**Implement GPS App in android studio.**

**MainActivity.java**

package com.example.gmap;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.location.Address;

import android.location.Geocoder;

import android.location.Location;

import android.os.Build;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Switch;

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.OnSuccessListener;

import java.util.List;

public class MainActivity extends AppCompatActivity {

public static final int Initial\_Interval = 30000;

private static final int PERMISSION\_FINE\_LOCATION = 99;

public static final int Fastest\_Interval = 5000;

Location currentLocation;

List<Location> savedLocations;

TextView tv\_lat, tv\_lon, tv\_altitude, tv\_accuracy, tv\_speed, tv\_sensor, tv\_updates, tv\_address, tv\_wayPointCounts;

Switch sw\_locationUpdates, sw\_gps;

Button btn\_newWayPoint, btn\_showWayPointList, btn\_showMap;

FusedLocationProviderClient fusedLocationProviderClient;

LocationRequest locationRequest;

LocationCallback locationCallBack;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

tv\_lat = findViewById(R.id.tv\_lat);

tv\_lon = findViewById(R.id.tv\_lon);

tv\_altitude = findViewById(R.id.tv\_altitude);

tv\_accuracy = findViewById(R.id.tv\_accuracy);

tv\_speed = findViewById(R.id.tv\_speed);

tv\_sensor = findViewById(R.id.tv\_sensor);

tv\_updates = findViewById(R.id.tv\_updates);

tv\_address = findViewById(R.id.tv\_address);

sw\_locationUpdates = findViewById(R.id.sw\_locationsupdates);

sw\_gps = findViewById(R.id.sw\_gps);

btn\_newWayPoint = findViewById(R.id.btn\_newWayPoint);

btn\_showWayPointList = findViewById(R.id.btn\_showWayPointsList);

btn\_showMap = findViewById(R.id.btn\_showMap);

tv\_wayPointCounts = findViewById(R.id.tv\_countOfCrumbs);

locationRequest = new LocationRequest();

locationRequest.setInterval(Initial\_Interval);

locationRequest.setFastestInterval(Fastest\_Interval);

locationRequest.setPriority(LocationRequest.PRIORITY\_BALANCED\_POWER\_ACCURACY);

locationCallBack = new LocationCallback() {

@Override

public void onLocationResult(LocationResult locationResult) {

super.onLocationResult(locationResult);

updateUIValues(locationResult.getLastLocation());

}

};

sw\_gps.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (sw\_gps.isChecked()) {

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

tv\_sensor.setText("Using GPS Sensors");

} else {

locationRequest.setPriority(LocationRequest.PRIORITY\_BALANCED\_POWER\_ACCURACY);

tv\_sensor.setText("Using Towers + WIFI");

}

}

});

sw\_locationUpdates.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if (sw\_locationUpdates.isChecked()) {

startLocationUpdates();

} else {

stopLocationUpdates();

}

}

});

btn\_newWayPoint.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

MyApplication myApplication = (MyApplication)getApplicationContext();

savedLocations = myApplication.getMyLocations();

savedLocations.add(currentLocation);

}

});

btn\_showWayPointList.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, ShowSavedLocationsList.class);

startActivity(intent);

}

});

btn\_showMap.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent intent = new Intent(MainActivity.this, MapsActivity.class);

startActivity(intent);

}

});

updateGPS();

}

private void stopLocationUpdates() {

tv\_updates.setText("Location is not being tracked");

tv\_lat.setText("Location is not being tracked");

tv\_lon.setText("Location is not being tracked");

tv\_speed.setText("Location is not being tracked");

tv\_address.setText("Location is not being tracked");

tv\_accuracy.setText("Location is not being tracked");

tv\_altitude.setText("Location is not being tracked");

tv\_sensor.setText("Location is not being tracked");

fusedLocationProviderClient.removeLocationUpdates(locationCallBack);

}

private void startLocationUpdates() {

tv\_updates.setText("Location is being tracked");

if (ActivityCompat.checkSelfPermission(this, Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED) {

fusedLocationProviderClient.requestLocationUpdates(locationRequest, locationCallBack, null);

