```java
package com.example.mysmsapplication;

//<?xml version="1.0" encoding="utf-8"?>
//        <manifest xmlns:android="http://schemas.android.com/apk/res/android"
//        xmlns:tools="http://schemas.android.com/tools">
//        <uses-permission android:name="android.permission.SEND_SMS"/>
//        <uses-feature android:name="android.hardware.telephony"
//        android:required="false"/>
import androidx.appcompat.app.AppCompatActivity;
import android.Manifest;
import android.app.PendingIntent;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private static final int MY_PERMISSIONS_REQUEST_SEND_SMS=123;
    private EditText recipientedtxt;
    private EditText messageedtxt;
    private Button sendBtn;
    private PendingIntent sentpi;
    private PendingIntent deliverypi;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

       recipientedtxt=findViewById(R.id.ed1);
       messageedtxt=findViewById(R.id.ed2);
       sendBtn=findViewById(R.id.btn);

       sentpi=PendingIntent.getBroadcast(this,0,new Intent("SMS_SENT"),
PendingIntent.FLAG_IMMUTABLE);
       deliverypi=PendingIntent.getBroadcast(this,1,new Intent("SMS_DELIVERED"
),PendingIntent.FLAG_IMMUTABLE);

       sendBtn.setOnClickListener(view -> {
           String recipient=recipientedtxt.getText().toString();
           String message=messageedtxt.getText().toString();

           if(recipient.isEmpty()||message.isEmpty()){
               Toast.makeText(MainActivity.this,"Receipent and message fields
cannot be empty",Toast.LENGTH_LONG).show();
               return;
           }

           if(checkSelfPermission(Manifest.permission.SEND_SMS)!=
PackageManager.PERMISSION_GRANTED){
               requestPermissions(new String[]{ Manifest.permission.SEND_SMS},
MY_PERMISSIONS_REQUEST_SEND_SMS);
           }
           else{
               sendSMS(recipient,message);
           }
```

```
55          });
56      }
57
58      public void sendSMS(String recipient,String message){
59          try{
60              SmsManager smsManager=SmsManager.getDefault();
61              smsManager.sendTextMessage(recipient,null,message,sentpi,
    deliverypi);
62              Toast.makeText(this,"Message sent successfully",Toast.LENGTH_LONG
    ).show();
63          }
64          catch (Exception e){
65              Toast.makeText(this,"Failed to send sms",Toast.LENGTH_LONG).show
    ();
66              e.printStackTrace();
67          }
68      }
69
70      @Override
71      public void onRequestPermissionsResult(int requestCode,String[]
    permissions,int[] grantResults){
72          super.onRequestPermissionsResult(requestCode,permissions,grantResults
    );
73
74          if(requestCode==MY_PERMISSIONS_REQUEST_SEND_SMS){
75              if(grantResults.length>0&& grantResults[0]==PackageManager.
    PERMISSION_GRANTED){
76                  String recipient=recipientedtxt.getText().toString();
77                  String message=messageedtxt.getText().toString();
78                  sendSMS(recipient,message);
79              }
80              else{
81                  Toast.makeText(this,"SMS Permission denied cannot send sms",
    Toast.LENGTH_LONG).show();
82              }
83          }
84      }
85  }
```