Ex.No: 1 GUI COMPONENTS, FONTS AND COLORS Date:

AIM:
To develop an application that uses GUI components, Font and Colours.

ALGORITHM:
1. Open ANDROID STUDIO.
2. Click File->new->New Project.
   Give the File name->Choose the Empty Activity and Package name and then click Finish button.
3. Go to res folder and select layout. Double click the activity_main.xml file.
4. Now you can see the Graphical layout window
5. Drag and drop the following components:
   - One TextView with text Hello world
   - Three Buttons with labeled as Change Text Size, Change Text Color and Change Text Style
6. Go to java folder. Double click the MainActivity.java file.
7. In java file write the activities done by the application such as, actions of buttons.
8. Finally run the android application.

PROGRAM:

activity_main.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="@color/purple_200"
   tools:context=".MainActivity">


   <TextView
      android:id="@+id/textView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="76dp"
      android:text="Hello World!"
      android:textSize="15sp"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent" />

   <Button
      android:id="@+id/button1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginBottom="80dp"
      android:text="Change Text Size"
```

```
        app:layout_constraintBottom_toTopOf="@+id/button2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />


<Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="80dp"
        android:text="Change Text Style"
        app:layout_constraintBottom_toTopOf="@+id/button3"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent" />

    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="160dp"
        android:text="Change Text Color"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent" />


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

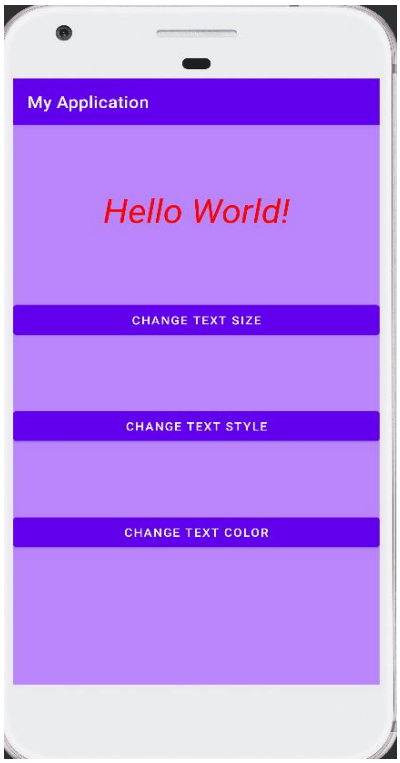
MainActivity.java:

```
package com.cse.myapplication;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    int font=20;
    TextView v;
    Button b,b1,b2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

```
v=findViewById(R.id.textView);
b=findViewById(R.id.button1);
b1=findViewById(R.id.button2);
b2=findViewById(R.id.button3);
b.setOnClickListener(new View.OnClickListener() {
   @Override
   public void onClick(View view) {
      font = font + 5;
      v.setTextSize(font);
      if(font == 40){
         font = 20;
      }

   }
});
b1.setOnClickListener(new View.OnClickListener() {
   int count=1;
   @Override
   public void onClick(View view)
    {
        switch(count){
         case 1:
            v.setTypeface(Typeface.MONOSPACE,Typeface.BOLD);
            break;
         case 2:
            v.setTypeface(Typeface.DEFAULT,Typeface.BOLD);
            break;
         case 3:
            v.setTypeface(Typeface.SANS_SERIF,Typeface.ITALIC);
            break;
         case 4:
            v.setTypeface(Typeface.MONOSPACE,Typeface.ITALIC);
            break;
         case 5:
            v.setTypeface(Typeface.DEFAULT,Typeface.BOLD_ITALIC);
            break;
      }
      count++;
      if(count==6){
         count=1;
      }
   }
});
b2.setOnClickListener(new View.OnClickListener() {
   int count = 1;
   @Override
   public void onClick(View view) {
      switch(count){
         case 1:
            v.setTextColor(Color.RED);
            break;
```

```
                case 2:
                    v.setTextColor(Color.GREEN);
                    break;
                case 3:
                    v.setTextColor(Color.YELLOW);
                    break;
                case 4:
                    v.setTextColor(Color.MAGENTA);
                    break;
                case 5:
                    v.setTextColor(Color.BLACK);
                    break;
                case 6:
                    v.setTextColor(Color.BLUE);
                    break;
                case 7:
                    v.setTextColor(Color.GRAY);
                    break;
            }
            count++;
            if(count == 8){
                count = 1;
            }

        }
    });
  }
}
```

OUTPUT:



RESULT: Thus an application that uses GUI components, Font and Colours is developed.

Ex.No: 2 LAYOUT MANAGERS AND EVENT LISTENERS Date:

AIM

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

PROCEDURE

1. Creating a **New project:**

- √ Open Android Stdio and then click on **File -> New -> New project.**
- √ Select the **Empty Activity** and type the name of the activity as ex2 **and Finish.**
- √ In MainActivity.java create views,refer views.
- √ Create Event listeners and intent to navigate to second activity and pass the values to the second activity.

2. Right click app→Java→com.example.ex2 and select new →java and type **SecondActivity** and click enter.

- ✓ Get the Intent from MainActivity to secondactivity.
- ✓ Getting the Values from First Activity using the Intent received
- ✓ Setting the Values to Intent.

3. Designing Layout for Main Activity:

- Click on **app -> res -> layout -> activity_main.xml.**
- √ Create Linear layout,Grid Layout,Relative layout and Text Views.
- √ Create second activity.xml to display the content created in activity_main.xml.
- √ Right click layout in res folder and select new →xml→Layoutxml file→type **second** and click enter.

4. Include the second activity in **androidManifest.xml.**

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:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="100dp">
      <TextView
         android:id="@+id/textView"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:layout_margin="30dp"
         android:text="Details Form"
         android:textSize="25sp"
         android:gravity="center"/>
```

```
</LinearLayout>
<GridLayout
   android:id="@+id/gridLayout"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:layout_marginTop="100dp"
   android:layout_marginBottom="200dp"
   android:columnCount="2"
   android:rowCount="3">
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="0"
      android:layout_column="0"
      android:text="Name"
      android:textSize="20sp"
      android:gravity="center"/>
   <EditText
      android:id="@+id/editText"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="0"
      android:layout_column="1"
      android:ems="10"/>
   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="1"
      android:layout_column="0"
      android:text="Reg.No"
      android:textSize="20sp"
      android:gravity="center"/>
   <EditText
      android:id="@+id/editText2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_row="1"
      android:layout_column="1"
      android:inputType="number"
```

```
        android:ems="10"/>
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="0"
        android:text="Dept"
        android:textSize="20sp"
        android:gravity="center"/>
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_row="2"
        android:layout_column="1"
        android:spinnerMode="dropdown"/>
  </GridLayout>
  <Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerInParent="true"
    android:layout_marginBottom="150dp"
    android:text="Submit"/>
</RelativeLayout>
```

MainActivity.java

```
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
public class MainActivity extends AppCompatActivity {
  EditText e1,e2;
  Button bt;
  Spinner s;
  String [] dept_array={"CSE","ECE","IT","Mech","Civil"};
  String name,reg,dept;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1= (EditText) findViewById(R.id.editText);
        e2= (EditText) findViewById(R.id.editText2);
        bt= (Button) findViewById(R.id.button);
        s= (Spinner) findViewById(R.id.spinner);
         ArrayAdapter adapter= new
ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,dept_array);
        s.setAdapter(adapter);
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                name=e1.getText().toString();
                reg=e2.getText().toString();
                dept=s.getSelectedItem().toString();
               Intent i = new Intent(MainActivity.this,SecondActivity.class);
               i.putExtra("name_key", name);
i.putExtra("reg_key",reg);
i.putExtra("dept_key", dept);
                startActivity(i);
            }
        });
    }
}
```

SecondActivity.java

```
import android.content.Intent;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class SecondActivity extends AppCompatActivity {
    TextView t1,t2,t3;
    String name,reg,dept;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second);
        t1= (TextView) findViewById(R.id.textView1);
        t2= (TextView) findViewById(R.id.textView2);
        t3= (TextView) findViewById(R.id.textView3);
        Intent i = getIntent();
        name=i.getStringExtra("name_key");
        reg=i.getStringExtra("reg_key");
```

```
        dept=i.getStringExtra("dept_key");
        t1.setText(name);
        t2.setText(reg);
        t3.setText(dept);
}
}
```

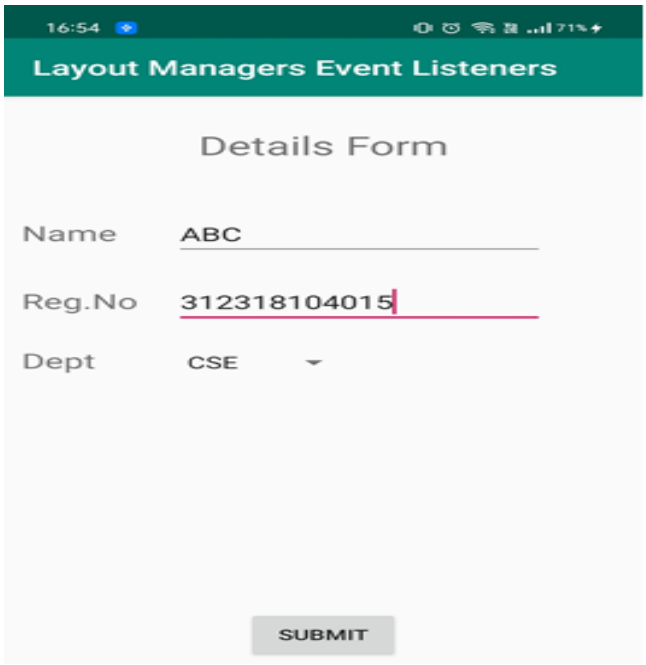
second.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context="com.example.exno2.SecondActivity"
   android:orientation="vertical"
   android:gravity="center">

<TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:text="New Text"
      android:textSize="30sp"/>

<TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:text="New Text"
      android:textSize="30sp"/>

<TextView
      android:id="@+id/textView3"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="20dp"
      android:text="New Text"
      android:textSize="30sp"/>
</LinearLayout>
```

In **androidManifest.xml**, add the second activity:

```
<activity android:label="@string/app_name" android:name="SecondActivity"/>
```

OUTPUT



RESULT

Thus a simple android application that uses Layout Managers and Event Listeners has been implemented successfully.

Ex.No: 3 **DRAWING GRAPHICAL PRIMITIVES** **Date:**

AIM

To develop a Simple Android Application that draws basic Graphical Primitives on the screen.

PROCEDURE

1. Creating a New project:

- Open Android Studio and then click on **File -> New -> New project.**
- Select the **Empty Activity** and click **Next and Finish**
- **In MainActivity.java Import graphics package.**
- Create a Bitmap, Setting the Bitmap as background for the ImageView, Creating the Canvas Object, Creating the Paint Object and set its color & TextSize
- Draw Square,Rectangle,Circle and line

2. Designing Layout for Main Activity:

- Click on **app -> res -> layout -> activity_main.xml.**
- **Create relative layout and Image view.**

SOURCE CODE

activity_main.xml

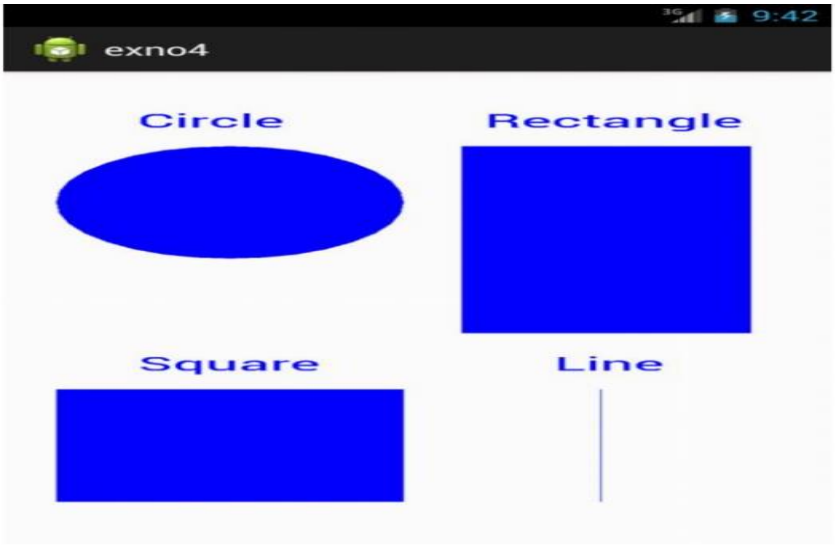
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
    <ImageView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
  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

MainActivity.java

```
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.drawable.BitmapDrawable;
import android.os.Bundle;
import android.widget.ImageView;

public class MainActivity extends Activity
{
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
         //Creating a Bitmap
        Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB_8888);
         //Setting the Bitmap as background for the ImageView
        ImageView i = (ImageView) findViewById(R.id.imageView);
        i.setBackgroundDrawable(new BitmapDrawable(bg));
         //Creating the Canvas Object
        Canvas canvas = new Canvas(bg);
         //Creating the Paint Object and set its color & TextSize
        Paint paint = new Paint();
        paint.setColor(Color.BLUE);
        paint.setTextSize(50);
         //To draw a Rectangle
        canvas.drawText("Rectangle", 420, 150, paint);
        canvas.drawRect(400, 200, 650, 700, paint);
         //To draw a Circle
        canvas.drawText("Circle", 120, 150, paint);
        canvas.drawCircle(200, 350, 150, paint);
         //To draw a Square
        canvas.drawText("Square", 120, 800, paint);
        canvas.drawRect(50, 850, 350, 1150, paint);
         //To draw a Line
        canvas.drawText("Line", 480, 800, paint);
        canvas.drawLine(520, 850, 520, 1150, paint);
    }
}
```

OUTPUT



RESULT

Thus a simple android application to draw basic graphical primitives on the screen is implemented successfully.

Ex.No: 4 **SIMPLE ANDROID APPLICATION THAT MAKES USE OF DATABASE** **Date:**

AIM

To develop a Simple Android Application that makes use of Database

PROCEDURE

1**. Creating a New project:**

- Open Android Studio and then click on **File -> New -> New project->Next->type application name->** select the **Empty Activity** and click **Next->Finish.**

2. **Designing layout for the Android Application:**

- Click on app -> res -> layout -> activity_main.xml.
- Create absolute layout with text views and buttons for retrieving the data from the database.
- Click on main activity and import android.database.sqlite.SQLiteDatabase.
- Create a method on create and set the content view.
- Create action listeners for buttons insert delete, update and view.
- Create database student and insert the values.
- Use cursors to retrieve rows from the database and display it on the application.
- Use if clause for the usage of buttons on click insert,delete,update and view.

SOURCE CODE

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="50dp"
      android:layout_y="20dp"
      android:text="Student Details"
      android:textSize="30sp" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="20dp"
      android:layout_y="110dp"
      android:text="Enter Rollno:"
      android:textSize="20sp" />

   <EditText
      android:id="@+id/Rollno"
      android:layout_width="150dp"
```

```
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="100dp"
    android:inputType="number"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
```

```
        android:layout_y="300dp"
        android:text="Insert"
        android:textSize="30dp" />

    <Button
        android:id="@+id/Delete"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="300dp"
        android:text="Delete"
        android:textSize="30dp" />


    <Button
        android:id="@+id/Update"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="25dp"
        android:layout_y="400dp"
        android:text="Update"
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />
</AbsoluteLayout>
```

MainActivity.java

```
import android.app.Activity;
import android.app.AlertDialog.Builder;
```

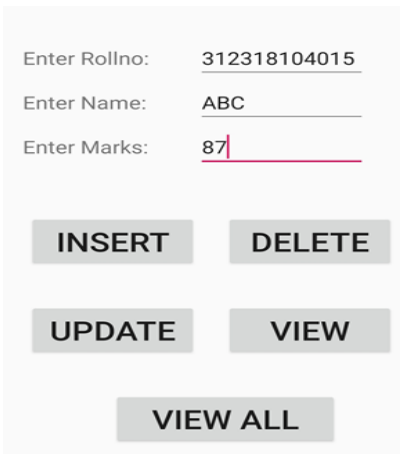
```
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener
{
   EditText Rollno,Name,Marks;
   Button Insert,Delete,Update,View,ViewAll;
   SQLiteDatabase db;
   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
       Rollno=(EditText)findViewById(R.id.Rollno);
      Name=(EditText)findViewById(R.id.Name);
      Marks=(EditText)findViewById(R.id.Marks);
      Insert=(Button)findViewById(R.id.Insert);
      Delete=(Button)findViewById(R.id.Delete);
      Update=(Button)findViewById(R.id.Update);
      View=(Button)findViewById(R.id.View);
      ViewAll=(Button)findViewById(R.id.ViewAll);
       Insert.setOnClickListener(this);
      Delete.setOnClickListener(this);
      Update.setOnClickListener(this);
      View.setOnClickListener(this);
      ViewAll.setOnClickListener(this);
       // Creating database and table
      db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
      db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR, marks VARCHAR);");
   }
   public void onClick(View view)
   {      // Inserting a record to the Student table
      if(view==Insert)
      {          // Checking for empty fields
         if(Rollno.getText().toString().trim().length()==0||
               Name.getText().toString().trim().length()==0||
               Marks.getText().toString().trim().length()==0)
```

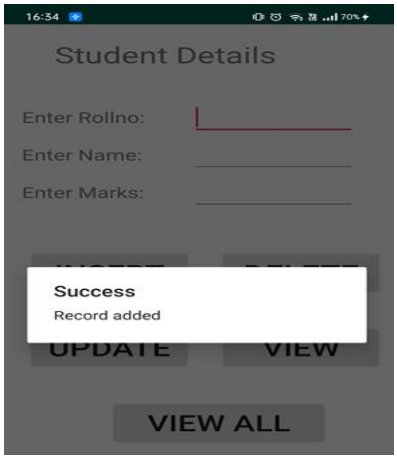
```
        {
           showMessage("Error", "Please enter all values");
           return;
        }
        db.execSQL("INSERT INTO student
VALUES('"+Rollno.getText()+"','"+Name.getText()+
              "','"+Marks.getText()+"');");
        showMessage("Success", "Record added");
        clearText();
     }      // Deleting a record from the Student table
     if(view==Delete)
     {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
           showMessage("Error", "Please enter Rollno");
           return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
        if(c.moveToFirst())
        {
           db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
           showMessage("Success", "Record Deleted");
        }
        else
        {
           showMessage("Error", "Invalid Rollno");
        }
        clearText();
     }      // Updating a record in the Student table
     if(view==Update)
     {
        // Checking for empty roll number
        if(Rollno.getText().toString().trim().length()==0)
        {
           showMessage("Error", "Please enter Rollno");
           return;
        }
        Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
        if(c.moveToFirst()) {
           db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
                 "' WHERE rollno='"+Rollno.getText()+"'");
```

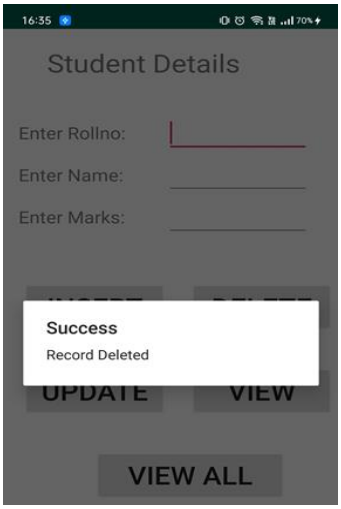
```
            showMessage("Success", "Record Modified");
          }
          else {
            showMessage("Error", "Invalid Rollno");
          }
          clearText();
        }      // Display a record from the Student table
        if(view==View)
        {
          // Checking for empty roll number
          if(Rollno.getText().toString().trim().length()==0)
          {
            showMessage("Error", "Please enter Rollno");
            return;
          }
          Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
          if(c.moveToFirst())
          {
            Name.setText(c.getString(1));
            Marks.setText(c.getString(2));
          }
          else
          {
            showMessage("Error", "Invalid Rollno");
            clearText();
          }
        }      // Displaying all the records
        if(view==ViewAll)
        {
          Cursor c=db.rawQuery("SELECT * FROM student", null);
          if(c.getCount()==0)
          {
            showMessage("Error", "No records found");
            return;
          }
          StringBuffer buffer=new StringBuffer();
          while(c.moveToNext())
          {
            buffer.append("Rollno: "+c.getString(0)+"\n");
            buffer.append("Name: "+c.getString(1)+"\n");
            buffer.append("Marks: "+c.getString(2)+"\n\n");
          }
          showMessage("Student Details", buffer.toString());
        }
```

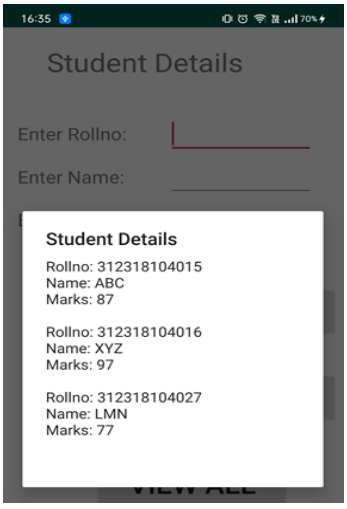
```
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

OUTPUT:









RESULT

Thus a simple android application that makes use of database is implemented successfully.

Ex. No: 5 **NOTIFICATION MANAGER** **Date:**

AIM

To develop a Simple Android Application for notification using Notification manager.

PROCEDURE

1. Creating a New project:

- Open Android Stdio and then click on File -> New -> New project.
- Select the Empty Activity and click Next and Finish.
- In MainActivity.java import notification manager and notification channel
- Create button and set on click listener for notification
- Create two methods makeNotificationChannel() and issueNotification()
- When the button it clicked notification will be displayed with help of notification channel and manager.

2. Designing Layout for Main Activity:

- Click on app -> res -> layout -> activity_main.xml.
- Create Constraint layout and buttons.

SOURCE CODE

activity_main.xml

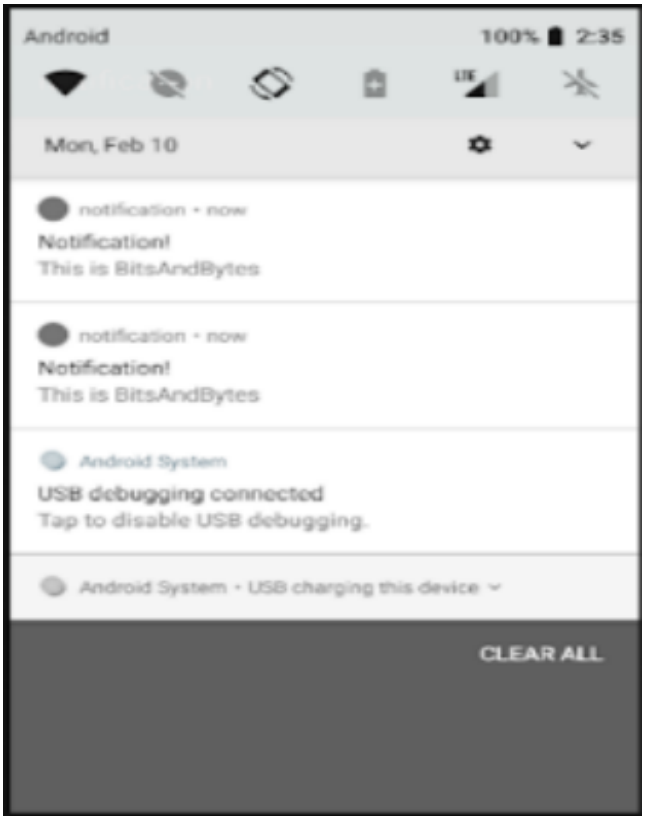
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <Button
      android:id="@+id/notify"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Notify me!"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.ex555;
import androidx.appcompat.app.AppCompatActivity;
import androidx.annotation.RequiresApi;
import androidx.core.app.NotificationCompat;
import android.app.NotificationChannel;
```

```
import android.app.NotificationManager;
import android.content.Context;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Button notify = findViewById(R.id.notify);
      notify.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            issueNotification();
         }
      });
   }
   @RequiresApi(api = Build.VERSION_CODES.O)
   void makeNotificationChannel(String id, String name, int importance)
   {
      NotificationChannel channel = new NotificationChannel(id, name, importance);
      channel.setShowBadge(true);
      NotificationManager notificationManager =
            (NotificationManager)getSystemService(NOTIFICATION_SERVICE);
      assert notificationManager != null;
      notificationManager.createNotificationChannel(channel);
   }
   void issueNotification()
   {
      if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
         makeNotificationChannel("CHANNEL_1", "Example channel",
NotificationManager.IMPORTANCE_DEFAULT);
      }
      NotificationCompat.Builder notification =
            new NotificationCompat.Builder(this, "CHANNEL_1")
                  .setSmallIcon(R.mipmap.ic_launcher)
                  .setContentTitle("Notification!")
                  .setContentText("This is BitsAndBytes")
                  .setNumber(3);
      NotificationManager notificationManager =
            (NotificationManager)getSystemService(NOTIFICATION_SERVICE);
      assert notificationManager != null;
      notificationManager.notify(1, notification.build());
      notificationManager.notify(2, notification.build());
   }
}
```

OUTPUT



RESULT

Thus a simple android application for notification using notification manager is implemented successfully.

Ex. No: 6 **MULTITHREADING** **Date:**

AIM

To develop a Simple Android Application that implements Multithreading.

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. Copy the Images and Paste it in "app -> res -> drawable" by pressing "right click mouse button on *drawable*" and selecting the "*Paste*" option.

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="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical" >

   <ImageView
      android:id="@+id/imageView"
      android:layout_width="250dp"
      android:layout_height="250dp"
      android:layout_margin="50dp"
      android:layout_gravity="center" />

   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_gravity="center"
      android:text="Load Image 1" />

   <Button
      android:id="@+id/button2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_margin="10dp"
      android:layout_gravity="center"
      android:text="Load image 2" />
</LinearLayout>
```

MainActivity.java

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
public class MainActivity extends AppCompatActivity
{
   ImageView img;
   Button bt1,bt2;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      bt1 = (Button)findViewById(R.id.button);
      bt2= (Button) findViewById(R.id.button2);
      img = (ImageView)findViewById(R.id.imageView);
      bt1.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            new Thread(new Runnable(){
               @Override
               public void run(){
                  img.post(new Runnable(){
                     @Override
                     public void run()
                     {
                        img.setImageResource(R.drawable.india1);
                     }
                  });
               }
            }).start();
         }
      });
      bt2.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v){
            new Thread(new Runnable()
            {
               @Override
               public void run()
               {
                  img.post(new Runnable()
                  {
                     @Override
                     public void run()
                     {
                        img.setImageResource(R.drawable.india2);
```

```
                    }
                });
            }
        }).start();
    }
});}}
```

OUTPUT

RESULT

Thus a simple android application that uses multithreading are implemented successfully.

Ex. No: 7 **GPS LOCATION** **Date:**

AIM

To develop an Android Application that uses GPS location information.

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
   - Type the Code for Activity_main.xml with button in relative layout.
3. Java Coding for the Android Application:
   - Click on app -> java -> MainActivity.
   - Create button for show location
   - Set Toast for displaying latitude and longitude information.
   - Call GPSTrace.java for getting GPS location by creating Location manager by enabling GPS provider and network provider.
4. Adding permissions in Manifest for the Android Application:
   - Click on app -> manifests -> AndroidManifest.xml.
   - Now include the INTERNET permissions in the AndroidManifest.xml file.

Selection of the location

Click three dots above the android and select the location. Now the area is indicated in red color. Click SAVEPOINT and note the latitude n longitude given in the right side. Finally click SET LOCATION and run the code again. Click showlocation in the android. It will display the latitude & longitude.

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:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">
   <Button
      android:id="@+id/show_Location"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Show Location"
      android:layout_centerVertical="true"
```

```
    android:layout_centerHorizontal="true" />
</RelativeLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   Button btnShowLocation;
   Context mContext;
   GPSTracker gps;

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      mContext = this;
      btnShowLocation=findViewById(R.id.show_Location);
      btnShowLocation.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            if (ContextCompat.checkSelfPermission(mContext,
                Manifest.permission.ACCESS_FINE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED &&
                ActivityCompat.checkSelfPermission(mContext,
                Manifest.permission.ACCESS_COARSE_LOCATION) !=
                PackageManager.PERMISSION_GRANTED) {
               ActivityCompat.requestPermissions(MainActivity.this,new
                String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
            }
             else {
               gps = new GPSTracker(mContext, MainActivity.this);
               // Check if GPS enabled
               if (gps.canGetLocation()) {
                  double latitude = gps.getLatitude();
                  double longitude = gps.getLongitude();
```

```
                // \n is for new line
                Toast.makeText(getApplicationContext(), "Yoshow_Locationur Location is -
\nLat: " + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
            } else {
                // Can't get location. GPS or network is not enabled.
                // Ask user to enable GPS/network in settings.
                Toast.makeText(mContext, "Location Services off!",
Toast.LENGTH_SHORT).show();
            }
          }
        }
      });
  }
  }
```

GPSTracker.java

```
  package com.example.ex7;
  import android.Manifest;
  import android.app.Activity;
  import android.app.Service;
  import android.content.Context;
  import android.content.DialogInterface;
  import android.content.Intent;
  import android.content.pm.PackageManager;
  import android.location.Location;
  import android.location.LocationListener;
  import android.location.LocationManager;
  import android.os.Bundle;
  import android.os.IBinder;
  import android.provider.Settings;
  import android.util.Log;
  import androidx.appcompat.app.AlertDialog;
  import androidx.core.app.ActivityCompat;
  import androidx.core.content.ContextCompat;
  public class GPSTracker extends Service {
    private Context mContext;
    // Flag for GPS status
    boolean isGPSEnabled = false;
    // Flag for network status
    boolean isNetworkEnabled = false;
    // Flag for GPS status
    boolean canGetLocation = false;
    Location location; // Location
    double latitude; // Latitude
    double longitude; // Longitude
    // The minimum distance to change Updates in meters
```

```
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1000;
  // 10 meters
    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute
    // Declaring a Location Manager
    protected LocationManager locationManager;
    Activity activity;
    public GPSTracker(Context context, Activity activity) {
      this.mContext = context;
      this.activity = activity;
      getLocation();
    }
    private boolean checkPermissions() {
      return ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED;
      // If we want background location on Android 10.0 and higher, use:
      // ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_BACKGROUND_LOCATION) ==
PackageManager.PERMISSION_GRANTED
    }
    public Location getLocation() {
      try {
        locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);
        // Getting GPS status
        isGPSEnabled = locationManager
            .isProviderEnabled(LocationManager.GPS_PROVIDER);
        // Getting network status
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);
        if (!isGPSEnabled && !isNetworkEnabled) {
          // No network provider is enabled
        } else {
          this.canGetLocation = true;
          if (isNetworkEnabled) {
            int requestPermissionsCode = 50;

locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES,
mLocationListener);
            Log.d("Network", "Network");
            if (locationManager != null) {
```

```
                    location =
locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
        }
        // If GPS enabled, get latitude/longitude using GPS Services
        if (isGPSEnabled) {
            if (location == null) {
                if (ContextCompat.checkSelfPermission(activity,
Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(activity,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
                    ActivityCompat.requestPermissions(activity, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 50);
                } else {
                    locationManager.requestLocationUpdates(
                            LocationManager.GPS_PROVIDER,
                            MIN_TIME_BW_UPDATES,
                            MIN_DISTANCE_CHANGE_FOR_UPDATES, mLocationListener);
                    Log.d("GPS Enabled", "GPS Enabled");
                    if (locationManager != null) {
                        location = locationManager
                                .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return location;
    }
    private final LocationListener mLocationListener = new LocationListener() {
        @Override
        public void onLocationChanged(final Location location) {
```

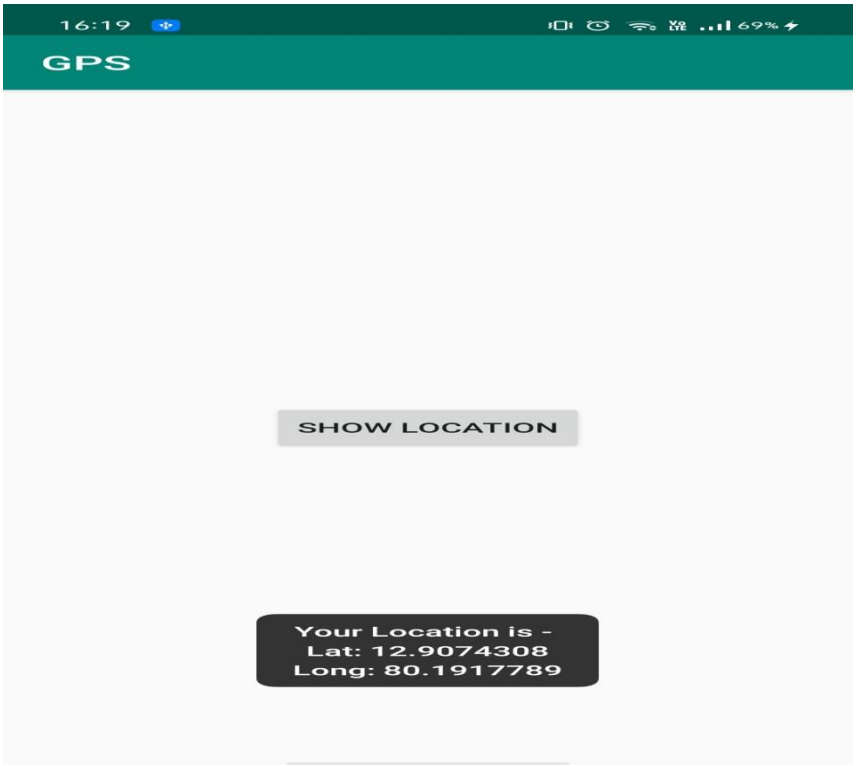
```
            if (location != null) {
                latitude = location.getLatitude();
                longitude = location.getLongitude();
            }
        }
        @Override
        public void onStatusChanged(String provider, int status, Bundle extras) {
        }
        @Override
        public void onProviderEnabled(String provider) {
        }
        @Override
        public void onProviderDisabled(String provider) {
        }
    };
    /**
     * Function to get latitude*
     */
    public double getLatitude() {
        if (location != null) {
            latitude = location.getLatitude();
        }
        // return latitude
        return latitude;
    }
    /* Function to get longitude     */

