Ex. No: 8

APPLICATION THAT WRITES DATA TO SD CARD

AIM

To develop an Android Application that writes data to SD card.

PROCEDURE

1. Creating a New project:

Open Android Studio and then click on File -> New -> New project. type the Application name and click Next.
Select the Empty Activity and click Next.

2. Designing layout for the Android Application:

Click on app -> res -> layout -> activity_main.xml

3. Adding permissions in Manifest for the Android Application

Click on app -> manifests -> AndroidManifest.xml
Include the WRITE_EXTERNAL_STORAGE permissions in the AndroidManifest.xml file

4. Java Coding for the Android Application

Click on app -> java -> MainActivity.
Create action listeners for test,and three buttons for read, write and clear.
Create new file myfile .txt using file input stream
Toast is initiated to create and show messages.
5. Update strings.xml in res folder.

SOURCE CODE:

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_marginTop="46dp"
    android:gravity="center"
    android:text="@string/add_text"
    android:textSize="24sp"
    android:textStyle="bold" />
<Button
    android:id="@+id/button4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignStart="@+id/button"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="245dp"
    android:layout_marginBottom="318dp"
    android:onClick="next"
    android:text="@string/click_to_view" />
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/button4"
    android:layout_alignBottom="@+id/button4"
    android:layout_alignParentStart="true"
    android:layout_marginStart="24dp"
    android:onClick="save"
```

```
        android:text="@string/write_data" />
    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/textView"
        android:layout_marginTop="16dp"
        android:ems="10"
        android:gravity="center_vertical|center"
        android:inputType="textMultiLine" />
</RelativeLayout>
```

MainActivity.java

```
package com.aravind.sd_card;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;


public class MainActivity extends AppCompatActivity {
    EditText editText;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = findViewById(R.id.editText2);
    }


    public void next(View view) {
        Intent intent = new Intent(MainActivity.this,
MainActivity2.class);
        startActivity(intent);
    }
    public void save(View view) {
        String info = editText.getText().toString();
        if(!(info.isEmpty())) {
            File folder = getExternalFilesDir("Android");
            File myFile = new File(folder, "sdcard.txt");
            writeData(myFile, info);
            editText.setText("");
        }
        else{
            Toast.makeText(getApplicationContext(), "Enter data",
Toast.LENGTH_SHORT).show();
        }
    }


    private void writeData(File myFile, String data) {
        FileOutputStream fileOutputStream = null;
        try {
            fileOutputStream = new FileOutputStream(myFile);
```

```
            fileOutputStream.write(data.getBytes());
            Toast.makeText(getApplicationContext(), "DATA WRITTEN IN
SDCARD" + myFile.getAbsolutePath(),
Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            if (fileOutputStream != null) {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

activity_main2.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:id="@+id/activity_main2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity2">

    <TextView
```

```
    android:id="@+id/getText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_marginTop="48dp"
    android:gravity="center"
    android:text=""
    android:textSize="28sp"
    android:textStyle="bold"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/button5"
    android:onClick="back"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="320dp"
    android:text="@string/back"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/button2"
    android:onClick="show"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="308dp"
```

```
        android:layout_marginBottom="324dp"
        android:text="@string/show_data"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity2.java

```
package com.aravind.sd_card;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;

public class MainActivity2 extends AppCompatActivity {
    TextView showText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        showText = findViewById(R.id.getText);
    }
    public void back(View view) {
        Intent intent = new Intent(MainActivity2.this,
MainActivity.class);
        startActivity(intent);
```

```
}

public void show(View view) {
    File folder = getExternalFilesDir("Android");
    File myFile = new File(folder, "sdcard.txt");
    String text = getdata(myFile);
    if (text != null) {
        showText.setText(text);
    } else {
        showText.setText("No Data");
    }
}

private String getdata(File myfile) {
    FileInputStream fileInputStream = null;
    try {
        fileInputStream = new FileInputStream(myfile);
        int i = -1;
        StringBuffer buffer = new StringBuffer();
        while ((i = fileInputStream.read()) != -1) {
            buffer.append((char) i);
        }
        return buffer.toString();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (fileInputStream != null) {
            try {
                fileInputStream.close();
```

```
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return null;
}}
```

AndroidManifest.xml

Add in 3rd line: **xmlns:tools=http://schemas.android.com/tools**

Add above <application>, the following:

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

android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        tools:ignore="ScopedStorage">
</uses-permission>
```

Add after first activity:

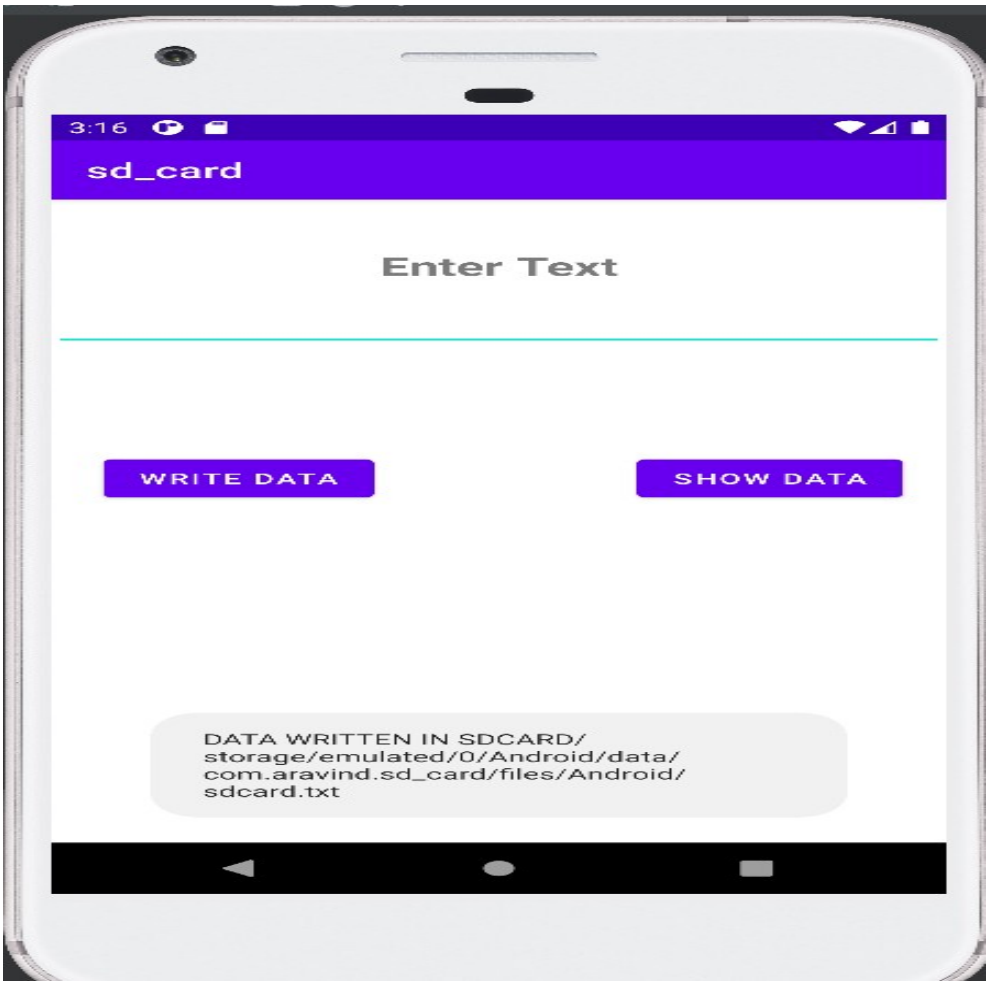
```
<activity
    android:name=".MainActivity2"
    android:exported="false">
</activity>
```

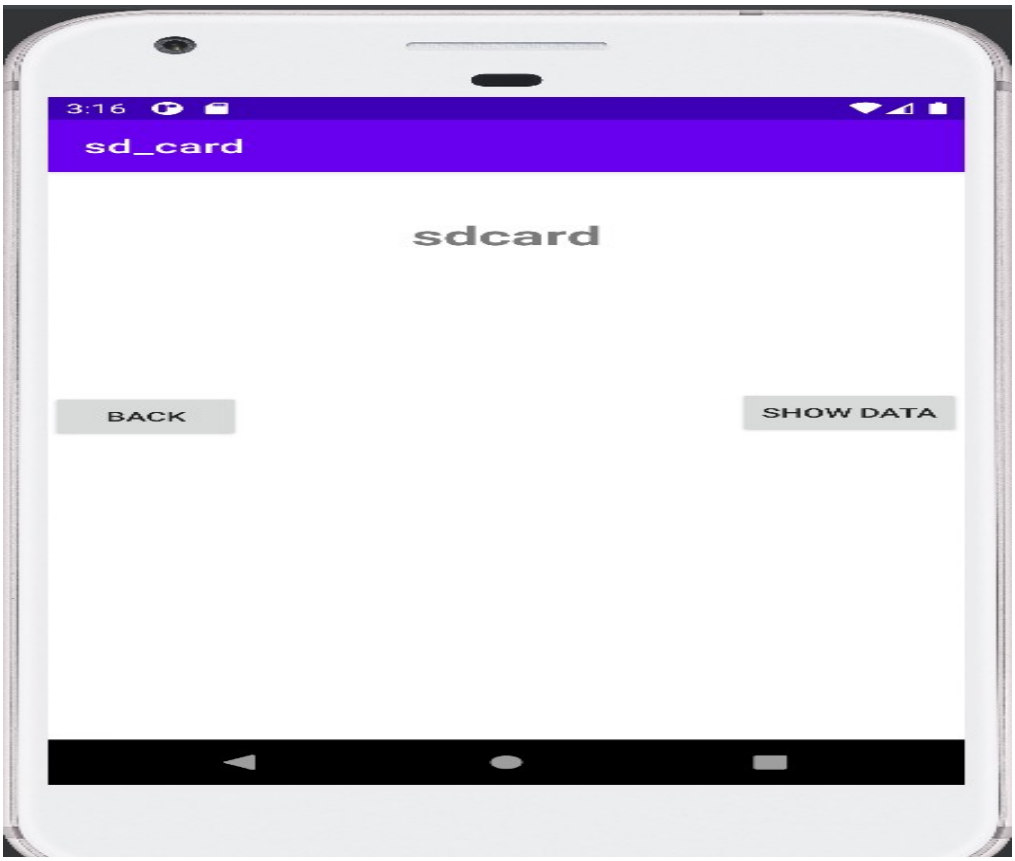
IN RES FOLDER, UPDATE IN strings.xml

