NAME:- DUBEY KARAN SANJEEV

CLASS:- B.E – 4

ROLL NO:- 04

BATCH:- A

Experiment 7

Develop a native application that uses GPS location information

**package** com.frsarker.locationfinder;

**import** androidx.annotation.NonNull;

**import** androidx.appcompat.app.AppCompatActivity;

### import androidx.core.app.ActivityCompat;

**import** android.Manifest;

**import** android.annotation.SuppressLint;

**import** android.content.Context;

**import** android.content.Intent;

**import** android.content.pm.PackageManager;

**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** 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;

**public class** MainActivity **extends** AppCompatActivity {

int PERMISSION\_ID = 44;

FusedLocationProviderClient **mFusedLocationClient**; TextView **latTextView**, **lonTextView**;

@Override **protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState); setContentView(R.layout.***activity\_main***);

**latTextView** = findViewById(R.id.***latTextView***); **lonTextView** = findViewById(R.id.***lonTextView***);

**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** {

**latTextView**.setText(location.getLatitude()+**""**); **lonTextView**.setText(location.getLongitude()+**""**);

}

}

}

);

} **else** {

Toast.*makeText*(**this**, **"Turn on 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(0); 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(); **latTextView**.setText(mLastLocation.getLatitude()+**""**); **lonTextView**.setText(mLastLocation.getLongitude()+**""**);

}

};

**private boolean** checkPermissions() { **if**

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

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

PackageManager.***PERMISSION\_GRANTED***) { **return true**;

}

### return false;

}

**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, String[] permissions, **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/andro](http://schemas.android.com/apk/res/android)**[**id"**](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:id="@+id/latTextView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Latitude: "/>

<**TextView android:id="@+id/lonTextView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"**

### android:text="Longitude: "/>

</**LinearLayout**>

# Output:

