**ASSIGNMENT NO.3**

2124UCEF1085

* **INTRODUCTION**

To create a Phone Call Program: A program to make phone calls. For instance, to develop a contact app that allows users to select a contact and press a "Call" button to initiate a call to the selected number.

* **TOOLS & TECHNOLOGIES USED**

Android Studio Language: (JAVA) ,

Open Android Studio and create a new project,

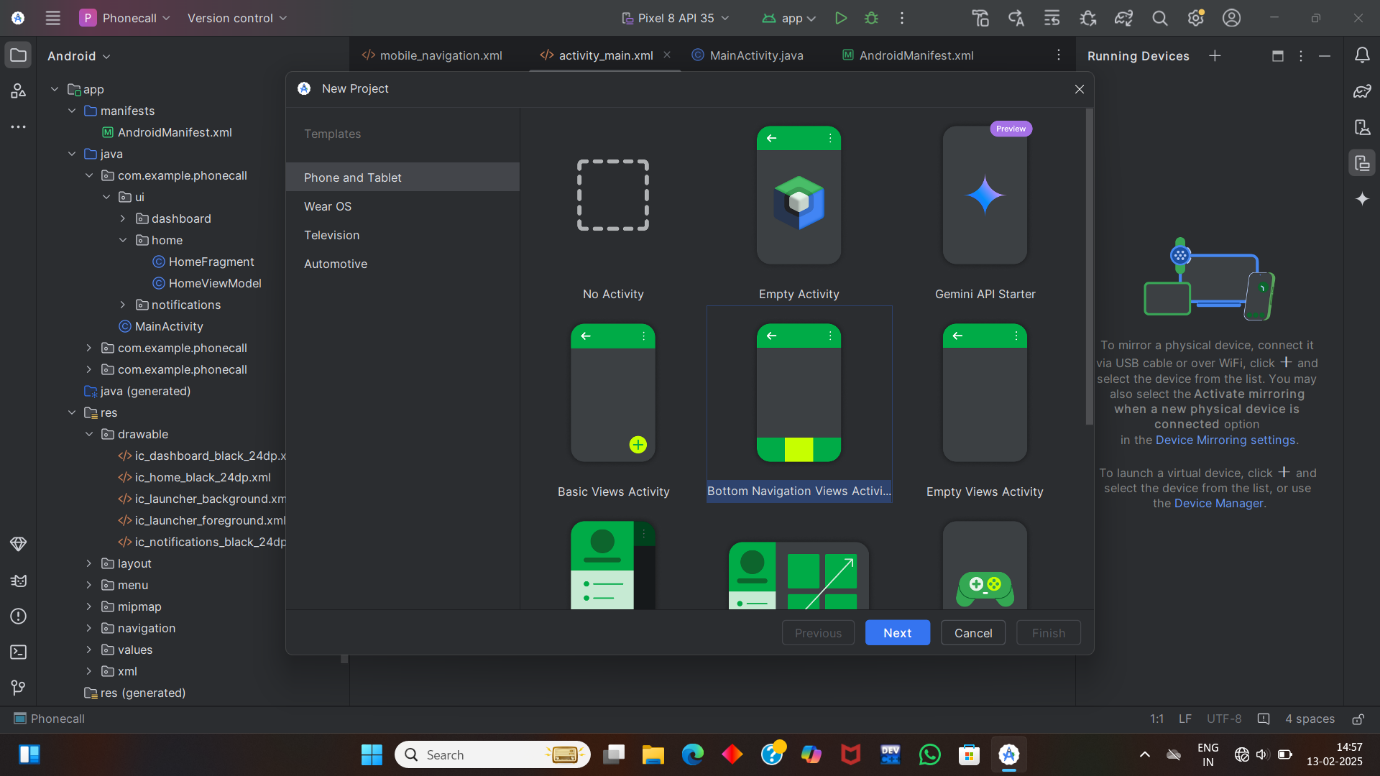
Emulator or Physical Device (Android API 35).

