# MAD MODEL LAB

### INTENT

### KOTLIN CODE:

package com.example.model\_lab

import android.annotation.SuppressLint

import android.content.Intent

import android.net.Uri

import android.os.Bundle

import android.widget.Button

import android.widget.Toast

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

class MainActivity : AppCompatActivity() {

@SuppressLint("MissingInflatedId")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

setContentView(R.layout.*activity\_main*)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->**

val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)

insets

**}**

val b1=findViewById<Button>(R.id.*button*);

val b2=findViewById<Button>(R.id.*button2*);

val b3=findViewById<Button>(R.id.*button3*);

b1.setOnClickListener**{**

val intent=Intent(Intent.*ACTION\_VIEW*);

intent.setData(Uri.parse("https://www.tce.edu/"));

startActivity(intent);

**}**

b2.setOnClickListener**{**

val intent1=Intent(this,MainActivity2::class.*java*)

startActivity(intent1)

**}**

b3.setOnClickListener**{**

Toast.makeText(this,"Vikasni is awesome",Toast.*LENGTH\_SHORT*).show();

**}**

}

}

### ACTIVITY.XML:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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:orientation="vertical"

android:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="406dp"

android:layout\_height="73dp"

android:orientation="horizontal"

tools:ignore="MissingConstraints"

tools:layout\_editor\_absoluteX="1dp"

tools:layout\_editor\_absoluteY="1dp"

>

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="IMPLICIT" />

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="EXPLICIT" />

</LinearLayout>

<LinearLayout

android:layout\_width="409dp"

android:layout\_height="655dp"

android:orientation="vertical"

tools:layout\_editor\_absoluteX="1dp"

tools:layout\_editor\_absoluteY="75dp">

<Button

android:id="@+id/button3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="TOAST MESSAGE" />

</LinearLayout>

</LinearLayout>

### MENU

### KOTLIN CODE:

package com.example.model\_lab

import android.annotation.SuppressLint

import android.os.Bundle

import android.view.\*

import android.widget.Button

import android.widget.PopupMenu

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

class MainActivity4 : AppCompatActivity() {

lateinit var textView: TextView

@SuppressLint("MissingInflatedId")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.*activity\_main4*)

val popupBtn = findViewById<Button>(R.id.*btn\_popup*)

textView = findViewById(R.id.*tv\_context*)

registerForContextMenu(textView)

popupBtn.setOnClickListener **{**

val popup = PopupMenu(this, **it**)

popup.*menuInflater*.inflate(R.menu.*pop\_menu*, popup.*menu*)

popup.setOnMenuItemClickListener **{**

Toast.makeText(this, "Vikasni on fire", Toast.*LENGTH\_LONG*).show()

true

**}**

popup.show()

**}**

}

override fun onCreateOptionsMenu(menu: Menu?): Boolean {

*menuInflater*.inflate(R.menu.*option\_menu*,menu)

return true

}

override fun onOptionsItemSelected(item: MenuItem): Boolean {

if (item.*itemId*==R.id.*opt1*){

Toast.makeText(this,"Hello there",Toast.*LENGTH\_LONG*).show()

}

return true

}

override fun onCreateContextMenu(

menu: ContextMenu?,

v: View?,

menuInfo: ContextMenu.ContextMenuInfo?

) {

super.onCreateContextMenu(menu, v, menuInfo)

*menuInflater*.inflate(R.menu.*context\_menu*, menu)

}

override fun onContextItemSelected(item: MenuItem): Boolean {

Toast.makeText(this,"Hello there",Toast.*LENGTH\_SHORT*).show()

return true

}

}

### ACTIVITY.XML:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent" android:gravity="center"

android:padding="20dp">

<Button

android:id="@+id/btn\_popup"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Show Popup Menu" />

<TextView

android:id="@+id/tv\_context"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Long press me for Context Menu"

android:padding="30dp"

android:textSize="16sp" />

</LinearLayout>

### SHARED PREFERENCE

### KOTLIN CODE:

val nameEdit = findViewById<EditText>(R.id.*editName*)

val regNoEdit = findViewById<EditText>(R.id.*editRegNo*)

val saveBtn = findViewById<Button>(R.id.*saveBtn*)

val outputText = findViewById<TextView>(R.id.*outputText*)

val pref=getSharedPreferences("MyPrefs", *MODE\_PRIVATE*);

saveBtn.setOnClickListener**{**

val name=nameEdit.*text*.toString();

val regno=regNoEdit.*text*.toString();

pref.edit().putString("name",name).putString("regno",regno).apply();

outputText.*text* = "Saved:\nName: $name\nReg No: $regno"

**}**

val name1=pref.getString("name","No name saved");

val regno1=pref.getString("regno","No regno saved");

outputText.*text* = "Saved:\nName: $name1\nReg No: $regno1";

### ACTIVITY.XML:

<EditText

android:id="@+id/editName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Name" />

<EditText

android:id="@+id/editRegNo"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter Reg No" />

<Button

android:id="@+id/saveBtn"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Save" />

<TextView

android:id="@+id/outputText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18sp"

android:paddingTop="20dp"/>

### SQLITE

### KOTLIN CODE:

package com.example.model\_lab

import android.app.AlertDialog

import android.content.ContentValues

import android.content.Context

import android.database.sqlite.SQLiteDatabase

import android.database.sqlite.SQLiteOpenHelper

import android.os.Bundle

import android.widget.Button

import android.widget.EditText

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import com.google.firestore.v1.Cursor

import java.lang.StringBuilder

class MainActivity3 : AppCompatActivity() {

lateinit var dbHelper: DBHelper

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.*activity\_main3*)

dbHelper=DBHelper(this)

val etId = findViewById<EditText>(R.id.*et\_id*)

val etName = findViewById<EditText>(R.id.*et\_name*)

val btnInsert = findViewById<Button>(R.id.*btn\_insert*)

val btnUpdate = findViewById<Button>(R.id.*btn\_update*)

val btnDelete = findViewById<Button>(R.id.*btn\_delete*)

val btnShow = findViewById<Button>(R.id.*btn\_show*)

btnInsert.setOnClickListener**{**

val id=etId.*text*.toString().*toIntOrNull*()

val name=etName.*text*.toString()

if (id!=null && name.*isNotEmpty*()){

val inserted=dbHelper.InsertData(id,name)

Toast.makeText(this,if (inserted) "Data is inserted" else "Data is not inserted",Toast.*LENGTH\_LONG*).show()

}

else{

Toast.makeText(this,"Error in inserting",Toast.*LENGTH\_LONG*).show()

}

**}**

btnUpdate.setOnClickListener**{**

val id=etId.*text*.toString().*toIntOrNull*()

val name=etName.*text*.toString()

if (id!=null && name.*isNotEmpty*()){

val updated=dbHelper.UpdateData(id,name)

Toast.makeText(this,if (updated) "Data is updated" else "Data is not updated",Toast.*LENGTH\_LONG*).show()

}

else{

Toast.makeText(this,"Error in updating",Toast.*LENGTH\_LONG*).show()

}

**}**

btnDelete.setOnClickListener**{**

val id=etId.*text*.toString().*toIntOrNull*()

if (id!=null){

val deleted=dbHelper.DeleteData(id)

Toast.makeText(this,if (deleted) "Data is deleted" else "Data is not deleted",Toast.*LENGTH\_LONG*).show()

}

else{

Toast.makeText(this,"Error in deleting",Toast.*LENGTH\_LONG*).show()

}

**}**

btnShow.setOnClickListener**{**

val cursor=dbHelper.getAllData()

val result=StringBuilder()

while (cursor.moveToNext()){

result.append("ID : ${cursor.getInt(0)} NAME: ${cursor.getString(1)}")

}

if (result.*isNotEmpty*()){

AlertDialog.Builder(this).setTitle("All data").setMessage(result.toString()).setPositiveButton("OK",**{**dialog,\_**->**dialog.dismiss()**}**).setCancelable(true).show()

}

else{

Toast.makeText(this,"Data is not found",Toast.*LENGTH\_LONG*).show()

}

**}**}

class DBHelper (context: Context):SQLiteOpenHelper(context,"MyDatabase.db",null,1){

override fun onCreate(db: SQLiteDatabase?) {

db?.execSQL("CREATE TABLE users (id INT PRIMARY KEY , name TEXT)")

}

override fun onUpgrade(db: SQLiteDatabase?, p1: Int, p2: Int){

db?.execSQL("DROP TABLE IF EXISTS users")

onCreate(db)

}

fun InsertData(id:Int,name:String):Boolean{

val db=this.*writableDatabase*

val cursor=db.rawQuery("Select \* from users where id =?", *arrayOf*(id.toString()))

if (cursor.*count*>0) return false

val contentValues=ContentValues().*apply* **{**

put("id",id)

put("name",name)

**}**

db.insert("users",null,contentValues)

return true

}

fun UpdateData(id:Int,name:String):Boolean{

val db=this.*writableDatabase*

val contentValues=ContentValues().*apply***{**

put("name",name)

**}**

val row=db.update("users",contentValues,"id=?", *arrayOf*(id.toString()))

return row>0

}

fun DeleteData(id:Int):Boolean{

val db=this.*writableDatabase*

val row=db.delete("users","id=?", *arrayOf*(id.toString()))

return row>0

}

fun getAllData(): android.database.Cursor {

val db=this.*readableDatabase*

return db.rawQuery("SELECT \* FROM users",null)

}

} }

**Firebase**  
  
package com.example.sample

import android.os.Bundle

import android.util.Log

import android.widget.Button

import android.widget.EditText

import android.widget.TextView

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import com.google.firebase.Firebase

import com.google.firebase.firestore.FieldValue

import com.google.firebase.firestore.firestore

class MainActivity : AppCompatActivity() {

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val itemNameEditText = findViewById<EditText>(R.id.name)

val age = findViewById<EditText>(R.id.age)

val contact = findViewById<EditText>(R.id.contact)

val addButton = findViewById<Button>(R.id.button)

val result = findViewById<TextView>(R.id.result)

val retrieveButton = findViewById<Button>(R.id.retrieveData)

val db = Firebase.firestore

addButton.setOnClickListener {

val itemName = itemNameEditText.text.toString().trim()

val itemAge = age.text.toString().trim()

val itemContact = contact.text.toString().trim()

if (itemName.isNotEmpty()) {

val userDocumentRef = db.collection("users").document("user")

userDocumentRef.update("items", FieldValue.arrayUnion(mapOf("name" to itemName, "age" to itemAge, "contact" to itemContact)))

.addOnSuccessListener {

Toast.makeText(this, "Item added to cart", Toast.LENGTH\_SHORT).show()

}

.addOnFailureListener { e ->

Log.e("MainActivity", "Error adding item to cart", e)

Toast.makeText(this, "Failed to add item to cart", Toast.LENGTH\_SHORT).show()

}

itemNameEditText.text.clear()

age.text.clear()

contact.text.clear()

} else {

Toast.makeText(this, "Item name cannot be empty", Toast.LENGTH\_SHORT).show()

}

}

retrieveButton.setOnClickListener {

db.collection("users").document("user").get()

.addOnSuccessListener { document ->

if (document.exists()) {

val items = document.get("items") as List<Map<String, Any>>

for (item in items) {

val itemName = item["name"] as? String

val itemAge = item["age"] as? String

val itemContact = item["contact"] as? String

result.append("Name: $itemName, Age: $itemAge, Contact: $itemContact\n")

}

}

}

}

}

}

### ACTIVITY.XML:

### SMS

### KOTLIN CODE:

val btn = findViewById<Button>(R.id.*sendBtn*)

val phone = findViewById<EditText>(R.id.*phoneEditText*)

val msg = findViewById<EditText>(R.id.*messageEditText*)

btn.setOnClickListener **{**

if (ActivityCompat.checkSelfPermission(

this,

Manifest.permission.*SEND\_SMS*

) != PackageManager.*PERMISSION\_GRANTED*

) {

ActivityCompat.requestPermissions(this, *arrayOf*(Manifest.permission.*SEND\_SMS*), 1)

} else {

SmsManager.getDefault()

.sendTextMessage(phone.*text*.toString(), null, msg.*text*.toString(), null, null)

Toast.makeText(this, "Sent sms", Toast.*LENGTH\_SHORT*).show()

