create a android application to generate two notification , one notication will be non-clickable and another is clickable (it will reload the current application) android studio kotlin

**Step 1: Set Up Your Project**

1. Open Android Studio and create a new project.
2. Choose an empty activity template and Kotlin as the programming language.

**Step 2: Design the Layout (Optional)**

Since this application does not require any specific UI, you can skip designing any layout files.

**Step 3: Implement Notification Handling**

In this step, you will implement the logic for generating notifications in your application.

**MainActivity.kt**

import android.app.NotificationChannel

import android.app.NotificationManager

import android.content.Context

import android.content.Intent

import android.os.Build

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import androidx.core.app.NotificationCompat

import kotlinx.android.synthetic.main.activity\_main.\*

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

btnNonClickableNotification.setOnClickListener { showNotification("Non-Clickable", "This is a non-clickable notification.", null) }

btnClickableNotification.setOnClickListener {

val intent = packageManager.getLaunchIntentForPackage(packageName)?.apply { flags = Intent.FLAG\_ACTIVITY\_CLEAR\_TOP }

showNotification("Clickable", "Click to reload the application.", intent)

}

}

private fun showNotification(channelId: String, contentText: String, intent: Intent?) {

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

getSystemService(NotificationManager::class.java).createNotificationChannel(

NotificationChannel(channelId, channelId, NotificationManager.IMPORTANCE\_DEFAULT)

)

}

val notificationBuilder = NotificationCompat.Builder(this, channelId)

.setSmallIcon(R.drawable.ic\_notification)

.setContentTitle("Notification")

.setContentText(contentText)

.setContentIntent(intent?.let { PendingIntent.getActivity(this, 0, it, PendingIntent.FLAG\_ONE\_SHOT) })

.setAutoCancel(true)

getSystemService(NotificationManager::class.java).notify(1, notificationBuilder.build())

}

}

### Step 4: Add Notification Icon

Make sure you have a notification icon (**ic\_notification.png**) in the **res/drawable** directory of your project.

### Step 5: Test Your Application

Run your application on an Android device or emulator. Click the buttons to generate the respective notifications and verify their behavio