

### 31) Switch and Toggle demo

#### SwitchToggleActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.CompoundButton;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivitySwitchToggleBinding;

public class SwitchToggleActivity extends AppCompatActivity {
    private ActivitySwitchToggleBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivitySwitchToggleBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.sw.setOnCheckedChangeListener((CompoundButton buttonView,
boolean isChecked) ->
            Toast.makeText(SwitchToggleActivity.this, "Switch: " +
isChecked, Toast.LENGTH_SHORT).show()
        );

        b.toggle.setOnCheckedChangeListener((CompoundButton buttonView,
boolean isChecked) ->
            Toast.makeText(SwitchToggleActivity.this, "Toggle: " +
isChecked, Toast.LENGTH_SHORT).show()
        );
    }
}
```

#### activity\_switch\_toggle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
```

```

android:layout_height="match_parent">
    <Switch android:id="@+id/sw" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Switch"/>
    <ToggleButton android:id="@+id/toggle"
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:textOff="Off"
android:textOn="On"/>
</LinearLayout>

```

### 32) TimePicker demo

#### TimePickerActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;
import android.widget.TimePicker;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityTimePickerBinding;

public class TimePickerActivity extends AppCompatActivity {
    private ActivityTimePickerBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityTimePickerBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.timePicker.setIs24HourView(true);
        b.btnGet.setOnClickListener(v -> {
            int h = b.timePicker.getHour();
            int m = b.timePicker.getMinute();
            Toast.makeText(this, "Selected: " + h + ":" +
String.format("%02d", m), Toast.LENGTH_SHORT).show();
        });
    }
}

```

#### activity\_time\_picker.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent">
    <TimePicker android:id="@+id/timePicker"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnGet"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Get Time"/>
</LinearLayout>

```

### 33) Spinner demo

#### SpinnerActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivitySpinnerBinding;

public class SpinnerActivity extends AppCompatActivity {
    private ActivitySpinnerBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivitySpinnerBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        String[] items = {"Red", "Green", "Blue"};
        b.spinner.setAdapter(new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_dropdown_item, items));
        b.spinner.setOnItemClickListener(new
        AdapterView.OnItemClickListener(){
            @Override public void onItemClick(AdapterView<?> parent,
            View view, int position, long id){
                Toast.makeText(SpinnerActivity.this, "Selected: " +
                items[position], Toast.LENGTH_SHORT).show();
            }
            @Override public void onNothingSelected(AdapterView<?>
            parent){}
        })
    }
}

```

```

        });
    }
}

```

#### activity\_spinner.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <Spinner android:id="@+id/spinner"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>

```

### 34) Working Thread using Runnable

#### RunnableThreadActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityRunnableThreadBinding;

public class RunnableThreadActivity extends AppCompatActivity {
    private ActivityRunnableThreadBinding b;
    private boolean running = false;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityRunnableThreadBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnStart.setOnClickListener(v -> {
            if (running) return;
            running = true;
            b.progressBar.setProgress(0);
            new Thread(() -> {

```

```

        for (int i=1;i<=100 && running;i++){
            final int val = i;
            runOnUiThread(() ->
b.progressBar.setProgress(val));
            try { Thread.sleep(50); } catch
(InterruptedExcepcion ignored) {}
        }
        runOnUiThread(() ->
Toast.makeText(RunnableThreadActivity.this, "Done",
Toast.LENGTH_SHORT).show());
        running = false;
    }).start();
});

    b.btnStop.setOnClickListener(v -> running = false);
}
}

```

#### activity\_runnable\_thread.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <ProgressBar android:id="@+id/progressBar"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnStart"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Start"/>
    <Button android:id="@+id/btnStop"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Stop"/>
</LinearLayout>

