ASSIGNMENT NO.9

GAURAV_BODKHE_2124UCEM1041

1. Introduction

An SMS sending program is a software application designed to enable users to send text messages directly from their mobile devices. It serves as a simple yet essential tool for communication, allowing users to input a phone number and message, and send the SMS with the click of a button. In this project, we will create an app that allows users to enter a recipient's phone number and compose a message, then click a "Send" button to deliver the SMS. The program will incorporate a user-friendly interface with fields for the phone number and message, along with a button to trigger the sending of the SMS. This functionality provides users with a convenient way to send text messages, utilizing the built-in telephony services of the Android platform.

2. Tools & Technologies Used

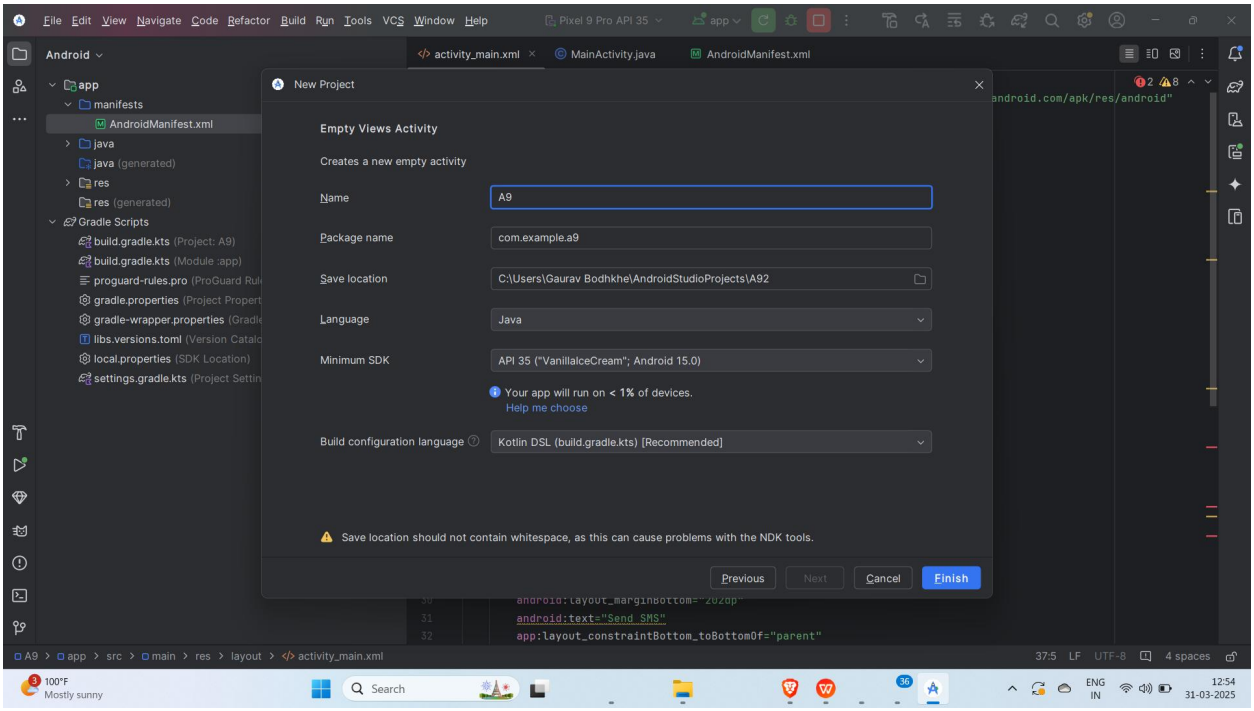
- Android Studio
- Java
- Emulator

3. Procedure & Steps

Step 1: Create a New Project

- Open Android Studio and create a new project.
- Choose an Empty Views Activity template.
- Set the project name and package name of your Application
- Select the programming language (Java).

