ATHARVA KALE **PRACTICAL NO. 7** **ROLL NO: 27**

AIM: Android program to work with google maps and location.

❖ THEORY:

1. Location:
   - A data class representing a geographic location.
   - The Location object represents a geographic location which can consist of a latitude, longitude, time stamp, and other information such as bearing, altitude and velocity.
   - All locations generated through LocationManager are guaranteed to have a valid latitude, longitude, and timestamp (both UTC time and elapsed real-time since boot).
   - Methods of this object are:

   1. float distanceTo(Location dest)- Returns the approximate distance in meters between this location and the given location.
   2. float getAccuracy()- Get the estimated accuracy of this location, in meters.
   3. double getAltitude()- Get the altitude if available, in meters above sea level.
   4. float getBearing()- Get the bearing, in degrees.
   5. double getLatitude()- Get the latitude, in degrees.
   6. double getLongitude()- Get the longitude, in degrees.

2. Android Google Map:
   - Android provides facility to integrate Google map in our application.

- Google map displays your current location, navigate location direction, search location etc.
- We can also customize Google map according to our requirement.
- Google map API provides several methods that help to customize Google map.
- These methods are as following:
  1) addCircle(CircleOptions options): This method add circle to map.
  2) addPolygon(PolygonOptions options): This method add polygon to map.
  3) addTileOverlay(TileOverlayOptions options): This method add tile overlay to the map.
  4) animateCamera(CameraUpdate update): This method moves the map according to the update with an animation.
  5) clear(): This method removes everything from the map.
  6) getMyLocation(): This method returns the currently displayed user location.
  7) moveCamera(CameraUpdate update): This method reposition the camera according to the instructions defined in the update.
  8) setTrafficEnabled(boolean enabled): This method set the traffic layer on or off.
  9) snapshot(GoogleMap.SnapshotReadyCallback callback): This method takes a snapshot of the map.

10) stopAnimation(): This method stops the camera animation if there is any progress.

A) Add marker method to be used in application students are creating.

- **CODE:**
  - **MapsActivity.java:**

```
package com.example.mapdemo;


import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Address;
import android.location.Criteria;
import android.location.Geocoder;
import android.location.Location;
import android.location.LocationManager;
import android.os.Build;
import android.os.Bundle;
import android.widget.Toast;


import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import androidx.fragment.app.FragmentActivity;


import com.google.android.gms.common.ConnectionResult;
import com.google.android.gms.common.api.GoogleApiClient;
import com.google.android.gms.location.LocationListener;
import com.google.android.gms.location.LocationRequest;
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.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.*;
import com.google.maps.android.SphericalUtil;

import java.io.IOException;
import java.util.List;
import java.util.Locale;
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback,
        GoogleApiClient.ConnectionCallbacks,
        GoogleApiClient.OnConnectionFailedListener,
        LocationListener {
    public static final int MY_PERMISSIONS_REQUEST_LOCATION = 99;
    GoogleApiClient mGoogleApiClient;
    Location mLastLocation;
    Marker mCurrLocationMarker;
    LocationRequest mLocationRequest;
    private GoogleMap mMap;
    Double distance;
    @Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_maps);

    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        checkLocationPermission();
    }
    SupportMapFragment mapFragment = (SupportMapFragment)
            getSupportFragmentManager()
                    .findFragmentById(R.id.map);

    mapFragment.getMapAsync(this);
}
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;
    mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
    mMap.getUiSettings().setZoomControlsEnabled(true);
    mMap.getUiSettings().setZoomGesturesEnabled(true);
    mMap.getUiSettings().setCompassEnabled(true);

    //Initialize Google Play Services
    if (android.os.Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {
            buildGoogleApiClient();
            mMap.setMyLocationEnabled(true);
          }
        } else {
          buildGoogleApiClient();
          mMap.setMyLocationEnabled(true);
        }
      }
      protected synchronized void buildGoogleApiClient() {
        mGoogleApiClient = new GoogleApiClient.Builder(this)
                .addConnectionCallbacks(this)
                .addOnConnectionFailedListener(this)
                .addApi(LocationServices.API)
                .build();
        mGoogleApiClient.connect();
      }
      @Override
      public void onConnected(Bundle bundle) {
        mLocationRequest = new LocationRequest();
        mLocationRequest.setInterval(1000);
        mLocationRequest.setFastestInterval(1000);
        mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_ACCURACY);
        if (ContextCompat.checkSelfPermission(this,
                Manifest.permission.ACCESS_FINE_LOCATION)
                == PackageManager.PERMISSION_GRANTED) {

LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,mLocationRequ
est, this);
        }
```

