Ex.No: 06

DEVELOP AN APPLICATION THAT USES RSS FEED

Aim:

To develop an application that uses RSS feed.

Procedure:

- Open Android Studio and then click on File -> New -> New project
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project.
- Click on app -> res -> layout -> activity_main.xml
- Now click on Text as shown below. Delete the code which is there and type the code as given below.
- Click on app -> manifests -> AndroidManifest.xml
- Now include the INTERNET permissions in the AndroidManifest.xml file
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Delete the code which is there and type the code as given below.
- Now run the application to see the output

Code:

MainActivity.java

```
package com.example.myapplication;
import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
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;
import java.util.List;
public class MainActivity extends ListActivity
{
    List headlines;
    List links;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        new MyAsyncTask().execute();
    }
    class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
    {
        @Override
        protected ArrayAdapter doInBackground(Object[] params)
        {
            headlines = new ArrayList();
            links = new ArrayList();
            try
            {
                URL url = new URL("https://codingconnect.net/feed");
                XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
                factory.setNamespaceAware(false);
                XmlPullParser xpp = factory.newPullParser();
                xpp.setInput(getInputStream(url), "UTF_8");
                boolean insideItem = false;
                int eventType = xpp.getEventType();
                while (eventType != XmlPullParser.END_DOCUMENT)
                {
                    if (eventType == XmlPullParser.START_TAG)
                    {
                        if (xpp.getName().equalsIgnoreCase("item"))
                        {
                            insideItem = true;
                        }
                        else if (xpp.getName().equalsIgnoreCase("title"))
                        {
                            if (insideItem)
                                headlines.add(xpp.nextText()); //extract the headline
                        }
                        else if (xpp.getName().equalsIgnoreCase("link"))
                        {
                            if (insideItem)
                                links.add(xpp.nextText()); //extract the link of article
                        }
                    }
                    else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
                    {
                        insideItem=false;
                    }
                    eventType = xpp.next(); //move to next element
                }
            }
            catch (MalformedURLException e)
            {
                e.printStackTrace();
            }
            catch (XmlPullParserException e)
            {
                e.printStackTrace();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
            return null;
        }
        protected void onPostExecute(ArrayAdapter adapter)
        {
            adapter = new ArrayAdapter(MainActivity.this,
android.R.layout.simple_list_item_1, headlines);
            setListAdapter(adapter);
        }
    }
    @Override
    protected void onListItemClick(ListView l, View v, int position, long id)
    {
        Uri uri = Uri.parse((links.get(position)).toString());
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }
    public InputStream getInputStream(URL url)
    {
        try
        {
            return url.openConnection().getInputStream();
        }
        catch (IOException e)
        {
            return null;
        }
    }
}
```

Activity_main.xml:

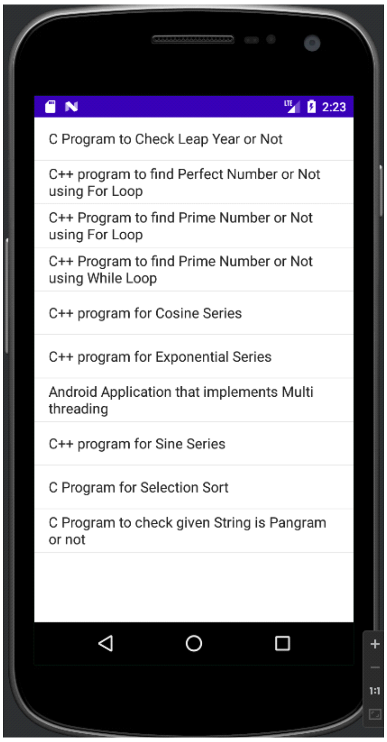
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <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.myapplication">
    <uses-permission android:name="android.permission.INTERNET"/>
    <application>
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:theme="@style/Theme.MyApplication">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Output:



C Program to Check Leap Year or Not

C++ program to find Perfect Number or Not using For Loop

C++ Program to find Prime Number or Not using For Loop

C++ Program to find Prime Number or Not using While Loop

C++ program for Cosine Series

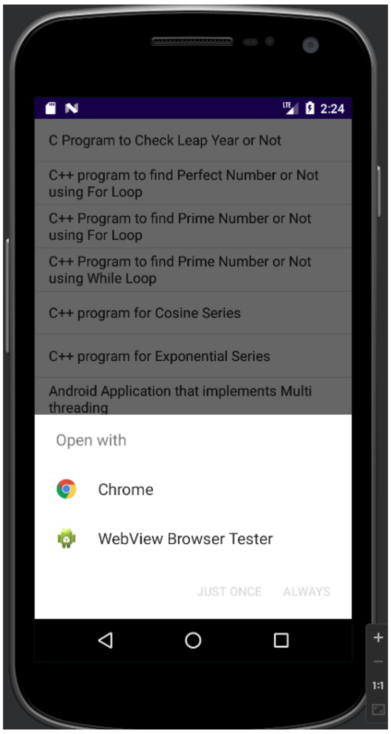
C++ program for Exponential Series

Android Application that implements Multi threading

C++ program for Sine Series

C Program for Selection Sort

C Program to check given String is Pangram or not



C Program to Check Leap Year or Not

C++ program to find Perfect Number or Not using For Loop

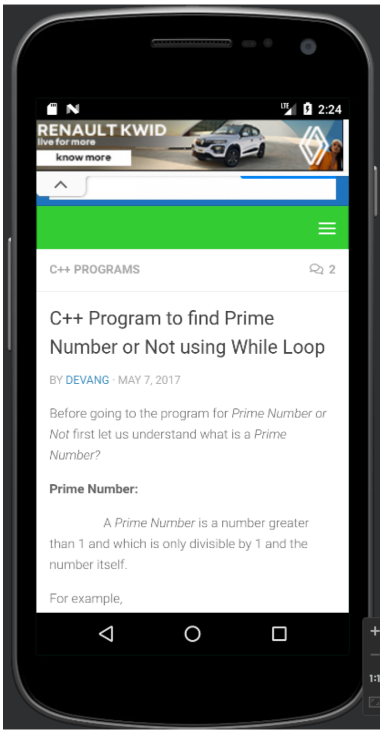
C++ Program to find Prime Number or Not using For Loop

C++ Program to find Prime Number or Not using While Loop

C++ program for Cosine Series

C++ program for Exponential Series

Android Application that implements Multi threading



C++ PROGRAMS 2

C++ Program to find Prime Number or Not using While Loop

BY DEVANG · MAY 7, 2017

Before going to the program for *Prime Number or Not* first let us understand what is a *Prime Number?*

Prime Number:

A *Prime Number* is a number greater than 1 and which is only divisible by 1 and the number itself.

For example,

Result:

Thus, an application that uses RSS feed has been implemented successfully.

Ex. No: 07

DEVELOP AN APPLICATION THAT IMPLEMENTS MULTI THREADING

Aim:

To develop an application that implements multithreading.

Procedure:

- Open Android Studio and then click on **File -> New -> New project.**
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project
- Click on app -> res -> layout -> activity_main.xml
- Now click on **Text** as shown below. Then delete the code which is there and type the code as given below.
- Click on app -> java -> com.example.exno7 -> MainActivity.
- Then delete the code which is there and type the code as given below.
- Before Running the Application, Copy the Images given below and Paste it in “app -> res -> drawable” by pressing “right click mouse button on drawable” and selecting the “Paste” option.
- Now run the application to see the output.

Code:

MainActivity.java

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

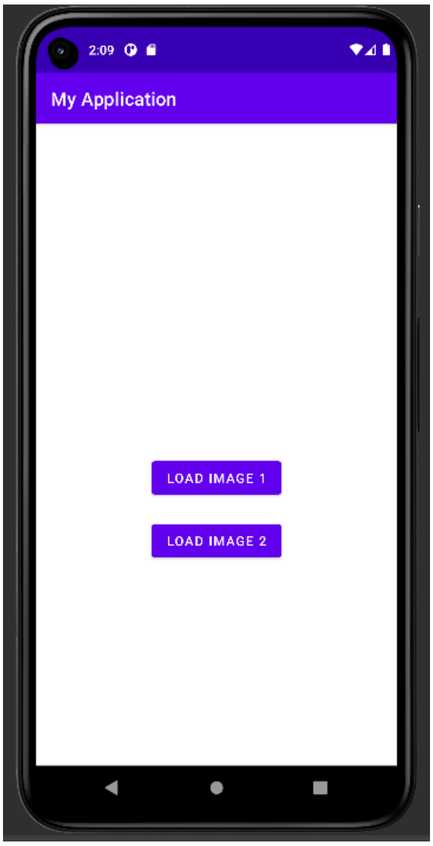
```
protected void onCreate(Bundle savedInstanceState)
{
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   bt1 = (Button)findViewById(R.id.button);
   bt2= (Button) findViewById(R.id.button2);
   img = (ImageView)findViewById(R.id.imageView);
   bt1.setOnClickListener(new 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);
                  }
```

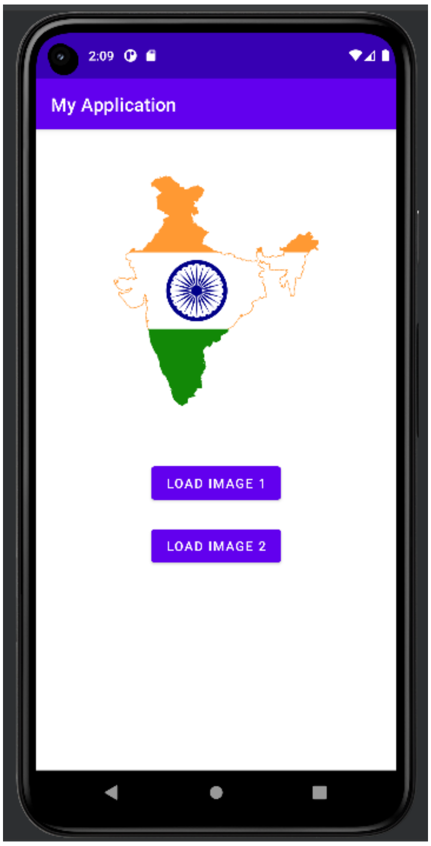
```
                        });
                    }
                }).start();
            }
        });
    }
}
```

