INDEX

DATE:-

SESSION-1 Review of Java Concepts, Download and Install Android Studio, Android Setup,Application components, Resources, Activities, Services.

Develop a Hello World Program

Introduction to Java:

JAVA was developed by James Gosling at Sun Microsystems in the year 1991, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs.

Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to C/C++.

History

Java's history is very interesting. It is a programming language created in 1991. James Gosling, Mike Sheridan, and Patrick Naughton, a team of Sun engineers known as the Green team initiated the Java language in 1991. Sun Microsystems released its first public implementation in 1996 as Java 1.0.

It provides no-cost -run-times on popular platforms. Java1.0 compiler was re-written in Java by Arthur Van Hoff to strictly comply with its specifications. With the arrival of Java 2, new versions had multiple configurations built for different types of platforms.

In 1997, Sun Microsystems approached the ISO standards body and later formalized Java, but it soon withdrew from the process. At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine as free, open- source software. On May 8, 2007, Sun finished the process, making all of its JVM's core code available under open-source distribution terms.

The principles for creating java were simple, robust, secured, high performance, portable, multi-threaded, interpreted, dynamic, etc. James Gosling in 1995 developed Java, who is known as the Father of Java. Currently, Java is used in mobile devices, internet programming, games, e-business, etc.

Java programming language is named JAVA. Why?

After the name OAK, the team decided to give a new name to it and the suggested words were Silk, Jolt, revolutionary, DNA, dynamic, etc. These all names were easy to spell and fun to say, but they all wanted the name to reflect the essence of technology. In accordance with James Gosling, Java the among the top names along with Silk, and since java was a unique name so most of them preferred it.

Java is the name of an island in Indonesia where the first coffee (named java coffee) was produced. And this name was chosen by James Gosling while having coffee near his office. Note that Java is just a name, not an acronym.

Java Terminology

Before learning Java, one must be familiar with these common terms of Java.

1**. Java Virtual Machine(JVM):** This is generally referred to as JVM. There are three execution phases of a program. They are written, compile and run the program.

- Writing a program is done by a java programmer like you and me.
- The compilation is done by the JAVAC compiler which is a primary Java compiler included in the Java development kit (JDK). It takes Java program as input and generates bytecode as output.
- In the Running phase of a program, JVM executes the bytecode generated by the compiler.
- Now, we understood that the function of Java Virtual Machine is to execute the bytecode produced by the compiler. Every Operating System has a different JVM but the output they produce after the execution of bytecode is the same across all the operating systems. This is why Java is known as a platform-independent language.

2. **Bytecode in the Development process**: As discussed, the Javac compiler of JDK compiles the java source code into bytecode so that it can be executed by JVM. It is saved as .class file by the compiler. To view the bytecode, a disassembler like javap can be used.

3. **Java Development Kit (JDK):** While we were using the term JDK, when we learn about bytecode and JVM . So, as the name suggests, it is a complete Java development kit that includes everything including compiler, Java Runtime Environment (JRE), java debuggers, java docs, etc. For

the program to execute in java, we need to install JDK on our computer in order to create, compile and run the java program.

4. **Java Runtime Environment (JRE)**: JDK includes JRE. JRE installation on our computers allows the java program to run, however, we cannot compile it. JRE includes a browser, JVM, applet supports, and plugins. For running the java program, a computer needs JRE.

5. **Garbage Collector**: In Java, programmers can't delete the objects. To delete or recollect that memory JVM has a program called Garbage Collector. Garbage Collectors can recollect the of objects that are not referenced. So, Java makes the life of a programmer easy by handling memory management. However, programmers should be careful about their code whether they are using objects that have been used for a long time. Because Garbage cannot recover the memory of objects being referenced.

6. **Class Path**: The class path is the file path where the java runtime and Java compiler look for .class files to load. By default, JDK provides many libraries. If you want to include external libraries, they should be added to the class path.

Primary/Main Features of Java

1. **Platform Independent**: Compiler converts source code to bytecode and then the JVM executes the bytecode generated by the compiler. This bytecode can run on any platform be it Windows, Linux, macOS which means if we compile a program on Windows, then we can run it on Linux and vice versa. Each operating system has a different JVM, but the output produced by all the OS is the same after the execution of bytecode. That is why we call java a platform-independent language.

2. **Object-Oriented Programming Language**: Organizing the program in the terms of collection of objects is a way of object-oriented programming, each of which represents an instance of the class. The four main concepts of Object-Oriented programming are:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

3. **Simple**: Java is one of the simple languages as it does not have complex features like pointers, operator overloading, multiple inheritances, Explicit memory allocation.

4. **Robust:** Java language is robust that means reliable. It is developed in such a way that it puts a lot of effort into checking errors as early as possible, that is why the java compiler is able to detect even those errors that are not easy to detect by another programming language. The main features of java that make it robust are garbage collection, Exception Handling, and memory allocation.

5. **Secure**: In java, we don't have pointers, and so we cannot access out-of-bound arrays i.e it shows ArrayIndexOutOfBound Exception if we try to do so. That's why several security flaws like stack corruption or buffer overflow are impossible to exploit in Java.

6. **Distributed**: We can create distributed applications using the java programming language. Remote Method Invocation and Enterprise Java Beans are used for creating distributed applications in java. The java programs can be easily distributed on one or more systems that are connected to each other through an internet connection.

7. **Multithreading**: Java supports multithreading. It is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU.

8. **Portable:** As we know, java code written on one machine can be run on another machine. The platform-independent feature of java in which its platform-independent bytecode can be taken to any platform for execution makes java portable.

9. **High Performance**: Java architecture is defined in such a way that it reduces overhead during the runtime and at some time java uses Just In Time (JIT) compiler where the compiler compiles code on-demand basics where it only compiles those methods that are called making applications to execute faster.

10. **Dynamic flexibility**: Java being completely object-oriented gives us the flexibility to add classes, new methods to existing classes and even creating new classes through sub-classes. Java even supports functions written in other languages such as C, C++ which are referred to as native methods.

11. **Sandbox Execution**: Java programs run in a separate space that allows user to execute their applications without affecting the underlying system with help of a bytecode verifier. Bytecode verifier also provides additional security as its role is to check the code for any violation access.

12. **Write Once Run Anywhere**: As discussed above java application generates '.class' file which corresponds to our applications(program) but contains code in binary format. It provides ease t

architecture-neutral ease as bytecode is not dependent on any machine architecture. It is the primary reason java is used in the enterprising IT industry globally worldwide.

13. **Power of compilation and interpretation:** Most languages are designed with purpose either they are compiled language, or they are interpreted language. But java integrates arising enormous power as Java compiler compiles the source code to bytecode and JVM executes this bytecode to machine OS- dependent executable code.

Minimum Theoretical Background

Step 1 - Setup Java Development Kit (JDK) You can download the latest version of Java JDK from Oracle's Java site: Java SE Downloads. You will find instructions for installing JDK in downloaded files, follow the given instructions to install and configure the setup. Finally, set PATH and JAVA_HOME environment variables to refer to the directory that contains java and javac, typically java_install_dir/bin and java_install_dir respectively. If you are running Windows and have installed the JDK in C:\jdk1.6.0_15, you would have to put the following line in your C:\autoexec.batfile.

set PATH=C:\jdk1.6.0_15\bin;%PATH% set JAVA_HOME=C:\jdk1.6.0_15

Step 2 - Setup Android SDK You can download the latest version of Android SDK from Android's official website: http://developer.android.com/sdk/index.html. If you are installing SDK on Windows machine, then you will find ainstaller_rXXwindows. exe, so just download and run this exe which will launch Android SDK Tool Setup wizard to guide you throughout the installation, so just follow the instructions carefully. Finally, you will have Android SDK Tools installed on your machine. If you are installing SDK either on Mac OS or Linux, check the instructions provided along with the downloaded android-sdk_rXX-macosx.zip file for Mac OS and androidsdk_rXX-linux.tgz file for Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 7 operating system.

Step 3 - Setup Android Development Tools (ADT) Plugin This step will help you in setting Android Development Tool plugin for Eclipse. Let's start with launching Eclipse and then, choose Help > Software Updates > Install New Software. This will display the following dialogue box.

Step 4 - Create Android Virtual Device to test your Android applications you will need a virtual Android device. So, before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager using Eclipse menu options Window > AVD Manager> which will launch

Android AVD Manager. Use New button to create a new Android Virtual Device and enter the following information, before clicking Create AVD button.
Android Development Tools (ADT) is a plugin for the Android studio that is designed to give you a powerful, integrated environment in which to build Android applications. ADT extends the capabilities of Android studio to let you quickly set up new Android projects, create an application UI, add components based on the Android Framework API, debug your applications using the Android SDK tools, and even export signed (or unsigned) APKs in order to distribute your application. Developing in Android studio with ADT is highly recommended and is the fastest way to get started.

With the guided project setup, it provides, as well as tools integration, custom XML editors, and debug output pane, applications. ADT gives you an incredible boost in developing Android applications.

In android studio students must be aware of the directory structure and the control flow of the program. Program should be either executed on the android mobile phones or on the suitable emulators. To execute a simple program, like to display Hello World on screen syntax of writing a program in android is pre-requisite as the programming language used is JAVA only. The main activity code is a Java file MainActivity.java. This is the actual application file which ultimately gets converted to a Dalvik executable and runs your application.

Following is the default code generated by the application wizard for Hello World! application:

AIM: To develop a Hello World Program

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.helloworld
2. Modify src/MainActivity.java file to add helloworld code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity:

```
package com.example.helloworld;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        TextView helloTextView = findViewById(R.id.helloTextView);
        helloTextView.setText("Hello World!");
    }
}
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <TextView
        android:id="@+id/helloTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="74sp"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

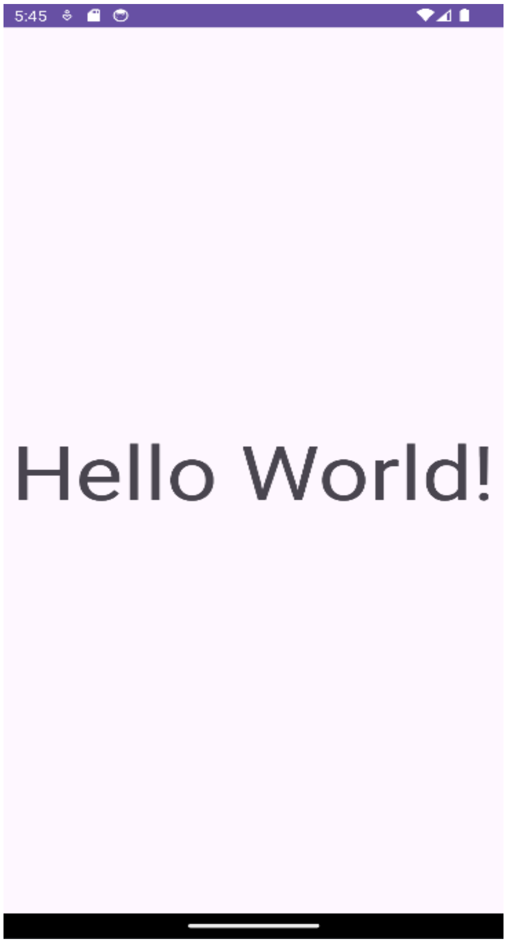
ActivityManifest.xml:

```
<?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:supportsRtl="true"
        android:theme="@style/Theme.Hello_World"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

The Manifest File: Whatever component you develop as a part of your application, you must declare all its components in a manifest file called AndroidManifest.xml which resides at the root of the application project directory. This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS. The <activity> tag is used to specify an activity and android:name attribute specifies the fully qualified class name of the Activity subclass and the android:label attributes specifies a string to use as the label for the activity. You can specify multiple activities using <activity> tags. The action for the intent filter is named android.intent.action.MAIN to indicate that this activity serves as the entry point for the application. The category for the intent- filter is named android.intent.category.LAUNCHER to indicate that the application can be launched from the device's launcher icon. The @string refers to the strings.xml file explained below. Hence, @string/app_name refers to the app_name string defined in the strings.xml file, which is "HelloWorld". Similar way, other strings get populated in the application. Following is the list of tags which you will use in your manifest file to specify different Android application components:

1) <activity>elements for activities
2) <service> elements for services
3) <receiver> elements for broadcast receivers
4) <provider> elements for content providers

OUTPUT:



Hello World!

RESULT:

Thus, the android application Hello World program is developed and tested using android studio.

DATE:

Session-II: Android User Interfaces: UI Layouts, UI Controls, Styles and Themes. Develop an application that uses GUI components, Font, and Colors

AIM: To develop an android application that uses GUI components, Font and Colours using android studio.

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.gui
2. Modify src/MainActivity.java file to add gui code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java

```
package com.example.gui;
import android.app.Activity;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity{
    float font=24;
    int i=1;
int fontType = Typeface.NORMAL; // Default font style
@Override
public void onCreate(Bundle savedInstance) {
    super.onCreate(savedInstance);
    setContentView(R.layout.activity_main);
    final TextView t1=(TextView)findViewById(R.id.textView1);
    Button b1=(Button)findViewById(R.id.button1);
    Button b2=(Button)findViewById(R.id.button2);
    Button b3=(Button)findViewById(R.id.button3);
    b1.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            t1.setTextSize(font);
            font=font+4;
            if(font==40)
                font=20;
        }
    });
    b2.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            switch(i){
                case 1:
                    t1.setTextColor(Color.parseColor("#0000FF"));
                    break;
                case 2:
                    t1.setTextColor(Color.parseColor("#00FF00"));
                    break;
                case 3:
                    t1.setTextColor(Color.parseColor("#FF0000"));
                    break;
                case 4:
                        t1.setTextColor(Color.parseColor("#F00000"));
                        break;
                }
                i++;
                if(i==5)
                    i=1;
            }
        });
        b3.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                switch(fontType){
                    case Typeface.NORMAL:
                        t1.setTypeface(null, Typeface.BOLD);
                        fontType = Typeface.BOLD;
                        break;
                    case Typeface.BOLD:
                        t1.setTypeface(null, Typeface.ITALIC);
                        fontType = Typeface.ITALIC;
                        break;
                    case Typeface.ITALIC:
                        t1.setTypeface(null, Typeface.NORMAL);
                        fontType = Typeface.BOLD_ITALIC;
                        break;
                }
            }
        });
    }
}
```

