LAB#06

INTRODUCTION TO XML AND INTERNET RESOURCES

Internet Resources

Android offers several ways to leverage Internet resources. You can use client-side APIs, such as the Google APIs, to interact directly with server processes and Webview to access internet services. You can process remote XML feeds to extract and process data using a Java-based XML parser, such as SAX or the XML Pull Parser.
Before you can access Internet resources, you need to add an INTERNET uses-permission node to your application manifest, as shown in the following XML snippet:
<uses-permission android:name=”android.permission.INTERNET”/>

Let’s Try

```
<?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=".MainActivity">

   <WebView
       android:id="@+id/wbv1"
       android:layout_width="match_parent"
       android:layout_height="match_parent"></WebView>

</android.support.constraint.ConstraintLayout>
```

```
package com.example.androidservices;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebView;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        WebView myWebView = (WebView) findViewById(R.id.wbv1);
        myWebView.loadUrl("http://www.gmail.com");


    }
}
```

Parsing XML data:

There are various XML parses available in the Android SDK in standard. Thus, you can use the following solutions:

- XMLPullParser API
- DOM Parser API
- SAX Parser API

Generate xml file in Asset folder and create employee.xml

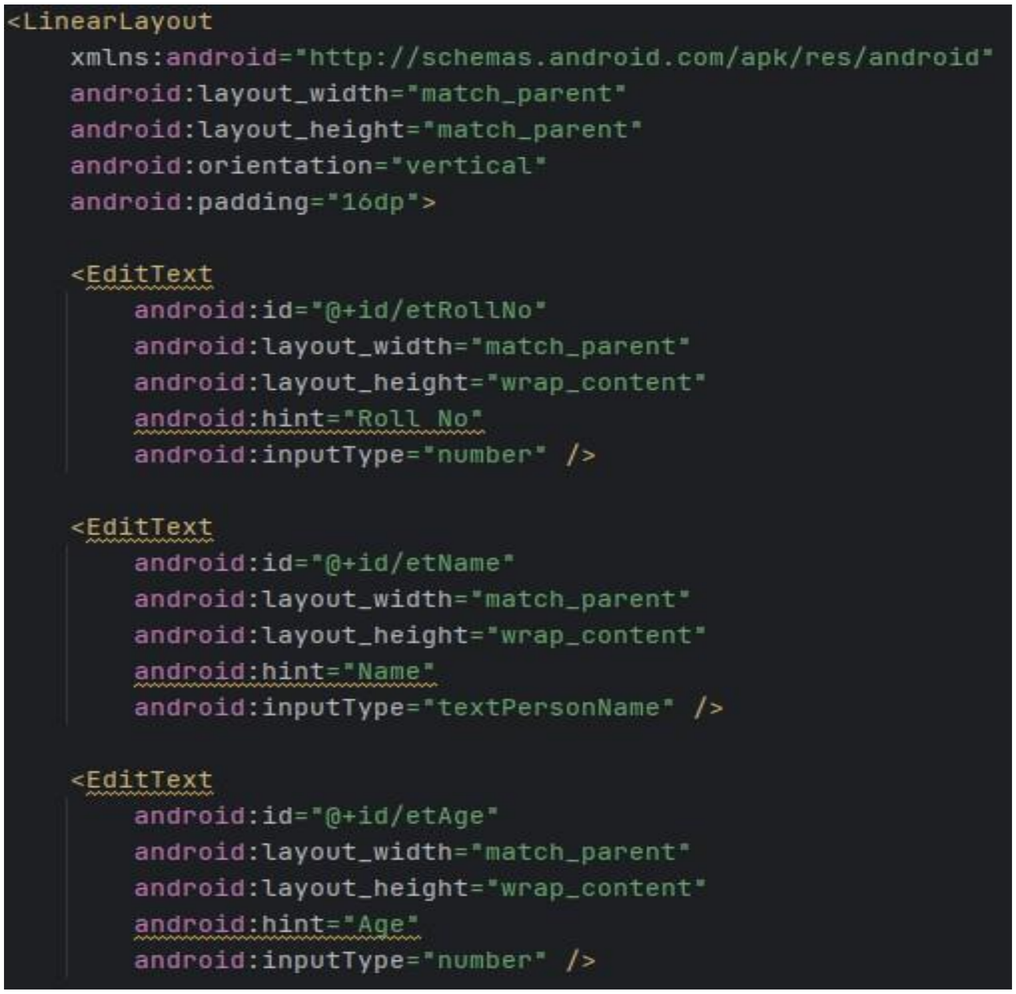
```
<?xml version="1.0"?>
<records>
    <employee>
        <name>Talha</name>
        <surname>Ahmed</surname>
        <salary>50000</salary>
    </employee>

    <employee>
        <name>Riaz </name>
        <surname>Ahmed</surname>
        <salary>60000</salary>
    </employee>

    <employee>
        <name>Babar</name>
        <surname>Khan</surname>
        <salary>70000</salary>
    </employee>

</records>
```

