**MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY**

**(An Autonomous Institution – UGC, Govt.of India)**

**Recognizes under 2(f) and 12(B) of UGC ACT 1956**

**(Affiliated to JNTUH, Hyderabad, Approved by AICTE –Accredited by NBA & NAAC-“A” Grade-ISO 9001:2015 Certified)**

****

**DEPARTMENT OF AERONAUTICAL ENGINEERING**

**APP DEVELOPMENT 2**

**Guide: Mrs. L. SUSHMA**

**ASSOCIATE PROFESSOR**

**NAME: SHAIK NAWAZ**

**Roll No: 20N31A2134**

**NAME: TIRUMALA SAI NITHIN**

**Roll No: 20N31A2135**

**NAME: V. HIMABINDU**

**Roll No: 20N31A2136**

**ABSTRACT**

The purpose of this project is to create an Android app for sending SMS messages using Intents in Android Studio. The app will allow users to compose and send SMS messages to their contacts using their phone's default messaging app. The app will have a simple and user-friendly interface with options to select the recipient, enter the message body, and send the message. The app will also incorporate features such as message history, message templates, and the ability to save and edit drafts. The app will be developed using Java programming language and will utilize the Android SDK and Android Studio development tools. The end result will be a fully functional SMS messaging app that can be downloaded and used on any Android device.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SL.NO.** | **CONTENT** | **PAGE NUMBER** |
| **1** | **INTRODUCTION** |  |
| **2** | WHAT IS ANDROID? |  |
| **3** | WHY ANDROID? |  |
| **4** | FEATURES OF ANDROID |  |
| **5** | ANDROID APPLICATIONS |  |
| **6** | CATEGORIES OF ANDROID APPLICATIONS |  |
| **7** | **WHAT IS ANDROID STUDIO?** |  |
| **8** | FEATURES OF ANDROID STUDIO |  |
| **9** | INSTALLING ANDROID STUDIO |  |
| **10** | STARTING A NEW PROJECT |  |
| **11** | **PROCEDURE:**  Steps to create an app to send SMS using intent |  |
| **12** | **RESULT:** Output Screens |  |

**INTRODUCTION**

Mobile devices have become an integral part of our lives, and the ability to communicate with others through SMS is a fundamental feature of these devices. With the rise of smartphone technology, app development has become more accessible, allowing developers to create applications that enhance our daily lives. In this project, we will be exploring how to develop an SMS sending app using intent in Android Studio.

Using Android Studio, we will create an application that allows users to send SMS messages to their contacts. We will leverage the intent feature of Android to facilitate communication between different components of our app, such as the UI, backend, and system applications. By utilizing the power of intents, we can create a seamless user experience that enhances the functionality of the app.

Our app will provide users with a simple and intuitive interface, allowing them to select a contact from their address book and compose an SMS message. We will also implement features such as message history, message templates, and message scheduling to enhance the app's usability.

Throughout this project, we will explore the fundamentals of Android app development, including UI design, intent handling, data storage, and system integration. Our goal is to provide users with a reliable and efficient SMS sending app that utilizes the latest features of Android Studio.

**WHAT IS ANDROID?**



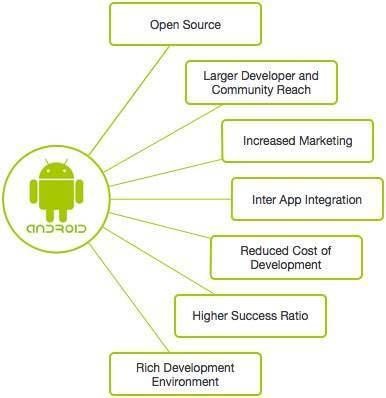
Android is an open source and Linux-based **Operating System** for mobile devices such as smartphones and tablet computers. Android was developed by the *Open Handset Alliance*, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

Android programming is based on Java programming language so if you have basic understanding on Java programming then it will be a fun to learn Android application development.

Why android?



**FEATURES OF ANDROID**

Android is a powerful operating system competing with Apple 4GS and supports great features. Few of them are listed below −

|  |  |
| --- | --- |
| **Sr.No.** | **Feature & Description** |
| 1 | **Beautiful UI**  Android OS basic screen provides a beautiful and intuitive user interface. |
| 2 | **Connectivity**  GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX. |
| 3 | **Storage** |

|  |  |
| --- | --- |
|  | SQLite, a lightweight relational database, is used for data storage purposes. |
| 4 | **Media support**  H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, WAV, JPEG, PNG, GIF, and BMP. |
| 5 | **Messaging**  SMS and MMS |
| 6 | **Web browser**  Based on the open-source Web Kit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3. |
| 7 | **Multi-touch**  Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero. |
| 8 | **Multi-tasking**  User can jump from one task to another and same time various application can run simultaneously. |
| 9 | **Resizable widgets**  Widgets are resizable, so users can expand them to show more content or shrink them to save space. |
| 10 | **Multi-Language**  Supports single direction and bi-directional text. |
| 11 | **GCM**  Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution. |

|  |  |
| --- | --- |
| 12 | **Wi-Fi Direct**  A technology that lets apps discover and pair directly, over a high- bandwidth peer-to-peer connection. |
| 13 | **Android Beam**  A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together. |

**ANDROID APPLICATIONS**

Android applications are usually developed in the Java language using the Android Software Development Kit.

Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**, **Opera Mobile Store**, **F-droid** and the **Amazon Appstore**.

Android powers hundreds of millions of mobile devices in more than 190 countries around the world. It's the largest installed base of any mobile platform and growing fast. Every day more than 1 million new Android devices are activated worldwide.

This tutorial has been written with an aim to teach you how to develop and package Android application. We will start from environment setup for Android application programming and then drill down to look into various aspects of Android applications.

**Categories of android applications**

In Android applications, categories are used to classify and organize apps in app stores and on devices. The following are some common categories used for Android applications:

**1. Books & Reference:** Apps that provide access to books, dictionaries, and other reference materials.

**2. Business:** Apps for managing finances, scheduling appointments, and other business-related tasks.

**3. Communication:** Apps for messaging, calling, and other forms of communication.

**4. Education:** Apps for learning, teaching, and studying.

**5. Entertainment:** Apps for watching videos, playing games, listening to music, and other forms of entertainment.

**6. Finance:** Apps for managing money, including budgeting, investing, and banking.

**7. Health & Fitness:** Apps for tracking fitness, nutrition, and other health-related activities.

**8. Lifestyle:** Apps for managing personal activities, including travel, shopping, and cooking.

**9. News & Magazines:** Apps for reading news articles and magazines.

**10. Productivity:** Apps for increasing productivity, including task management and note-taking.

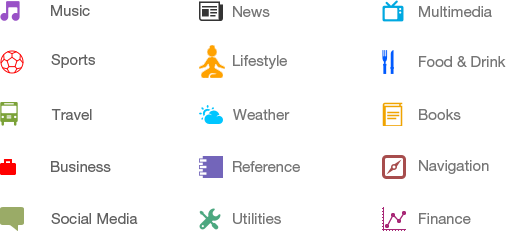
**11. Social:** Apps for connecting with friends and family, including social media and messaging apps.

**12. Sports:** Apps for tracking and watching sports events.

**13. Tools:** Apps that provide various tools, including calculators, weather forecasts, and system utilities.

**14. Travel & Local:** Apps for finding and booking travel accommodations and activities.

**15. Weather:** Apps for providing weather forecasts and alerts.



**What is android studio?**

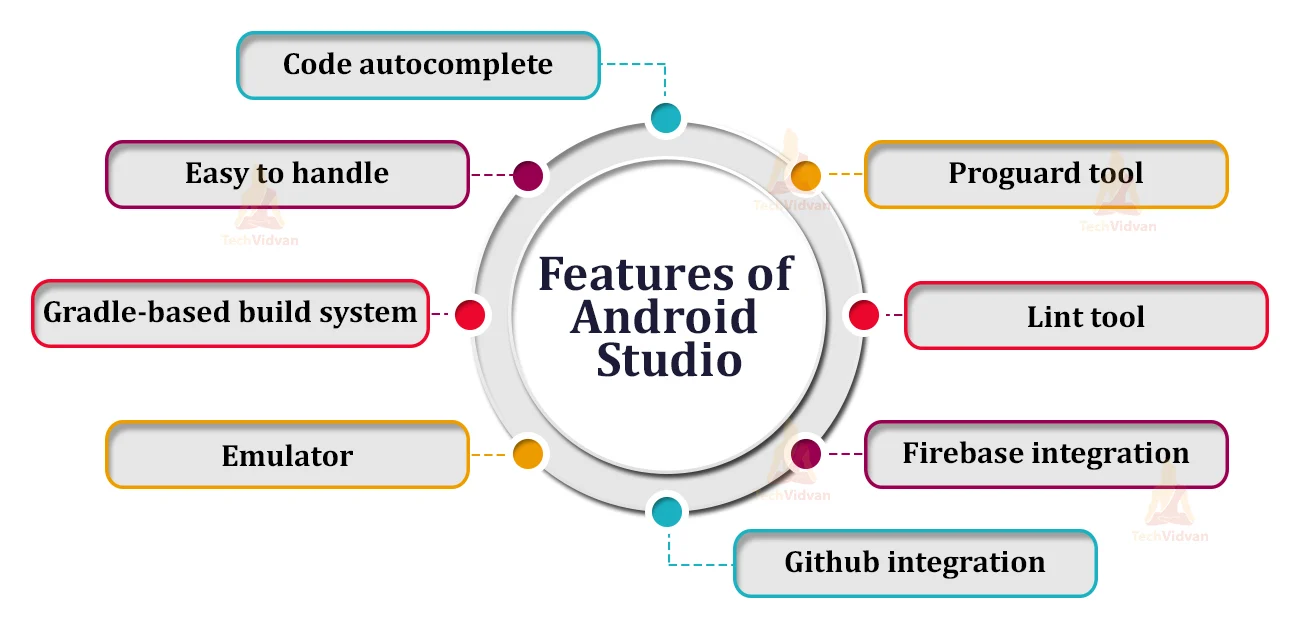
****

**Android Studio** is the official  Integrated Development Unit (IDE) for Google’s Android Operating Systems, built on JetBrains’IntelliJ IDEA software and designed specifically for  Android Development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the  Eclipse Android Development Tools(E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O  conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.

On May 7, 2019, Kotlin  replaced Java  as Google's preferred language for Android app development. Java is still supported, as is C++.

**Features of android studio**

****

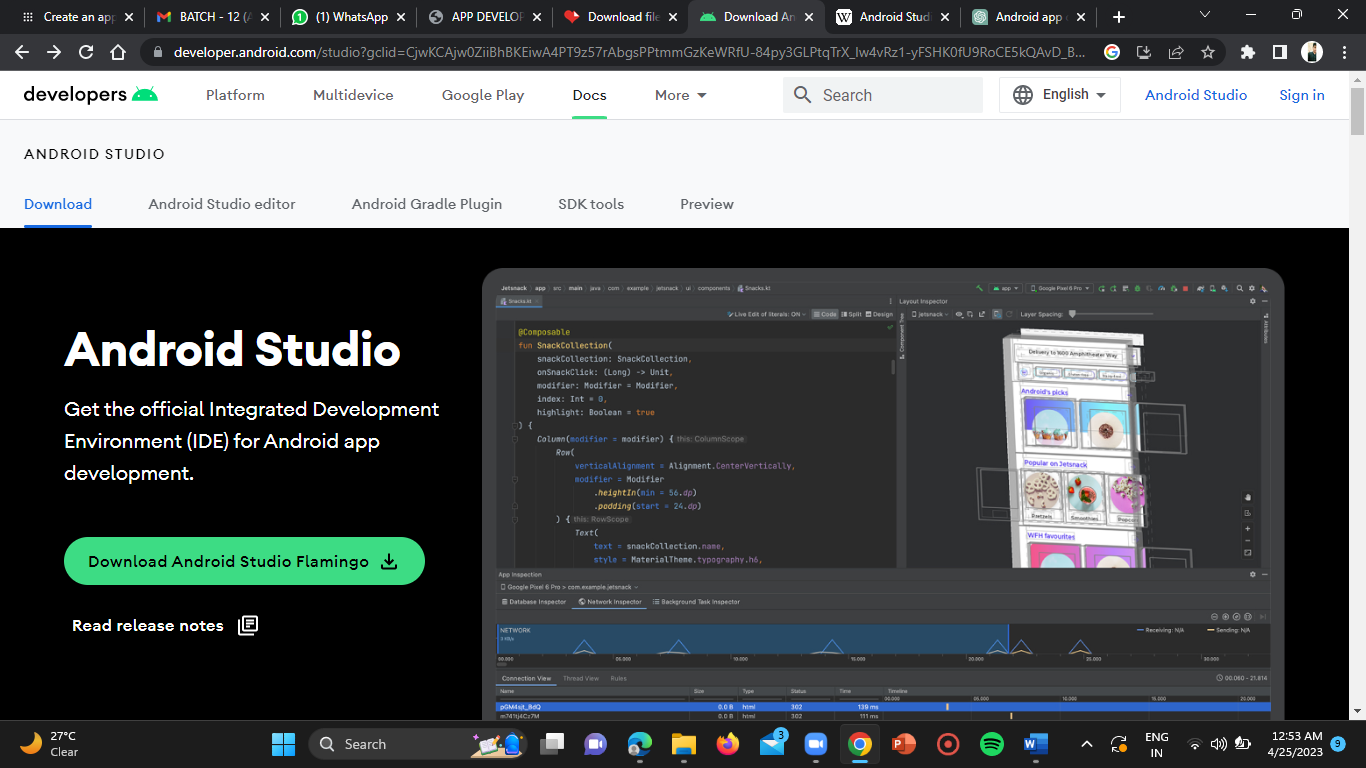
The following features are provided in the current stable version:

* Gradle-based build support
* Android-specific  refactoring and quick fixes
* Lint tools to catch performance, usability, version compatibility and other problems
* ProGuard integration and app-signing capabilities
* Template-based wizards to create common Android designs and components
* A rich [layout editor](https://en.wikipedia.org/wiki/Graphical_user_interface_builder) that allows users to drag-and-drop UI components, option to previewlayouts on multiple screen configurations.
* Support for building Android Wear  apps
* Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine.
* Android Virtual Device (Emulator) to run and debug apps in the Android studio.

**Installing android studio**

**System Requirement –** First your system OS must be either Windows, Max OS X or Linux with below requirement:

* Microsoft Windows 10/8.1/8/7/Vista/2003/XP (32 or 64 bit)
* Mac OS X 10.8.5 or higher, up to 10.10 to up 10.10.2 up 10.10.3 on 10.10.5 (Yosemite)
* GNOME or KDE or Unity desktop on Ubuntu or Fedora or GNU/Linux Debian
* Minimum RAM: 2GB
* Recommended RAM: 4GB
* Disk Space: 500 MB disk space
* Android SDK Space Reqirement: At least 1 GB for Android SDK, emulator system images, and caches
* JDK: Java Development Kit (JDK) 7 or higher
* Screen Resolution: 1280×800 minimum screen resolution
* Prefer faster processor according to your budget

****

**Android Studio:** The second thing you need is to [download Android Studio](https://abhiandroid.com/androidstudio/how-to-download-android-studio) on your system and install it. It is available for free download on Windows, Mac OS X and Linux OS.

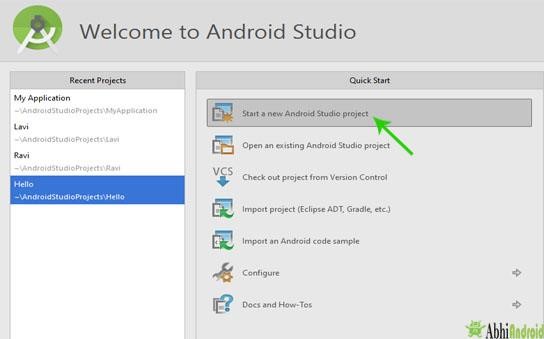
Create/start a new project in android studio

We create or start a new project in android studio to create a new Mobile Application for Android. To create a new project we need to enter the Application name, Company Domain, choose Phone and Tablet, fill Minimum SDK, Activity details and Activity Name & Layout Name in the text box. Below is the step by step guide.