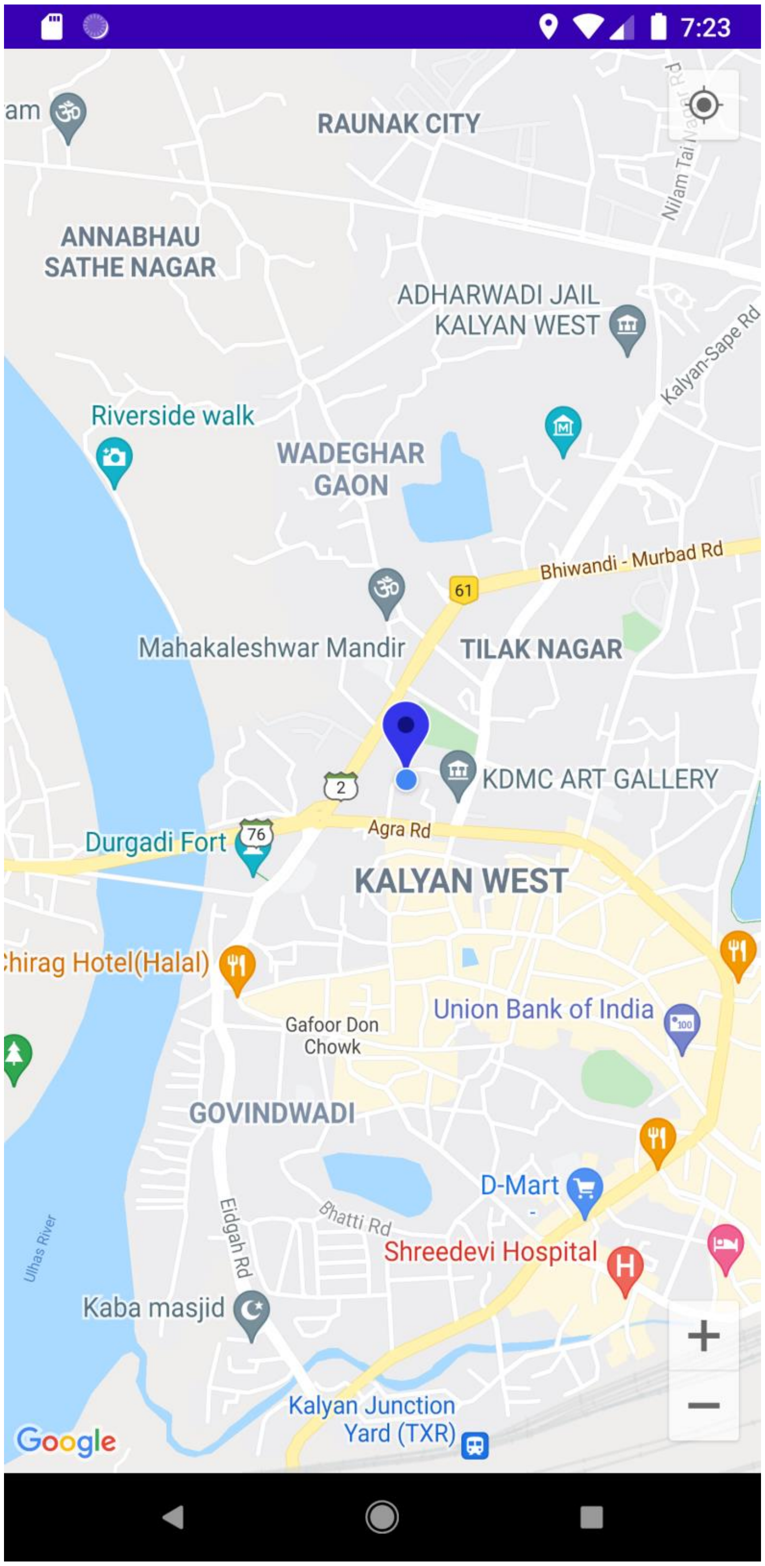
```
  }
  @Override
  public void onConnectionSuspended(int i) {
  }
  @Override
  public void onLocationChanged(Location location) {
    mLastLocation = location;
    if (mCurrLocationMarker != null) {
      mCurrLocationMarker.remove();
    }      LatLng latLng = new LatLng(location.getLatitude(), location.getLongitude());
    MarkerOptions markerOptions = new MarkerOptions();
    markerOptions.position(latLng);
    LocationManager locationManager = (LocationManager)
        getSystemService(Context.LOCATION_SERVICE);
    String provider = locationManager.getBestProvider(new Criteria(), true);
    if (ActivityCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION)
            != PackageManager.PERMISSION_GRANTED) {
      return;
    }
    Location locations = locationManager.getLastKnownLocation(provider);
    List<String> providerList = locationManager.getAllProviders();
    if (null != locations && null != providerList && providerList.size() > 0) {
      double longitude = locations.getLongitude();
      double latitude = locations.getLatitude();
```

```
        Geocoder geocoder = new Geocoder(getApplicationContext(),
                Locale.getDefault());
        try {
            List<Address> listAddresses = geocoder.getFromLocation(latitude,
                    longitude, 1);
            if (null != listAddresses && listAddresses.size() > 0) {
                String state = listAddresses.get(0).getAdminArea();
                String country = listAddresses.get(0).getCountryName();
                String subLocality = listAddresses.get(0).getSubLocality();
                markerOptions.title("" + latLng + "," + subLocality + "," + state
                        + "," + country);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

markerOptions.icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE_BL
UE));
    mCurrLocationMarker = mMap.addMarker(markerOptions);
    mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
    mMap.animateCamera(CameraUpdateFactory.zoomTo(11));
    if (mGoogleApiClient != null) {
        LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
                this);
    }
  }
  @Override
  public void onConnectionFailed(ConnectionResult connectionResult) {
  }
  public boolean checkLocationPermission() {
    if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {
      if (ActivityCompat.shouldShowRequestPermissionRationale(this,
          Manifest.permission.ACCESS_FINE_LOCATION)) {
        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            MY_PERMISSIONS_REQUEST_LOCATION);
      } else {
        ActivityCompat.requestPermissions(this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            MY_PERMISSIONS_REQUEST_LOCATION);
      }
      return false;
    } else {
      return true;
    }
  }
  @Override
  public void onRequestPermissionsResult(int requestCode, String permissions[], int[]
grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    switch (requestCode) {
      case MY_PERMISSIONS_REQUEST_LOCATION: {
        if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            if (ContextCompat.checkSelfPermission(this,
                    Manifest.permission.ACCESS_FINE_LOCATION)
                    == PackageManager.PERMISSION_GRANTED) {
                if (mGoogleApiClient == null) {
                    buildGoogleApiClient();
                }
                mMap.setMyLocationEnabled(true);
            }
        } else {
            Toast.makeText(this, "permission denied",
                    Toast.LENGTH_LONG).show();
        }
        return;
}}}}
```