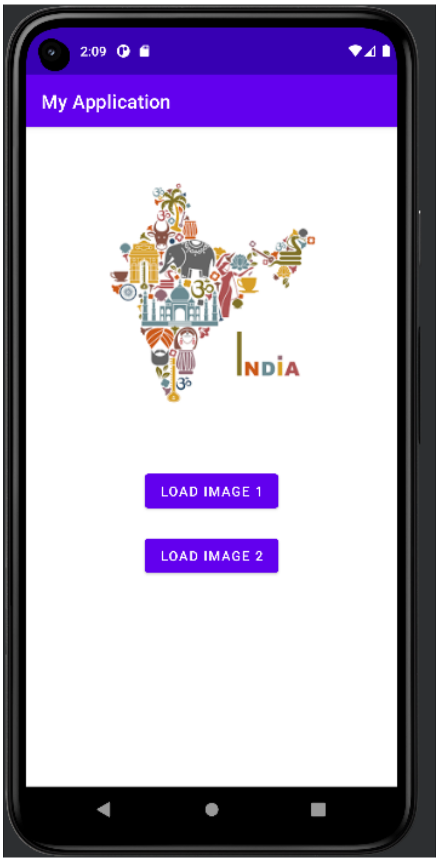
activity_main.xml

```
<?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>
```

Output:







Result:

Thus, an application that implements multithreading has been implemented successfully.

Ex.No: 08

DEVELOP A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION

Aim:

To develop a native application that uses GPS location information.

Procedure:

- Open Android Studio and then click on File -> New -> New project
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project.
- Click on app -> res -> layout -> activity_main.xml
- Now click on Text as shown below. Delete the code which is there and type the code as given below.
- Click on app -> manifests -> AndroidManifest.xml
- Now include the INTERNET permissions in the AndroidManifest.xml file
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Delete the code which is there and type the code as given below.
- Now run the application to see the output

Code:

Main_Activity.java

```
package com.example.myapplication;
import android.app.Activity;
import android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;

import android.widget.Toast;

public class LbsGeocodingActivity extends Activity {

    private static final long MINIMUM_DISTANCE_CHANGE_FOR_UPDATES = 1; //
in Meters

    private static final long MINIMUM_TIME_BETWEEN_UPDATES = 1000; // in
Milliseconds

    protected LocationManager locationManager;

    protected Button retrieveLocationButton;

    @Override

    public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.main);

        retrieveLocationButton = (Button) findViewById(R.id.retrieve_location_button);

        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);

        locationManager.requestLocationUpdates(

                LocationManager.GPS_PROVIDER,

                MINIMUM_TIME_BETWEEN_UPDATES,

                MINIMUM_DISTANCE_CHANGE_FOR_UPDATES,

                new MyLocationListener()

        );

    retrieveLocationButton.setOnClickListener(new OnClickListener() {

            @Override

            public void onClick(View v) {

                showCurrentLocation();

            }

    });

    }

    protected void showCurrentLocation() {

        Location location =
locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);
```

```
        if (location != null) {
            String message = String.format(
                    "Current Location \n Longitude: %1$s \n Latitude: %2$s",
                    location.getLongitude(), location.getLatitude()
            );
            Toast.makeText(LbsGeocodingActivity.this, message,
                    Toast.LENGTH_LONG).show();
        }
    }
    private class MyLocationListener implements LocationListener {
        public void onLocationChanged(Location location) {
            String message = String.format(
                    "New Location \n Longitude: %1$s \n Latitude: %2$s",
                    location.getLongitude(), location.getLatitude()
            );
            Toast.makeText(LbsGeocodingActivity.this, message,
Toast.LENGTH_LONG).show();
        }
        public void onStatusChanged(String s, int i, Bundle b) {
            Toast.makeText(LbsGeocodingActivity.this, "Provider status changed",
                    Toast.LENGTH_LONG).show();
        }
        public void onProviderDisabled(String s) {
            Toast.makeText(LbsGeocodingActivity.this,
                    "Provider disabled by the user. GPS turned off",
                    Toast.LENGTH_LONG).show();
        }
        public void onProviderEnabled(String s) {
            Toast.makeText(LbsGeocodingActivity.this,
                    "Provider enabled by the user. GPS turned on",
                    Toast.LENGTH_LONG).show();
        }
    }
}
```

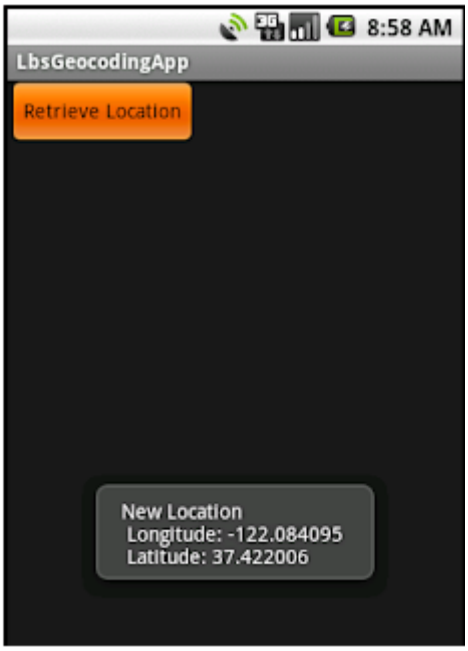
activity_main.xml

```
<?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">
<Button
 android:id="@+id/retrieve_location_button"
 android:text="Retrieve Location"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"/>
</LinearLayout>
```

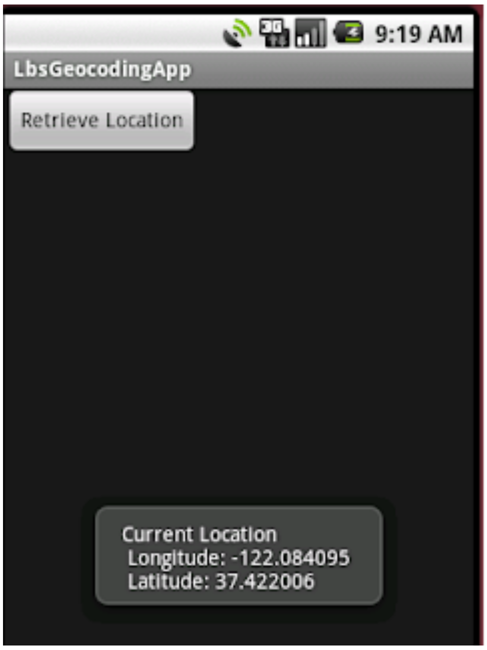
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.javacodegeeks.android.lbs"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".LbsGeocodingActivity"
                  android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
 <uses-permission
android:name="android.permission.ACCESS_MOCK_LOCATION" />
 <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-sdk android:minSdkVersion="3" />
</manifest>
```

Output:



New Location
Longitude: -122.084095
Latitude: 37.422006



Current Location
Longitude: -122.084095
Latitude: 37.422006

Result:

Thus, a native application that uses GPS location information has been implemented successfully.

Ex.No: 9

IMPLEMENT AN APPLICATION THAT WRITES DATA TO THE SD CARD

Aim:

To implement an application that writes data to the SD card

Procedure:

- Open Android Studio and then click on **File -> New -> New project.**
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project.
- Click on app -> res -> layout -> activity_main.xml.
- Now click on Text as shown below. Then delete the code which is there and type the code as given below.
- Click on app -> manifests -> AndroidManifest.xml
- Now include the WRITE_EXTERNAL_STORAGE permissions in the AndroidManifest.xml file as shown below
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Then delete the code which is there and type the code as given below.
- Now run the application to see the output

Code:

MainActivity.java

```
package com.example.myapplication;

import android.os.Bundle;

import androidx.appcomapt.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
public class MainActivity extends AppCompatActivity
{
    EditText e1;
    Button write,read,clear;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        e1= (EditText) findViewById(R.id.editText);
        write= (Button) findViewById(R.id.button);
        read= (Button) findViewById(R.id.button2);
        clear= (Button) findViewById(R.id.button3);
        write.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message=e1.getText().toString();
                try
                {
                    File f=new File("/sdcard/myfile.txt");
                    f.createNewFile();
                    FileOutputStream fout=new FileOutputStream(f);
                    fout.write(message.getBytes());
                    fout.close();
                    Toast.makeText(getBaseContext(),"Data Written in SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH_LONG).show();
                }
            }
        });
        read.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                String message;
                String buf = "";
                try
                {
                    File f = new File("/sdcard/myfile.txt");
                    FileInputStream fin = new FileInputStream(f);
                    BufferedReader br = new BufferedReader(new InputStreamReader(fin));
                    while ((message = br.readLine()) != null)
                    {
                        buf += message;
                    }
                    e1.setText(buf);
                    br.close();
                    fin.close();
                    Toast.makeText(getBaseContext(),"Data Recived from
SDCARD",Toast.LENGTH_LONG).show();
                }
                catch (Exception e)
                {
                    Toast.makeText(getBaseContext(), e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            }
        });
        clear.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v)
            {
                e1.setText("");
            }
        });
    }
}
```

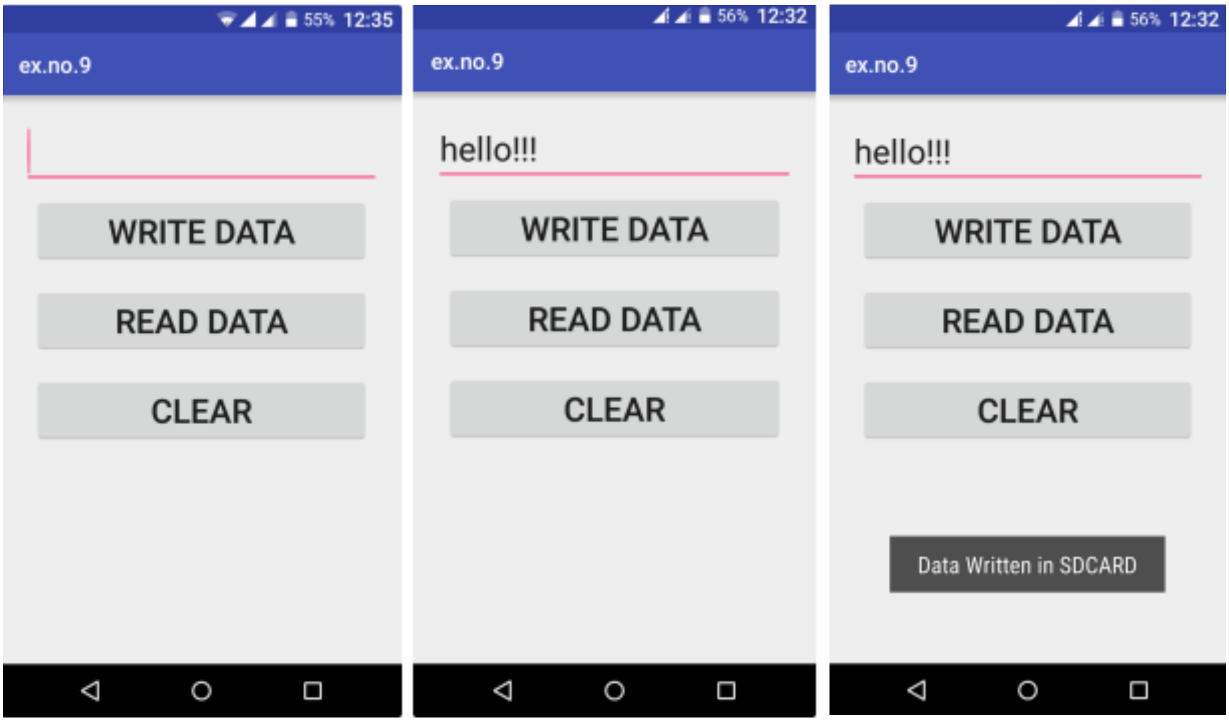
activity_main.xml

```
<?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:layout_margin="20dp"
    android:orientation="vertical">
    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30dp" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Write Data"
        android:textSize="30dp" />
    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Read data"
        android:textSize="30dp" />
    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Clear"
        android:textSize="30dp" />
</LinearLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno9" >
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
    <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>
```

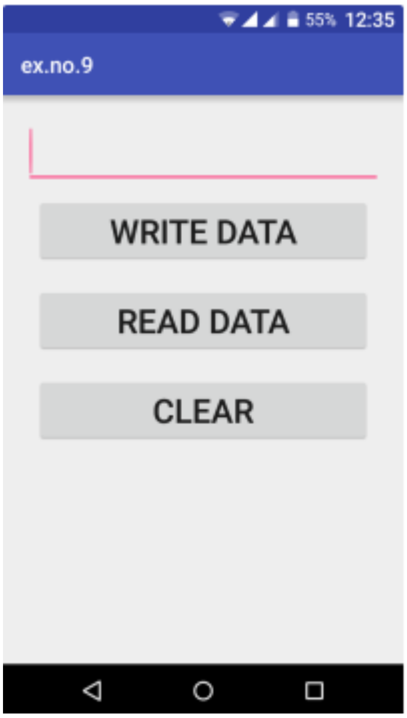
Output:

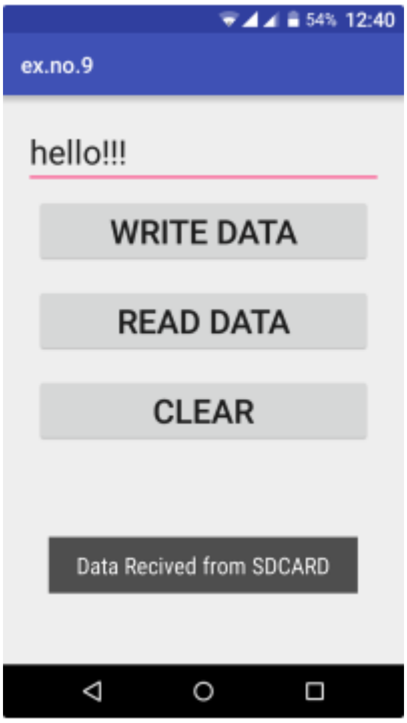


hello!!!

hello!!!

Data Written in SDCARD





hello!!!

Data Recived from SDCARD

Result:

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

Ex.No: 10

IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE

Aim:

To implement an application that creates an alert upon receiving a message.

Procedure:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project.
- Click on File -> New -> Activity -> Empty Activity.
- Type the Activity Name as SecondActivity and click Finish button. Thus Second Activity For the application is created.
- Click on app -> res -> layout -> activity_main.xml.
- Now click on Text as shown below. Then delete the code which is there and type the code as given below.
- Click on app -> java -> com.example.myapplication -> MainActivity.
- Now run the application to see the output.

Code:

MainActivity.java

