

|  |
| --- |
| **BATCH AND ROLL NO:** Q5 42441 |
| **EXPERIMENT NO.10** |
| **TITLE:** Design a mobile app using Google Map and GPS to trace the location. |
| **DATE OF PERFORMANCE:** |
| **DATE OF SUBMISSION:** |

**Title:** Design a mobile app using Google Map and GPS to trace the location.

# Requirements:

1 Android studio

1. Google Play service Packages

# Theory:

**Introduction**

In the ever-connected world of mobile applications, harnessing the power of location-based services has become essential for creating dynamic and context-aware applications. This lab focuses on designing a mobile application that integrates Google Maps and Global Positioning System (GPS) functionalities, enabling users to trace their location and visualize it on a map. The fusion of Google Maps and GPS empowers developers to craft applications that provide real-time location-based information, fostering an enriched user experience.

**Objective of the Lab:** The primary objective of this lab is to guide you through the process of designing a mobile application that leverages Google Maps and GPS technology. By the end of this lab, you should be adept at implementing features such as obtaining real-time location updates, displaying the user's location on a Google Map, and incorporating additional functionalities to enhance the overall location tracking experience.

# Components of the Application:

* 1. **Google Maps Integration:**
     + The application will integrate Google Maps, allowing users to view and interact with a map interface.
     + Developers will utilize the Google Maps API to embed the map and leverage its rich features for location-based interactions.

# GPS Location Tracking:

* + - The application will utilize the device's GPS functionality to trace and update the user's real-time location.
    - GPS data will be used to dynamically update the user's marker on the Google Map.



# Lab Prerequisites:

* Basic understanding of mobile application development concepts.
* Familiarity with the chosen development environment (e.g., Android Studio).
* Prior knowledge of programming languages such as Java (for Android)

# Steps:

**Step 1: Set Up Your Development Environment**

* Ensure that you have Android Studio installed and configured on your machine.
* Create a new project in Android Studio.

# Step 2: Obtain Google Maps API Key

* Obtain a Google Maps API key from the Google Cloud Console.
* Enable the "Maps SDK for Android" for your project.

# Step 3: Add Google Maps SDK to Your Project

* Open the build.gradle file (Module: app) and add the following dependency: implementation 'com.google.android.gms:play-services-maps:17.0.1'

# Step 4: Design the User Interface

* Open the XML layout file associated with your main activity (e.g., activity\_main.xml).
* Add a SupportMapFragment or MapView element to your layout to display the Google Map.

# Step 5: Implement Google Maps Integration

* Open the Java file associated with your main activity (e.g., MainActivity.java).
* Initialize the Google Map and set up its features, such as zoom controls and markers.

# Step 6: Implement GPS Location Tracking

* Request permission for accessing the device's location in the AndroidManifest.xml.
* Implement a LocationListener to receive location updates.

# Step 7: Test Your Application

* Run your application on an emulator or a physical device.
* Verify that the Google Map is displayed, and the user's location is updated on the map as they move.



# XML Code:

activity\_main.xml:

<?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:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:orientation="vertical" tools:context=".MainActivity">

<fragment android:id="@+id/mapFragment"

android:name="com.google.android.gms.maps.SupportMapFragment" android:layout\_width="match\_parent"

android:layout\_height="0dp" android:layout\_weight="1" />

<Button

android:id="@+id/btnTrackLocation" android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:text="Track Location" android:background="#2196F3" android:textColor="@android:color/white" android:padding="16dp"/>

</LinearLayout>

google\_maps\_api.xml:

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

<resources>

<string name="google\_maps\_key">My\_API\_Key</string>

</resources>

AndroidManifest.xml:

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

<manifest xmlns:android="<http://schemas.android.com/apk/res/android>" package="com.example.expt10\_42441">

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

<uses-permission android:name="android.permission.INTERNET" />

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



<application android:allowBackup="true" android:icon="@mipmap/ic\_launcher" android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round" android:supportsRtl="true" android:theme="@style/AppTheme">

<meta-data android:name="com.google.android.geo.API\_KEY" android:value="@string/google\_maps\_key" />

<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>

# Java Code:

MainActivity.java:

package com.example.expt10\_42441; import android.Manifest;

import android.content.pm.PackageManager; import android.location.Location;

import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.Toast;

import androidx.annotation.NonNull; import androidx.core.app.ActivityCompat;

import androidx.fragment.app.FragmentActivity;

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.maps.CameraUpdateFactory; import com.google.android.gms.maps.GoogleMap;

import com.google.android.gms.maps.OnMapReadyCallback;



import com.google.android.gms.maps.SupportMapFragment; import com.google.android.gms.maps.model.LatLng;

import com.google.android.gms.maps.model.MarkerOptions;

public class MainActivity extends FragmentActivity implements OnMapReadyCallback { private static final int *LOCATION\_PERMISSION\_REQUEST\_CODE* = 1;

private GoogleMap mMap;

private FusedLocationProviderClient mFusedLocationClient; private LocationRequest mLocationRequest;

private LocationCallback mLocationCallback; private Button btnTrackLocation;

private boolean isTracking = false;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*);

// Initialize Location Services

mFusedLocationClient = LocationServices.*getFusedLocationProviderClient*(this);

// Initialize Map Fragment

SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()

.findFragmentById(R.id.*mapFragment*); mapFragment.getMapAsync(this);

// Track Location Button

btnTrackLocation = findViewById(R.id.*btnTrackLocation*); btnTrackLocation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) { toggleLocationTracking();

}

});

createLocationRequest(); createLocationCallback();

