**App 10: Develop an Android app to display the addresses present around current location**

**MainActivity.java:**

package com.example.currentlocation;

import android.Manifest;

import android.annotation.SuppressLint;

import android.content.Context;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Address;

import android.location.Geocoder;

import android.location.Location;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.Looper;

import android.provider.Settings;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import com.google.android.gms.location.FusedLocationProviderClient;

import com.google.android.gms.location.LocationCallback;

import com.google.android.gms.location.LocationRequest;

import com.google.android.gms.location.LocationResult;

import com.google.android.gms.location.LocationServices;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import java.io.IOException;

import java.util.List;

import java.util.Locale;

public class MainActivity extends AppCompatActivity {

FusedLocationProviderClient mFusedLocationClient;

TextView latitudeTextView, longTextView, country, locality, address;

int PERMISSION\_ID = 44;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

latitudeTextView = findViewById(R.id.latTextView);

longTextView = findViewById(R.id.lonTextView);

country = findViewById(R.id.country);

locality = findViewById(R.id.locality);

address = findViewById(R.id.address);

mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

getLastLocation();

}

@SuppressLint("MissingPermission")

private void getLastLocation() {

if (checkPermissions()) {

if (isLocationEnabled()) {

mFusedLocationClient.getLastLocation().addOnCompleteListener(new OnCompleteListener<Location>() {

@Override

public void onComplete(@NonNull Task<Location> task) {

Location location = task.getResult();

if (location == null) {

requestNewLocationData();

} else {

Locale l = getResources().getConfiguration().getLocales().get(0);

Geocoder geocoder = new Geocoder(MainActivity.this, l);

try {

List<Address> addresses = geocoder.getFromLocation(location.getLatitude(), location.getLongitude(), 1);

latitudeTextView.setText(location.getLatitude() + "");

longTextView.setText(location.getLongitude() + "");

country.setText(addresses.get(0).getLocale().getDisplayCountry());

locality.setText(addresses.get(0).getLocality());

address.setText(addresses.get(0).getAddressLine(0));

} catch (IOException e) {

e.printStackTrace();

}

}

}

});

} else {

Toast.makeText(this, "Please turn on" + " your location...", Toast.LENGTH\_LONG).show();

Intent intent = new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS);

startActivity(intent);

}

}

else {

requestPermissions();

}

}

@SuppressLint("MissingPermission")

private void requestNewLocationData() {

LocationRequest mLocationRequest = new LocationRequest();

mLocationRequest.setPriority(LocationRequest.PRIORITY\_HIGH\_ACCURACY);

mLocationRequest.setInterval(5);

mLocationRequest.setFastestInterval(0);

mLocationRequest.setNumUpdates(1);

mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);

mFusedLocationClient.requestLocationUpdates(mLocationRequest, mLocationCallback, Looper.myLooper());

}

private LocationCallback mLocationCallback = new LocationCallback() {

@Override

public void onLocationResult(LocationResult locationResult) {

Location mLastLocation = locationResult.getLastLocation();

latitudeTextView.setText("Latitude: " + mLastLocation.getLatitude() + "");

longTextView.setText("Longitude: " + mLastLocation.getLongitude() + "");

}

};

// method to check for permissions

private boolean checkPermissions() {

return ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_COARSE\_LOCATION) == PackageManager.PERMISSION\_GRANTED && ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED;

}

private void requestPermissions() {

ActivityCompat.requestPermissions(this, new String[]{

Manifest.permission.ACCESS\_COARSE\_LOCATION,

Manifest.permission.ACCESS\_FINE\_LOCATION}, PERMISSION\_ID);

}

private boolean isLocationEnabled() {

LocationManager locationManager = (LocationManager) getSystemService(Context.LOCATION\_SERVICE);

return locationManager.isProviderEnabled(LocationManager.GPS\_PROVIDER) || locationManager.isProviderEnabled(LocationManager.NETWORK\_PROVIDER);

}

@Override

public void

onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == PERMISSION\_ID) {

if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

getLastLocation();

}

}

}

@Override

public void onResume() {

super.onResume();

if (checkPermissions()) {

getLastLocation();

}

}

}

**Activity\_main.xml:**

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

<LinearLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:gravity="center"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-black"

android:text="Latitude:"

android:textColor="@color/black"

android:textSize="20dp" />

<TextView

android:id="@+id/latTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Latitude will be here! "

android:textColor="@color/black"

android:textSize="16dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-black"

android:text="Longitude:"

android:textSize="20dp"

android:textColor="@color/black"/>

<TextView

android:id="@+id/lonTextView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Longitude will be here! "

android:textColor="@color/black"

android:textSize="16dp"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-black"

android:text="Country:"

android:textSize="20dp"

android:textColor="@color/black"/>

<TextView

android:id="@+id/country"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Country will be here! "

android:textColor="@color/black"

android:textSize="16dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-black"

android:text="Locality:"

android:textSize="20dp"

android:textColor="@color/black"/>

<TextView

android:id="@+id/locality"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Locality will be here! "

android:textColor="@color/black"

android:textSize="16dp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="sans-serif-black"

android:text="Address:"

android:textSize="20dp"

android:textColor="@color/black"/>

<TextView

android:id="@+id/address"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="address will be here! "

android:textColor="@color/black"

android:textSize="16dp" />

</LinearLayout>

**OUTPUT:**