***How To Start / Create a New Project In Android Studio:***

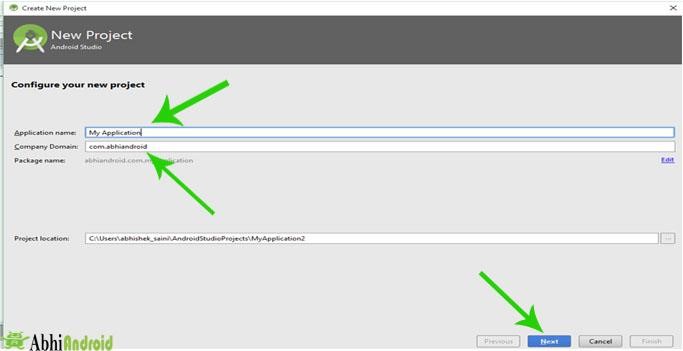
To create a new project perform the following steps:

**Step 1:** Firstly, open the Android Studio. You will see “Welcome to Android Studio” on your computer screen. After that you click on “Start a new Android Studio project”.

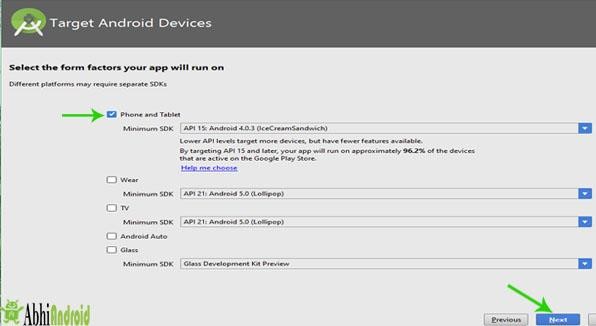


**Step 2:** After that you will see “New Project Android Studio” on your computer screen. Here enter the “Application name” and Type the “Company Domain” in the text box and then Click Next button.

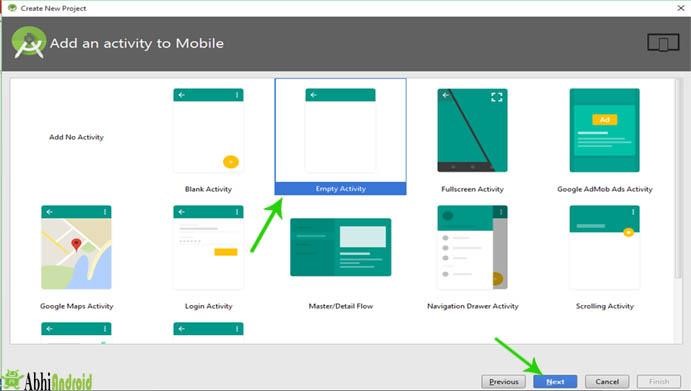
**Important Note:** Company domain is used to uniquely identify the App. So two App with same Application name will have different company domain. Another thing to know is that company domain is written in reverse. Ex: com.abhiandroid



**Step 3:** After that you will see “Target Android Devices” on your computer screen and here select the form factors your app will run on. Select “Phone and Tablet” field and enter the “Minimum SDK” in the text box and then Click Next button.

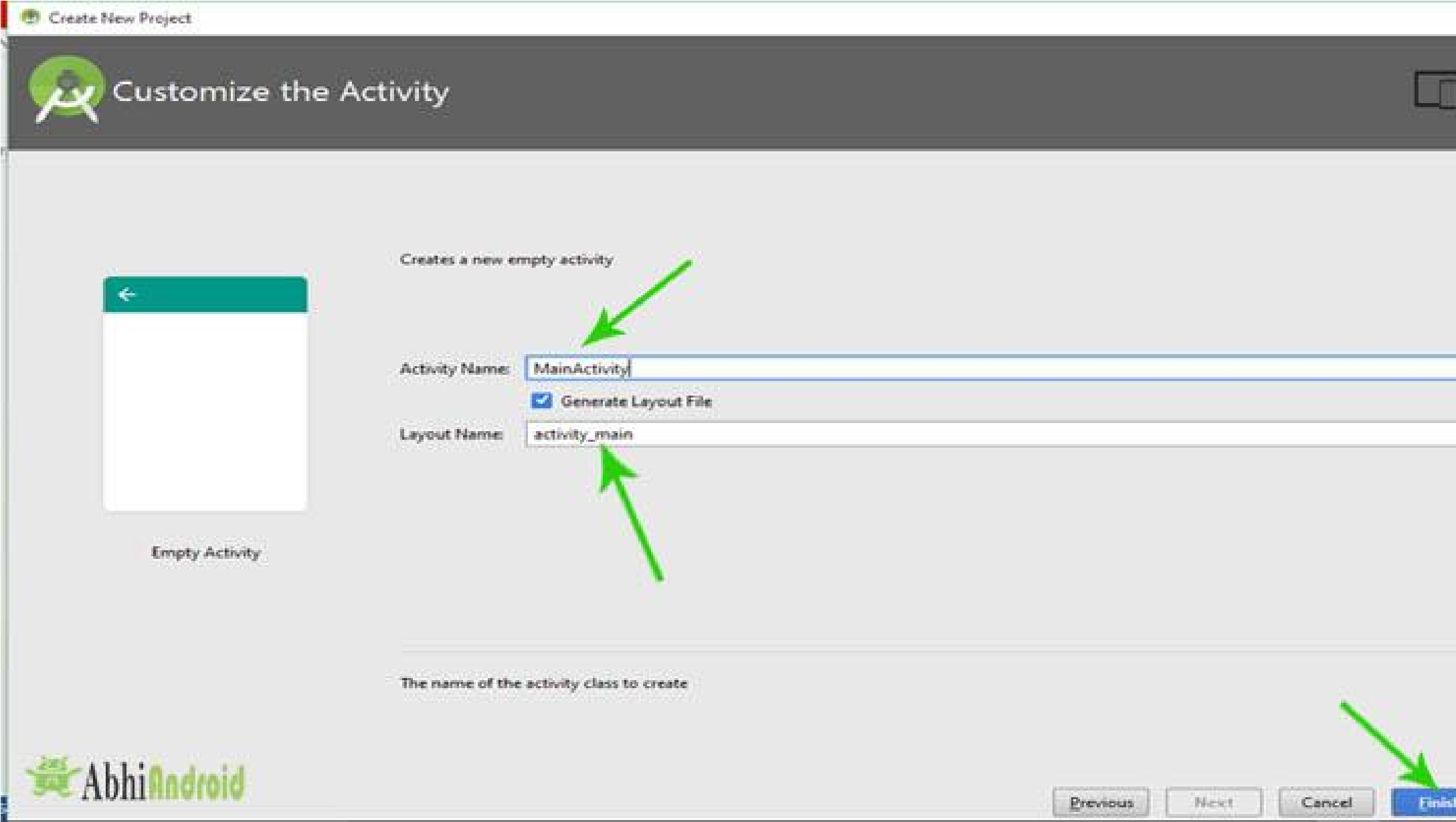


**Step 4:** After that you will see Add an activity to Mobile on your computer screen and Select the “Activity” as per requirement and then Click Next button. We selected Empty Activity.



**Step 5:** After that you will see Customize the Activity on your computer screen and enter the “Activity Name” in the text box and enter the “Layout Name” in the text box and then Click Finish button. Here JAVA and XML file will be created.

**Activity Name –** Name of JAVA file for programming

**Layout Name –** Name of XML file for designing UI

***How To Start A New Project When A Project Is Already O***

***pen In Android Studio:***

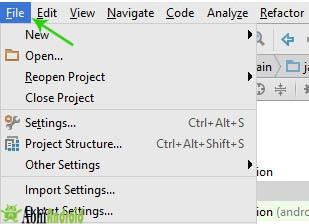
**Step 1**

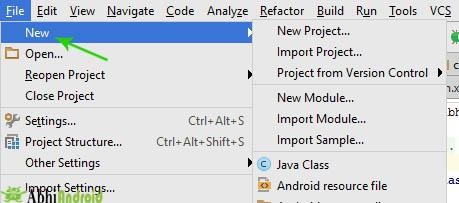
:

Firstly, we will Click on File.

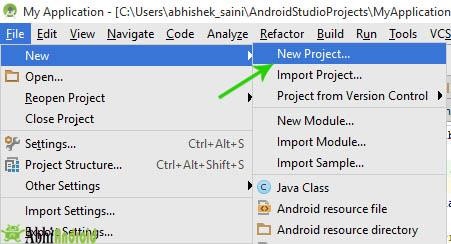
**Step 2:**

After that we will Click on New.





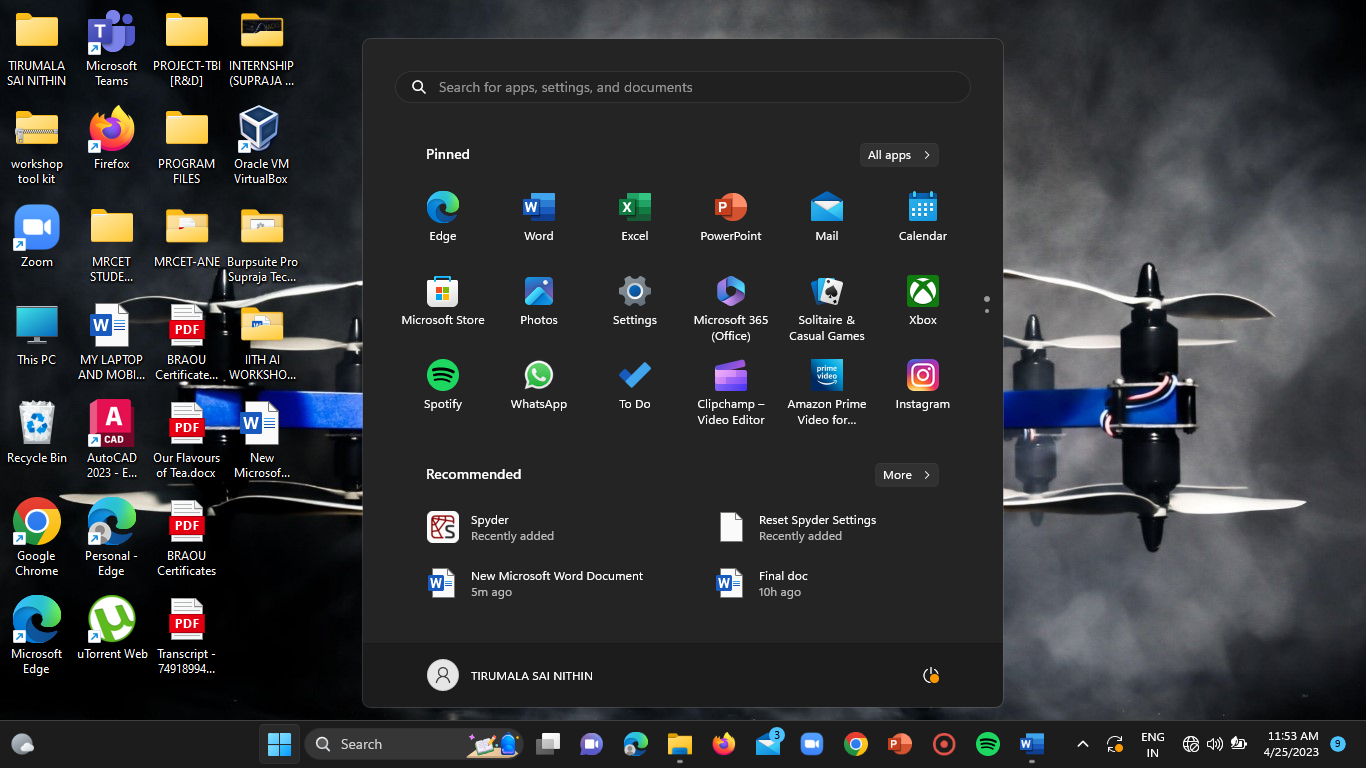
**Step 3:** After that we will Click on New Project.



STEPS TO CREATE AN APP TO SEND SMS USING INTENT

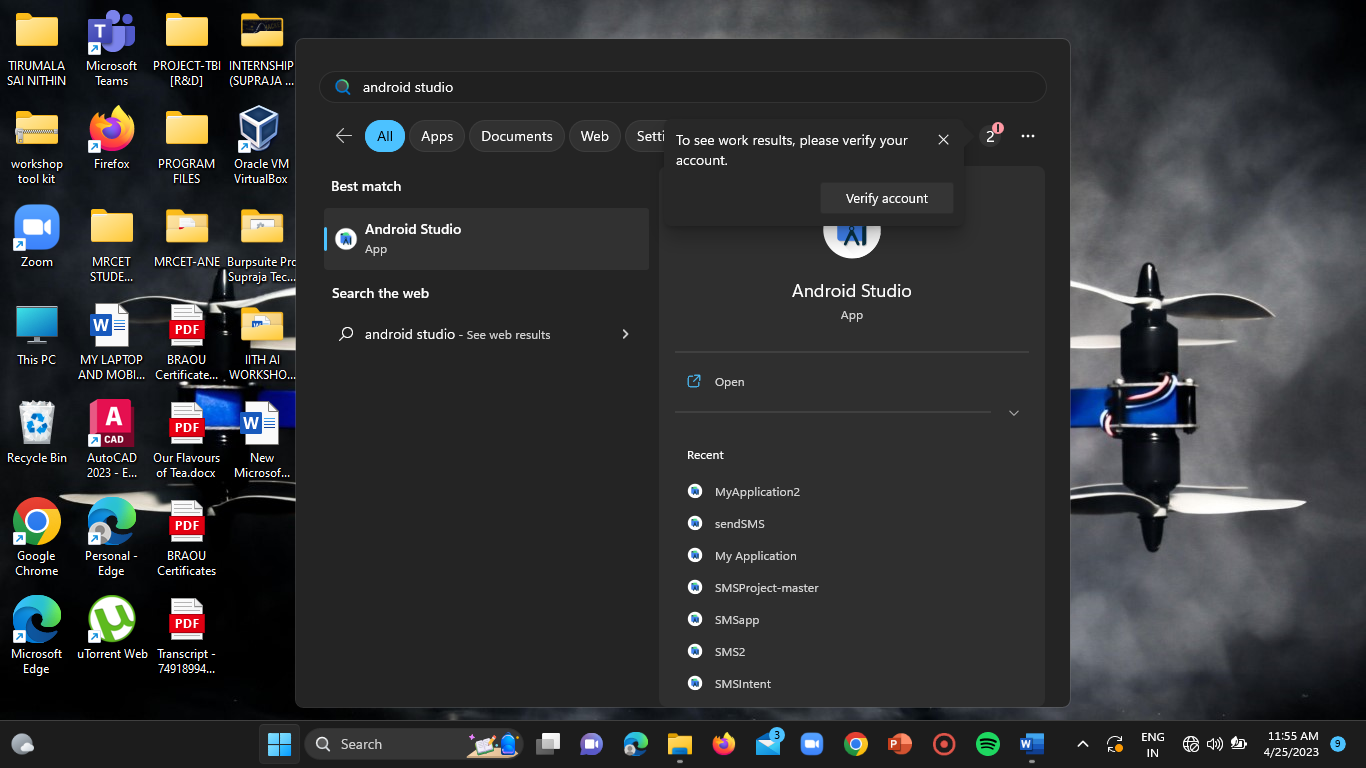
**STEP-1:**

Click on start button on your PC.



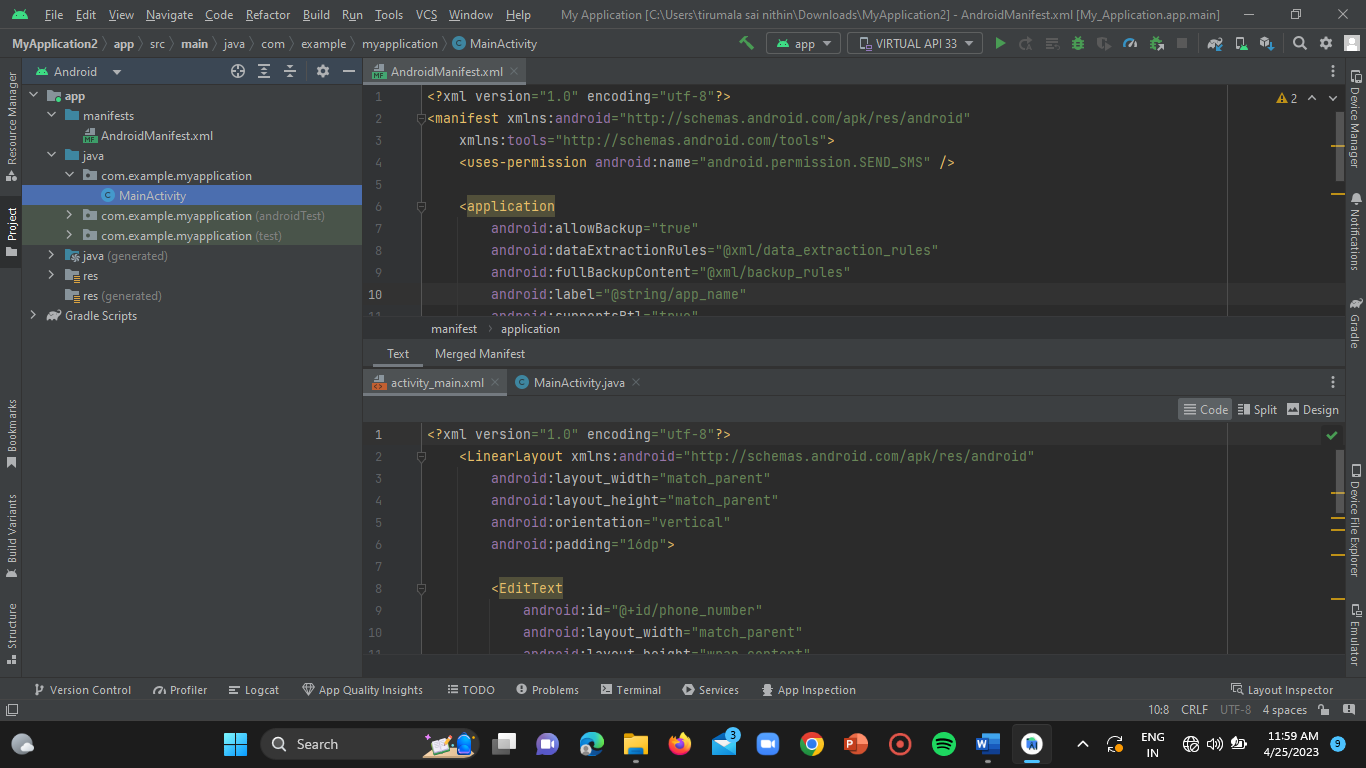
**STEP-2:**

Enter “ANDROID STUDIO” in the search bar and click on enter.



