PRACTICAL: 13

AIM:

Create a Check-Out application to find the hottest places people are leaving. Use the Google Places API for Android to manage nearby locations and Firebase to store and synchronize data across devices in real time.

THEORY:

Google Place API

The Places API is a service that returns information about places using HTTP requests. Places are defined within this API as establishments, geographic locations, or prominent points of interest.

The following place requests are available:

- Place Search returns a list of places based on a user's location or search string.
- Place Details returns more detailed information about a specific place, including user reviews.
- Place Photos provides access to the millions of place-related photos stored in Google's Place database.
- Place Autocomplete automatically fills in the name and/or address of a place as users type.
- Query Autocomplete provides a query prediction service for text-based geographic searches, returning suggested queries as users type.

Each of the services is accessed as an HTTP request, and returns either an JSON or XML response.

All requests to a Places service must use the https:// protocol, and include an API key.

The Places API uses a place ID to uniquely identify a place.

CODE:

Program: activity_main.xml

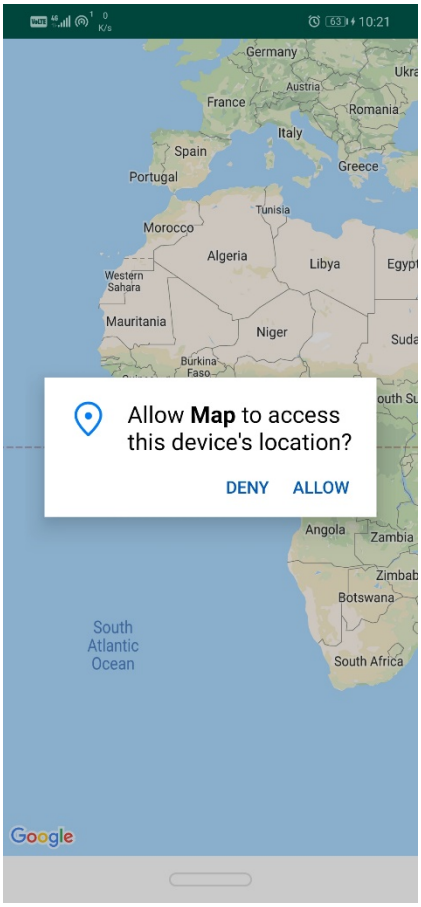
```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="match_parent"
    android:layout_width="match_parent">

    <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="in.ac.charusat.cspit.it.practical_13.MapsActivity" />

</FrameLayout>
```

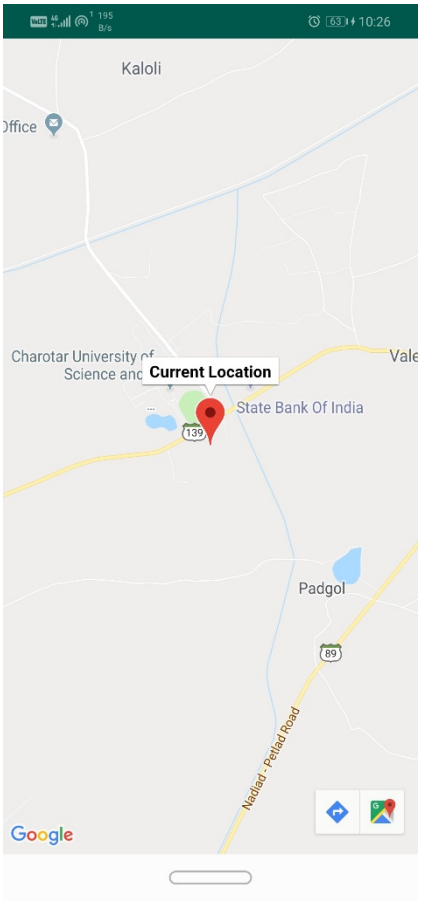
Program: MainActivity.java

```
package in.ac.charusat.cspit.it.map;

import android.Manifest;
import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.support.annotation.NonNull;
import android.support.v4.app.ActivityCompat;
import android.support.v4.app.FragmentActivity;
import android.os.Bundle;
import android.view.View;
import android.view.ViewTreeObserver;
import android.widget.Button;
import android.widget.Toast;

import com.google.android.gms.location.FusedLocationProviderClient;
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.LatLngBounds;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

   private GoogleMap mMap;
   private LatLngBounds.Builder mBounds;
   private FusedLocationProviderClient mFusedLocationClient;

   @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);
```

```
        mBounds = new LatLngBounds.Builder();

        mFusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
    }

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

        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
                && ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, 1);

        } else {
            // already permission granted
            mFusedLocationClient.getLastLocation().addOnSuccessListener(this, location -> {
                if (location != null) {
                    LatLng ll = new LatLng(location.getLatitude(), location.getLongitude());
                    addPointToViewPort(ll);
                }
            });
        }
    }

    private void addPointToViewPort(LatLng newPoint) {
        mBounds.include(newPoint);
        mMap.addMarker(new MarkerOptions().position(newPoint).title("Current Location"));
        mMap.animateCamera(CameraUpdateFactory.newLatLngBounds(mBounds.build(), 15));
    }
}
```

OUTPUT:





CONCLUSION:

Google maps are very helpful in our day to day life. So We had learn about how to use google map in android device and implement it using google API. The main aim of this practical cannot be achieved because they are asking to enable the key and for that they are charging some amount so we made only simple app that shows current location.