Screenshot:

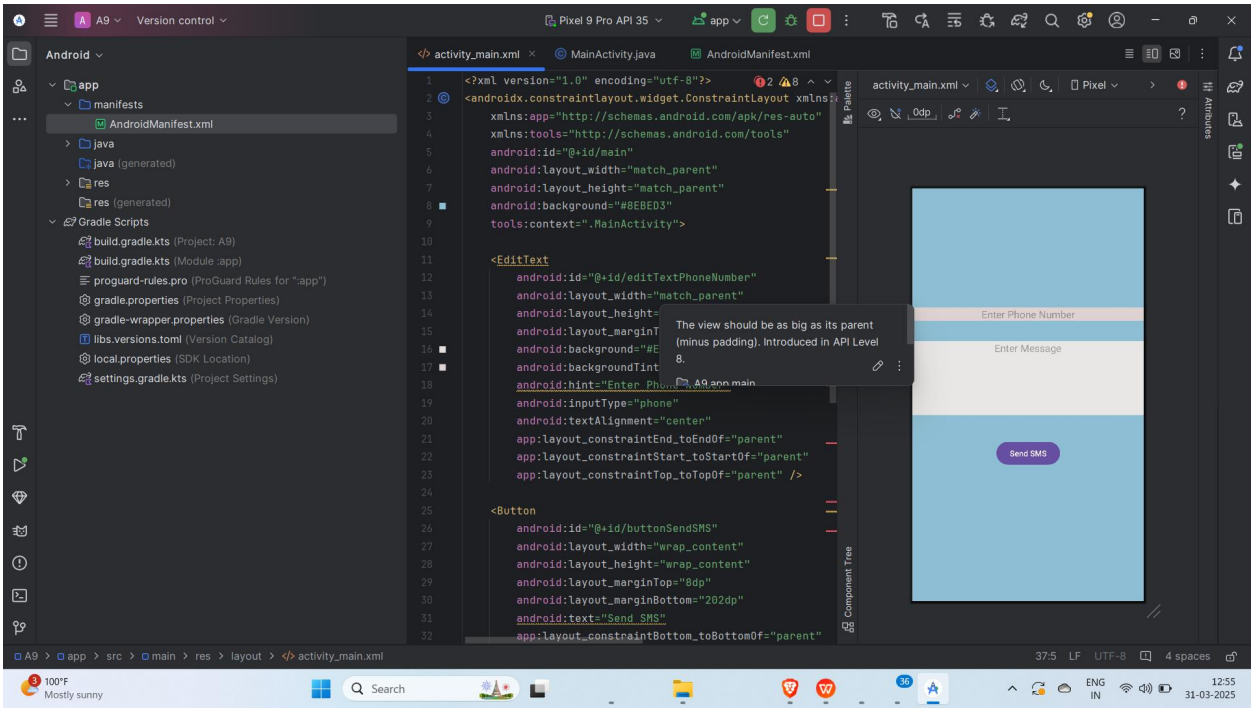


Step 2: Designing the UI

- Open *activity_main.xml* and design the layout using XML.

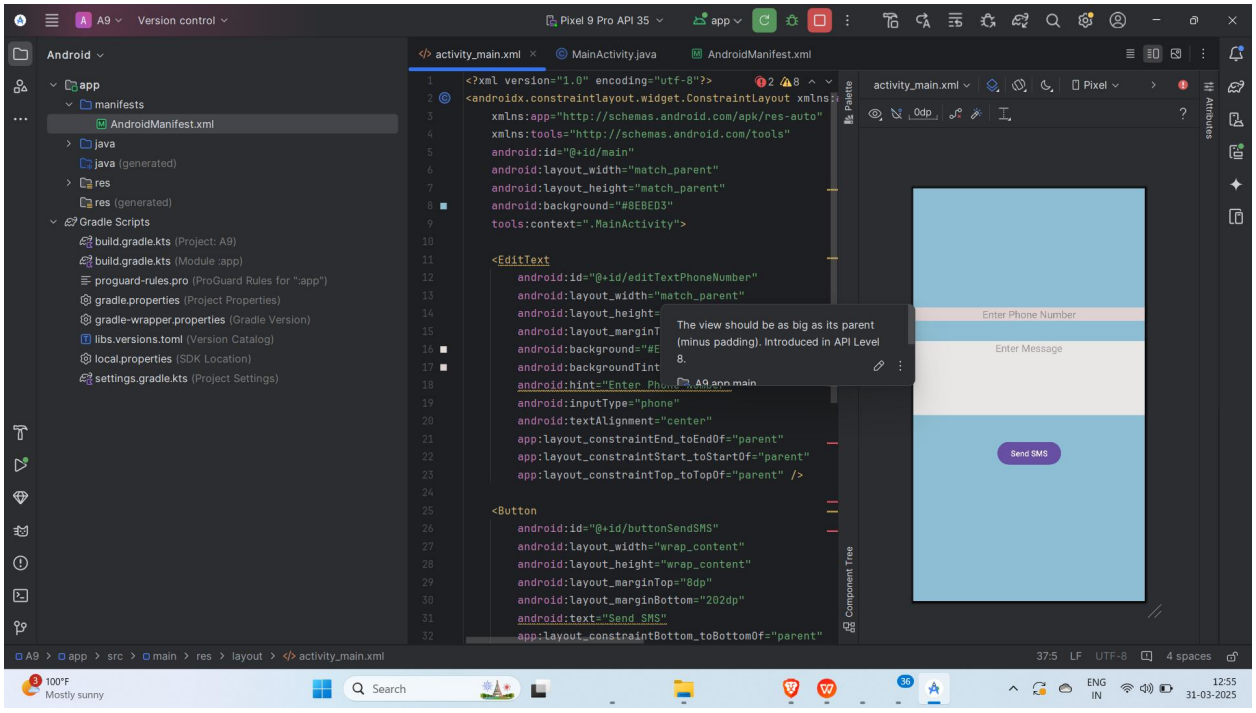
Add UI components such as EditText for *phone number and message*, and a Button to *send the SMS*

