Practical:11

AIM-Consider Android Application created in Practical-10 and add Sqlite feature such that it is storing data of persons which is received in JSON Format. There should be two buttons: after pressing first button then data of persons loads from sqlite database. After pressing second button, it receives data from network database in JSON format and loads data in ListView or RecyclerView.

1. Create MainActivity according to below UI design.
2. Follow steps and Copy codes from Practical-10
3. Create Class DatabaseHelper for Sqlite Database
4. Create class to store Companion Object for Sqlite database table name, column names.
5. Add some supported function in MainActivity for Sqlite database.
6. Add main_menu in menu folder of resource folder.
7. Add main_menu.xml file to toolbar of Activity as Option menu.
8. Add two buttons with vector icons.
9. Call appropriate method of mainactivity after pressing buttons of toolbar.

Submitted By:- Harshil_Ghadiya
Enrollment number:- 21012021031

Department of Information Technology

Activity_main.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns: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"
android:orientation="vertical"
android:elevation="10dp">


<com.google.android.material.appbar.AppBarLayout
android:layout_width="match_parent"
android:layout_height="wrap_content">
<com.google.android.material.appbar.MaterialToolbar
android:id="@+id/toolbar"
android:layout_width="match_parent"
android:layout_height="?attr/actionBarSize"
app:menu="@menu/menu">
<TextView
android:layout_width="match_parent"
android:layout_height="match_parent"
android:text="@string/app_name"
android:textSize="16sp"
android:textStyle="bold"
android:gravity="center_vertical"/>
</com.google.android.material.appbar.MaterialToolbar>
</com.google.android.material.appbar.AppBarLayout>

<androidx.recyclerview.widget.RecyclerView
android:id="@+id/recyclerView"
android:layout_width="match_parent"
android:layout_height="match_parent" />


<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal"
android:elevation="20dp"
android:layout_marginTop="-80dp"
android:layout_marginRight="20dp"
android:layout_gravity="end">
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
android:id="@+id/btnSwap"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/baseline_autorenew_24"
app:fabCustomSize="60dp"
tools:ignore="SpeakableTextPresentCheck" />
</LinearLayout>

</LinearLayout>
```

Activity_maps.xml :-

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

Contact_item.xml :-

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="wrap_content"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:orientation="vertical">

<com.google.android.material.card.MaterialCardView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:cardElevation="10dp"
android:layout_margin="10dp"
android:layout_gravity="center">

<LinearLayout
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:orientation="horizontal">
```

```
<ImageView
android:id="@+id/imageView"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_margin="10dp"
android:src="@drawable/baseline_person_24"
android:background="@drawable/round_shape"/>


<LinearLayout
android:layout_width="240dp"
android:layout_height="wrap_content"
android:orientation="vertical"
android:layout_margin="5dp">
<TextView
android:id="@+id/name"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="16sp"
android:textStyle="bold"
android:text="Guerra Rodgers"/>

<TextView
android:id="@+id/mobile"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="14sp"
android:text="+919289635723"/>

<TextView
android:id="@+id/emailid"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="14sp"
android:text="guerra_rodgers@gnu.ac.in"/>

<TextView
android:id="@+id/address"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="88 College Place, Umapine, Oregon"
android:textSize="14sp" />
</LinearLayout>

<LinearLayout
```

```
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
android:gravity="center">
<ImageView
android:id="@+id/btnLocation"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginRight="10dp"
android:layout_marginLeft="10dp"
android:layout_marginBottom="5dp"
android:background="@drawable/blue_round_shape"
android:src="@drawable/baseline_location_on_24"/>

<ImageView
android:id="@+id/btnDelete"
android:layout_width="40dp"
android:layout_height="40dp"
android:layout_gravity="center"
android:layout_marginRight="10dp"
android:layout_marginLeft="10dp"
android:background="@drawable/red_round_shape"
android:src="@drawable/baseline_delete_24"/>

</LinearLayout>

</LinearLayout>
</com.google.android.material.card.MaterialCardView>
</LinearLayout>
```

Menu.xml:-

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

<item
android:id="@+id/sqliteDB"
android:title="Button 1"
app:showAsAction="always"
android:icon="@drawable/baseline_window_24" />

<item
```

```
android:id="@+id/jsonDB"
android:title="Button 2"
app:showAsAction="always"
android:icon="@drawable/baseline_autorenew_24" />
</menu>
```

MainActivity.kt:-