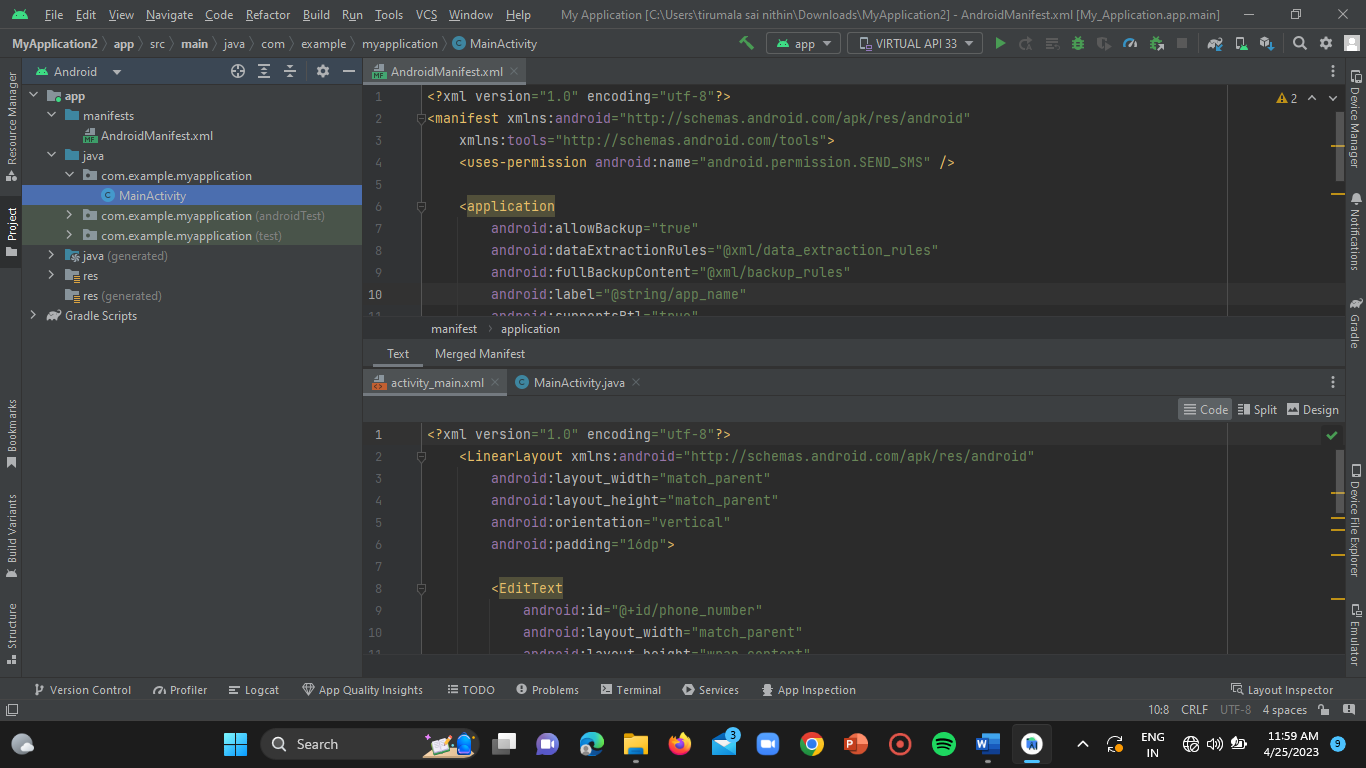
**STEP-3:**

Click on open under “ANDROID STUDIO”. Then the android studio application will be opened.



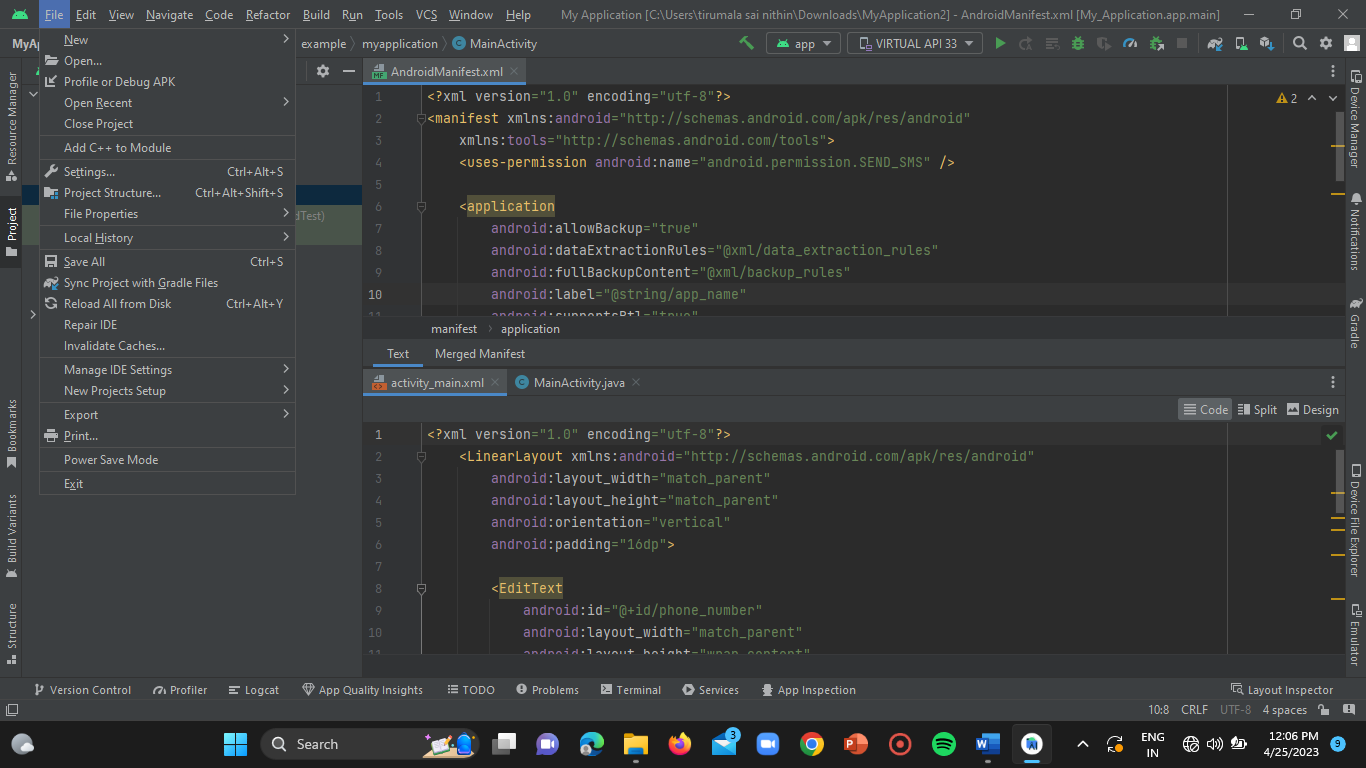
**STEP-4:**

Click on the “FILE” option located at the left top of the homepage of android studio.



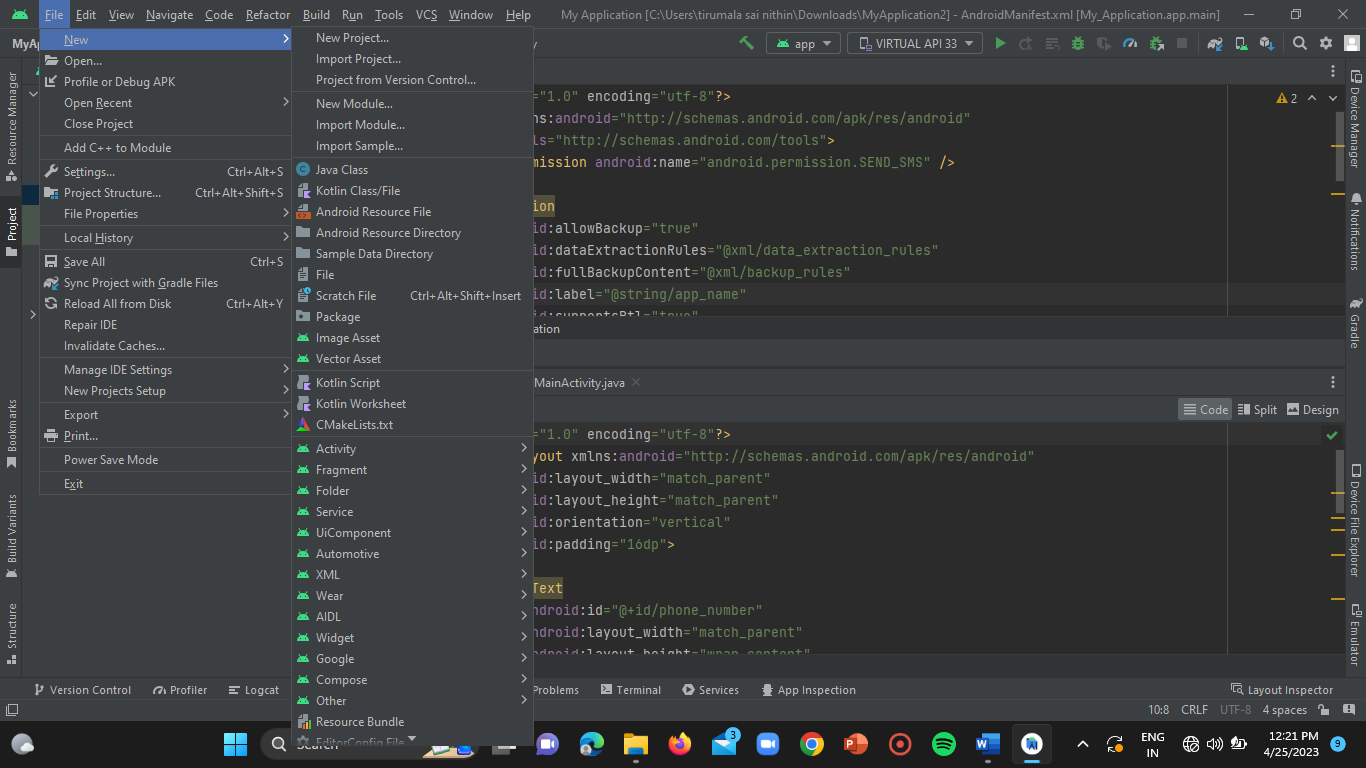
**STEP-6:**

The dialogue box of the file menu gets opened and then select the “NEW” option from the menu.



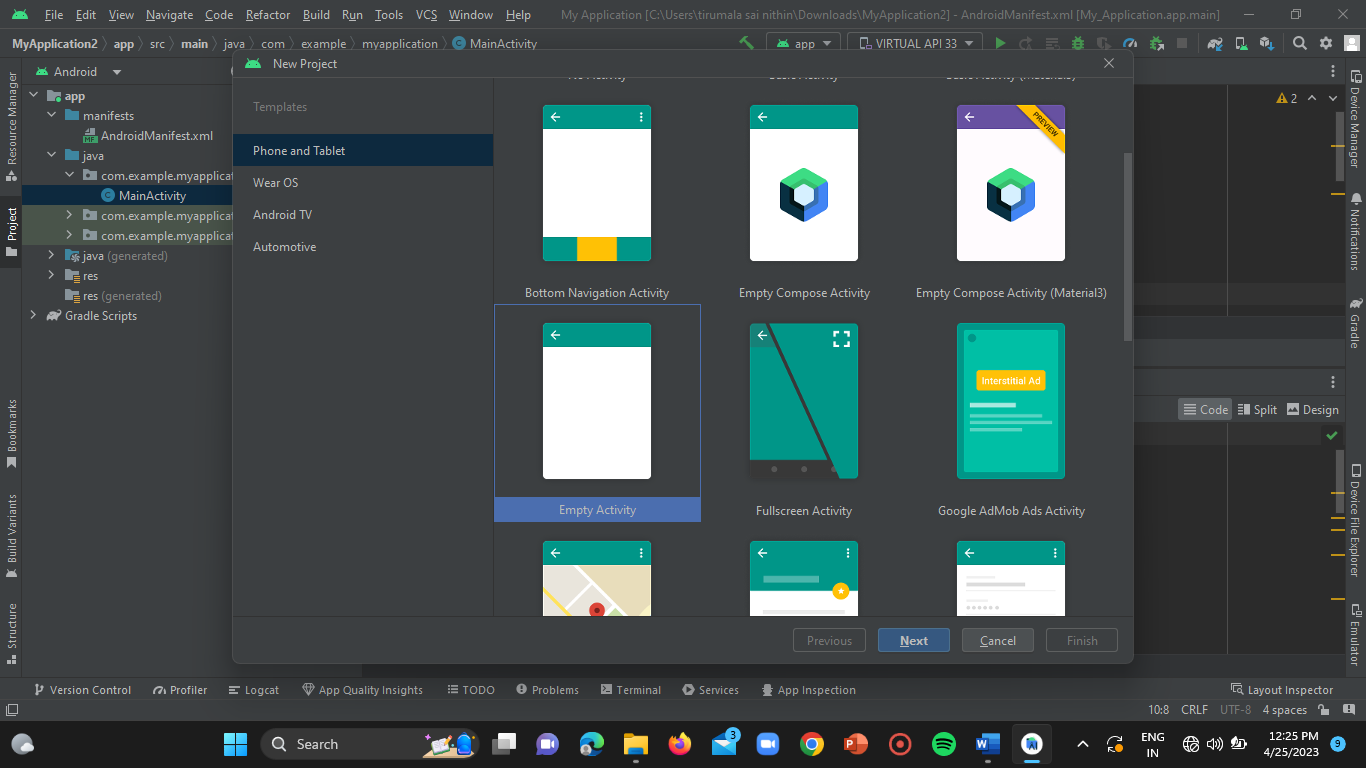
**STEP-7:**

Select “NEW PROJECT” from the menu.