Screenshot:



Step 3: Writing the Code

- Open *Activity_main.xml*
- Implement functionality such as *Button,Textview.*
- Use necessary Android components like *Buttons,textview* etc
- **Screenshot:**



- **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/main"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    android:background="#8EBED3"

    tools:context=".MainActivity">


    <EditText
```

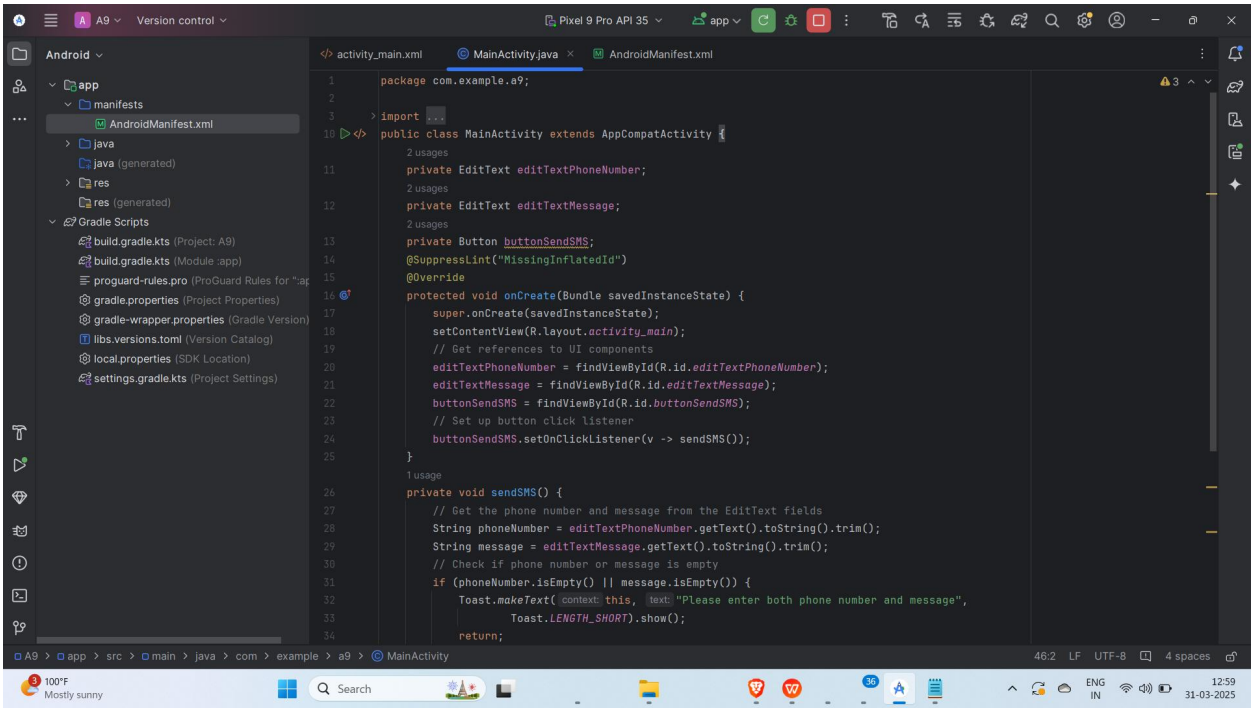
```
    android:id="@+id/editTextPhoneNumber"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="211dp"
    android:background="#E7DDDD"
    android:backgroundTint="#DED2D2"
    android:hint="Enter Phone Number"
    android:inputType="phone"
    android:textAlignment="center"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Button
    android:id="@+id/buttonSendSMS"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="202dp"
