getName() {
return Name;
}

```

    public void setName(String name) {
        Name = name;
    }

    public String getPhoneNumber() {
        return PhoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        PhoneNumber = phoneNumber;
    }
}

```

In Data

- > Create Java file Name DatabaseHandler
- > Extends with SQLiteOpenHelper
- > Create Thier 2 override methods
- > Create One Constructor

```

package Data;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DatabaseHandler extends SQLiteOpenHelper {
    public DatabaseHandler(@Nullable Context context, @Nullable String name, @Nullable
    SQLiteDatabase.CursorFactory factory, int version) {
        super(context, name, factory, version);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

    }
}

```

In Utils

- > create Util java file
- > Inside Util class We Create Static Final Variable for referencing in our DatabaseHandler
- > Create DataBase Version
- > Create contact Table Coloum name

```
package Utils;

public class Util {

    //DataBaseVersion

    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "contactDB";
    public static final String TABLE_NAME = "contacts";

    //Contact Table Columns Name

    public static final String KEY_ID = "id";
    public static final String KEY_NAME = "name";
    public static final String KEY_PHONE_NUMBER = "phone_number";

}
```

- > Go to DataBaseHandler
- > Add the DataBaseverion & DataBaseName in Parameter

(DatabaseHandler)

```
package Data;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

import Utils.Util;

public class DatabaseHandler extends SQLiteOpenHelper {
    public DatabaseHandler( Context context) {
        super(context, Util.DATABASE_NAME, null, Util.DATABASE_VERSION  );
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {

        // SQL- Structured Query Language
        // Create String CREATE_CONTACT_TABLE
        // Pass the ALL KEY with Their Type

        String CREATE_CONTACT_TABLE = "CREATE TABLE" + Util.TABLE_NAME + "(" +
            Util.KEY_ID + "INTEGER PRIMARY KEY" + Util.KEY_NAME + "TEXT" +
            Util.KEY_PHONE_NUMBER + "TEXT" + ")";

        // Execute the DataBase Code
        // Reference the Database sqLiteDatabase instance here
```

```
        //Execute SQL
        //Pass the String

        sqLiteDatabase.execSQL(CREATE_CONTACT_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {

        // For Upgrade the table
        // We Drop the Existing table
        // And Create table again

        sqLiteDatabase.execSQL (Util.TABLE_NAME );

        //CREATE TABLE AGAIN
        onCreate(sqLiteDatabase);
    }
}
```



27: (Grocery list App With Crud)

- 1: Start New Project
- 2: Select Activity
- 3: Create the 5 Packages
- 4: Activities,Data,Model,UI,Util
- 5: Drag your MainActivity.java and Others inside the Activites
- 6: There are 2 xml files in resources
- 7: And their code is Written in Activity_main.xml
- 8: Layout of content is define in Activity_main.xml like example

```
<include layout="@layout/content_main" />
```

9: Now Center your FloatingActionButton which is given at the bottom of Design in content_main.xml

10: for changes go to Activitu_main.xml file and do the changes like example

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fab"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center" -----> change occured here, We put Center
Instead of bottom
    android:layout_margin="@dimen/fab_margin"
    app:srcCompat="@android:drawable/ic_input_add" /> -----> change occured here
for + input sign
```

11: Add popup.xml file in layout

12: Change root to LinearLayout

13: Drag and Drop the Cardview inside LinearLayout

14: Do the changes in popip.xml Add the RelativeLayout inside the LinearLayout

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
```

```

        android:id="@+id/layout_id"-----> Here We add id
        android:layout_width="match_parent"
        android:layout_height="match_parent">

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

<RelativeLayout

    android:layout_width="match_parent" -----> Width is match_parent
    android:layout_height="wrap_content">-----> height is Wrap_content
    >

<TextView

    android:id="@+id/title"
    android:text="@string/enter_item"
    android:textSize="18sp"
    android:textStyle="italic"
    android:layout_centerHorizontal="true"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
<EditText
    android:id="@+id/GroceryItem"
    android:layout_below="@+id/title" -----> used for set the text below the textview
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/hint_item"
    />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/GroceryOTY"
    android:hint="@string/Quantity_hint"
    android:layout_below="@id/GroceryItem" ---> used for set the text below the
Edittext
    android:layout_marginTop="15dp"
    />
<Button
    android:id="@+id/saveButton"
    android:text="@string/save_title"
    android:layout_below="@id/GroceryOTY" ---> same
    android:layout_marginTop="5dp"
    android:background="#E91E63"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textColor="#E6DEF3"
    />

</RelativeLayout>

</androidx.cardview.widget.CardView>
</LinearLayout>

