**ANDROID LABORATORY - PROGRAMS**

**Part - A**

Q1. In this lab we will be learning how to use and extend the Android user interface library.

a. Views, View Groups, Layouts, and Widgets are and how they relate to each other.

b. How to declare and reference resources in code.

c. How to navigate between multiple activities.

d. How to share the data between the activities.

e. Explore life-cycle methods of an activity.

f. How to use Events and Event Listeners.

g. How to create Toast Notifications.

Code:

**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="com.example.helloworld.MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

package com.example.helloworld;

import androidx.appcompat.app.AppCompatActivity;

//import android.os.Bundle;

//import android.app.Activity;

import android.os.Bundle;

import android.util.Log;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Log.d("lifecycle","onCreate invoked");

}

@Override

protected void onStart() {

super.onStart();

Log.d("lifecycle","onStart invoked");

}

@Override

protected void onResume() {

super.onResume();

Log.d("lifecycle","onResume invoked");

}

@Override

protected void onPause() {

super.onPause();

Log.d("lifecycle","onPause invoked");

}

@Override

protected void onStop() {

super.onStop();

Log.d("lifecycle","onStop invoked");

}

@Override

protected void onRestart() {

super.onRestart();

Log.d("lifecycle","onRestart invoked");

}

@Override

protected void onDestroy() {

super.onDestroy();

Log.d("lifecycle","onDestroy invoked");

}

}

Q2. You will expand on your knowledge of the Android user interface library.

a. How to declare layouts statically as an xml resource.

b. How to create custom Views from scratch to suit a specific need.

c. How to create Options and Context Menus.

d. How to use ListAdapter and ArrayAdapter to bind data source to a List View.

e. How to create AlertDialog and progress Dialog in your activity.

**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"

tools:visibility="visible">

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Long Click on Me"

android:textSize="25sp"

android:visibility="visible"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.java**

package com.example.contextmenu;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.ContextMenu;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private Button button;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

button = findViewById(R.id.button);

registerForContextMenu(button);

}

@Override

public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {

getMenuInflater().inflate(R.menu.menu,menu);

super.onCreateContextMenu(menu, v, menuInfo);

}

@Override

public boolean onContextItemSelected(@NonNull MenuItem item) {

Toast.makeText(this, ""+item.getTitle(), Toast.LENGTH\_SHORT).show();

return super.onContextItemSelected(item);

}

}

menu.xml

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

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:title="@string/one"

/>

<item

android:title="@string/two"

/>

<item

android:title="@string/three"

/>

<item

android:title="@string/four"

/>

<item

android:title="@string/five"

/>

</menu>

Q3. You will be persisting data using an SQLite Database and preserving the state of

an application during its lifecycle.

a. How to save & restore data as Application Preferences (Shared Preference).

b. How to save & restore data as Instance State.

c. How to create and manage an SQLiteDatabase in Android.

d. How to insert, update, remove, and retrieve data from an SQLite Database.

e. Display data using RecyclerView.

**MainActivity.java**

package com.example.sqllitetry;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.database.Cursor;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

EditText name, contact, dob;

Button insert, update, delete, view;

DBHelper DB;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

name = findViewById(R.id.name);

contact = findViewById(R.id.contact);

dob = findViewById(R.id.dob);

insert = findViewById(R.id.btnInsert);

update = findViewById(R.id.btnUpdate);

delete = findViewById(R.id.btnDelete);

view = findViewById(R.id.btnView);

DB = new DBHelper(this);

insert.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String nameTXT = name.getText().toString();

String contactTXT = contact.getText().toString();

String dobTXT = dob.getText().toString();

Boolean checkinsertdata = DB.insertuserdata(nameTXT, contactTXT, dobTXT);

if(checkinsertdata==true)

Toast.makeText(MainActivity.this, "New Entry Inserted", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "New Entry Not Inserted", Toast.LENGTH\_SHORT).show();

} });

update.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String nameTXT = name.getText().toString();

String contactTXT = contact.getText().toString();

String dobTXT = dob.getText().toString();

Boolean checkupdatedata = DB.updateuserdata(nameTXT, contactTXT, dobTXT);

if(checkupdatedata==true)

