**Ex. No. : 8a Date: 28/04/2025**

**Register No.: 221701050 Name: D Sathiya Sri**



**Send SMS**

**Aim**

Develop an application to send SMS.

***Procedure:***

1. Create a New Android Project

* Open Android Studio and create a new project with Empty Activity.
* Choose Kotlin as the programming language.

1. Add Permissions in AndroidManifest.xml

* Add the SEND\_SMS permission in the AndroidManifest.xml.

1. Design the Layout (activity\_main.xml)

* Add EditText fields for entering the phone number and message.
* Add a Button to trigger the SMS sending action.

1. Request Runtime Permissions (For Android 6.0 and Above)

* Request the SEND\_SMS permission at runtime.

1. Set Up Button Click Listener

* Set an OnClickListener on the Send SMS button to gather the phone number and message.

1. Send SMS Using SmsManager

* Use SmsManager to send the SMS to the specified phone number.

1. Handle Success and Error Responses

* Optionally, display Toast messages to indicate whether the SMS was sent successfully or if there was an error.

1. Run and Test the Application

* Build and run the app on a physical device to test the SMS sending functionality.

***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.exp\_8a">

<uses-permission android:name="android.permission.SEND\_SMS"/>

<uses-permission android:name="android.permission.READ\_PHONE\_STATE" />

<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.Exp8a"

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>

</application>

</manifest>

***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:padding="20dp"

android:orientation="vertical">

<EditText

android:id="@+id/editTextPhone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Phone Number"

android:inputType="phone" />

<EditText

android:id="@+id/editTextMessage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Message"

android:inputType="textMultiLine"

android:minLines="3"

android:gravity="top" />

<Button

android:id="@+id/buttonSend"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Send SMS" />

</LinearLayout>

***MainActivity.kt***

package com.example.exp\_8a

import android.Manifest

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

import androidx.appcompat.app.AppCompatActivity

import androidx.core.app.ActivityCompat

import androidx.core.content.ContextCompat

class MainActivity : AppCompatActivity() {

private lateinit var editTextPhone: EditText

private lateinit var editTextMessage: EditText

private lateinit var buttonSend: Button

private val SMS\_PERMISSION\_CODE = 100

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

editTextPhone = findViewById(R.id.editTextPhone)

editTextMessage = findViewById(R.id.editTextMessage)

buttonSend = findViewById(R.id.buttonSend)

buttonSend.setOnClickListener {

if (checkPermission()) {

sendSMS()

} else {

requestPermission()

}

}

}

private fun checkPermission(): Boolean {

val result = ContextCompat.checkSelfPermission(this, Manifest.permission.SEND\_SMS)

return result == PackageManager.PERMISSION\_GRANTED

}

private fun requestPermission() {

ActivityCompat.requestPermissions(

this,

arrayOf(Manifest.permission.SEND\_SMS),

SMS\_PERMISSION\_CODE

)

}

private fun sendSMS() {

val phoneNumber = editTextPhone.text.toString()

val message = editTextMessage.text.toString()

if (phoneNumber.isNotEmpty() && message.isNotEmpty()) {

try {

val smsManager = SmsManager.getDefault()

smsManager.sendTextMessage(phoneNumber, null, message, null, null)

Toast.makeText(this, "SMS Sent Successfully!", Toast.LENGTH\_SHORT).show()

} catch (e: Exception) {

Toast.makeText(this, "Failed to send SMS: ${e.message}", Toast.LENGTH\_LONG).show()

}

} else {

Toast.makeText(this, "Please enter phone number and message", Toast.LENGTH\_SHORT).show()

}

}

// Handle permission result

override fun onRequestPermissionsResult(

requestCode: Int, permissions: Array<out String>, grantResults: IntArray

) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults)

if (requestCode == SMS\_PERMISSION\_CODE) {

if (grantResults.isNotEmpty() && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

sendSMS()

} else {

Toast.makeText(this, "Permission denied", Toast.LENGTH\_SHORT).show()

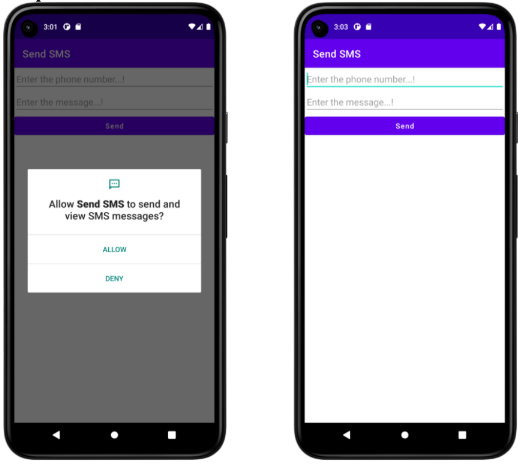
}

}

}

}

***Output***

******

**Result:**

| The app successfully sends SMS messages to a specified phone number by using the SmsManager class, with proper permissions and runtime handling. |
| --- |