```
package com.example.myapplication;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity
{
   Button notify;
   EditText e;
   @Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      notify= (Button) findViewById(R.id.button);
      e= (EditText) findViewById(R.id.editText);
      notify.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
         {
            Intent intent = new Intent(MainActivity.this, SecondActivity.class);
            PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);
            Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic_launcher).setContentIntent(pending).build();
            NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
            noti.flags |= Notification.FLAG_AUTO_CANCEL;
            manager.notify(0, noti);
         }
      });
```

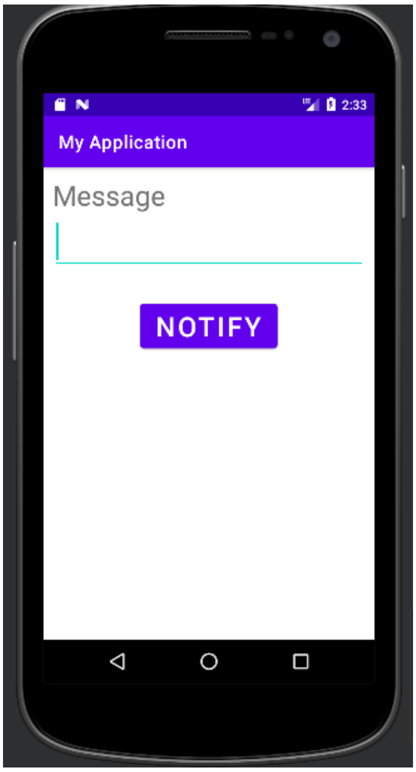
```
    }
}
```

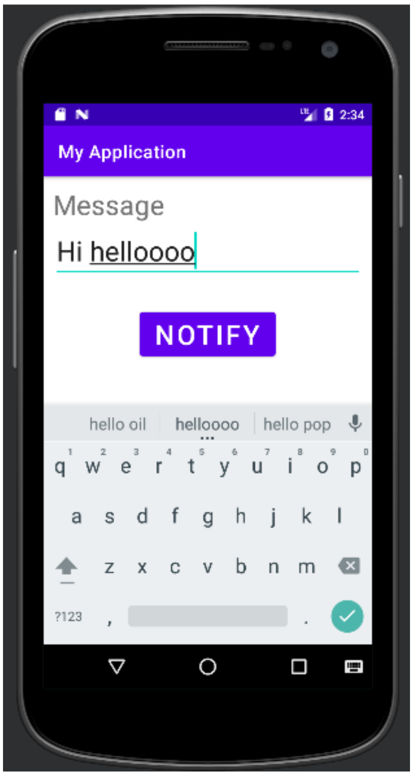
activity_main.xml

```
<?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:layout_margin="10dp"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message"
        android:textSize="30sp" />
    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:singleLine="true"
        android:textSize="30sp" />
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:layout_gravity="center"
        android:text="Notify"
        android:textSize="30sp"/>
```

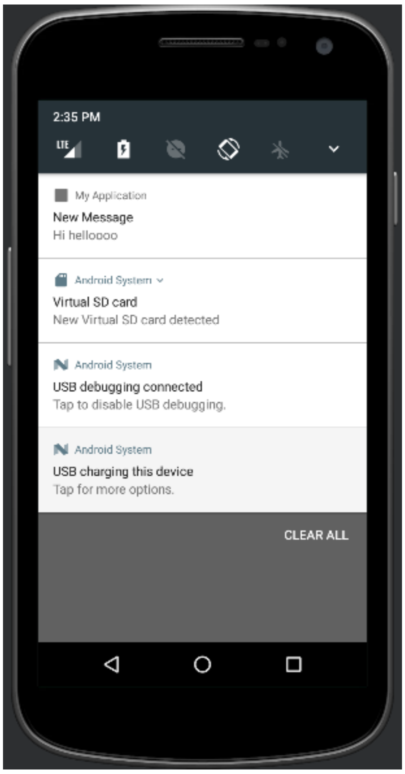
```
</LinearLayout>
```

Output:





Hi helloooo



My Application
New Message
Hi helloooo

Android System
Virtual SD card
New Virtual SD card detected

Android System
USB debugging connected
Tap to disable USB debugging.

Android System
USB charging this device
Tap for more options.

Result:

Thus Android Application that creates an alert upon receiving a message is developed and executed successfully.

Ex. No: 11

WRITE A MOBILE APPLICATION THAT CREATES ALARM CLOCK

Aim:

To write a mobile application that creates an alarm clock.

Procedure:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “My Application″ 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. It will take some time to build and load the project.
- Click on File -> New -> Activity -> Empty Activity.
- Type the Activity Name as AlarmReceiver and click Finish button. Thus Second Activity For the application is created.
- Click on app -> res -> layout -> activity_main.xml. Then delete the code which is there and type the code as given below.
- Click on app -> manifests -> AndroidManifest.xml
- Now change the activity tag to receiver tag in the AndroidManifest.xml file as shown below
- Click on app -> java -> com.example.myapplication -> MainActivity. Then delete the code which is there and type the code as given below.
- Click on app -> java -> com.example.myapplication -> AlarmReceiver. Then delete the code which is there and type the code as given below.
- Now run the application to see the output.

Code:

MainActivity.java

```
package com.example.myapplication;

import android.app.AlarmManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
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 = (TimePicker) 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.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
            calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
            Intent intent = new Intent(this, AlarmReceiver.class);
            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);
            }
            alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time, 10000,
pendingIntent);
        }
        else
        {
            alarmManager.cancel(pendingIntent);
            Toast.makeText(MainActivity.this, "ALARM OFF",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