    public double getLongitude() {
        if (location != null) {
            longitude = location.getLongitude();
        }
        return longitude;
    }
    /* Function to check GPS/Wi-Fi enabled    */
    public boolean canGetLocation() {
        return this.canGetLocation;

    }
    /**
     * Function to show settings alert dialog.
     * On pressing the Settings button it will launch Settings Options.
     */
    @Override
    public IBinder onBind(Intent arg0) {

        return null;
```

```
    }
  }
```

Activitymanifest.xml

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

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

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

OUTPUT



RESULT

Thus a simple android application that uses GPS location information is implemented successfully.

Ex.No:8 **WRITING DATA TO SD CARD** **Date:**

AIM:
To implement an application that writes data to the SD card.

ALGORITHM:
1. Open ANDROID STUDIO.
2. Click File->new->New Project.
3. Give the File name->Choose the Blank Activity and Package name and then click Finish button.

activity_main.xml:

1. Create a EditText box.
2. Create 3 buttons to Save, Show and Clear from the SD card.

MainActivity.java:
1. Declare the EditText and Buttons in the onCreate method.
2. Create a file object to store the External Storage .
3. Create a directory and check if it already exists , if not exists make directory and input a .txt file .
4. Create three methods to read , save and clear the data.
5. In read method use a buffered reader to store the data that is being retrieved and display it in the text field.
6. In save method using OutputStream write the data into the file.

PROGRAM:

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:

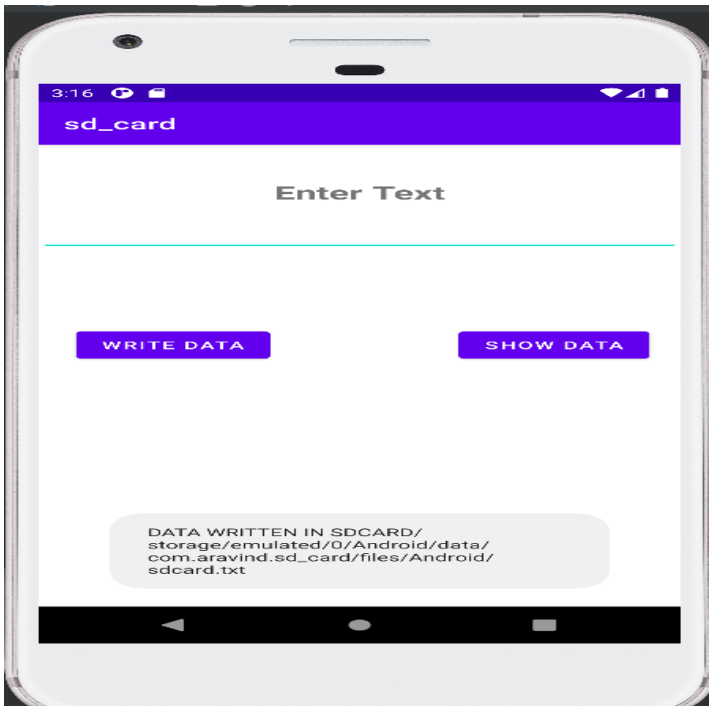
```
<activity
   android:name=".MainActivity2"
   android:exported="false">
</activity>
```

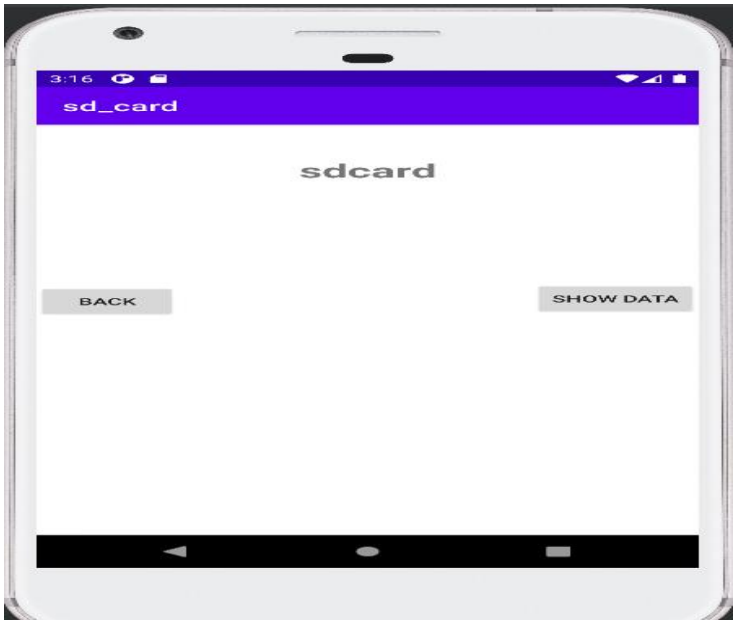
IN RES FOLDER, UPDATE IN strings.xml

```
<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** **Date:**

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) {
```

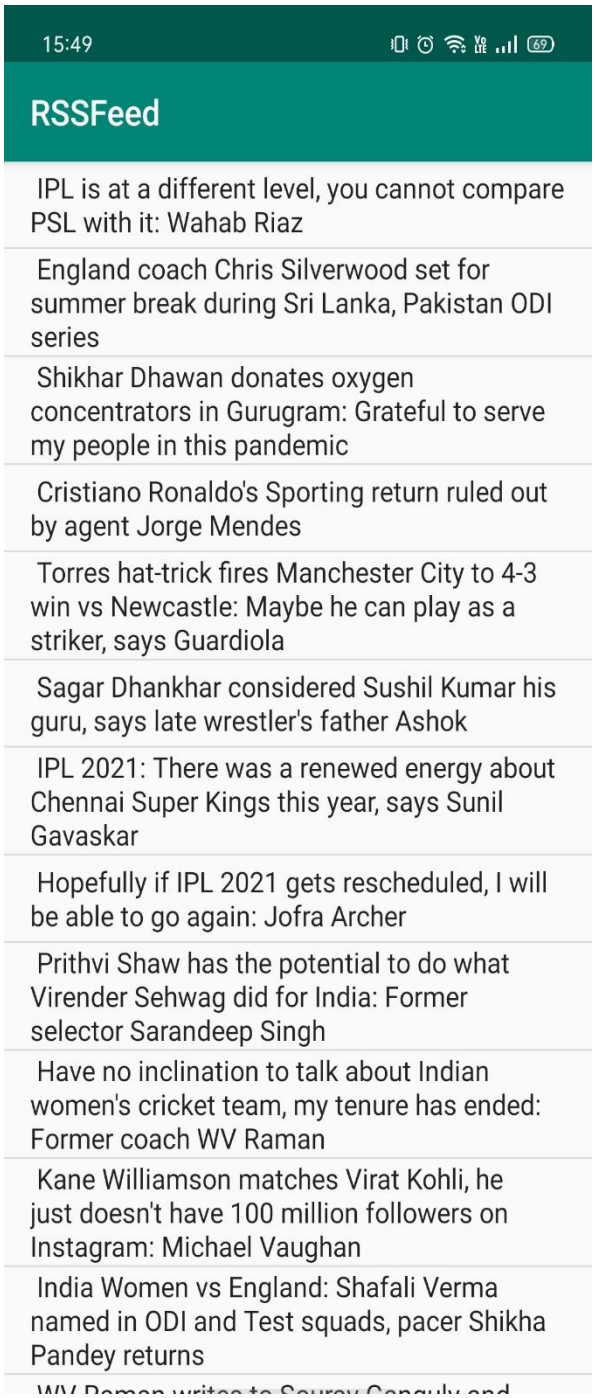
```
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvRss =  findViewById(R.id.listView);
        titles = new ArrayList<>();
        links = new ArrayList<>();