- **Activity_maps.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:map="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/map"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MapsActivity" />
```

- **OUTPUT:**


7:23
RAUNAK CITY
ANNABHAU SATHE NAGAR
ADHARWADI JAIL KALYAN WEST
Riverside walk
WADEGHAR GAON
Bhiwandi - Murbad Rd
Mahakaleshwar Mandir
TILAK NAGAR
KDMC ART GALLERY
Agra Rd
Durgadi Fort
KALYAN WEST
Chirag Hotel(Halal)
Union Bank of India
Gafoor Don Chowk
GOVINDWADI
D-Mart
Bhatti Rd
Shreedevi Hospital
Eidgah Rd
Kaba masjid
Ulhas River
Kalyan Junction Yard (TXR)
Google

ATHARVA KALE PRACTICAL NO. 7 ROLL NO: 27

B) Calculate route distance between two locations.

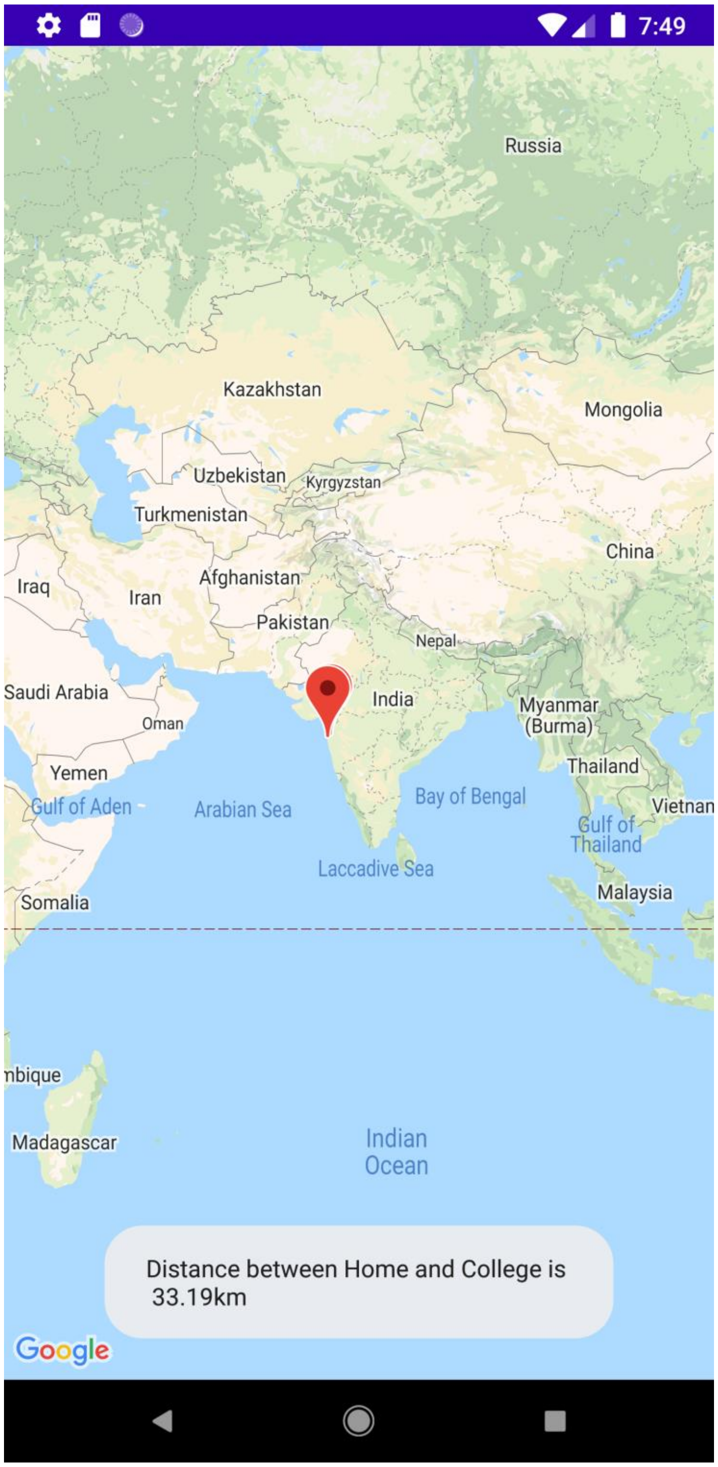
- **CODE:**
  - **MapsActivity.java:**