Here we use XMLPullParser API for access XML data in Activity.

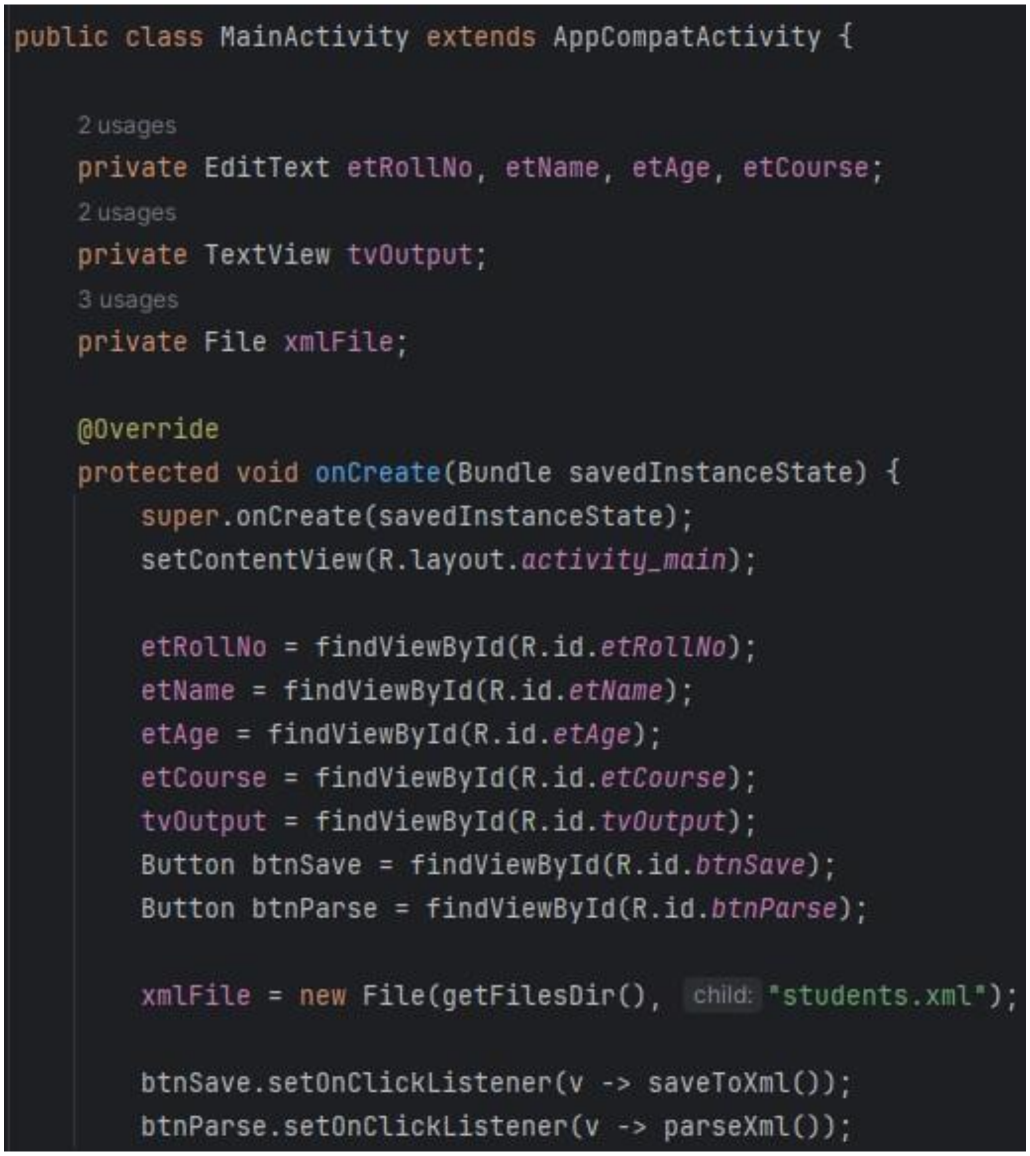
```
XmlPullParserFactory  parserFactory = XmlPullParserFactory.newInstance();
XmlPullParser parser = parserFactory.newPullParser();
InputStream is = getAssets().open("employee.xml");
```

