**Aim:- Programming Security and Permissions.**

**activity\_main.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:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
   
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Request Permissions at Runtime"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.21" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**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.INTERNET" />  
 <uses-permission android:name="android.permission.CAMERA"/>  
 <uses-permission android:name="android.permission.READ\_CONTACTS"/>  
 <uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>  
 <uses-permission android:name="android.permission.SEND\_SMS"/>  
 <uses-permission android:name="android.permission.READ\_CALENDAR"/>  
 <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.Securitypermission"  
 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>  
  
 <meta-data  
 android:name="android.app.lib\_name"  
 android:value="" />  
 </activity>  
 </application>  
  
</manifest>

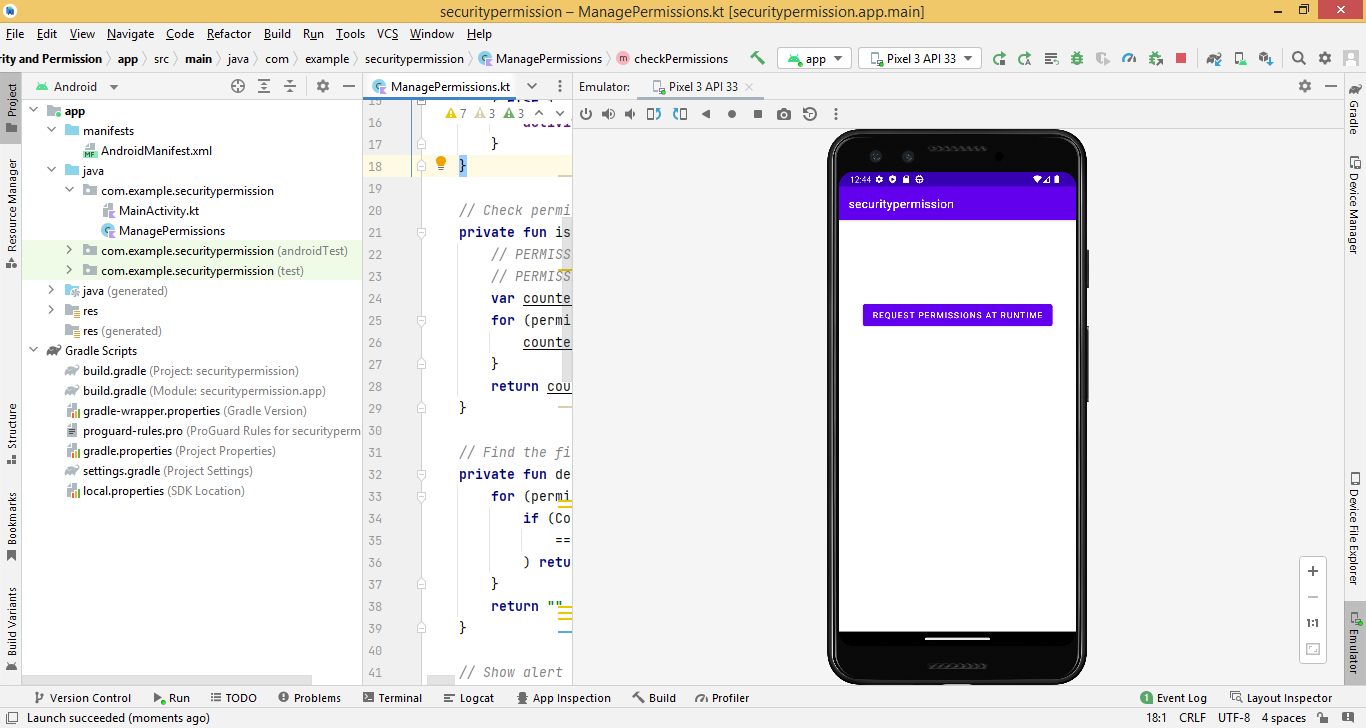
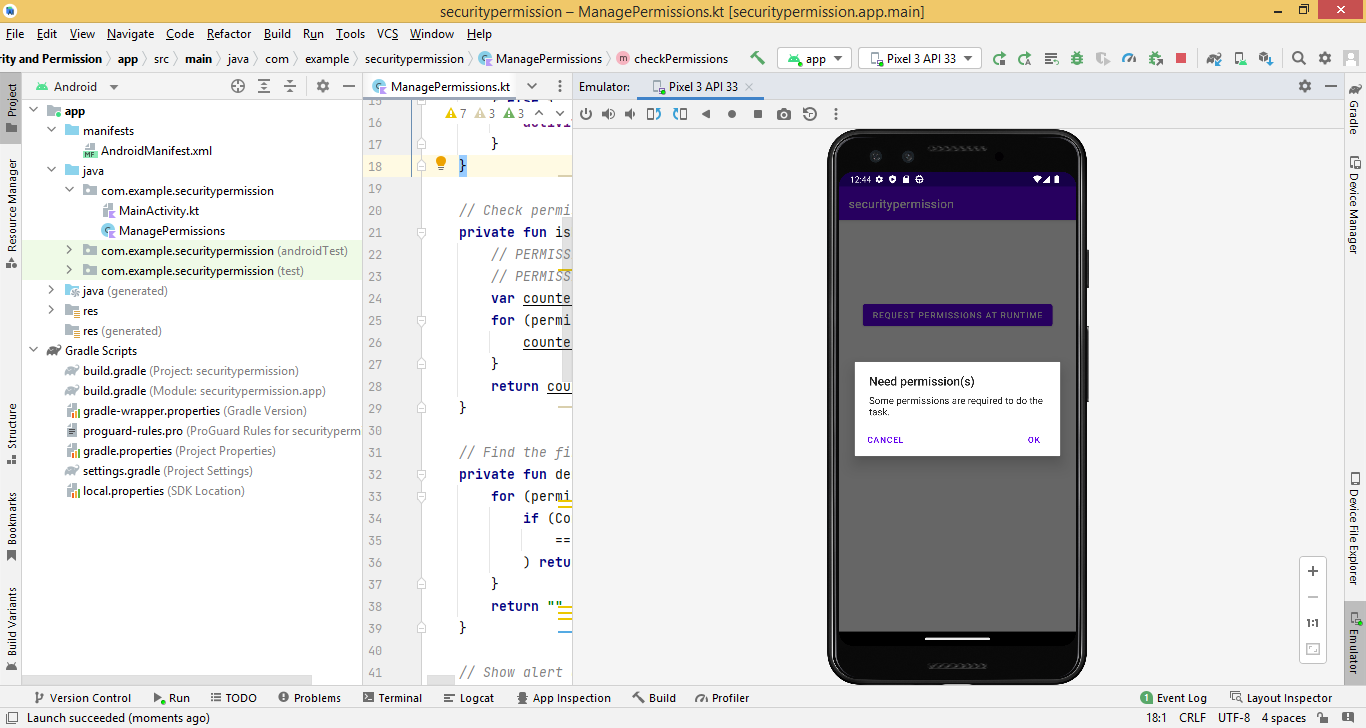
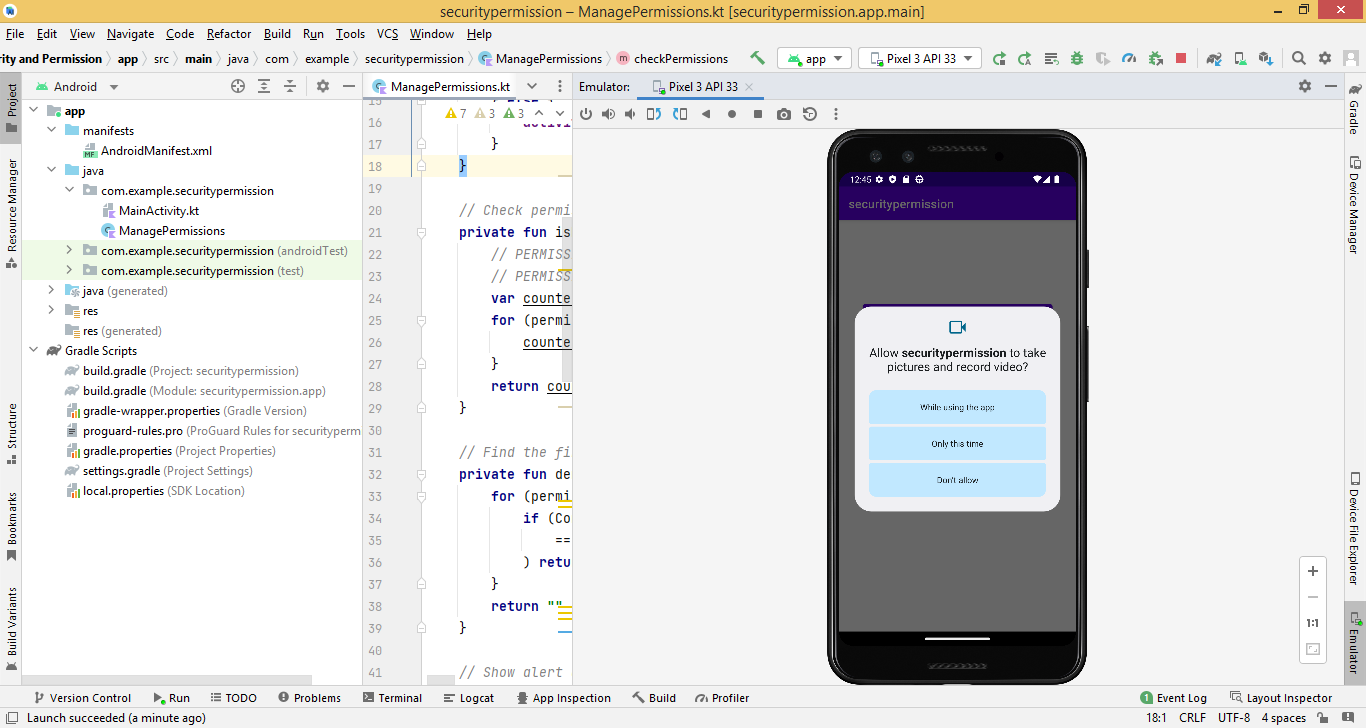
**MainActivity.kt -**