```

### 35) Geocoding - get address from lat,long (uses Geocoder)

#### GeocodeActivity.java

```

package com.example.madpracticals;

import android.location.Address;

```

```

import android.location.Geocoder;
import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityGeocodeBinding;

import java.util.List;
import java.util.Locale;

public class GeocodeActivity extends AppCompatActivity {
    private ActivityGeocodeBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityGeocodeBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnFind.setOnClickListener(v -> {
            try {
                double lat =
Double.parseDouble(b.etLat.getText().toString().trim());
                double lon =
Double.parseDouble(b.etLon.getText().toString().trim());
                Geocoder g = new Geocoder(this, Locale.getDefault());
                List<Address> list = g.getFromLocation(lat, lon, 1);
                if (list != null && !list.isEmpty()) {
                    Address a = list.get(0);
                    b.tv.setText(a.getAddressLine(0));
                } else b.tv.setText("No address found");
            } catch (Exception e) {
                e.printStackTrace();
                Toast.makeText(this, "Invalid input or Geocoder not
available", Toast.LENGTH_SHORT).show();
            }
        });
    }
}

```

#### activity\_geocode.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"

```

```

android:layout_height="match_parent">
    <EditText android:id="@+id/etLat" android:hint="Latitude"
android:inputType="numberDecimal" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <EditText android:id="@+id/etLon" android:hint="Longitude"
android:inputType="numberDecimal" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnFind"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Find Address"/>
    <TextView android:id="@+id/tv" android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_marginTop="8dp"/>
</LinearLayout>

```

### 36) Phone call using Intent (opens dialer)

#### CallActivity.java

```

package com.example.madpracticals;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityCallBinding;

public class CallActivity extends AppCompatActivity {
    private ActivityCallBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityCallBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnCall.setOnClickListener(v -> {
            String num = b.etNumber.getText().toString().trim();
            if (num.isEmpty()) return;
            Intent i = new Intent(Intent.ACTION_DIAL, Uri.parse("tel:"
+ Uri.encode(num)));
            startActivity(i);
        });
    }
}

```

#### activity\_call.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etNumber" android:hint="Phone number"
android:inputType="phone" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnCall"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Call"/>
</LinearLayout>
```

#### 37) DatePicker demo

##### DatePickerActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.DatePicker;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityDatePickerBinding;

public class DatePickerActivity extends AppCompatActivity {
    private ActivityDatePickerBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityDatePickerBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnGet.setOnClickListener(v -> {
            int d = b.datePicker.getDayOfMonth();
            int m = b.datePicker.getMonth() + 1;
            int y = b.datePicker.getYear();
            Toast.makeText(this, "Selected: " + d + "/" + m + "/" + y,
Toast.LENGTH_SHORT).show();
        });
    }
}
```



#### activity\_date\_picker.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <DatePicker android:id="@+id/datePicker"
android:layout_width="wrap_content"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnGet"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Get Date"/>
</LinearLayout>
```

### 38) JSON Employee (alternative) - parse JSON and show in ListView

#### JsonAltActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityJsonAltBinding;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;

public class JsonAltActivity extends AppCompatActivity {
    private ActivityJsonAltBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityJsonAltBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        String json =
        "[{"id":1,"name":"Aman"},{"id":2,"name":"Riya"},{"id":3,"name":"Rahul"}
        ]";

        ArrayList<String> arr = new ArrayList<>();
        try {
            JSONArray a = new JSONArray(json);
            for (int i=0;i<a.length();i++){
```

```

        JSONObject o = a.getJSONObject(i);
        arr.add(o.getInt("id") + " - " + o.getString("name"));
    }
} catch (Exception e){ e.printStackTrace(); }
b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
}
}

```

#### activity\_json\_alt.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent"/>
</LinearLayout>

```

### 39) Working Thread using Handler (post updates to UI)

#### HandlerThreadActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivityHandlerThreadBinding;

public class HandlerThreadActivity extends AppCompatActivity {
    private ActivityHandlerThreadBinding b;
    private Handler handler = new Handler(Looper.getMainLooper());

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityHandlerThreadBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
    }
}

```