****

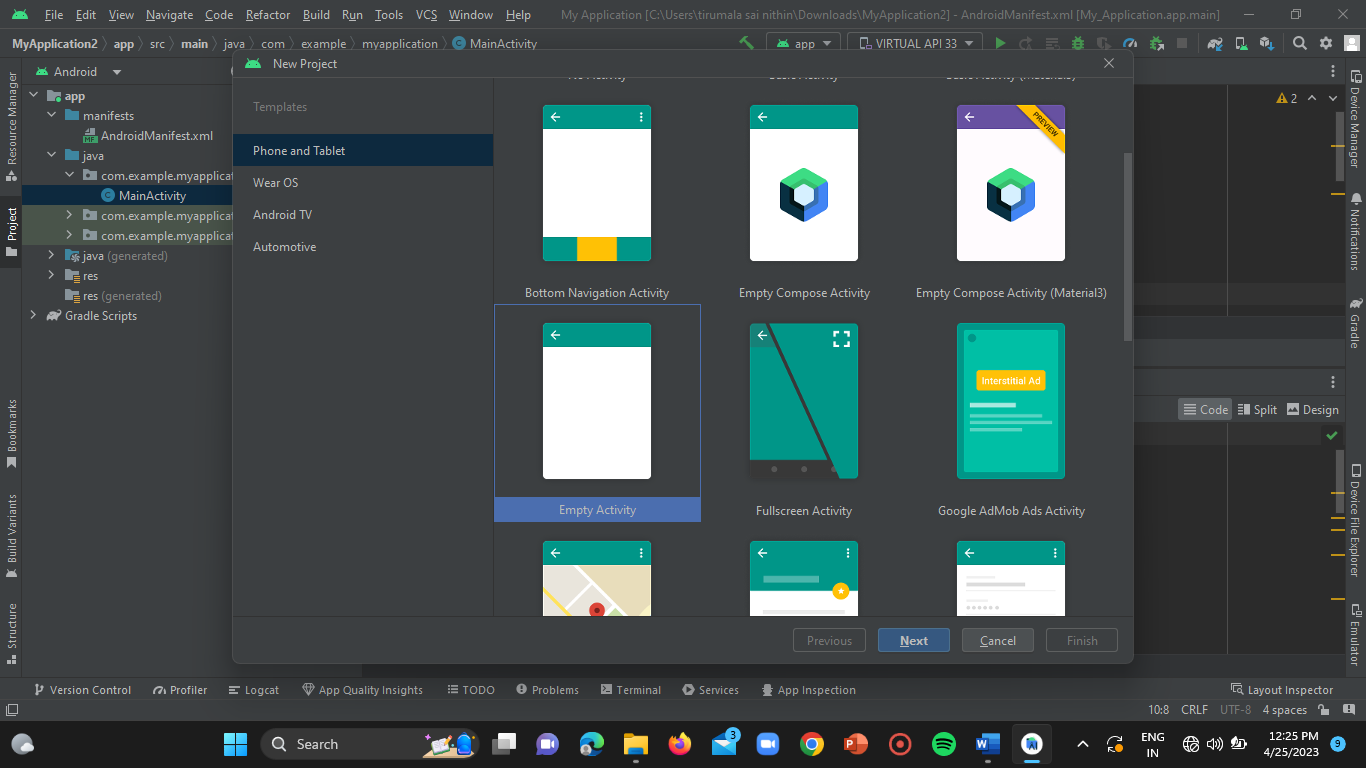
**STEP-8:**

Select the “BLANK ACTIVITY” from the project layout menu.



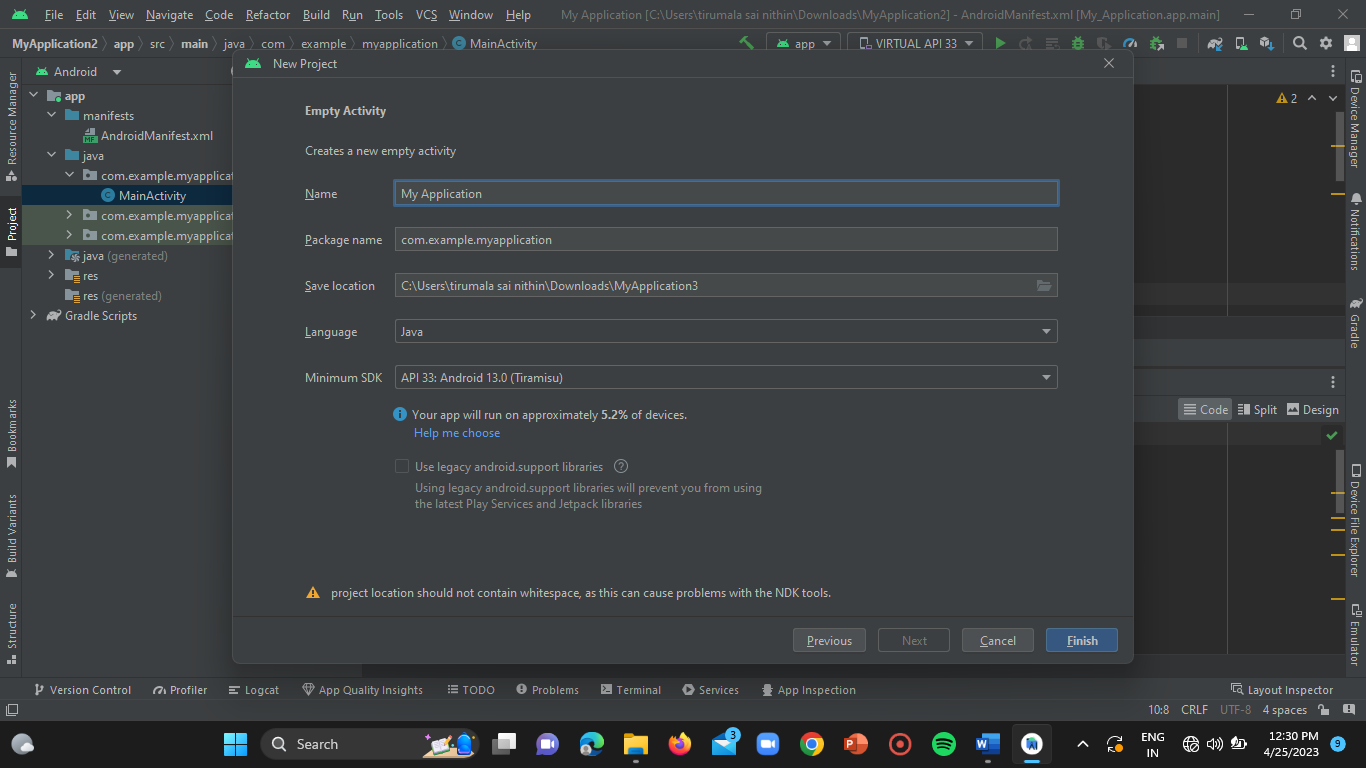
**STEP-9:**

Click on “NEXT”.



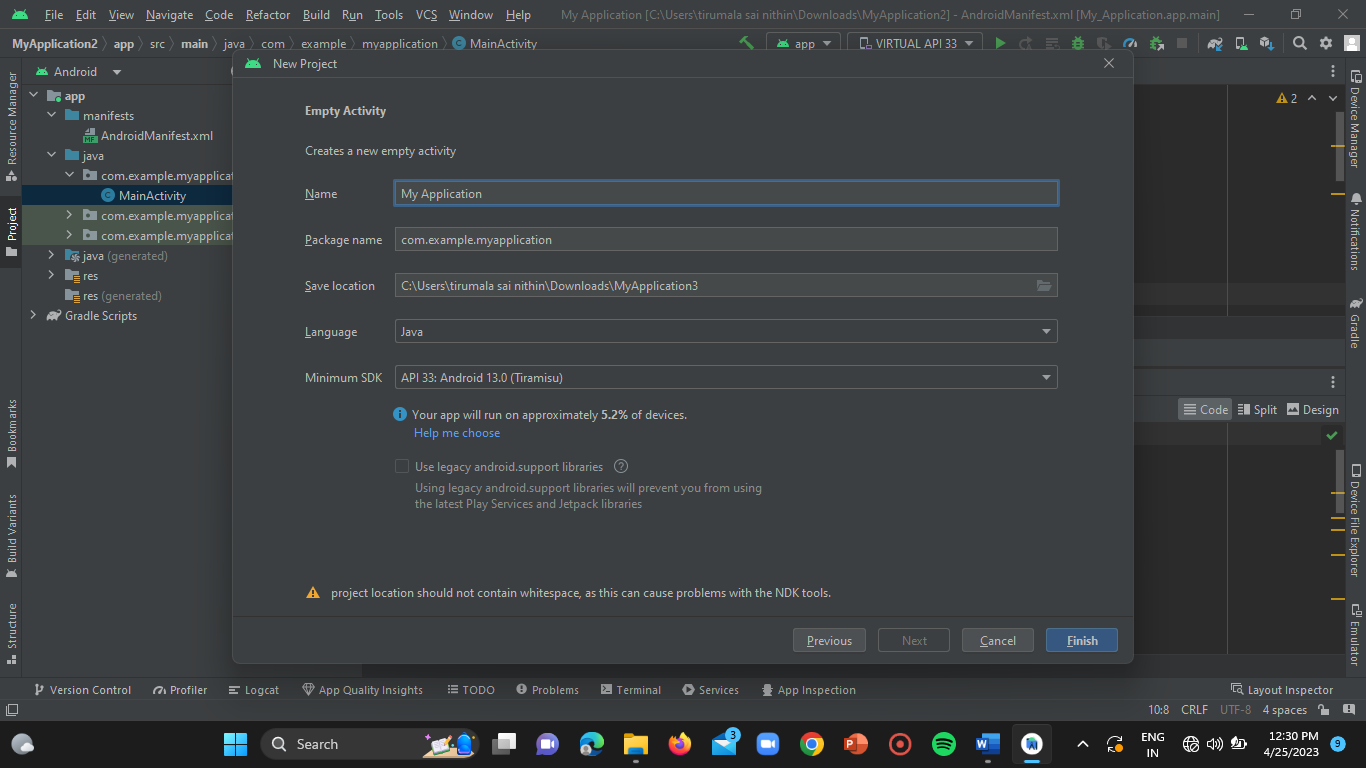
**STEP-10:**

Enter your project name in the name input bar.



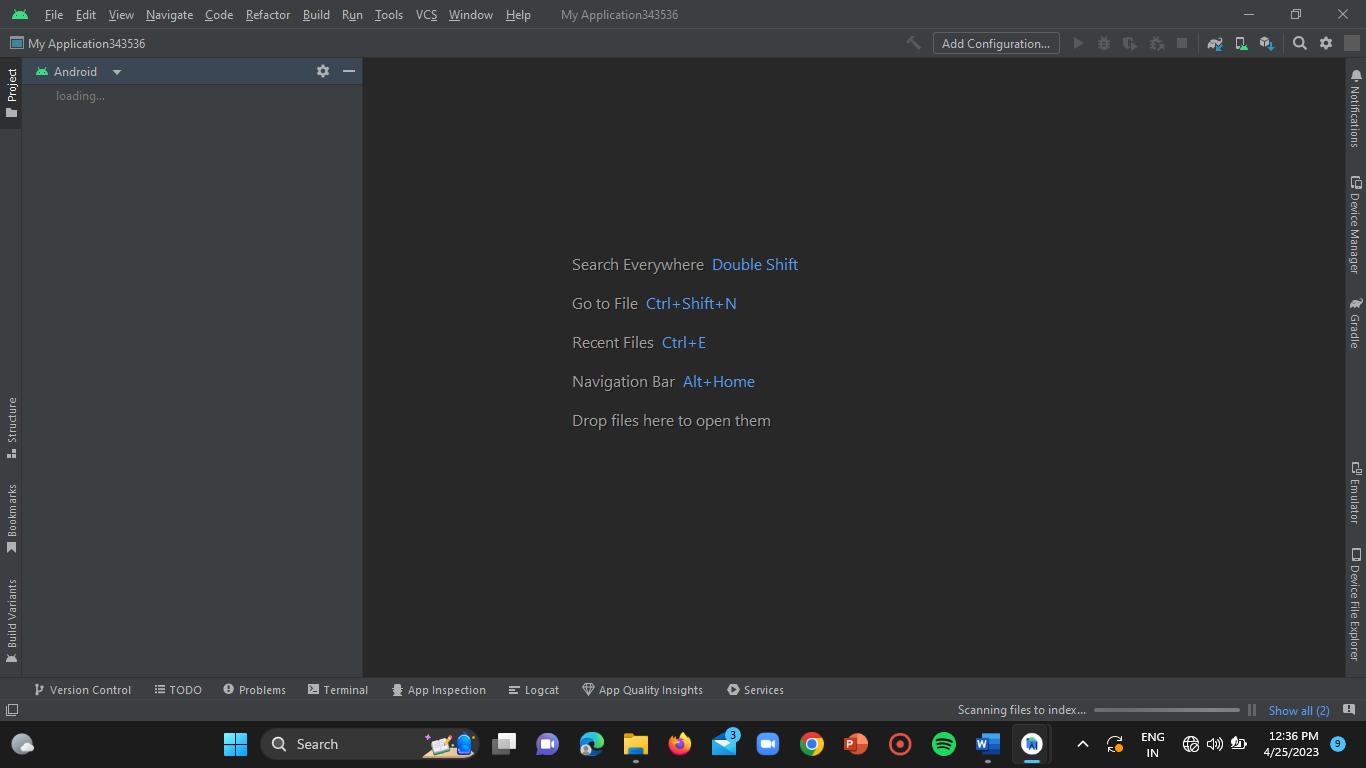
**STEP-11:**

Then click on “FINISH” button. Your new project gets opened and you can edit it based on your project requirements.



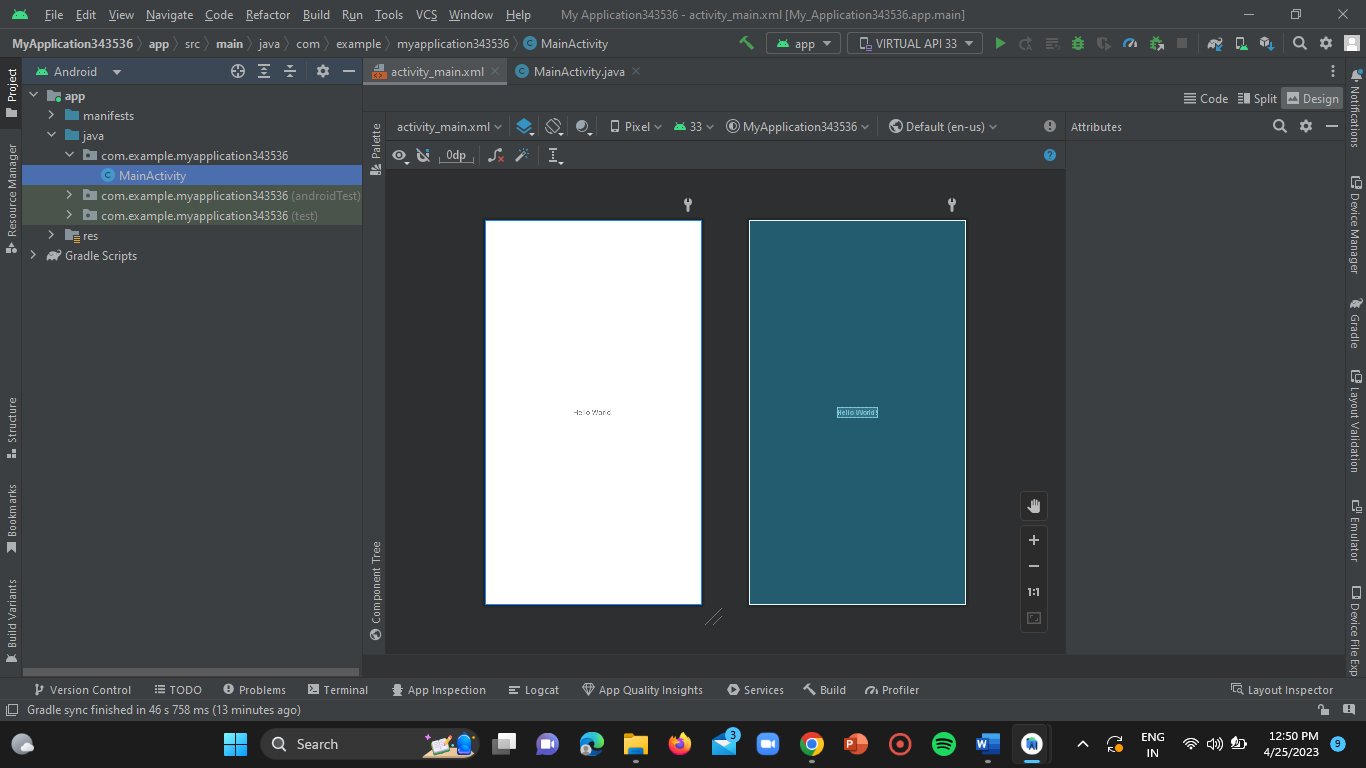
**STEP-12:**

Wait until all your project libraries get loaded.



**STEP-13:**

Click on “MANIFEST” option on the left side of the homepage.