Lab Task:

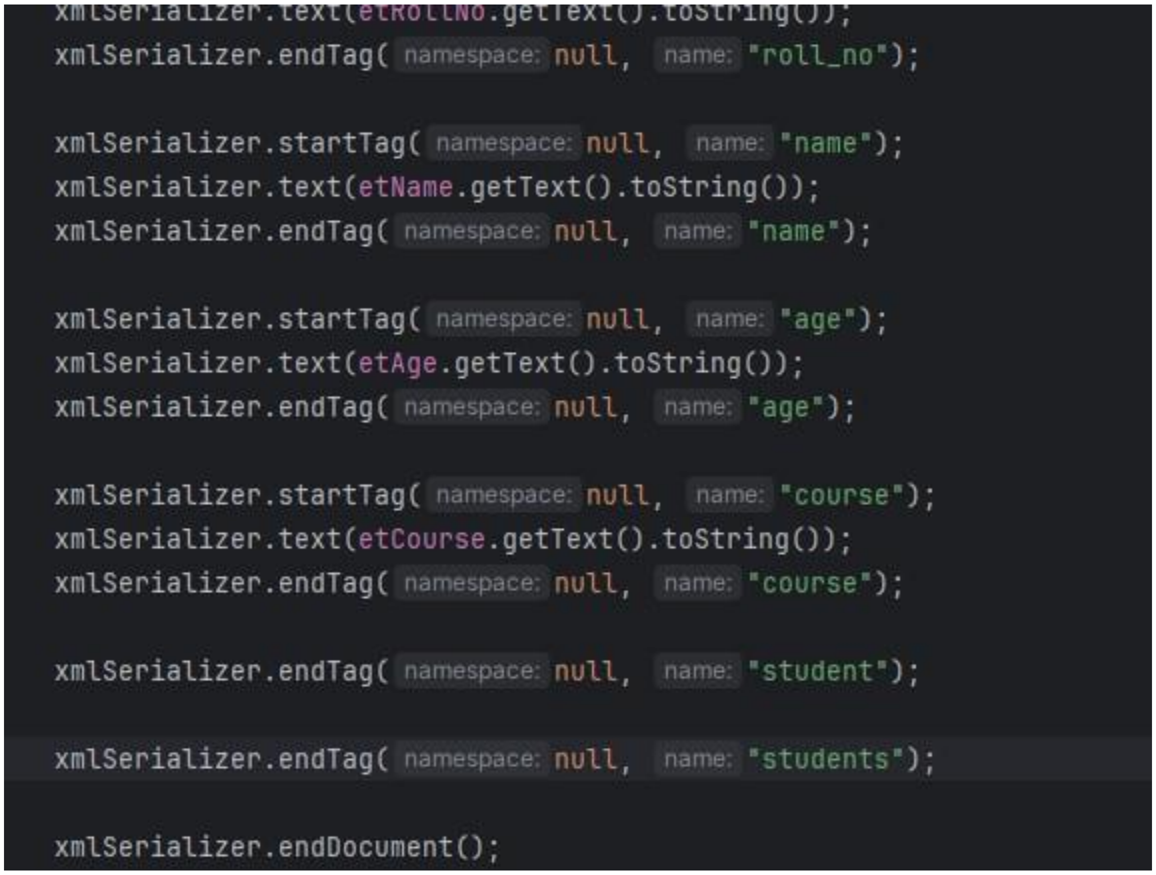
1) Generate Android app which generate students.xml file with roll no, name, age, and course and parse xml data.