Toast.makeText(MainActivity.this, "Entry Updated", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "New Entry Not Updated", Toast.LENGTH\_SHORT).show();

} });

delete.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

String nameTXT = name.getText().toString();

Boolean checkudeletedata = DB.deletedata(nameTXT);

if(checkudeletedata==true)

Toast.makeText(MainActivity.this, "Entry Deleted", Toast.LENGTH\_SHORT).show();

else

Toast.makeText(MainActivity.this, "Entry Not Deleted", Toast.LENGTH\_SHORT).show();

} });

view.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Cursor res = DB.getdata();

if(res.getCount()==0){

Toast.makeText(MainActivity.this, "No Entry Exists", Toast.LENGTH\_SHORT).show();

return;

}

StringBuffer buffer = new StringBuffer();

while(res.moveToNext()){

buffer.append("Name :"+res.getString(0)+"\n");

buffer.append("Contact :"+res.getString(1)+"\n");

buffer.append("Date of Birth :"+res.getString(2)+"\n\n");

}

AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

builder.setCancelable(true);

builder.setTitle("User Entries");

builder.setMessage(buffer.toString());

builder.show();

} });

}}

**DBHelper.java**

package com.example.sqllitetry;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DBHelper extends SQLiteOpenHelper {

public DBHelper(Context context) {

super(context, "Userdata.db", null, 1);

}

@Override

public void onCreate(SQLiteDatabase DB) {

DB.execSQL("create Table Userdetails(name TEXT primary key, contact TEXT, dob TEXT)");

}

@Override

public void onUpgrade(SQLiteDatabase DB, int i, int ii) {

DB.execSQL("drop Table if exists Userdetails");

}

public Boolean insertuserdata(String name, String contact, String dob)

{

SQLiteDatabase DB = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put("name", name);

contentValues.put("contact", contact);

contentValues.put("dob", dob);

long result=DB.insert("Userdetails", null, contentValues);

if(result==-1){

return false;

}else{

return true;

}

}

public Boolean updateuserdata(String name, String contact, String dob)

{

SQLiteDatabase DB = this.getWritableDatabase();

ContentValues contentValues = new ContentValues();

contentValues.put("contact", contact);

contentValues.put("dob", dob);

Cursor cursor = DB.rawQuery("Select \* from Userdetails where name = ?", new String[]{name});

if (cursor.getCount() > 0) {

long result = DB.update("Userdetails", contentValues, "name=?", new String[]{name});

if (result == -1) {

return false;

} else {

return true;

}

} else {

return false;

}

}

public Boolean deletedata (String name)

{

SQLiteDatabase DB = this.getWritableDatabase();

Cursor cursor = DB.rawQuery("Select \* from Userdetails where name = ?", new String[]{name});

if (cursor.getCount() > 0) {

long result = DB.delete("Userdetails", "name=?", new String[]{name});

if (result == -1) {

return false;

} else {

return true;

}

} else {

return false;

}

}

public Cursor getdata ()

{

SQLiteDatabase DB = this.getWritableDatabase();

Cursor cursor = DB.rawQuery("Select \* from Userdetails", null);

return cursor;

}

}

**Activity\_main.xml**

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

<RelativeLayout 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"

android:padding="10dp"

tools:context=".MainActivity">

<TextView

android:id="@+id/texttitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Please enter the details below"

android:textSize="24dp"

android:layout\_marginTop="20dp"/>

<EditText

android:id="@+id/name"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/texttitle"

android:hint="Name"

android:inputType="textPersonName"

android:textSize="24dp" />

<EditText

android:id="@+id/contact"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/name"

android:hint="Contact"

android:inputType="number"

android:textSize="24dp" />

<EditText

android:id="@+id/dob"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/contact"

android:hint="Date of Birth"

android:inputType="number"

android:textSize="24dp" />

<Button

android:id="@+id/btnInsert"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/dob"

android:layout\_marginTop="30dp"

android:text="Insert New Data"

android:textSize="24dp" />

<Button

android:id="@+id/btnUpdate"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/btnInsert"

android:text="Update Data"

android:textSize="24dp" />

<Button

android:id="@+id/btnDelete"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/btnUpdate"

android:text="Delete Existing Data"

android:textSize="24dp" />

<Button

android:id="@+id/btnView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/btnDelete"

android:text="View Data"

android:textSize="24dp" />

</RelativeLayout>

Q4.Develop an app to capture a photo and store it into SDCard, extend this app to display

all the photos capture in the grid view.

a. How to use the Camera.

b. How to write data to the SD card.