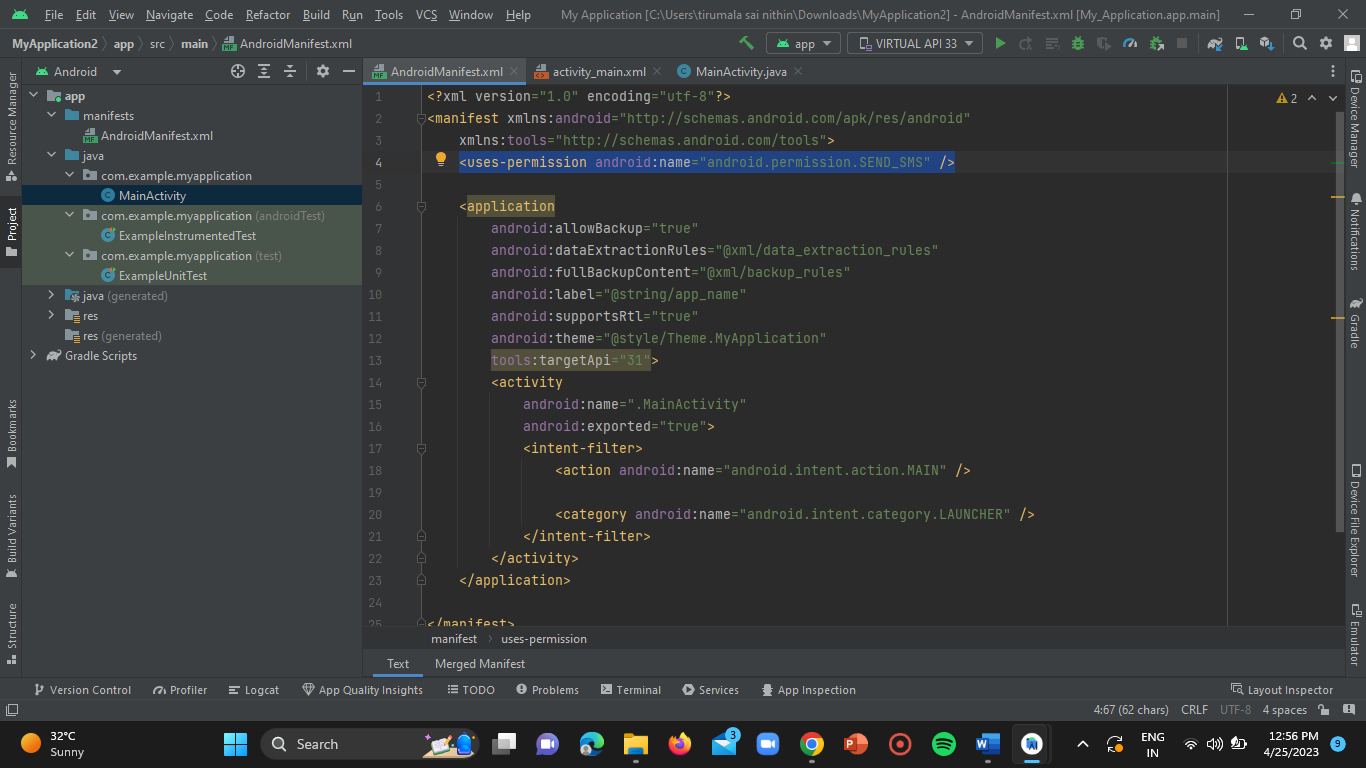
**STEP-14:**

Double click on “AndroidManifest.xml” file.



**STEP-15:**

Now add the **“SEND\_SMS”** permission to your **“AndroidManifest.xml”** file.

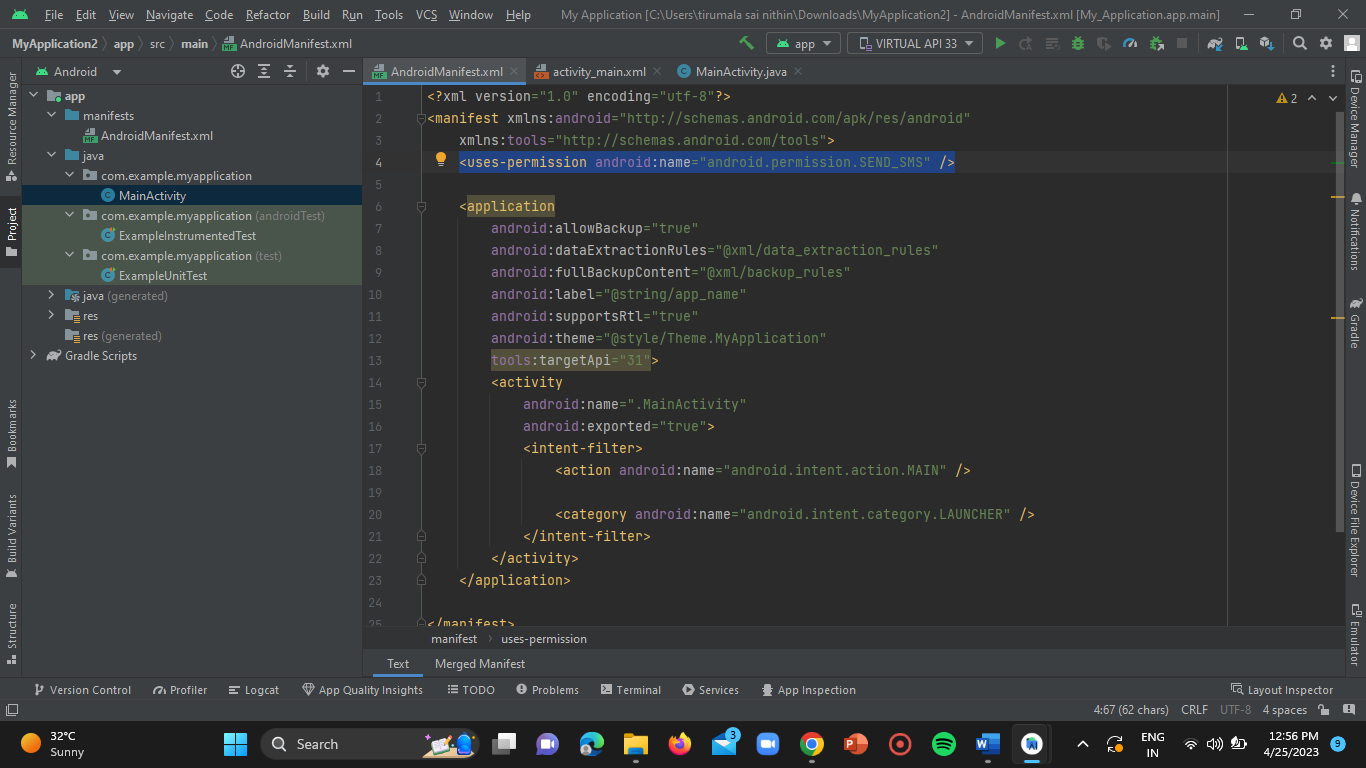


**CODE:**

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

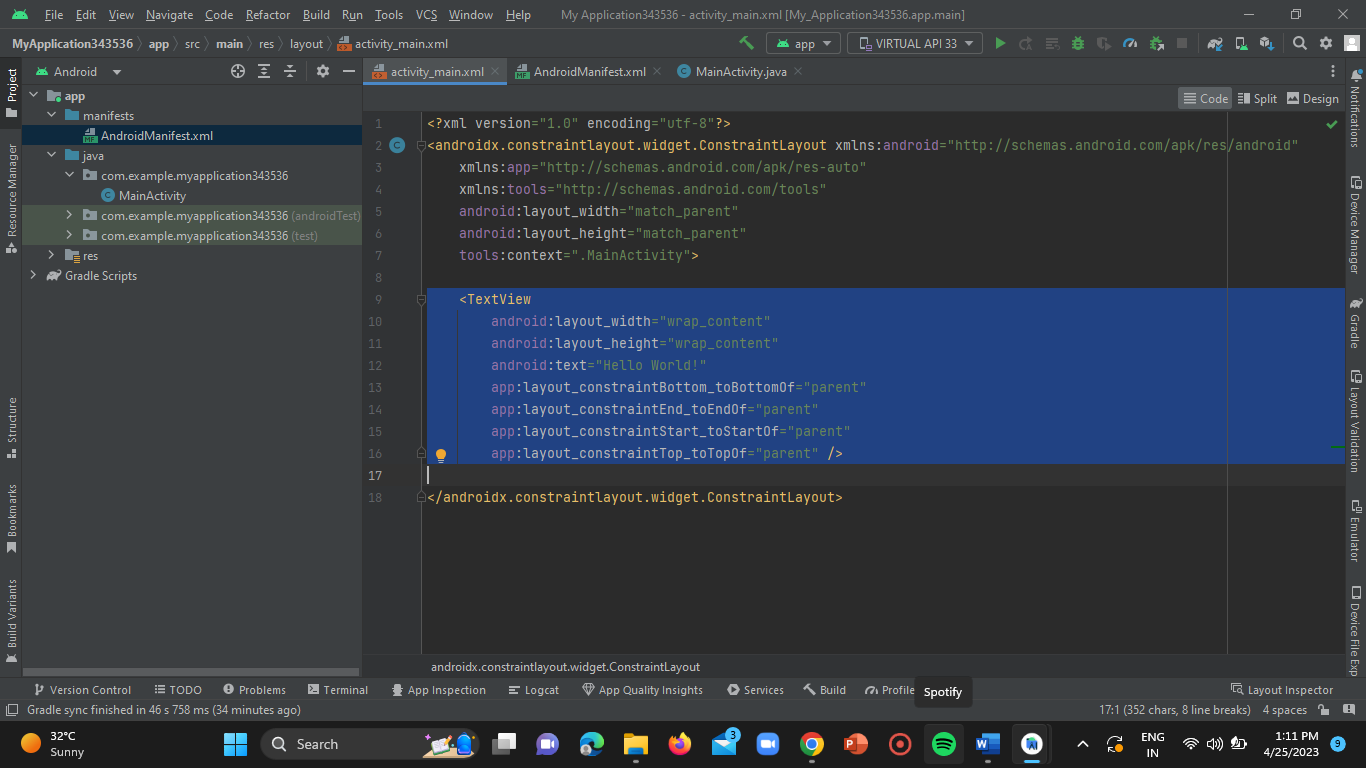
**STEP-16:**

Click and open the **“activity\_main.xml”** page on home screen of the project.



**STEP-17:**

Remove the <TEXT VIEW> code from the code file.



**STEP-18:**

Create a layout for your activity with two EditTexts (for the phone number and message) and a Button (to send the SMS).

**CODE:**

<?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"

android:padding="16dp">

<EditText

android:id="@+id/phone\_number"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Phone number" />

<EditText

android:id="@+id/message"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Message" />

<Button

android:id="@+id/send\_sms"

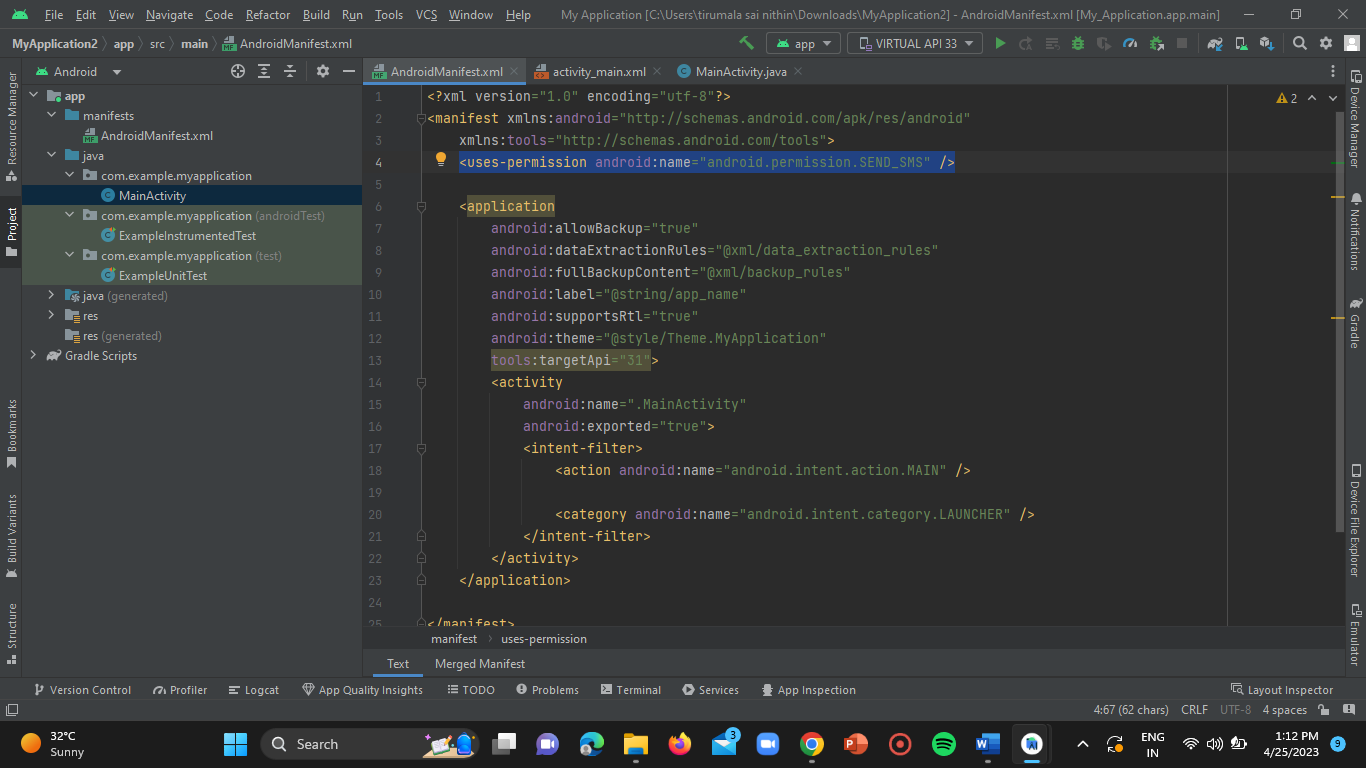
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

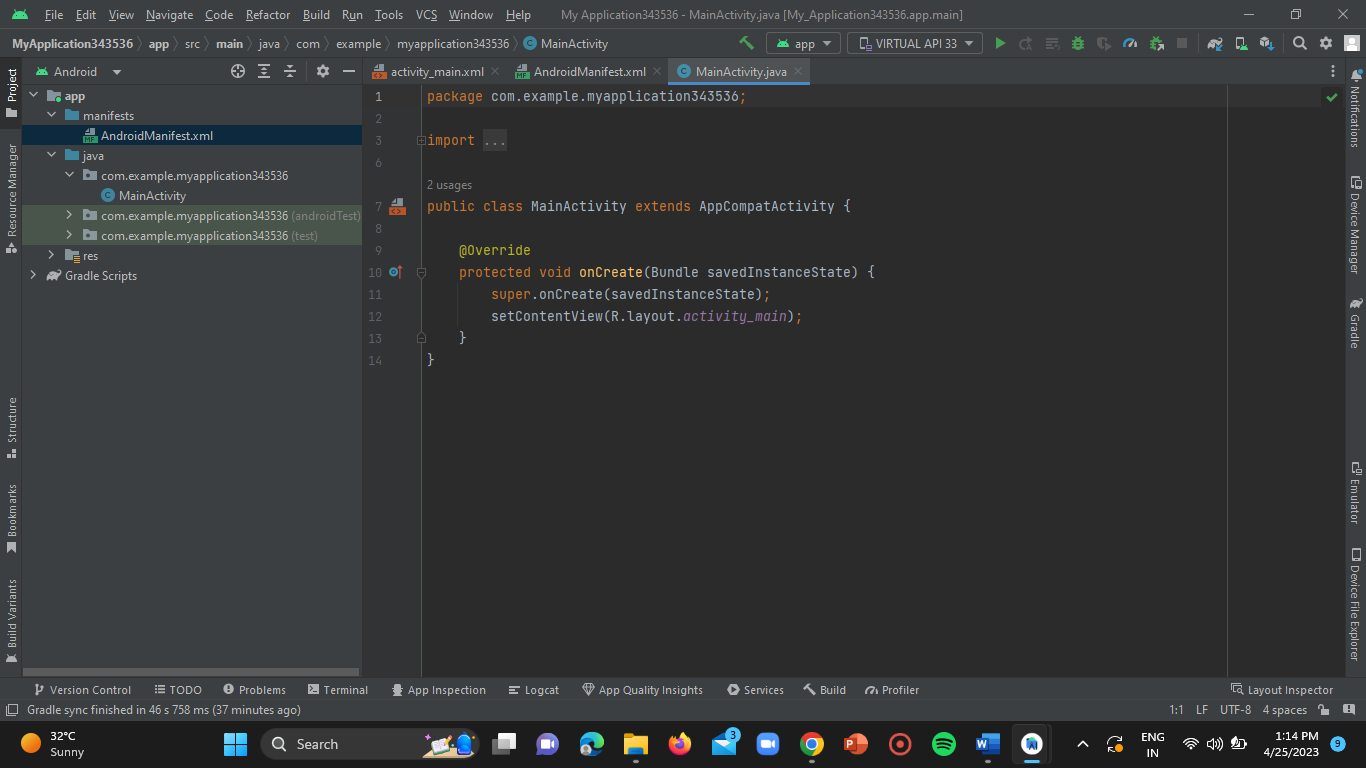
android:text="Send SMS" />

</LinearLayout>



**STEP-18:**

Open **“MainActivity.java”** file.



