

Lab Manuals for Mobile Application Development



Raja Jait Singh Government
Polytechnic, Neemka

Prepared by
Department Of Computer Science

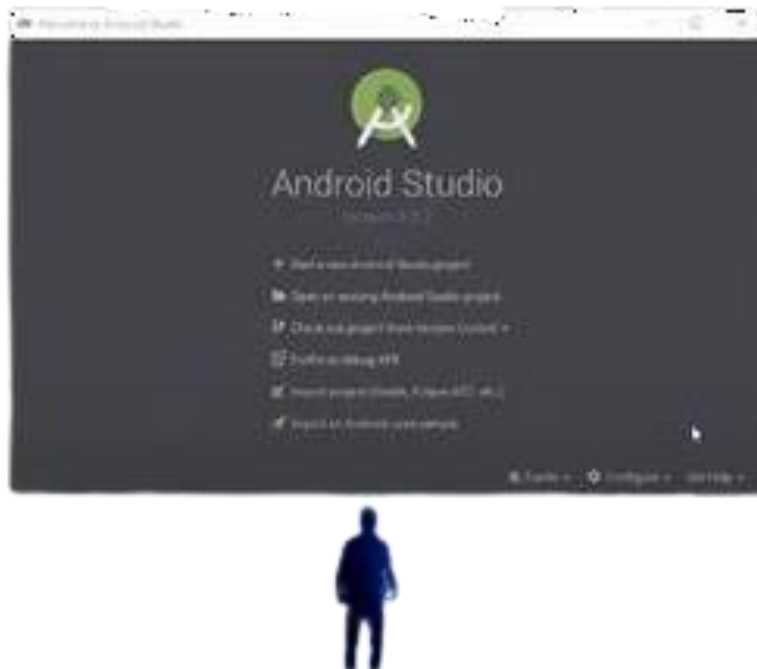
WHAT IS MOBILE DEVELOPMENT?

Mobile application development is the set of processes and procedures involved in writing software for small, wireless computing devices, such as Smartphone's and other hand-held devices.

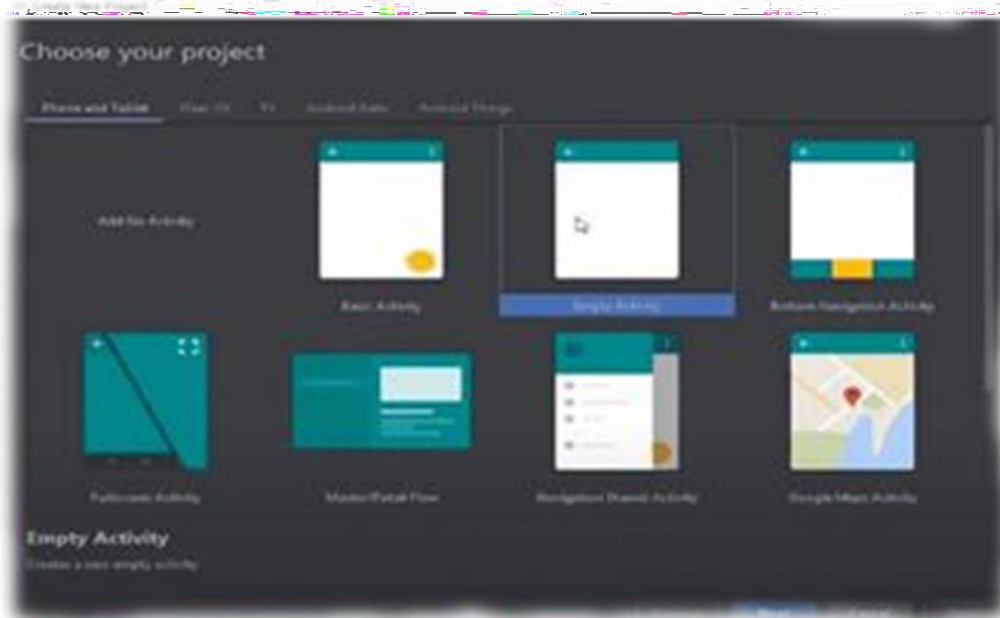
What is Android Studio?

In recent times, Android became the world's most popular operating system for various reasons. As an Android programmer, I want to share what the Android Studio is? Android Studio is an IDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. The Android Studio uses Gradle to manage your project, a Build Automation Tool.

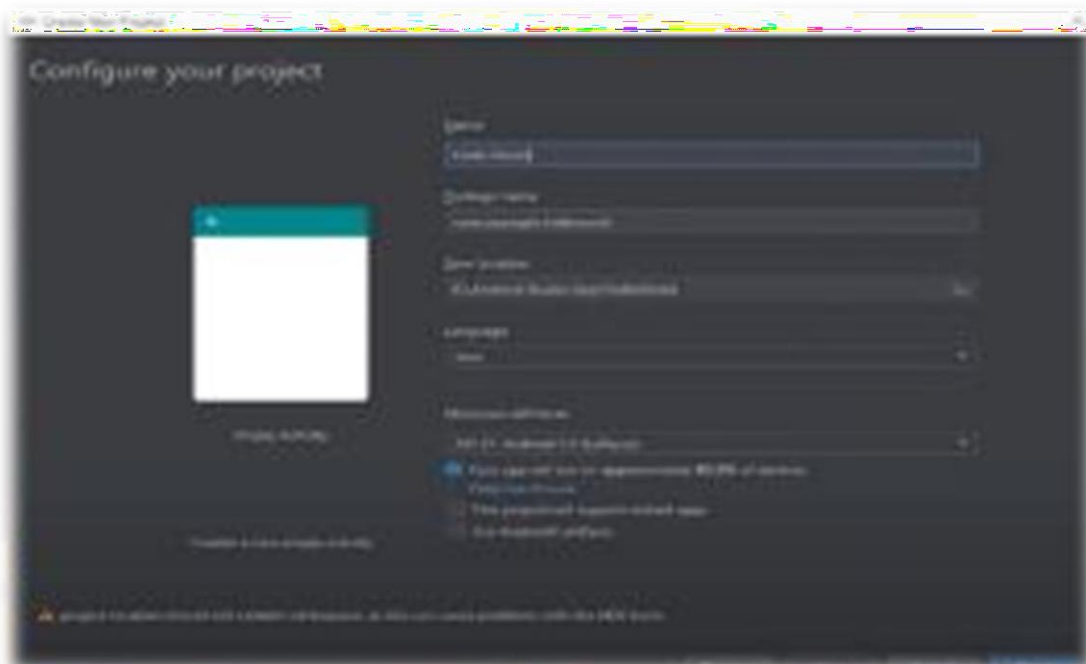
- Step 1. Open your Android Studio after installing, and this following screen appears after starting



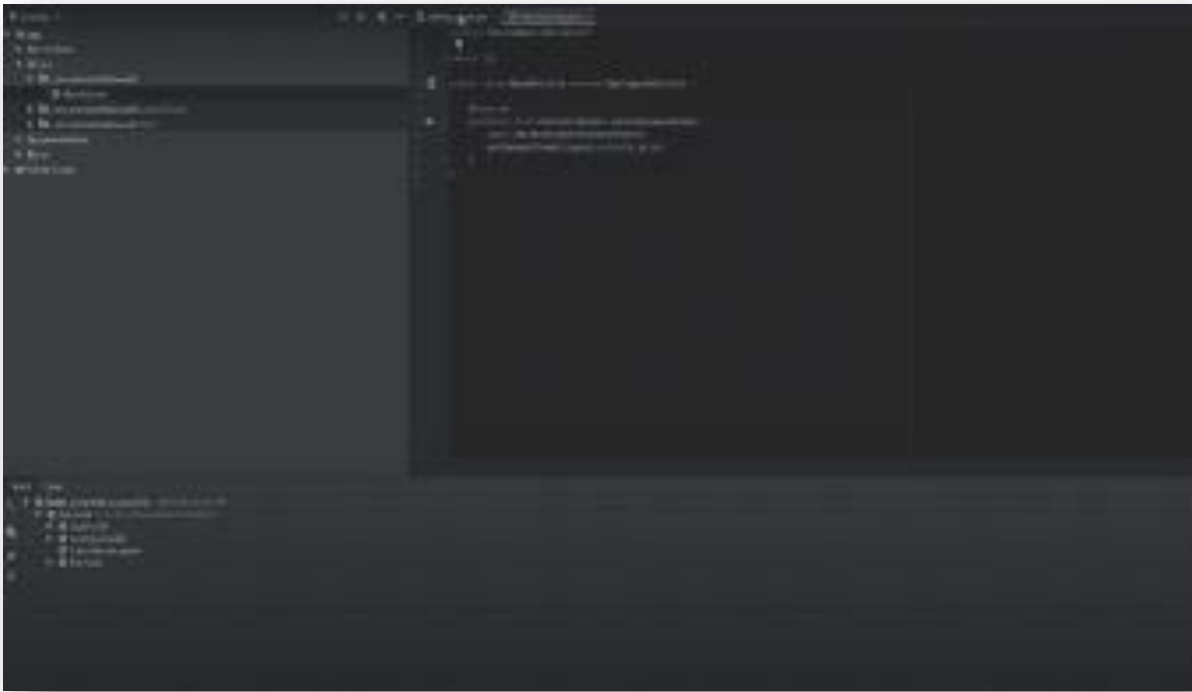
- Step 2. Click on "Start a New Android Project" and choose the activity we want the project to be. For me, I'm selecting an empty activity.