**activity\_main.xml**

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

<LinearLayout

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"

android:gravity="center\_horizontal"

android:orientation="vertical"

tools:context=".MainActivity">

<ImageView

android:id="@+id/imgCamera"

android:layout\_width="400dp"

android:layout\_height="240dp"

android:scaleType="fitXY" />

<Button

android:id="@+id/btnCamera"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="21dp"

android:text="Open Camera"/>

</LinearLayout>

**MainActivity.java**

package com.example.camerasd;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.graphics.Bitmap;

import android.os.Bundle;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

private final int CAMERA\_REQ\_CODE = 100;

ImageView imgCamera;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

imgCamera = findViewById(R.id.imgCamera);

Button btnCamera = findViewById(R.id.btnCamera);

btnCamera.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent iCamera = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

startActivityForResult(iCamera, CAMERA\_REQ\_CODE);

}

});

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == RESULT\_OK) {

if (requestCode == CAMERA\_REQ\_CODE) {

//for camera

Bitmap img = (Bitmap) (data.getExtras().get("data"));

imgCamera.setImageBitmap(img);

}

}

}

}

4b) Android select an image from the gallery or storage

**AndriodMainfest.xml**

<!--//Adding Read External storage Permission-->

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

**MainActivity.java**

package com.example.galleryapp;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import android.Manifest;

import android.app.Activity;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.net.Uri;

import android.os.Build;

import android.os.Bundle;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.Toast;

import java.io.InputStream;

public class MainActivity extends AppCompatActivity

{

private static final int REQUEST\_CODE\_STORAGE\_PERMISSION = 1;

private static final int REQUEST\_CODE\_SELECT\_IMAGE = 2;

private ImageView imageSelected;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

imageSelected = findViewById(R.id.selectedImage);

findViewById(R.id.buttonSelectedImage).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (ContextCompat.checkSelfPermission(

getApplicationContext(), Manifest.permission.READ\_EXTERNAL\_STORAGE

) != PackageManager.PERMISSION\_GRANTED) {

ActivityCompat.requestPermissions(

MainActivity.this,

new String[]{Manifest.permission.READ\_EXTERNAL\_STORAGE},

REQUEST\_CODE\_STORAGE\_PERMISSION);

} else {

selectImage();

}

}

});

}

private void selectImage()

{

Intent intent = new Intent(Intent.ACTION\_PICK,MediaStore.Images.Media.EXTERNAL\_CONTENT\_URI);

if(intent.resolveActivity(getPackageManager()) != null){

startActivityForResult(intent,REQUEST\_CODE\_SELECT\_IMAGE);

}

}

@Override

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

{

super.onRequestPermissionsResult(requestCode,permissions,grantResults);

if(requestCode == REQUEST\_CODE\_STORAGE\_PERMISSION && grantResults.length > 0)

{

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

{

selectImage();

}

else

{

Toast.makeText(this, "Permission Denied", Toast.LENGTH\_SHORT).show();

}

}

}

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(requestCode == REQUEST\_CODE\_SELECT\_IMAGE && resultCode == RESULT\_OK){

if(data != null){

Uri selectedImageUri = data.getData();

if(selectedImageUri != null){

try{

InputStream inputStream = getContentResolver().openInputStream(selectedImageUri);

Bitmap bitmap = BitmapFactory.decodeStream(inputStream);

imageSelected.setImageBitmap(bitmap);

}catch (Exception exception){

Toast.makeText(this,exception.getMessage(), Toast.LENGTH\_SHORT).show();

}

}

}

}

}

}

**activitymain.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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity">

<Button

android:id="@+id/buttonSelectedImage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/select\_image"/>

<ImageView

android:id="@+id/selectedImage"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

android:contentDescription="@string/app\_name" />

</LinearLayout>

Q5. Create an application to demonstrate few key features of the Android framework. In

particular, the application demonstrates how to send SMS text messages.

a. How to send SMS text messages.

b. How to dial using an in-built dialer

c. How to send email.

**activitymain.xml:**

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

<RelativeLayout 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="com.example.smsemailphone.MainActivity">

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Sending SMS Example"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true"

android:textSize="30dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Send Sms"

android:id="@+id/btnSendSMS"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="48dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Send Email"

android:id="@+id/btnSendEmail"

android:layout\_below="@+id/btnSendSMS"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="48dp" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Phone"

android:id="@+id/btnDialPhone"

android:layout\_marginTop="54dp" />

</RelativeLayout>

**MainActivity:**

package com.example.smsemailphone;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private static final int MY\_PERMISSIONS\_REQUEST\_SEND\_SMS = 0;

Button sendBtn;

Button btnSendEmail;

Button btnPhone;

String phoneNo;

String message;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

sendBtn = (Button) findViewById(R.id.btnSendSMS);

btnSendEmail = (Button) findViewById(R.id.btnSendEmail);

btnPhone = (Button) findViewById(R.id.btnDialPhone);

sendBtn.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

sendSMSMessage();

}

});

btnSendEmail.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

sendEmail();

}

});

btnPhone.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

PhoneDial();

}

});

}

protected void sendEmail() {

Intent emailIntent = new Intent(Intent.ACTION\_SEND);

emailIntent.setData(Uri.parse("mailto:"));

emailIntent.setType("text/plain");

emailIntent.putExtra(Intent.EXTRA\_EMAIL, new String[]{"ba.mohan@gmail.com"});

emailIntent.putExtra(Intent.EXTRA\_SUBJECT, "subject Test");

emailIntent.putExtra(Intent.EXTRA\_TEXT, "Message Body Test");

startActivity(emailIntent);

}

protected void sendSMSMessage() {

Intent sendIntent = new Intent(Intent.ACTION\_VIEW);

sendIntent.putExtra("sms\_body", "default content");

sendIntent.setType("vnd.android-dir/mms-sms");

startActivity(sendIntent);

Toast.makeText(getApplicationContext(), "SMS sent.",

Toast.LENGTH\_LONG).show();

}

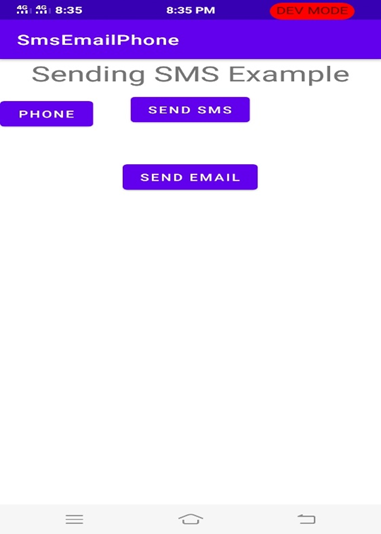
protected void PhoneDial() {

Intent intent = new Intent(Intent.ACTION\_DIAL);

startActivity(intent); }

}

Output:



Q6. Develop an app that include broadcast Receiver to receive the miss calls from the Known

number and display it to the user using notification services. This same app should also fetch

phone number from the inbuilt contacts using the concept of content provider.

a. How to use broadcast receiver and notifications.

b. How to use content providers.

6a:

**Activity\_main.xml**

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity">  
  
 <Button  
 android:id="@+id/buttonNotify"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/notify\_me"  
 tools:layout\_editor\_absoluteX="33dp"  
 tools:layout\_editor\_absoluteY="40dp"  
 tools:ignore="MissingConstraints" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

**Main\_activity.java**

package com.example.sixthapp;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.app.TaskStackBuilder;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Bundle;  
import android.widget.Button;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.NotificationCompat;  
  