    android:text="Send SMS"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/editTextMessage" />
```

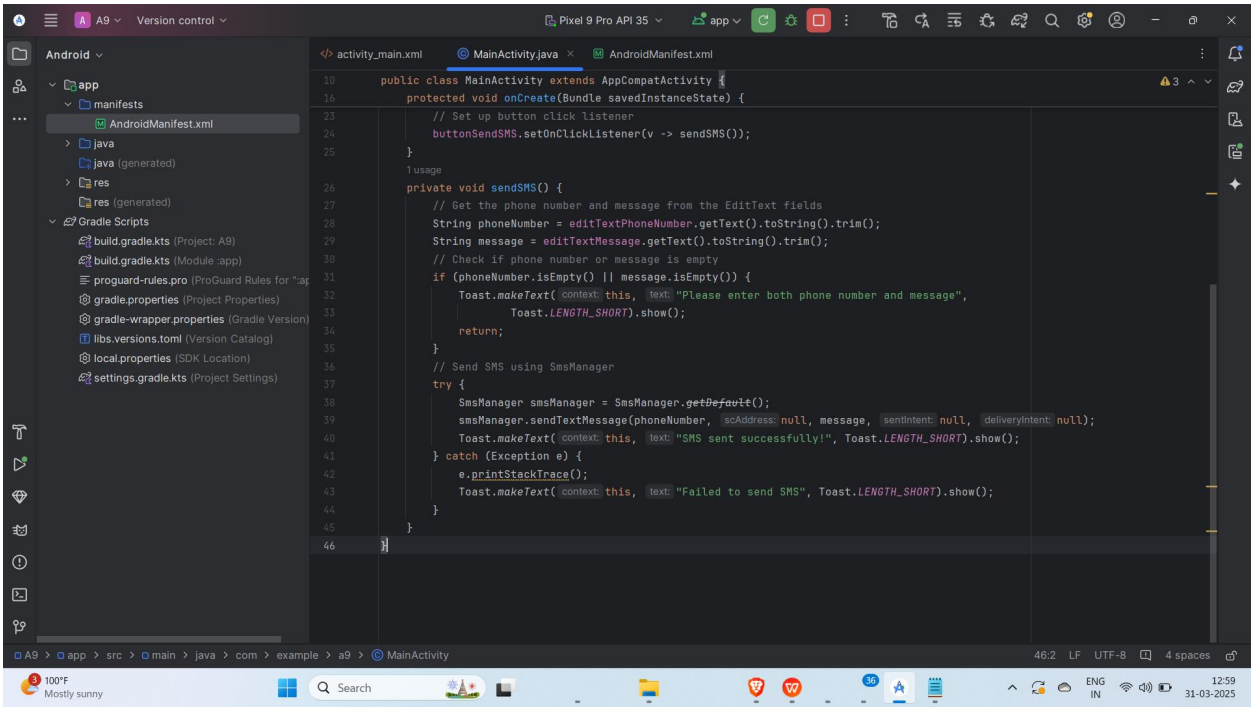
```
    <EditText
        android:id="@+id/editTextMessage"
        android:layout_width="411dp"
        android:layout_height="131dp"
        android:background="#E9E6E6"
        android:ems="10"
        android:gravity="start|top"
        android:hint="Enter Message"
        android:inputType="textMultiLine"
        android:textAlignment="center"
        app:layout_constraintBottom_toTopOf="@+id/buttonSendSMS"
        app:layout_constraintTop_toBottomOf="@+id/editTextPhoneNumber"
        tools:layout_editor_absoluteX="0dp" />

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

Step 4: Writing the Backend Code (java)

- Open MainActivity.java
- Implement Futionalities like When we give the Input of Mobile number As well as the message it will be get Delivered to that phone Number when Click on **“Send SMS”** .