```
package com.example.practical7_a;

import android.os.Bundle;
import android.widget.Toast;
import androidx.fragment.app.FragmentActivity;
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;
import com.google.maps.android.SphericalUtil;


public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {
   private GoogleMap mMap;
   LatLng home = new LatLng(19.24760301731887, 73.1221433756462);
   LatLng college = new LatLng(19.04589540226512, 72.88918131108983);
   Double distance;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_maps);
      // Obtain the SupportMapFragment and get notified
      // when the map is ready to be used.
      SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager().findFragmentById(R.id.map);
      mapFragment.getMapAsync(this);
   }
   @Override
   public void onMapReady(GoogleMap googleMap) {
      mMap = googleMap;
      // on below line we are calculating the distance between sydney and brisbane
      distance = SphericalUtil.computeDistanceBetween(home, college);
      googleMap.addMarker(new MarkerOptions()
            .position(home)
            .title("Marker in Home"));

      googleMap.addMarker(new MarkerOptions()
            .position(college)
            .title("Marker in College"));
      Toast.makeText(this, "Distance between Home and College is \n " + String.format("%.2f", distance /
1000) + "km", Toast.LENGTH_SHORT).show();
   }
}
```

- **Activity_maps.xml:**

```
<?xml version="1.0" encoding="utf-8"?>
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:map="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/map"
   android:name="com.google.android.gms.maps.SupportMapFragment"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MapsActivity" />
```

- **OUTPUT:**



Distance between Home and College is 33.19km

❖ **CONCLUSION:**

Hence we successfully implemented google maps and location.