@SuppressWarnings("ALL")  
public class MainActivity extends AppCompatActivity {  
 Button but;  
 Context context=MainActivity.this;  
 String CHANNEL\_ID="my\_channel\_01";  
 CharSequence name = "my\_channel";  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 but= findViewById(R.id.*buttonNotify*);  
 NotificationManager notificationManager = (NotificationManager) context.getSystemService(Context.*NOTIFICATION\_SERVICE*);  
  
 but.setOnClickListener(v -> {  
 NotificationChannel mChannel = new NotificationChannel(CHANNEL\_ID, name, NotificationManager.*IMPORTANCE\_HIGH*);  
 notificationManager.createNotificationChannel(mChannel);  
  
 Intent notificationIntent = new Intent(context, MainActivity.class);  
  
 TaskStackBuilder stackBuilder = TaskStackBuilder.*create*(context);  
 stackBuilder.addParentStack(MainActivity.class);  
 stackBuilder.addNextIntent(notificationIntent);  
  
 PendingIntent pendingIntent = stackBuilder.getPendingIntent(0, PendingIntent.*FLAG\_IMMUTABLE*);  
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(context,CHANNEL\_ID);  
  
 Notification notification = builder.setContentTitle("Notification Title")  
 .setContentText("Notification Text")  
 .setTicker("Test!")  
 .setSmallIcon(R.mipmap.*ic\_launcher*)  
 .setAutoCancel(true)  
 .setContentIntent(pendingIntent).build();  
  
 notificationManager.notify(0, notification);  
 });  
 }  
}

Program 6b:

**Activity\_main.xml**

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/button"

tools:layout\_editor\_absoluteX="33dp"

tools:layout\_editor\_absoluteY="40dp"

tools:ignore="MissingConstraints" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Main\_activity.java**

package com.example.sixthbapp;

import android.Manifest;

import android.annotation.SuppressLint;

import android.app.Activity;

import android.content.Intent;

import android.database.Cursor;

import android.net.Uri;

import android.os.Bundle;

import android.provider.ContactsContract;

import android.widget.Button;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

@SuppressWarnings("ALL")

public class MainActivity extends AppCompatActivity {

public static final int RequestPermissionCode = 1 ;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Button button = findViewById(R.id.button);

EnableRuntimePermission();

button.setOnClickListener(v -> {

Intent intent = new Intent(Intent.ACTION\_PICK, ContactsContract.Contacts.CONTENT\_URI);

startActivityForResult(intent, 1);

});

}

public void EnableRuntimePermission(){

if (ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this, Manifest.permission.READ\_CONTACTS))

{

Toast.makeText(MainActivity.this,"CONTACTS permission allows us to Access CONTACTS app", Toast.LENGTH\_LONG).show();

} else {

ActivityCompat.requestPermissions(MainActivity.this,new String[]{ Manifest.permission.READ\_CONTACTS}, RequestPermissionCode);

}

}

@Override

public void onActivityResult(int reqCode, int resultCode, Intent data){

super.onActivityResult(reqCode, resultCode, data);

if (reqCode == 1) {

if (resultCode == Activity.RESULT\_OK) {

Uri contactData = data.getData();

try (Cursor c = managedQuery(contactData, null, null, null, null)) {

if (c.moveToFirst()) {

String id = c.getString(c.getColumnIndexOrThrow(ContactsContract.Contacts.\_ID));

@SuppressLint("Range")

String hasPhone = c.getString(c.getColumnIndex(ContactsContract.Contacts.HAS\_PHONE\_NUMBER));

if (hasPhone.equalsIgnoreCase("1")) {

Cursor phones = getContentResolver().query(ContactsContract.CommonDataKinds.Phone.CONTENT\_URI, null, ContactsContract.CommonDataKinds.Phone.CONTACT\_ID + " = " + id, null, null);

phones.moveToFirst();

@SuppressLint("Range")

String cNumber = phones.getString(phones.getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));

Toast.makeText(getApplicationContext(), cNumber, Toast.LENGTH\_SHORT).show();

phones.close();

}

}

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

}

}

**AndriodManifest.xml**

   <uses-permission android:name="android.permission.READ\_CONTACTS"/>

7.Design an android app to fetch the JSON data from the internet and display the data using

listView.