**STEP-19:**

Now add the below mentioned code to the .java file.

**CODE:**

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button sendSmsButton = findViewById(R.id.send\_sms);

sendSmsButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sendSms();

}

});

}

private void sendSms() {

EditText phoneNumberEditText = findViewById(R.id.phone\_number);

EditText messageEditText = findViewById(R.id.message);

String phoneNumber = phoneNumberEditText.getText().toString();

String message = messageEditText.getText().toString();

Intent smsIntent = new Intent(Intent.ACTION\_VIEW);

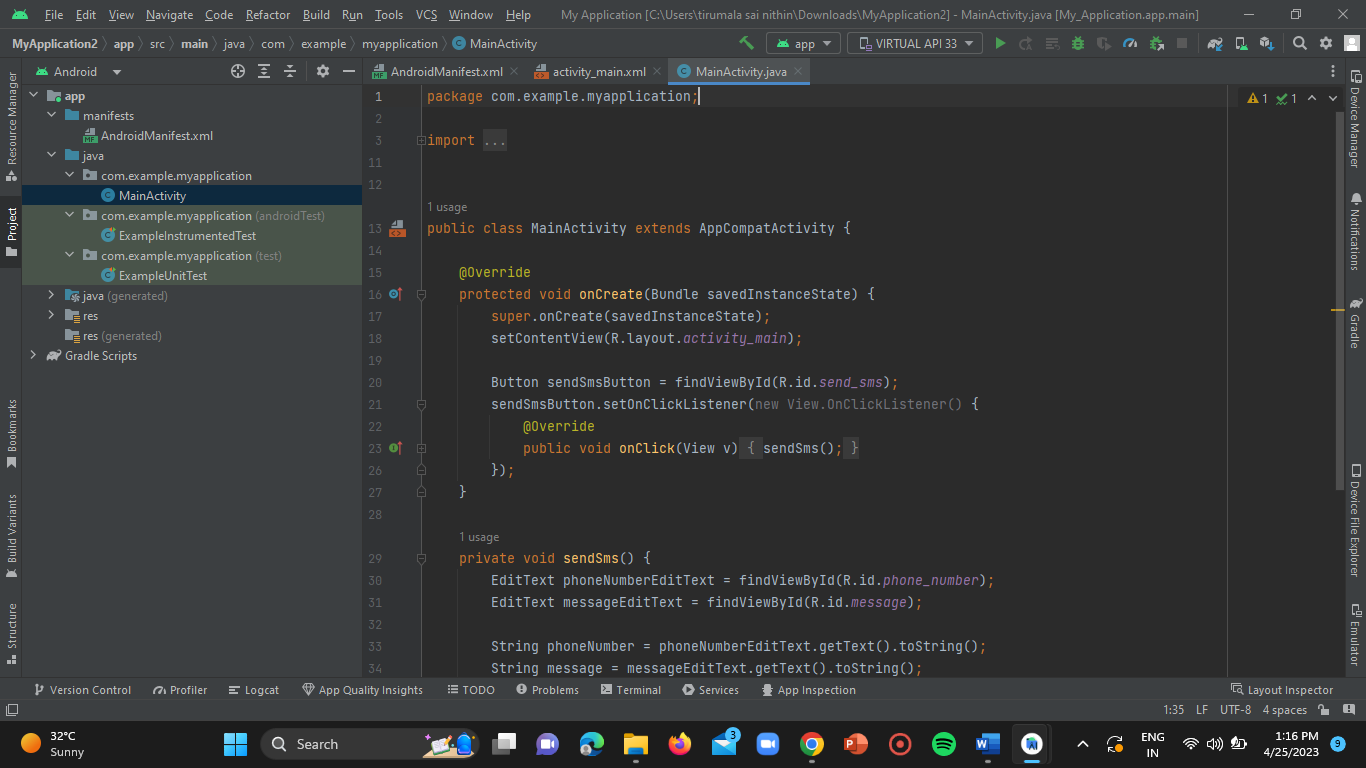
smsIntent.setData(Uri.parse("smsto:" + phoneNumber));

smsIntent.putExtra("sms\_body", message);

startActivity(smsIntent);

}

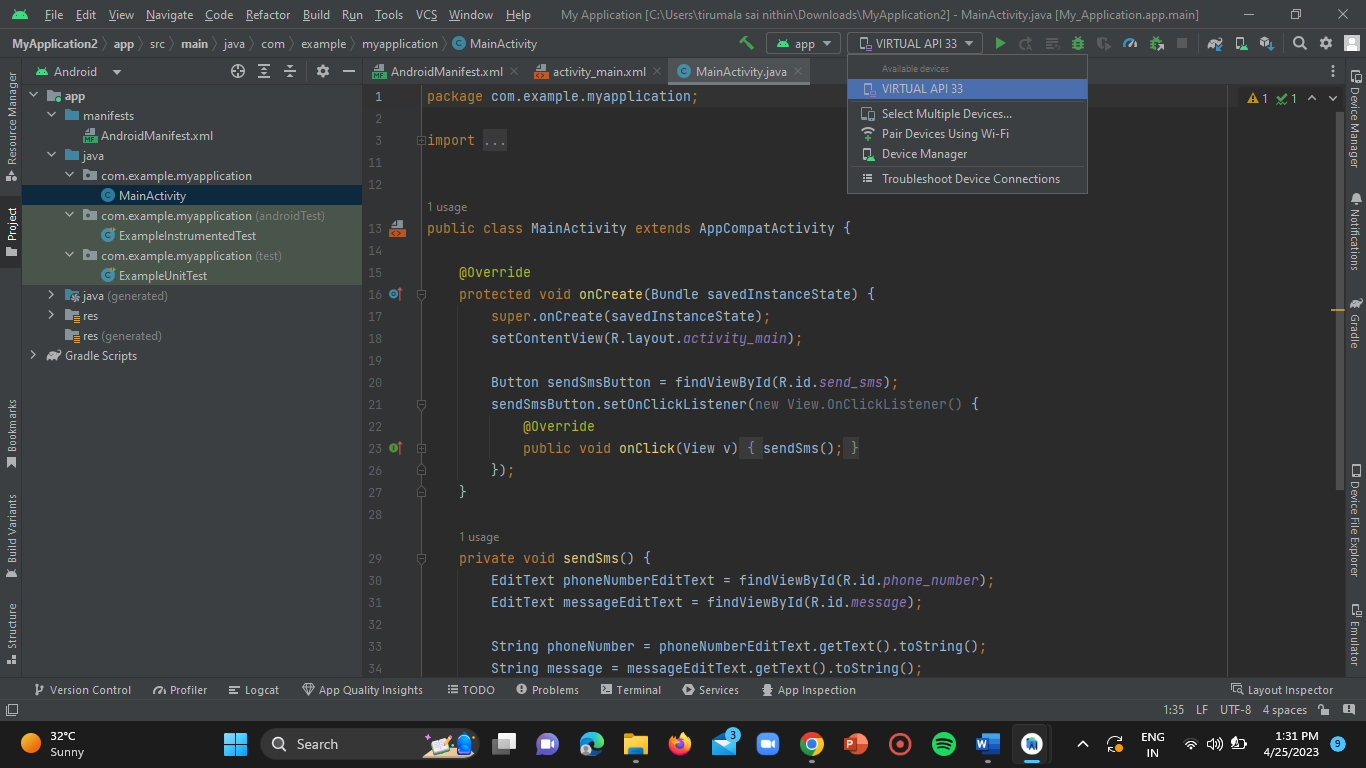
}



This code initializes the button click listener and calls the **sendSms()** method when the button is clicked. The **sendSms()** method retrieves the phone number and message from the EditTexts, creates an Intent with the **ACTION\_VIEW** action, sets the data to the phone number, and adds the message as an extra. Finally, it starts the activity with the Intent, which will open the default SMS app with the phone number and message pre-filled.Now, when you run the app, you can enter a phone number and message, and then click the "Send SMS" button to send the SMS using the default SMS app on your Android device.

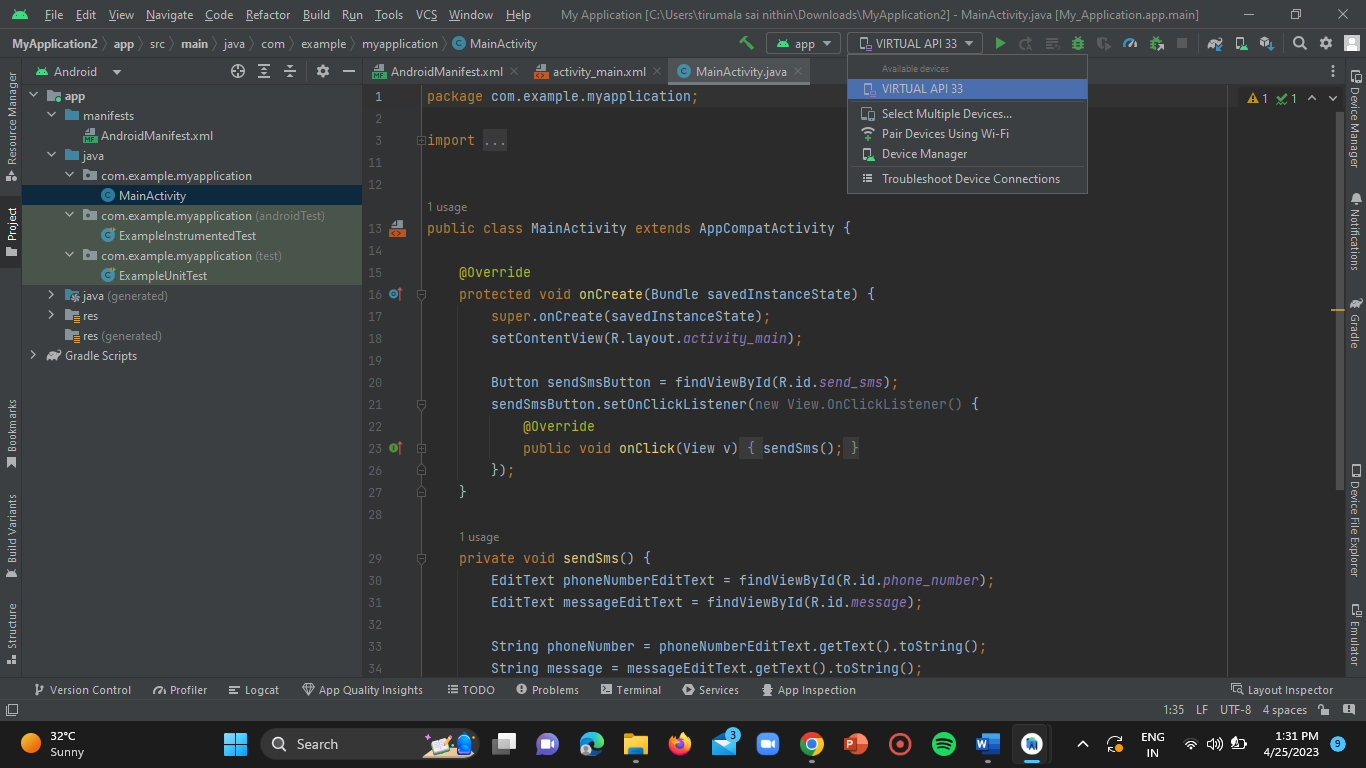
**STEP-20:**

Now click on the device pairing option.



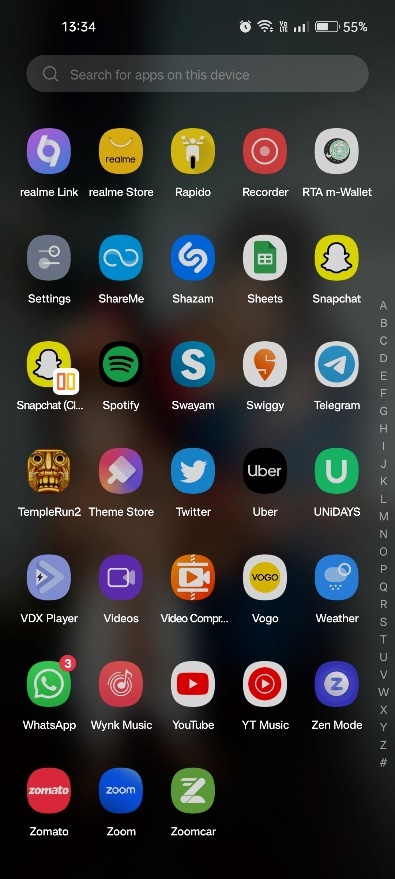
**STEP-21:**

Click on **“pair devices using wifi”** option.



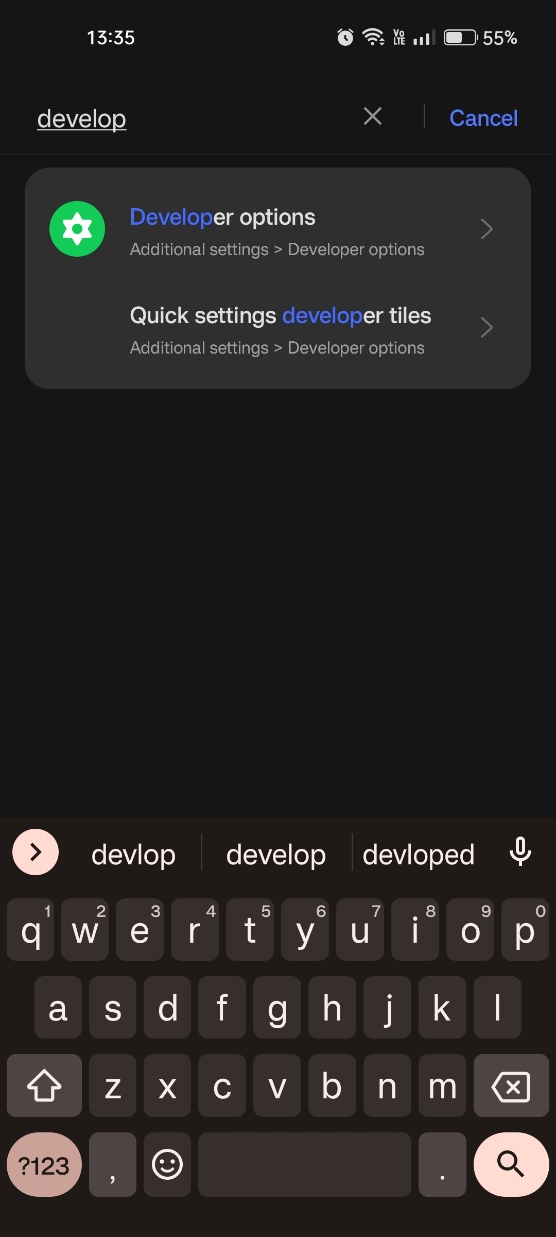
**STEP-22:**

Open the app menu in your mobile and click on settings.



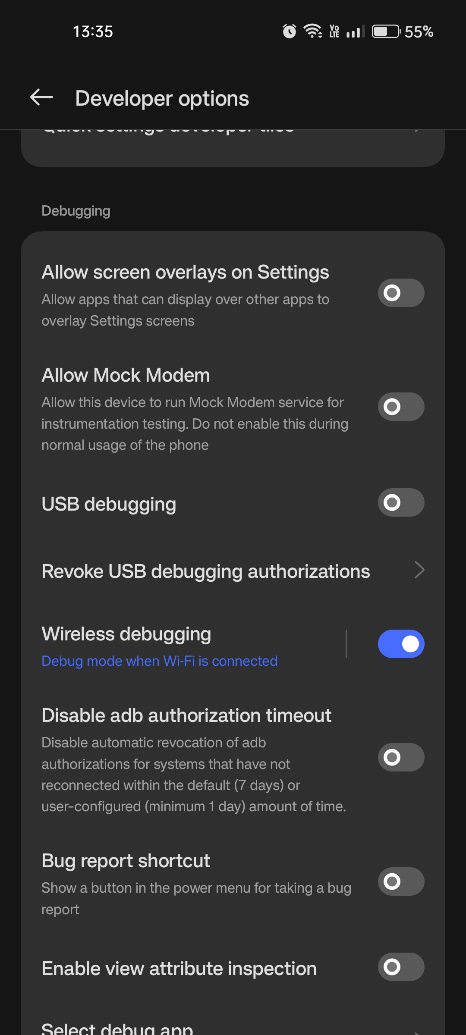
**STEP-23:**

Search for “developer options” and open it.