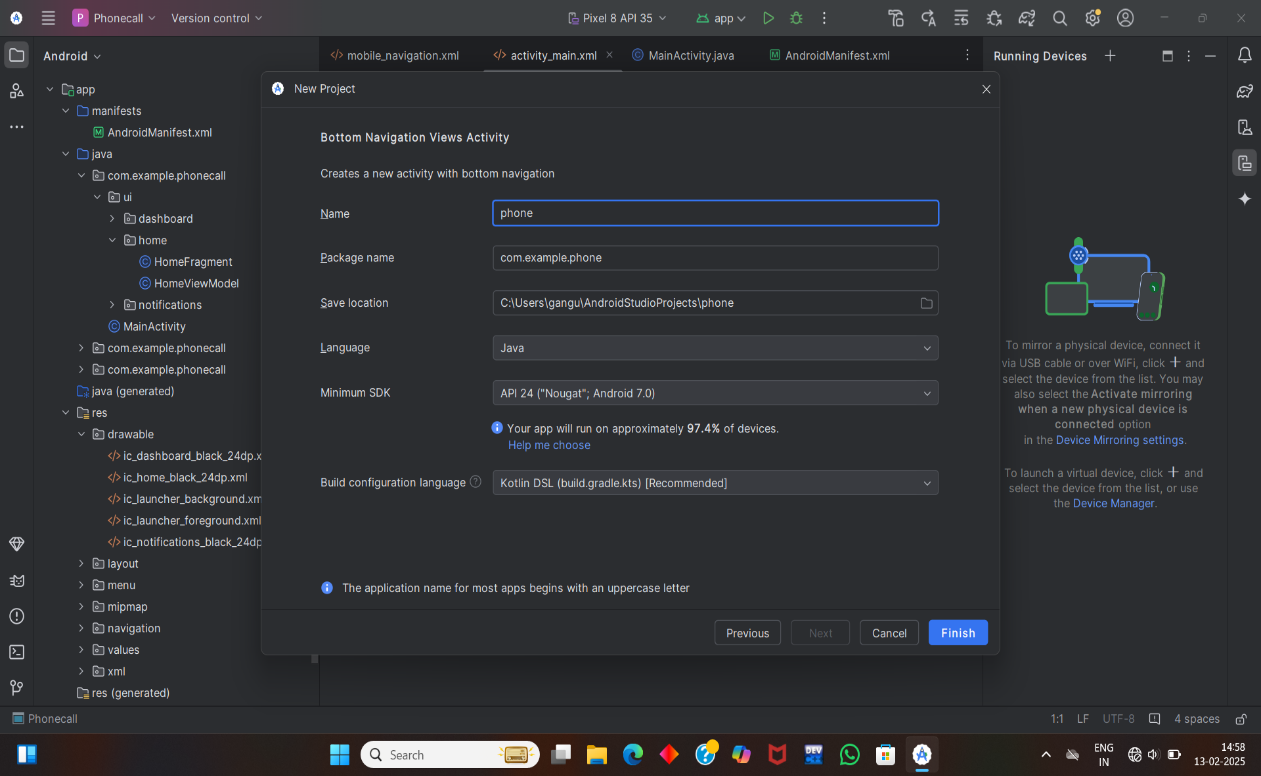
* **PROCEDURE & STEPS**

**Step:1 -Create a New Project**

-Choose Bottom Navigation Activity template.

-Set the project name and package name.

-Select the programming language(JAVA). 