c. Employee data is stored in the internet. (use Async Task)

d. When app sends the request to the server, the server should provide data in json

format.

e. The client app should fetch this data and display using listview.

https://api.androidhive.info/contacts/

{

"contacts": [

{

"id": "c200",

"name": "Ravi Tamada",

"email": "ravi@gmail.com",

"address": "xx-xx-xxxx,x - street, x - country",

"gender" : "male",

"phone": {

"mobile": "+91 0000000000",

"home": "00 000000",

"office": "00 000000"

}

},

**Main\_Activity.java**

public class MainActivity extends AppCompatActivity {

Button b;

ListView lv;

ArrayList<HashMap<String, String>>contactList;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

contactList = new ArrayList<>();

lv= (ListView) findViewById(R.id.list);

b= (Button) findViewById(R.id.fetch);

b.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

String strUrl = "https://api.androidhive.info/contacts/";

new UrlHandler().execute(strUrl);

}

});

}

public class UrlHandler extends AsyncTask<String, Integer, String> {

@Override

protected void onPostExecute(String s) {

super.onPostExecute(s);

ListAdapter adapter = new SimpleAdapter(MainActivity.this, contactList,

R.layout.list\_item, new String[]{ "id","name","email"},

new int[]{R.id.cid,R.id.cname, R.id.cemail});

lv.setAdapter(adapter);

}

@Override

protected String doInBackground(String... params) {

String json\_response = null;

try {

URL url = new URL(params[0]);

HttpURLConnection connection = (HttpURLConnection) url.openConnection();

connection.setRequestMethod("GET");

connection.connect();

InputStream in = new BufferedInputStream(connection.getInputStream());

json\_response = convertStreamToString(in);

if (json\_response != null) {

try {

JSONObject jsonObj = new JSONObject(json\_response);

// Getting JSON Array node

JSONArray contacts = jsonObj.getJSONArray("contacts");

// looping through All Contacts

for (int i = 0; i < contacts.length(); i++) {

JSONObject c = contacts.getJSONObject(i);

String id = c.getString("id");

String name = c.getString("name");

String email = c.getString("email");

// tmp hash map for single contact

HashMap<String, String> contact = new HashMap<>();

// adding each child node to HashMap key => value

contact.put("id", id);

contact.put("name", name);

contact.put("email", email);

// adding contact to contact list

contactList.add(contact);

}

} catch (JSONException e) {

Log.e("error", "Json parsing error: " + e.getMessage());

}

} else {

Log.e("error", "Couldn't get json from server.");

}

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

return null;

}

private String convertStreamToString(InputStream is) {

BufferedReader reader = new BufferedReader(new InputStreamReader(is));

StringBuilder sb = new StringBuilder();

String line;

try {

while ((line = reader.readLine()) != null) {

sb.append(line).append('\n');

}

} catch (IOException e) {

e.printStackTrace();

}

return sb.toString();

}

}

}

**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"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context="nmit.mohan.com.myapplication.MainActivity">

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="Fetch Data"

android:id="@+id/fetch"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/textView"

/>

<ListView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/list"

></ListView>

</LinearLayout>

**list\_item.xml**

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

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/cid"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/cname"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/cemail"

/>

</LinearLayout>

**Android\_Mainfest.xml**

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

Q8.Develop an android app on Google Map, and should provide following functions.

a. How to incorporate Google Maps into an application.

b. How to register for and receive GPS location information.

c. How to create Google Maps Overlays.

d. Accept city name from user and marks it on map.

e. Explore features like Zoom and map types.

**MapsActivity.java**

package com.example.program8;  
  
import androidx.fragment.app.FragmentActivity;  
  
import android.location.Address;  
import android.location.Geocoder;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
  
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.program8.databinding.ActivityMapsBinding;  
  
import java.io.IOException;  
import java.util.List;  
  
public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {  
  
 private GoogleMap mMap;  
 private TextView tv;  
 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);  
 }  
  
 /\*\*  
 \* 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  
 public void onMapReady(GoogleMap googleMap) {  
 mMap = googleMap;  
  