```

        b.btnStart.setOnClickListener(v -> {
            b.tv.setText("Working...");
            new Thread(() -> {
                for (int i=1;i<=5;i++){
                    int step = i;
                    handler.post(() -> b.tv.append("\nStep " + step));
                    try { Thread.sleep(500); } catch
(InterruptedExcepcion ignored) {}
                }
                handler.post(() -> b.tv.append("\nDone"));
            }).start();
        });
    }
}

```

#### activity\_handler\_thread.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <TextView android:id="@+id/tv" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnStart"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Start"/>
</LinearLayout>

```

#### 40) First class students - SQLite query (percentage > threshold)

##### StudentsDb.java

```

package com.example.madpracticals;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import androidx.annotation.Nullable;

public class StudentsDb extends SQLiteOpenHelper {

```

```

        private static final String DB_NAME = "students2.db";
        private static final int DB_VER = 1;
        private static final String TABLE = "Student";

        public StudentsDb(@Nullable Context context) { super(context,
DB_NAME, null, DB_VER); }

        @Override
        public void onCreate(SQLiteDatabase db) {
            db.execSQL("CREATE TABLE " + TABLE + " (roll INTEGER PRIMARY
KEY, name TEXT, percent REAL)");
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldV, int newV) {
            db.execSQL("DROP TABLE IF EXISTS " + TABLE);
            onCreate(db);
        }

        public void insert(int roll, String name, double p) {
            ContentValues cv = new ContentValues();
            cv.put("roll", roll); cv.put("name", name); cv.put("percent",
p);
            getWritableDatabase().insert(TABLE, null, cv);
        }

        public Cursor getFirstClass(double minPercent) {
            return getReadableDatabase().rawQuery("SELECT * FROM " + TABLE
+ " WHERE percent >= ? ORDER BY percent DESC", new
String[] {String.valueOf(minPercent)});
        }
    }
}

```

#### [FirstClassActivity.java](#)

```

package com.example.madpracticals;

import android.database.Cursor;
import android.os.Bundle;
import android.widget.AdapterView;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityFirstClassBinding;
import java.util.ArrayList;

public class FirstClassActivity extends AppCompatActivity {

```

```

private ActivityFirstClassBinding b;
private StudentsDb db;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    b = ActivityFirstClassBinding.inflate(getLayoutInflater());
    setContentView(b.getRoot());
    db = new StudentsDb(this);

    // sample inserts if empty
    Cursor c = db.getFirstClass(0);
    if (!c.moveToFirst()) {
        db.insert(1,"A",85.0);
        db.insert(2,"B",72.0);
        db.insert(3,"C",91.5);
    }
    c.close();

    b.btnShow.setOnClickListener(v -> {
        Cursor cur = db.getFirstClass(75.0);
        ArrayList<String> arr = new ArrayList<>();
        while (cur.moveToNext()) arr.add(cur.getInt(0) + " - " +
cur.getString(1) + " - " + cur.getDouble(2));
        cur.close();
        b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
    });
}
}

```

#### activity\_first\_class.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <Button android:id="@+id/btnShow"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Show First Class"/>
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"

```

```
android:layout_height="match_parent"/>
</LinearLayout>
```

#### 41) Dynamic Spinner (populate programmatically)

##### DynamicSpinnerActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.AdapterView;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityDynamicSpinnerBinding;
import java.util.ArrayList;

public class DynamicSpinnerActivity extends AppCompatActivity {
    private ActivityDynamicSpinnerBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityDynamicSpinnerBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        ArrayList<String> items = new ArrayList<>();
        items.add("One"); items.add("Two"); items.add("Three");
        b.spinner.setAdapter(new ArrayAdapter<>(this,
        android.R.layout.simple_spinner_dropdown_item, items));
    }
}
```

##### activity\_dynamic\_spinner.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="12dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <Spinner android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

## 42) AutoCompleteTextView demo

### AutoCompleteActivity.java

```
package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityAutoCompleteBinding;

public class AutoCompleteActivity extends AppCompatActivity {
    private ActivityAutoCompleteBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityAutoCompleteBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        String[] names = {"Aman", "Aadil", "Aarav", "Riya", "Rohit"};
        b.auto.setAdapter(new ArrayAdapter<>(this,
        android.R.layout.simple_dropdown_item_1line, names));
        b.auto.setThreshold(1);
    }
}
```