}

**}**

### ACTIVITY.XML:

<EditText

android:id="@+id/phoneEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter phone number"

android:inputType="phone" />

<EditText

android:id="@+id/messageEditText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Enter message"

android:inputType="textMultiLine"

android:minLines="3"

android:gravity="top" />

<Button

android:id="@+id/sendBtn"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Send SMS"

android:layout\_marginTop="16dp" />

### NOTIFICATION

private val channel\_id="Simple\_channel\_id";

@SuppressLint("MissingInflatedId")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

setContentView(R.layout.*activity\_main2*)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->**

val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)

insets

**}**

val notifybtn=findViewById<Button>(R.id.*notifyBtn*)

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

val name="Simple notification"

val descriptionText="Hi vikasni"

val importance=NotificationManager.*IMPORTANCE\_DEFAULT*

val channel=NotificationChannel(channel\_id,name,importance).*apply***{**

*description*=descriptionText

**}**

val notificationManager:NotificationManager=getSystemService(Context.*NOTIFICATION\_SERVICE*) as NotificationManager

notificationManager.createNotificationChannel(channel)}

notifybtn.setOnClickListener**{**

val intent= Intent(this,MainActivity::class.*java*)

val pendingIntent=PendingIntent.getActivity(this,0,intent,PendingIntent.*FLAG\_IMMUTABLE*)

val builder1=NotificationCompat.Builder(this,channel\_id).setSmallIcon(android.R.drawable.*ic\_dialog\_info*).setAutoCancel(true).setContentTitle("Simple Notification").setContentText("Hey Vikasni").setColorized(true).setContentIntent(pendingIntent);

*with*(NotificationManagerCompat.from(this))**{**

if (ActivityCompat.checkSelfPermission(

this@MainActivity2,

Manifest.permission.*POST\_NOTIFICATIONS*

) != PackageManager.*PERMISSION\_GRANTED*

) {

// *TODO: Consider calling*

// *ActivityCompat#requestPermissions*

// here to request the missing permissions, and then overriding

// public void onRequestPermissionsResult(int requestCode, String[] permissions,

// int[] grantResults)

// to handle the case where the user grants the permission. See the documentation

// for ActivityCompat#requestPermissions for more details.

return@setOnClickListener

}

notify(1,builder1.build())

**}**

**}**

### LOCATION SERVICE AND GEOCODING

### KOTLIN CODE:

fusedLPC=LocationServices.getFusedLocationProviderClient(this)

locationText = findViewById(R.id.*locationText*)

btn1 = findViewById(R.id.*getLocationBtn*)

btn1.setOnClickListener **{**

getYourCurrentLocation()

**}** }

private fun getYourCurrentLocation() {

if ((ActivityCompat.checkSelfPermission(this,Manifest.permission.*ACCESS\_FINE\_LOCATION*)!=PackageManager.*PERMISSION\_GRANTED*) && (ActivityCompat.checkSelfPermission(this,Manifest.permission.*ACCESS\_COARSE\_LOCATION*)!=PackageManager.*PERMISSION\_GRANTED*)){

ActivityCompat.requestPermissions(this, *arrayOf*(Manifest.permission.*ACCESS\_FINE\_LOCATION*),909)

return

}

val task=fusedLPC.getLastLocation();

task.addOnSuccessListener **{**

if (**it**!=null){

val lat=**it**.*latitude*;

val lon=**it**.*longitude*;

try{

val geocoder=Geocoder(this,Locale.getDefault())

val addressList=geocoder.getFromLocation(lat,lon,1)

val address= if(!addressList.*isNullOrEmpty*()) addressList[0]?.getAddressLine(0)

else "Address not found"

locationText.*text*="$address,lat:$lat,lon:$lon"

}

catch (e:Exception){

Toast.makeText(this,"Can't find the location",Toast.*LENGTH\_LONG*).show()

}

}

else{

Toast.makeText(this,"Location is not found",Toast.*LENGTH\_LONG*).show()

}

**}**

### ACTIVITY.XML:

<Button

android:id="@+id/getLocationBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Get Location" />

