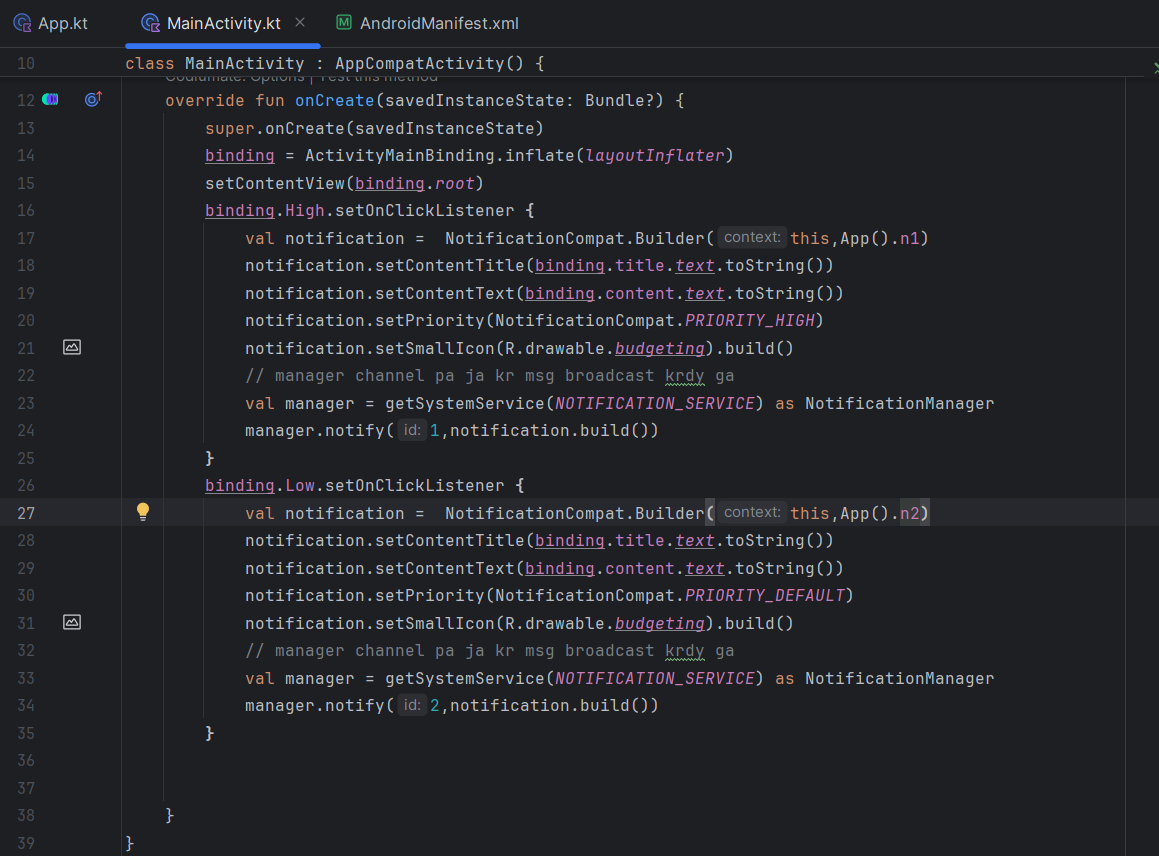
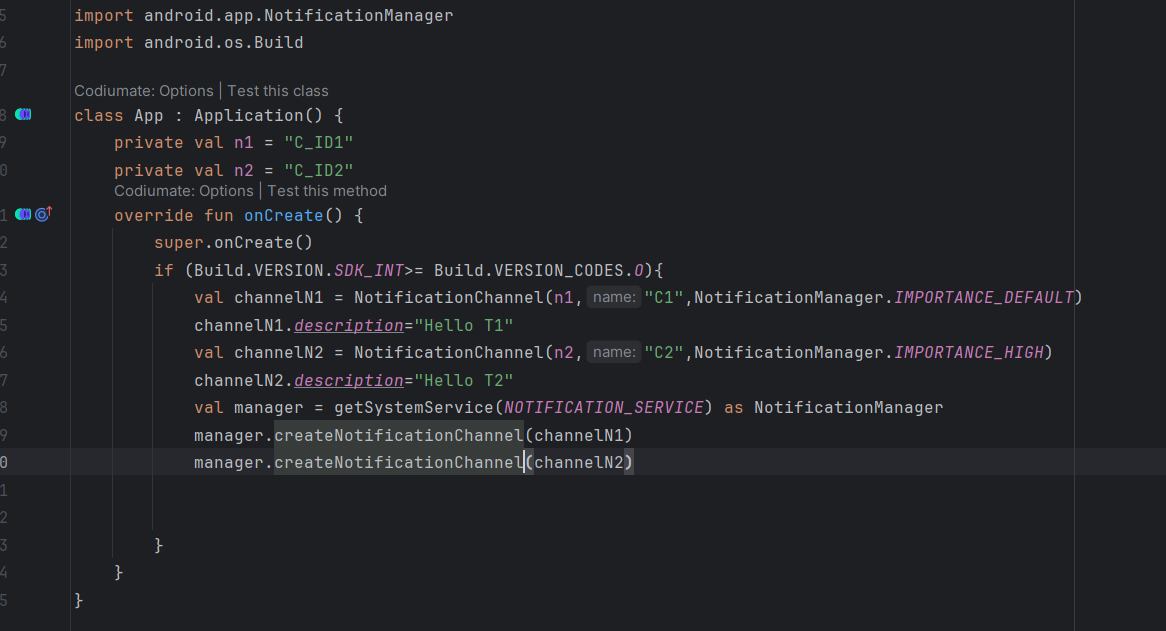
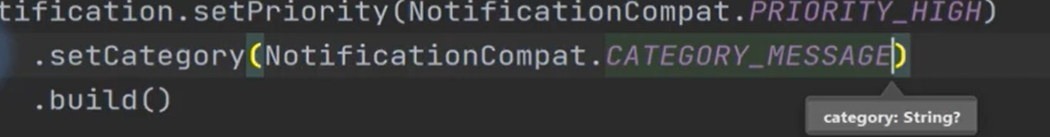
Notificaitons,

We create App: Application class notification for that purpose, we create channels from api 26 as before we work with same channel so all passed as same







package com.talhaatif.notificationimplement  
  
import android.app.NotificationManager  
import android.app.PendingIntent  
import android.content.Intent  
import android.graphics.Color  
import android.os.Bundle  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.app.NotificationCompat  
import com.talhaatif.notificationimplement.databinding.ActivityMainBinding  
  
  
class MainActivity : AppCompatActivity() {  
 private lateinit var binding: ActivityMainBinding  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 binding = ActivityMainBinding.inflate(*layoutInflater*)  
 setContentView(binding.*root*)  
 binding.High.setOnClickListener **{** val notification = NotificationCompat.Builder(this,App().n1)  
 notification.setContentTitle(binding.title.*text*.toString())  
 notification.setContentText(binding.content.*text*.toString())  
 notification.setSmallIcon(R.drawable.*budgeting*)  
 notification.setPriority(NotificationCompat.*PRIORITY\_HIGH*)  
 .setCategory(NotificationCompat.*CATEGORY\_MESSAGE*)  
 .setColor(Color.*MAGENTA*) // we can define only three actions for notifications  
 .addAction(R.drawable.*budgeting*,"back",null)  
 .addAction(R.drawable.*budgeting*,"play",null)  
 .addAction(R.drawable.*budgeting*,"next",null)  
 .setOnlyAlertOnce(true) // aik hi dfa aye bs  
 .setAutoCancel(true) // khudi cancel hujaye  
 .build()  
 // manager channel pa ja kr msg broadcast krdy ga  
 val manager = getSystemService(*NOTIFICATION\_SERVICE*) as NotificationManager  
 manager.notify(1,notification.build())  
 **}** binding.Low.setOnClickListener **{** val notification = NotificationCompat.Builder(this,App().n2)  
 notification.setContentTitle(binding.title.*text*.toString())  
 notification.setContentText(binding.content.*text*.toString())  
 notification.setSmallIcon(R.drawable.*budgeting*)  
 val intent = Intent(this,MainActivity::class.*java*)  
 // pending inntent trigger huga future m like jab notiificaiton pa click kry tb  
 // when we flag as zero tb ham data ni dal skty via notification  
 val pendingIntent = PendingIntent.getActivity(this,0,intent,0)  
 // notification pa jab click kruga tu notification gyb huga or screen open hugi yani app  
 // jab action pa clcik kruga tu notiifcation gyb ni huga or screen open b hugi  
 notification.setPriority(NotificationCompat.*PRIORITY\_DEFAULT*)  
 .setCategory(NotificationCompat.*CATEGORY\_MESSAGE*)  
 .setColor(Color.*MAGENTA*) // we can define only three actions for notifications  
 .setContentIntent(pendingIntent)  
 .addAction(R.drawable.*budgeting*,"back",pendingIntent)  
 .addAction(R.drawable.*budgeting*,"play",null)  
 .addAction(R.drawable.*budgeting*,"next",null)  
 .setOnlyAlertOnce(true) // aik hi dfa aye bs  
 .setAutoCancel(true) // khudi cancel hujaye  
 .build()  
  
 // manager channel pa ja kr msg broadcast krdy ga  
 val manager = getSystemService(*NOTIFICATION\_SERVICE*) as NotificationManager  
 manager.notify(2,notification.build())  
 **}** }  
}