

Lab 4: Google Map

Objective

- To implement google map in android application

Lab work

MainActivity.java

```
package com.example.googlemap;
```

```
import android.*;
import android.app.*;
import android.content.pm.*;
import android.location.*;
import android.os.Bundle;
import androidx.annotation.*;
import androidx.appcompat.app.*;
import androidx.core.content.*
import com.google.android.gms.location.*;
import com.google.android.gms.maps.*;
import com.google.android.gms.tasks.*;

// google map sdk
public class MapActivity extends AppCompatActivity implements OnMapReadyCallback {

    private GoogleMap googleMap;
    private FusedLocationProviderClient locationProviderClient;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_map);
        locationProviderClient = LocationServices.getFusedLocationProviderClient(this);
        if (ContextCompat.checkSelfPermission(this,
            android.Manifest.permission.ACCESS_FINE_LOCATION) !=
            PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 1);
        }
        SupportMapFragment mapFragment = (SupportMapFragment)
        getSupportFragmentManager().findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
    }
    @Override public void onMapReady(@NonNull GoogleMap googleMap) {
        this.googleMap = googleMap;
        UpdateLocation();
    }
    @Override
    public void onRequestPermissionsResult(int requestCode,
```

```

@NonNull
String[] permissions,
int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 1) {
        if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
            UpdateLocation();
        }
    }
}

private void UpdateLocation() {
    if (ActivityCompat.checkSelfPermission(this, android.Manifest.permission.ACCESS_FINE_LOCATION)
    != PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission(this,
    android.Manifest.permission.ACCESS_COARSE_LOCATION) !=
        PackageManager.PERMISSION_GRANTED) {
        return;
    }
    locationProviderClient.getLastLocation().addOnSuccessListener(this, new
    OnSuccessListener<Location>() {
        @Override
        \
        public void onSuccess(Location location) {if (location == null) {
return;
        }
        LatLng currentLat = new LatLng(location.getLatitude(),location.getLongitude());
        googleMap.addMarker(new MarkerOptions().position(currentLat).title("you is here"));
        googleMap.moveCamera(CameraUpdateFactory.newLatLngZoom(currentLat,15));
        }
    });
}
}
}

```

activity_main.xml

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

```

<androidx.fragment.app.FragmentContainerView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:name="com.google.android.gms.maps.SupportMapFragment"
    android:id="@+id/map"
    xmlns:android="http://schemas.android.com/apk/res/android" />

```

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.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:supportRtl="true"
android:theme="@style/Theme.GoogleMap"
tools:targetApi="31">

    <meta-data android:name="com.google.android.geo.API_KEY"
        android:value="AIzaSyBiK29ZZJ9hjSho0c9UhX1hKujvygh8UVY"/>

    <activity android:name=".MapActivity" android:exported="true">
        <intent-filter >
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>

    </activity>

</application>

</manifest>
```

Output:



Discussion and conclusion:

In this lab, we implement google map using API key. We implement the OnMapReadyCallBack method to add google map feature in the application. In this lab we try to pin points on our own current location and add marker there as You are here.

Lab 5: Simple Sqlite operations on android

1. Create a sqlite database named “noteApp.db”.
2. Create a table named “notes” with following columns:
 - a. `_id` (autoincrement primary key)
 - b. UUID string
 - c. Title string
 - d. Description string
3. Create two buttons in MainActivity
 - a. Insert
 - b. List
4. When clicked on insert, make a database operation to insert dummy data on notes table.
 - a. Insert at least 5 dummy notes
5. When clicked on List, make a database operation to query all data on notes table and show it in a listView just below the buttons.

MainActivity.java

```
package com.example.noteapp;
```

```
import android.database.*;
import android.os.*;
import android.view.*;
import android.widget.*;
import androidx.appcompat.app.*;
import java.util.*;
```

```
public class MainActivity extends AppCompatActivity {
    private NoteDbHelper dbHelper;
    private ListView listView;
```

```
    @Override protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        dbHelper = new NoteDbHelper(this);
        listView = findViewById(R.id.listView);
        Button insertButton = findViewById(R.id.insertButton);
        Button listButton = findViewById(R.id.listButton);
        // Insert dummy notes when clicking the insert button
        insertButton.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
                public void onClick(View v) {
                    dbHelper.insertDummyData();
                }
        });
```