```
package com.example.mad_practical_11_21012021031

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.Menu
import android.view.MenuItem
import android.widget.Toast
import androidx.appcompat.widget.Toolbar
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.floatingactionbutton.FloatingActionButton
importkotlinx.coroutines.CoroutineScope
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import org.json.JSONArray
import org.json.JSONException
import org.json.JSONObject

class MainActivity : AppCompatActivity() {
lateinit var recyclerView : RecyclerView
lateinit var databaseHelper: DatabaseHelper
   override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
setContentView(R.layout.activity_main)

databaseHelper = DatabaseHelper(this)
valtoolBar : Toolbar = findViewById(R.id.toolbar)

setSupportActionBar(toolBar)

valfetchBtn : FloatingActionButton = findViewById(R.id.btnSwap)

recyclerView = findViewById(R.id.recyclerView)
fetchBtn.setOnClickListener{
CoroutineScope(Dispatchers.IO).launch {
try {
```

```
val data = HttpRequest().makeServiceCall(
                    "https://api.json-generator.com/templates/qjeKFdjkXCdK/data",
                    "rbn0rerl1k0d3mcwgw7dva2xuwk780z1hxvyvrb1"
                )
withContext(Dispatchers.Main) {
try {
                        if(data != null)
                        {
runOnUiThread{getPersonDetailsFromJson(data)}
}
                    }
                    catch (e: Exception)
                    {
e.printStackTrace()
                    }
}
}
            catch (e: Exception)
            {
e.printStackTrace()
            }
}
      }
}

   override fun onCreateOptionsMenu(menu: Menu): Boolean {
menuInflater.inflate(R.menu.menu, menu)
      return true
   }

   override fun onOptionsItemSelected(item: MenuItem): Boolean {
      when (item.itemId) {
R.id.sqliteDB-> {
Toast.makeText(this@MainActivity, "Clicked on item at menu!",
Toast.LENGTH_SHORT).show()
            return true
         }
R.id.jsonDB-> {
valpersonList: ArrayList<Person> = databaseHelper.getAllPersons()
recyclerView.layoutManager= LinearLayoutManager(this)
recyclerView.adapter= PersonAdapter(this, personList)
            return true
         }
         else -> return super.onOptionsItemSelected(item)
      }
   }
```

```
    private fun getPersonDetailsFromJson(sJson: String?)
    {
valpersonList = ArrayList<Person>()
        try {
valjsonArray = JSONArray(sJson)
            for(i in 0 until jsonArray.length())
            {
valjsonObject = jsonArray[i] as JSONObject
val person = Person(jsonObject)
personList.add(person)
            }
recyclerView.layoutManager= LinearLayoutManager(this)
recyclerView.adapter= PersonAdapter(this, personList)
        }
        catch (e: JSONException)
        {
e.printStackTrace()
        }
    }
}
```

MapsActivity.kt :-

```
package com.example.mad_practical_11_21012021031

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import com.example.mad_practical_11_21012021031.databinding.ActivityMapsBinding

importcom.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

class MapsActivity : AppCompatActivity(), OnMapReadyCallback {

    private lateinit var mMap: GoogleMap
    private lateinit var binding: ActivityMapsBinding

    private val TAG = "MapActivity"
    private var lat = -34.0
    private var log = 151.0
    private var title = "Marker in Sydney"
```

```
    override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)

valobj = intent.getSerializableExtra("Object") as Person
Log.i(TAG, "onCreate: Object:$obj")
lat = obj.latitude
        log = obj.longitude
        title = obj.name

        binding = ActivityMapsBinding.inflate(layoutInflater)
setContentView(binding.root)

        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
valmapFragment = supportFragmentManager
.findFragmentById(R.id.map) as SupportMapFragment
mapFragment.getMapAsync(this)
    }

/**
     * Manipulates the map once available.
     * This callback is triggered when the map is ready to be used.
     * This is where we can add markers or lines, add listeners or move the camera. In this case,
     * we just add a marker near Sydney, Australia.
     * If Google Play services is not installed on the device, the user will be prompted to install
     * it inside the SupportMapFragment. This method will only be triggered once the user has
     * installed Google Play services and returned to the app.
     */
override fun onMapReady(googleMap: GoogleMap) {
mMap = googleMap

        // Add a marker in Sydney and move the camera
valsydney = LatLng(lat, log)
mMap.addMarker(MarkerOptions().position(sydney).title(title))
//        mMap.moveCamera(CameraUpdateFactory.newLatLng(sydney))
mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(sydney, 8.0f))
    }
}
```

Person.kt :-

```
package com.example.mad_practical_11_21012021031

import org.json.JSONObject
import java.io.Serializable
```

```
class Person (var id: String,
              var name: String,
              var emailId: String,
              var phoneNo: String,
              var address: String,
              var latitude: Double,
              var longitude: Double):Serializable{
    constructor(jsonObject: JSONObject):this("","","","","",0.0,0.0) {
        id = jsonObject.getString("id")
emailId = jsonObject.getString("email")
phoneNo = jsonObject.getString("phone")
valprofileJson = jsonObject.getJSONObject("profile")
        name = profileJson.getString("name")
        address = profileJson.getString("address")
vallocationJson = profileJson.getJSONObject("location")
        latitude = locationJson.getDouble("lat")
        longitude = locationJson.getDouble("long")
    }
}
```

HttpRequest.kt :-

```
package com.example.mad_practical_11_21012021031

import android.util.Log
import java.io.BufferedInputStream
import java.io.BufferedReader
import java.io.IOException
import java.io.InputStream
import java.io.InputStreamReader
import java.lang.Exception
import java.lang.StringBuilder
import java.net.HttpURLConnection
import java.net.MalformedURLException
import java.net.ProtocolException
import java.net.URL

class HttpRequest {
    private val TAG = "HttpRequest"

    fun makeServiceCall(reqUrl: String?, token: String?=null): String? {
        var response: String? = null
        try {
```

```
valurl = URL(reqUrl)
val conn = url.openConnection() as HttpURLConnection
            if (token != null)
            {
conn.setRequestProperty("Authorization","Bearer $token")
conn.setRequestProperty("Content-Type","application/json")
            }
conn.requestMethod= "GET"
            response = convertStreamToString(BufferedInputStream(conn.inputStream))
        }
        catch (e : MalformedURLException)
        {
Log.e(TAG, "MalformedURLException: " + e.message)
        }
        catch (e : ProtocolException)
        {
Log.e(TAG, "ProtocolException: " + e.message)
        }
        catch (e : IOException)
        {
Log.e(TAG, "IOException: " + e.message)
        }
        catch (e: Exception)
        {
Log.e(TAG, "Exception: " + e.message)
        }
        return response
    }

    private fun convertStreamToString(`is`: InputStream):String
    {
val reader = BufferedReader(InputStreamReader(`is`))
valsb = StringBuilder()
        var line: String?=null
        try {
            while (reader.readLine().also { line = it } != null)
            {
sb.append(line).append('\n')
            }
        }
        catch (e : IOException)
        {
Log.i(TAG, "convertStreamToString: $line")
e.printStackTrace()
        }
        finally {
            try {
```

```
            `is`.close()
          }
          catch (e: IOException)
          {
e.printStackTrace()
          }
        }
        return sb.toString()
    }
}
```

PersonAdapter.kt :-

```
package com.example.mad_practical_11_21012021031

import android.annotation.SuppressLint
import android.content.Context
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import android.widget.Toast
import androidx.recyclerview.widget.RecyclerView
import java.io.Serializable
class PersonAdapter(private val context: Context, private val array: ArrayList<Person>):
RecyclerView.Adapter<PersonAdapter.PersonViewHolder>(){
lateinit var databaseHelper: DatabaseHelper
init {
        // Initialize the databaseHelper here
databaseHelper = DatabaseHelper(context)
    }
    inner class PersonViewHolder(valitemView: View): RecyclerView.ViewHolder(itemView)
    {
valnameTxt : TextView = itemView.findViewById(R.id.name)
valemailTxt : TextView = itemView.findViewById(R.id.emailid)
valphoneTxt : TextView = itemView.findViewById(R.id.mobile)
valaddressTxt : TextView = itemView.findViewById(R.id.address)
valmapBtn : ImageView = itemView.findViewById(R.id.btnLocation)
valdeleteBtn : ImageView = itemView.findViewById(R.id.btnDelete)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): PersonViewHolder {
val view = LayoutInflater.from(parent.context).inflate(R.layout.contact_item, parent, false)
        return PersonViewHolder(view)
    }
```

```
    override fun getItemCount(): Int {
        return array.size
    }
    @SuppressLint("NotifyDataSetChanged")
    override fun onBindViewHolder(holder: PersonViewHolder, position: Int) {
val person = array[position]
holder.nameTxt.text= person.name
holder.emailTxt.text= person.emailId
holder.phoneTxt.text= person.phoneNo
holder.addressTxt.text= person.address
valobj = person as Serializable
databaseHelper.insertPerson(person)
holder.mapBtn.setOnClickListener{
Intent(this@PersonAdapter.context, MapsActivity::class.java).apply {
putExtra("Object",obj)
this@PersonAdapter.context.startActivity(this)
}
      }
holder.deleteBtn.setOnClickListener{
val count = databaseHelper.deletePerson(person.id)
          if(count > 0)
          {
Toast.makeText(this.context, "${person.name}'s details deleted successfully!",
Toast.LENGTH_SHORT).show()
array.removeAt(position)
notifyDataSetChanged()
          }
}
}
}
```

DatabaseHelper.kt :-

```
package com.example.mad_practical_11_21012021031

import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
import org.json.JSONObject
class DatabaseHelper(context: Context): SQLiteOpenHelper(context, DATABASE_NAME, null,
    DATABASE_VERSION){
```

```
    companion object{
        private constval DATABASE_VERSION = 1
        private constval DATABASE_NAME = "persons.db"
        private constval TABLE_NAME = "person"
        private constval COLUMN_ID = "id"
        private constval COLUMN_PERSON_NAME = "person_name"
        private constval COLUMN_PERSON_EMAIL_ID = "person_email_id"
        private constval COLUMN_PERSON_PHONE_NO = "person_phone_no"
        private constval COLUMN_PERSON_ADDRESS = "person_address"
        private constval COLUMN_PERSON_GPS_LAT = "person_lat"
        private constval COLUMN_PERSON_GPS_LONG = "person_long"
    }
    override fun onCreate(db: SQLiteDatabase?) {
val CREATE_TABLE = ("CREATE TABLE " + TABLE_NAME + "("
                + COLUMN_ID + " TEXT PRIMARY KEY,"
                + COLUMN_PERSON_NAME + " TEXT,"
                + COLUMN_PERSON_EMAIL_ID + " TEXT,"
                + COLUMN_PERSON_PHONE_NO + " TEXT,"
                + COLUMN_PERSON_ADDRESS + " TEXT,"
                + COLUMN_PERSON_GPS_LAT + " REAL,"
                + COLUMN_PERSON_GPS_LONG + " REAL)")

        if (db != null) {
db.execSQL(CREATE_TABLE)
        }
    }
    override fun onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) {
        if (db != null) {
db.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        }
    }
    fun insertPerson(person: Person) : Long
    {
valdb =writableDatabase
valcontentValues = ContentValues()
contentValues.put(COLUMN_ID,person.id)
contentValues.put(COLUMN_PERSON_NAME,person.name)
contentValues.put(COLUMN_PERSON_EMAIL_ID,person.emailId)
contentValues.put(COLUMN_PERSON_PHONE_NO,person.phoneNo)
contentValues.put(COLUMN_PERSON_ADDRESS,person.address)
contentValues.put(COLUMN_PERSON_GPS_LAT,person.latitude)
contentValues.put(COLUMN_PERSON_GPS_LONG,person.longitude)
val count = db.insert(TABLE_NAME, null, contentValues)
db.close()
        return count
    }
    fun deletePerson(personId: String) : Int
```

```
    {
valdb = writableDatabase
val selection ="$COLUMN_ID = ?"
valselectionArgs = arrayOf(personId)
val count =db.delete(TABLE_NAME,selection,selectionArgs)
db.close()
        return count
    }
    @SuppressLint("Range")
    fun getAllPersons() :ArrayList<Person>
    {
valpersonList = arrayListOf<Person>()
valdb =readableDatabase
var query = "SELECT * FROM $TABLE_NAME"
        var cursor : Cursor =db.rawQuery(query,null)
        while (cursor.moveToNext())
        {
            var id : String = cursor.getString(cursor.getColumnIndex(COLUMN_ID))
            var name : String =
cursor.getString(cursor.getColumnIndex(COLUMN_PERSON_NAME))
            var email : String =
cursor.getString(cursor.getColumnIndex(COLUMN_PERSON_EMAIL_ID))
            var phone : String =
cursor.getString(cursor.getColumnIndex(COLUMN_PERSON_PHONE_NO))
            var address: String =
cursor.getString(cursor.getColumnIndex(COLUMN_PERSON_ADDRESS))

            var latitude : Double =
cursor.getDouble(cursor.getColumnIndex(COLUMN_PERSON_GPS_LAT))
            var longitude: Double =
cursor.getDouble(cursor.getColumnIndex(COLUMN_PERSON_GPS_LONG))
valjsonObject = JSONObject()
jsonObject.put("id", id)
jsonObject.put("email", email)
jsonObject.put("phone", phone)
valprofileJson = JSONObject()
profileJson.put("name", name) // You'll need to fill in the actual name value here
profileJson.put("address", address)
vallocationJson = JSONObject()
locationJson.put("lat", latitude)
locationJson.put("long", longitude)
profileJson.put("location", locationJson)
jsonObject.put("profile", profileJson)
val person = Person(jsonObject)
personList.add(person)
        }
cursor.close()
```