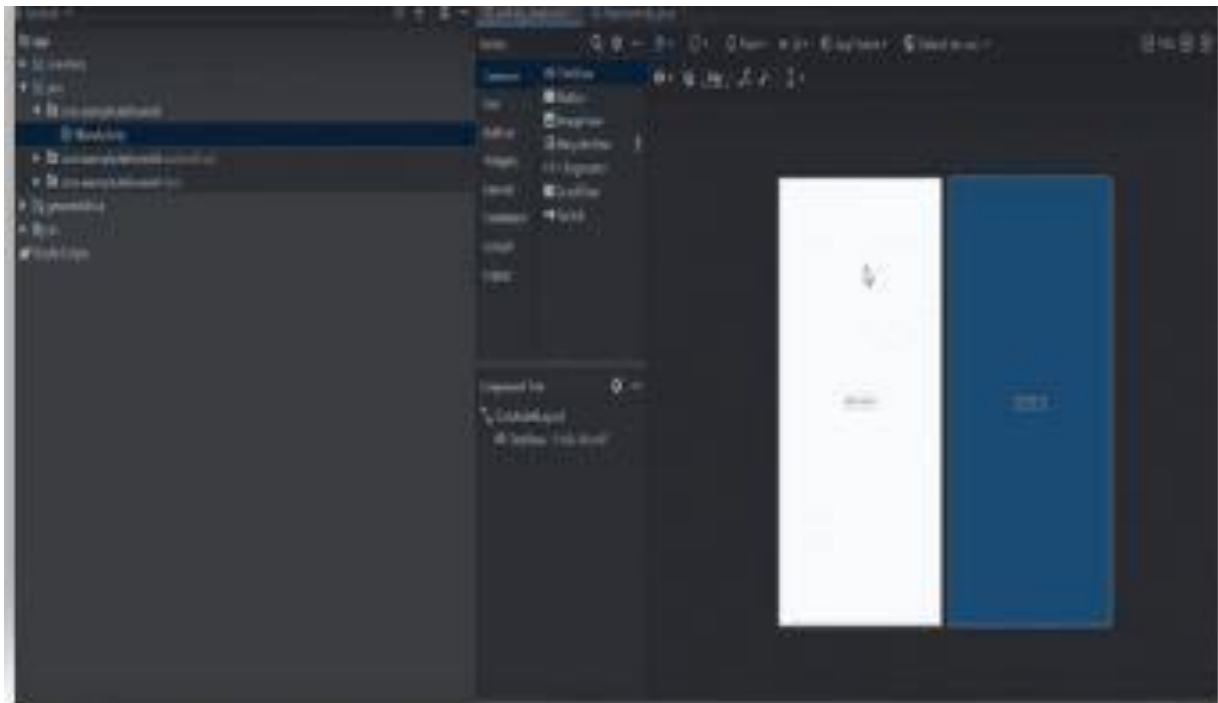
Step 3. Configure your project by changing the project's name and the desired language the user wants to code with. Android Studio supports two languages, i.e., Java and Kotlin. The user can also choose their desired API, like which android version they want to run, and then click on finish.



Step 4. Wait until your Gradle build finishes successfully and find your java and .xml file from folders of the app, as shown in the given image.



Step 5. Click onto your .xml file to see the initial layout of the file like the given image. You can drag and drop from the palette's items and change your layout by dragging and dropping.



Step 6. At last, run your file by the run button on the top right corner in your emulator configured.

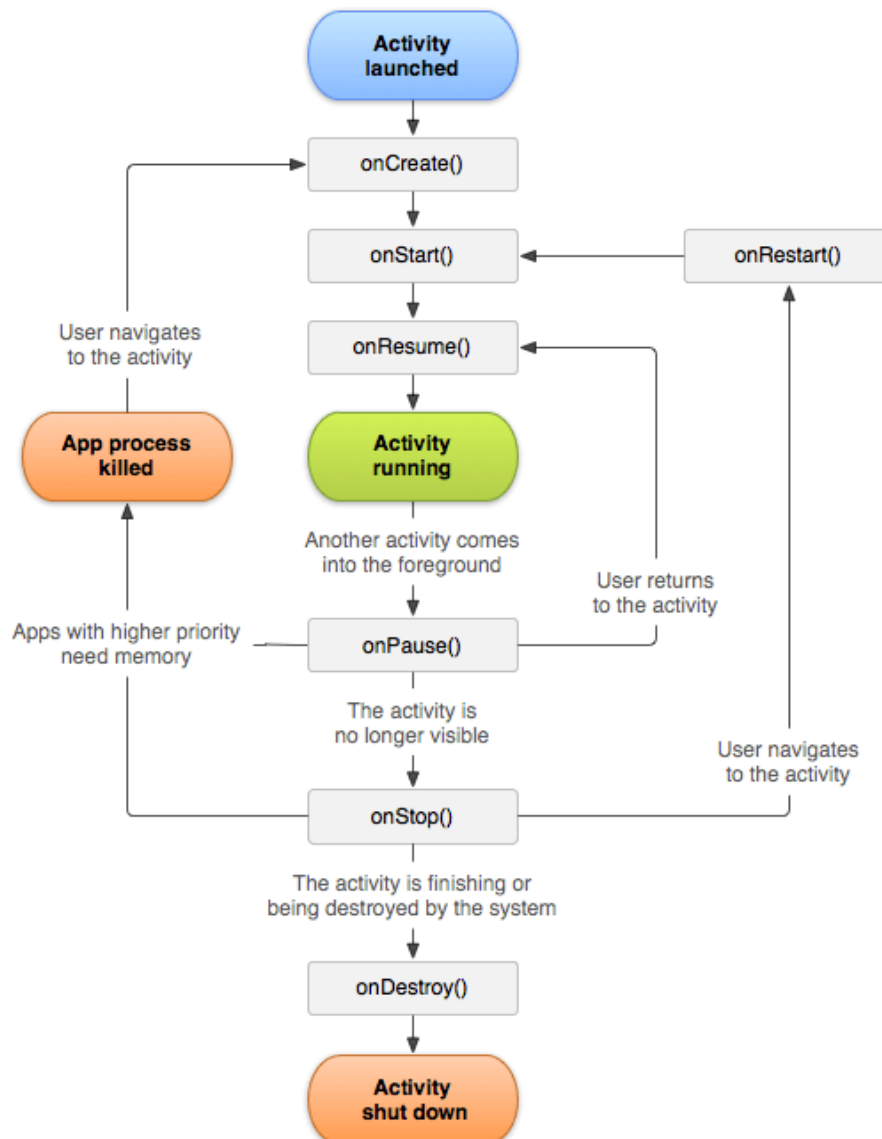
PRACTICAL 1

WRITE A PROGRAM TO DEMONSTRATE ACTIVITY (APPLICATION LIFE CYCLE).

ANDROID: Android is an Operating system and Programming Platform developed by Google for mobile Phones and other mobile devices, such as tablets.

Application Life Cycle: An Android activity goes through six major lifecycle stages or callbacks. These are: onCreate(), onStart(), onResume(), onPause(), onStop(), and onDestroy(). The system invokes each of these callbacks as an activity enters a new state.

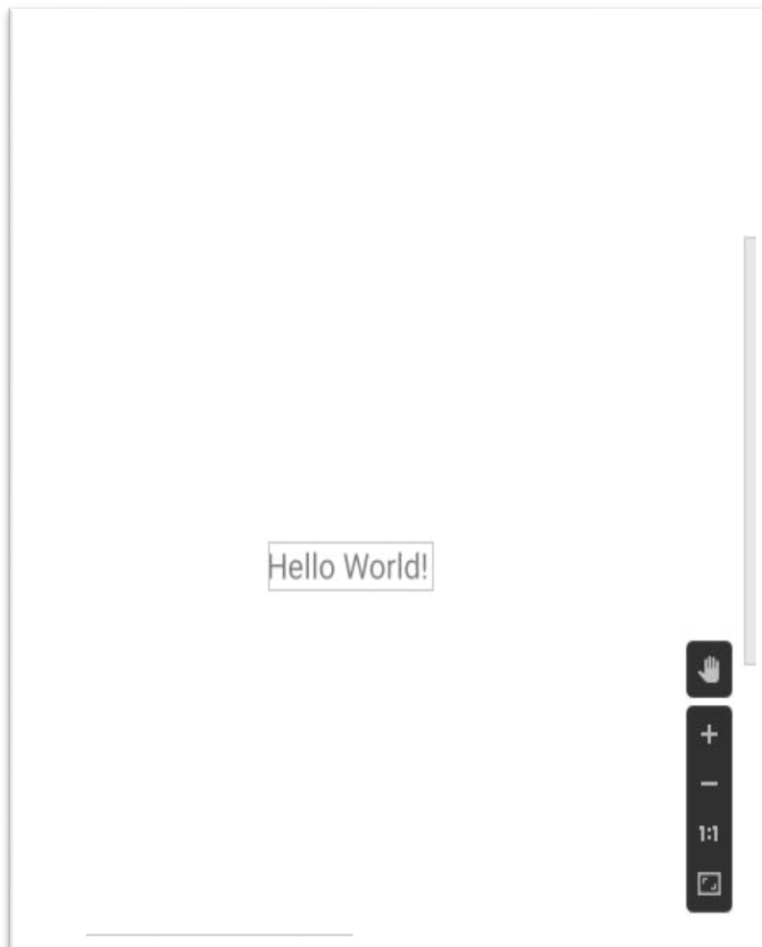
ANDROID ACTIVITY LIFECYCLE



FILE: ACTIVITY MAIN.XML

```
activity_main.xml x MainActivity.kt x
Code Split Design
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   xmlns:app="http://schemas.android.com/apk/res-auto"
4   xmlns:tools="http://schemas.android.com/tools"
5   android:layout_width="match_parent"
6   android:layout_height="match_parent"
7   tools:context=".MainActivity">
8
9   <TextView
10    android:layout_width="wrap_content"
11    android:layout_height="wrap_content"
12    android:text="Hello World!"
13    app:layout_constraintBottom_toBottomOf="parent"
14    app:layout_constraintEnd_toEndOf="parent"
15    app:layout_constraintStart_toStartOf="parent"
16    app:layout_constraintTop_toTopOf="parent" />
17
18 </androidx.constraintlayout.widget.ConstraintLayout>
```

DESIGN:



PRACTICAL 2

✚ Write a program to demonstrate different type of layouts.

Layouts: A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and View Group objects.

❖ There are different types of layouts:-

- Linear Layout: It is a view group that aligns all children in a single direction, vertically or horizontally.
- Relative Layout: Relative Layout is a view group that displays child views in relative positions.
- Table Layout: Table Layout is a view that groups views into rows and columns.
- Absolute Layout: Absolute Layout enables you to specify the exact location of its children.
- Frame Layout: The Frame Layout is a placeholder on screen that you can use to display a single view
- List View: List View is a view group that displays a list of scrollable items.
- Grid View: Grid View is a View Group that displays items in a two-dimensional, scrollable grid

PRACTICAL 3

Write a Program to implement simple calculator using Text View, Edit View, Option Button and Button

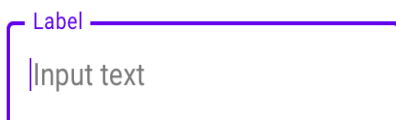
Create a simple calculator which can perform basic arithmetic operations like addition, subtraction, multiplication, or division depending upon the user input.

Text View



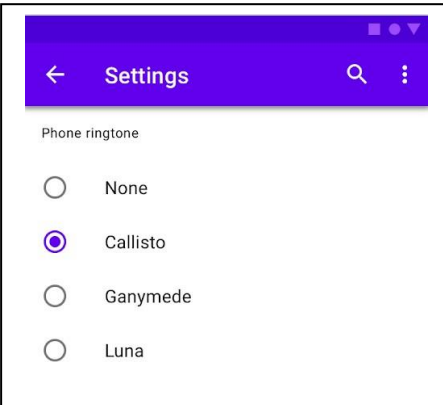
Text View:- This class is used to display text on the android application screen. it also allows user to Optionally edit it .Although it Contains text editing operations, the basic class does not allow editing.

Edit Text View



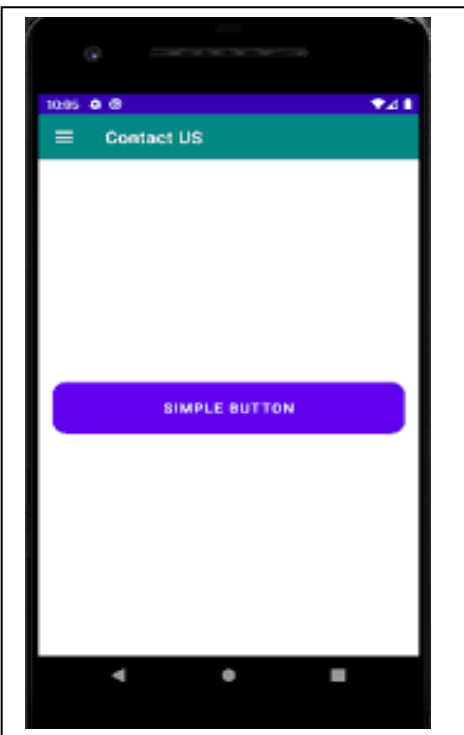
Edit Text View:- An Edit text is an overlay over Text View that configures itself to be Editable. Edit text is used when you want to have a text field in your Application. Where user can enter any text .It can be either Single line or Multi-Line.

Option Menu



Option Menu:- Option menu is a primary collection of menu items for an activity and it is useful to implement action that have a global impact on the app, such as Settings, Search etc.

Button View



Button View :- Button is a component which can be pressed or clicked by the user to perform an action. The main usage of button view is that whenever we click a Button, we can set a method that will handle that specific button request and will carry out the necessary action.

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@color/teal_700"
tools:context=".MainActivity">
```

<EditText

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter no. 1" android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.407"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.123" />
```

<EditText

```
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Enter no.2" android:inputType="number"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.407"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText1"
    app:layout_constraintVertical_bias="0.073" />
```

<TextView

```
    android:id="@+id/textresult"
    android:layout_width="96dp"
    android:layout_height="30dp"
    android:text="RESULT"
    android:textSize="24sp"
    android:textStyle="bold|italic"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.26"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editText2"
    app:layout_constraintVertical_bias="0.076" />
```

<Button

```
android:id="@+id/button"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="48dp"  
android:layout_marginBottom="40dp"  
android:onClick="btnsum" android:text="SUM"  
app:layout_constraintBottom_toTopOf="@+id/button2"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.753"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toBottomOf="@+id/textresult"  
app:layout_constraintVertical_bias="0.0" />
```

<Button

```
android:id="@+id/button2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="40dp"  
android:onClick="btnsub" android:text="SUB"  
app:layout_constraintBottom_toTopOf="@+id/button3"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.752"  
app:layout_constraintStart_toStartOf="parent" />
```

<Button

```
android:id="@+id/button3"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:onClick="btnmulty" android:text="MULTY"  
app:layout_constraintBottom_toTopOf="@+id/button4"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.752"  
app:layout_constraintStart_toStartOf="parent"  
app:layout_constraintTop_toTopOf="parent"  
app:layout_constraintVertical_bias="0.932" />
```

<Button

```
android:id="@+id/button4"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginBottom="87dp"  
android:onClick="btndiv" android:text="DIV"  
app:layout_constraintBottom_toBottomOf="parent"  
app:layout_constraintEnd_toEndOf="parent"  
app:layout_constraintHorizontal_bias="0.752"  
app:layout_constraintStart_toStartOf="parent" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

MAIN ACTIVITY.JAVA:-

```
package com.example.simplecalculator;

import androidx.appcompat.app.AppCompatActivity;import
    android.os.Bundle;

import android.view.View; import
    android.widget.EditText;import
    android.widget.TextView;

public class MainActivity extends AppCompatActivity {private
    EditText etn1;

private EditText etn2; private
    TextView tvResult;

@Override

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

    etn1 = (EditText) findViewById(R.id.editText1);etn2 = (EditText)
        findViewById(R.id.editText2);

    tvResult = (TextView) findViewById(R.id.textresult);
}

public void btnsum(View view) {

int n1 = Integer.parseInt(etn1.getText().toString());int n2 =
    Integer.parseInt(etn2.getText().toString());int sum = n1 + n2;
    tvResult.setText(String.valueOf(sum));

}

public void btnsub(View view) {

int n1 = Integer.parseInt(etn1.getText().toString());int n2 =
    Integer.parseInt(etn2.getText().toString());int sub = n1 - n2;
    tvResult.setText(String.valueOf(sub));