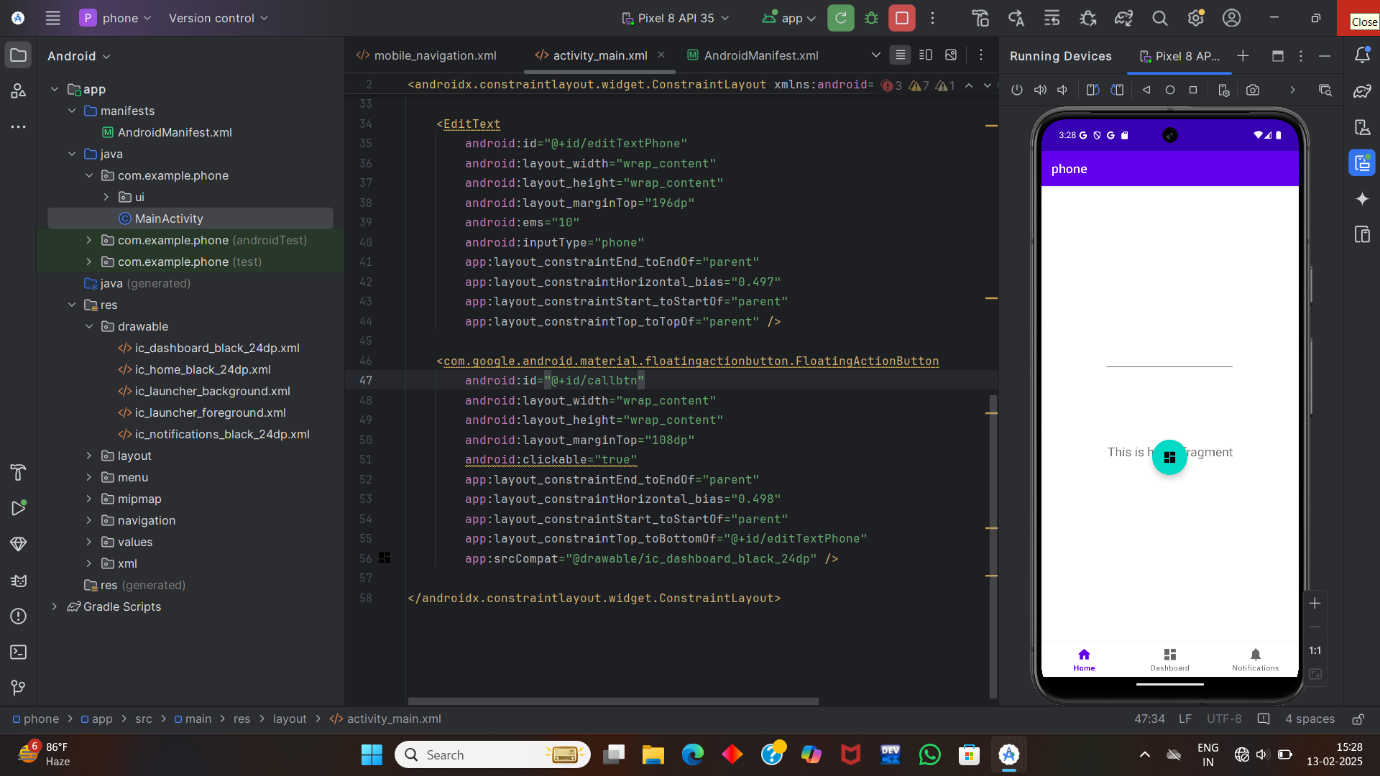


**Step 2: Designing the UI**

Open activity\_main.xml and design the layout using XML.

Add UI components such as input Field for user to input a number and a floating

Action Button which consist a icon of phone.



**Step 3: Writing the code:**

activity\_main.xml:

<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/container"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:paddingTop="?attr/actionBarSize">  
  
 <com.google.android.material.bottomnavigation.BottomNavigationView  
 android:id="@+id/nav\_view"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="0dp"  
 android:layout\_marginEnd="0dp"  
 android:background="?android:attr/windowBackground"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:menu="@menu/bottom\_nav\_menu" />  
  
 <fragment  
 android:id="@+id/nav\_host\_fragment\_activity\_main"  
 android:name="androidx.navigation.fragment.NavHostFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:defaultNavHost="true"  
 app:layout\_constraintBottom\_toTopOf="@id/nav\_view"  
 app:layout\_constraintLeft\_toLeftOf="parent"  
 app:layout\_constraintRight\_toRightOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:navGraph="@navigation/mobile\_navigation" />  
  
 <EditText  
 android:id="@+id/editTextPhone"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="168dp"  
 android:autofillHints=""  
 android:ems="10"  
 android:inputType="phone"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.497"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 tools:ignore="LabelFor" />  
  
 <com.google.android.material.floatingactionbutton.FloatingActionButton  
 android:id="@+id/callbtn"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="56dp"  
 android:clickable="true"  
 android:focusable="true"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.498"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/editTextPhone"  
 app:srcCompat="@drawable/ic\_dashboard\_black\_24dp"  
 tools:ignore="ContentDescription" />  
  
</androidx.constraintlayout.widget.ConstraintLayout<?xml version="1.0" encoding="utf-8"?>

