

- Install Android Studio and Run Hello World

Android Studio is Google's IDE for Android apps. It contains tools for development, debugging, testing, and performance

that make it faster and easier to develop apps.

We can test our apps with a large range of preconfigured emulators or on your own mobile device, and build production APKs for publication.

For the latest information on system requirements and installation instructions, refer to the documentation at developer.android.com.

To run with Android Studio:

We may need to install the Java Development Kit - Java 7 or better.

Install Android Studio

Android Studio is available for Windows, Mac, and Linux computers. The installation is similar for all platforms.

1.1. Installing the Java Development Kit

On your computer, open a terminal window.

Type `java -version`

The output includes a line:

```
Java(TM) SE Runtime Environment (build1.X.0_05-b13)
```

X is the version number to look at.

If this is 7 or greater, you can move on to installing Android Studio.

If you see a Java SE version is below 7 or if Java is not installed, you need to install the latest version of the

Java SE development kit before installing Android Studio.

To download the Java Standard Edition () Development Kit (JDK):

Go to the [Oracle Java SE downloads page](http://www.oracle.com/technetwork/java/javase-downloads).

Click the Java SE Downloads icon to open the Java SE Development Kit 8 Downloads page.

In the box for the latest Java SE Development kit, you need to accept the License Agreement in order to proceed. Then download the version appropriate for the computer you are developing on.

Important: Do not go to the demos and samples (the menus look very similar, so make sure to read the heading at the top).

Install the development kit. Once the installation of the JDK is completed — it should only take a few minutes — you can confirm it's correct by checking the Java version from the command line.

Open a terminal window and enter `Type java -version` again to verify that installation has been successful.

Set the `JAVA_HOME` environment variable to the installation directory of the JDK.

Windows:

Set `JAVA_HOME` to the installation location.

Start > Control Panel > System > Advanced System Settings > Environment Variables System Variables > New

Variable name: `JAVA_HOME`

Variable value: `C:\Program Files\Java\jdk1.7.0_80` (or whatever version your installation is!)

If the variable already exists, update it to this version of the JDK.

Verify your `JAVA_HOME` variable from a `cmd.exe` terminal: `echo %JAVA_HOME%`

See also: https://docs.oracle.com/cd/E19182-01/820-7851/inst_cli_jdk_javahome_t/

1.2. Installing Android Studio

Navigate to the Android developers site and follow the instructions to download and install Android Studio.

Accept the default configurations for all steps.

Make sure that all components are selected for installation.

After finishing the install, the Setup Wizard will download and install some additional components.

Be patient, this might take some time depending on your Internet speed, and some of the steps may seem redundant.

When the download completes, Android Studio will start, and you are ready to create your first project

MainActivity.java

```
package com.example.ncet.myapplication;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

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

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="com.example.ncet.myapplication.MainActivity">

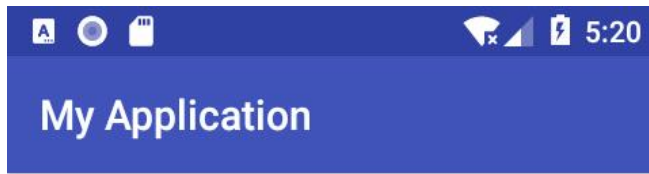
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="150dp"
        android:text="Hello World!"

    />
```

```
<ImageView
    android:layout_width="200dp"
    android:layout_height="100dp"
    android:layout_marginLeft="100dp"
    android:layout_marginRight="200dp"
    android:layout_marginTop="40dp"
    android:src="@drawable/images" />
```

```
</LinearLayout>
```

Output:-



Hello World!