}

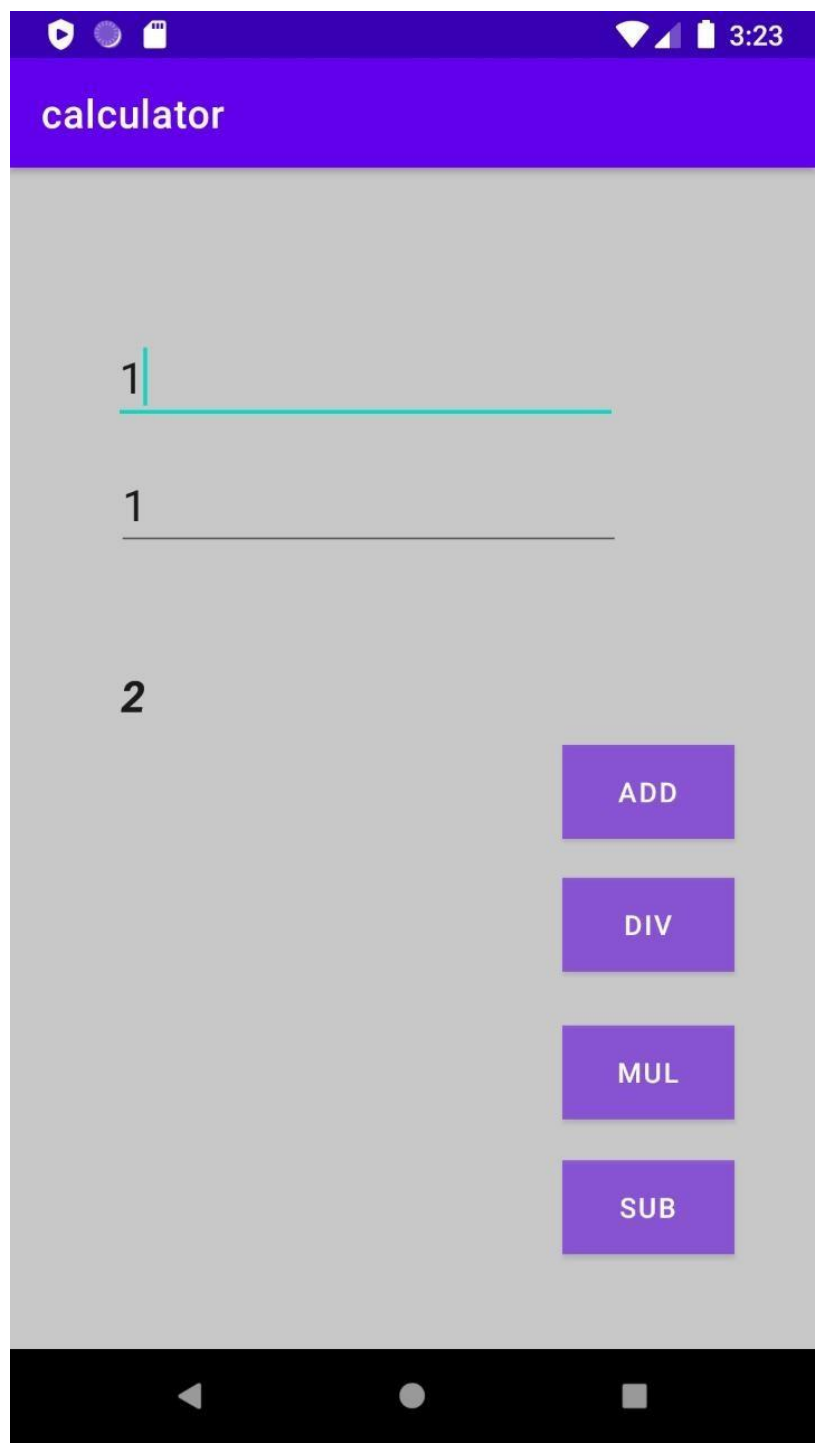
public void btnmulty(View view) {

int n1 = Integer.parseInt(etn1.getText().toString());int n2 =
    Integer.parseInt(etn2.getText().toString());int multy = n1 * n2;
    tvResult.setText(String.valueOf(multy));

}
```

```
public void btndiv(View view) {  
  
int n1 = Integer.parseInt(etn1.getText().toString());int n2 =  
Integer.parseInt(etn2.getText().toString());int div = n1 / n2;  
tvResult.setText(String.valueOf(div));  
  
}g  
  
}
```

DESIGN:



PRACTICAL 4

✚ WRITE A PROGRAM TO DEMONSTRATE LIST VIEW.

LIST VIEW: - Android List View is a view which contains the group of items and displays in a scrollable list.

ACTIVITY.MAIN_XML

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="listview.example.com.listview.MainActivity">

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="fill_parent"
        />
</android.support.constraint.ConstraintLayout>
```

MYlist.XML

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Medium Text"
    android:textStyle="bold">
```

```
android:textAppearance="?android:attr/textAppearanceMedium"  
android:layout_marginLeft="10dp"  
android:layout_marginTop="5dp"  
android:padding="2dp"  
android:textColor="#4d4d4d"  
/>
```

Strings.xml

```
<resources>  
<string name="app_name">ListView</string>  
<string-array name="array_technology">  
    <item>Android</item>  
    <item>Java</item>  
    <item>Php</item>  
    <item>Hadoop</item>  
    <item>Sap</item>  
    <item>Python</item>  
    <item>Ajax</item>  
    <item>C++</item>  
    <item>Ruby</item>  
    <item>Rails</item>  
    <item>.Net</item>  
    <item>Perl</item>  
</string-array>  
</resources>
```

MainActivity.java

```
package listview.example.com.listview;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    ListView listView;
    TextView textView;
    String[] listItem;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        listView=(ListView)findViewById(R.id.listView);
        textView=(TextView)findViewById(R.id.textView);
        listItem = getResources().getStringArray(R.array.array_technology);

        final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
        android.R.layout.simple_list_item_1, android.R.id.text1, listItem);

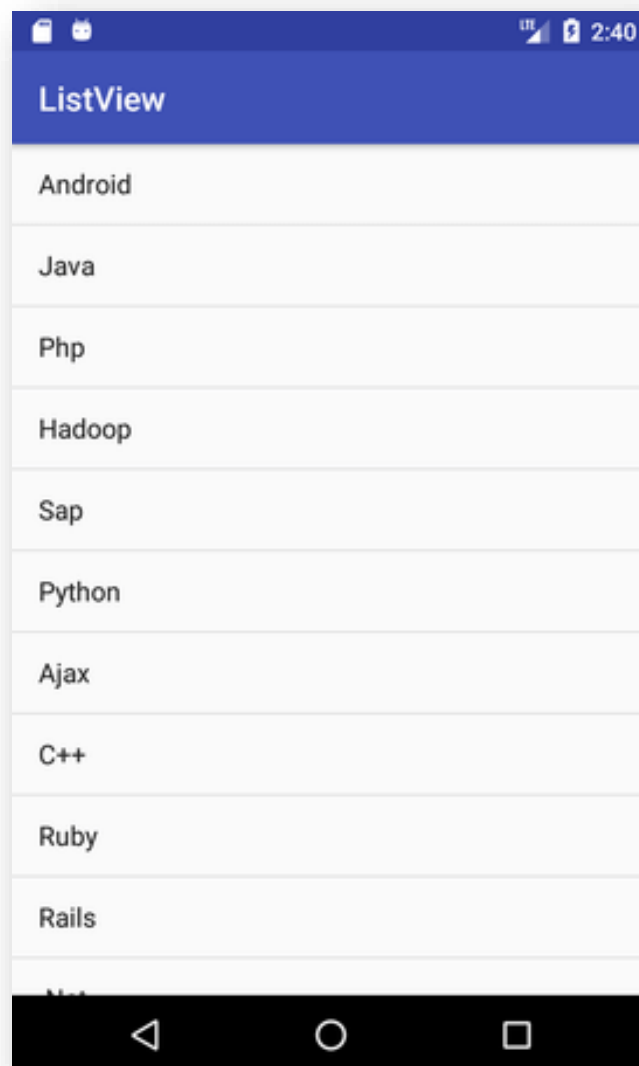
        listView.setAdapter(adapter);

        listView.setOnItemClickListener(new AdapterView.OnItemClickListener
        () {
            @Override
```



```
public void onItemClick(AdapterView<?> adapterView, View view,
int position, long l) {
    // TODO Auto-generated method stub
    String value=adapter.getItem(position);
    Toast.makeText(getApplicationContext(),value,Toast.LENGTH_S
SHORT).show();
}
});
}
```

DESIGN:



PRACTICAL 5

Write a program to demonstrate photo gallery.

Gallery View: Gallery is a view used to show items in a center locked, horizontal scrolling list and user will select a view than user selected view will be shown in the center of the Horizontal list.

Adapter used to fill Image in Gallery:

Adapter work as a bridge between Adapter View and data source.

Activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="#333333"
    android:orientation="vertical"
    ">

    <RelativeLayout

        android:layout_width="match_parent"
        android:layout_height="56dp">

        <TextView

            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/gallery_number"
            android:layout_centerVertical="true"
            android:layout_marginLeft="16dp"
            android:textColor="@android:color/white"
            android:textSize="18dp"
            android:textStyle="bold"/>

        </RelativeLayout>

        <androidx.recyclerview.widget.RecyclerView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/recyclerView_gallery_images"/>

    </LinearLayout>
```

Main_Activity.Java

```
package com.example.gallery;

import androidx.annotation.NonNull;
import
androidx.appcompat.app.AppCompatActivity;import
androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import
androidx.recyclerview.widget.GridLayoutManager;import
androidx.recyclerview.widget.RecyclerView;

import android.Manifest;
import
android.content.pm.PackageManager;import
android.os.Bundle;
import
android.widget.TextView;import
android.widget.Toast;

import java.util.List;

public class MainActivity extends AppCompatActivity

    {RecyclerView recyclerView;
    GalleryAdapter
    galleryAdapter;List<String>
    images;
    TextView gallery_number;

    private static final int My_READ_PERMISSION_CODE =

    101;@Override

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

        gallery_number = findViewById(R.id.gallery_number);
        recyclerView =
        findViewById(R.id.recyclerView_gallery_images);

        if

            (ContextCompat.checkSelfPermission(MainActivity
            .this,
            Manifest.permission.READ_EXTERNAL_STORAGE)!=
PackageManager.PERMISSION_GRANTED){

                ActivityCompat.requestPermissions(MainActivity.this,
                    new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
My_READ_PERMISSION_CODE);

            }
            else {
                loadimages();
            }
        }
    }
}
```

```

    }
}

private void loadimages(){
    recyclerView.setHasFixedSize(true)
    ;
    recyclerView.setLayoutManager(new GridLayoutManager(this,
    4));images = ImagesGallery.lisOfImages(this);
    galleryAdapter = new GalleryAdapter(this, images,
newGalleryAdapter.PhotoListener() {
        @Override
        public void OnPhotoClick(String Path) {
            Toast.makeText(MainActivity.this,""+Path,
Toast.LENGTH_SHORT).show();

        }
    });
    recyclerView.setAdapter(galleryAdapter);
    gallery_number.setText("Photos
    ("+images.size()+")");
}

@Override
public void onRequestPermissionsResult(int requestCode,
@NonNullString[] permissions, @NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions,
grantResults);
    if(requestCode ==My_READ_PERMISSION_CODE){
        if(grantResults[0] ==
PackageManager.PERMISSION_GRANTED){

            Toast.makeText(this, "Read external storage permission
granted",Toast.LENGTH_SHORT).show();

            loadimage();
        }
        else{
            Toast.makeText(this,"Read external storage permission
denied",Toast.LENGTH_SHORT).show();

        }
    }
}

private void loadimage() {
}
}

```

GalleryAdapter.java:

```

package com.example.galllery;

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

import androidx.annotation.NonNull;
import