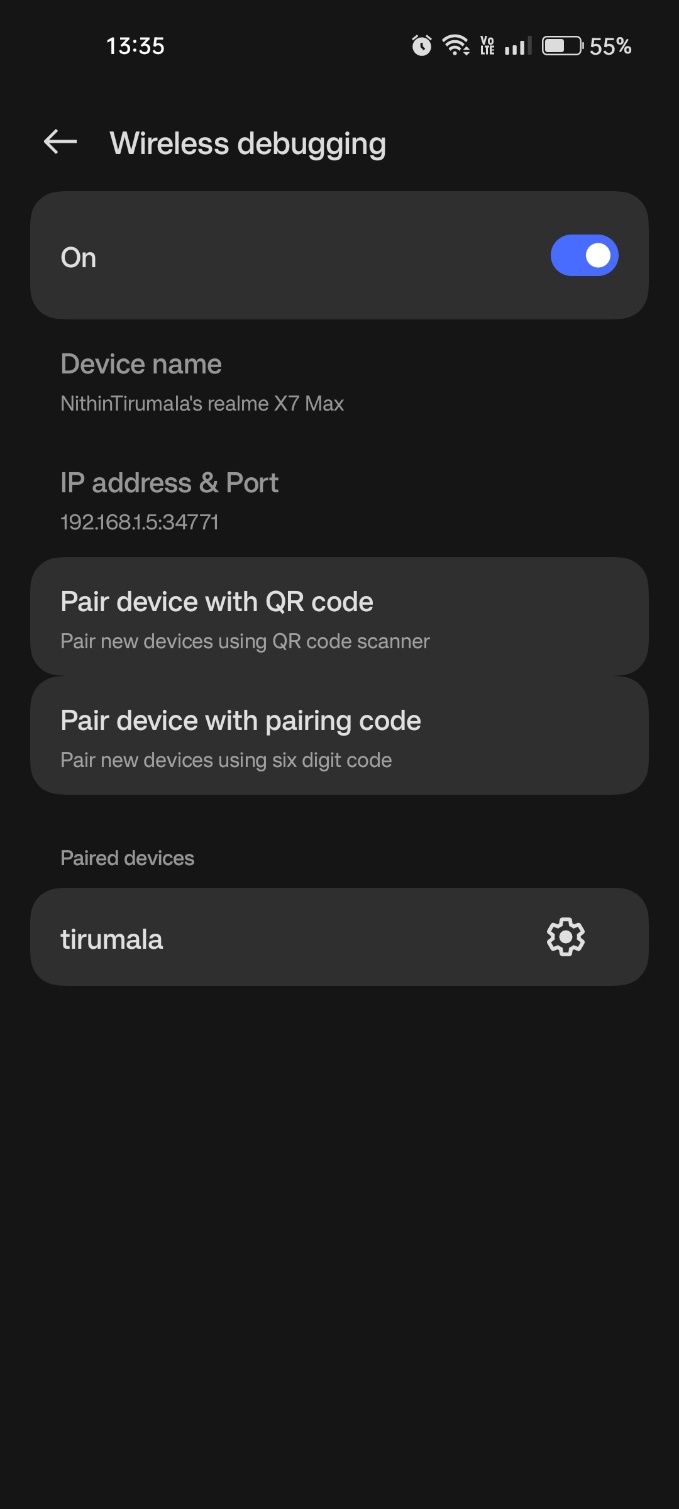
**STEP-24:**

Enable the developer options and open it. Then enable the “wireless debugging” option.



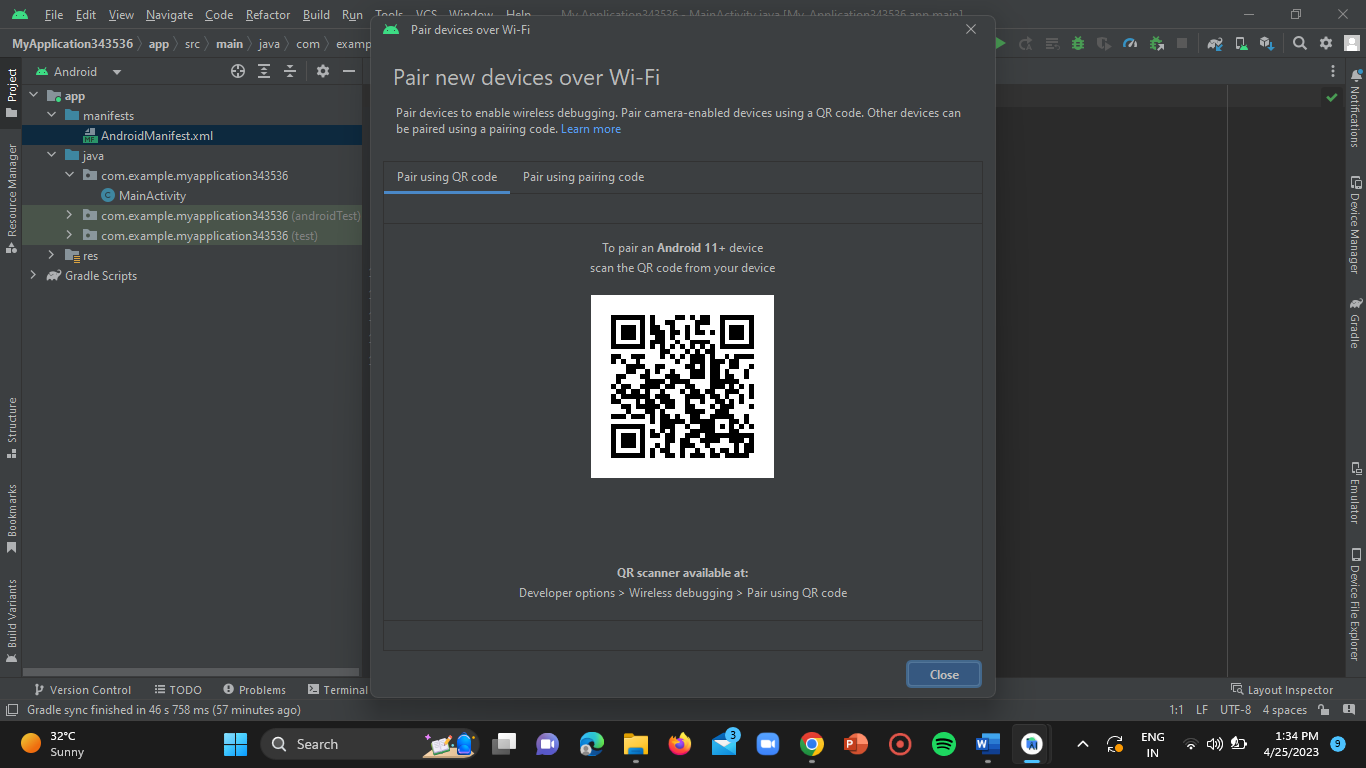
**STEP-25:**

Now click on the option **“pair device with QR”**.



**STEP-26:**

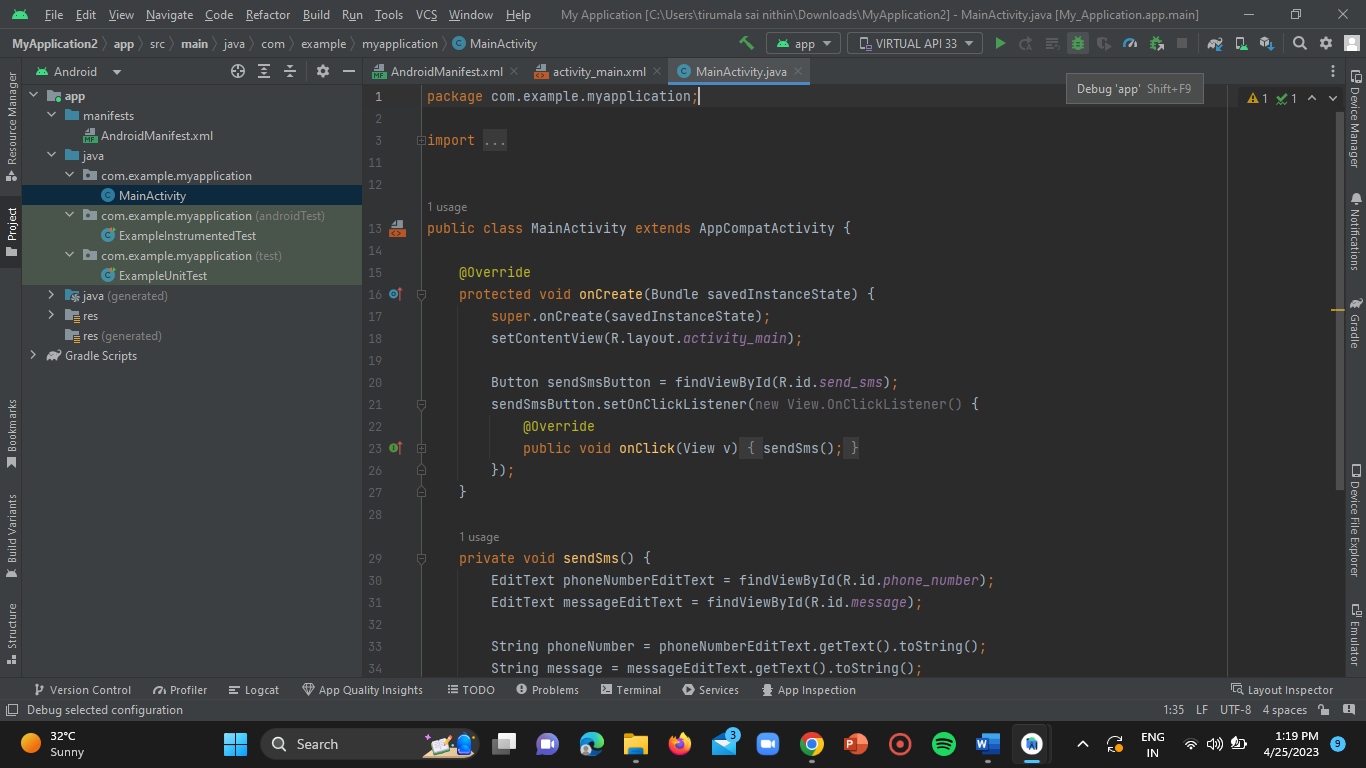
Now scan the QR code which is displayed in the android studio as mentioned below.



Then the mobile gets connected to your PC.

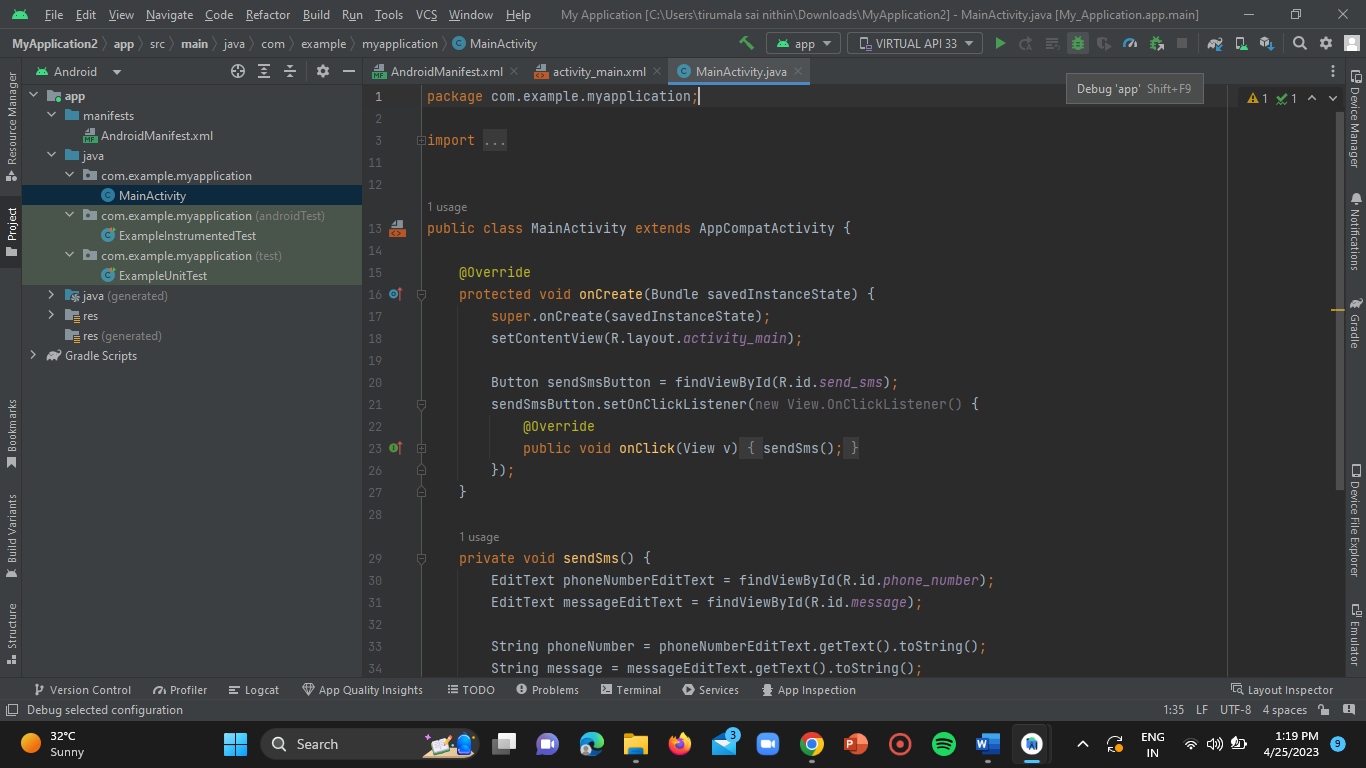
**STEP-27:**

Now click on the **“DEBUG”** option.



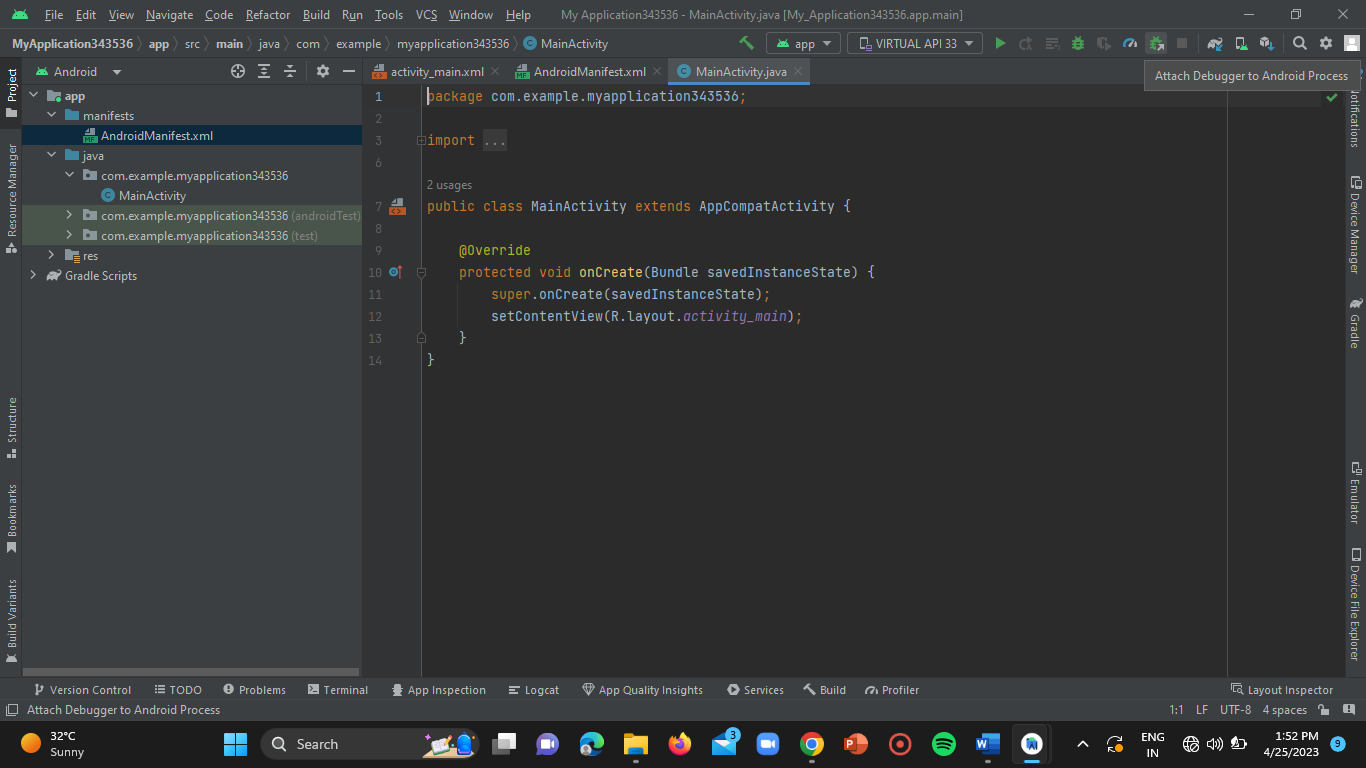
**STEP-28:**

Once check for the errors by clicking on the “RUN” option.