}

}

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

switch (requestCode){

case PERMISSION\_FINE\_LOCATION:

if(grantResults[0] == PackageManager.PERMISSION\_GRANTED){

updateGPS();

}

else{

Toast.makeText(getApplicationContext(), "This app requires permission to be granted in order to work properly", Toast.LENGTH\_LONG).show();

}

}

}

private void updateGPS(){

fusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this);

if(ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS\_FINE\_LOCATION) == PackageManager.PERMISSION\_GRANTED){

fusedLocationProviderClient.getLastLocation().addOnSuccessListener(this, new OnSuccessListener<Location>() {

@Override

public void onSuccess(Location location) {

updateUIValues(location);

currentLocation = location;

}

});

}

else{

if(Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.M){

requestPermissions(new String[] {Manifest.permission.ACCESS\_FINE\_LOCATION}, PERMISSION\_FINE\_LOCATION);

}

}

}

private void updateUIValues(Location location) {

tv\_lat.setText(String.valueOf(location.getLatitude()));

tv\_lon.setText(String.valueOf(location.getLongitude()));

tv\_accuracy.setText(String.valueOf(location.getAccuracy()));

if(location.hasAltitude()){

tv\_altitude.setText(String.valueOf(location.getAltitude()));

}else{

tv\_altitude.setText("Not Available");

}

if(location.hasSpeed()){

tv\_speed.setText(String.valueOf(location.getSpeed()));

}else{

tv\_speed.setText("Not Available");

}

Geocoder geocoder = new Geocoder(this);

try {

List<Address> addresses = geocoder.getFromLocation(location.getLatitude(), location.getLongitude(), 1);

tv\_address.setText(addresses.get(0).getAddressLine(0));

}catch (Exception e){

e.printStackTrace();

}

MyApplication applicationContext = (MyApplication) getApplicationContext();

savedLocations = applicationContext.getMyLocations();

tv\_wayPointCounts.setText(Integer.toString(savedLocations.size()));

}

}

**MyApplication.java**

package com.example.gmap;

import android.app.Application;

import android.location.Location;

import java.util.ArrayList;

import java.util.List;

public class MyApplication extends Application {

private static MyApplication singleton;

private List<Location> myLocations;

public MyApplication getInstance(){

return singleton;

}

public void onCreate(){

super.onCreate();

singleton = this;

myLocations = new ArrayList<>();

}

public List<Location> getMyLocations() {

return myLocations;

}

public void setMyLocations(List<Location> myLocations) {

this.myLocations = myLocations;

}

}

**ShowSavedLocationsList.java**

package com.example.gmap;

import androidx.appcompat.app.AppCompatActivity;

import android.location.Location;

import android.os.Bundle;

import android.widget.ArrayAdapter;

import android.widget.ListView;

import java.util.List;

public class ShowSavedLocationsList extends AppCompatActivity {

ListView lv\_savedLocations;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_show\_saved\_locations\_list);

lv\_savedLocations = findViewById(R.id.lv\_wayPoints);

MyApplication myApplication = (MyApplication) getApplicationContext();

List<Location> savedLocations = myApplication.getMyLocations();

lv\_savedLocations.setAdapter(new ArrayAdapter<Location>(this, android.R.layout.simple\_list\_item\_1, savedLocations));

}

}

**MapsActivity.java**

package com.example.gmap;

import androidx.fragment.app.FragmentActivity;

import android.location.Location;

import android.os.Bundle;

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;

import com.example.gmap.databinding.ActivityMapsBinding;

import java.util.List;

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

private GoogleMap mMap;

private ActivityMapsBinding binding;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

binding = ActivityMapsBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

// 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;

MyApplication myApplication = (MyApplication) getApplicationContext();

List<Location> savedLocations = myApplication.getMyLocations();

// Add a marker in Sydney and move the camera

LatLng location = new LatLng(savedLocations.get(0).getLatitude(), savedLocations.get(0).getLongitude());

mMap.addMarker(new MarkerOptions().position(location).title("Marker in Sydney"));

mMap.moveCamera(CameraUpdateFactory.newLatLng(location));

}

}

**Output:**

  