```
db.close()
        return personList
    }
}
```

PersonDBTableData.kt :-

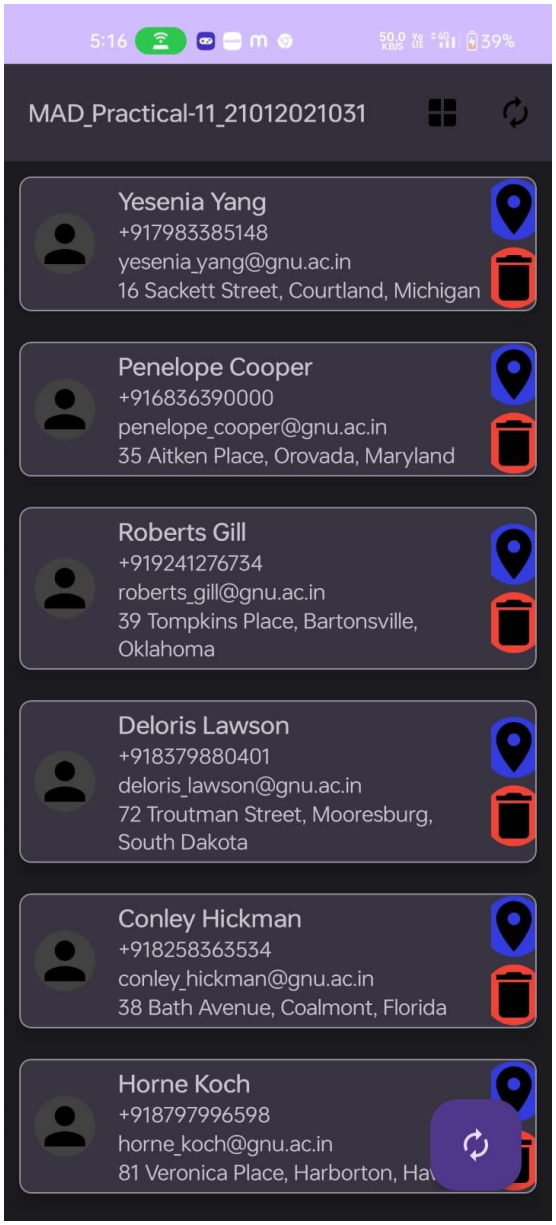
```
package com.example.mad_practical_11_21012021031

class PersonDBTableData {
    companion object {
constval TABLE_NAME = "persons"
constval COLUMN_ID = "id"
constval COLUMN_PERSON_NAME = "person_name"
constval COLUMN_PERSON_EMAIL_ID = "person_email_id"
constval COLUMN_PERSON_PHONE_NO = "person_phone_no"
constval COLUMN_PERSON_ADDRESS = "person_address"
constval COLUMN_PERSON_GPS_LAT = "person_lat"
constval COLUMN_PERSON_GPS_LONG = "person_long"