activity_main.xml

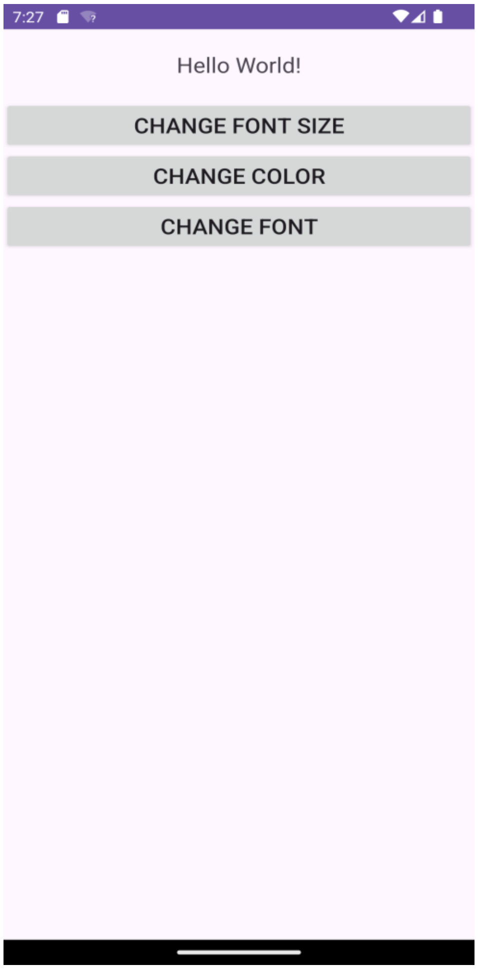
```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20sp"
        android:gravity="center"
        android:text="Hello World!"
        android:textSize="20sp"
        android:textStyle="bold"/>
    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="change font size"
        android:textSize="20sp"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="change color"
        android:textSize="20sp"/>
    <Button
        android:id="@+id/button3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="change font"
        android:textSize="20sp"/>
</LinearLayout>
```

AndroidManifest:

```
<?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:supportsRtl="true"
        android:theme="@style/Theme.Gui"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

OUTPUT:



Hello World!

RESULT:

Thus, the android application GUI components, Font size, Font style and Font Colors is developed and tested using android studio.

DATE:-

Session-III: Android Event Handling, Drag and Drop, Notifications.
Develop a basic Calculator application that uses Layout Managers and event listeners.

AIM: To develop an android application that uses Layout Managers and event listeners using android studio.

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.calculator
2. Modify src/MainActivity.java file to add calculator code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.xml

```
package com.example.calculator;
import android.app.Activity;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends Activity implements View.OnClickListener {
    EditText input1;
    EditText input2;
    Button addition;
    Button subtraction;
    Button multiplication;
    Button division;
    TextView tvResult;
    String opera = " ";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        input1 = (EditText) findViewById(R.id.etNum1);
        input2 = (EditText) findViewById(R.id.etNum2);
        addition = (Button) findViewById(R.id.btnAdd);
        subtraction =(Button) findViewById(R.id.btnSub);
        multiplication = (Button)findViewById(R.id.btnMult);
        division = (Button)findViewById(R.id.btnDiv);
        tvResult = (TextView) findViewById(R.id.tvResult);
        addition.setOnClickListener(this);
        subtraction.setOnClickListener(this);
        multiplication.setOnClickListener(this);
        division.setOnClickListener(this);
    }
    @Override
    public void onClick(View v) {
        float num1 = 0;
        float num2 = 0;
        float result = 0;
        if (TextUtils.isEmpty(input1.getText().toString()) ||
TextUtils.isEmpty(input2.getText().toString()))
```

```
            return;
        num1 = Float.parseFloat(input1.getText().toString());
        num2 = Float.parseFloat(input2.getText().toString());
        int id = v.getId();
        if(id==R.id.btnAdd){
            opera = "+";
            result = num1 + num2;
        }
        else if(id==R.id.btnSub){
            opera = "-";
            result = num1 - num2;
        }
        else if(id==R.id.btnMult){
            opera = "*";
            result = num1 * num2;
        }
        else if(id==R.id.btnDiv){
            opera = "/";
            result = num1 / num2;
        }
        else
        {
            Toast.makeText(getApplicationContext(),"invalid input",Toast.LENGTH_LONG).show();
        }
        tvResult.setText(num1 + " " + opera + " " + num2 + " = " + result);
    }
}
```

activity_main.xml:

```
<?xmlVersion ="1.0" encoding="utf-8?>
```

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/linearLayout1"
        android:layout_marginLeft="10pt"
        android:layout_marginRight="10pt"
        android:layout_marginTop="3pt">
        <EditText
            android:layout_weight="1"
            android:layout_height="wrap_content"
            android:layout_marginRight="5pt"
            android:id="@+id/etNum1"
            android:layout_width="match_parent"
            android:inputType="numberDecimal">
        </EditText>
        <EditText
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:layout_marginStart="5pt"
            android:id="@+id/etNum2"
            android:layout_width="match_parent"
            android:inputType="numberDecimal">
        </EditText>
    </LinearLayout>
    <LinearLayout
```

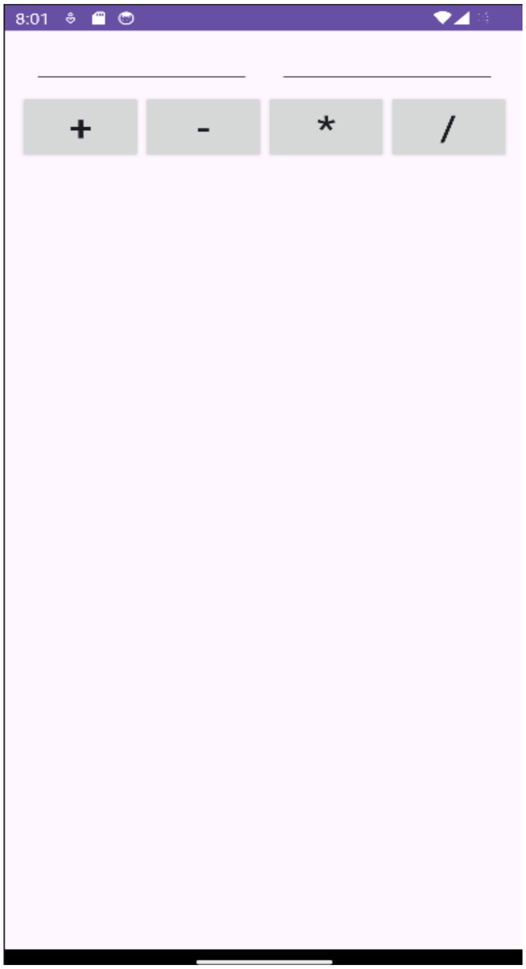
```
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/linearLayout2"
android:layout_marginTop="3pt"
android:layout_marginLeft="5pt"
android:layout_marginRight="5pt">
<Button
   android:layout_height="wrap_content"
   android:layout_width="match_parent"
   android:layout_weight="1"
   android:text="+"
   android:textSize="15pt"
   android:id="@+id/btnAdd">
</Button>
<Button
   android:layout_height="wrap_content"
   android:layout_width="match_parent"
   android:layout_weight="1"
   android:text="-"
   android:textSize="15pt"
   android:id="@+id/btnSub">
</Button>
<Button
   android:layout_height="wrap_content"
   android:layout_width="match_parent"
   android:layout_weight="1"
   android:text="*"
   android:textSize="15pt"
   android:id="@+id/btnMult">
</Button>
```

```
        <Button
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:layout_weight="1"
            android:text="/"
            android:textSize="15pt"
            android:id="@+id/btnDiv">
        </Button>
    </LinearLayout>
    <TextView
        android:id="@+id/tvResult"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="5pt"
        android:layout_marginTop="3pt"
        android:layout_marginRight="5pt"
        android:gravity="center_horizontal"
        android:textSize="12pt"></TextView>
</LinearLayout>
```

