activity\_main.xml

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

<RelativeLayout 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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/location\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:padding="16dp"

android:text="Click the button to get your location"

android:textAlignment="center"

android:textSize="20sp" />

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/fab\_get\_location"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentEnd="true"

android:layout\_alignParentBottom="true"

android:layout\_margin="24dp"

android:src="@android:drawable/ic\_menu\_mylocation"

android:contentDescription="Get Location" />

</RelativeLayout>

Add to AndroidManifest.xml:

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />

<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />

MainActivity.java

package com.example.gpslocationapp;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.pm.PackageManager;

import android.location.Location;

import android.os.Bundle;

import android.os.Looper;

import android.view.View;

import android.widget.TextView;

import android.widget.Toast;

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.location.Priority;

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

import com.google.android.material.floatingactionbutton.FloatingActionButton;

public class MainActivity extends AppCompatActivity {

private static final int LOCATION\_PERMISSION\_REQUEST\_CODE = 1;

private FusedLocationProviderClient fusedLocationProviderClient;

private LocationCallback locationCallback;

private TextView locationTextView;

private FloatingActionButton getLocationButton;

private boolean isRequestingLocation = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

locationTextView = findViewById(R.id.location\_text);

getLocationButton = findViewById(R.id.fab\_get\_location);

fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);

// Create location callback to handle location updates

locationCallback = new LocationCallback() {

@Override

public void onLocationResult(@NonNull LocationResult locationResult) {

if (locationResult == null) {

return;

}

// Get the most recent location

Location location = locationResult.getLastLocation();

if (location != null) {

double latitude = location.getLatitude();

double longitude = location.getLongitude();

String locationString = "Latitude: " + latitude + "\nLongitude: " + longitude;

locationTextView.setText(locationString);

// Stop location updates after getting one location

stopLocationUpdates();

}

}

};

getLocationButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

getCurrentLocation();

}

});

}

private void getCurrentLocation() {

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

requestLocationPermission();

return;

}

if (isRequestingLocation) {

return; // Already requesting

}

locationTextView.setText("Getting location...");

// First, try to get last known location (quick)

fusedLocationProviderClient.getLastLocation()

.addOnSuccessListener(this, new OnSuccessListener<Location>() {

@Override

public void onSuccess(Location location) {

if (location != null) {

// Got cached location

double latitude = location.getLatitude();

double longitude = location.getLongitude();

String locationString = "Latitude: " + latitude + "\nLongitude: " + longitude;

locationTextView.setText(locationString);

} else {

// No cached location, request fresh location

requestNewLocation();

}

}

});

}

private void requestNewLocation() {

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

return;

}

isRequestingLocation = true;

// Create location request for high accuracy

LocationRequest locationRequest = new LocationRequest.Builder(

Priority.PRIORITY\_HIGH\_ACCURACY, 1000) // Update interval: 1 second

.setMinUpdateIntervalMillis(500) // Fastest update interval: 0.5 seconds

.setMaxUpdates(1) // Stop after getting 1 update

.build();

// Request location updates

fusedLocationProviderClient.requestLocationUpdates(

locationRequest,

locationCallback,

Looper.getMainLooper()

);

// Set timeout in case location takes too long

locationTextView.postDelayed(new Runnable() {

@Override

public void run() {

if (isRequestingLocation) {

stopLocationUpdates();

locationTextView.setText("Location request timed out. Please ensure GPS is enabled and try again.");

}

}

}, 10000); // 10 second timeout

}

private void stopLocationUpdates() {

if (isRequestingLocation) {

fusedLocationProviderClient.removeLocationUpdates(locationCallback);

isRequestingLocation = false;

}

}

private void requestLocationPermission() {

ActivityCompat.requestPermissions(this,

new String[]{Manifest.permission.ACCESS\_FINE\_LOCATION},

LOCATION\_PERMISSION\_REQUEST\_CODE);

}

@Override

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

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if (requestCode == LOCATION\_PERMISSION\_REQUEST\_CODE) {

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

getCurrentLocation();

} else {

Toast.makeText(this, "Location permission denied", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

protected void onPause() {

super.onPause();

stopLocationUpdates();

}

}