```

15: Create the popupclass inside the MainActivity.java

```
private void createPopupDialog(){
```

```
}
```

16: Call the PopUpDialog() inside the FloatigActionButton

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                .setAction("Action", null).show();

                createPopupDialog(); -----> Here we call the Popupidialog class
        }
    });
}
```

17: Here We instantiate our Ids and Create a dialog and inflate the view and other quantity inside PopupDialog class

```
package Activities;
```

```
import android.os.Bundle;
```

```
import com.example.grocerylist.R;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import com.google.android.material.snackbar.Snackbar;
```

```
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.widget.Toolbar;
```

```
import android.view.View;
```

```
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private AlertDialog.Builder dialogBuilder;
    private AlertDialog dialog;
    private EditText groceryItem;
    private EditText quantity;
    private Button saveButton;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```

        setContentView(R.layout.activity_main);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();

                createPopupDialog();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    private void createPopupDialog(){

        dialogBuilder = new AlertDialog.Builder(this); -----> Instantiate the
class and pass this
        View view = getLayoutInflater().inflate(R.layout.popup,null); -----> get the
popup.xml view by LayoutInflater
        groceryItem = (EditText) view.findViewById(R.id.GroceryItem); ----->
view.findViewById cuz we call with view
        quantity = (EditText) view.findViewById(R.id.GroceryQTY);
        saveButton = (Button) view.findViewById(R.id.saveButton);

        dialogBuilder.setView(view); ----> here We set the View
        dialog = dialogBuilder.create(); ----> here we create the dialog
        dialog.show(); ----> here We show the dialog

        saveButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                //TODO: Save to db
            }
        });
    }

```

//TODO: Go to the Next Screen

saveGroceryToDB(view); -----> call the method
here and declare class outside

```
    }  
  
    });  
}  
  
private void saveGroceryToDB(View view) { -----> Here We  
declare the class  
  
}  
}
```

- 17: create grocery class inside the (Model)
- 18: grocery class allow us to create grocery object
- 19: Create Constructor and Create getter and setter method

package Model;

public class Grocery {

```
    private String name;  
    private String quantity;  
    private String dateItemAdded;  
    private int id;
```

/// Create Empty as well as All Constructor

```
    public Grocery() {  
    }  
    public Grocery(String name, String quantity, String dateItemAdded, int id) {  
        this.name = name;  
        this.quantity = quantity;  
        this.dateItemAdded = dateItemAdded;  
        this.id = id;  
    }  
}
```

/// Create all getter and setter

```
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getQuantity() {  
        return quantity;  
    }  
}
```

```

    public void setQuantity(String quantity) {
        this.quantity = quantity;
    }

    public String getDateItemAdded() {
        return dateItemAdded;
    }

    public void setDateItemAdded(String dateItemAdded) {
        this.dateItemAdded = dateItemAdded;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}

```

20: Create DatabaseHandler.java class in DATA

Create Constants.java class in Util ----> This class Will hold the all Constant Variable to Create a Database

Constant.java

package Util;

```

public class Constants {

    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "groceryListDB";
    public static final String TABLE_NAME = "groceryTBL";

    // Table Column

    public static final String KEY_ID = "id";
    public static final String KEY_GROCERY_ITEM = "grocery_item";
    public static final String KEY_QUANTITY_NUMBER = "quantity_number";
    public static final String KEY_DATE_NAME = "date_added";
}

```

21: Go to DatabaseHandler

--> extends or we can say inheritance the SQLiteOpenHelper
 --> Implements the methods
 --> Construct the Constructor
 --> Erase the remaining field expect Context context cuz we only need this

```

public class DatabaseHandler extends SQLiteOpenHelper { ----> Here We extends the SQLite

    private Context ctx; -----> Here we instantiate Constant
    public DatabaseHandler(Context context) { -----> Here We pass

```

the context