 // Add a marker in Bengaluru and move the camera  
 LatLng Bengaluru = new LatLng(13, 78);  
 mMap.addMarker(new MarkerOptions().position(Bengaluru).title("Marker in Bengaluru"));  
 mMap.moveCamera(CameraUpdateFactory.newLatLng(Bengaluru));  
 }  
  
 public void setmMap(GoogleMap mMap) {  
 this.mMap = mMap;  
 }  
  
 public void onSearch(View view) {  
 List<Address> addressList = null;  
 EditText et\_location = (EditText) findViewById(R.id.et1);  
 String location = et\_location.getText().toString();  
 if (location != null || location.equals("")) {  
 Geocoder geocoder = new Geocoder(this);  
  
 try {  
  
  
 addressList = geocoder.getFromLocationName(location, 1);  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 Address address = addressList.get(0);  
 LatLng latLng = new LatLng(address.getLatitude(), address.getLongitude());  
 mMap.addMarker(new MarkerOptions().position(latLng).title(location));  
 mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));  
 }  
 }  
  
 public void onType(View view) {  
 if (mMap.getMapType() == GoogleMap.MAP\_TYPE\_NORMAL) {  
 mMap.setMapType(GoogleMap.MAP\_TYPE\_SATELLITE);  
 } else {  
 mMap.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);  
 }  
 }  
  
 public void onZoom(View view) {  
 if (view.getId() == R.id.zoomin) {  
 mMap.animateCamera(CameraUpdateFactory.zoomIn());  
 }  
 if (view.getId() == R.id.zoomout) {  
 mMap.animateCamera(CameraUpdateFactory.zoomOut());  
 }  
  
 }  
}

**Activity\_maps.xml**

<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:layout\_width="match\_parent"  
 android:orientation="vertical" android:layout\_height="400dp">  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:id="@+id/ll1">  
  
 <EditText  
 android:id="@+id/et1"  
 android:layout\_width="196dp"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/searchbut"  
 android:layout\_width="98dp"  
 android:layout\_height="wrap\_content"  
 android:onClick="onSearch"  
 android:text="Search" />  
  
 <Button  
 android:id="@+id/typebut"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:onClick="onType"  
 android:text="Type" />  
  
 </LinearLayout>  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:layout\_below="@id/ll1"  
 android:id="@+id/linearLayout"  
 android:layout\_alignParentBottom="true">  
  
 <fragment xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/map"  
 android:layout\_below="@id/ll1"  
 android:name="com.google.android.gms.maps.SupportMapFragment"  
 android:layout\_width="343dp"  
 android:layout\_height="match\_parent"  
 tools:context="com.example.mohan.demomaps.MapsActivity" />  
  
 <LinearLayout  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <Button  
 android:id="@+id/zoomin"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="69dp"  
 android:onClick="onZoom"  
 android:text="+" />  
  
 <Button  
 android:id="@+id/zoomout"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="68dp"  
 android:onClick="onZoom"  
 android:text="-" />  
 </LinearLayout>  
 </LinearLayout>  
</RelativeLayout>

**AndroidMainfest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.program8">  
  
 <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION" />  
 <uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION" />  
 <uses-permission android:name="android.permission.INTERNET"/>  
 <uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>  
 <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />  
 <uses-permission android:name="com.google.android.providers.gsf.permission.READ\_GSERVICES" />  
  
  
 <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.Program8">  
  
  
  
 <!--  
 TODO: Before you run your application, you need a Google Maps API key.  
  
 To get one, follow the directions here:  
  
 https://developers.google.com/maps/documentation/android-sdk/get-api-key  
  
 Once you have your API key (it starts with "AIza"), define a new property in your  
 project's local.properties file (e.g. MAPS\_API\_KEY=Aiza...), and replace the  
 "YOUR\_API\_KEY" string in this file with "${MAPS\_API\_KEY}".  
 -->  
 <meta-data  
 android:name="com.google.android.geo.API\_KEY"  
 android:value="AIzaSyAPTYkumdx\_qL8KCypbGI5asIsNRXmt2p8" />  
  
 <activity  
 android:name=".MapsActivity"  
 android:exported="true"  
 android:label="@string/title\_activity\_maps">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
  
</manifest>