```
<resources>
    <string name="app_name">sd_card</string>
    <string name="add_text">Enter Text</string>
    <string name="click_to_view">Show Data</string>
    <string name="write_data">Write Data</string>
```

```
    <string name="show_data">Show Data</string>
    <string name="back">back</string>
</resources>
```

OUTPUT





RESULT

Thus an android application that writes data to SD card has been implemented successfully.

Ex. No: 9

RSS FEED

AIM

To develop an Android Application that uses RSS Feed.

PROCEDURE

1. Creating a New project:

   Open Android Studio and then click on File -> New -> New project.
   Then type the Application name and click Next.
   Then select the Minimum SDK as shown below and click Next.
   Then select the Empty Activity and click Next.
   Finally click Finish.

2. Designing layout for the Android Application:

   Click on app -> res -> layout -> activity_main.xml

Create linear layout with ListView.

3. Adding permissions in Manifest for the Android Application:

Click on app -> manifests -> AndroidManifest.xml.
Now include the INTERNET permissions in the AndroidManifest.xml file

4. Java Coding for the Android Application

Click on app -> java -> MainActivity.
Create URL and get the XML from an input stream
Returns the type of current event: START_TAG, END_TAG
Extract the link and the URL
Define the array adapters and onclick listeners.

SOURCE CODE

activity_main.xml

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

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.exno6" >
<uses-permission android:name="android.permission.INTERNET"/>
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
```

```
android:label="@string/app_name"
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>
</manifest>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;

import java.io.IOException;
```

```
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {

    ListView lvRss;

    ArrayList<String> titles;
    ArrayList<String> links;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvRss =  findViewById(R.id.listView);
        titles = new ArrayList<>();
        links = new ArrayList<>();

        lvRss.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> adapterView, View view, int
i, long l) {
                String url = links.get(i).substring(1);
                Uri uri = Uri.parse(url);
                Intent intent = new Intent(Intent.ACTION_VIEW, uri);
                startActivity(intent);
            }
        });
```

```
        new ProcessInBackground().execute();
    }
    public InputStream getInputStream(URL url)
    {
        try
        {
            //openConnection() returns instance that represents a connection to the remote object referred      to by the URL
            //getInputStream() returns a stream that reads from the open connection
            return url.openConnection().getInputStream();
        }
        catch (IOException e)
        {
            return null;
        }
    }

    public class ProcessInBackground extends AsyncTask<Integer, Void, Exception>
    {
        ProgressDialog progressDialog = new ProgressDialog(MainActivity.this);
        Exception exception = null;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();

            progressDialog.setMessage("Busy loading rss feed...please wait...");
            progressDialog.show();
```

