activity_main:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical"
    android:padding="25dp"
    android:background="@color/background">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="What is the weather like for you"
        android:textColor="@color/textColor"
        android:textSize="30sp"
        android:gravity="center"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btVar1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter"
        android:background="@drawable/gradient_btn"
        android:layout_gravity="center"
        android:layout_margin="50dp"/>

</LinearLayout>
```

MainActivity:

```
package com.example.lessontenapp

import android.Manifest
import android.annotation.SuppressLint
import android.content.pm.PackageManager
import android.location.Location
import android.os.Bundle
import android.widget.Button
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.view.ViewCompat
```

```
import androidx.core.view.WindowInsetsCompat
import com.android.volley.Request
import com.android.volley.toolbox.StringRequest
import com.android.volley.toolbox.Volley
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationServices
import org.json.JSONException
import org.json.JSONObject



class MainActivity : AppCompatActivity() {
    var api_key = "7a5106c25a048f3bffc21bd3cdca508c"

    private lateinit var btVar1: Button
    private lateinit var textView: TextView

    private lateinit var fusedLocationClient: FusedLocationProviderClient

    private val LOCATION_PERMISSION_REQUEST_CODE = 1

    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
        }

        textView = findViewById(R.id.textView)
        btVar1 = findViewById(R.id.btVar1)

        fusedLocationClient =
            LocationServices.getFusedLocationProviderClient(this)

        btVar1.setOnClickListener {
            checkForPermission()
        }
    }

    private fun checkForPermission() {
        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,
                    Manifest.permission.ACCESS_COARSE_LOCATION),
                LOCATION_PERMISSION_REQUEST_CODE)
        } else {
            obtainLocation()
        }
    }

    override fun onRequestPermissionsResult(requestCode: Int, permissions:
    Array<out String>, grantResults: IntArray) {
        super.onRequestPermissionsResult(requestCode, permissions,
            grantResults)
        if (requestCode == LOCATION_PERMISSION_REQUEST_CODE) {
            if (grantResults.isNotEmpty() && grantResults[0] ==
                PackageManager.PERMISSION_GRANTED) {
                obtainLocation()
            } else {
                Toast.makeText(this, "Разрешение отклонено",
                    Toast.LENGTH_SHORT).show()
            }
        }
    }

    @SuppressLint("MissingPermission")
    private fun obtainLocation() {
        fusedLocationClient.lastLocation
            .addOnSuccessListener { location: Location? ->
                if (location != null) {
                    val weatherUrl =

"https://api.openweathermap.org/data/2.5/weather?lat=${location.latitude}&lon=${lo
cation.longitude}&units=metric&appid=${api_key}"
                    getTemp(weatherUrl)
                } else {
                    Toast.makeText(this, "Не удалось получить местоположение",
Toast.LENGTH_SHORT).show()
                }
            }
            .addOnFailureListener { exception ->
                Toast.makeText(this, "Location Permission not granted",
                    Toast.LENGTH_SHORT).show()
            }
    }

    private fun getTemp(url: String) {
```

```
        val queue = Volley.newRequestQueue(this)

        val stringReq = StringRequest(
            Request.Method.GET, url, { response ->
                try {
                    val obj = JSONObject(response)

                    val main: JSONObject = obj.getJSONObject("main")
                    val temperature = main.getString("temp")
                    val humidity = main.getString("humidity") // Получаем
влажность

                    val city = obj.getString("name")

                    textView.text = "${temperature} °C in ${city},
humidity:${humidity}%"
                } catch (e: JSONException) {
                    textView.text = "Data parsing error!"
                }
            },
            { textView.text = "Request error!" }
        )

        queue.add(stringReq)
    }
}
```

colors:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="textColor">#00ADB5</color>
    <color name="background">#222831</color>
</resources>
```

AndroidManifest:

```
<?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.ACCESS_COARSE_LOCATION"/>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <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.LessonTenApp"
        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>
        </activity>
    </application>

</manifest>
```

build.gradle:

```
plugins {
    alias(libs.plugins.android.application)
    alias(libs.plugins.kotlin.android)
}

android {
    namespace = "com.example.lessontenapp"
    compileSdk = 35

    defaultConfig {
        applicationId = "com.example.lessontenapp"
        minSdk = 28
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
```

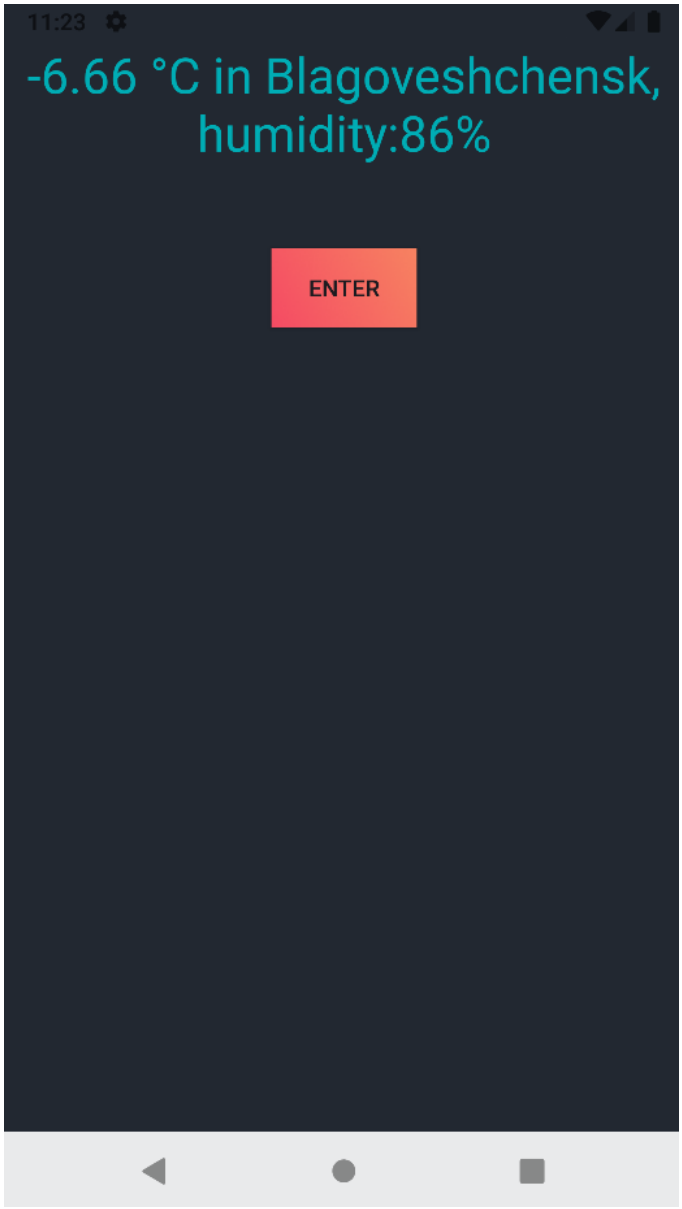
```
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
    kotlinOptions {
        jvmTarget = "11"
    }
}

dependencies {

    implementation(libs.androidx.core.ktx)
    implementation(libs.androidx.appcompat)
    implementation(libs.material)
    implementation(libs.androidx.activity)
    implementation(libs.androidx.constraintlayout)
    implementation(libs.volley)
    implementation(libs.androidx.tools.core)
    implementation(libs.play.services.location)

    testImplementation(libs.junit)
    androidTestImplementation(libs.androidx.junit)
    androidTestImplementation(libs.androidx.espresso.core)
}
```

result:


11:23
-6.66 °C in Blagoveshchensk, humidity:86%
ENTER