2. Create and Start Activity Lifecycle and Instance State

MainActivity.java

```
package com.example.ncet.activitylifecycle;

import android.provider.Settings;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
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(this, "activity created", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onStart() {
        super.onStart();
        Toast.makeText(this, "activity started", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onResume() {
        super.onResume();
        Toast.makeText(this, "activity resumed", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onPause() {
        super.onPause();
        Toast.makeText(this, "activity paused", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onStop() {
        super.onStop();
        Toast.makeText(this, "activity stopped", Toast.LENGTH_SHORT).show();
    }
}
```

```

    }

    @Override
    protected void onRestart() {
        super.onRestart();
        Toast.makeText(this, "activity restarted", Toast.LENGTH_SHORT).show();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        Toast.makeText(this, "Destroyed", Toast.LENGTH_SHORT).show();
    }
}

```

activity_main.xml

```

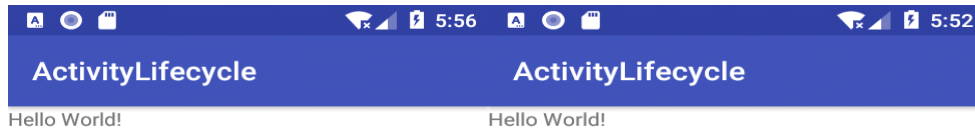
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
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

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

</LinearLayout>

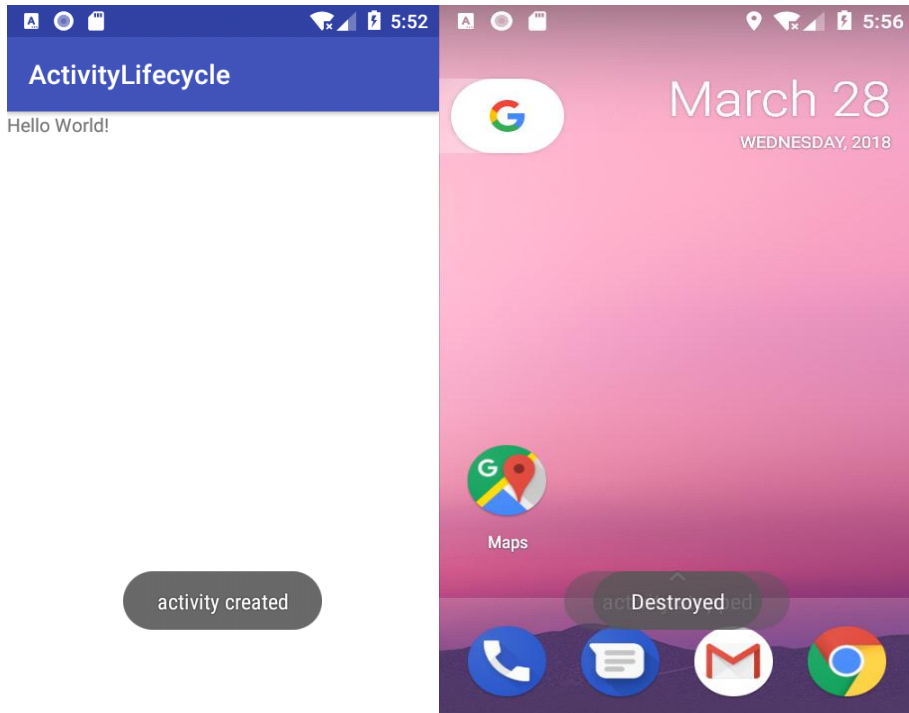
```

Output:-



activity started

activity resumed



3. Create implicit intents

Activity1.java

```

package com.example.ncet.implicitintent;

import android.content.Intent;
import android.net.Uri;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class Activity1 extends AppCompatActivity {
    EditText editText1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_1);
        final EditText editText1=(EditText)findViewById(R.id.editText1);
        Button button1=(Button)findViewById(R.id.button1);
        button1.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
                String url=editText1.getText().toString();
                Intent intent=new Intent(Intent.ACTION_VIEW);
                intent.setData(Uri.parse("http://www.nagarjunareresults.com"));
                startActivity(intent);
            }
        });
    }
}

```

activity_1.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="com.example.ncet.implicitintent.Activity1">

    <EditText
        android:id="@+id/editText1"
        android:layout_width="157dp"
        android:layout_height="wrap_content" />