```

        // List all notes when clicking the list button
        listButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                List<String> notes = getAllNotes();
                ArrayAdapter<String> adapter = new ArrayAdapter<>(MainActivity.this,
                    android.R.layout.simple_list_item_1, notes);
                listView.setAdapter(adapter);
            }
        });
    }

    // Retrieve all notes from the database and convert to a list
    private List<String> getAllNotes() {
        List<String> notesList = new ArrayList<>();
        Cursor cursor = dbHelper.getAllNotes();

        if (cursor.moveToFirst()) {
            do {
                String title = cursor.getString(cursor.getColumnIndex("title"));
                String description = cursor.getString(cursor.getColumnIndex("description"));
                notesList.add(title + ": " + description);
            } while (cursor.moveToNext());
        }
        cursor.close();
        return notesList;
    }
}

```

NoteDbHelper.java

```
package com.example.noteapp;
```

```

import android.content.*;
import android.database.*;
import android.database.sqlite.*;
import android.widget.*;
import java.util.UUID;

```

```

public class NoteDbHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "noteApp.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NOTES = "notes";

```

```

    // Columns
    private static final String COLUMN_ID = "_id";
    private static final String COLUMN_UUID = "uuid";
    private static final String COLUMN_TITLE = "title";
    private static final String COLUMN_DESCRIPTION = "description";

```

```

    private final Context context;
    public NoteDbHelper(Context context) {
super(context, DATABASE_NAME, null, DATABASE_VERSION);
this.context = context;
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NOTES + " (" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_UUID + " TEXT, " +
            COLUMN_TITLE + " TEXT, " +
            COLUMN_DESCRIPTION + " TEXT)";
        db.execSQL(createTable);
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
db.execSQL("DROP TABLE IF EXISTS " + TABLE_NOTES);
onCreate(db);
    }
    @Override
    public void onOpen(SQLiteDatabase db) {
super.onOpen(db);
// Clear all previous data
        db.execSQL("DELETE FROM " + TABLE_NOTES);
    }
// Insert Dummy Data with unique descriptions
    public void insertDummyData() {
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        // Array of unique descriptions
        String[] descriptions = {
            "Advance Java Programming Note",
            "Mobile Programming Note",
            "Network Programming Note",
            "Applied Economics Note",
            "Distributed System Note"
        };

        for (int i = 1; i <= descriptions.length; i++) {
            contentValues.put(COLUMN_UUID, UUID.randomUUID().toString());
            contentValues.put(COLUMN_TITLE, "Note " + i);
            contentValues.put(COLUMN_DESCRIPTION, descriptions[i - 1]);
// Assign unique description
            db.insert(TABLE_NOTES, null, contentValues);
        }
// Use the context to show a Toast message
        Toast.makeText(context, "Data inserted", Toast.LENGTH_SHORT).show();
    }

```

```

    }

    // Insert Dummy Data with dynamic descriptions (optional)
    public void insertDynamicDummyData() {

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues contentValues = new ContentValues();

        for (int i = 1; i <= 5; i++) {
            String uniqueDescription = "CodyNoteApp " + i + " - " + System.currentTimeMillis();
            contentValues.put(COLUMN_UUID, UUID.randomUUID().toString());
            contentValues.put(COLUMN_TITLE, "Note " + i);
            contentValues.put(COLUMN_DESCRIPTION, uniqueDescription);
            db.insert(TABLE_NOTES, null, contentValues);
        }

        Toast.makeText(context, "Dynamic data inserted", Toast.LENGTH_SHORT).show();
    }
    // Query All Notes
    public Cursor getAllNotes() {
        SQLiteDatabase db = this.getReadableDatabase();
        return db.rawQuery("SELECT * FROM " + TABLE_NOTES, null);
    }
}
activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Note Application"
        android:textSize="25dp"
        android:textAlignment="center"
        android:layout_margin="16dp"
        />