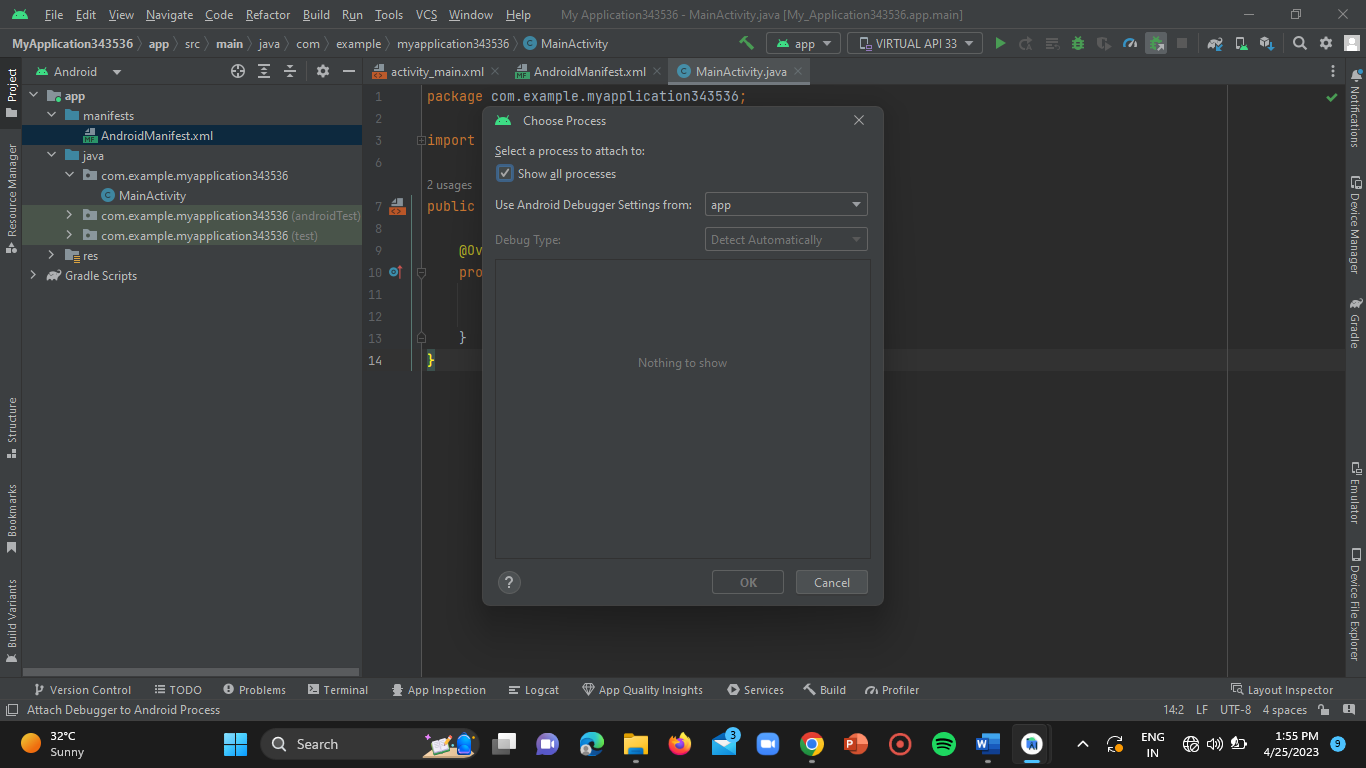
**STEP-29:**

Now click on the “attach debugger” option in the homepage of the project at the top.



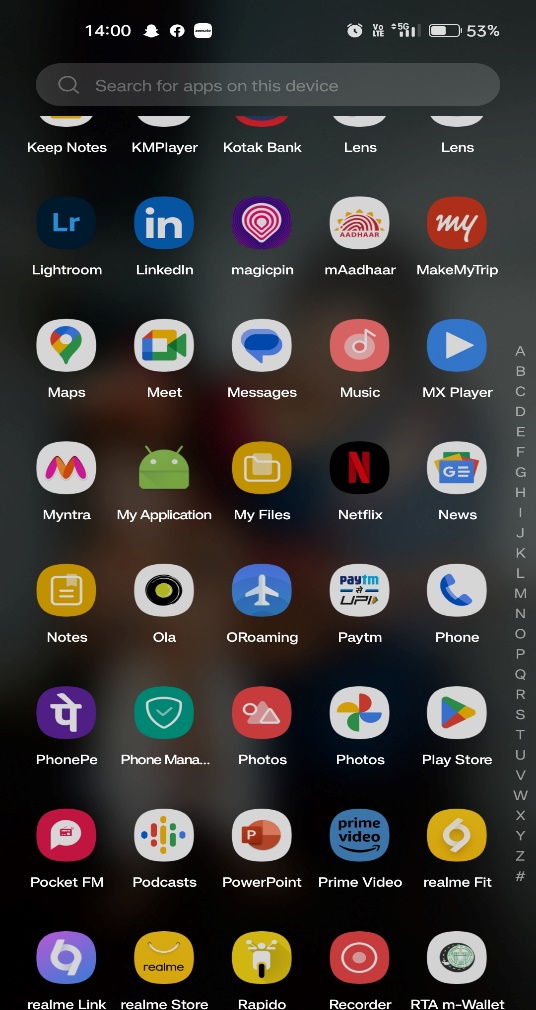
**STEP-30:**

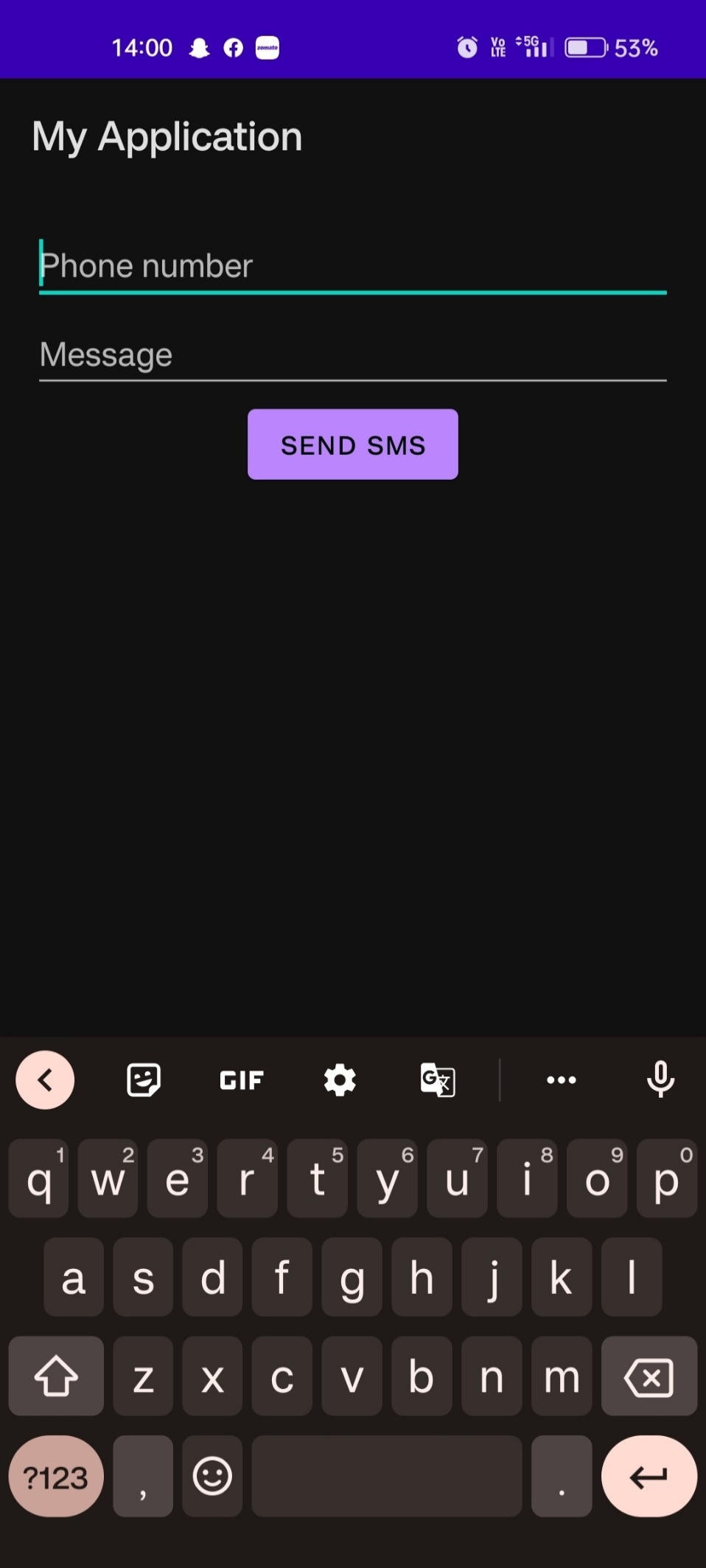
The debug dialogue box gets opened. Select the checkbox mentioning “show all processes” and then select the device below and click on “ok” and wait until the app gets launched in your connected mobile phone.



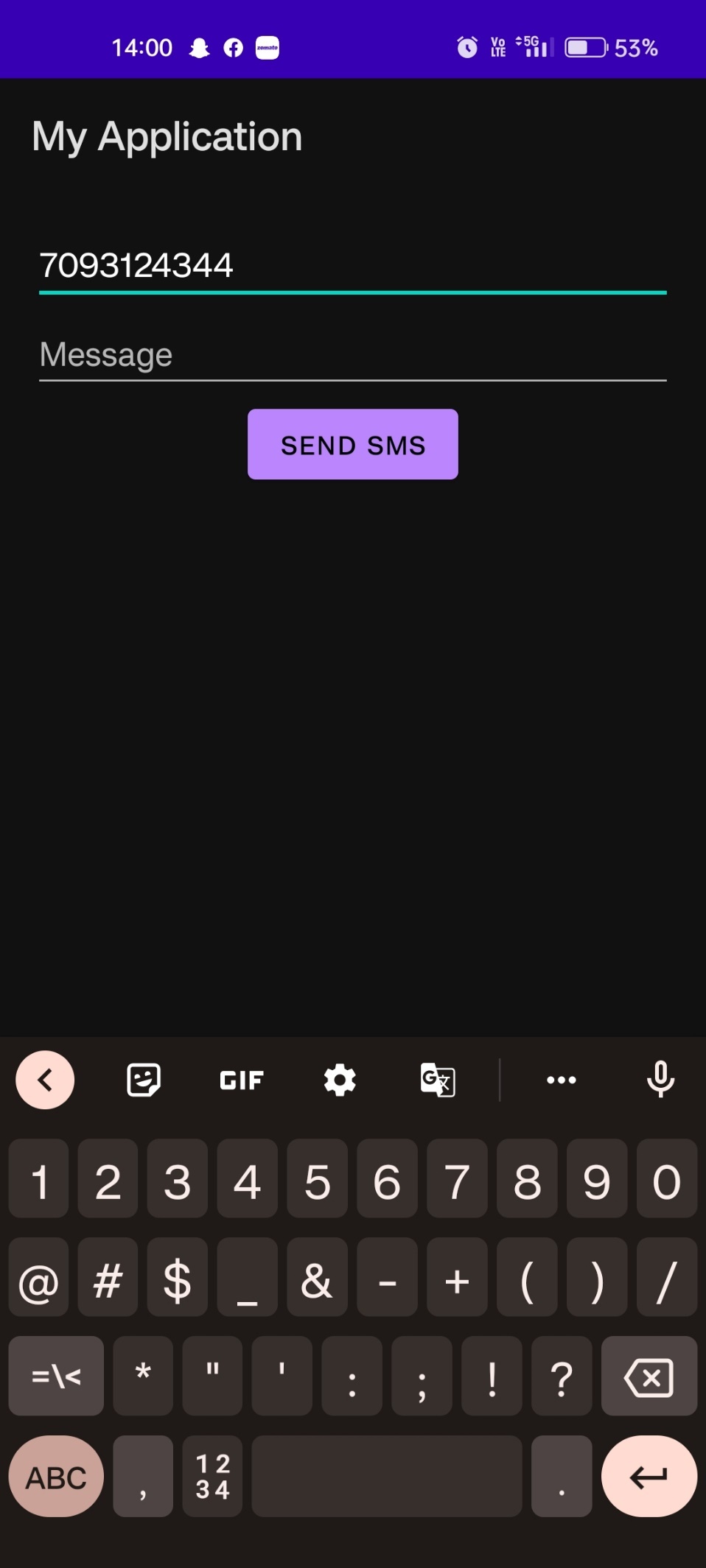
**RESULT (OUTPUT SCREEN)**

1) Open your device apps menu and click on the newly created “**MyApplication**” app.

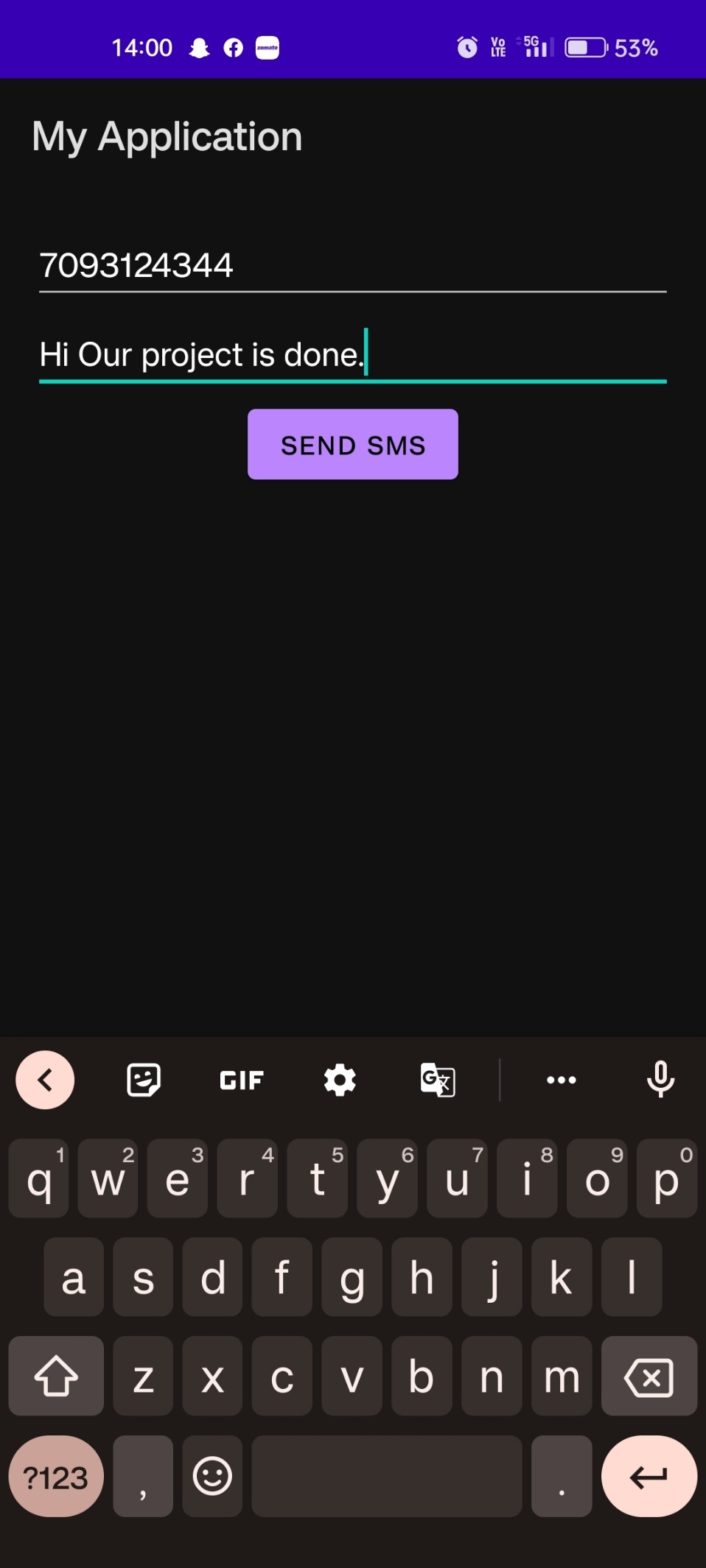
****



2) Enter the mobile number in the “phone number” input bar.



3) Enter the message you want to send to that contact in the “message” input bar.



4) Click on the “SEND SMS” button.