    <Button

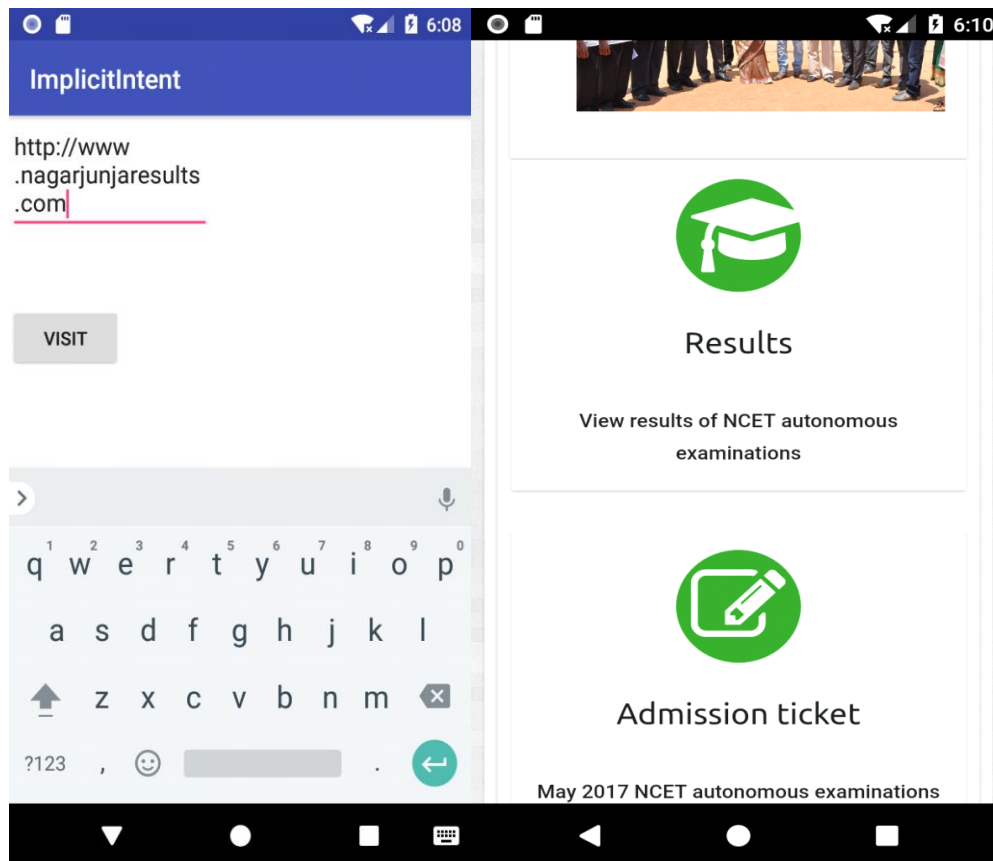
```



```
android:id="@+id/button1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_below="@+id/editText1"  
android:layout_centerHorizontal="true"  
android:layout_marginTop="54dp"  
android:text="Visit" />
```

</LinearLayout>

Output:-



4. Make your First Interactive UI Using Layouts and Text View Elements

MainActivity.java

```
package com.example.ncet.login;
```

```
import android.support.v7.app.AppCompatActivity;
```

```

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

public class MainActivity extends AppCompatActivity {
    EditText username,password;

    Button b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username = (EditText) findViewById(R.id.t1);
        password = (EditText) findViewById(R.id.t2);
        b = (Button) findViewById(R.id.b1);

        b.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {

                if(username.getText().toString().equals("admin") &&
password.getText().toString().equals("123456"))
                    Toast.makeText(getApplicationContext(),"logged
in",Toast.LENGTH_SHORT).show();// Set your own toast message
                else
                    Toast.makeText(getApplicationContext(),"logged
out",Toast.LENGTH_SHORT).show()

            }

        });

    }

}

```

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="com.example.ncet.login.MainActivity"  
android:background="#DAAA" >
```