    <Button
        android:id="@+id/insertButton"
        android:layout_width="wrap_content"

```



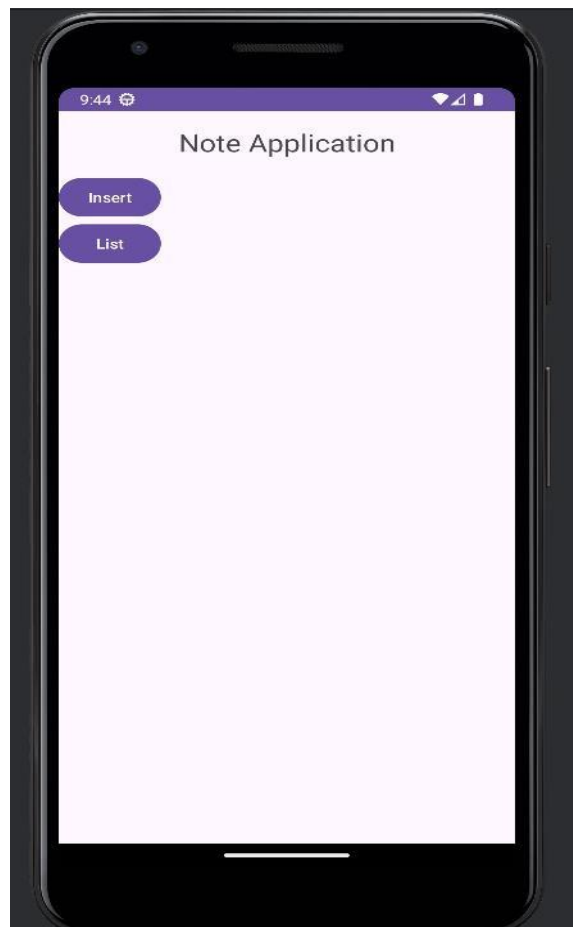
```
android:layout_height="wrap_content"  
android:text="Insert" />
```

```
<Button  
android:id="@+id/listButton"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:text="List"  
/>
```

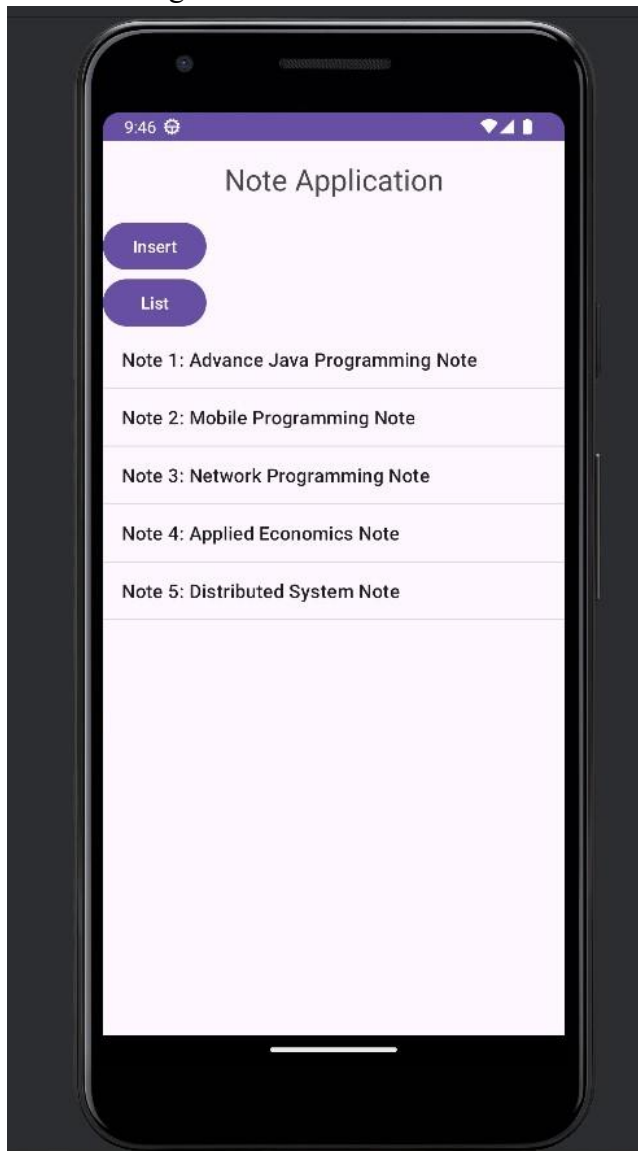
```
<ListView  
android:id="@+id/listView"  
android:layout_width="match_parent"  
android:layout_height="match_parent"  
/>  
</LinearLayout>
```

Output:

First view of the app



After clicking on List button we can see list of an item like this



Conclusion:

In this lab, we learn about the database SQLite to store data. We insert dummy data in the database and show that data in list form. We create two button as Insert and List. The insert button function as it help to insert data in the database and List button function as to show the list of item store in the database.

Lab:6

Develop an android application to input your Name, Age, gender, email address, phone number and a submit button. When clicked on the button, show this information on another activity. Perform given validation on the following input fields:

Name:	Must not be empty, must satisfy the following expression: "FirstName LastName".
Email	Must not be empty, must be a valid email address
Phone Number	Must not be empty
Gender	One gender must be selected

Note: You can use Google's material UI for better looking Input fields and error messages.

MainActivity.java