AndroidManifest:

```
<?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:supportsRtl="true"
        android:theme="@style/Theme.Calculator"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



RESULT:

Thus, the android application Calculator is developed and tested using android studio.

DATE:-

SESSION-IV: Alert Dialogues, Clipboard, Animation

Develop an application that draws basic graphical primitives on the screen.

AIM:

To write an application that draws basic graphical primitives on the screen using android studio.

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.graphical
2. Modify src/MainActivity.java file to add graphical code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.xml

```
package com.example.graphical;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;
import  ndroid.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity
{
   @Override
```

```
protected void onCreate(Bundle savedInstanceState)
{
   super.onCreate(savedInstanceState);
   setContentView(new myView(this));
}
private class myView extends View
{
   public myView(Context Context)
   {
      super(Context);
   }
   @Override
   protected void onDraw(Canvas Canvas)
   {
      super.onDraw(Canvas);
      Paint paint = new Paint();
      paint.setTextSize(40);
      Canvas.drawText(“Circle”,52,30,paint);
      paint.setColor(Color.GREEN);
      Canvas.drawCircle(100,150,100,paint);
      paint.setColor(Color.RED);
      Canvas.drawText(“Rectangle”,255,30,paint);
      paint.setColor(Color.YELLOW);
      Canvas.drawRect(250,50,400,350,paint);
      paint.setColor(Color.GREEN);
      Canvas.drawText(“SQUARE”,55,430,paint);
      paint.setColor(Color.BLUE);
      Canvas.drawRect(50,450,150,550,paint);
      paint.setColor(Color.GREEN);
      Canvas.drawText(“LINE”,255,430,paint);
```

```
            paint.setColor(Color.CYAN);
            Canvas.drawLine(250,500,350,500,paint);
        }
    }
}
```

activity_main.xml:

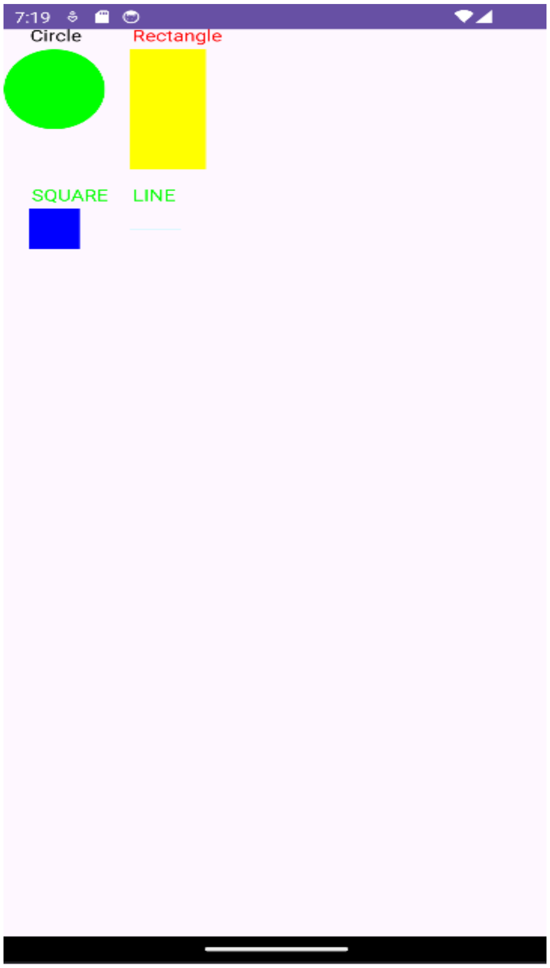
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

AndroidManifest:

```
<?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:supportsRtl="true"
        android:theme="@style/Theme.Graphical"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



RESULT:

Thus, the application that draws basic graphical primitives on the screen is developed and tested using android studio.

DATE:-

SESSION-V: Sending SMS, Phone calls.

Develop an application for SMS and Phone Calls.

SMS

AIM: To develop an application for SMS

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.sms
2. Modify src/MainActivity.java file to add SMS code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java

```
package com.example.sms;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    EditText phonenumber, message;
    Button send;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        send = findViewById(R.id.button);
        phonenumber = findViewById(R.id.editText);
        message = findViewById(R.id.editText2);
        send.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                String number = phonenumber.getText().toString();
                String msg = message.getText().toString();
                if (isValidIndianNumber(number)) {
                    try {
                        SmsManager smsManager = SmsManager.getDefault();
                        smsManager.sendTextMessage(number, null, msg, null, null);
                        Toast.makeText(getApplicationContext(), "Message Sent",
Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(), "Failed to send message",
Toast.LENGTH_LONG).show();
                    }
                } else {
                    Toast.makeText(getApplicationContext(), "Please enter a valid Indian number",
Toast.LENGTH_LONG).show();
                }
            }
        });
    }
    private boolean isValidIndianNumber(String number) {
        return number.startsWith("+91");
    }
}
```