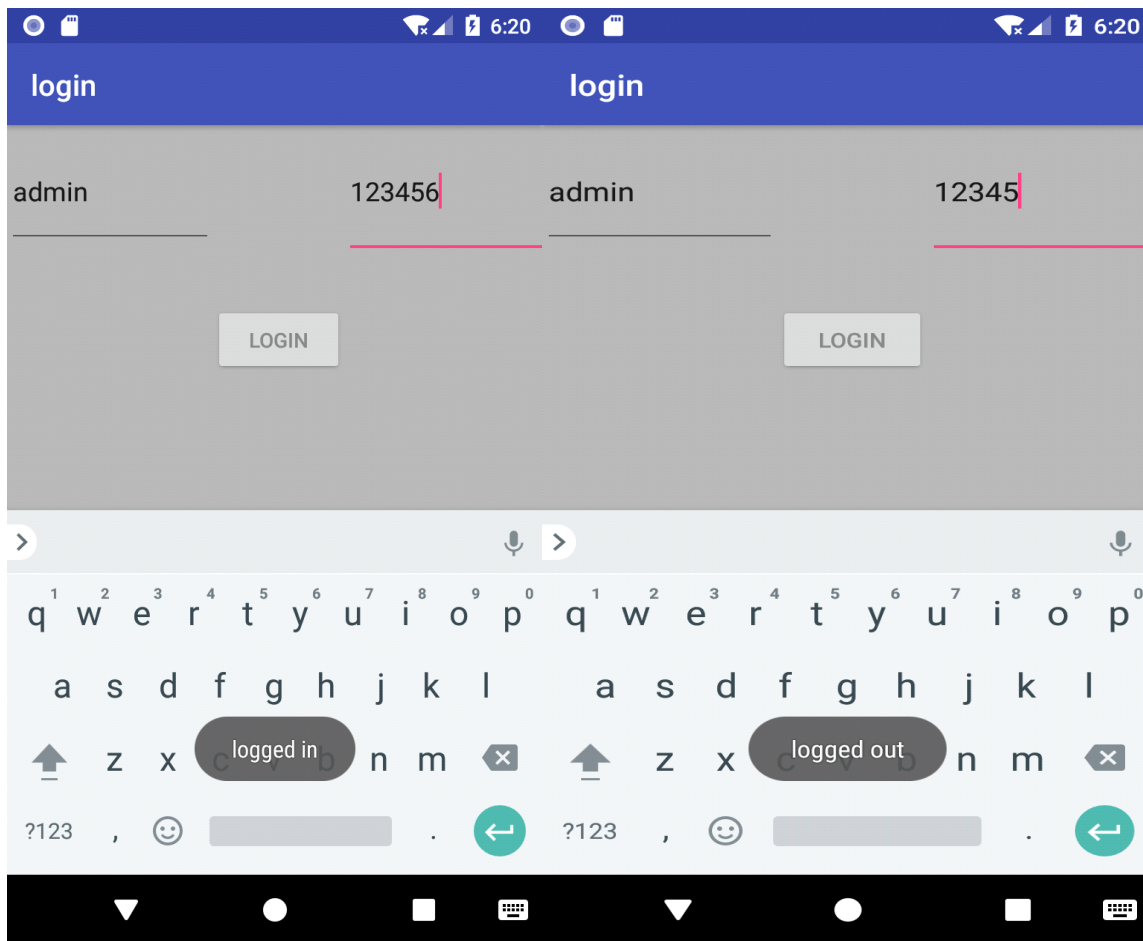
```
<EditText  
    android:id="@+id/t1"  
    android:layout_width="wrap_content"  
    android:layout_height="77dp"  
    android:hint="enter user name" />
```

```
<Button  
    android:id="@+id/b1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="100dp"  
    android:hint="LOGIN"  
    />
```

```
<EditText  
    android:id="@+id/t2"  
    android:layout_width="200dp"  
    android:layout_height="91dp"  
    android:hint="password" />
```

```
</LinearLayout>
```

Output:-



5. Using an Options Menu

MainActivity.java

```
package com.example.ncet.optionsmenu;

import android.content.DialogInterface;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
```

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

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add(0,0,0,"Show alter dialog");
    menu.add(0,1,1,"Show Toast Message");
    menu.add(0,2,2,"Close Activity");
    return true;
}
```

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()){
        case 0: {
            AlertDialog.Builder dialogBuilder = new AlertDialog.Builder(MainActivity.this);
            dialogBuilder.setMessage("Do you want to close the app?")
                .setCancelable(false)

                .setPositiveButton("Yes", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        finish();
                    }
                })
                .setNegativeButton("No", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        dialog.cancel();
                    }
                });

            AlertDialog alert = dialogBuilder.create();
            alert.setTitle("AlertDialogExample");
            alert.show();
        }
        break;
        case 1:
            Toast.makeText(MainActivity.this,"This is a Toast
Message",Toast.LENGTH_LONG).show();
            break;
        case 2:
            finish();
            break;
        default:
```

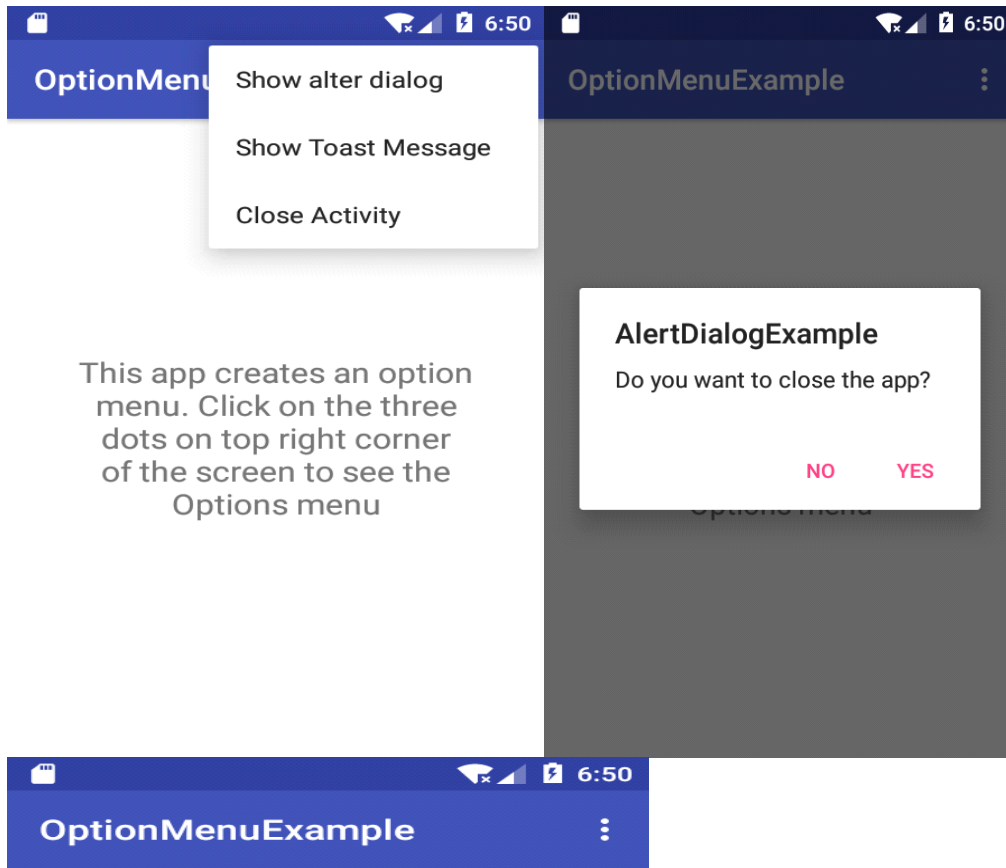
```
        break;
    }
    return false;
}
}
```