<TextView

android:id="@+id/locationText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Location will appear here"

android:paddingTop="20dp"

android:textSize="16sp"/>

### DIALOGUE, PROGRESS, DATE PICKER, TIME PICKER:

### KOTLIN CODE:

package com.example.model\_lab

import android.annotation.SuppressLint

import android.app.DatePickerDialog

import android.app.TimePickerDialog

import android.os.Bundle

import android.os.Handler

import android.os.Looper

import android.widget.Button

import android.widget.ProgressBar

import android.widget.Toast

import androidx.activity.enableEdgeToEdge

import androidx.appcompat.app.AlertDialog

import androidx.appcompat.app.AppCompatActivity

import androidx.core.view.ViewCompat

import androidx.core.view.WindowInsetsCompat

import java.util.Calendar

class MainActivity2 : AppCompatActivity() {

val handler=Handler(Looper.getMainLooper());

@SuppressLint("MissingInflatedId")

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

*enableEdgeToEdge*()

setContentView(R.layout.*activity\_main2*)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.*main*)) **{** v, insets **->**

val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())

v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)

insets

**}**

val dateBtn = findViewById<Button>(R.id.*datePickerBtn*)

val timeBtn = findViewById<Button>(R.id.*timePickerBtn*)

dateBtn.setOnClickListener **{**

val calender=Calendar.getInstance()

val year=calender.get(Calendar.*YEAR*)

val month=calender.get(Calendar.*MONTH*)

val day=calender.get(Calendar.*DAY\_OF\_MONTH*)

DatePickerDialog(this,**{**\_,y,m,d**->**

Toast.makeText(this,"Date is : $d/${m+1}/$y",Toast.*LENGTH\_SHORT*).show()

**}**,year,month,day).show()

**}**

timeBtn.setOnClickListener **{**

val calendar=Calendar.getInstance()

val hour=calendar.get(Calendar.*HOUR\_OF\_DAY*)

val min=calendar.get(Calendar.*MINUTE*)

TimePickerDialog(this,**{**\_,h,m**->**

Toast.makeText(this,"TIME IS : $h:$m",Toast.*LENGTH\_SHORT*).show()**}**,hour,min,true).show()

**}**

val b1=findViewById<Button>(R.id.*button4*);

b1.setOnClickListener**{**

incrementProgress();

**}**

val b2=findViewById<Button>(R.id.*button5*);

b2.setOnClickListener**{**

showDialogAlert()

**}**

}

private fun showDialogAlert() {

val builder=AlertDialog.Builder(this);

builder.setTitle("DO YOU WANT TO PASS?").setMessage("I THINK IT IS DIFFICULT ....").setPositiveButton("YES")**{**dialog,\_**->**dialog.dismiss()**}**.setNegativeButton("NO")**{**dialog,\_**->**dialog.dismiss()**}**.show();

}

private fun incrementProgress() {

val prog=findViewById<ProgressBar>(R.id.*progressBar*);

Thread**{**

for (i in 1..100){

Thread.sleep(50);

handler.post**{**prog.*progress*=i;**}**

}

**}**.start()

}

}

### ACTIVITY.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:id="@+id/main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity2">

<LinearLayout

android:layout\_width="409dp"

android:layout\_height="729dp"

android:orientation="vertical"

tools:layout\_editor\_absoluteX="1dp"

tools:layout\_editor\_absoluteY="1dp"

tools:ignore="MissingConstraints">

<ProgressBar

android:id="@+id/progressBar"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="200dp"

android:layout\_gravity="center"

android:layout\_marginTop="20dp"

android:layout\_height="wrap\_content" />

<Button

android:id="@+id/button4"

android:layout\_width="150dp"

android:layout\_gravity="center"

android:layout\_marginTop="40dp"

android:layout\_height="wrap\_content"

android:text="progress" />

<Button

android:id="@+id/button5"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="40dp"

android:layout\_gravity="center"

android:text="dialogue" />

<Button

android:id="@+id/datePickerBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Pick a Date"

android:layout\_gravity="center"

android:layout\_marginTop="30dp" />

<Button

android:id="@+id/timePickerBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Pick a Time"

android:layout\_gravity="center"

android:layout\_marginTop="20dp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>