Screenshot:





Code(java):

package com.example.a9;

```
import android.annotation.SuppressLint;
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;
public class MainActivity extends AppCompatActivity {
    private EditText editTextPhoneNumber;
    private EditText editTextMessage;
    private Button buttonSendSMS;
    @SuppressLint("MissingInflatedId")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Get references to UI components
        editTextPhoneNumber = findViewById(R.id.editTextPhoneNumber);
        editTextMessage = findViewById(R.id.editTextMessage);
        buttonSendSMS = findViewById(R.id.buttonSendSMS);
        // Set up button click listener
        buttonSendSMS.setOnClickListener(v -> sendSMS());
    }
```

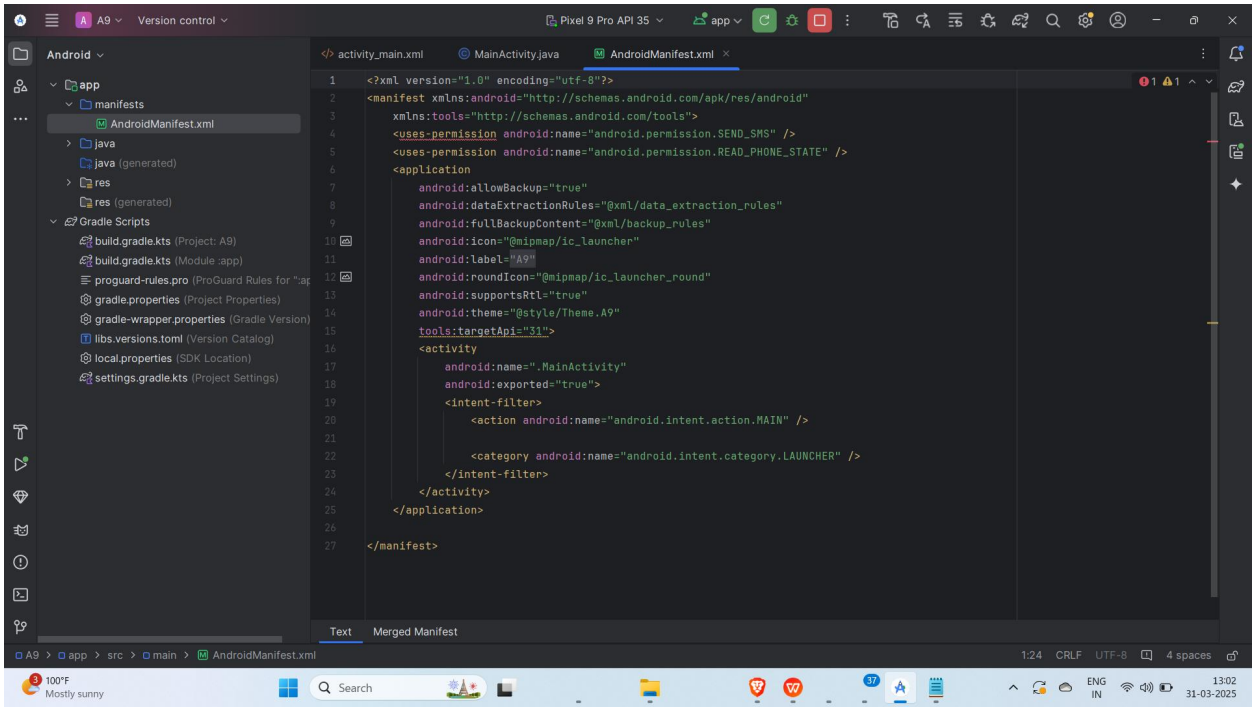
```
    private void sendSMS() {
        // Get the phone number and message from the EditText fields
        String phoneNumber = editTextPhoneNumber.getText().toString().trim();
        String message = editTextMessage.getText().toString().trim();
        // Check if phone number or message is empty
        if (phoneNumber.isEmpty() || message.isEmpty()) {
            Toast.makeText(this, "Please enter both phone number and message",
                    Toast.LENGTH_SHORT).show();
            return;
        }
        // Send SMS using SmsManager
        try {
            SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(phoneNumber, null, message, null, null);
            Toast.makeText(this, "SMS sent successfully!", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
            Toast.makeText(this, "Failed to send SMS", Toast.LENGTH_SHORT).show();
        }
    }
}
```

Step 5: Taking Permission To open the Camera

- Open AndroidManifest.xml

- Write the code for the Giving Permission to Sending the SMS

Screenshot:



Code:

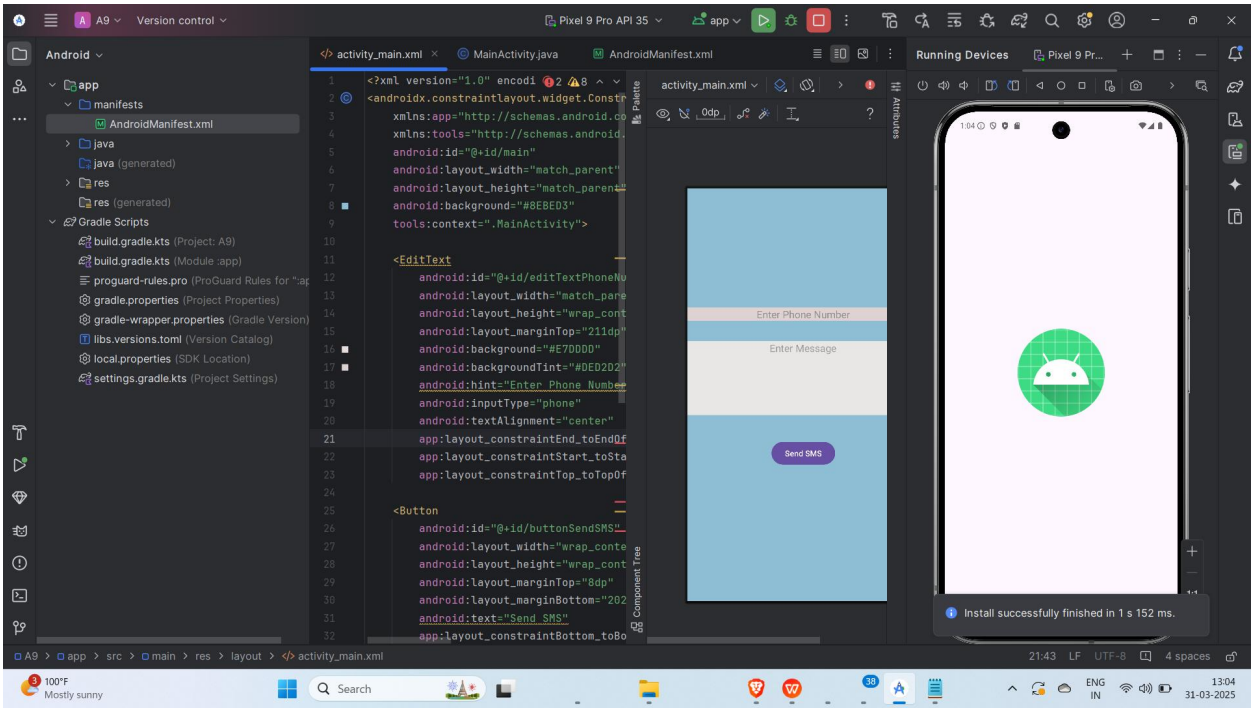
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.SEND_SMS" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <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.A9"
        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>
```

Step 6: Running the Application on Emulator

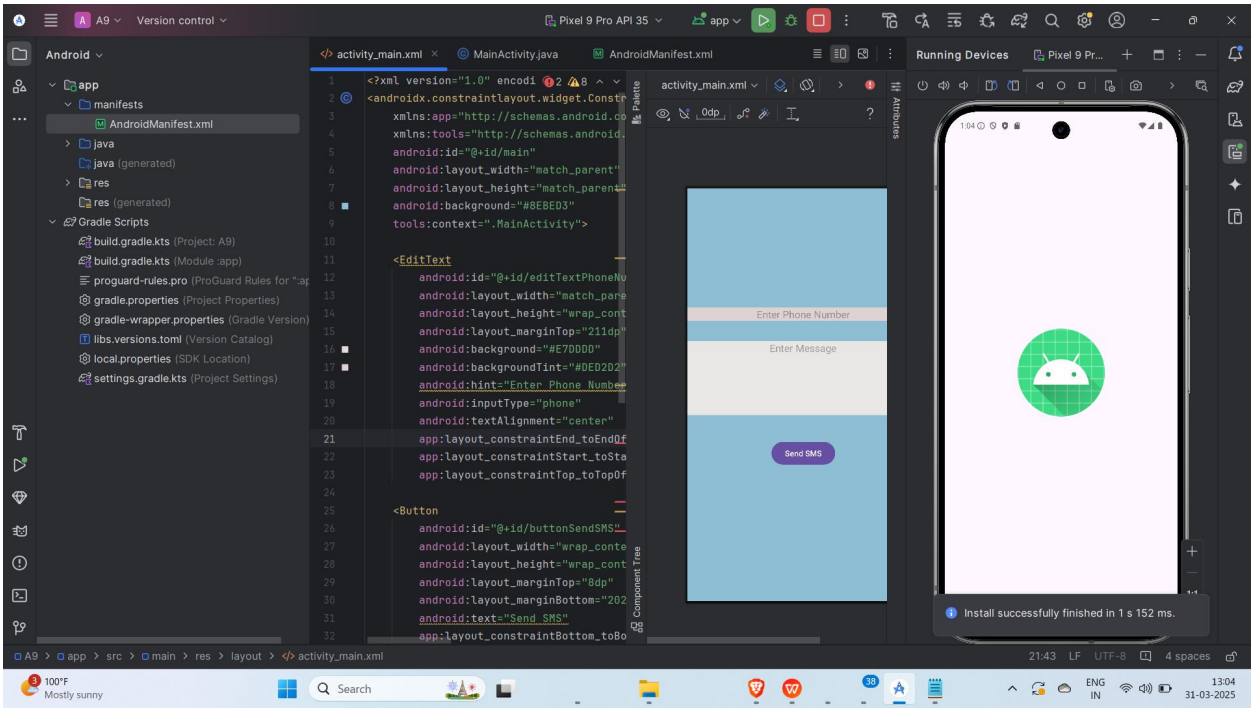
- Click on the **Run** button in Android Studio.
- Select the emulator and launch the app.

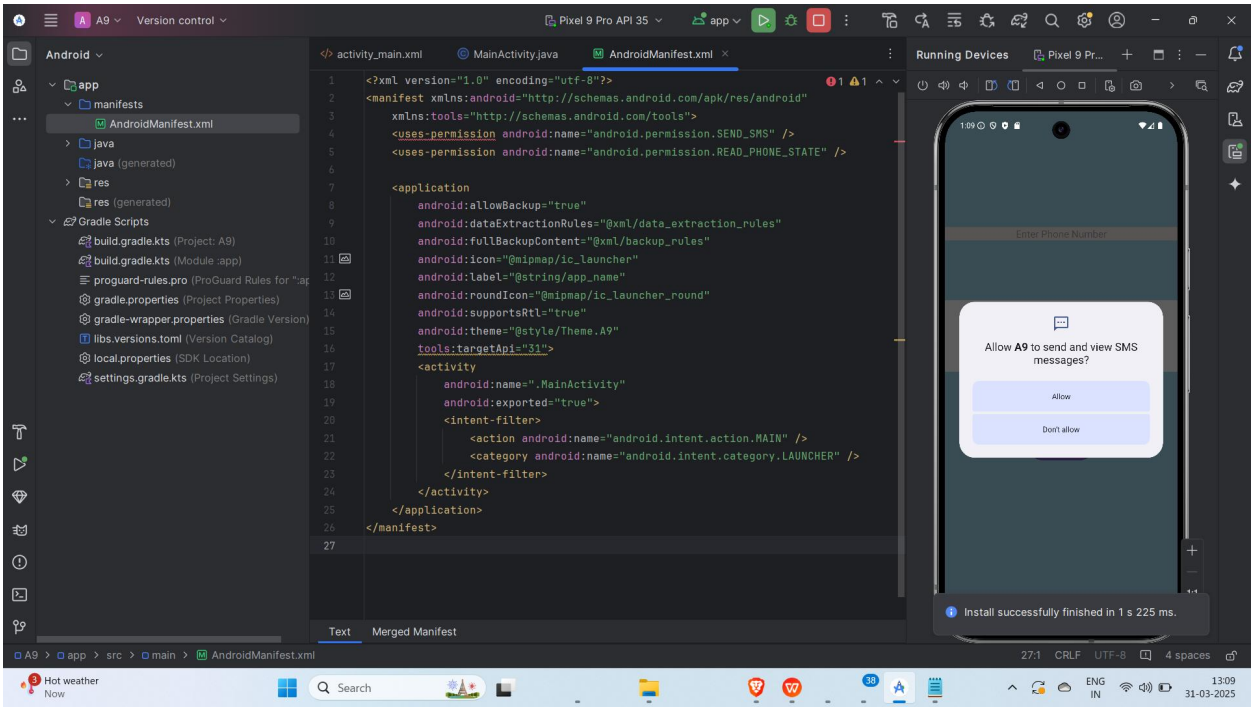
  Screenshot:

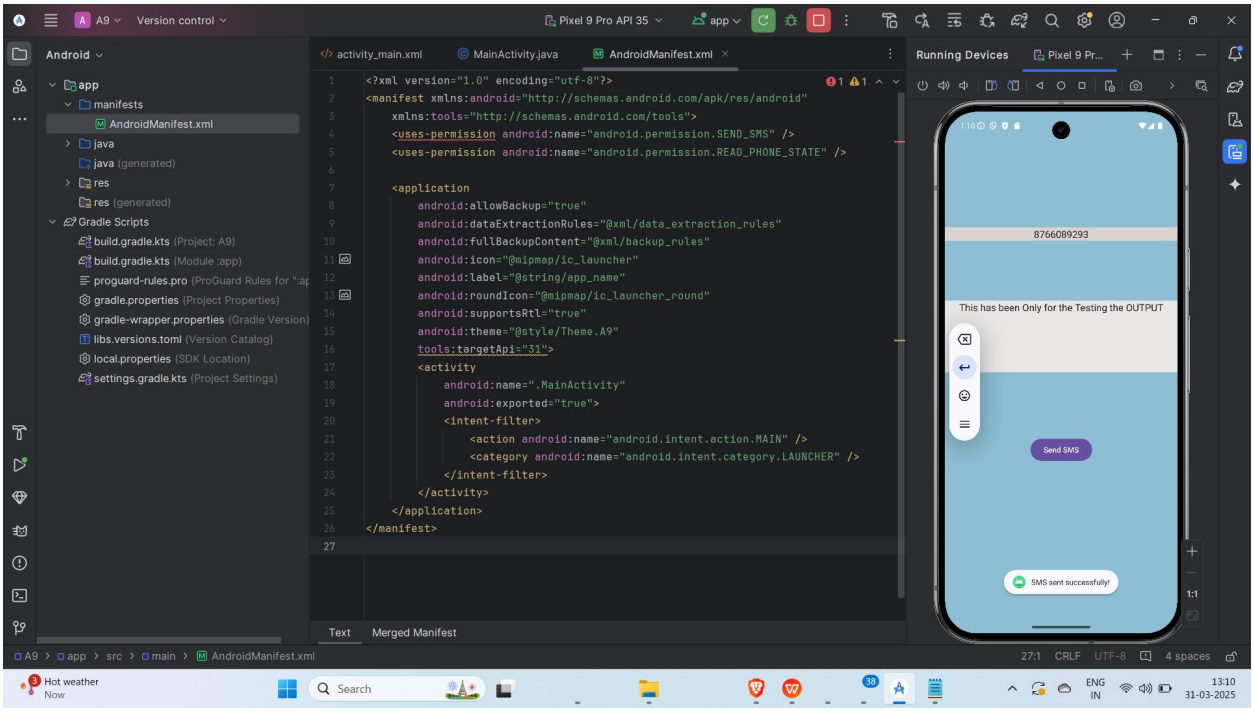


Step 6: Testing & Output

- Test different functionalities of the app.
- Giving Input of Mobile Number and SMS Message.
- **Screenshot:**


activity_main.xml
MainActivity.java
AndroidManifest.xml
<?xml version="1.0" encodi
<androidx.constraintlayout.widget.Constr
xmlns:app="http://schemas.android.co
xmlns:tools="http://schemas.android.
android:id="@+id/main"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="#8EBED3"