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="wrap_content"
    android:layout_gravity="center"
    android:orientation="vertical"
    tools:context="com.example.ncet.optionsmenu.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:padding="20dp"
        android:gravity="center_horizontal"
        android:text="@string/description"
        />
</LinearLayout>
```

Output:-



This app creates an option menu. Click on the three dots on top right corner of the screen to see the Options menu

This app creates an option menu. Click on the three dots on top right corner of the screen to see the Options menu

This is a Toast Message

6. Set and retrieve shared preferences

MainActivity.java

```

package com.example.ncet.sharedpreferences;

import android.app.Activity;
import android.content.Context;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends Activity {
    SharedPreferences sharedPreferences;
    TextView name;
    TextView email;
    public static final String mypreference = "mypref";
    public static final String Name = "nameKey";
    public static final String Email = "emailKey";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        name = (TextView) findViewById(R.id.etName);
        email = (TextView) findViewById(R.id.etEmail);
        sharedPreferences = getSharedPreferences(mypreference,
            Context.MODE_PRIVATE);
        if (sharedPreferences.contains(Name)) {
            name.setText(sharedPreferences.getString(Name, ""));
        }
        if (sharedPreferences.contains(Email)) {
            email.setText(sharedPreferences.getString(Email, ""));
        }
    }

    public void Save(View view) {
        String n = name.getText().toString();
        String e = email.getText().toString();
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString(Name, n);
        editor.putString(Email, e);
        editor.commit();
    }

    public void clear(View view) {

```



```

name = (TextView) findViewById(R.id.etName);
email = (TextView) findViewById(R.id.etEmail);
name.setText("");
email.setText("");

}

public void Get(View view) {
name = (TextView) findViewById(R.id.etName);
email = (TextView) findViewById(R.id.etEmail);
sharedpreferences = getSharedPreferences(mypreference,
Context.MODE_PRIVATE);

if (sharedpreferences.contains(Name)) {
name.setText(sharedpreferences.getString(Name, ""));
}
if (sharedpreferences.contains(Email)) {
email.setText(sharedpreferences.getString(Email, ""));
}
}
}
}
}

```

activity_main.xml

```

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

<Button
android:id="@+id/btnSave"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerVertical="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:onClick="Save"
android:text="Save" />

<Button
android:id="@+id/btnRetr"
android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:onClick="Get"
android:text="Retrieve" />
```

```
<Button
    android:id="@+id/btnClear"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/etEmail"
    android:layout_centerVertical="true"
    android:onClick="clear"
    android:text="Clear" />
```

```
<EditText
    android:id="@+id/etEmail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Email"
    android:inputType="textEmailAddress"
    android:layout_below="@+id/etName"
    android:layout_marginTop="20dp"
/>
```

```
<EditText
    android:id="@+id/etName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Name"
    android:inputType="text"
/>
```

```
</RelativeLayout>
```

Output:-



Name priya

Email shanmugapriya09@gmail.com



7. Implement a simple content provider

MainActivity.java

```
package com.example.ncet.cp;

import android.Manifest;
import android.database.Cursor;
import android.provider.ContactsContract;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
```

```

import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ListView;

import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ListView lv;
    ArrayList al=new ArrayList();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActivityCompat.requestPermissions(MainActivity.this,new String[]
{Manifest.permission.READ_CONTACTS},1);

        setContentView(R.layout.activity_main);
        lv=(ListView)findViewById(R.id.list);

        Cursor
c=getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT_URI,null,n
ull,null,null);
        while (c.moveToNext()){

            String
name=c.getString(c.getColumnIndex(ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME)
);
            al.add(name);
        }
lv.setAdapter(new
ArrayAdapter(MainActivity.this,android.R.layout.simple_list_item_1,al));
    }
}

```

activity_main.xml

```

<RelativeLayout 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:paddingTop="16dp"
    android:paddingRight="16dp"
    android:paddingLeft="16dp"
    android:paddingBottom="16dp"
    tools:context="com.example.ncet.cp.MainActivity">

    <ListView

```

```
        android:id="@+id/list"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp" />
</RelativeLayout>
```

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ncet.cp">
<uses-permission android:name="android.permission.READ_CONTACTS"/> //include this
line//
    <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">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

Output:-