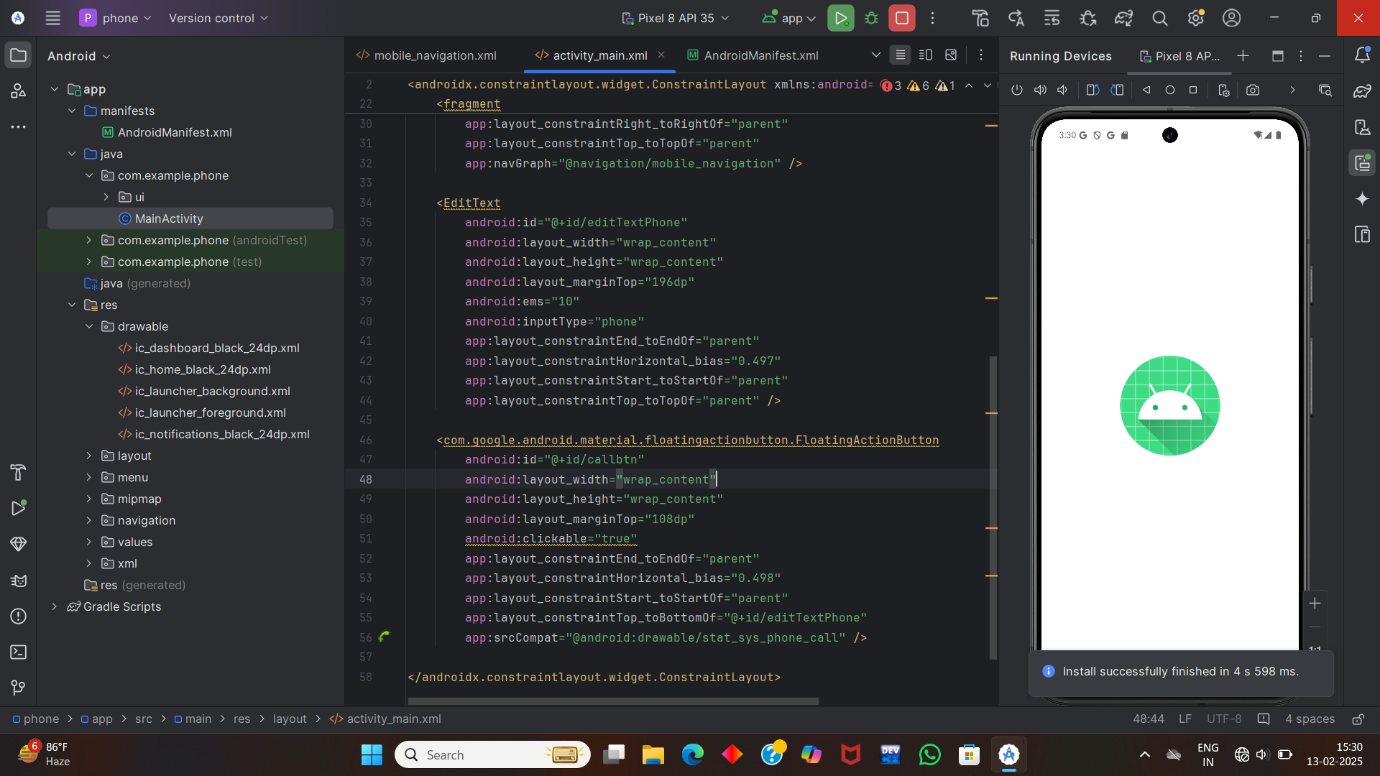
AndroidManifest.xml

<manifestxmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.phone"  
 xmlns:tools="http://schemas.android.com/tools">  
<uses-permission android:name="android.permission.CALL\_PHONE"  
 tools:ignore="PermissionImpliesUnsupportedChromeOsHardware" />  
 <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.Phone"  
 tools:targetApi="31">  
 <activity  
 android:name=".MainActivity"  
 android:exported="true"  
 android:label="@string/app\_name"  
 tools:ignore="RedundantLabel">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 <meta-data  
 android:name="preloaded\_fonts"  
 android:resource="@array/preloaded\_fonts" />  
 </application>  
  
