EXP NO: 1 DISPLAY A TOAST MESSAGE

DATE:

AIM:

To display a toast message in android application

ALGORITHM:

Step 1: Start the program

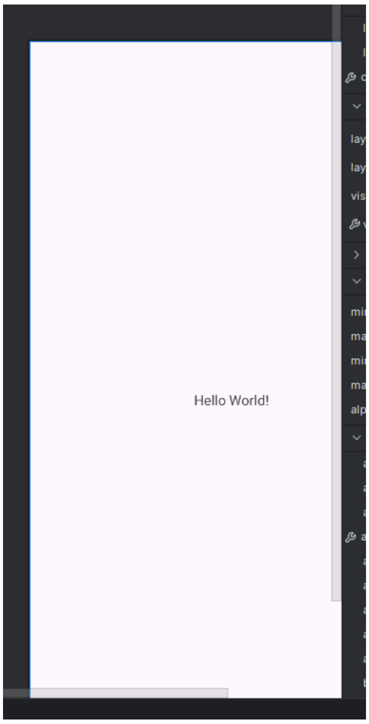
Step 2: Create New file and put your file name

Step 3: Click Empty View Activity

Step 4: In MainActivity.java, give the source code for Toast message.

Step 5: Run the Program.

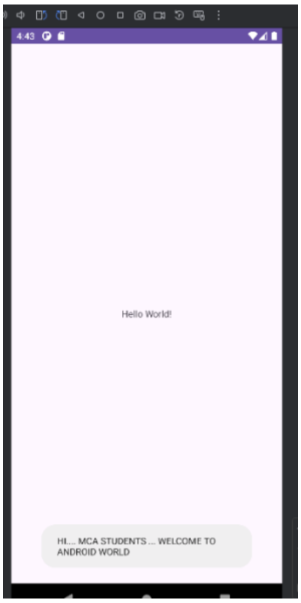
DESIGN VIEW INPUT WINDOW:



CODE: MainActivity.java

```
package com.example.kktoastmessage;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Toast.makeText(this,&quot;HI.... MCA STUDENTS ... WELCOME TO ANDROID WORLD&quot;,
Toast.LENGTH_LONG).show();
}
```

OUTPUT



```
Hello World!

HI.... MCA STUDENTS ... WELCOME TO
ANDROID WORLD
```

RESULT:

Thus, The Android Application has been developed and run successfully.

EX NO: 2 DESIGN LOGIN PAGE AND DISPLAY MESSAGE USING TOAST MESSAGE AND VALIDATE USER

DATE:

AIM:

To design login page and display message using toast message and validate the user.

ALGORITHM:

Step 1: Start the program

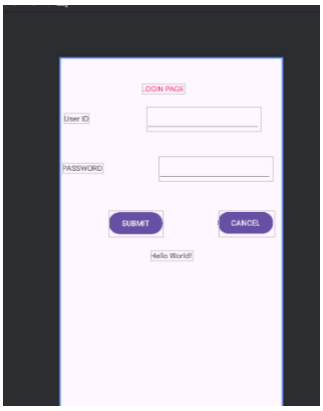
Step 2: Choose the 2 TextBox, 2 Buttons drag and drop in design page.

Step 3: Set the widgets for all components

Step 4: Give the source code for login page inside a MainActivity.java

Step 5: Run the program

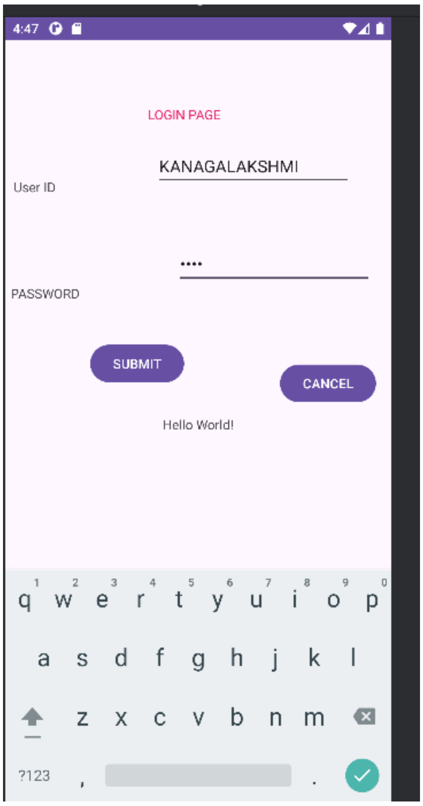
DESIGN VIEW INPUT WINDOW:



CODE: MainActivity.java

```
package com.example.loginpage;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.os.Bundle;
import java.text.BreakIterator;
public class MainActivity extends AppCompatActivity {
    EditText u, p;
    TextView disp;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        u = (EditText) findViewById(R.id.uid);
        p = (EditText) findViewById(R.id.pwd);
       /* disp=(EditText) findViewById(R.id.tv2);*/
    }public void logFun(View v){
        String uname = u.getText().toString();
        String password = p.getText().toString();
        /*if (uname.contentEquals("admin") && pwd.contentEquals("kanagalakshmi")
)*/Toast.makeText(getApplicationContext(),"welcome" +
uname,Toast.LENGTH_LONG).show();
            disp.setText(u.getText());
            /* else
            Toast.makeText(getApplicationContext(),"invalid
user",Toast.LENGTH_LONG).show();*/
    }}
```

OUTPUT:



LOGIN PAGE

User ID: KANAGALAKSHMI

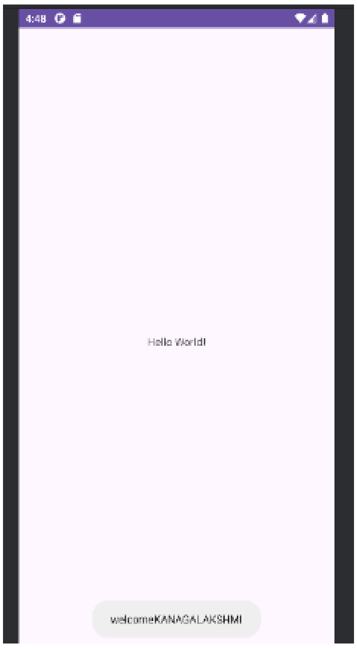
PASSWORD: ••••

SUBMIT

CANCEL

Hello World!

OUTPUT:



Hello World!

welcomeKANAGALAKSHMI

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO: 3 Date and Time picker and an Image Demo

DATE:

AIM:

To display a current date and time picker and an image demo

ALGORITHM:

Step 1: Create New Project

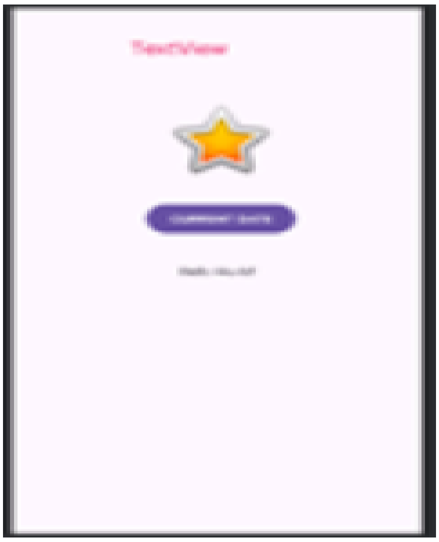
Step 2: Select Empty View Activity

Step 3: Add Four Components Text View, Button, Image View, Calendar

Step 4: In MainActivity.java, give the source code for date and time picker

Step 5: Run the program.