```
super(context, Constants.DATABASE_NAME,null,Constants.DATABASE_VERSION); --  
-----> Pass the DB name ,factory null, DB Version  
    this.ctx = context; -----> In constructor  
}
```

-----> Now Create the Database

```
@Override  
public void onCreate(SQLiteDatabase sqLiteDatabase) {  
  
    String CREATE_GROCERY_DATABASE = "CREATE_TABLE " +  
Constants.TABLE_NAME + "(" -----> DB having their Statements  
    + Constants.KEY_ID + " INTEGER_PRIMARY_KEY," +  
Constants.KEY_GROCERY_ITEM + " TEXT,"  
    + Constants.KEY_QTY_NUMBER + " TEXT,"  
    + Constants.KEY_DATE_NAME + " LONG);";  
  
}
```

-----> Create the TABLE with all fields and column the we need

-----> Go to Upgrade and Execute the SQL

-----> On create Table again just we pass our DB

```
@Override  
public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {  
  
    sqLiteDatabase.execSQL("DROP TABLE IF EXISTS " + Constants.TABLE_NAME );  
    onCreate(sqLiteDatabase);  
}
```

-----> We have to Create the Methods for Doing the CRUD Create-Read-Update-Delete Operation

// CRUD Operation - We have to Create the classes

```
// Add Grocery  
public void AddGrocery(Grocery grocery) {
```

SQLiteDatabase sqLiteDatabase = this.getWritableDatabase(); -->we able to write a data from this method

ContentValues values = new ContentValues(); -->Content value object is used for adding our data with key Key & value

```
values.Put(Constants.KEY_GROCERY_ITEM, grocery.getname());  
values.Put(Constants.KEY_QTY_NUMBER, grocery.getQuantity());  
values.Put(Constants.KEY_DATE_NAME, java.lang.System.currentTimeMillis()); -->This  
the method for live Current time
```

// Now Insert our value into the Table row

```
sqLiteDatabase.insert(Constants.TABLE_NAME,null,values); -->null is used for  
ColumnHack
```

-> We Insert our Content value inside parameter

```
Log.d("Saved!!","Saved to Database"); --> Only for log purpose
```

```

    }

// Get Grocery
private Grocery getGrocery(int id){

    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase(); -->we able to write a
data from this method

    Cursor cursor = sqLiteDatabase.query(Constant.TABLE_NAME,new
String[]{Constant.KEY_ID,
                                Constant.KEY_GROCERY_ITEM,
                                Constant.KEY_QTY_NUMBER,
                                Constant.KEY_DATE_NAME},
                                // Selection by KEY_ID
                                Constant.KEY_ID + "=?",
                                new String[]{String.valueOf(id
)},null,null,null,null); --> groupby:null, having:null, orderby:null, limit:null

    if(cursor != null)
        cursor.moveToFirst();

        // Create New Grocery Object

        Grocery grocery = new Grocery();

        grocery.SetId(Integer.parseInt(cursor.getString()(cursor.getColumnIndex(Constants.KEY_ID))))
        ;// We Set the ID not but converted into String using parseInt and then we get string and then
        we get our columnIndex

        grocery.SetName(cursor.getString()(cursor.getColumnIndex(Constant.KEY_GROCERY_ITEM)))
        ;

        grocery.SetQuantity(cursor.getString()(cursor.getColumnIndex(Constant.KEY_QTY_NUMBER))
        );

        // We have another field called KEY_DATE_NAME that the save the time in
        mileSecond or systemTime
        // We Want something which is Readable
        // Covert TimeStamp to something Readable

        java.lang.text    DateFormat    dateFormat    =    java.lang.text
DateFormat.getDateInstance();
        String    formattedDate    =    dateFormat    format(new
Date(cursor.getLong(cursor.getColumnIndex(Constant.KEY_DATE_NAME))).getTime());

        grocery.setDateItemAdded(formatedDate);

        return grocery;
    }

// Get All Grocery

    public List<Grocery> getAllGrocery(){

```



```
        return null;
    }

// Update Grocery

    public int updateGrocery(Grocery grocery) {

        return 0;
    }

// Delete Grocery

    public void deleteGrocery(int id){

    }

//Get Count

    public int getGroceriesCount(){

        return 0;
    }
}

}
```