        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) {
```

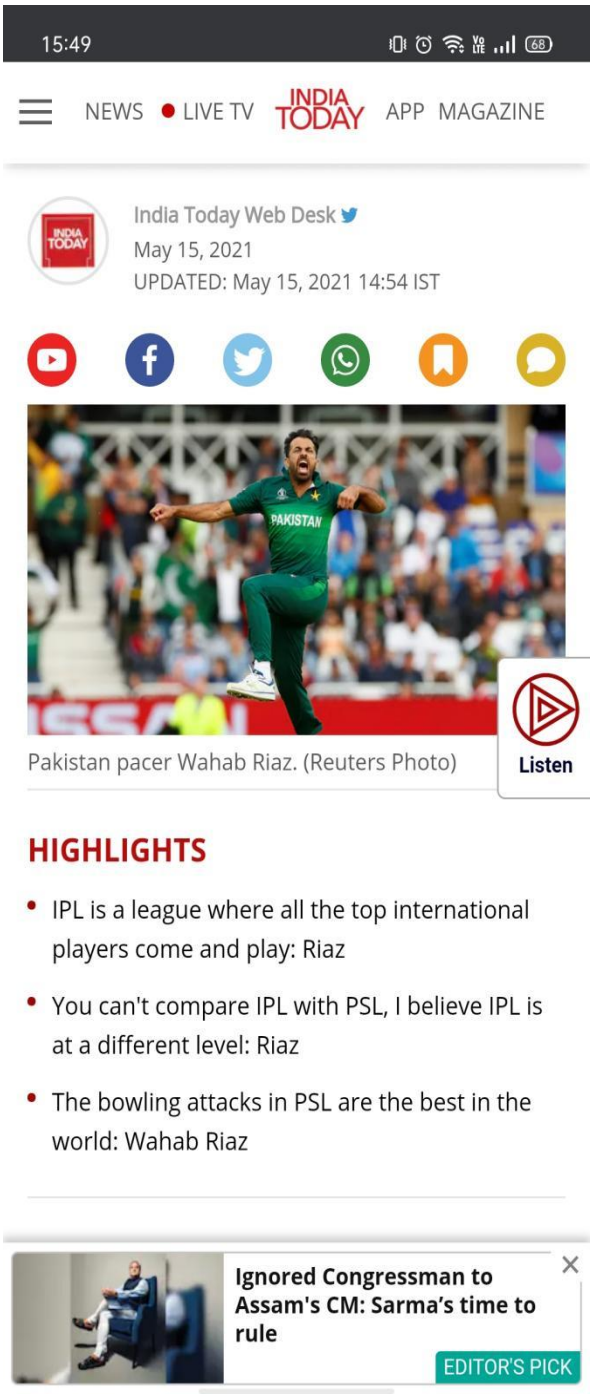
```
        try
        {
           URL url = new URL("https://www.indiatoday.in/rss/1206550");
           //creates new instance of PullParserFactory that can be used to create XML pull
parsers
           XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
           //Specifies whether 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.

Ex. No: 10 **ALARM CLOCK** **Date:**

AIM

To develop an Android Application that creates an Alarm Clock.

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. Creating Second Activity for the Android Application:
   - Click on File -> New -> Activity -> Empty Activity.
   - Type the Activity Name as AlarmReceiver and click Finish button.
3. Designing layout for the Android Application:
   - Click on app -> res -> layout -> activity_main.xml.
   - Create Relative layout with Toggle button and Time Picker
4. Changes in Manifest for the Android Application:
   - Click on app -> manifests -> AndroidManifest.xml
   - Now change the activity tag to receiver tag in the AndroidManifest.xml file.

SOURCE CODE

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity">
   <!-- Added Time Picker just to pick alarm time -->
   <TimePicker
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_gravity="center"
      android:id="@+id/timePicker"/>
   <!-- Added Toggle Button to set the alarm on or off-->
   <ToggleButton
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/toggleButton"
      android:layout_gravity="center"
      android:layout_margin="20dp"
      android:checked="false"
      android:onClick="OnToggleClicked" />
</LinearLayout>
```

MainActivity.java

```
package com.ex.alarm;
import androidx.appcompat.app.AppCompatActivity;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.TimePicker;
import android.widget.Toast;
import android.widget.ToggleButton;
import java.util.Calendar;

public class MainActivity extends AppCompatActivity {
   TimePicker alarmTimePicker;
   PendingIntent pendingIntent;
   AlarmManager alarmManager;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      alarmTimePicker = findViewById(R.id.timePicker);
      alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
   }

   public void OnToggleClicked(View view){
      long time;
      if(((ToggleButton)view).isChecked()){
         Toast.makeText(MainActivity.this,"ALARM ON",Toast.LENGTH_SHORT).show();
         Calendar calendar = Calendar.getInstance();
         // calendar is called to get current time in hour and minute
         calendar.set(Calendar.HOUR_OF_DAY,alarmTimePicker.getCurrentHour());
         calendar.set(Calendar.MINUTE,alarmTimePicker.getCurrentMinute());
         // using intent i have class AlarmReceiver class which inherits BroadCastReceiver
         Intent intent = new Intent(this,AlarmReceiver.class);
         // we call broadcast using pendingIntent
         pendingIntent = PendingIntent.getBroadcast(this,0,intent,0);
         time = (calendar.getTimeInMillis() - (calendar.getTimeInMillis() % 60000));
         if(System.currentTimeMillis() > time ){
            if(Calendar.AM_PM == 0)
               time = time + (1000*60*60*12);
            else
               time = time + (1000*60*60*24);
         }
```

```
        // Alarm rings continuously until toggle button button is turned off

alarmManager.setRepeating(AlarmManager.RTC_WAKEUP,time,10000,pendingIntent);
      }
      else{
         alarmManager.cancel(pendingIntent);
         Toast.makeText(MainActivity.this,"ALARM OFF",Toast.LENGTH_SHORT).show();
      }
   }
}
```

AlarmReceiver.java

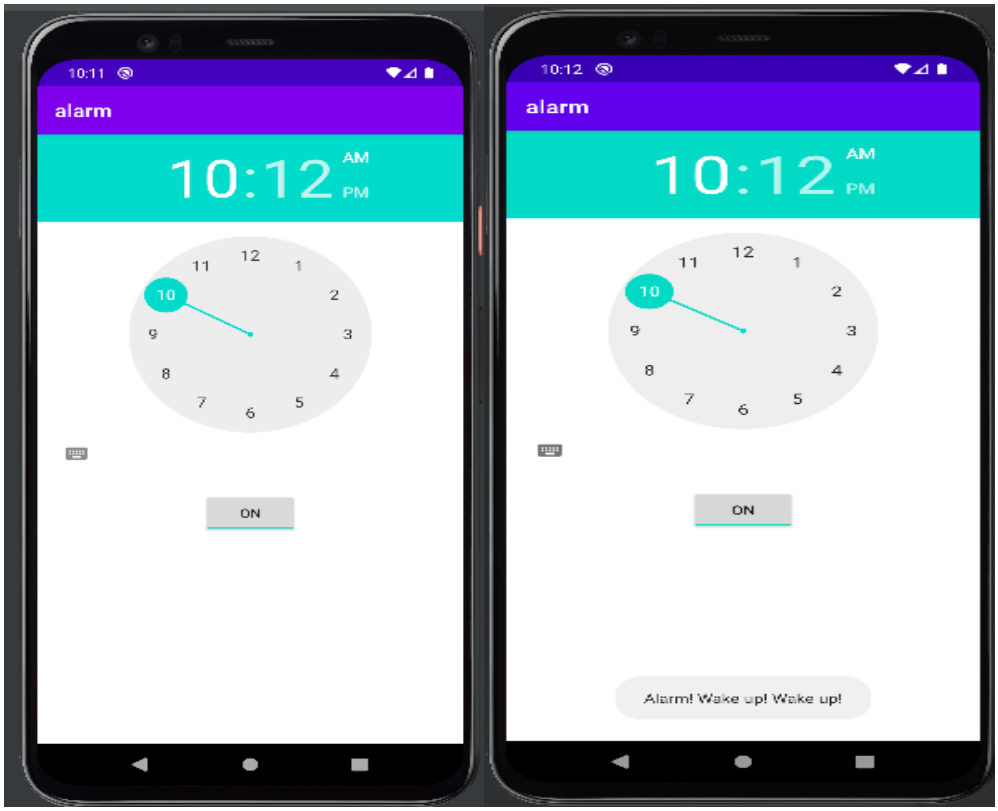
```
package com.ex.alarm;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Vibrator;
import android.widget.Toast;
public class AlarmReceiver extends BroadcastReceiver {
   //implement OnReceive Method
   @Override
   public void onReceive(Context context, Intent intent) {
      // we will use vibrator first
      Vibrator vibrator = (Vibrator) context.getSystemService(Context.VIBRATOR_SERVICE);
      vibrator.vibrate(4000);
      Toast.makeText(context,"Alarm! Wake up! Wake up!",Toast.LENGTH_SHORT).show();
      Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
      if(alarmUri == null){
         alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
      }
      // setting default ringtone
      Ringtone ringtone = RingtoneManager.getRingtone(context,alarmUri);
      ringtone.play();
   }
}
```

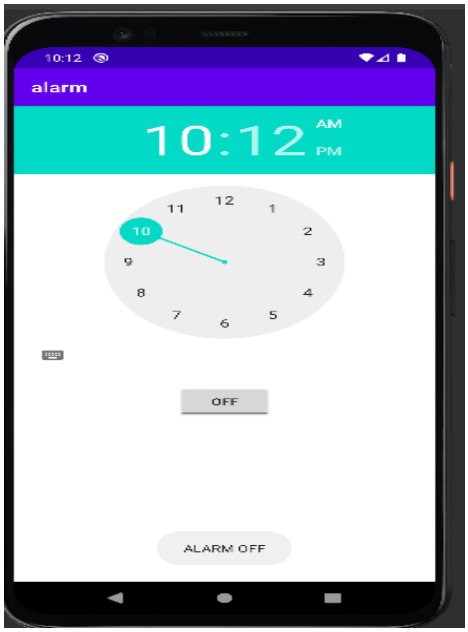
Manifest.xml