DESIGN VIEW INPUT WINDOW:



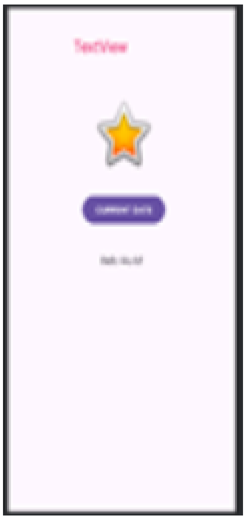
CODE: XML

```
<?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:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="138dp"
        android:layout_marginTop="39dp"
        android:layout_marginEnd="215dp"
        android:text="TextView"
        android:textColor="#E91E63"
        android:textSize="24sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

<Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="164dp"
        android:layout_marginTop="204dp"
        android:layout_marginEnd="157dp"
            android:text="CURRENT DATE AND TIME"
        android:onClick="showTime"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

<ImageView
        android:id="@+id/imageView3"
        android:layout_width="265dp"
        android:layout_height="117dp"
        android:layout_marginStart="76dp"
        android:layout_marginTop="124dp"
        android:layout_marginEnd="70dp"
        android:layout_marginBottom="28dp"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@android:drawable/btn_star_big_on" />

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

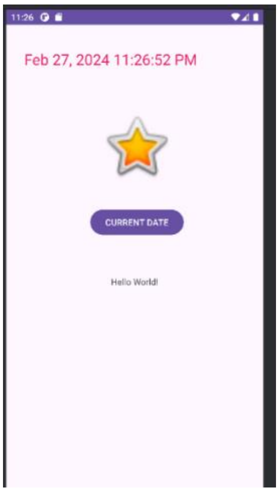
CODE: MainActivity.java

```
package com.example.kktimerandimage;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.TextView;
import com.example.kktimerandimage.R;

import com.example.kktimerandimage.R;

import java.text.DateFormat;

import java.util.Date;

public class MainActivity extends AppCompatActivity {

@Override

 protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_main);

}

public void showTime(View v)

{

TextView tv=(TextView)findViewById(R.id.textView);

Date d=new Date();

String s=DateFormat.getDateTimeInstance().format(d);

tv.setText(s);

}}
```

OUTPUT



TextView



Feb 27, 2024 11:26:52 PM

Hello World!

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO: 4 Implicit Intent (connecting SRM WebSite)

DATE:

AIM:

To Implicity connecting to SRM Website using Intent in Android Application

ALGORITHM:

Step 1: Create New Project

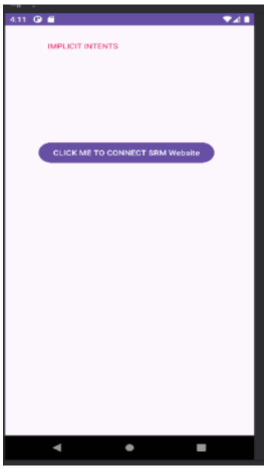
Step 2 : Select Empty View Activity

Step 3: Add the TextView, Button components.

Step 4: In the MainActivity.java, implement the code to handle the button click and create an implicit intent to open a web browser.

Step 5: Run the app.

DESIGN VIEW INPUT WINDOW:



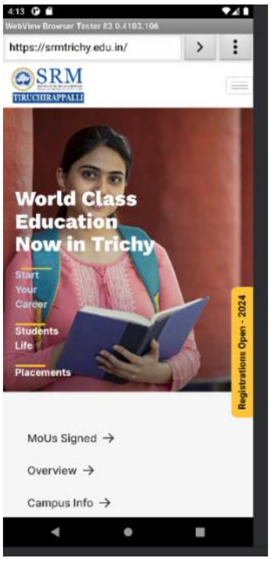
CODE: XML

```
<?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"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="70dp"
        android:layout_marginTop="32dp"
        android:layout_marginEnd="334dp"
        android:text="IMPLICIT INTENTS"
        android:textColor="#E91E63"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.0" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="155dp"
        android:layout_marginTop="135dp"
        android:layout_marginEnd="166dp"
        android:layout_marginBottom="497dp"
        android:text="CLICK ME TO CONNECT SRM Website"
android:onClick="connect"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />


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

CODE: MainActivity.java

```
package com.example.kkintentintext;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.net.Uri;

import android.os.Bundle;
import android.view.View;

import android.net.Uri;
import android.os.Bundle;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

    }
    public void connect(View v)
    {
        Intent i=new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse("https://srmtrichy.edu.in/"));
        startActivity(i);
    }

}
```

OUTPUT



WebView Browser Tester 83.0.4103.106

https://srmtrichy.edu.in/

World Class Education Now in Trichy

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO: 5 Explicit Intent (connecting Activity1 to activity2)

DATE:

AIM:

To Explicity connecting Activity 1 to Activity 2 using intent in Android Application.

ALGORITHM:

Step 1: Open Android Studio and create a new project with an Empty Activity.

Step 2: Add the Textview and Button components

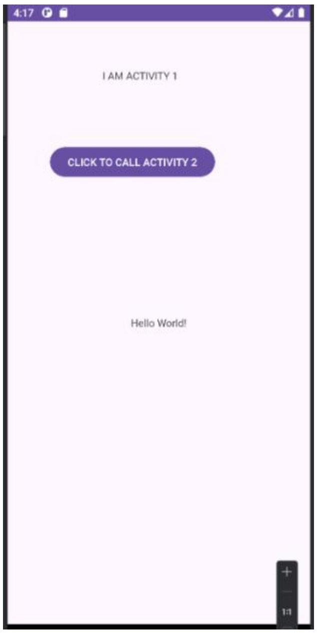
Step 3: Create A New Activity

1. Right-click on the java folder in your project.
2. Choose New-> Activity->Empty Activity

Step 4: In the MainActivity.java, implement the code to handle the button click and create an explicit intent to open SecondActivity.

Step 5: Run the app.

DESIGN VIEW INPUT WINDOW:



CODE: 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:id="@+id/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" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="125dp"
        android:layout_marginTop="145dp"
        android:layout_marginEnd="197dp"
        android:layout_marginBottom="163dp"
        android:onClick="show"
        android:text="CLICK TO CALL ACTIVITY 2"
        app:layout_constraintBottom_toTopOf="@+id/textView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="151dp"
        android:layout_marginTop="53dp"
        android:layout_marginEnd="202dp"
        android:layout_marginBottom="73dp"
        android:text="I AM ACTIVITY 1"
        app:layout_constraintBottom_toTopOf="@+id/button"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

CODE: XML (Activity2)

```
<?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=".MainActivity2">

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="131dp"
        android:layout_marginTop="218dp"
        android:layout_marginEnd="123dp"
        android:layout_marginBottom="465dp"
        android:text="SHOW MESSAGE"
        android:onClick="display"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="287dp"
        android:layout_height="13dp"
        android:layout_marginStart="197dp"
        android:layout_marginTop="107dp"
        android:layout_marginEnd="157dp"
        android:layout_marginBottom="339dp"
        android:text="TextView"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.491"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/button2"
        app:layout_constraintVertical_bias="1.0" />

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="164dp"
        android:layout_marginTop="47dp"
        android:layout_marginEnd="143dp"
        android:layout_marginBottom="152dp"
        android:text="I AM ACTIVITY 2"
        app:layout_constraintBottom_toTopOf="@+id/button2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

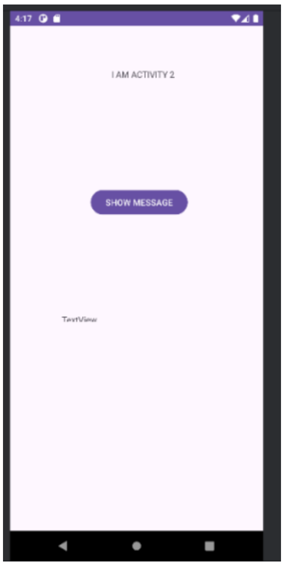
CODE: MainActivity.java

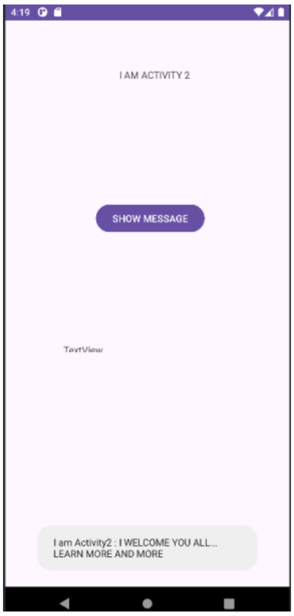
```
package com.example.kkexplicit;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;

import android.os.Bundle;
import android.view.View;

import android.net.Uri;
import android.os.Bundle;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

    public void show(View v) {
        Intent i = new Intent(getApplicationContext(), MainActivity2.class);
        startActivity(i);
    }
}
```

CODE: MainActivity.java Activity2

```
package com.example.kkexplicit;
import androidx.appcompat.app.AppCompatActivity;
import android.view.View;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity2 extends AppCompatActivity {
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
    }
    public void display(View v)
    {

        Toast.makeText(getApplicationContext(),"I am Activity2 : I WELCOME YOU
ALL... LEARN MORE AND MORE",Toast.LENGTH_LONG).show();

   }}
```

OUTPUT





I am Activity2 : I WELCOME YOU ALL...
LEARN MORE AND MORE

RESULT:

Thus, The Android Application has been developed and run successfully.

Ex No: 6   STUDENTS REGISTRATION FORM USING LISTVIEW

DATE:

AIM:

To develop an android App to implement LinearLayout and ListView.

ALGORITHM:

Step 1: Create a New project -> Empty view activity ->Finish

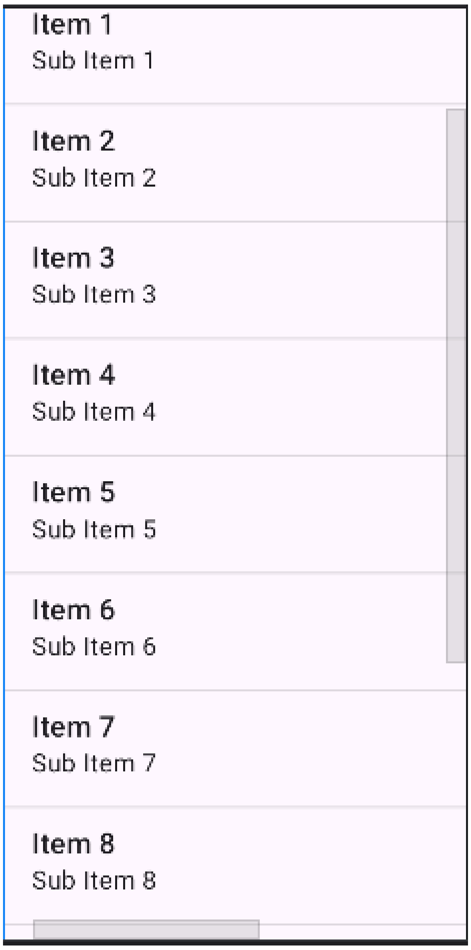
Step 2: Add the following components, LinearLayout and ListView

Step 3: Develop the MainActivity.java to add items in the ListView using ArrayAdaptor class.

Step 4: Build and Run the App.

Step 5: Stop the Project.

DESIGN VIEW INPUT WINDOW:



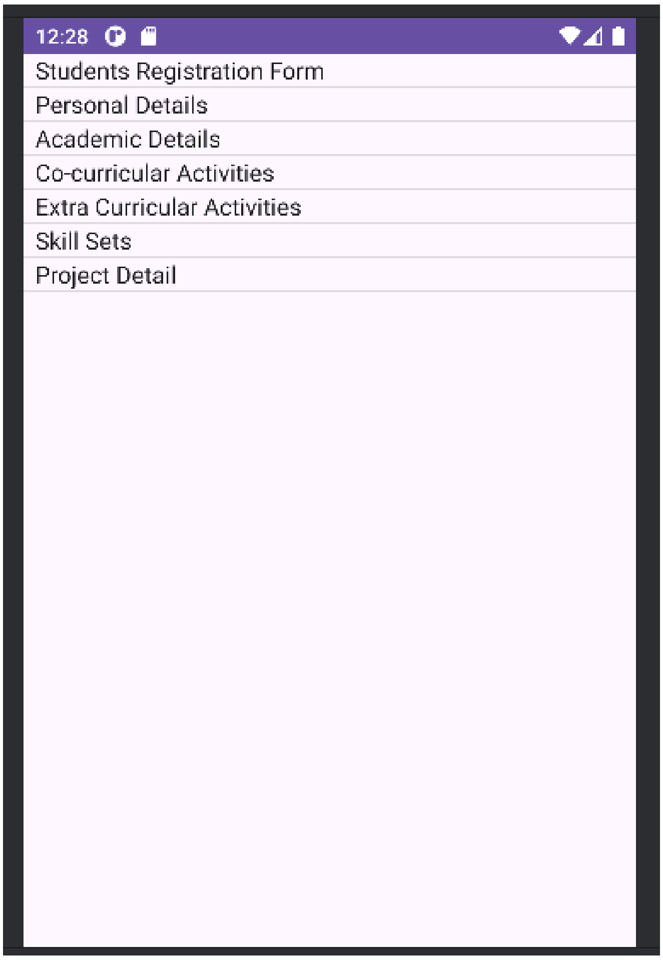
CODE: XML

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

<androidx.constraintlayout.widget.ConstraintLayoutxmlns: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">

<LinearLayout

        android:layout_width="409dp"

        android:layout_height="665dp"

        android:layout_marginStart="1dp"

        android:layout_marginTop="1dp"

        android:layout_marginEnd="1dp"

        android:orientation="vertical"

        app:layout_constraintBottom_toBottomOf="parent"

        app:layout_constraintEnd_toEndOf="parent"

        app:layout_constraintHorizontal_bias="0.0"

        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toBottomOf="@+id/textView"

        app:layout_constraintVertical_bias="0.0"></LinearLayout>

<ListView

        android:id="@+id/list"

        android:layout_width="match_parent"

        android:layout_height="match_parent" />

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

CODE: MainActivity.java

```
package com.example.kklistview;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
public class MainActivity extends AppCompatActivity {
ListView l;
   //Spinner s;

String registration[]= { "Students Registration Form","Personal Details", "Academic
Details", "Co-curricular Activities","Extra Curricular Activities", "Skill Sets",
"Project Detail"};
@Override
   protected void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      l = findViewById(R.id.list);
      ArrayAdapter<String> arr;

     arr=new
ArrayAdapter(MainActivity.this,android.R.layout.simple_spinner_item,registration);
      l.setAdapter(arr);
   }
}
```

OUTPUT



12:28

Students Registration Form
Personal Details
Academic Details
Co-curricular Activities
Extra Curricular Activities
Skill Sets
Project Detail

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO: 7 IMPLEMENT CONTEXT MENU

DATE:

AIM:

To develop an android Application to implement Context Menu.

ALGORITHM:

Step 1: Create a New Project in the Android Studio
Step 2: Select an Empty View Window
Step 3: Add the following components : TextView, RelativeLayout
Step 4: In the MainActivity.java, implement the code for context menu.
Step 5: Run the program.

CODE: XML (Design)

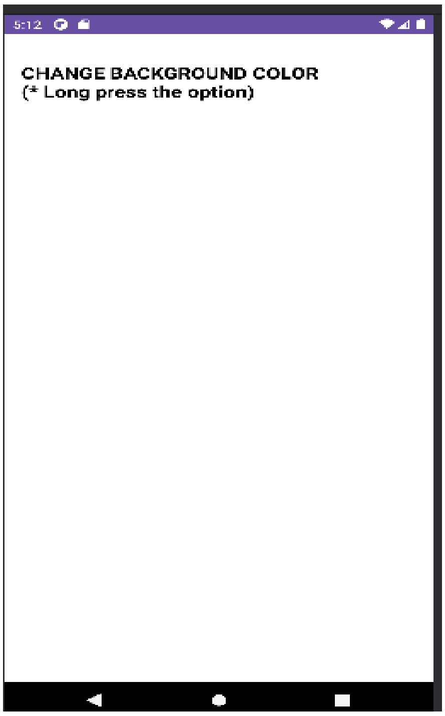
```
<?xml version="1.0" encoding="utf-8"?>
<!-- Relative Layout to display all the details -->
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/relLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#fff"
    android:padding="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp"
        android:text="Long press me!"
        android:textColor="#000"
        android:textSize="20sp"
        android:textStyle="bold" />
</RelativeLayout>
```

CODE: MainActivity.java

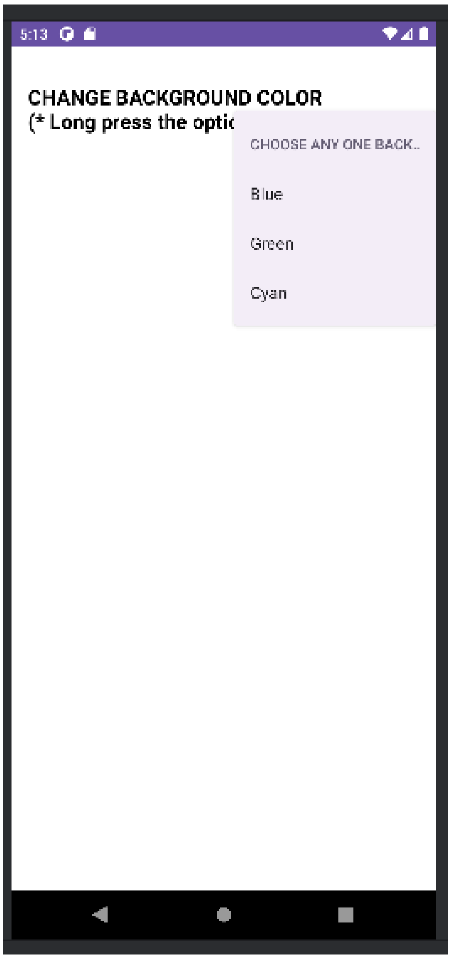
```
          import android.graphics.Color;
          import android.os.Bundle;
          import android.view.ContextMenu;
          import android.view.MenuItem;
          import android.view.View;
          import android.widget.RelativeLayout;
          import android.widget.TextView;
          import androidx.appcompat.app.AppCompatActivity;
          public class MainActivity extends AppCompatActivity {
              TextView textView;
              RelativeLayout relativeLayout;
             @Override
              protected void onCreate(Bundle savedInstanceState) {
                  super.onCreate(savedInstanceState);
                  setContentView(R.layout.activity_main);
```

```
 // Link those objects with their respective id's that we have given in .XML
file
        textView = (TextView) findViewById(R.id.textView);
        relativeLayout = (RelativeLayout) findViewById(R.id.relLayout);
  // here you have to register a view for context menu you can register any
view
          // like listview, image view, textview, button etc
        registerForContextMenu(textView);
  }
      @Override
    public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
        super.onCreateContextMenu(menu, v, menuInfo);
        // you can set menu header with title icon etc
        menu.setHeaderTitle("Choose a color");
        // add menu items
        menu.add(0, v.getId(), 0, "Yellow");
        menu.add(0, v.getId(), 0, "Gray");
        menu.add(0, v.getId(), 0, "Cyan");
    }
// menu item select listener
    @Override
    public boolean onContextItemSelected(MenuItem item) {
        if (item.getTitle() == "Yellow") {
            relativeLayout.setBackgroundColor(Color.YELLOW);
        } else if (item.getTitle() == "Gray") {
            relativeLayout.setBackgroundColor(Color.GRAY);
        } else if (item.getTitle() == "Cyan") {
            relativeLayout.setBackgroundColor(Color.CYAN);
        }
        return true;
    }
}
```

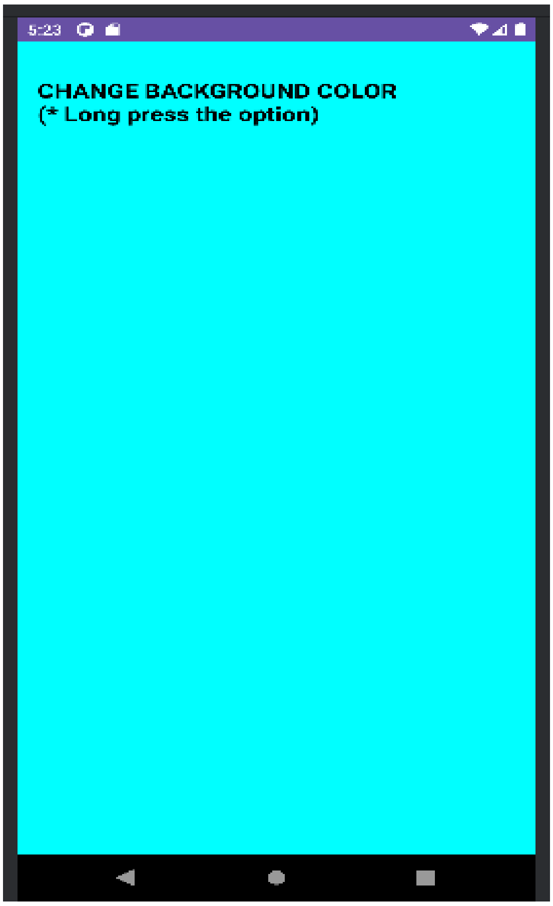
OUTPUT:



CHANGE BACKGROUND COLOR
(* Long press the option)



CHANGE BACKGROUND COLOR
(* Long press the optic



5:23

CHANGE BACKGROUND COLOR
(* Long press the option)

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO: 8 **SHARED PREFERENCES**

DATE:

AIM:

To implement the shared preference in Android Application.

ALGORITHM:

STEP 1: The first thing we need to do is to create one shared preferences file per app.

STEP 2: Name it with the package name of your app- unique and easy to associate with the app.

STEP 3: When you want to get the values, call the getSharedPreferences() method.

STEP4: To write to a SharedPreference file we can create a SharedPreferences.Editor by calling edit() on your sharedpreference.

STEP 5: To retrieve values from a shared preferences file, call methods such as getInt() and getString()

CODE: XML

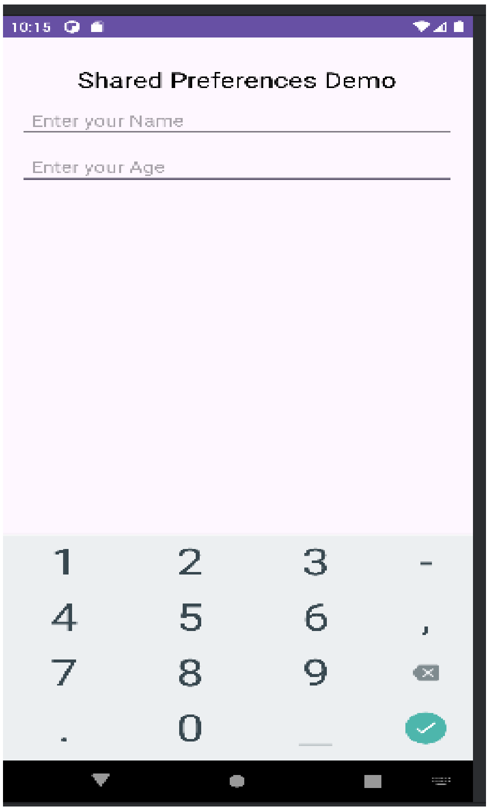
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    tools:ignore="HardcodedText">
  <TextView
        android:id="@+id/textview"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="32dp"
        android:text="Shared Preferences Demo"
        android:textColor="@android:color/black"
        android:textSize="24sp" />
  <!--EditText to take the data from the user and save the data in SharedPreferences-
->
    <EditText
        android:id="@+id/edit1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textview"
        android:layout_marginStart="16dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:hint="Enter your Name"
        android:padding="10dp" />
    <!--EditText to take the data from the user and save the data in
SharedPreferences-->

    <EditText

        android:id="@+id/edit2"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:layout_below="@+id/edit1"

        android:layout_marginStart="16dp"

        android:layout_marginTop="8dp"

        android:layout_marginEnd="16dp"

        android:hint="Enter your Age"

        android:inputType="number"

        android:padding="10dp" />

</RelativeLayout>
```

CODE: MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
   private EditText name, age;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      name = findViewById(R.id.edit1);
      age = findViewById(R.id.edit2);
   }

   // Fetch the stored data in onResume() Because this is what will be called when
the app opens again
   @Override
   protected void onResume() {
      super.onResume();
      // Fetching the stored data from the SharedPreference
```

```
        SharedPreferences sh = getSharedPreferences("MySharedPref",
MODE_PRIVATE);
        String s1 = sh.getString("name", "");
        int a = sh.getInt("age", 0);

        // Setting the fetched data in the EditTexts
        name.setText(s1);
        age.setText(String.valueOf(a));
    }

    // Store the data in the SharedPreference in the onPause() method
    // When the user closes the application onPause() will be called and data will be
stored
    @Override
    protected void onPause() {
        super.onPause();
        // Creating a shared pref object with a file name "MySharedPref" in private
mode
        SharedPreferences sharedPreferences =
getSharedPreferences("MySharedPref", MODE_PRIVATE);
        SharedPreferences.Editor myEdit = sharedPreferences.edit();

        // write all the data entered by the user in SharedPreference and apply
        myEdit.putString("name", name.getText().toString());
        myEdit.putInt("age", Integer.parseInt(age.getText().toString()));
        myEdit.apply();
    }
}
```

OUTPUT:

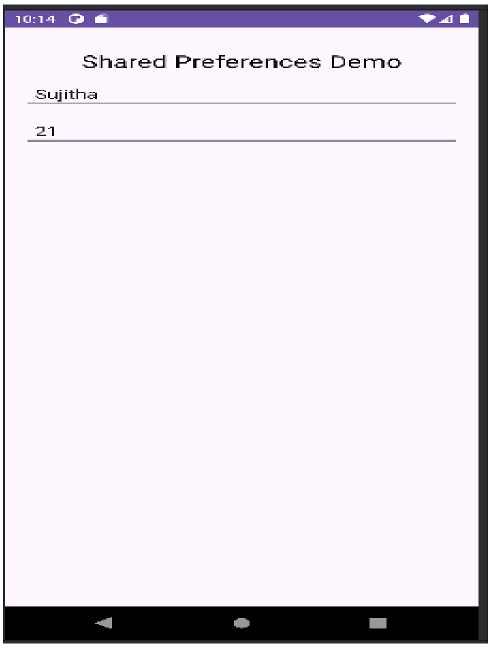


10:15

Shared Preferences Demo

Enter your Name

Enter your Age

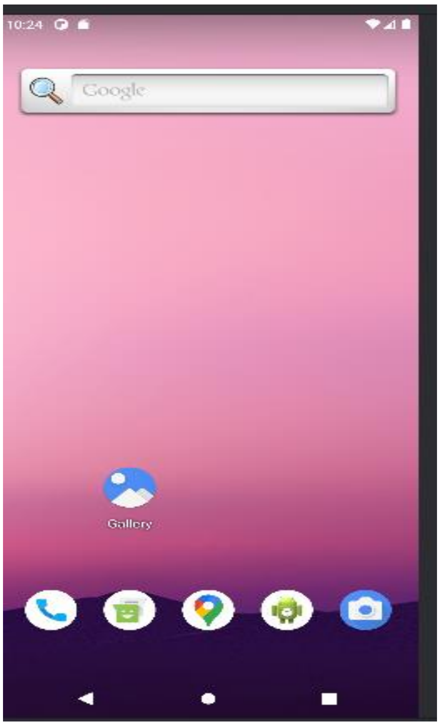


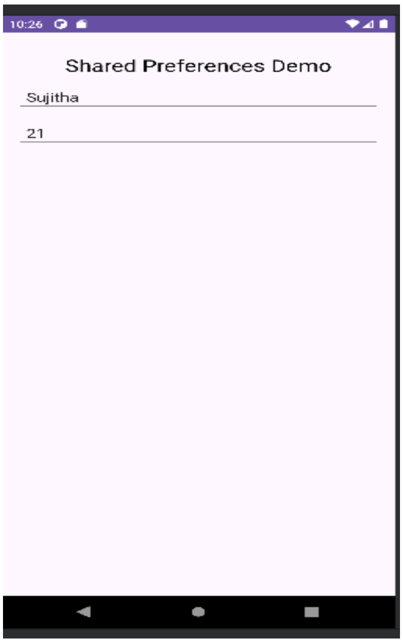
10:14

Shared Preferences Demo

Sujitha

21





Shared Preferences Demo

Sujitha

21

RESULT:

Thus, The Android Application has been developed and run successfully.

Ex No:9 **SQLite BANK DATABASE**

DATE:

AIM:

To implement the SQlite bank database in android application.

ALGORITHM:

STEP 1: Create a class that extends SQLiteOpenHelper to manage the creation and version management of the database.

STEP 2: Implement methods in your application to perform database operations using SQLiteDatabase class.

STEP 3: You can then use this BankDataSource class in your Android activities

STEP 4: Use SQL commands to create the database schema.

STEP 5: Create tables for each entity with the appropriate attributes.

STEP 6: Apply normalization techniques to organize the data efficiently.

CODE: XML

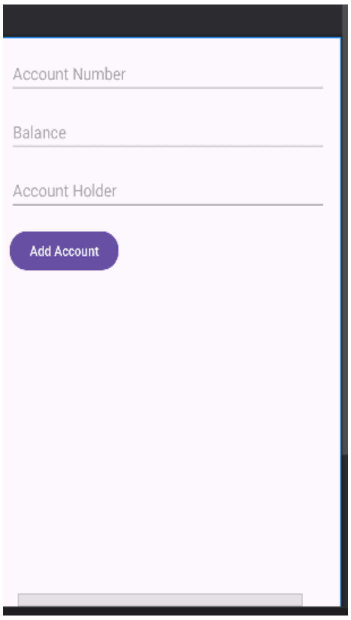
```
<!-- activity_main.xml -->
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
 <EditText
        android:id="@+id/editTextAccountNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Account Number"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"/>
    <EditText
        android:id="@+id/editTextBalance"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Balance"
        android:layout_below="@id/editTextAccountNumber"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"/>
<EditText
        android:id="@+id/editTextAccountHolder"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Account Holder"
        android:layout_below="@id/editTextBalance"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"/>
<Button
        android:id="@+id/buttonAddAccount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add Account"
        android:layout_below="@id/editTextAccountHolder"
        android:layout_marginTop="16dp"
        android:layout_marginStart="16dp"
        android:layout_marginEnd="16dp"/>
</RelativeLayout>
```

CODE: MainActivity.java