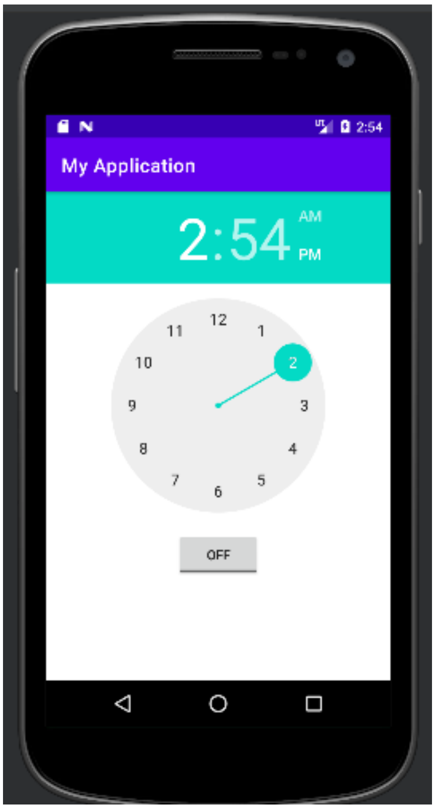
AndroidManifest.xml

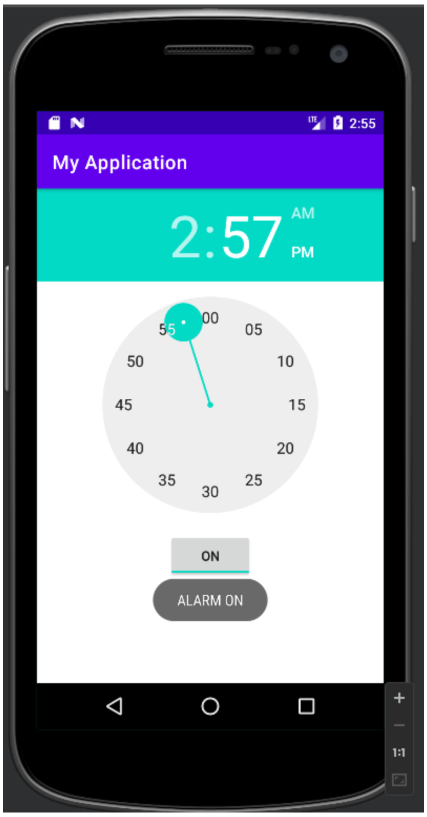
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.myapplication">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <receiver android:name=".AlarmReceiver" >
        </receiver>
    </application>
</manifest>
```

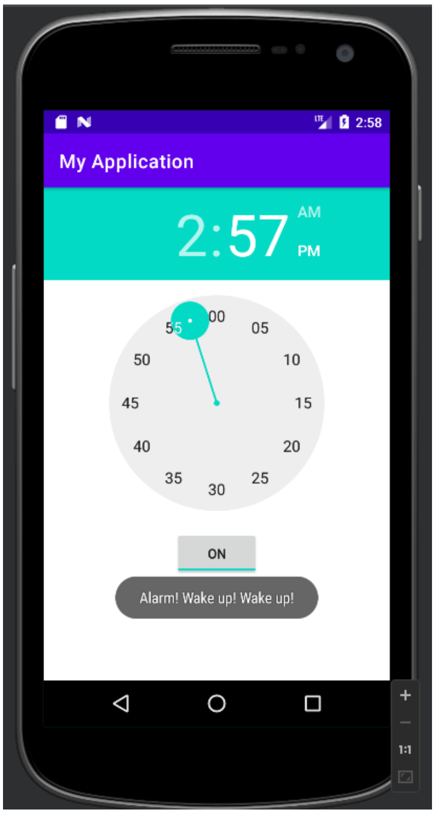
AlarmReceiver.java

```
package com.example.myapplication;
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.widget.Toast;
public class AlarmReceiver extends BroadcastReceiver
{
    @Override
    public void onReceive(Context context, Intent intent)
    {
        Toast.makeText(context, "Alarm! Wake up! Wake up!",
Toast.LENGTH_LONG).show();
        Uri alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        if (alarmUri == null)
        {
            alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
        }
        Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
        ringtone.play();
    }
}
```

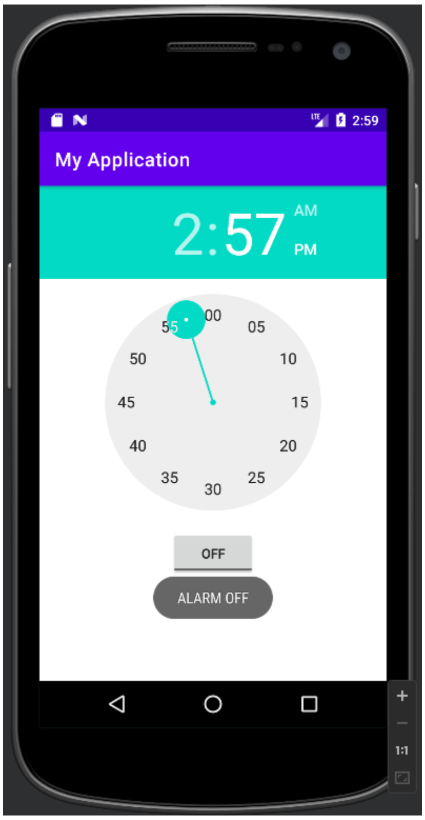
Output:







Alarm! Wake up! Wake up!



ALARM OFF

Result:

Thus, Android Application that creates Alarm Clock is developed and executed successfully.

Ex. No: 12

WRITE A MOBILE APPLICATION TO SEND EMAIL

Aim:

To write a mobile application to send email.

Procedure:

- Open Android Studio and then click on File -> New -> New project.
- Then type the Application name as “ex.no.12″ 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. It will take some time to build and load the project.
- Click on File -> New -> Activity -> Empty Activity.
- Click on app -> res -> layout -> activity_main.xml. Then delete the code which is there and type the code as given below.
- Click on app -> manifests -> AndroidManifest.xml
- Now change the activity tag to receiver tag in the AndroidManifest.xml file as shown below
- Now run the application to see the output.

Code:

MainActivity.java