```
BEFORE TO  <application TYPE THE FOLLOWING LINE:
   <uses-permission android:name="android.permission.VIBRATE" />
        AFTER TO  </activity> TYPE THE FOLLOWING LINE:
   <receiver android:name=".AlarmReceiver">
```

```
    </receiver>
    </application>
</manifest>
```

OUTPUT:





RESULT

Thus an Android Application that creates an Alarm Clock is implemented successfully.

Ex. No: 11 **NATIVE CALCULATOR** **Date:**

AIM

To develop an Android Application for Native Calculator.

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 two edit text
- Create another Linear Layout for four buttons add ,sub, multiply and divide

3. Java Coding for the Android Application

- Click on app -> java -> MainActivity.
- Declare edit texts for getting two numbers
- Declare four buttons for addition, subtraction, Multiplication and division operation.
- Buffer the views and set the listeners
- Dheck if the fields are empty and read EditText and fill variables with numbers

Define the button that has been clicked and performs the corresponding operation using switch case to perform particular operation.

SOURCE CODE

MainActivity.java

```
package com.ex.calc;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
   Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bp,bm,bmu,bdo,bd,be,bclr;
   EditText tx;     float val1=0,val2=0,res=0;     String op="";
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      b1=(Button)findViewById(R.id.one);
      b2=(Button)findViewById(R.id.two);
      b3=(Button)findViewById(R.id.three);
      b4 =(Button)findViewById(R.id.four);
      b5=(Button)findViewById(R.id.five);
      b6=(Button)findViewById(R.id.six);
```

```
    b7=(Button)findViewById(R.id.seven);
    b8=(Button)findViewById(R.id.eight);
    b9=(Button)findViewById(R.id.nine);
    b0=(Button)findViewById(R.id.zero);
    bp=(Button)findViewById(R.id.plus);
    bm=(Button)findViewById(R.id.minus);
    bmu=(Button)findViewById(R.id.multi);
    bdo=(Button)findViewById(R.id.dot);
    bd=(Button)findViewById(R.id.divide);
    be=(Button)findViewById(R.id.equal);
    bclr=(Button)findViewById(R.id.clear);
    tx=(EditText)findViewById(R.id.txt);
}
public void show(View v){
    Button b=(Button)v;
    String set=b.getText().toString();
    tx.append(set);
}
public void calc(View v){
    Button b=(Button)v;
    op=b.getText().toString();
    val1=Float.parseFloat(tx.getText().toString());
    tx.setText("");
}
public void equal(View v){
    val2=Float.parseFloat(tx.getText().toString());
    switch(op) {
        case "+":
            res =val1+val2;
            tx.setText(String.valueOf(res));
            break;
        case "-":
            res =val1-val2;
            tx.setText(String.valueOf(res));
            break;
        case "*":
            res =val1*val2;
            tx.setText(String.valueOf(res));
            break;
        case "/":
            res =val1/val2;
            tx.setText(String.valueOf(res));
            break;
        default:
    }
```

```
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:paddingTop="5dp"
    android:orientation="vertical"
    android:paddingBottom="5dp"
    tools:context=".MainActivity">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <EditText
            android:layout_width="fill_parent"
            android:layout_height="50dp"
            android:layout_marginLeft="25dp"
            android:gravity="right"
            android:focusableInTouchMode="true"
            android:id="@+id/txt"
            android:layout_marginRight="25dp"
            tools:ignore="RtlHardcoded" />
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:layout_marginTop="20dp"
        android:orientation="horizontal">
        <Button
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:text="9"
            android:onClick="show"
            android:id="@+id/nine"
            android:textSize="30dp"/>
        <Button
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:onClick="show"
            android:id="@+id/eight"
            android:text="8"
```

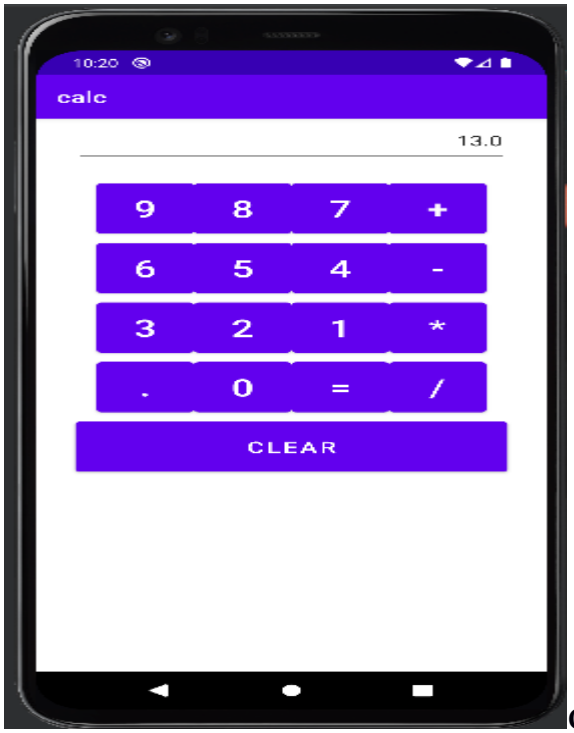
```
        android:textSize="30dp"/>
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="7"
        android:onClick="show"
        android:id="@+id/seven"
        android:textSize="30dp"/>
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:onClick="calc"
        android:id="@+id/plus"
        android:text="+"
        android:textSize="30dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:onClick="show"
        android:id="@+id/six"
        android:text="6"
        android:textSize="30dp"/>
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="5"
        android:onClick="show"
        android:id="@+id/five"
        android:textSize="30dp"/>
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="4"
        android:onClick="show"
        android:id="@+id/four"
        android:textSize="30dp"/>
    <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
```

```
        android:text="-"
        android:onClick="calc"
        android:id="@+id/minus"
        android:textSize="30dp"/>
    </LinearLayout>
    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:orientation="horizontal">
      <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="3"
        android:onClick="show"
        android:id="@+id/three"
        android:textSize="30dp"/>
      <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="2"
        android:onClick="show"
        android:id="@+id/two"
        android:textSize="30sp"/>
      <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="1"
        android:onClick="show"
        android:id="@+id/one"
        android:textSize="30sp"/>
      <Button
        android:layout_width="75dp"
        android:layout_height="75dp"
        android:text="*"
        android:onClick="calc"
        android:id="@+id/multi"
        android:textSize="30sp"/>
    </LinearLayout>
    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:gravity="center"
      android:orientation="horizontal">
      <Button
```