tools:context=".MainActivity">
<EditText
android:id="@+id/editTextPhoneNu
android:layout_width="match_pare
android:layout_height="wrap_cont
android:layout_marginTop="211dp"
android:background="#E7DDDD"
android:backgroundTint="#DED2D2"
android:hint="Enter Phone Number
android:inputType="phone"
android:textAlignment="center"
app:layout_constraintEnd_toEndOf
app:layout_constraintStart_toSta
app:layout_constraintTop_toTopOf
<Button
android:id="@+id/buttonSendSMS"
android:layout_width="wrap_conte
android:layout_height="wrap_cont
android:layout_marginTop="8dp"
android:layout_marginBottom="202
android:text="Send SMS"
app:layout_constraintBottom_toBo
Enter Phone Number
Enter Message
Send SMS
Install successfully finished in 1 s 152 ms.


activity_main.xml
MainActivity.java
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">
<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.A9"
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>
Text
Merged Manifest
Allow A9 to send and view SMS messages?
Allow
Don't allow
Install successfully finished in 1 s 225 ms.



4. Conclusion

I successfully developed a simple Android app that allows users to send SMS messages. Through this project, I gained hands-on experience in implementing SMS functionality using SmsManager in Android. I designed an intuitive interface where users can enter a phone number and message, then send it with a single click.

One challenge I encountered was handling SMS permissions, especially for newer Android versions that require runtime permission requests. However, I overcame this by studying Android documentation and implementing the necessary permission-handling logic.

This project enhanced my Android development skills, particularly in working with system services like SMS. I also improved my understanding of Android permissions, UI design, and debugging. Overall, this experience has increased my confidence in building more functional and user-friendly Android application.