```
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // define objects for edit text and button
    Button button;
    EditText sendto, subject, body;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Getting instance of edittext and button
    sendto = findViewById(R.id.editText1);
    subject = findViewById(R.id.editText2);
    body = findViewById(R.id.editText3);
    button = findViewById(R.id.button);

    // attach setOnClickListener to button with Intent object define in it
    button.setOnClickListener(view -> {
        String emailsend = sendto.getText().toString();
        String emailsubject = subject.getText().toString();
        String emailbody = body.getText().toString();

        // define Intent object with action attribute as ACTION_SEND
        Intent intent = new Intent(Intent.ACTION_SEND);

        // add three fields to intent using putExtra function
        intent.putExtra(Intent.EXTRA_EMAIL, new String[]{emailsend});
        intent.putExtra(Intent.EXTRA_SUBJECT, emailsubject);
        intent.putExtra(Intent.EXTRA_TEXT, emailbody);

        // set type of intent
        intent.setType("message/rfc822");

        // startActivity with intent with chooser as Email client using
createChooser function
        startActivity(Intent.createChooser(intent, "Choose an Email client :"));
    });
    }
}
```

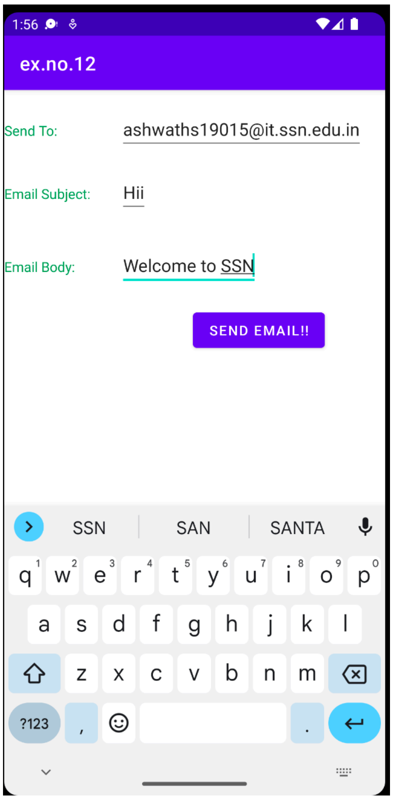
activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Relative Layout -->
<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">

    <!-- Edit text for email id -->
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_alignParentRight="true"
        android:layout_marginTop="18dp"
        android:layout_marginRight="22dp" />

    <!-- Edit text for email subject -->
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText1"
        android:layout_alignLeft="@+id/editText1"
        android:layout_marginTop="20dp" />

    <!-- Edit text for email body -->
    <EditText
```

```
        android:id="@+id/editText3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText2"
        android:layout_alignLeft="@+id/editText2"
        android:layout_marginTop="30dp" />

<!-- text Views for label -->
<TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText1"
        android:layout_alignBottom="@+id/editText1"
        android:layout_alignParentLeft="true"
        android:text="Send To:"
        android:textColor="#0F9D58" />

<TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBaseline="@+id/editText2"
        android:layout_alignBottom="@+id/editText2"
        android:layout_alignParentLeft="true"
        android:text="Email Subject:"
        android:textColor="#0F9D58" />

<TextView
        android:id="@+id/textView3"
```

```
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBaseline="@+id/editText3"
            android:layout_alignBottom="@+id/editText3"
            android:text="Email Body:"
            android:textColor="#0F9D58" />

        <!-- Button to send email -->
        <Button
            android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/editText3"
            android:layout_alignLeft="@+id/editText3"
            android:layout_marginLeft="76dp"
            android:layout_marginTop="20dp"
            android:text="Send email!!" />
    </RelativeLayout>
```

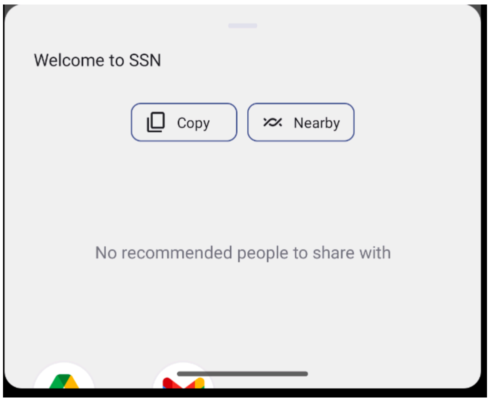
Output:



Send To: ashwaths19015@it.ssn.edu.in

Email Subject: Hii

Email Body: Welcome to SSN



Welcome to SSN

No recommended people to share with

Result:

The android studio application for sending email is developed and executed successfully .