</manifest>

MainActivity.Java

package com.example.phone;  
  
import static com.example.phone.R.id.*callbtn*;  
import static com.example.phone.R.layout.*activity\_main*;  
  
import android.annotation.SuppressLint;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.net.Uri;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import androidx.core.content.ContextCompat;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import android.Manifest;  
import com.google.android.material.floatingactionbutton.FloatingActionButton;  
  
public class MainActivity extends AppCompatActivity {  
  
 EditText phoneNo;  
 FloatingActionButton callbtn;  
 static int *PERMISSION\_CODE*=100;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 phoneNo = findViewById(R.id.*editTextPhone*);  
 callbtn = findViewById(R.id.*callbtn*);  
 if (ContextCompat.*checkSelfPermission*(MainActivity.this, Manifest.permission.*CALL\_PHONE*) != PackageManager.*PERMISSION\_GRANTED*)  
 ActivityCompat.*requestPermissions*(MainActivity.this, new String[]{Manifest.permission.*CALL\_PHONE*}, *PERMISSION\_CODE*);  
 callbtn.setOnClickListener(new View.OnClickListener(){  
 @Override  
 public void onClick(View v){  
 String phoneno=phoneNo.getText().toString();  
 Intent i=new Intent(Intent.*ACTION\_CALL*);  
 i.setData(Uri.*parse*("tel:"+phoneno));  
 startActivity(i);  
  
 }  
 });  
  
  
  
 }  
}  
  
**Step 4: Running the Application on Emulator**

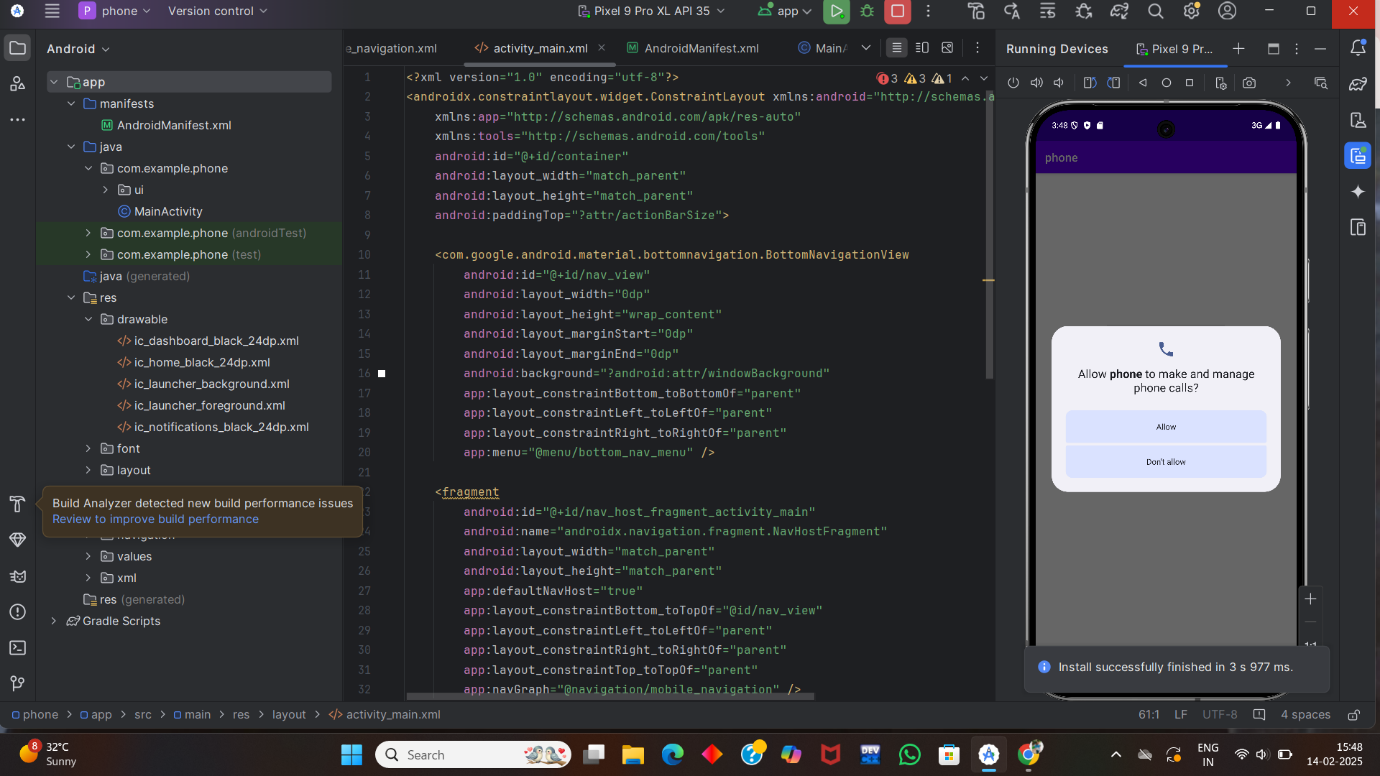
Click on the Run button in Android Studio.