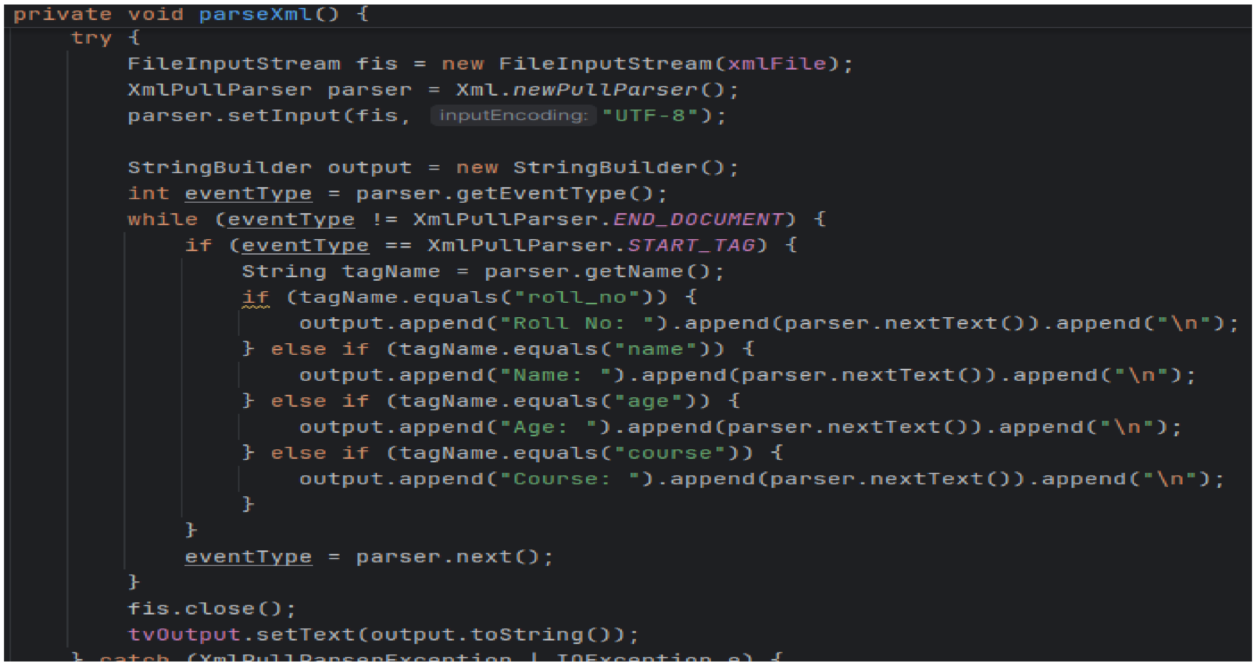
```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etRollNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Roll No"
        android:inputType="number" />

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/etAge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:inputType="number" />
```



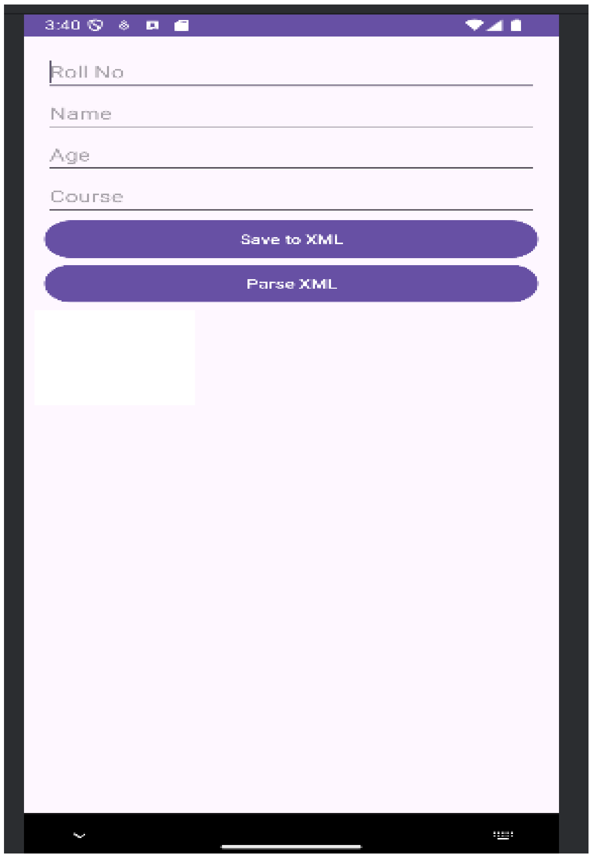
```
public class MainActivity extends AppCompatActivity {

    2 usages
    private EditText etRollNo, etName, etAge, etCourse;
    2 usages
    private TextView tvOutput;
    3 usages
    private File xmlFile;

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

        etRollNo = findViewById(R.id.etRollNo);
        etName = findViewById(R.id.etName);
        etAge = findViewById(R.id.etAge);
        etCourse = findViewById(R.id.etCourse);
        tvOutput = findViewById(R.id.tvOutput);
        Button btnSave = findViewById(R.id.btnSave);
        Button btnParse = findViewById(R.id.btnParse);

        xmlFile = new File(getFilesDir(),  child: "students.xml");

        btnSave.setOnClickListener(v -> saveToXml());
        btnParse.setOnClickListener(v -> parseXml());
```