```
package com.example.personalinfoapp;

import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Patterns;
import android.view.View;
import android.widget.*;

import androidx.appcompat.app.AppCompatActivity;
import com.google.android.material.textfield.TextInputEditText;

public class MainActivity extends AppCompatActivity {
    private TextInputEditText nameInput, ageInput, emailInput, phoneInput;
    private RadioGroup genderRadioGroup;
    private Button submitButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize Views
        nameInput = findViewById(R.id.inputName);
```

```

ageInput = findViewById(R.id.inputAge);
emailInput = findViewById(R.id.inputEmail);
phoneInput = findViewById(R.id.inputPhone);
genderRadioGroup = findViewById(R.id.genderRadioGroup);
submitButton = findViewById(R.id.submitButton);

submitButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (validateInputs()) {
            // Get selected gender
            int genderId = genderRadioGroup.getCheckedRadioButtonId();
            RadioButton selectedGender = findViewById(genderId);

            // Create intent to pass data to SecondActivity
            Intent intent = new Intent(MainActivity.this, SecondActivity.class);
            intent.putExtra("name", nameInput.getText().toString());
            intent.putExtra("age", ageInput.getText().toString());
            intent.putExtra("email", emailInput.getText().toString());
            intent.putExtra("phone", phoneInput.getText().toString());
            intent.putExtra("gender", selectedGender.getText().toString());
            startActivity(intent);
        }
    }
});

private boolean validateInputs() {
    boolean isValid = true;

    // Name validation
    String namePattern = "^[A-Za-z]+\\s[A-Za-z]+$";
    if (TextUtils.isEmpty(nameInput.getText()) || !nameInput.getText().toString().matches(namePattern)) {
        nameInput.setError("Enter valid full name (FirstName LastName)");
        isValid = false;
    }

    // Email validation
    if (TextUtils.isEmpty(emailInput.getText()) ||
        !Patterns.EMAIL_ADDRESS.matcher(emailInput.getText()).matches()) {
        emailInput.setError("Enter a valid email");
        isValid = false;
    }

    // Gender validation
    if (genderRadioGroup.getCheckedRadioButtonId() == -1) {
        Toast.makeText(this, "Please select a gender", Toast.LENGTH_SHORT).show();
    }
}

```

```

isValid = false;
    }

    return isValid;
} }

```

activity_main.xml

```

<?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">

    <com.google.android.material.textfield.TextInputLayout
    android:id="@+id/inputNameLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Name"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent">

        <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/inputName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>

    <com.google.android.material.textfield.TextInputLayout
    android:id="@+id/inputAgeLayout"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Age"
    app:layout_constraintTop_toBottomOf="@id/inputNameLayout"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent">

        <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/inputAge"
        android:inputType="number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
    </com.google.android.material.textfield.TextInputLayout>

```

```
<com.google.android.material.textfield.TextInputLayout
android:id="@+id/inputEmailLayout"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:hint="Email"
app:layout_constraintTop_toBottomOf="@id/inputAgeLayout"
app:layout_constraintStart_toStartOf="parent"

app:layout_constraintEnd_toEndOf="parent">
```

```
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/inputEmail"
android:inputType="textEmailAddress"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</com.google.android.material.textfield.TextInputLayout>
```

```
<com.google.android.material.textfield.TextInputLayout
android:id="@+id/inputPhoneLayout"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:hint="Phone Number"
app:layout_constraintTop_toBottomOf="@id/inputEmailLayout"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent">
```

```
<com.google.android.material.textfield.TextInputEditText
android:id="@+id/inputPhone"
android:inputType="phone"
android:layout_width="match_parent"
android:layout_height="wrap_content"/>
</com.google.android.material.textfield.TextInputLayout>
```

```
<RadioGroup
android:id="@+id/genderRadioGroup"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
app:layout_constraintTop_toBottomOf="@id/inputPhoneLayout"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent">
```