```
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.content.ContentValues;
import android.content.Context;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
public class BankDatabaseHelper extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "bank.db";
    private static final int DATABASE_VERSION = 1;
// Table name and column names
    public static final String TABLE_ACCOUNTS = "accounts";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_ACCOUNT_NUMBER = "account_number";
    public static final String COLUMN_BALANCE = "balance";
    public static final String COLUMN_ACCOUNT_HOLDER = "account_holder";
// SQL statement to create the accounts table
    private static final String DATABASE_CREATE = "create table " +
TABLE_ACCOUNTS + " ("
            + COLUMN_ID + " integer primary key autoincrement, "
            + COLUMN_ACCOUNT_NUMBER + " text not null, "
            + COLUMN_BALANCE + " real not null, "
            + COLUMN_ACCOUNT_HOLDER + " text not null);";
public BankDatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
 @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(DATABASE_CREATE);
    }
@Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_ACCOUNTS);
        onCreate(db);
    }
}
public class BankDataSource {
    private SQLiteDatabase database;
    private BankDatabaseHelper dbHelper;
public BankDataSource(Context context) {
        dbHelper = new BankDatabaseHelper(context);
    }
 public void open() throws SQLException {
        database = dbHelper.getWritableDatabase();
    }
```

```
public void close() {
        dbHelper.close();
    }
public long createAccount(String accountNumber, double balance, String accountHolder) {
        ContentValues values = new ContentValues();
        values.put(BankDatabaseHelper.COLUMN_ACCOUNT_NUMBER, accountNumber);
        values.put(BankDatabaseHelper.COLUMN_BALANCE, balance);
        values.put(BankDatabaseHelper.COLUMN_ACCOUNT_HOLDER, accountHolder);
        return database.insert(BankDatabaseHelper.TABLE_ACCOUNTS, null, values);
    }
BankDataSource dataSource = new BankDataSource(this);
dataSource.open();
long accountId = dataSource.createAccount("1234567890", 1000.00, "John Doe");
dataSource.close();
}
```

OUTPUT


Account Number

Balance

Account Holder

RESULT:

Thus, The Android Application has been developed and run successfully.

EXP NO:10 STUDENT APP USING SQLITE

DATE:

AIM:

To create an Student Application using Android studio for maintaining student details in sqlite**.**

ALGORITHM:

Step 1: Start the Android Studio.

Step 2: File -> New -> New Project ->Application Name ->StudentAppNext-

>Finish.

Step 3: Design the App (activity_main.xml) by adding the following controls. 4Button,2 Edit Text, 2 Text View.

Step 4: Assign the name for onClick event as inf,delf,updatef,viewf. Step 5:Write coding for MainActivity.java.

Step 5: Run the program Build -> BuildBundles/apk(s)->locate.

Step 6: Now the App is installed and verified in Bluestack.

Step 7:Close the project.

CODE: XML

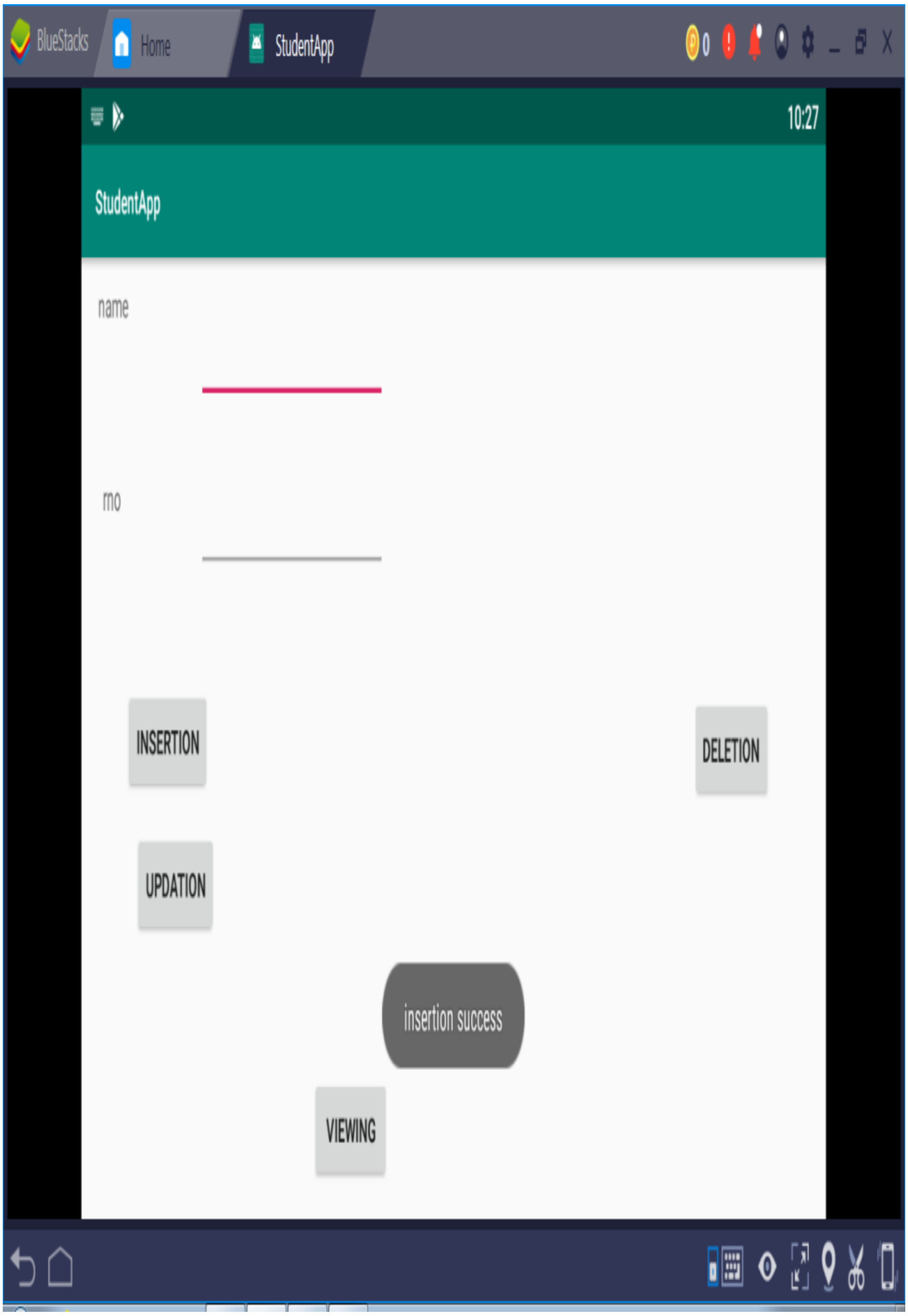
```
 <?xml version="1.0" encoding="utf-8"?>

<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=".MainActivity">


    <TextView
        android:id="@+id/textView"
        android:layout_width="84dp"
        android:layout_height="51dp"

        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="19dp"
        android:layout_marginLeft="19dp"
        android:layout_marginTop="10dp" android:text="name"
        />


    <TextView
        android:id="@+id/textView2"
        android:layout_width="75dp"
        android:layout_height="44dp"

        android:layout_alignEnd="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_marginTop="92dp"
        android:layout_marginEnd="3dp"
        android:text="rno" />


    <EditText
        android:id="@+id/editText"