```

```

androidx.recyclerview.widget.RecyclerView;import
com.bumptech.glide.Glide;

import java.util.List;

public class GalleryAdapter extends
RecyclerView.Adapter<GalleryAdapter.Viewholder>
{

    private Context context;
    private List<String>
    images;
    protected PhotoListener photoListener;

    public GalleryAdapter(Context context, List<String> image,
PhotoListenerphotoListener){
        this.context =
        context;this.images =
        images;
        this.photoListener = photoListener;
    }

    @NonNull
    @Override
    public Viewholder onCreateViewHolder(@NonNull ViewGroup parent,
int viewType) {
        return new Viewholder(
            LayoutInflater.from(context).inflate(R.layout.gallery_item,
parent, false)
        );
    }

    @Override
    public void onBindViewHolder(@NonNull Viewholder holder, int position)
    {
        final String image = images.get(position);

        Glide.with(context).load(image).into(holder.image)

        ;

        holder.itemView.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                photoListener.OnPhotoClick(image)
                ;
            }
        });
    }

    @Override
    public int getItemCount()
    {
        return images.size();
    }

    public class Viewholder extends RecyclerView.ViewHolder

        {ImageView image;

```

```

        public ViewHolder(@NonNull View itemView)
            {super(itemView);
            image = itemView.findViewById(R.id.image);
        }
    }
    public interface PhotoListener{
        void OnPhotoClick(String
        Path);
    }
}

```

AndroidManifest.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.galllery">

    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <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/Theme.Galllery">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

<uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <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/Theme.Galllery">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

```

Themes.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.galllery">

    <uses-permission
        android:name="android.permission.READ_EXTERNAL_STORAGE"/>

    <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/Theme.Galllery">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

Gallery_item.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-
auto" android:padding="1dp">

<ImageView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:scaleType="centerCrop"
    android:contentDescription="Gallery
image"
    app:layout_constraintDimensionRatio="1"
    android:id="@+id/image"
    tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

ImagesGallery.java:

```
package com.example.galllery;

import
android.content.Context; import
android.database.Cursor; import
android.net.Uri;
import android.provider.MediaStore;

import
java.lang.reflect.Array; import
java.util.ArrayList;

public class ImagesGallery {

    public static ArrayList<String> lisOfImages(Context
context){Uri uri;
Cursor cursor;
int column_index_data, column_index_folder_name;
ArrayList<String> listofAllImages = new
ArrayList<>();String ablosutePathOfImage;
uri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;

String[] projection =
{MediaStore.MediaColumns.DATA,
MediaStore.Images.Media.BUCKET_DISPLAY_NAME};

String orderBy = MediaStore.Video.Media.DATE_TAKEN;
cursor = context.getContentResolver().query(uri,
projection,null,null, orderBy+"DESC");
column_index_data =
cursor.getColumnIndexOrThrow(MediaStore.MediaColumns.DATA)
;

// get folder name
//column_index_folder_name =
cursor.getColumnIndexOrThrow(MediaStore.Images.Media.BUCKET_DISPLAY_NAME)
;

while (cursor.moveToNext()){
ablosutePathOfImage = cursor.getString(column_index_data);

listofAllImages.add(ablosutePathOfImage);
}

return listofAllImages;
}
}
```


DESIGN:



Practical -6

Write a program to demonstrate Date picker and Time picker

- ≡ **Time Picker:** Android Time Picker allows you to select the time of the day in either 24 hour or AM/PM mode. The time consists of hours, minutes and clock format.
- ≡ **Date Picker:** Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides Date Picker and Date Picker Dialog components.

(Activity_main.xml):

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

    <EditText
        android:hint="SELECT DATE"
        android:id="@+id/date_input"
        android:inputType="none"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <EditText
        android:hint="SELECT TIME"
        android:id="@+id/time_input"
        android:inputType="none"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

    <EditText
        android:hint="SELECT DATE and
        TIME"
        android:id="@+id/date_time_input"
        android:inputType="none"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

MainActivity.java:

```
Package com.example.androiddatetimepickerdialog;

import androidx.appcompat.app.AppCompatActivity;

import android.app.DatePickerDialog;

import android.app.TimePickerDialog;
import android.os.Bundle;
import android.text.InputType;
import android.view.View;
import android.widget.DatePicker;
import android.widget.EditText;
import android.widget.TimePicker;

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.SimpleTimeZone;

    public class MainActivity extends
        AppCompatActivity {EditText date_in;
EditText time_in;
    EditText date_time_in;

        @Override
        protected void onCreate(Bundle
            savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);
            date_in=findViewById(R.id.date_input);
            time_in=findViewById(R.id.time_input);
            date_time_in=findViewById(R.id.date_time_i
                nput);

            date_time_in.setInputType(InputType.TYPE_N
                ULL );
            date_in.setInputType(InputType.TYPE_NULL);
            time_in.setInputType(InputType.TYPE_NULL);

            date_in.setOnClickListener(new
                View.OnClickListener() {@Override
                public void onClick(View
                    view) {
                    showDateDialog(date_i
                        n);
                }
            });
            time_in.setOnClickListener(new
                View.OnClickListener() {@Override
                public void onClick(View
                    view) {
                    showTimeDialog(time_i
                        n);
                }
            });
            date_time_in.setOnClickListener(new
                View.OnClickListener() {@Override
                public void onClick(View view)
                    {
                    showDateTimeDialog(date_ti
                        me_in);
                }
            });
        }
    }
```

```

    }
    private void showDateTimeDialog(EditText
        date_time_in){ Calendar
        calendar=Calendar.getInstance();
        DatePickerDialog.OnDateSetListener
        dateSetListener=new
DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker datePicker, int
year, int month, int dayofmonth) {
                calendar.set(Calendar.YEAR,year);
                calendar.set(Calendar.MONTH,month);
                calendar.set(Calendar.DAY_OF_MONTH,dayofm
onth);

                TimePickerDialog.OnTimeSetListener
timeSetListener=newTimePickerDialog.OnTimeSetListener() {
                    @Override

                    public void onTimeSet(TimePicker
timePicker, int hourofday, int minute) {
                        calendar.set(Calendar.HOUR_OF_DAY,hourof
day);
                        calendar.set(Calendar.MINUTE,minute);

                        SimpleDateFormat
simpleDateFormat=newSimpleDateFormat("yy-MM-
dd/HH:mm");

date_time_in.setText(simpleDateFormat.format(calendar.getTime()));
                    }
                };
                new
TimePickerDialog(MainActivity.this,timeSetListener,calendar.get(Calendar
.HOUR_OF_DAY),calendar.get(Calendar.MINUTE),false).show();
            }
        };
        new
DatePickerDialog(MainActivity.this,dateSetListener,calendar.get(Calendar
.YEAR),calendar.get(Calendar.MONTH),calendar.get(Calendar.DAY_OF_MONTH)).sh
ow(
    );
    }
    private void showTimeDialog(EditText
        time_in){Calendar
        calendar=Calendar.getInstance();
        TimePickerDialog.OnTimeSetListener
timeSetListener=newTimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker timePicker, int hourofday, int
minute) {

calendar.set(Calendar.HOUR_OF_DAY,hourofday);
calendar.set(Calendar.MINUTE,minute);
                SimpleDateFormat simpleDateFormat=new

                SimpleDateFormat("HH:mm");

                time_in.setText(simpleDateFormat.format(calendar.getTime()));
            }
        };
        new
TimePickerDialog(MainActivity.this,timeSetListener,calendar.get(Calendar
.HOUR_OF_DAY),calendar.get(Calendar.MINUTE),false).show();
    }
    private void showDateDialog(EditText
        date_in){Calendar
        calendar=Calendar.getInstance();
        DatePickerDialog.OnDateSetListener

```

```

dateSetListener=newDatePickerDialog.OnDateSetListener() {
    @Override
    public void onDateSet(DatePicker datePicker, int
year, intmonth, int dayofmonth) {
        calendar.set(Calendar.YEAR,year);
        calendar.set(Calendar.MONTH,month);
        calendar.set(Calendar.DAY_OF_MONTH,dayofmonth);
        SimpleDateFormat simpleDateFormat=new
SimpleDateFormat("yy-
MM-dd");

        date_in.setText(simpleDateFormat.format(calendar.getTime()));
    }
};
new DatePickerDialog(MainActivity.this,dateSetListener,calendar.get
(Calendar.YEAR),calendar.get(Calendar.MONTH),calendar.get(Calendar.DAY_OF_MONTH))
.show(
);
}

```

DESIGN: -



PRACTICAL 7

Develop an simple application with context menu.

Android Context Menu:- Android context menu appears when user press long click on the element. It is also known as floating menu.

(ACTIVITY_MAIN.XML)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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"

    <ListView
        android:layout_width="368dp"
        android:layout_height="495dp"
        android:id="@+id/listView"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"
        android:layout_marginTop="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="h
ttp://schemas.android.com/apk/res/android"
```

MainActivity.java

```
package com.contextmenu;
import android.support.app.AppCompatActivity;
import android.os.Bundle;
import android.view.ContextMenu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    ListView listView;
    String contacts[]={"Ajay","Sachin","Sumit","Tarun","Yogesh"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        listView=(ListView)findViewById(R.id.listView);
        ArrayAdapter<String> adapter=new ArrayAdapter<String>(this,android
.R.layout.simple_list_item_1,contacts);
        listView.setAdapter(adapter);
        // Register the ListView for Context menu
        registerForContextMenu(listView);
    }
    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, Context
Menu.ContextMenuInfo menuInfo)
    {
        super.onCreateContextMenu(menu, v, menuInfo);
```