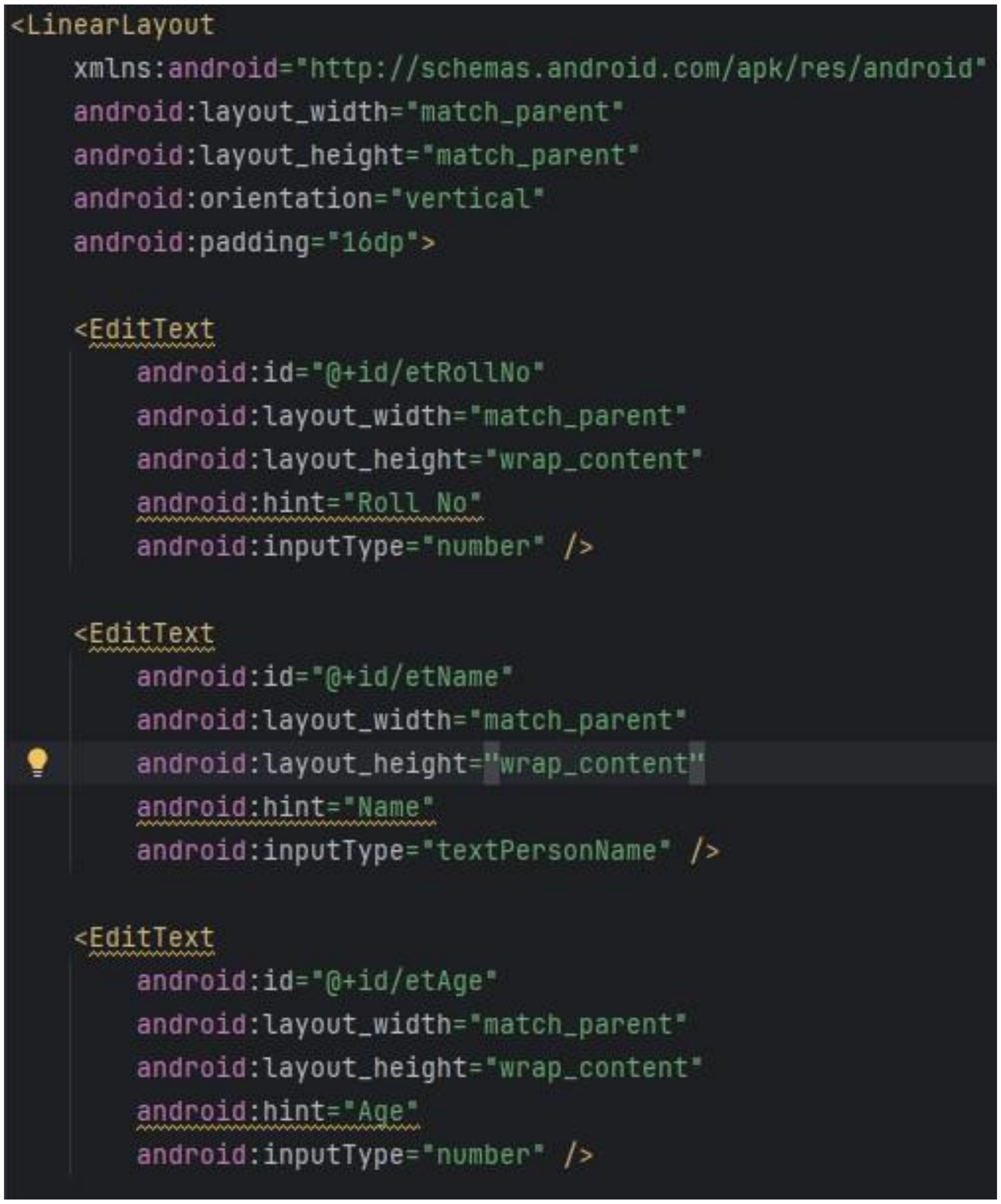
```
xmlSerializer.text(etRollNo.getText().toString());
xmlSerializer.endTag( namespace: null,  name: "roll_no");

xmlSerializer.startTag( namespace: null,  name: "name");
xmlSerializer.text(etName.getText().toString());
xmlSerializer.endTag( namespace: null,  name: "name");

xmlSerializer.startTag( namespace: null,  name: "age");
xmlSerializer.text(etAge.getText().toString());
xmlSerializer.endTag( namespace: null,  name: "age");

xmlSerializer.startTag( namespace: null,  name: "course");
xmlSerializer.text(etCourse.getText().toString());
xmlSerializer.endTag( namespace: null,  name: "course");

xmlSerializer.endTag( namespace: null,  name: "student");

xmlSerializer.endTag( namespace: null,  name: "students");

xmlSerializer.endDocument();
```



```
private void parseXml() {
    try {
        FileInputStream fis = new FileInputStream(xmlFile);
        XmlPullParser parser = Xml.newPullParser();
        parser.setInput(fis,  inputEncoding: "UTF-8");

        StringBuilder output = new StringBuilder();
        int eventType = parser.getEventType();
        while (eventType != XmlPullParser.END_DOCUMENT) {
            if (eventType == XmlPullParser.START_TAG) {
                String tagName = parser.getName();
                if (tagName.equals("roll_no")) {
                    output.append("Roll No: ").append(parser.nextText()).append("\n");
                } else if (tagName.equals("name")) {
                    output.append("Name: ").append(parser.nextText()).append("\n");
                } else if (tagName.equals("age")) {
                    output.append("Age: ").append(parser.nextText()).append("\n");
                } else if (tagName.equals("course")) {
                    output.append("Course: ").append(parser.nextText()).append("\n");
                }
            }
            eventType = parser.next();
        }
        fis.close();
        tvOutput.setText(output.toString());
    } catch (XmlPullParserException | IOException e) {
```