AndroidManifest.xml

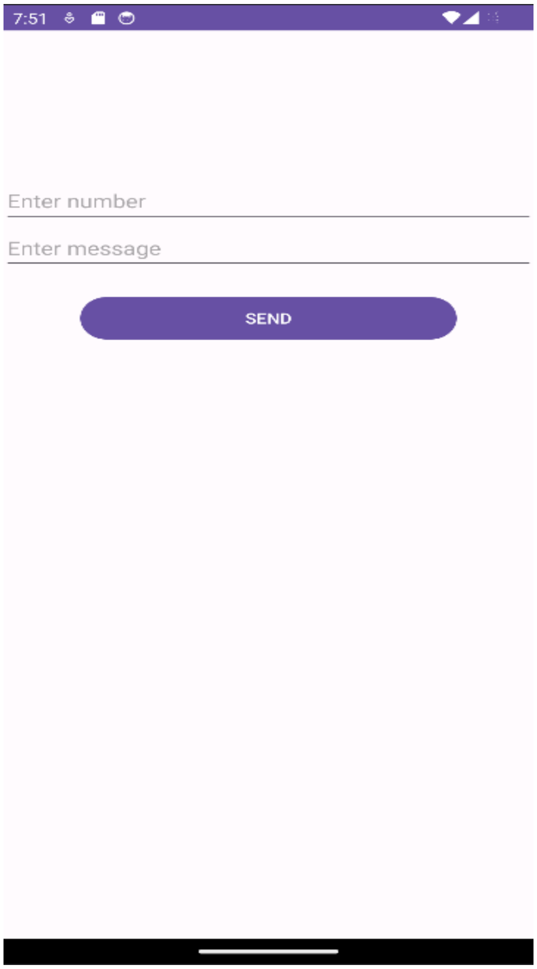
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:dist="http://schemas.android.com/apk/distribution"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.sms">
    <uses-feature
        android:name="android.hardware.telephony"
        android:required="false" />
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <dist:module dist:instant="true" />
    <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.Sms">
        <activity
            android:name="com.example.sms.MainActivity"
            android:exported="true"
            tools:ignore="MissingClass">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

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:orientation="vertical"
    android:layout_marginTop="140dp"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter number"
        android:inputType="textPersonName" />

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Enter message"
        android:inputType="textPersonName" />
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="60dp"
        android:layout_marginRight="60dp"
        android:text="SEND" />
</LinearLayout>
```

OUTPUT:



RESULT:

The application that SMS on the screen is developed and tested using android studio.

PHONE CALLS:

AIM: To develop an application for Phone Calls

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.phone_calls
2. Modify src/MainActivity.java file to add Phone calls code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM

MainActivity.java:

```
package com.example.phone_calls;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_CALL_PHONE = 1;
    private EditText editTextPhoneNumber;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTextPhoneNumber = findViewById(R.id.editTextPhoneNumber);
    }
    public void makePhoneCall(View view) {
        String phoneNumber = editTextPhoneNumber.getText().toString().trim();
        if (!phoneNumber.isEmpty()) {
            if (ContextCompat.checkSelfPermission(MainActivity.this,
Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(MainActivity.this, new
String[]{Manifest.permission.CALL_PHONE}, REQUEST_CALL_PHONE);
            } else {
                startPhoneCall(phoneNumber);
            }
        } else {
            Toast.makeText(this, "Please enter a phone number", Toast.LENGTH_SHORT).show();
        }
    }
    private void startPhoneCall(String phoneNumber) {
        Intent intent = new Intent(Intent.ACTION_CALL);
        intent.setData(Uri.parse("tel:" + phoneNumber));
        startActivity(intent);
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == REQUEST_CALL_PHONE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
                String phoneNumber = editTextPhoneNumber.getText().toString().trim();
                if (!phoneNumber.isEmpty()) {
                    startPhoneCall(phoneNumber);
                } else {
                    Toast.makeText(this, "Please enter a phone number", Toast.LENGTH_SHORT).show();
                }
            } else {
                Toast.makeText(this, "Permission Denied", Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

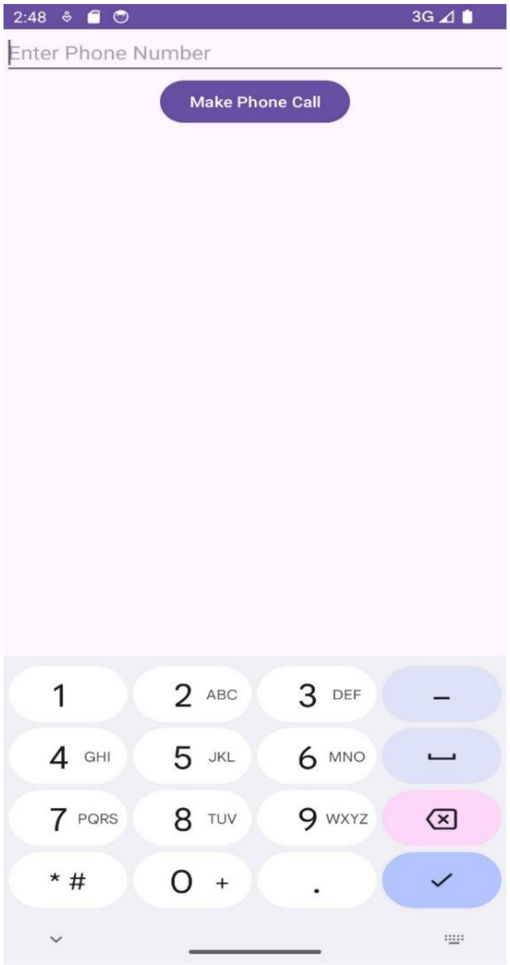
activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <EditText
        android:id="@+id/editTextPhoneNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Phone Number"
        android:inputType="phone" />
    <Button
        android:id="@+id/btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Make Phone Call"
        android:layout_below="@id/editTextPhoneNumber"
        android:layout_centerHorizontal="true"
        android:onClick="makePhoneCall" />
</RelativeLayout>
```

ActivityManifest:

```
<?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-feature
        android:name="android.hardware.telephony"
        android:required="false" />
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Phone_calls"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



RESULT:

The application that Phone Calls on the screen is developed and tested using android studio.

DATE:-

Session-VI: Android Audio Capture, Audio Manager, Audio Complete.
Develop a program for Audio Capture.

AIM: To develop an application for Audio Capture

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.audio
2. Modify src/MainActivity.java file to add audio code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java:

```
package com.example.audio;
import android.Manifest;
import android.content.pm.PackageManager;
import android.media.MediaRecorder;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import java.io.IOException;
```

```
public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_PERMISSION_CODE = 100;
    private MediaRecorder mediaRecorder;
    private String audioFilePath;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Request necessary permissions
        requestPermissions();
        // Set the path to store the audio file
        audioFilePath = getExternalCacheDir().getAbsolutePath() + "/audio.3gp";
    }
    private void requestPermissions() {
        if (ContextCompat.checkSelfPermission(this, Manifest.permission.RECORD_AUDIO) !=
PackageManager.PERMISSION_GRANTED ||
                ContextCompat.checkSelfPermission(this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.RECORD_AUDIO,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, REQUEST_PERMISSION_CODE);
        }
    }
    public void startRecording(View view) {
        try {
            mediaRecorder = new MediaRecorder();
            mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
            mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
            mediaRecorder.setOutputFile(audioFilePath);
            mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
            mediaRecorder.prepare();
            mediaRecorder.start();
            // Toggle visibility of buttons
            Button startButton = findViewById(R.id.startRecordingButton);
            Button stopButton = findViewById(R.id.stopRecordingButton);
            startButton.setVisibility(View.GONE);
            stopButton.setVisibility(View.VISIBLE);
            Toast.makeText(this, "Recording started", Toast.LENGTH_SHORT).show();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public void stopRecording(View view) {
        mediaRecorder.stop();
        mediaRecorder.release();
        mediaRecorder = null;
        // Toggle visibility of buttons
        Button startButton = findViewById(R.id.startRecordingButton);
        Button stopButton = findViewById(R.id.stopRecordingButton);
        startButton.setVisibility(View.VISIBLE);
        stopButton.setVisibility(View.GONE);
        Toast.makeText(this, "Recording stopped", Toast.LENGTH_SHORT).show();
    }
}
```

activity_main.xml:

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

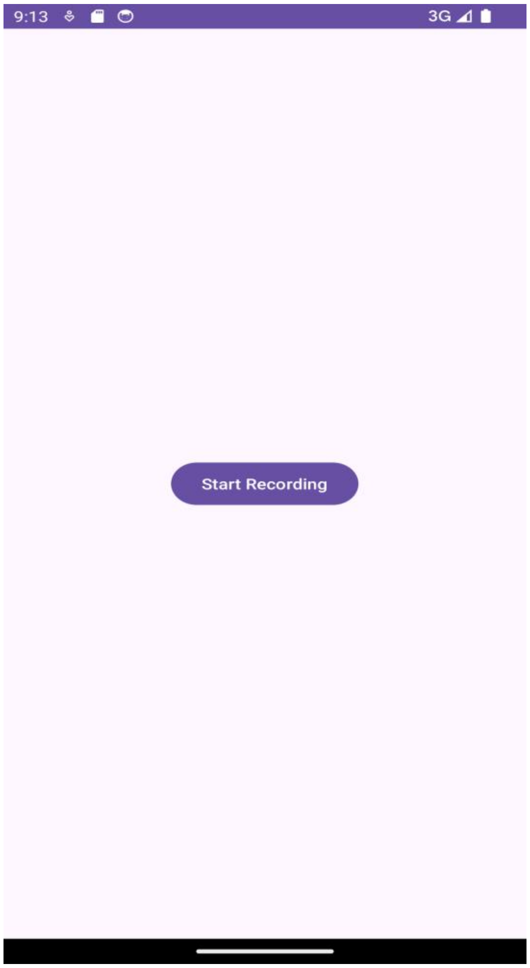
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/startRecordingButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Recording"
        android:layout_centerInParent="true"
        android:onClick="startRecording" />
    <Button
        android:id="@+id/stopRecordingButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Stop Recording"
        android:layout_below="@id/startRecordingButton"
        android:layout_centerHorizontal="true"
        android:onClick="stopRecording"
        android:visibility="gone" />
</RelativeLayout>
```

ActivityManifest:

```
<?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.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
```

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

OUTPUT:



RESULT:

The application that Audio capture on the screen is developed and tested using android studio.

DATE:-

Session-VII: SqLite (CRUD).

Develop an Android Program to connect Database and Develop Database Operations using SQLite.

AIM: To develop an Android Program to connect Database and Develop Database Operations using SQLite.

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.sqlite
2. Modify src/MainActivity.java file to add sqlite code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java:

```
package com.example.sqlite;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.view.View;
import android.os.Bundle;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends Activity implements OnClickListener {
   EditText editEmpid,editName,editSalary;
```

```
    Button btnAdd,btnDelete,btnModify,btnView,btnViewAll;
    SQLiteDatabase db;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editEmpid=(EditText)findViewById(R.id.editEmpid);
        editName=(EditText)findViewById(R.id.editName);
        editSalary=(EditText)findViewById(R.id.editSalary);
        btnAdd=(Button)findViewById(R.id.btnAdd);
        btnDelete=(Button)findViewById(R.id.btnDelete);
        btnModify=(Button)findViewById(R.id.btnModify);
        btnView=(Button)findViewById(R.id.btnView);
        btnViewAll=(Button)findViewById(R.id.btnViewAll);
        btnAdd.setOnClickListener(this);
        btnDelete.setOnClickListener(this);
        btnModify.setOnClickListener(this);
        btnView.setOnClickListener(this);
        btnViewAll.setOnClickListener(this);
        db=openOrCreateDatabase("Employee DB",Context.MODE_PRIVATE,null);
        db.execSQL("CREATE TABLE IF NOT EXISTS employee (empid VARCHAR, name
VARCHAR, salary VARCHAR)");
    }
    public void onClick(View view)
    {
        if(view==btnAdd)
        {
            if(editEmpid.getText().toString().trim().length()==0 ||
editName.getText().toString().trim().length()==0 || editSalary.getText().toString().trim().length()==0)
            {
                showMessage("Error","please enter all values");
                return;
            }
            db.execSQL("INSERT INTO employee VALUES('" + editEmpid.getText() + "','" +
editName.getText() + "','" + editSalary.getText() + "')");
            showMessage("success","Record added");
            clearText();
        }
        if(view==btnDelete)
        {
            if(editEmpid.getText().toString().trim().length()==0)
            {
                showMessage("Error","Please enter Employee id");
                return;
            }
            Cursor c = db.rawQuery("SELECT * FROM employee WHERE empid = ?", new
String[]{editEmpid.getText().toString()});
            if (c.moveToFirst()) {
                do {
                    db.execSQL("DELETE FROM employee WHERE empid = ?", new
String[]{editEmpid.getText().toString()});
                } while (c.moveToNext());
                showMessage("Success", "Record(s) deleted");
            } else {
                showMessage("Error", "Invalid Employee ID");
            }
            c.close();
            clearText();
        }
        if(view==btnModify)
        {
            if(editEmpid.getText().toString().trim().length()==0)
            {
                showMessage("Error","Please enter Employee id");
                return;
            }
            Cursor c = db.rawQuery("SELECT * FROM employee WHERE empid = '" +
editEmpid.getText() + "'", null);
            if(c.moveToFirst())
            {
                db.execSQL("UPDATE employee SET name='" + editName.getText() + "', salary='" +
editSalary.getText() + "' WHERE empid='" + editEmpid.getText() + "'");
                showMessage("success","Record modified");
            }
            else {
                showMessage("Error","Invalid Employee id");
            }
            clearText();
        }
        if (view == btnView) {
            String empId = editEmpid.getText().toString().trim();
            if (empId.length() == 0) {
                showMessage("Error", "Please enter Employee ID");
                return;
            }
            Cursor c = db.rawQuery("SELECT * FROM employee WHERE empid = ?", new
String[]{empId});
            try {
                if (c.moveToFirst()) {
                    editName.setText(c.getString(1));
                    editSalary.setText(c.getString(2));
                } else {
                    showMessage("Error", "Invalid Employee ID");
                }
            } finally {
                c.close(); // Close the cursor after using it
            }
            clearText();
        }
        if(view==btnViewAll)
        {
            Cursor c=db.rawQuery("SELECT * FROM employee ",null);
            if(c.getCount()==0)
            {
                showMessage("Error","No records found");
                return;
            }
            StringBuffer buffer=new StringBuffer();
            while(c.moveToNext())
            {
                buffer.append("Employee id:"+c.getString(0)+"\n");
                buffer.append("Name:"+c.getString(1)+"\n");
                buffer.append(" Salary:"+c.getString(2)+"\n");
            }
            showMessage("Employee details Details",buffer.toString());
        }
    }
    public void showMessage(String title,String message)
    {
        Builder builder=new Builder(this);
```