```
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:onClick="show"
            android:id="@+id/dot"
            android:text="."
            android:textSize="30sp"/>
        <Button
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:text="0"
            android:onClick="show"
            android:id="@+id/zero"
            android:textSize="30sp"/>
        <Button
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:text="="
            android:onClick="equal"
            android:id="@+id/equal"
            android:textSize="30sp"/>
        <Button
            android:layout_width="75dp"
            android:layout_height="75dp"
            android:text="/"
            android:onClick="calc"
            android:id="@+id/divide"
            android:textSize="30sp"/>
    </LinearLayout>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
        <Button
            android:layout_width="fill_parent"
            android:layout_height="75dp"
            android:text="CLEAR"
            android:onClick="calc"
            android:id="@+id/clear"
            android:layout_marginLeft="25dp"
            android:layout_marginRight="25dp"
            android:textSize="20sp"/>
    </LinearLayout>
</LinearLayout>
```

OUTPUT

Click "/" first and then type 5.



Output is displayed after clicking "=" : 65/5=13. Click "CLEAR" to do the next calculation.

RESULT

Thus an Android Application for Native Calculator has been implemented successfully.

Ex. No 12 **Online Shopping** **Date:**

AIM

To create an online shopping application using android

ALGORITHM

1. Create the layouts for the app.
2. Create the Product Class
3. Create the ShoppingCartHelper class
4. Create the Item Layout
5. Create the ProductAdapter class
6. Create the CatalogActivity
7. Create the ProductDetails Activity
8. Finally, create the ShoppingCart Activity

Program

catalog.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="fill_parent"
android:layout_height="fill_parent" android:background="#ffffff">
<TextView android:id="@+id/TextView01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:textColor="#000000"
android:textSize="24dip" android:layout_margin="5dip" android:text="Product
Catalog"></TextView>
<ListView android:layout_height="wrap_content"
android:layout_weight="1" android:id="@+id/ListViewCatalog"
android:layout_width="fill_parent" android:background="#ffffff"
android:clickable="true" android:cacheColorHint="#ffffff">
</ListView>
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_margin="5dip"
android:layout_gravity="right" android:id="@+id/ButtonViewCart"
android:text="View Shopping Cart"></Button>
</LinearLayout>
```

productdetails.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent" android:layout_width="fill_parent"
android:background="#ffffff" android:orientation="vertical">
<LinearLayout android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/LinearLayoutHeader"
android:orientation="horizontal">
<ImageView android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/ImageViewProduct"
android:adjustViewBounds="true" android:scaleType="fitXY"
android:src="@drawable/deadoralive" android:layout_margin="5dip"></ImageView>
<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/TextViewProductTitle"
android:layout_gravity="center" android:layout_margin="5dip"
```

```
android:textSize="26dip" android:text="Dead or Alive"
android:textColor="#000000"></TextView>
</LinearLayout>
<TextView android:layout_height="wrap_content"
android:id="@+id/TextViewProductDetails"
android:layout_width="fill_parent" android:layout_margin="5dip"
android:layout_weight="1" android:textColor="#000000" android:text="Product
description"></TextView>
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_margin="5dip"
android:layout_gravity="right" android:id="@+id/ButtonAddToCart"
android:text="Add to Cart"></Button>
</LinearLayout>
```

shoppingcart.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_height="fill_parent"
android:layout_width="fill_parent" android:background="#ffffff">

<TextView android:id="@+id/TextView01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:textColor="#000000"
android:textSize="24dip" android:layout_margin="5dip" android:text="Shopping
Cart"></TextView>
<ListView android:layout_height="wrap_content"
android:layout_weight="1" android:id="@+id/ListViewCatalog"
android:layout_width="fill_parent" android:background="#ffffff"
android:cacheColorHint="#ffffff" android:clickable="true"
android:choiceMode="multipleChoice">

</ListView>
<LinearLayout android:id="@+id/LinearLayout01"
android:layout_width="wrap_content" android:layout_height="wrap_content"
android:orientation="horizontal" android:layout_gravity="right"
android:layout_margin="5dip">
<Button android:layout_width="wrap_content"
android:layout_height="wrap_content" android:id="@+id/ButtonRemoveFromCart"
android:text="Remove from Cart"></Button>
<Button android:id="@+id/Button02" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Proceed to Checkout"></Button>
</LinearLayout>
</LinearLayout>
```

Mainactivity.java

```
package com.dreamdom.tutorials.shoppingcart;
import android.graphics.drawable.Drawable;
public class Product {
public String title;
public Drawable productImage;
public String description;
public double price;
public boolean selected;
```

```
public Product(String title, Drawable productImage, String description,
double price) {
this.title = title;
this.productImage = productImage;
this.description = description;
this.price = price;
}
}
```

ShoppingCartHelper.java

```
package com.dreamdom.tutorials.shoppingcart;
import java.util.List;
import java.util.Vector;
import android.content.res.Resources;
public class ShoppingCartHelper {
public static final String PRODUCT_INDEX = "PRODUCT_INDEX";
private static List<Product> catalog;
private static List<Product> cart;
public static List<Product> getCatalog(Resources res){
if(catalog == null) {
catalog = new Vector<Product>();
catalog.add(new Product("Dead or Alive", res
.getDrawable(R.drawable.deadoralive),
"Dead or Alive by Tom Clancy with Grant Blackwood", 29.99));
catalog.add(new Product("Switch", res
.getDrawable(R.drawable.switchbook),
"Switch by Chip Heath and Dan Heath", 24.99));
catalog.add(new Product("Watchmen", res
.getDrawable(R.drawable.watchmen),
"Watchmen by Alan Moore and Dave Gibbons", 14.99));
}
return catalog;
}

public static List<Product> getCart() {
if(cart == null) {
cart = new Vector<Product>();
}
return cart;
}
}
```

item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="wrap_content" android:orientation="horizontal"
android:layout_width="fill_parent" android:id="@+id/LinearLayoutItem">
<ImageView android:layout_margin="5dip" android:id="@+id/ImageViewItem"
android:layout_height="wrap_content" android:layout_width="100dip"></ImageView>
<TextView android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_gravity="center"
android:layout_margin="5dip" android:id="@+id/TextViewItem"
```

```
android:textSize="26dip" android:text="Book Title" android:textColor="#000000"
android:minLines="2" android:maxWidth="150dip"></TextView>
<TextView android:id="@+id/TextView01" android:layout_width="wrap_content"
android:layout_height="wrap_content" android:layout_weight="1"></TextView>
<CheckBox android:layout_height="wrap_content"
android:layout_margin="5dip" android:id="@+id/CheckBoxSelected"
android:focusable="false" android:clickable="false"
android:layout_gravity="center" android:layout_width="wrap_content"></CheckBox>
</LinearLayout>
```

ProductAdapter.java

```
package com.dreamdom.tutorials.shoppingcart;
import java.util.List;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.CheckBox;
import android.widget.ImageView;
import android.widget.TextView;

public class ProductAdapter extends BaseAdapter {
private List<Product> mProductList;
private LayoutInflater mInflater;
private boolean mShowCheckbox;
public ProductAdapter(List<Product> list, LayoutInflater inflater, boolean showCheckbox) {
mProductList = list;
mInflater = inflater;
mShowCheckbox = showCheckbox;
}
@Override
public int getCount() {
return mProductList.size();
}
@Override
public Object getItem(int position) {
return mProductList.get(position);
}
@Override
public long getItemId(int position) {
return position;
}
@Override
public View getView(int position, View convertView, ViewGroup parent) {
final ViewItem item;

if (convertView == null) {
convertView = mInflater.inflate(R.layout.item,
null);
item = new ViewItem();
```

```
item.productImageView = (ImageView) convertView
.findViewById(R.id.ImageViewItem);
item.productTitle = (TextView) convertView.findViewById(R.id.TextViewItem);
item.productCheckbox = (CheckBox) convertView.findViewById(R.id.CheckBoxSelected);
convertView.setTag(item);
} else {
item = (ViewItem) convertView.getTag();
}
Product curProduct = mProductList.get(position);
item.productImageView.setImageDrawable(curProduct.productImage);
item.productTitle.setText(curProduct.title);
if(!mShowCheckbox) {
item.productCheckbox.setVisibility(View.GONE);
} else {
if(curProduct.selected == true)
item.productCheckbox.setChecked(true);
else
item.productCheckbox.setChecked(false);
}
return convertView;
}
private class ViewItem {
ImageView productImageView;
TextView productTitle;
CheckBox productCheckbox;
}
}
```

OUTPUT :

Result

Thus the Online shopping using android has been sucessfully executed.