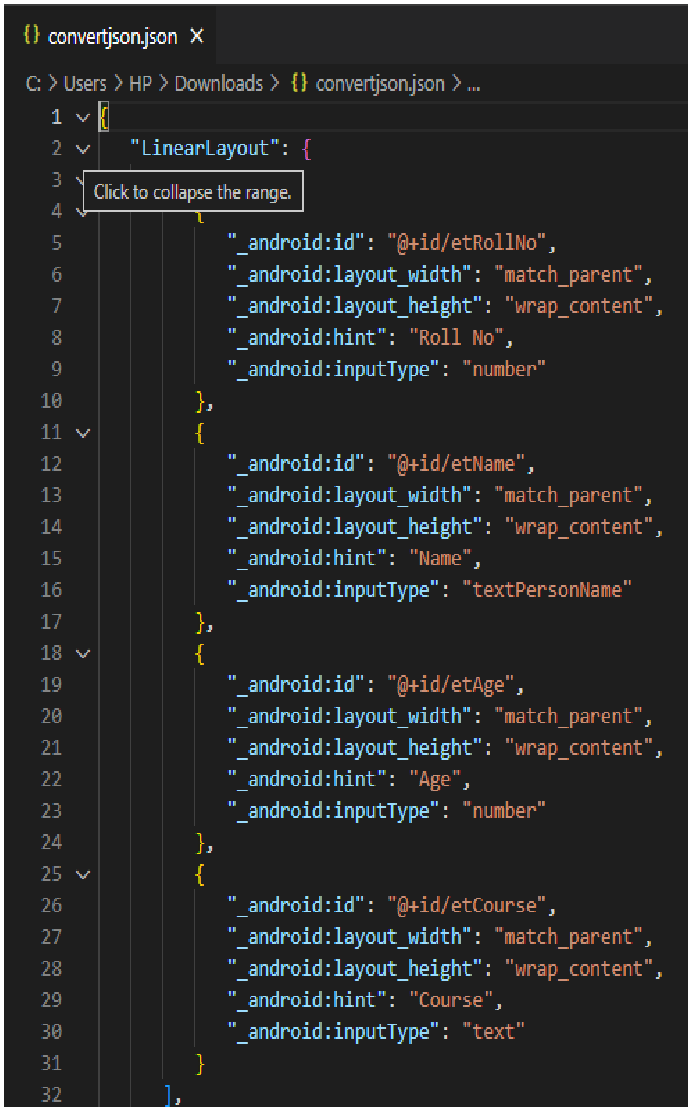

3:40
Roll No
Name
Age
Course
Save to XML
Parse XML

2) Use any web Api to convert Xml data into meaning full form.

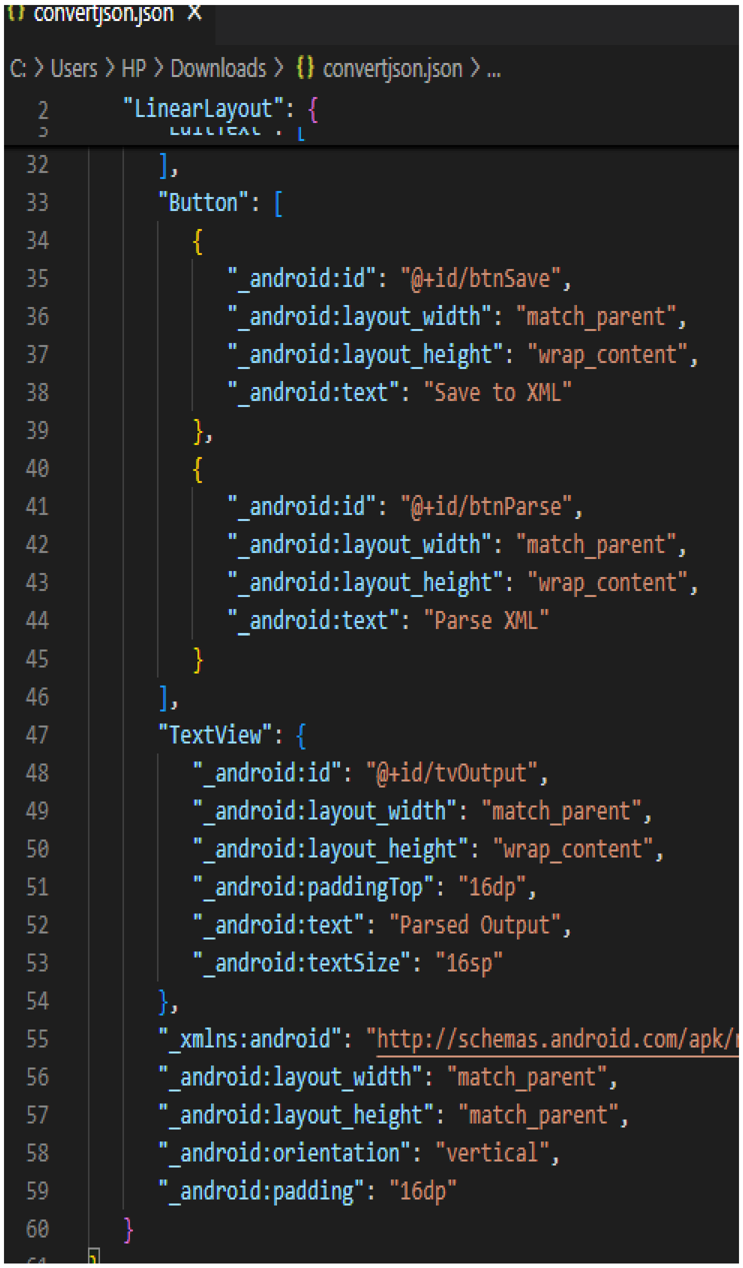


```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <EditText
        android:id="@+id/etRollNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Roll No"
        android:inputType="number" />

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/etAge"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Age"
        android:inputType="number" />
```

Xml data converted to meaningful data:


convertjson.json ×
C: > Users > HP > Downloads > convertjson.json > ...
Click to collapse the range.

```
{
    "LinearLayout": {
            {
                "_android:id": "@+id/etRollNo",
                "_android:layout_width": "match_parent",
                "_android:layout_height": "wrap_content",
                "_android:hint": "Roll No",
                "_android:inputType": "number"
            },
            {
                "_android:id": "@+id/etName",
                "_android:layout_width": "match_parent",
                "_android:layout_height": "wrap_content",
                "_android:hint": "Name",
                "_android:inputType": "textPersonName"
            },
            {
                "_android:id": "@+id/etAge",
                "_android:layout_width": "match_parent",
                "_android:layout_height": "wrap_content",
                "_android:hint": "Age",
                "_android:inputType": "number"
            },
            {
                "_android:id": "@+id/etCourse",
                "_android:layout_width": "match_parent",
                "_android:layout_height": "wrap_content",
                "_android:hint": "Course",
                "_android:inputType": "text"
            }
        ],
```


{} convertjson.json X

C: > Users > HP > Downloads > {} convertjson.json > ...

```
    "LinearLayout": {
      ],
      "Button": [
        {
          "_android:id": "@+id/btnSave",
          "_android:layout_width": "match_parent",
          "_android:layout_height": "wrap_content",
          "_android:text": "Save to XML"
        },
        {
          "_android:id": "@+id/btnParse",
          "_android:layout_width": "match_parent",
          "_android:layout_height": "wrap_content",
          "_android:text": "Parse XML"
        }
      ],
      "TextView": {
        "_android:id": "@+id/tvOutput",
        "_android:layout_width": "match_parent",
        "_android:layout_height": "wrap_content",
        "_android:paddingTop": "16dp",
        "_android:text": "Parsed Output",
        "_android:textSize": "16sp"
      },
      "_xmlns:android": "http://schemas.android.com/apk/
      "_android:layout_width": "match_parent",
      "_android:layout_height": "match_parent",
      "_android:orientation": "vertical",
      "_android:padding": "16dp"
    }
```