```
        builder.setCancelable(true);
        builder.setTitle(title);
        builder.setMessage(message);
        builder.show();
    }
    public void clearText()
    {
        editEmpid.setText(" ");
        editSalary.setText(" ");
        editEmpid.requestFocus();
    }
}
```

ACTIVITY_MAIN.XML:

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout"
    android:stretchColumns="0"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <TextView
        android:text="@string/title"
        android:layout_x="110dp"
        android:layout_y="10dp"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
    <TextView
        android:text="@string/empid"
        android:layout_x="30dp"
```

```
     android:layout_y="50dp"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"/>
 <EditText
     android:id="@+id/editEmpid"
     android:layout_width="150dp"
     android:layout_height="40dp"
     android:layout_x="152dp"
     android:layout_y="52dp"
     android:inputType="number" />
 <TextView
     android:text="@string/name"
     android:layout_x="30dp"
     android:layout_y="100dp"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"/>
 <EditText
     android:id="@+id/editName"
     android:inputType="text"
     android:layout_x="150dp"
     android:layout_y="100dp"
     android:layout_width="150dp"
     android:layout_height="40dp"/>
<TextView
     android:text="@string/salary"
     android:layout_x="30dp"
     android:layout_y="150dp"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"/>
```

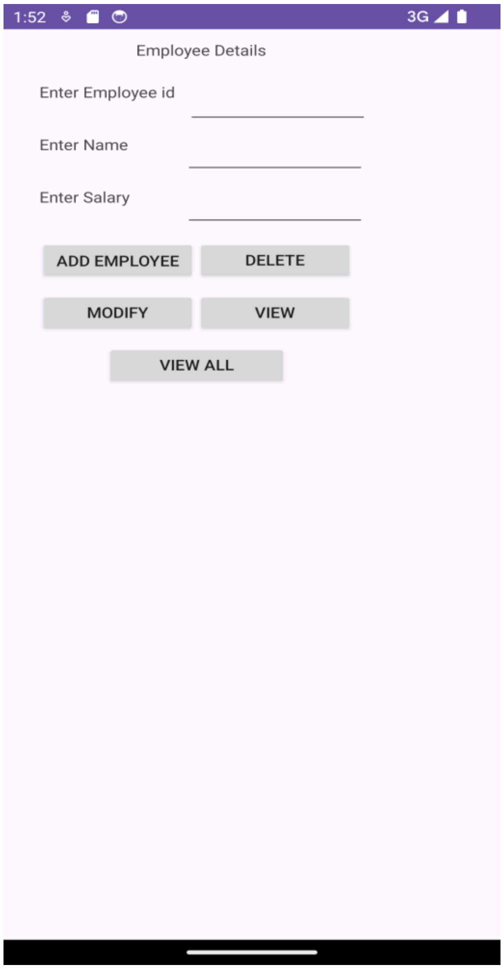
```
<EditText
    android:id="@+id/editSalary"
    android:inputType="number"
    android:layout_x="150dp"
    android:layout_y="150dp"
    android:layout_width="150dp"
    android:layout_height="40dp"/>
  <Button
    android:id="@+id/btnAdd"
    android:text="@string/add"
    android:layout_x="30dp"
    android:layout_y="200dp"
    android:layout_width="130dp"
    android:layout_height="40dp"/>
  <Button
    android:id="@+id/btnDelete"
    android:text="@string/delete"
    android:layout_x="160dp"
    android:layout_y="200dp"
    android:layout_width="130dp"
    android:layout_height="40dp"/>
 <Button
    android:id="@+id/btnModify"
    android:text="@string/modify"
    android:layout_x="30dp"
    android:layout_y="250dp"
    android:layout_width="130dp"
    android:layout_height="40dp"/>
 <Button
    android:id="@+id/btnView"
```

```
        android:text="@string/view"
        android:layout_x="160dp"
        android:layout_y="250dp"
        android:layout_width="130dp"
        android:layout_height="40dp"/>
    <Button
        android:id="@+id/btnViewAll"
        android:text="@string/view_all"
        android:layout_x="85dp"
        android:layout_y="300dp"
        android:layout_width="150dp"
        android:layout_height="40dp"/>
</AbsoluteLayout>
```

String.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Employee detail</string>
    <string name="hello">Hello world,Employee detail Activity!</string>
    <string name="title">Employee Details</string>
    <string name="empid">Enter Employee id</string>
    <string name="name">Enter Name</string>
    <string name="salary">Enter Salary</string>
    <string name="add">Add Employee</string>
    <string name="delete">Delete Employee</string>
    <string name="modify">Modify Employee</string>
    <string name="view">view Employee</string>
    <string name="view_all">View All Employees</string>
</resources>
```

OUTPUT:



1:52 3G

Employee Details

Enter Employee id

Enter Name

Enter Salary

RESULT:

The application that SQLite on the screen is developed and tested using android studio**.**

DATE:-

Session-VIII: Text to Speech

Develop an Application for Text to Speech.

AIM: To develop an application for Text to Speech.

PROCEDURE:

5. You will use Android studio to create an Android application under a package com.example.textspeech
6. Modify src/MainActivity.java file to add textspeech code.
7. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
8. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java

```
package com.example.textspeech;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Locale;
public class MainActivity extends AppCompatActivity {
    private TextToSpeech textToSpeech;
    private EditText editText;
    private Button btnSpeak;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = findViewById(R.id.editText);
        btnSpeak = findViewById(R.id.btnSpeak);
        textToSpeech = new TextToSpeech(getApplicationContext(), new
TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if (status != TextToSpeech.ERROR) {
                    textToSpeech.setLanguage(Locale.US);
                } else {
                    Toast.makeText(MainActivity.this, "Initialization failed",
Toast.LENGTH_SHORT).show();
                }
            }
        });
        btnSpeak.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String text = editText.getText().toString();
                if (!text.isEmpty()) {
                    textToSpeech.speak(text, TextToSpeech.QUEUE_FLUSH, null, null);
                } else {
                    Toast.makeText(MainActivity.this, "Please enter some text",
Toast.LENGTH_SHORT).show();
```

