Experiment 07: To develop an android application that uses GPS Location Information.

PART A

A.1 Aim: To develop an android application that uses GPS Location Information.

A.2 Objectives: **To introduce students with various tools like** Android Studio, NS2, Wire-shark, Cisco packet tracer, WAP supported browser etc.

A.3 Outcome: **After successful completion of this experiment students will be able to** develop an android application that uses GPS Location Information. **A.4Theory:**

SOFTWARE:

- Android Studio
- The Android SDK (Starter Package)
- Gradle
- Java Development Kit (JDK) 5

DESCRIPTION:

1. Open android studio and select new android project .
2. Give project name and select next
3. Choose the android version.
4. Enter the package name. package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side and select our project.

6. Go to res folder and select layout. Double click the main.xml file
7. Now you can see the Graphics layout window.

INPUT:

MAINACTIVITY.JAVA

```
package com.gpslocation;

//import android.R; import android.view.View;
import android.widget.Button; import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity; import android.os.Bundle;


public class MainActivityextends AppCompatActivity{

	/** Called when the activity is first created. */ Button
	btnShowLocation; GPStracegps;

	@Override
	public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);
		btnShowLocation=(Button)findViewById(R.id.show_Location);
		btnShowLocation.setOnClickListener(new View.OnClickListener() {
			@Override
			public void onClick(View v) {
				// TODO Auto-generated method stub gps=new
				GPStrace(MainActivity.this);
				if(gps.canGetLocation()) { double
				latitude=gps.getLatitude(); double
					longitude=gps.getLongtiude();
					Toast.makeText(getApplicationContext(),"Your Location is
\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH_LONG).show();
				}
				else { gps.showSettingAlert();
				}

			}
		});
	}
}
```

ACTIVITY_MAIN.XML

```
<?xml version="1.0" encoding="utf-8"?>

<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"andro
        id:id="@+id/relativeLayout1" android:layout_width="match_parent"
        android:layout_height="match_parent">

        <Button android:id="@+id/show_Location"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show_Location"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

</RelativeLayout>
```

GPSTRACE.JAVA

```
package com.gpslocation;

import android.app.AlertDialog; import
android.app.Service; import android.content.Context;
import android.content.DialogInterface; import
android.content.Intent; import
android.location.Location; import
android.location.LocationListener; import
android.location.LocationManager; import
android.os.Bundle; import android.os.IBinder;
import android.provider.Settings;


public class GPStraceextends Service implements LocationListener{ private final Context
        context; booleanisGPSEnabled=false; booleancanGetLocation=false;
        booleanisNetworkEnabled=false; Location location;
        double latitude; double longtitude;
        private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES=10; private static
        final long MIN_TIME_BW_UPDATES=1000*60*1; protected
```

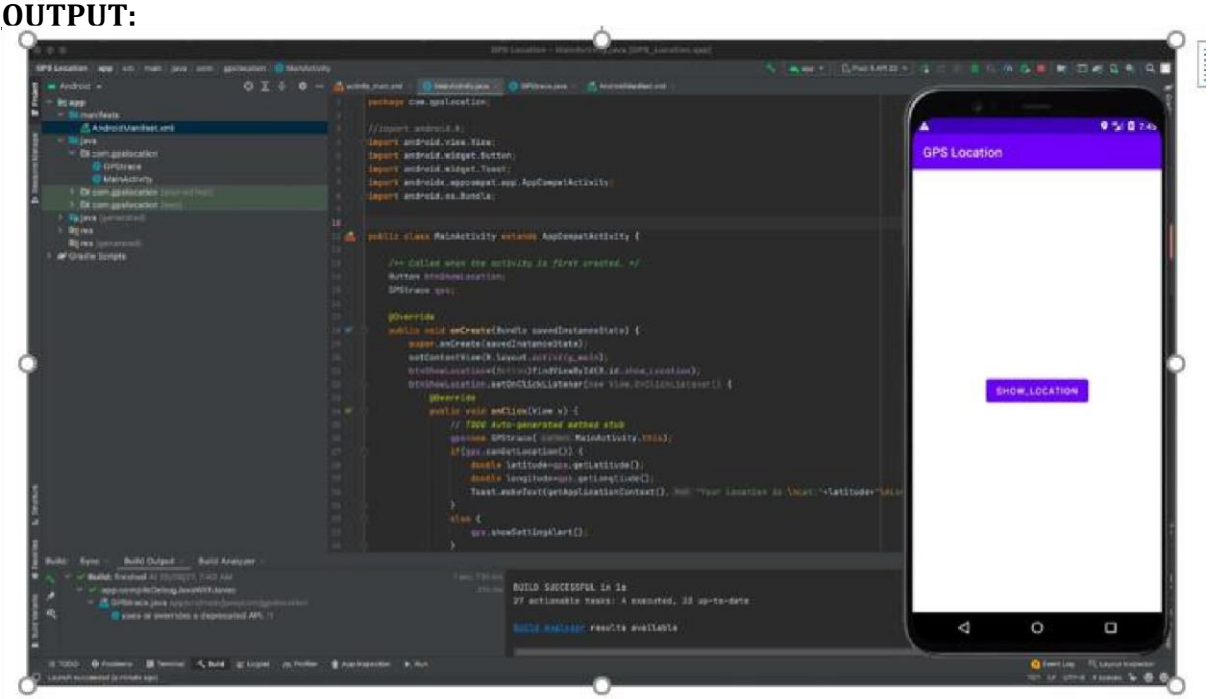
```
        LocationManagerlocationManager; public
        GPStrace(Context context) {
        this.context=context; getLocation();
        }
        public Location getLocation() { try { locationManager=(LocationManager)
context.getSystemService(LOCATION_SERVICE);

isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER
);

isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK_ PROVIDER);


                    if(!isGPSEnabled&& !isNetworkEnabled) {
                    }
                    else { this.canGetLocation=true;
                        if(isNetworkEnabled) {

locationManager.requestLocationUpdates(                    LocationManager.NETWORK_PROVIDER,
MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
                        }
                        if(locationManager!=null) {

location=locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVI DER);
                            if(location !=null) { latitude=location.getLatitude();
longtitude=location.getLongitude();
                            }
                        }
                    }


                    if(isGPSEnabled) { if(location==null)
                        {

locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TIM  E_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES, this); if(locationManager!=null) {

location=locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER)
;
                                if(location!=null) { latitude=location.getLatitude();
                                    longtitude=location.getLongitude();
                                }
                            }
                        }
```

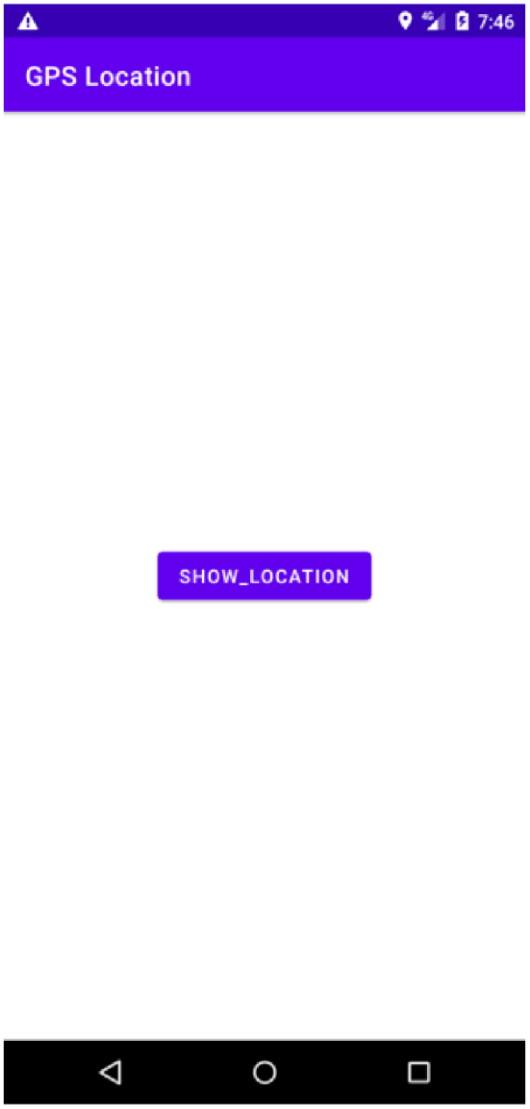
```
                }
        }
        catch(Exception e) {
                e.printStackTrace();
        }
        return location;
}

public void stopUsingGPS() { if(locationManager!=null) {
        locationManager.removeUpdates(GPStrace.this);
        }
}

public double getLatitude() {
        if(location!=null) {
        latitude=location.getLatitude();
        }
        return latitude;
}

public double getLongtiude() {
        if(location!=null) {
        longtitude=location.getLatitude();
        }
        return longtitude;
}

public booleancanGetLocation() { return this.canGetLocation;
}

public void showSettingAlert() {
        AlertDialog.BuilderalertDialog= new AlertDialog.Builder(context);
```

ANDROIDMANIFEST.XML

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"package="com.gpsl
    ocation">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GPSLocation">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <uses-permission android:name="android.permission.INTERNET"/>

</manifest>
```

OUTPUT:




7:46
GPS Location
SHOW_LOCATION


7:46
GPS Location
SHOW_LOCATION
Your Location is
Lat:37.421998333333335
Long:37.421998333333335

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. B30	Name: Pranjal Bhatt
Class :TE COMPS B	Batch:B2
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software Code written by student/steps:

MainActivity.java:

```
package com.example.exp7;
import android.location.Address;
import android.location.Geocoder;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.io.IOException;
import java.util.List;

public class MainActivity extends AppCompatActivity {
   private static final String TAG = MainActivity.class.getSimpleName();
```

```
  private TextView latitudeTextView, longitudeTextView;
  private EditText editCity;
  private Button getLocationButton;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    latitudeTextView = findViewById(R.id.latitudeTextView);
    longitudeTextView = findViewById(R.id.longitudeTextView);
    getLocationButton = findViewById(R.id.getLocationButton);
    editCity = findViewById(R.id.editTextCity);
    getLocationButton.setOnClickListener(v-> getCoordinates());
  }
  public void getCoordinates() {
    String city = editCity.getText().toString().trim();
    if (city.isEmpty()) {
      Toast.makeText(this, "Please enter a city name", Toast.LENGTH_SHORT).show();
      return;
    }

    new Thread(() -> { // Run in a background thread
      Geocoder geocoder = new Geocoder(this);
      try {
        List<Address> addresses = geocoder.getFromLocationName(city, 1);
        if (addresses == null || addresses.isEmpty()) {
          runOnUiThread(() -> Toast.makeText(this, "Couldn't Fetch Coordinates",
Toast.LENGTH_SHORT).show());
          return;
        }
        Address address = addresses.get(0);
        double latitude = address.getLatitude();
        double longitude = address.getLongitude();

        runOnUiThread(() -> { // Update UI on main thread
          latitudeTextView.setText(String.valueOf(latitude));
          longitudeTextView.setText(String.valueOf(longitude));
        });
```

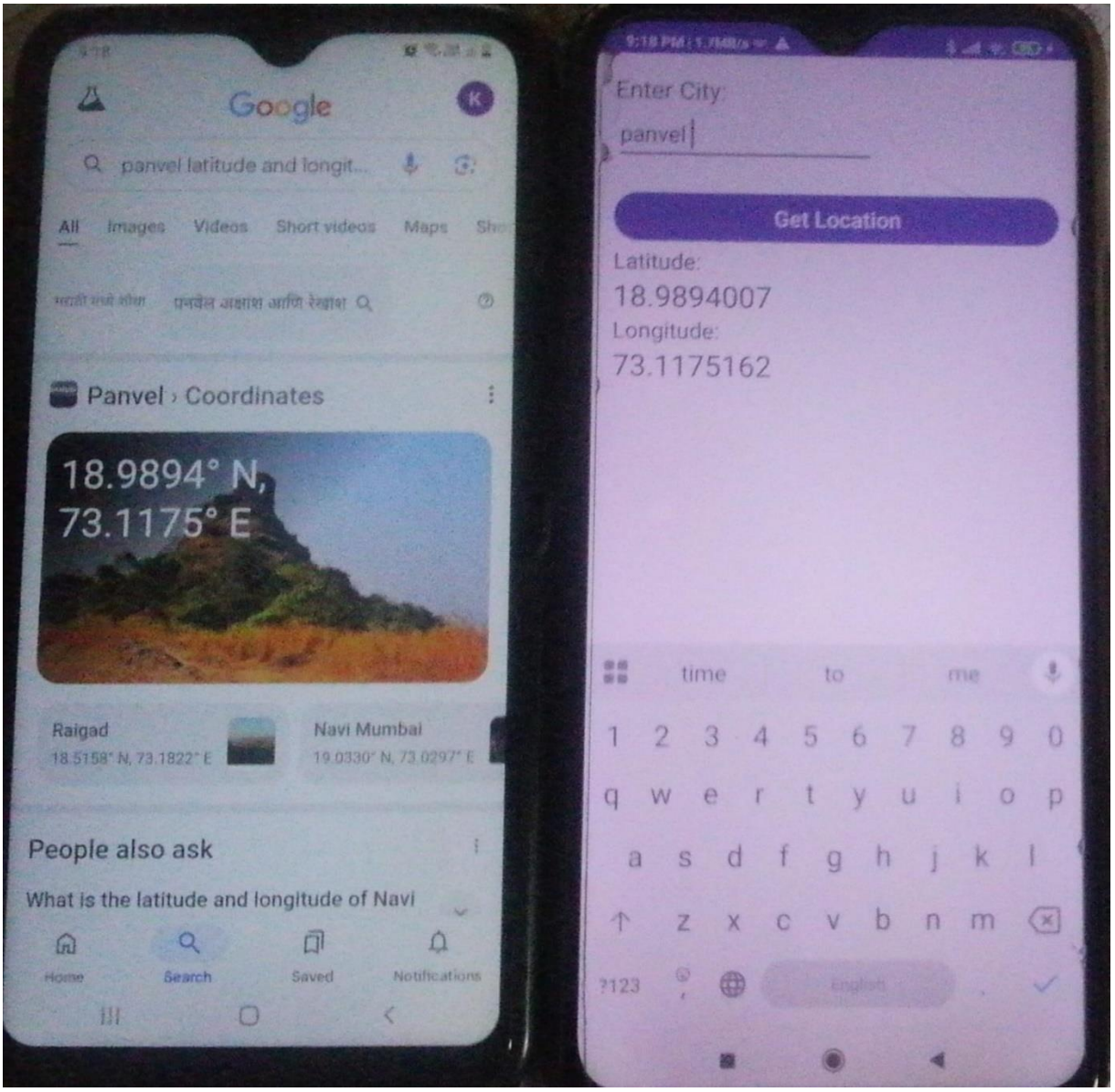
```
        } catch (IOException e) {
            runOnUiThread(() -> Toast.makeText(this, "Error fetching coordinates",
Toast.LENGTH_SHORT).show());
        }
      }).start();
   }

}
```

AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools">

   <uses-permission android:name="android.permission.INTERNET" />
   <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
   <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

   <application
      android:allowBackup="true"
      android:dataExtractionRules="@xml/data_extraction_rules"
      android:fullBackupContent="@xml/backup_rules"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:roundIcon="@mipmap/ic_launcher_round"
      android:supportsRtl="true"
      android:theme="@style/Theme.Exp7"
      tools:targetApi="31">
      <activity
         android:name=".MainActivity"
         android:exported="true">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
```

</manifest>

B.2 Input and Output:



B.3 Observations and learning:

During the experiment, an Android application was developed using Android Studio to fetch and display GPS location information. The setup involved configuring Android Studio, JDK, and Gradle, followed by creating a new project. Necessary permissions for accessing location services were added to the AndroidManifest.xml file. The application successfully retrieved latitude and longitude coordinates using the device's GPS sensor and displayed them on the screen. Testing was conducted using an emulator with a simulated location or a physical device with GPS enabled.

B.4 Conclusion:

The experiment demonstrated the practical implementation of GPS-based location tracking in Android applications. Students gained experience with integrating Android's Location API and handling permissions for accessing user location. By completing this project, they understood how real-time location services work and how they can be used in mobile applications for various purposes like navigation, tracking, and location-based services.

B.5 Question of Curiosity

1) Explain different steps required to build up this GPS project?

- **Set Up Development Environment:**
  - Install Android Studio, Java Development Kit (JDK), and Android SDK.
  - Configure Gradle dependencies.
- **Create a New Android Project:**
  - Open Android Studio and select "New Project."
  - Choose an appropriate project template (e.g., Empty Activity).
  - Enter the application name, package name, and select the minimum Android version.
  - Click “Finish” to create the project.
- **Request Permissions in `AndroidManifest.xml`:**
- **Modify** activity_main.xml **to Display Location:**
- Implement Location Retrieval Logic in Java/Kotlin:
- Handle Permission Requests:
- **Test the Application:**
  - Run the application on an emulator with a simulated GPS location.
  - Alternatively, test on a physical device with GPS enabled.
- **Debug and Optimize:**
  - Check logs in Logcat for debugging.
  - Improve error handling and UI updates for better user experience.