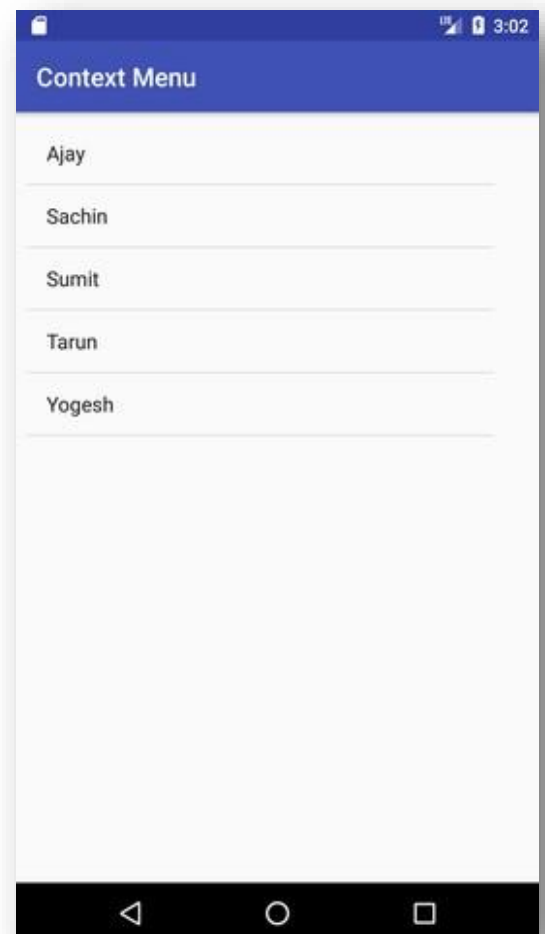


```

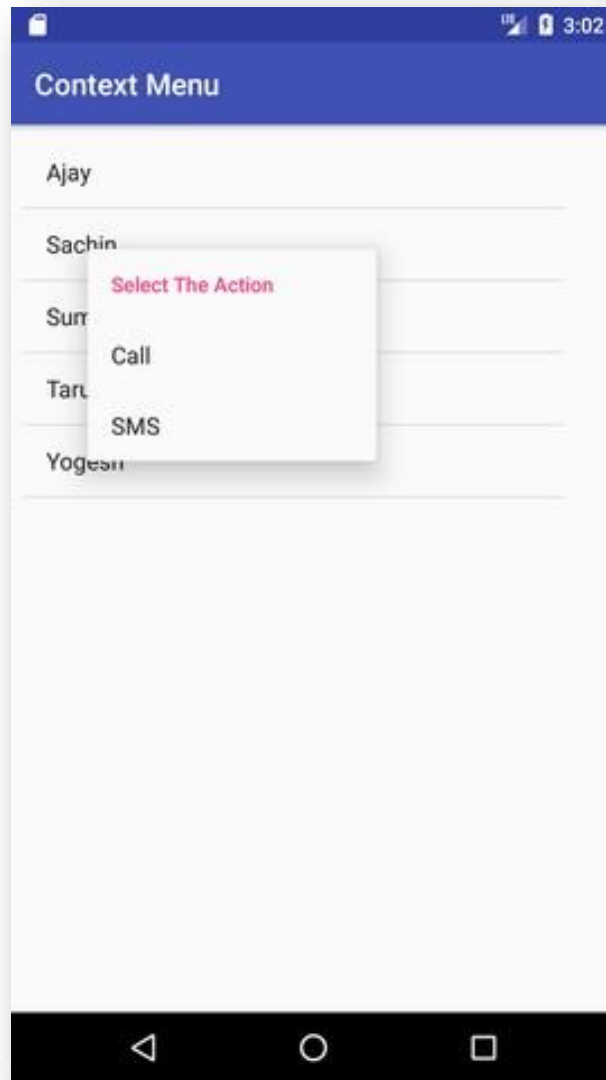
MenuInflater inflater = getMenuInflater();
inflater.inflate(R.menu.menu_main, menu);
menu.setHeaderTitle("Select The Action");
}
@Override
public boolean onOptionsItemSelected(MenuItem item){
    if(item.getItemId()==R.id.call){
        Toast.makeText(getApplicationContext(),"calling code",Toast.LENGTH
_LONG).show();
    }
    else if(item.getItemId()==R.id.sms){
        Toast.makeText(getApplicationContext(),"sending sms code",Toast.LE
NGTH_LONG).show();
    }else{
        return false;
    }
    return true;
}
}
}

```

DESIGN: -



Output after long press on Menu:



PRACTICAL 8

Develop an application to send SMS

(activity_main.xml):

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

    <EditText
        android:id="@+id/number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="phone number"
        android:inputType="phone"
        android:layout_margin="8dp"/>

    <EditText
        android:id="@+id/message"
        android:layout_width="match_parent"
        android:layout_height="100dp"
        android:hint="type message"
        android:inputType="textLongMessage"
        android:gravity="top"
        android:layout_margin="8dp"/>

    <Button
        android:id="@+id/send"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="send"
        android:gravity="center_horizontal"/>

</LinearLayout>
```

Main_Activity.java:

```
package com.example.sms;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

import android.widget.Toast

;import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    private EditText number,message;
    private Button send;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        number = findViewById(R.id.number);
        message =
        findViewById(R.id.message);send =
        findViewById(R.id.send);
        send.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (Build.VERSION.SDK_INT>=Build.VERSION_CODES.M){
                    if (checkSelfPermission(Manifest.permission.SEND_SMS)==
PackageManager.PERMISSION_GRANTED){

                        sendSMS();
                    }
                    else {
                        requestPermissions(new String[]
{Manifest.permission.SEND_SMS}, 1);
                    }
                }
            }
        });
    }

    private void sendSMS(){
        String phoneNo=number.getText().toString().trim();
        String SMS= message.getText().toString().trim();

        try {
            SmsManager smsManager=SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNo,null,SMS,null,null);
            Toast.makeText(this,"Message
issent",Toast.LENGTH_SHORT).show();

        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(this,"Failed to
sendmessage",Toast.LENGTH_SHORT).show();
        }
    }
}
```

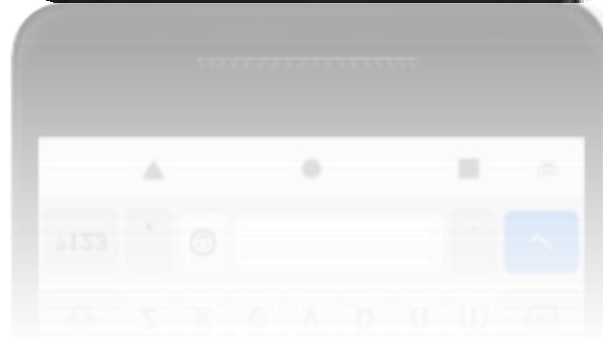
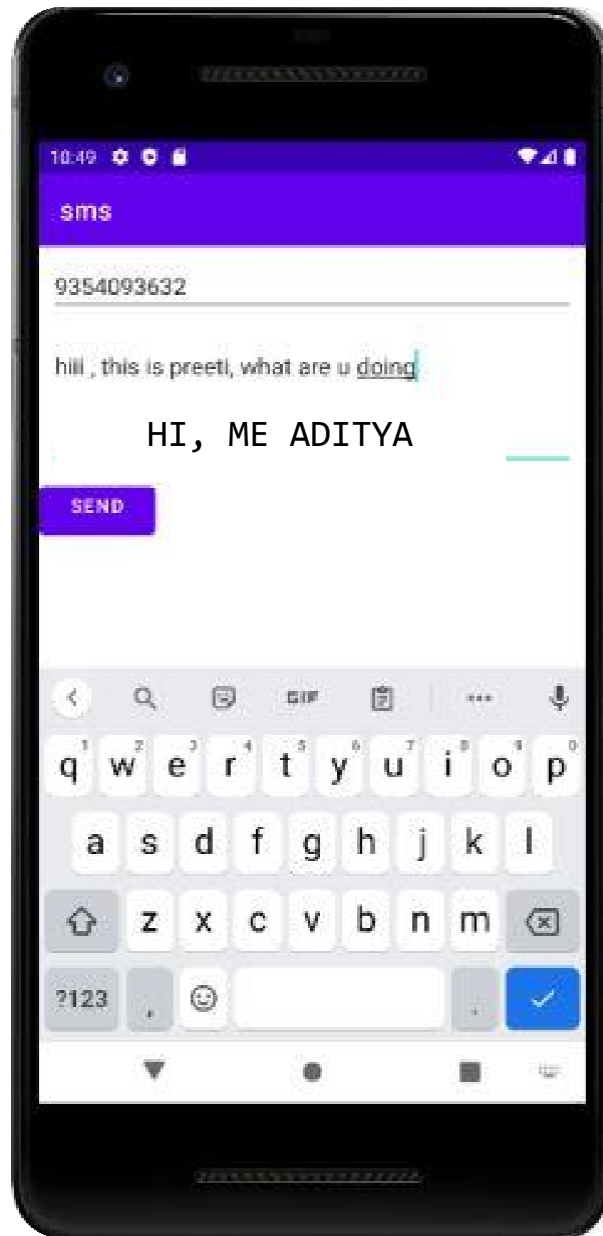
```
}
```

AndroidManifest.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/andro
id"xmlns:tools="http://schemas.android.com/tools"
    package="com.example.sms">
<uses-permission android:name="android.permission.SEND_SMS"/>
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extracti
on_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launche
r_round"android:supportsRtl="true"
    android:theme="@style/Theme.Sms"
    tools:targetApi="31">
<activity
    android:name=".MainAct
ivity"
    android:exported="true
">
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

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

Design:



PRACTICAL 9

Write A Program to View ,Edit Contact.

Main_Activity.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">

    <EditText
        android:id="@+id/editText"
        android:layout_width="230dp"
        android:layout_height="67dp"
        android:layout_marginStart="88dp"
        android:layout_marginTop="64dp"
        android:ems="10"
        android:hint="Enter Display Name ..."
        android:inputType="textPersonName"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <EditText
        android:id="@+id/editTextNumber"
        android:layout_width="228dp"
        android:layout_height="72dp"
        android:layout_marginStart="84dp"
        android:layout_marginTop="53dp"
        android:ems="10"
        android:hint="Enter New Number ..."
        android:inputType="number"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editText" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="108dp"
        android:layout_marginTop="56dp"
        android:onClick="buttonUpdateContact"
        android:text="Update Contact"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/editTextNumber" />
</androidx.constraintlayout.widget.ConstraintLayout>
```


Main_Activity.Java

```
package com.example.myapplication;

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

import android.Manifest;
import android.content.ContentProviderOperation;
import android.content.OperationApplicationException;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.os.RemoteException;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    private EditText editTextName, editTextNumber;

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

        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.WRITE_CONTACTS},
            PackageManager.PERMISSION_GRANTED);

        editTextName = findViewById(R.id.editText);
        editTextNumber = findViewById(R.id.editTextNumber);
    }

    public void buttonUpdateContact(View view){

        ArrayList<ContentProviderOperation> contentProviderOperations
            = new ArrayList<ContentProviderOperation>();

        contentProviderOperations.add(ContentProviderOperation
            .newUpdate(ContactsContract.Data.CONTENT_URI)
            .withSelection(ContactsContract.Data.DISPLAY_NAME + " = ? AND " +
                ContactsContract.Data.MIMETYPE + " = ?",
                new String[]{editTextName.getText().toString(),
                    ContactsContract.CommonDataKinds.Phone.CONTENT_ITEM_TYPE})
            .withValue(ContactsContract.CommonDataKinds.Phone.NUMBER,
                editTextNumber.getText().toString())
            .withValue(ContactsContract.CommonDataKinds.Phone.TYPE,
                ContactsContract.CommonDataKinds.Phone.TYPE_MOBILE)
            .build());
```

```

    try {
getContentResolver().applyBatch(ContactsContract.AUTHORITY, contentProviderOperations);
        Toast.makeText(this, "Contact Updated Successfully",
Toast.LENGTH_LONG).show();
    } catch (OperationApplicationException e) {
        e.printStackTrace();
    } catch (RemoteException e) {
        e.printStackTrace();
    }
}
}
}

```

Manifest File.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">

    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>

    <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/Theme.UpdateOrEditContacts">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>

```

Design:-

The screenshot shows an Android application interface with a blue header bar containing the text "Test_2016-08-22_add_new_Contact" and system icons for Wi-Fi, battery, and the time 6:00. Below the header, the text "Hello World!" is displayed. The main content area contains four input fields: "Firstname" with a text input field containing "Firstname", "Familyname" with a text input field containing "Familyname", "Telephone" with an empty text input field, and "Email:" with an empty text input field. A grey button labeled "ADD CONTACT" is positioned below the input fields. At the bottom of the screen, the standard Android navigation bar is visible with back, home, and recent apps icons.

PRACTICAL 10

Write a program to send e-mail

E-MAIL: E-mail is defined as the transmission of messages on the Internet. It is one of the most commonly used features over communications networks that may contain text, files, images, or other attachments. Generally, it is information that is stored on a computer sent through a network to a specified individual or group of individuals.

Email messages are conveyed through email servers; it uses multiple protocols within the TCP/IP suite.

Advantages of Email

- ≅ **Cost-effective**
- ≅ **Speed and simplicity**
- ≅ **Mass sending**

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:orientation="vertical" >
    <EditText
        android:id="@+id/txtTo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="To"/>
    <EditText
        android:id="@+id/txtSub"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Subject"/>
    <EditText
        android:id="@+id/txtMsg"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="top"
```

```

        android:hint="Message"/>
<Button
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:text="Send"
    android:id="@+id/btnSend"/>
</LinearLayout>

```

Main_Activity.Java

```

package com.example.myapplication;

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

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private EditText eTo;
    private EditText eSubject;
    private EditText eMsg;
    private Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        eTo = (EditText)findViewById(R.id.txtTo);
        eSubject = (EditText)findViewById(R.id.txtSub);
        eMsg = (EditText)findViewById(R.id.txtMsg);
        btn = (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(Intent.ACTION_SEND);
                it.putExtra(Intent.EXTRA_EMAIL, new String[]{eTo.getText().toString()});
                it.putExtra(Intent.EXTRA_SUBJECT, eSubject.getText().toString());
                it.putExtra(Intent.EXTRA_TEXT, eMsg.getText());
                it.setType("message/rfc822");
                startActivity(Intent.createChooser(it, "Choose Mail App"));
            }
        });
    }
}

```

DESIGN:

The image shows a mobile application interface for sending an email. At the top, there is a purple header bar with the text "Send e mail" in white. Below the header, the interface is divided into three sections: "To", "subject", and "message". Each section has a horizontal line indicating an input field. The "To" field is currently active, with a blue cursor and a blue underline. At the bottom of the screen, there is a purple button with the text "SEND" in white. The entire interface is framed by a black border, and the bottom of the screen shows the standard Android navigation bar with a back arrow, a home circle, and a recent apps square.

4:32 [Settings] [Google] [Shield] [Battery]

Send e mail

To

subject

message

SEND

PRACTICAL 11

Write a program to demonstrate a service

Service:- A `Service` is an application component that can perform long-running operations in the background. It does not provide a user interface.

Main_Activity.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"
android:background="#168BC34A"
tools:context=".MainActivity">

<LinearLayout
    android:id="@+id/linearLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_centerVertical="true"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0"
    tools:ignore="MissingConstraints">

<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="170dp"

    android:text="@string/heading"
    android:textAlignment="center"
    android:textAppearance="@style/TextAppearance.AppCompat.Large"
    android:textColor="@android:color/holo_green_dark"
    android:textSize="36sp"
    android:textStyle="bold" />

<Button
    android:id="@+id/startButton"
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginStart="20dp"
android:layout_marginTop="10dp"
android:layout_marginEnd="20dp"
android:layout_marginBottom="20dp"
android:background="#4CAF50"
```

```
android:text="@string/startButtonText"
android:textAlignment="center"
android:textAppearance="@style/TextAppearance.AppCompat.Display1"
android:textColor="#FFFFFF"
android:textStyle="bold" />
```

<Button

```
android:id="@+id/stopButton"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:layout_marginStart="20dp"
android:layout_marginTop="10dp"
android:layout_marginEnd="20dp"
android:layout_marginBottom="20dp"
android:background="#4CAF50"
```

```
android:text="@string/stopButtonText"
android:textAlignment="center"
android:textAppearance="@style/TextAppearance.AppCompat.Display1"
android:textColor="#FFFFFF"
android:textStyle="bold" />
```

</LinearLayout>

Activity_main.java

```
package com.example.myapplication;

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

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
```



```

// declaring objects of Button class
private Button start, stop;

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

    // assigning ID of startButton
    // to the object start
    start = (Button) findViewById( R.id.startButton );

    // assigning ID of stopButton
    // to the object stop
    stop = (Button) findViewById( R.id.stopButton );

    // declaring listeners for the
    // buttons to make them respond
    // correctly according to the process
    start.setOnClickListener( this );
    stop.setOnClickListener( this );
}

public void onClick(View view) {

    // process to be performed
    // if start button is clicked
    if(view == start){

        // starting the service
        startService(new Intent( this, NewService.class ) );
    }

    // process to be performed
    // if stop button is clicked
    else if (view == stop){

        // stopping the service
        stopService(new Intent( this, NewService.class ) );
    }
}
}
}

```

NEW SERVICE.XML

```
package com.example.myapplication;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
import android.provider.Settings;
import androidx.annotation.Nullable;

public class NewService extends Service {

    // declaring object of MediaPlayer
    private MediaPlayer player;

    @Override

    // execution of service will start
    // on calling this method
    public int onStartCommand(Intent intent, int flags, int startId) {

        // creating a media player which
        // will play the audio of Default
        // ringtone in android device
        player = MediaPlayer.create( this, Settings.System.DEFAULT_RINGTONE_URI
);

        // providing the boolean
        // value as true to play
        // the audio on loop
        player.setLooping( true );

        // starting the process
        player.start();

        // returns the status
        // of the program
        return START_STICKY;
    }

    @Override

    // execution of the service will
    // stop on calling this method
    public void onDestroy() {
        super.onDestroy();

        // stopping the process
        player.stop();
    }
}
```

```
    @Nullable
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

Manifest.xml

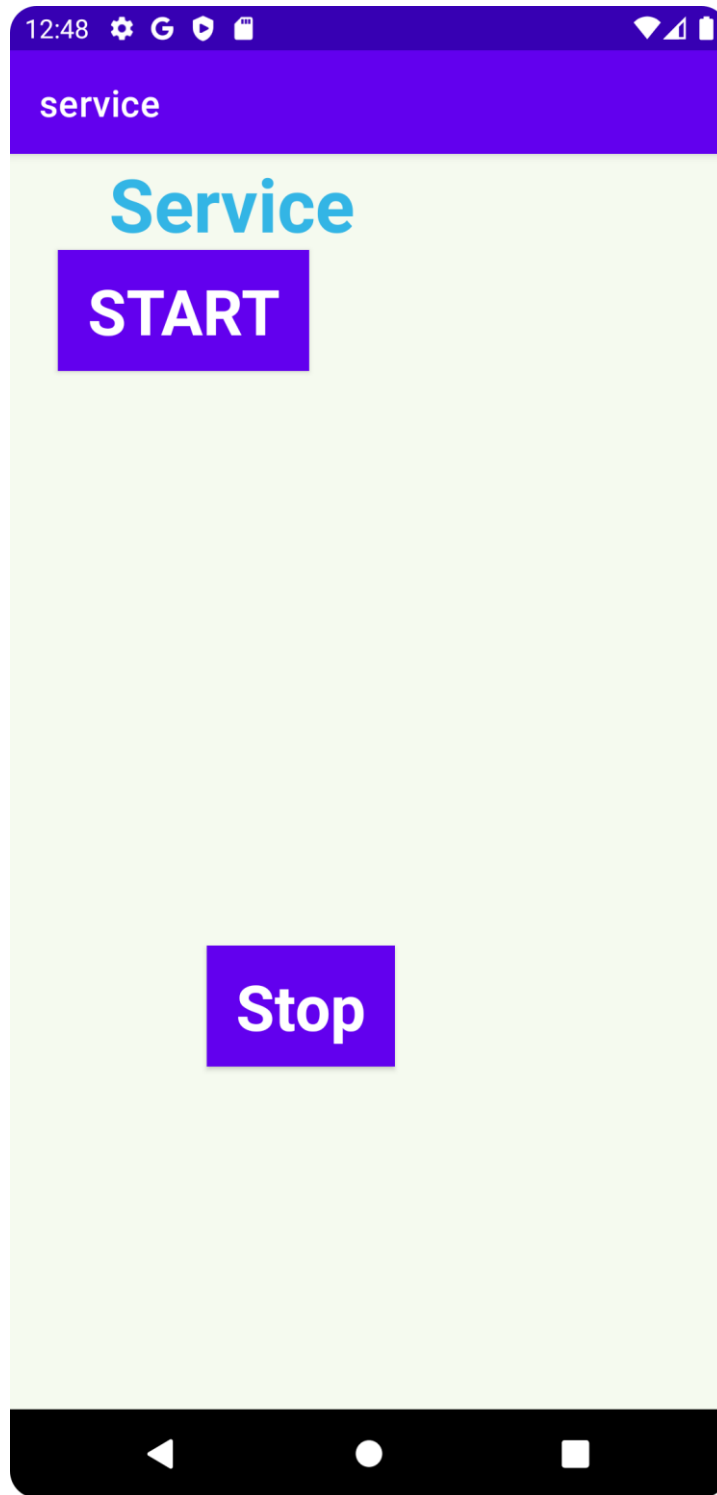
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapplication">

    <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/Theme.MyApplication">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

DESIGN:



PRACTICAL 12

Write a program to demonstrate web view to display web site.

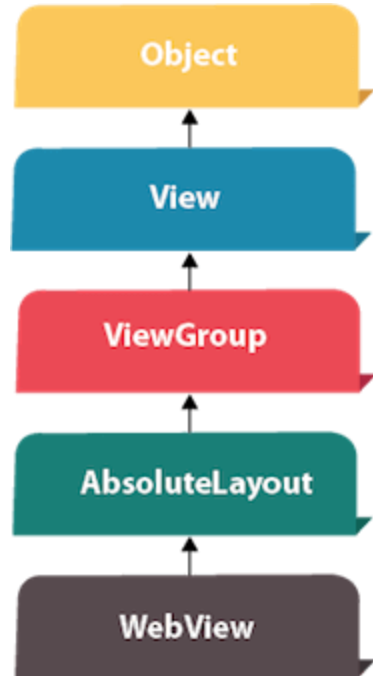
Android Web View

Android Web View is used to display web page in android. The web page can be loaded from same application or URL. It is used to display online content in android activity.

Android Web View uses webkit engine to display web page.

The `android.webkit.WebView` is the subclass of Absolute Layout class.

The **load Url()** and **load Data()** methods of Android Web View class are used to load and display web page.



activity_main.xml

File: activity_main.xml

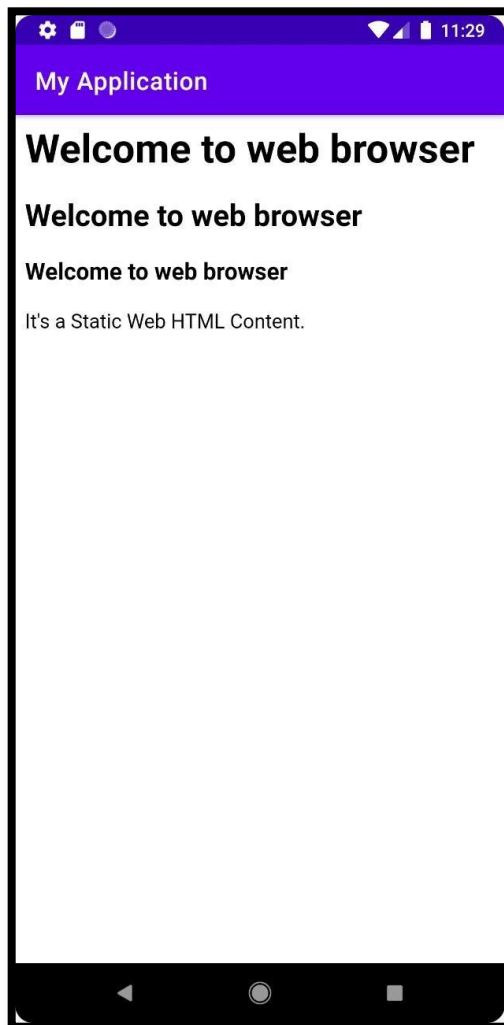
```
<?xml version="1.0" encoding="utf-8"?>
<WebView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/webview"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"/>
```

Activity class

File: MainActivity.java

```
activity_main.xml x MainActivity.kt x
1 package com.example.myapplication
2
3 import android.os.Bundle
4 import android.view.View
5 import android.webkit.WebView
6 import androidx.appcompat.app.AppCompatActivity
7
8 class MainActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_main)
12         val wv = findViewById<View>(R.id.webview) as WebView
13         val customHtml = "<html><body><h1>Welcome to web browser</h1>" +
14             "<h2>Welcome to web browser</h2><h3>Welcome to web browser</h3>" +
15             "<p>It's a Static Web HTML Content.</p>" +
16             "</body></html>"
17         wv.loadData(customHtml, mimeType: "text/html", encoding: "UTF-8")
18     }
19 }
```

OUTPUT:-



PRACTICAL 13

Write a program to display map of given location/position using map view.

Android Google Map :- Google Maps is a Web-based service that provides detailed information about geographical regions and sites around the world.

Activity_Main.xml

```
<fragment xmlns:android="http://schemas.android.com/apk/res/andro
id"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"

    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="example.com.mapexample.MapsActivity" />
```

Build.gradle

```
implementation 'com.google.android.gms:play-services-maps:17.0.0'
implementation 'com.google.android.gms:play-services-location:17.0.0'
```

Main_Activity.java

```
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
```

```
import android.os.Bundle;
import android.widget.Toast;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationServices;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.android.gms.tasks.Task;
import androidx.annotation.NonNull;
import androidx.core.app.ActivityCompat;
import androidx.fragment.app.FragmentActivity;
public class MainActivity extends FragmentActivity implements
OnMapReadyCallback {
    Location currentLocation;
    FusedLocationProviderClient fusedLocationProviderClient;
    private static final int REQUEST_CODE = 101
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        fusedLocationProviderClient =
LocationServices.getFusedLocationProviderClient(this);
        fetchLocation();
    }
    private void fetchLocation() {
        if (ActivityCompat.checkSelfPermission(
            this, Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(
            this, Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
```



```

        ActivityCompat.requestPermissions(this, new
String[] {Manifest.permission.ACCESS_FINE_LOCATION}, REQUEST_CODE);
        return;
    }
    Task<Location> task = fusedLocationProviderClient.getLastLocation();
    task.addOnSuccessListener(new OnSuccessListener<Location>() {
        @Override
        public void onSuccess(Location location) {
            if (location != null) {
                currentLocation = location;
                Toast.makeText(getApplicationContext(),
currentLocation.getLatitude() + "" + currentLocation.getLongitude(),
Toast.LENGTH_SHORT).show();
                SupportMapFragment supportMapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.myMap);
                assert supportMapFragment != null;
                supportMapFragment.getMapAsync(MainActivity.this);
            }
        }
    });
}
@Override
public void onMapReady(GoogleMap googleMap) {
    LatLng latLng = new LatLng(currentLocation.getLatitude(),
currentLocation.getLongitude());
    MarkerOptions markerOptions = new
MarkerOptions().position(latLng).title("I am here!");
    googleMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
    googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 5));
    googleMap.addMarker(markerOptions);
}
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[]
permissions, @NonNull int[] grantResults) {

```

```

switch (requestCode) {
    case REQUEST_CODE:
        if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            fetchLocation();
        }
        break;
    }
}
}
}

```

Strings.xml

```

<resources>
    <string name="app_name">Sample</string>
    <string name="map_key" translatable="false">Enter your google API key
here</string>
</resources>

```

AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="app.com.sample">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <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">

```

```
<meta-data android:name="com.google.android.geo.API_KEY"
android:value="@string/map_key"/>
<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>
```

DESIGN: -