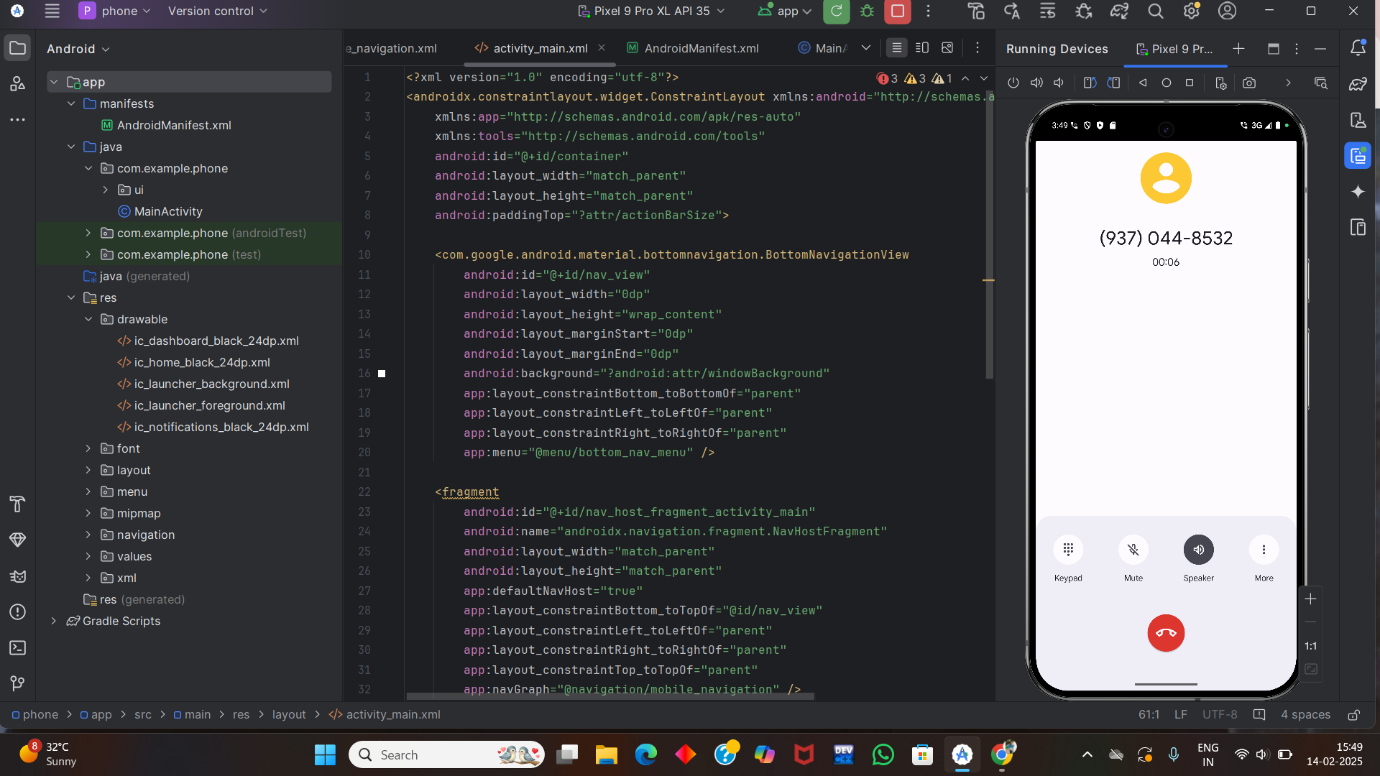
Select the emulator and launch the app. 

**Step 5: Testing & Output**

1.Test different functionalities of the app.

2.Capture the output results.





**Step 6: Conclusion**

I have Completed assignment Successfully of writing the Code for Phone call programme, I have demonstrated my understanding of user interface design principles and data registration process.