```
<RadioButton
android:id="@+id/maleRadioButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Male"/>
```

```
<RadioButton
```

```

        android:id="@+id/femaleRadioButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female"/>
    </RadioGroup>

    <Button
        android:id="@+id/submitButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Submit"
        app:layout_constraintTop_toBottomOf="@id/genderRadioGroup"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>

```

SecondActivity.java

```
package com.example.personalinfoapp;
```

```
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class SecondActivity extends AppCompatActivity {
    private TextView nameTextView, ageTextView, emailTextView, phoneTextView, genderTextView;
```

```

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        // Initialize Views
        nameTextView = findViewById(R.id.nameTextView);
        ageTextView = findViewById(R.id.ageTextView);
        emailTextView = findViewById(R.id.emailTextView);
        phoneTextView = findViewById(R.id.phoneTextView);
        genderTextView = findViewById(R.id.genderTextView);
    }

```

```

    // Get data from intent
    String name = getIntent().getStringExtra("name");
    String age = getIntent().getStringExtra("age");
    String email = getIntent().getStringExtra("email");
    String phone = getIntent().getStringExtra("phone");
    String gender = getIntent().getStringExtra("gender");

```

```

    // Set text to TextViews

```

```

nameTextView.setText("Name: " + name);
ageTextView.setText("Age: " + age);
emailTextView.setText("Email: " + email);
phoneTextView.setText("Phone: " + phone);
genderTextView.setText("Gender: " + gender);
    }
}

```

second_activity.xml

```

<?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=".SecondActivity"
>
    <TextView
android:id="@+id/nameTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Name"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"/>

    <TextView
android:id="@+id/ageTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content" android:text="Age"
app:layout_constraintTop_toBottomOf="@id/nameTextView"
app:layout_constraintStart_toStartOf="parent"/>

    <TextView
android:id="@+id/emailTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Email"
app:layout_constraintTop_toBottomOf="@id/ageTextView"
app:layout_constraintStart_toStartOf="parent"/>

    <TextView
android:id="@+id/phoneTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Phone"

```



```
app:layout_constraintTop_toBottomOf="@id/emailTextView"
app:layout_constraintStart_toStartOf="parent"/>
```

```
    <TextView
        android:id="@+id/genderTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Gender"
        app:layout_constraintTop_toBottomOf="@id/phoneTextView"
        app:layout_constraintStart_toStartOf="parent"/>
```