### activity\_auto\_complete.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:padding="12dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <AutoCompleteTextView android:id="@+id/auto"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

## 43) Square of number - compute and show in another activity

### SquareInputActivity.java

```
package com.example.madpracticals;

import android.content.Intent;
```

```

import android.os.Bundle;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivitySquareInputBinding;

public class SquareInputActivity extends AppCompatActivity {
    private ActivitySquareInputBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivitySquareInputBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());

        b.btnCompute.setOnClickListener(v -> {
            String s = b.etNum.getText().toString().trim();
            if (s.isEmpty()) { Toast.makeText(this, "Enter
number", Toast.LENGTH_SHORT).show(); return; }
            int n = Integer.parseInt(s);
            Intent i = new Intent(this, SquareResultActivity.class);
            i.putExtra("result", n * n);
            startActivity(i);
        });
    }
}

```

#### **SquareResultActivity.java**

```

package com.example.madpracticals;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivitySquareResultBinding;

public class SquareResultActivity extends AppCompatActivity {
    private ActivitySquareResultBinding b;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivitySquareResultBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        int res = getIntent().getIntExtra("result", 0);
        b.tv.setText("Square = " + res);
    }
}

```



```

    }
}

```

#### activity\_square\_input.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <EditText android:id="@+id/etNum" android:inputType="number"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
    <Button android:id="@+id/btnCompute"
android:layout_width="match_parent"
android:layout_height="wrap_content" android:text="Compute"/>
</LinearLayout>

```

#### activity\_square\_result.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <TextView android:id="@+id/tv" android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</LinearLayout>

```

### 44) JSON Student Data List - parse and show

#### JsonStudentListActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;
import android.widget.ArrayAdapter;
import androidx.appcompat.app.AppCompatActivity;
import
com.example.madpracticals.databinding.ActivityJsonStudentListBinding;
import org.json.JSONArray;
import org.json.JSONObject;
import java.util.ArrayList;

public class JsonStudentListActivity extends AppCompatActivity {

```

```

        private ActivityJsonStudentListBinding b;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            b =
ActivityJsonStudentListBinding.inflate(getLayoutInflater());
            setContentView(b.getRoot());

            String json =
"{"students":[{"roll":1,"name":"A"}, {"roll":2,"name":"B"}]}";
            ArrayList<String> arr = new ArrayList<>();
            try {
                JSONObject root = new JSONObject(json);
                JSONArray s = root.getJSONArray("students");
                for (int i=0;i<s.length();i++){
                    JSONObject o = s.getJSONObject(i);
                    arr.add(o.getInt("roll") + " - " +
o.getString("name"));
                }
            } catch (Exception e){ e.printStackTrace(); }
            b.listView.setAdapter(new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, arr));
        }
}

```

#### activity\_json\_student\_list.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:padding="12dp"
android:layout_width="match_parent"
android:layout_height="match_parent">
    <ListView android:id="@+id/listView"
android:layout_width="match_parent"
android:layout_height="match_parent"/>
</LinearLayout>

```

### 45) Counter increment demo (simple UI)

#### CounterActivity.java

```

package com.example.madpracticals;

import android.os.Bundle;

```

```
import androidx.appcompat.app.AppCompatActivity;
import com.example.madpracticals.databinding.ActivityCounterBinding;
```

```
public class CounterActivity extends AppCompatActivity {
    private ActivityCounterBinding b;
    private int count = 0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        b = ActivityCounterBinding.inflate(getLayoutInflater());
        setContentView(b.getRoot());
        b.btnInc.setOnClickListener(v -> b.tv.setText("Count: " +
++count));
    }
}
```

#### activity\_counter.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:gravity="center"
    android:padding="12dp" android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView android:id="@+id/tv" android:text="Count: 0"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:textSize="20sp"/>
    <Button android:id="@+id/btnInc" android:text="Increment"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" android:layout_marginTop="12dp"/>
</LinearLayout>
```