val CREATE_TABLE = ("CREATE TABLE "+ TABLE_NAME + "("
                + COLUMN_ID + " TEXT PRIMARY KEY,"
                + COLUMN_PERSON_NAME + " TEXT,"
                + COLUMN_PERSON_EMAIL_ID + " TEXT,"
                + COLUMN_PERSON_PHONE_NO + " TEXT,"
                + COLUMN_PERSON_ADDRESS + " TEXT,"
                + COLUMN_PERSON_GPS_LAT + " REAL,"
                + COLUMN_PERSON_GPS_LONG + " REAL"
                + ")")
    }
}
```

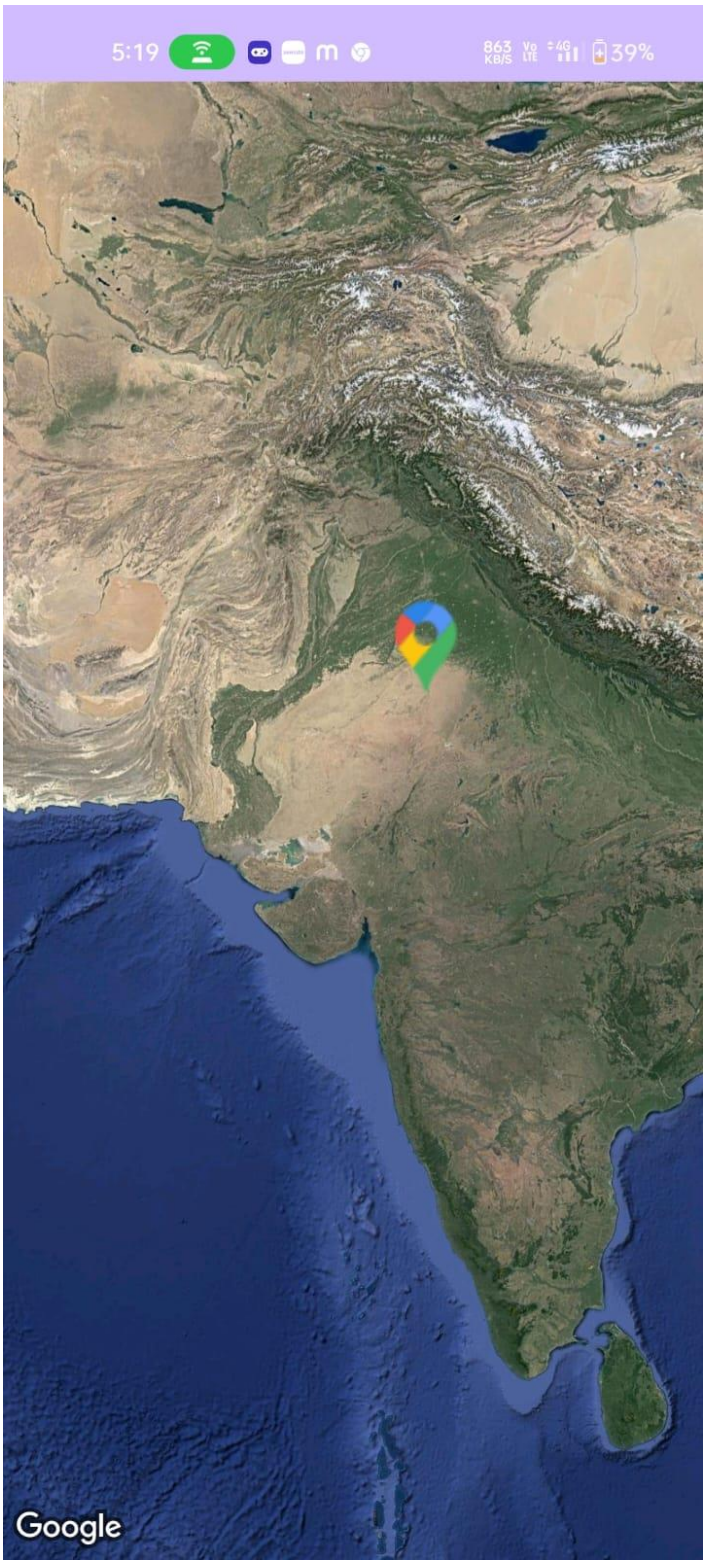
AndroidManifest.xml :-

```
<meta-data
android:name="com.google.android.geo.API_KEY"
android:value="AIzaSyBVgO1713xHQ9FVaLQiV1Pp0AiA3ndOckw" />
```

Output :-


5:16
39%
MAD_Practical-11_21012021031


5:16
39%
MAD_Practical-11_21012021031
Yesenia Yang
+917983385148
yesenia_yang@gnu.ac.in
16 Sackett Street, Courtland, Michigan
Penelope Cooper
+916836390000
penelope_cooper@gnu.ac.in
35 Aitken Place, Orovada, Maryland
Roberts Gill
+919241276734
roberts_gill@gnu.ac.in
39 Tompkins Place, Bartonsville, Oklahoma
Deloris Lawson
+918379880401
deloris_lawson@gnu.ac.in
72 Troutman Street, Mooresburg, South Dakota
Conley Hickman
+918258363534
conley_hickman@gnu.ac.in
38 Bath Avenue, Coalmont, Florida
Horne Koch
+918797996598
horne_koch@gnu.ac.in


5:19
39%
Google