```
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
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">
```

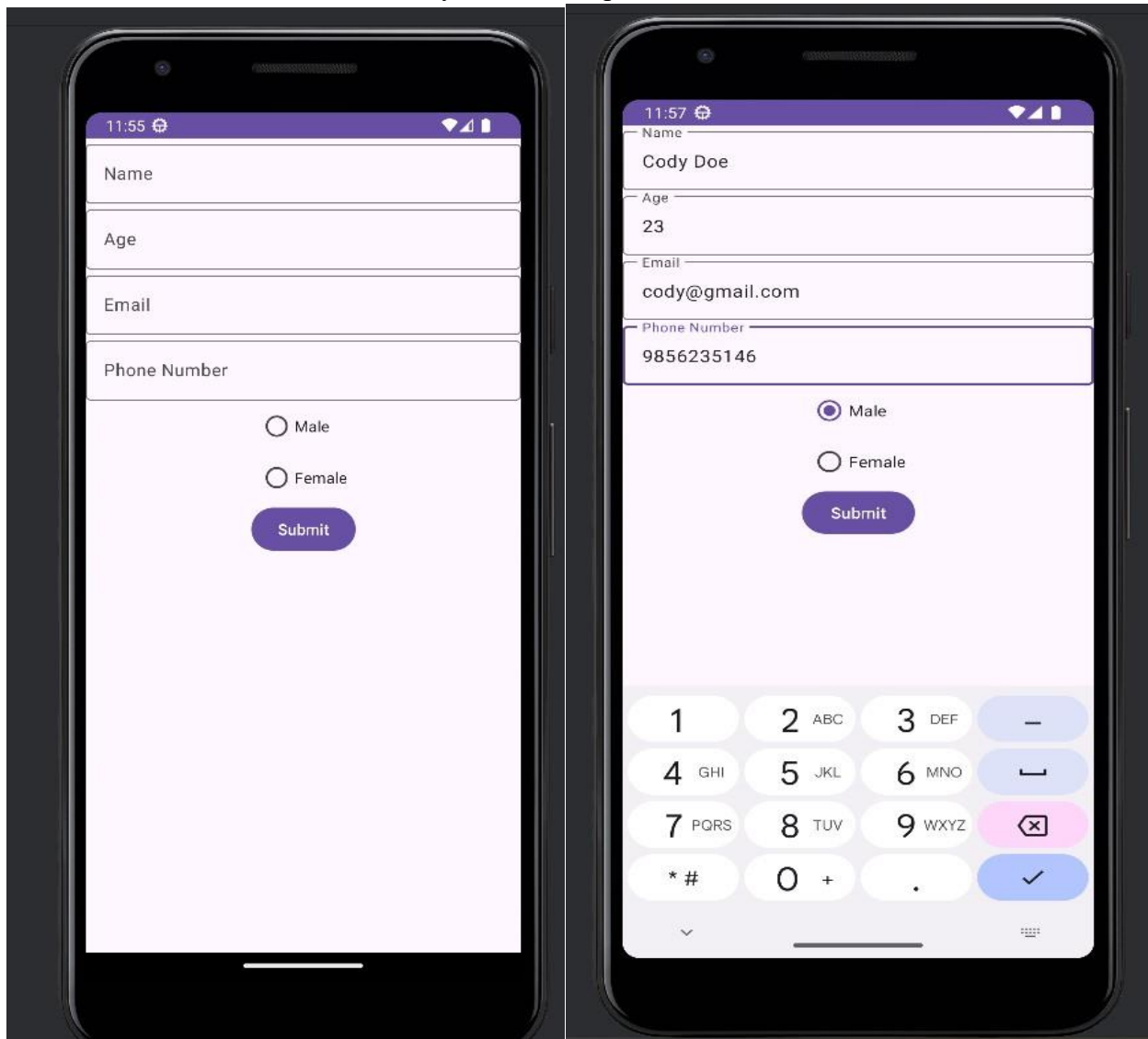
```
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true" android:theme="@style/Theme.PersonalInfoApp" tools:targetApi="31">
        <activity
            android:name=".SecondActivity"
            android:exported="false" />
        <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>

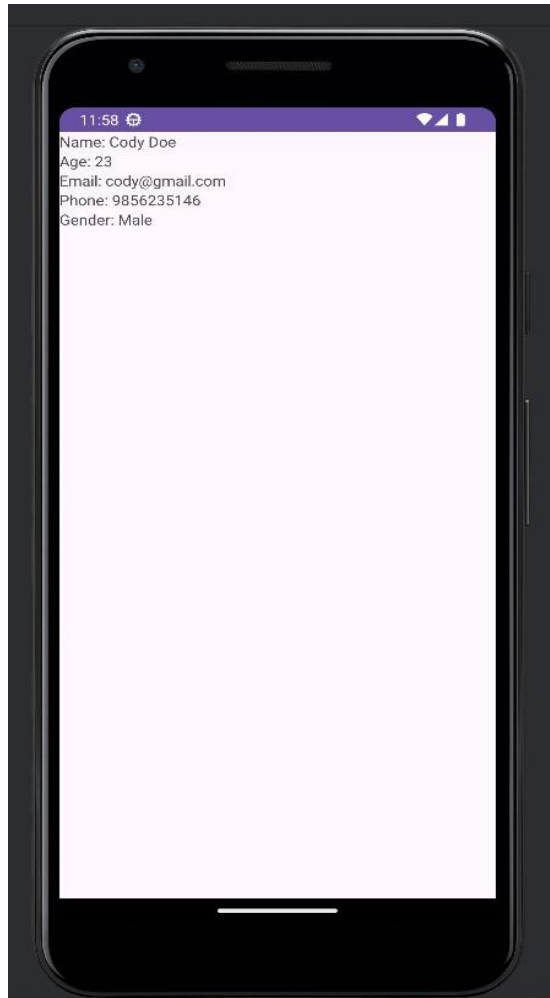
</manifest>
```

Output:

This is the initial view of mainActivity create a simple form



After clicking on submit we goto secondActivity and that activity show the information inputed in the form



Conclusion:

In this lab, we learn about the multiple activity and how to handle them. we create a simple form and validate them and that form information is submitted and show in another activity.

Intent class is used to build connection between to activity. Main activity has simple form and second activity is used to show the details of form input.