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/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: