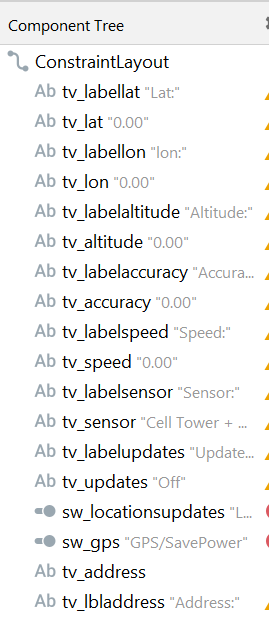
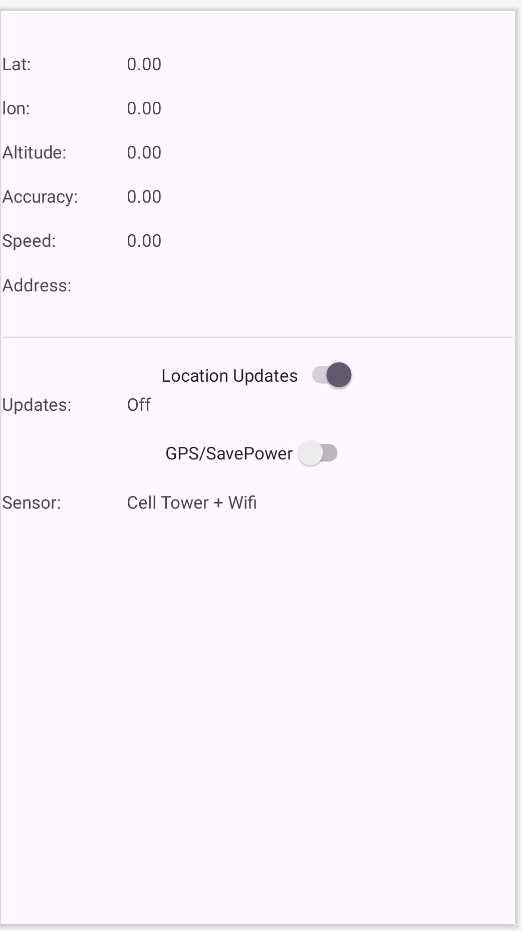
Choose API34. (At least API 31 to run the program)

Step1: Make the layout:



*<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <TextView  
 android:id="@+id/tv\_labellat"  
 android:layout\_width="100dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="Lat:"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/tv\_lat"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 android:text="0.00"  
 app:layout\_constraintStart\_toEndOf="@+id/tv\_labellat"  
 app:layout\_constraintTop\_toTopOf="parent" />

………(to be completed)

<Switch  
 android:id="@+id/sw\_locationsupdates"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="16dp"  
 android:checked="true"  
 android:text="Location Updates"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/divider" />

………(to be completed)

<View  
 android:id="@+id/divider"  
 android:layout\_width="409dp"  
 android:layout\_height="1dp"  
 android:layout\_marginTop="32dp"  
 android:background="?android:attr/listDivider"  
 app:layout\_constraintTop\_toBottomOf="@+id/tv\_address"  
 tools:layout\_editor\_absoluteX="1dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Step2:

Add dependencies in build.gradle.kts:

*implementation*("com.google.android.gms:play-services-location:17.0.0")

Step3:

Add user permission in AndroidManifest.xml:

<uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"/>

**Step4:**

**Coding in MainActivity.java:**

Define variables:

TextView tv\_lat, tv\_lon, tv\_altitude, tv\_accuracy, ………(to be completed)

private static final int *PERMISSIONS\_FINE\_LOCATION*=99;  
 Switch sw\_locationupdates, sw\_gps;  
 boolean updateOn = false;  
 LocationRequest locationRequest;

FusedLocationProviderClient fusedLocationProviderClient;

@Override

**Coding in OnCreate:**

tv\_lat=findViewById(R.id.*tv\_lat*);  
tv\_lon=findViewById(R.id.*tv\_lon*);  
tv\_altitude=findViewById(R.id.*tv\_altitude*);  
………(to be completed)

sw\_gps=findViewById(R.id.*sw\_gps*);  
sw\_locationupdates=findViewById(R.id.*sw\_locationsupdates*);  
  
locationRequest = new LocationRequest();  
locationRequest.setInterval(30000);  
locationRequest.setFastestInterval(5000);  
locationRequest.setPriority(LocationRequest.*PRIORITY\_BALANCED\_POWER\_ACCURACY*);

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");  
 }  
  
 }  
  
});  
updateGPS();

}

**Step 5 Check permission grant:**

@Override  
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {  
 super.onRequestPermissionsResult(requestCode, permissions, grantResults);  
  
 switch (requestCode){  
 case *PERMISSIONS\_FINE\_LOCATION*:  
 if (grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){  
 updateGPS();  
 }else{  
 Toast.*makeText*(this, "This app requires permission", Toast.*LENGTH\_SHORT*).show();  
 finish();  
 }  
 break;  
 }  
  
  
}

**Step 6: Add update GPS function:**

private void updateGPS(){  
  
 fusedLocationProviderClient = LocationServices.*getFusedLocationProviderClient*(MainActivity.this);  
if (ActivityCompat.*checkSelfPermission*(MainActivity.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);  
 }  
 });  
}else{  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*){  
 requestPermissions(new String[] {android.Manifest.permission.*ACCESS\_FINE\_LOCATION*},*PERMISSIONS\_FINE\_LOCATION*);  
 }  
}  
}

**Step 7:Add update UI Value:**

private void updateUIValues(Location location){  
 tv\_lat.setText(String.*valueOf*(location.getLatitude()));  
 tv\_lon………(to be completed)

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…….  
 }else{  
 tv\_speed.setText("Not available");}  
  
}

Step 8: update UI value with Address

Geocoder geocoder =new Geocoder(MainActivity.this);  
  
try {  
 List<Address> addresses=geocoder.getFromLocation(location.getLatitude(),  
 location.getLongitude(),1);  
 tv\_address.setText(addresses.get(0).getAddressLine(0));  
}  
catch (Exception e)  
  
{  
 tv\_address.setText("Unable to get street address");  
}