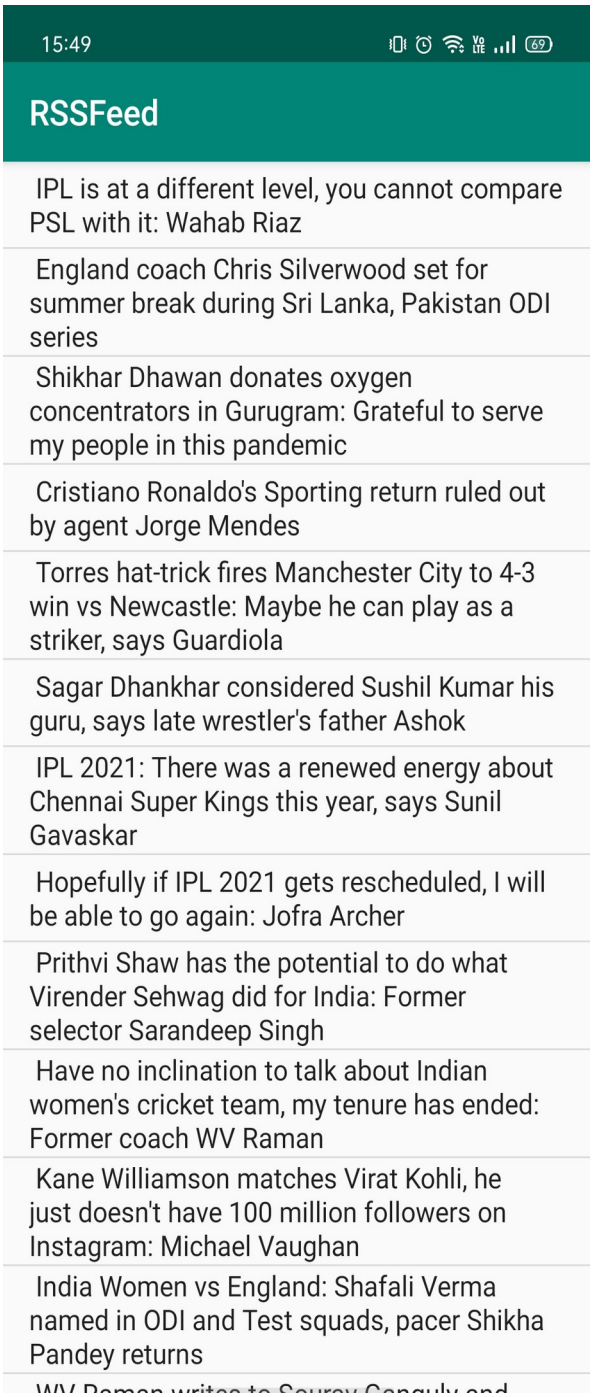
```
    }
    @Override
    protected Exception doInBackground(Integer... params) {
       try
       {
          URL url = new URL("https://www.indiatoday.in/rss/1206550");
          //creates new instance of PullParserFactory that can be used to
create XML pull parsers
          XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
          //Specifies whether the parser produced by this factory will provide
support
          //for XML namespaces
          factory.setNamespaceAware(false);
          //creates a new instance of a XML pull parser using the currently
configured
          //factory features
          XmlPullParser xpp = factory.newPullParser();
          // We will get the XML from an input stream
          xpp.setInput(getInputStream(url), "UTF_8");
          /* We will parse the XML content looking for the "<title>" tag
which appears inside the "<item>" tag.
           * We should take into consideration that the rss feed name is also
enclosed in a "<title>" tag.
           * Every feed begins with these lines:
"<channel><title>Feed_Name</title> etc."
           * We should skip the "<title>" tag which is a child of "<channel>"
tag,
           * and take into consideration only the "<title>" tag which is a
child of the "<item>" tag
           *
           * In order to achieve this, we will make use of a boolean variable
called "insideItem".
```

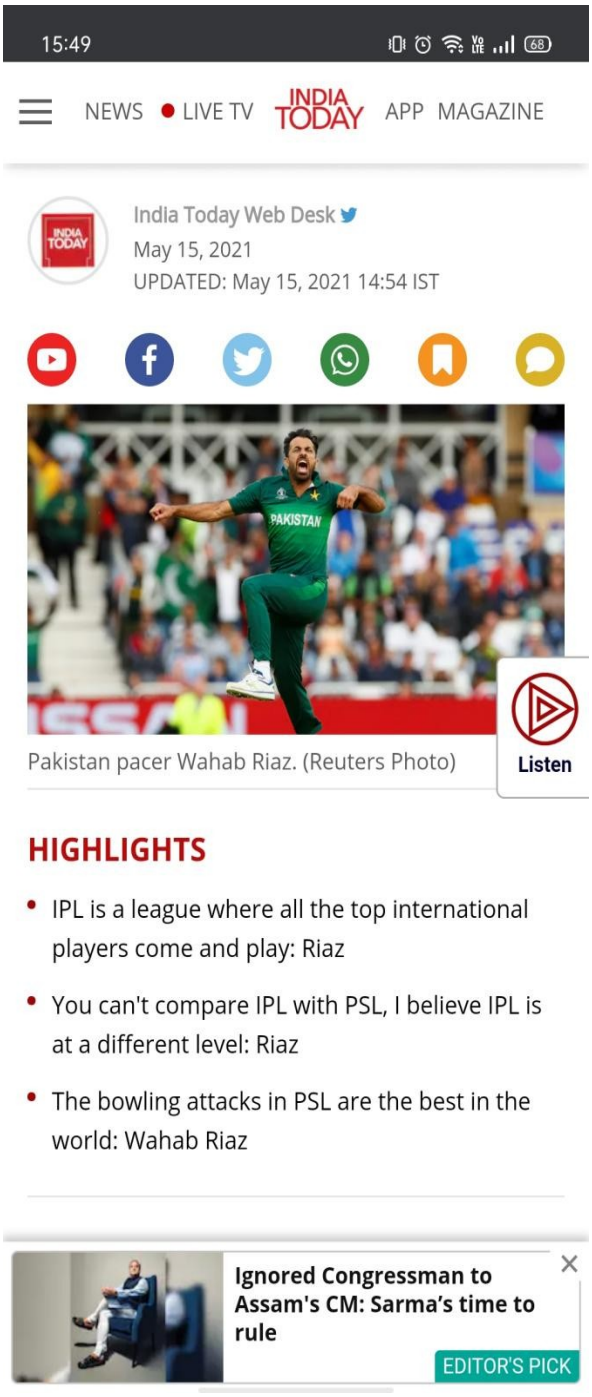
```
         */
        boolean insideItem = false;


        // Returns the type of current event: START_TAG, END_TAG,
START_DOCUMENT,              END_DOCUMENT etc..
        int eventType = xpp.getEventType(); //loop control variable
        while (eventType != XmlPullParser.END_DOCUMENT)
        {
           //if we are at a START_TAG (opening tag)
           if (eventType == XmlPullParser.START_TAG)
           {
              //if the tag is called "item"
              if (xpp.getName().equalsIgnoreCase("item"))
              {
                 insideItem = true;
              }
              //if the tag is called "title"
              else if (xpp.getName().equalsIgnoreCase("title"))
              {
                 if (insideItem)
                 {
                    // extract the text between <title> and </title>
                    titles.add(xpp.nextText());
                 }
              }
              //if the tag is called "link"
              else if (xpp.getName().equalsIgnoreCase("link"))
              {
                 if (insideItem)
```

```
                {
                    // extract the text between <link> and </link>
                    links.add(xpp.nextText());
                }
            }
        }
        //if we are at an END_TAG and the END_TAG is called "item"
        else if (eventType == XmlPullParser.END_TAG &&
            xpp.getName().equalsIgnoreCase("item"))
                {
            insideItem = false;
                }
        eventType = xpp.next(); //move to next element
    }
  }
  catch (MalformedURLException e)
  {
    exception = e;
  }
  catch (XmlPullParserException e)
  {
    exception = e;
  }
  catch (IOException e)
  {
    exception = e;
  }
  return exception;
}
```

```
        @Override
        protected void onPostExecute(Exception s) {
            super.onPostExecute(s);


            ArrayAdapter<String> adapter = new
ArrayAdapter<>(MainActivity.this,
                    android.R.layout.simple_list_item_1, titles);
            lvRss.setAdapter(adapter);
            progressDialog.dismiss();
        }
    }
}
```

OUTPUT





RESULT

Thus an Android Application that uses RSS feed was implemented successfully.