```
                }
            }
        });
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        if (textToSpeech != null) {
            textToSpeech.stop();
            textToSpeech.shutdown();
        }
    }
}
```

activity_main.xml

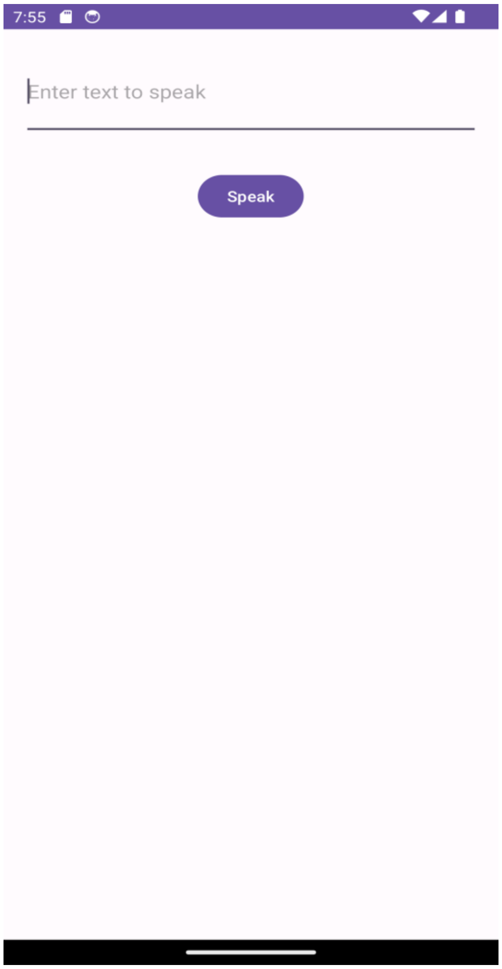
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter text to speak"
        android:layout_margin="16dp"
        android:inputType="textMultiLine"
        android:minLines="3" />
    <Button
```

```
        android:id="@+id/btnSpeak"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editText"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="Speak" />
</RelativeLayout>
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_CALENDAR" />
    <uses-permission android:name="com.android.vending.BILLING" />
    <uses-permission android:name="android.permission.RECORD_AUDIO" />
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.TextSpeech"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



RESULT:

The application that Text Speech on the screen is developed and tested using android studio.

DATE:-

Session-IX: Google Maps

Develop an application for identifying the current location.

AIM: To develop an application for identifying the current location.

PROCEDURE:

1. You will use Android studio to create an Android application under a package com.example.current_location
2. Modify src/MainActivity.java file to add current_location code.
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity:

```
package com.example.current_location;
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import com.google.android.gms.location.FusedLocationProviderClient;
import com.google.android.gms.location.LocationCallback;
import com.google.android.gms.location.LocationRequest;
import com.google.android.gms.location.LocationResult;
import com.google.android.gms.location.LocationServices;
public class MainActivity extends AppCompatActivity {
    private static final int REQUEST_PERMISSION_CODE = 1001;
    private FusedLocationProviderClient fusedLocationClient;
    private LocationCallback locationCallback;
    private boolean isTracking = false;
    private Button startStopButton;
    private TextView latitudeTextView, longitudeTextView;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        startStopButton = findViewById(R.id.startStopButton);
        latitudeTextView = findViewById(R.id.latitudeTextView);
        longitudeTextView = findViewById(R.id.longitudeTextView);
        fusedLocationClient = LocationServices.getFusedLocationProviderClient(this);
        if (checkPermission()) {
            startLocationUpdates();
        } else {
            requestPermission();
        }
    }
    private boolean checkPermission() {
        return ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) == PackageManager.PERMISSION_GRANTED
&&
                ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) ==
PackageManager.PERMISSION_GRANTED;
    }
    private void requestPermission() {
        ActivityCompat.requestPermissions(this, new
String[]{Manifest.permission.ACCESS_FINE_LOCATION,
Manifest.permission.ACCESS_COARSE_LOCATION}, REQUEST_PERMISSION_CODE);
    }
    public void startStopTracking(View view) {
        if (!isTracking) {
            startLocationUpdates();
            startStopButton.setText("Stop Tracking");
            isTracking = true;
        } else {
            stopLocationUpdates();
            startStopButton.setText("Start Tracking");
            isTracking = false;
        }
    }
    private void startLocationUpdates() {
        LocationRequest locationRequest = LocationRequest.create();
        locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);
        locationRequest.setInterval(5000); // 5 seconds
        locationCallback = new LocationCallback() {
            @Override
            public void onLocationResult(LocationResult locationResult) {
                if (locationResult == null) {
                    return;
                }
                for (Location location : locationResult.getLocations()) {
                    // Update UI with location data
                    latitudeTextView.setText("Latitude: " + location.getLatitude());
                    longitudeTextView.setText("Longitude: " + location.getLongitude());
                }
            }
        };
        if (ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED
&& ActivityCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    ActivityCompat#requestPermissions
            // here to request the missing permissions, and then overriding
            //   public void onRequestPermissionsResult(int requestCode, String[] permissions,
            //                                          int[] grantResults)
            // to handle the case where the user grants the permission. See the documentation
            // for ActivityCompat#requestPermissions for more details.
            return;
        }
        fusedLocationClient.requestLocationUpdates(locationRequest, locationCallback, null);
    }
    private void stopLocationUpdates() {
        fusedLocationClient.removeLocationUpdates(locationCallback);
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if (requestCode == REQUEST_PERMISSION_CODE) {
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED &&
                    grantResults[1] == PackageManager.PERMISSION_GRANTED) {
                startLocationUpdates();
            } else {
                Toast.makeText(this, "Permission denied. Cannot track location.",
Toast.LENGTH_SHORT).show();
            }
        }
    }
}
```

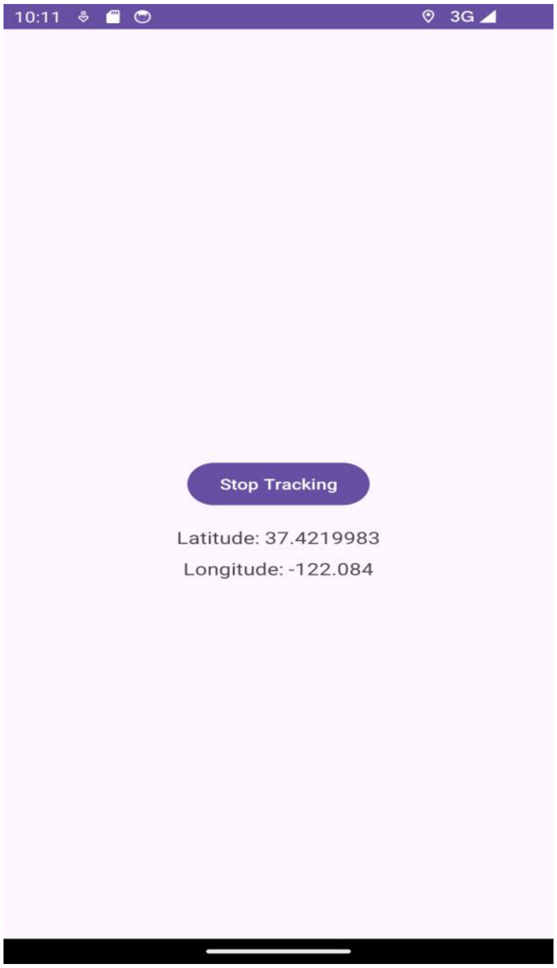
activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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/startStopButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:text="Start Tracking"
        android:onClick="startStopTracking"/>
    <TextView
        android:id="@+id/latitudeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/startStopButton"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="16dp"
        android:text="Latitude: "
        android:textSize="16sp"/>
    <TextView
        android:id="@+id/longitudeTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/latitudeTextView"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="8dp"
        android:text="Longitude: "
        android:textSize="16sp"/>
</RelativeLayout>
```

ActivityManifest.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_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Current_location"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



Latitude: 37.4219983

Longitude: -122.084

RESULT:

The application that Google Maps on the screen is developed and tested using android studio