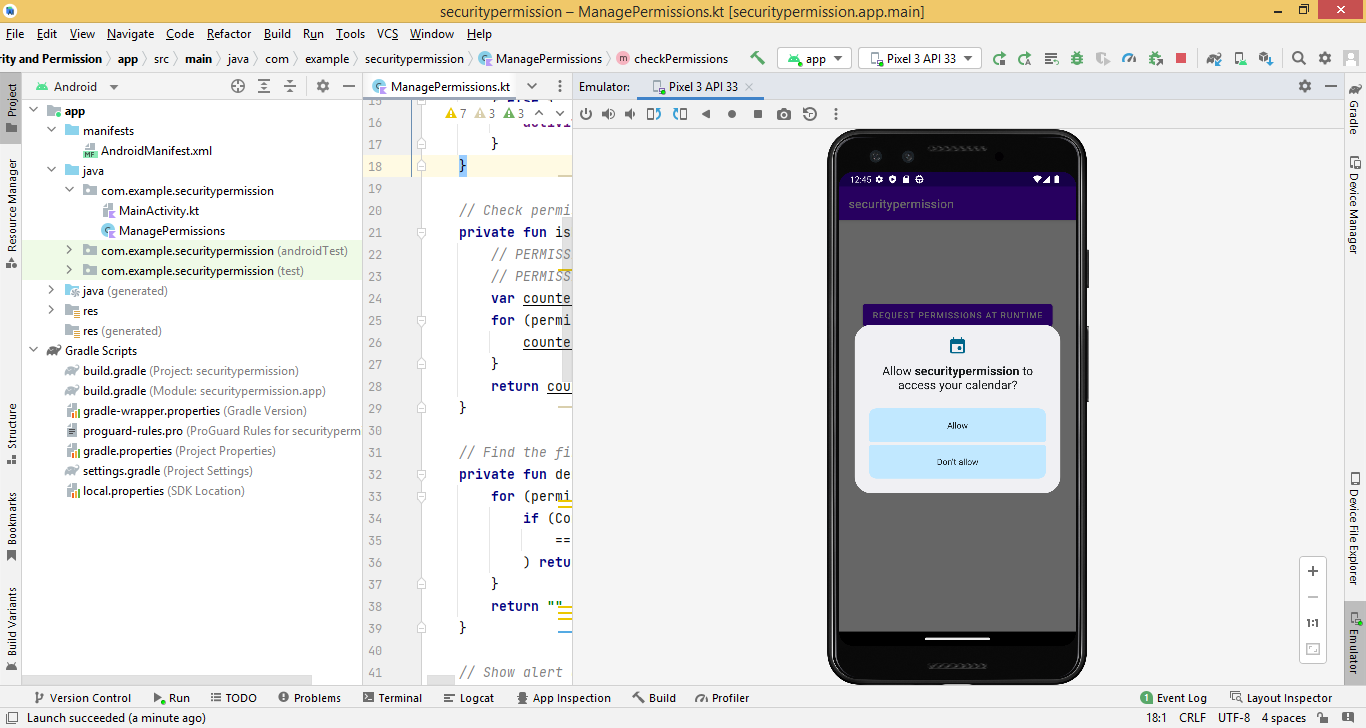
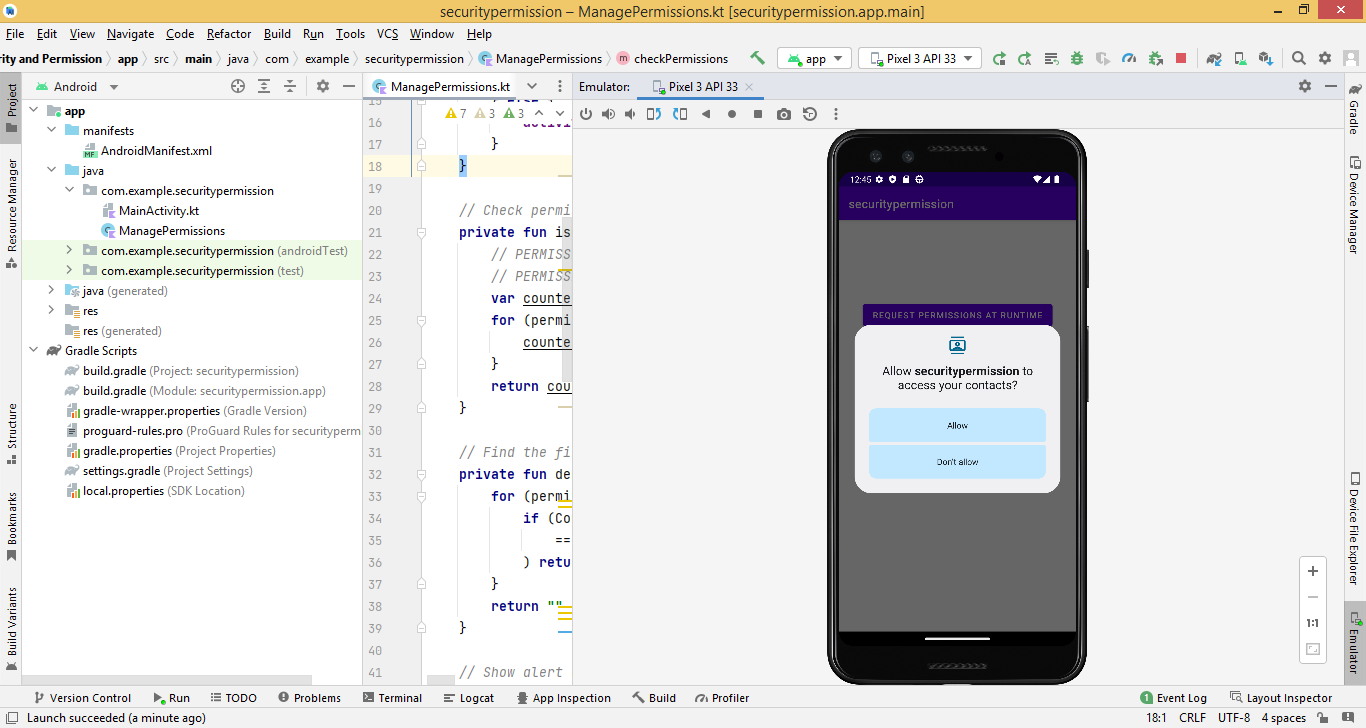
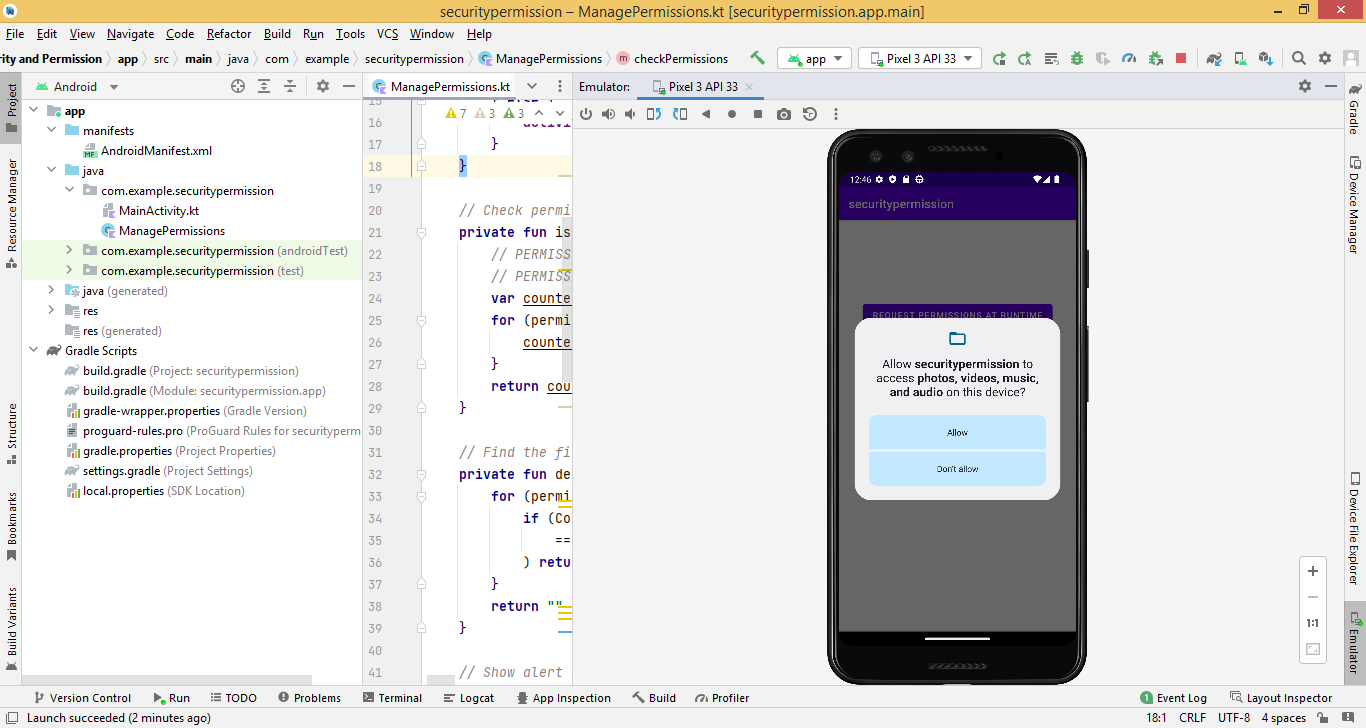
package com.example.securitypermission  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.Manifest  
import android.content.Context  
import android.os.Build  
import android.widget.Toast  
import kotlinx.android.synthetic.main.activity\_main.\*  
  
class MainActivity : AppCompatActivity() {  
 private val PermissionRequestCode = 123  
 private lateinit var managePermissions: ManagePermissions  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val list= *listOf*<String>(  
 Manifest.permission.*CAMERA*,  
 Manifest.permission.*READ\_CONTACTS*,  
 Manifest.permission.*READ\_EXTERNAL\_STORAGE*,  
 Manifest.permission.*SEND\_SMS*,  
 Manifest.permission.*READ\_CALENDAR* )  
  
 managePermissions = ManagePermissions(this,list,PermissionRequestCode)  
 button.setOnClickListener {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*)  
 managePermissions.checkPermissions()  
 }  
 }  
  
 override fun onRequestPermissionsResult(  
 requestCode: Int,  
 permissions: Array<out String>,  
 grantResults: IntArray) {  
 when(requestCode){  
 PermissionRequestCode->{  
 val isPermissionGranted = managePermissions.processPermissionResult(requestCode,permissions,grantResults)  
  
 if (isPermissionGranted)  
 {  
 Toast.makeText(this,"Permisssion Granted",Toast.*LENGTH\_SHORT*).show()  
 }  
 else  
 {  
 Toast.makeText(this, "Permission Denied", Toast.*LENGTH\_SHORT*).show()  
 }  
 return  
 }  
 }  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults)  
 }  
}  
fun Context.toast(message: String)  
{  
 Toast.makeText(this, "", Toast.*LENGTH\_SHORT*).show()  
}

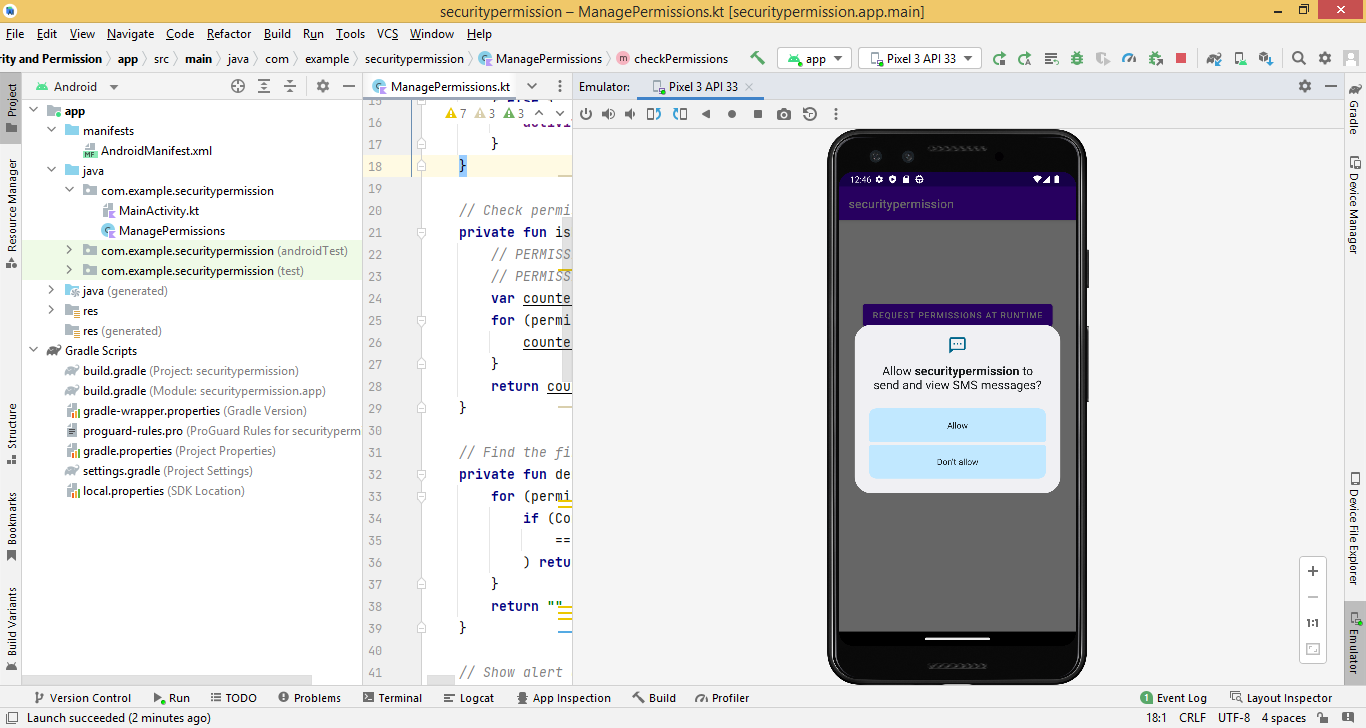
**ManagePermissions.kt –**

package com.example.securitypermission  
  
import android.app.Activity  
import android.content.pm.PackageManager  
import android.app.AlertDialog  
import androidx.core.app.ActivityCompat  
import androidx.core.content.ContextCompat  
  
class ManagePermissions(val activity: Activity,val list: List<String>,val code:Int) {  
 *// Check permissions at runtime* fun checkPermissions()   
 {  
 if (isPermissionsGranted() != PackageManager.*PERMISSION\_GRANTED*) {  
 showAlert()  
 } else {  
 activity.*toast*("Permissions already granted.")  
 }  
 }  
  
 *// Check permissions status* private fun isPermissionsGranted(): Int {  
 *// PERMISSION\_GRANTED : Constant Value: 0  
 // PERMISSION\_DENIED : Constant Value: -1* var counter = 0;  
 for (permission in list) {  
 counter += ContextCompat.checkSelfPermission(activity, permission)  
 }  
 return counter  
 }  
  
 *// Find the first denied permission* private fun deniedPermission(): String {  
 for (permission in list) {  
 if (ContextCompat.checkSelfPermission(activity, permission)  
 == PackageManager.*PERMISSION\_DENIED* ) return permission  
 }  
 return ""  
 }  
  
 *// Show alert dialog to request permissions* private fun showAlert() {  
 val builder = AlertDialog.Builder(activity)  
 builder.setTitle("Need permission(s)")  
 builder.setMessage("Some permissions are required to do the task.")  
 builder.setPositiveButton("OK", { dialog, which -> requestPermissions() })  
 builder.setNeutralButton("Cancel", null)  
 val dialog = builder.create()  
 dialog.show()  
 }  
  
 *// Request the permissions at run time* private fun requestPermissions() {  
 val permission = deniedPermission()  
 if (ActivityCompat.shouldShowRequestPermissionRationale(activity, permission)) {  
 *// Show an explanation asynchronously* activity.*toast*("Should show an explanation.")  
 } else {  
 ActivityCompat.requestPermissions(activity, list.*toTypedArray*(), code)  
 *// Process permissions result* fun processPermissionsResult(  
 requestCode: Int, permissions: Array<String>,  
 grantResults: IntArray  
 ): Boolean {  
 var result = 0  
 if (grantResults.*isNotEmpty*()) {  
 for (item in grantResults) {  
 result += item  
 }  
 }  
 if (result == PackageManager.*PERMISSION\_GRANTED*) return true  
 return false  
 }  
 }  
 }  
  
 fun processPermissionResult(  
 requestCode: Int,  
 permissions: Array<out String>,  
 grantResults: IntArray  
 ): Boolean {  
 *TODO*("Not yet implemented")  
 }  
}

**Output:**

****

****

****