}

private void createLocationRequest() { mLocationRequest = LocationRequest.*create*()

.setInterval(100000)

.setFastestInterval(50000)

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

}



private void createLocationCallback() { mLocationCallback = new LocationCallback() {

@Override

public void onLocationResult(LocationResult locationResult) { if (locationResult == null) {

return;

}

for (Location location : locationResult.getLocations()) { updateMapLocation(location);

}

}

};

}

private void toggleLocationTracking() { if (checkLocationPermission()) {

if (!isTracking) { startLocationUpdates();

btnTrackLocation.setText("Stop Tracking"); isTracking = true;

} else {

stopLocationUpdates(); btnTrackLocation.setText("Track Location"); isTracking = false;

}

}

}

private void startLocationUpdates() { try {

mFusedLocationClient.requestLocationUpdates(mLocationRequest, mLocationCallback, null);

} catch (SecurityException e) { e.printStackTrace();

}

}

private void stopLocationUpdates() { mFusedLocationClient.removeLocationUpdates(mLocationCallback);

}

private boolean checkLocationPermission() {

if (ActivityCompat.*checkSelfPermission*(this, Manifest.permission.*ACCESS\_FINE\_LOCATION*)

!= PackageManager.*PERMISSION\_GRANTED*) {

ActivityCompat.*requestPermissions*(this,

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

*LOCATION\_PERMISSION\_REQUEST\_CODE*);

return false;

}

return true;

}



@Override

public void onMapReady(GoogleMap googleMap) { mMap = googleMap;

if (checkLocationPermission()) { mMap.setMyLocationEnabled(true);

}

mMap.getUiSettings().setZoomControlsEnabled(true);

}

private void updateMapLocation(Location location) {

LatLng currentLocation = new LatLng(location.getLatitude(), location.getLongitude());

mMap.clear(); // Clear previous markers mMap.addMarker(new MarkerOptions()

.position(currentLocation)

.title("Your Current Location")); mMap.moveCamera(CameraUpdateFactory.*newLatLngZoom*(currentLocation, 15));

Toast.*makeText*(this,

"Location Updated: " + location.getLatitude() + ", "

+ location.getLongitude(), Toast.*LENGTH\_SHORT*).show();

}

@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*) {

if (checkLocationPermission()) { mMap.setMyLocationEnabled(true);

}

} else {

Toast.*makeText*(this,

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

}

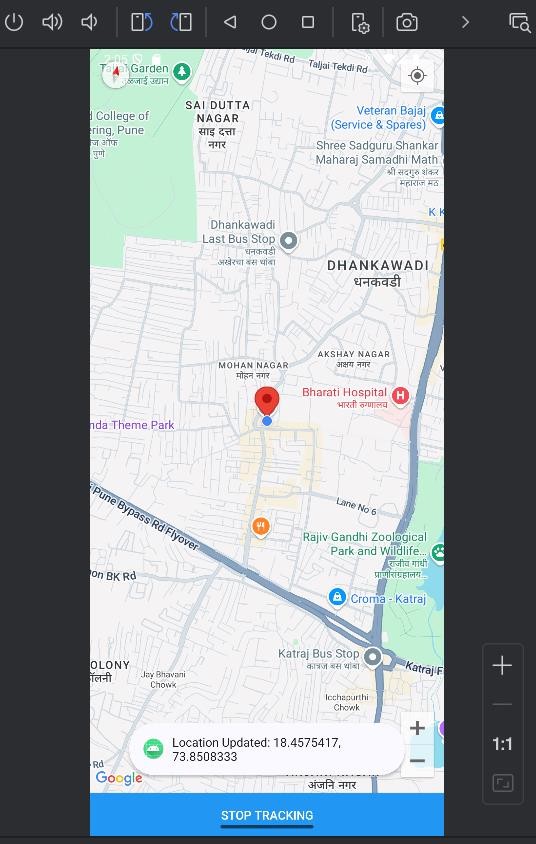
}

}

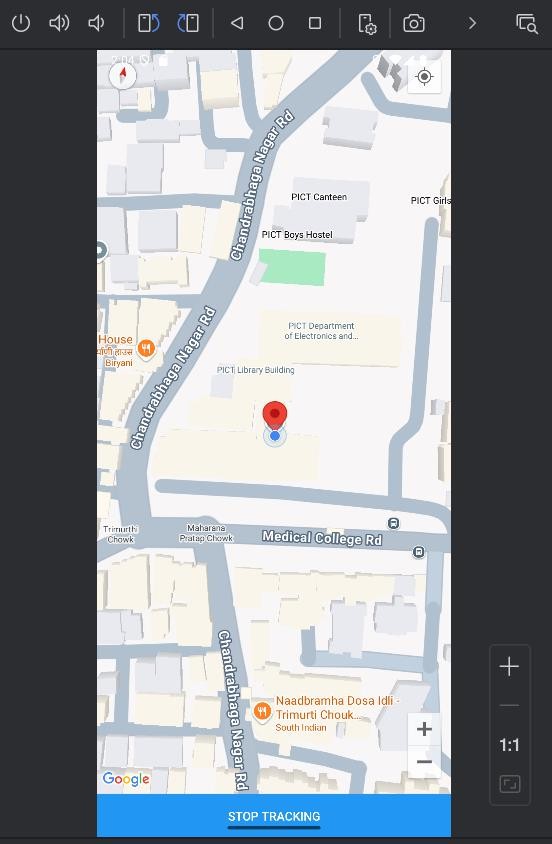
}



# Output:

****

****



**Conclusion:**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**

**……………………………………………………………………………………………**