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/textView"
        android:layout_marginStart="31dp"
        android:layout_marginBottom="-3dp"
        android:layout_toEndOf="@+id/textView2"
        android:ems="10"
        android:inputType="textPersonName" />


    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/editText"
android:layout_alignBottom="@+id/textView2"
android:ems="10" android:inputType="textPersonName"
/>


    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="51dp"
        android:layout_marginLeft="51dp"
        android:layout_marginTop="181dp"
        android:onClick="inf"
        android:text="insertion" />


    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignTop="@+id/button"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_marginTop="3dp"
        android:layout_marginEnd="64dp"
        android:layout_marginRight="64dp"
        android:onClick="delf"
        android:text="deletion" />


    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-8dp"
        android:layout_marginBottom="118dp"
        android:onClick="updatef"
        android:text="updation" />


    <Button
        android:id="@+id/button4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignEnd="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_marginTop="346dp"
        android:layout_marginEnd="-250dp"
        android:onClick="viewf"

android:text="viewing" />

 </RelativeLayout>
```

CODE: MainActivity.java

```
 package com.example.admin.studentapp;

 import android.content.Context;

 import android.database.Cursor;

 import android.database.sqlite.SQLiteDatabase;
 import android.support.v7.app.AppCompatActivity;
 import android.os.Bundle;
 import android.view.View;
 import android.widget.Button;
 import android.widget.EditText;
 import android.widget.TextView;
 import android.widget.Toast;


 public class MainActivity extends AppCompatActivity {

    EditText name,rno;

    Button i,d,u,v;
    SQLiteDatabase db;



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


        name=(EditText)findViewById(R.id.editText);

        rno=(EditText) findViewById(R.id.editText2);

        i=(Button)findViewById(R.id.button);
        d=(Button)findViewById(R.id.button2);
        u=(Button)findViewById(R.id.button3);
        v=(Button)findViewById(R.id.button4);
    db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
    db.execSQL("CREATE TABLE IF NOT EXISTS student(name VARCHAR,rno
VARCHAR);");
  }

  public void inf(View v)
  {
    db.execSQL("INSERT INTO student
VALUES('"+name.getText()+"','"+rno.getText()+"');");



    Toast.makeText(getApplicationContext(),"insertion
success",Toast.LENGTH_LONG).show();
    clearText();
  }

  public void delf(View v)
  {
    Cursor c=db.rawQuery("SELECT * FROM student WHERE
rno='"+rno.getText()+"'", null);
    if(c.moveToFirst())
    {
      db.execSQL("DELETE FROM student WHERE rno='"+rno.getText()+"'");
      Toast.makeText(getApplicationContext(),"record
deleted",Toast.LENGTH_LONG).show();
    }
    else
    {
      Toast.makeText(getApplicationContext(),"invalid roll
number",Toast.LENGTH_LONG).show();
    }
    clearText();

  }

  public void viewf(View v)
  {

    Cursor c=db.rawQuery("SELECT * FROM student WHERE
rno='"+rno.getText()+"'", null);
    if(c.moveToFirst())
    {
      name.setText(c.getString(0));
```

```
     }
     else

     {
       Toast.makeText(getApplicationContext(),"No such
 record",Toast.LENGTH_LONG).show();

       clearText();
     }
   public void updatef(View v)

   {

     Cursor c=db.rawQuery("SELECT * FROM student WHERE
 rno='"+rno.getText()+"'", null);



     if(c.moveToFirst()) {
       db.execSQL("UPDATE student SET name='" + name.getText() + "' WHERE
 rno='" + rno.getText() + "'");


       Toast.makeText(getApplicationContext(),"Updation
 success",Toast.LENGTH_LONG).show();


     }
     else {

       Toast.makeText(getApplicationContext(),"No such
 record",Toast.LENGTH_LONG).show();

     }
     clearText();

   }

public void clearText()

   {
     rno.setText("");

     name.setText("");
     name.requestFocus();
     }

}



 }
```

OUTPUT:



RESULT:

Thus, The Android Application has been developed and run successfully.

Ex No: 11 DISPLAY A SATELLITE VIEW IN GOOGLE MAP

DATE:

AIM:

To display a Satellite View in Google map using Android Application.

ALGORITHM:

Step 1: Start the Android Studio.

Step 2: File -> New -> New Project ->Application Name ->Next->Finish.
Step 3: Add the following components: Linear layout, Mapview element.

Step 4: In MainActivity.java, give the source code for Satellite view.

Step 5: Run the program.

CODE: XML

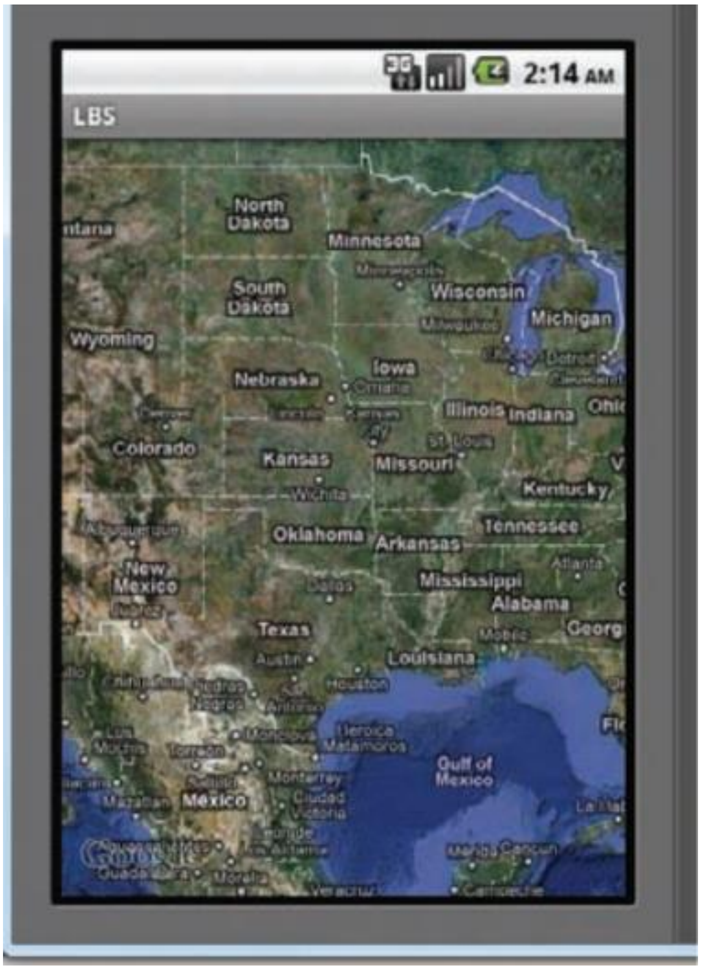
```
<?xml version=”1.0” encoding=”utf-8”?>
<LinearLayout xmlns:android=”http://schemas.android.com/apk/res/android”
android:orientation=”vertical”
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”>
<com.google.android.maps.MapView
android:id=”@+id/mapView”
android:layout_width=”fill_parent”
android:layout_height=”fill_parent”
android:enabled=”true”
android:clickable=”true”
android:apiKey=”<YOUR KEY>” />
</LinearLayout>
```

CODE : MainActivity.java

```
@Override
public void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
mapView = (MapView) findViewById(R.id.mapView);
mapView.setBuiltInZoomControls(true);
mapView.setSatellite(true);
}
```

OUTPUT:



RESULT:

Thus, The Android Application has been developed and run successfully.

Ex No: 12 DISPLAY A STREET VIEW IN GOOGLE MAP

DATE:

AIM:

To display a Street View in Google map using Android Application.

ALGORITHM:

Step 1: Start the Android Studio.

Step 2: File -> New -> New Project ->Application Name ->Next->Finish.
Step 3: Add the following components: Linear layout, Mapview element.

Step 4: In MainActivity.java, give the source code for Street view.

Step 5: Run the program.

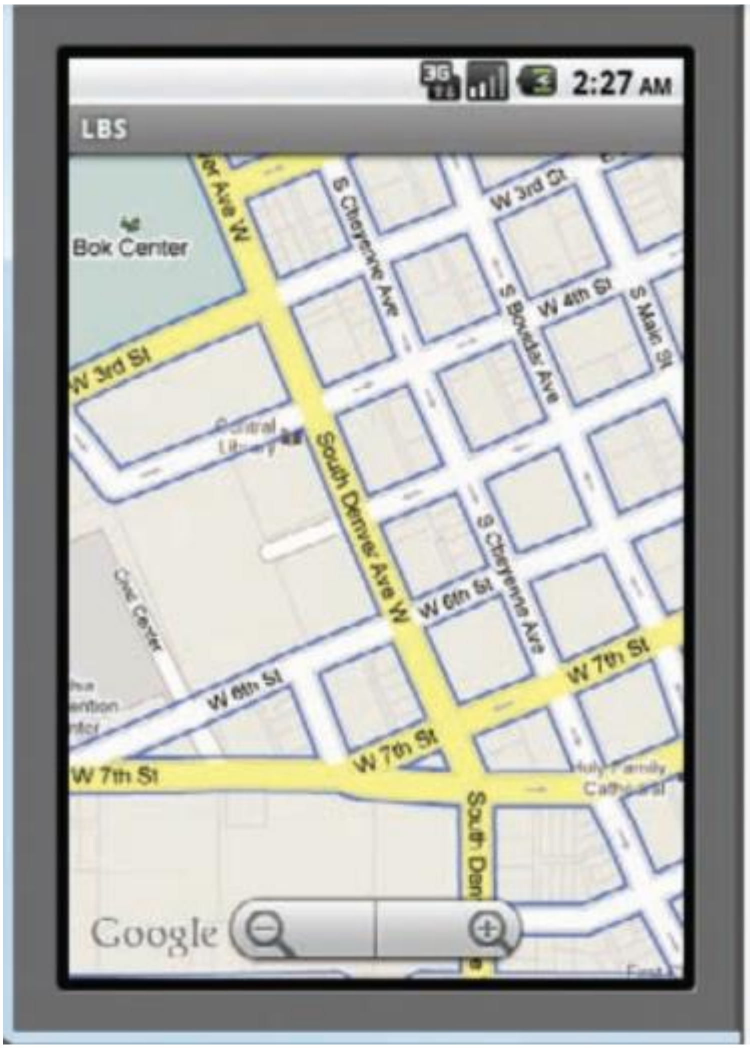
CODE: XML

```
<?xml version=”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”>

<com.google.android.maps.MapView

android:id=”@+id/mapView”

android:layout_width=”fill_parent”

android:layout_height=”fill_parent”

android:enabled=”true”

android:clickable=”true”

android:apiKey=”<YOUR KEY>” />

</LinearLayout>
```

CODE: MainActivity.java

```
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
mapView = (MapView) findViewById(R.id.mapView);
mapView.setBuiltInZoomControls(true);
mapView.setSatellite(true);
mapView.setStreetView(true);
}
```

OUTPUT



RESULT:

Thus, The Android Application has been developed and run successfully.

Ex No: 13 DISPLAY A TRAFFIC CONDITION VIEW IN GOOGLE MAP

DATE:

AIM:

To display a Traffic View in Google map using Android Application.

ALGORITHM:

Step 1: Start the Android Studio.

Step 2: File -> New -> New Project ->Application Name ->Next->Finish.
Step 3: Add the following components: Linear layout, Mapview element.

Step 4: In MainActivity.java, give the source code for traffic view.

Step 5: Run the program.

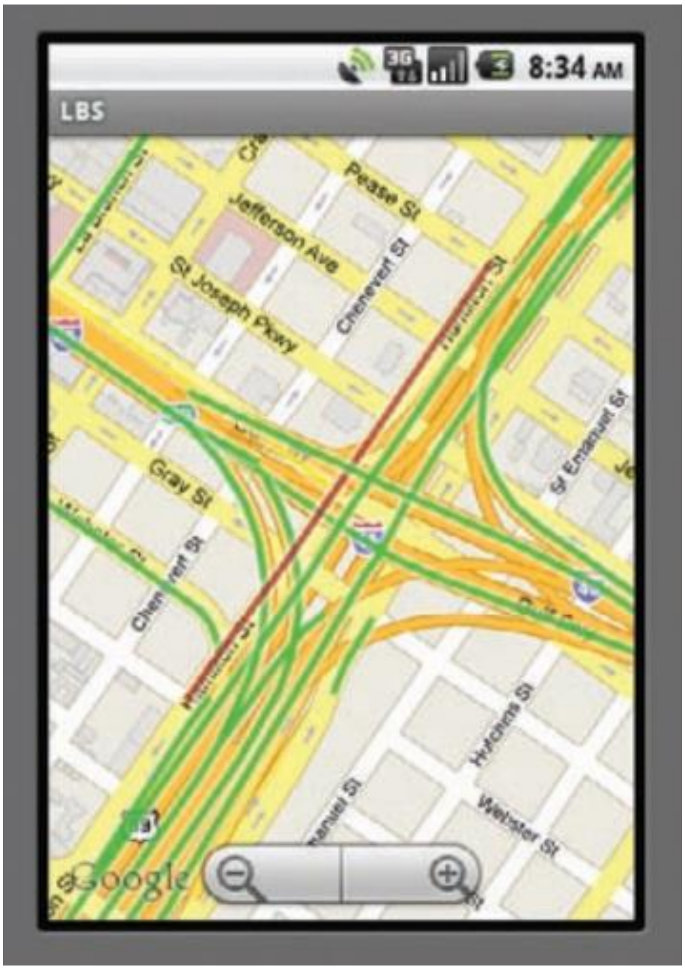
CODE: MainActivity.java

```
package net.learn2develop.LBS;

import android.app.Activity;

import android.os.Bundle;

import android.view.KeyEvent;

import com.google.android.maps.MapActivity;

import com.google.android.maps.MapController;

import com.google.android.maps.MapView;

import com.google.android.maps.GeoPoint;


public class MainActivity extends MapActivity {

MapView mapView;

MapController mc;


GeoPoint p;

/** Called when the activity is first created. */

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);


mapView = (MapView) findViewById(R.id.mapView);

mapView.setBuiltInZoomControls(true);

//mapView.setSatellite(true);
```

```
mapView.setStreetView(true);

mc = mapView.getController();

String coordinates[] = {“1.352566007”, “103.78921587”};

double lat = Double.parseDouble(coordinates[0]);

double lng = Double.parseDouble(coordinates[1]);

p = new GeoPoint(

(int) (lat * 1E6),

(int) (lng * 1E6));

mc.animateTo(p);

mc.setZoom(13);

mapView.invalidate();

}

public boolean onKeyDown(int keyCode, KeyEvent event)

{

 MapController mc = mapView.getController();

switch (keyCode)

{

case KeyEvent.KEYCODE_3:

mc.zoomIn();

break;

case KeyEvent.KEYCODE_1:

mc.zoomOut();

break;

}

return super.onKeyDown(keyCode, event);
```

```
}

@Override

protected boolean isRouteDisplayed() {

// TODO Auto-generated method stub

return false;

} }
```

OUTPUT



RESULT:

Thus, The Android Application has been developed and run successfully.