PRACTICAL: 8

AIM:

Create an application with the help of fragment.

THEORY:

Fragments:

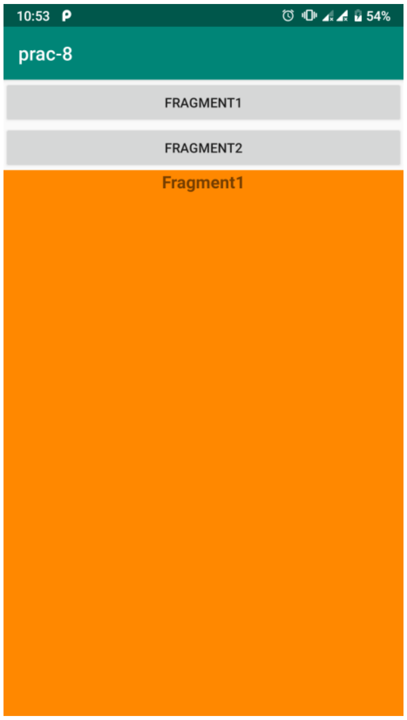
- A fragment has its own layout and its own behaviour with its own life cycle callbacks.
- You can add or remove fragments in an activity while the activity is running.
- You can combine multiple fragments in a single activity to build a multi-pane UI.
- A fragment can be used in multiple activities.
- Fragment life cycle is closely related to the life cycle of its host activity which means when the activity is paused, all the fragments available in the activity will also be stopped.
- A fragment can implement a behaviour that has no user interface component.
- Fragments were added to the Android API in Honeycomb version of Android which API version 11.

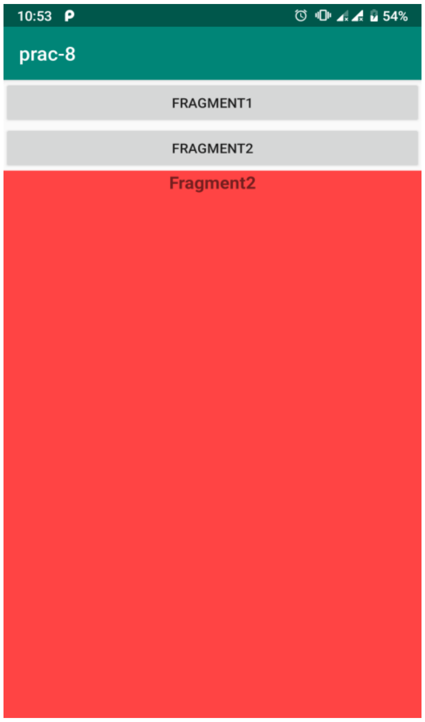
CODE:

```
//activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="com.example.administrator.prac_8.MainActivity">

    <Button
        android:id="@+id/fragment1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="changeFragment"
        android:text="fragment1" />

    <Button
        android:id="@+id/fragment2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="changeFragment"
        android:text="fragment2" />

    <fragment
        android:id="@+id/fragment_view"
        android:name="com.example.administrator.prac_8.Fragment1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"></fragment>
</LinearLayout>
```

```
//fragment_fragment1.xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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=".Fragment1">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@android:color/holo_orange_dark"
        android:text="Fragment1"
        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold" />

</FrameLayout>


//fragment_fragment2.xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout 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=".Fragment2">

    <!-- TODO: Update blank fragment layout -->
    <TextView
        android:id="@+id/p"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@android:color/holo_red_light"
        android:text="Fragment2"
        android:textAlignment="center"
        android:textSize="18sp"
        android:textStyle="bold" />

</FrameLayout>


//MainActivity.java
package com.example.administrator.prac_8;

import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class MainActivity extends AppCompatActivity {

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

    }

    public void changeFragment(View view){
        Fragment fragment;

        if( view == findViewById(R.id.fragment1)){
            fragment = new Fragment1();
            FragmentManager fm = getSupportFragmentManager();
            FragmentTransaction ft = fm.beginTransaction();
            ft.replace(R.id.fragment_view, fragment);
            ft.commit();
        }

        if( view == findViewById(R.id.fragment2)){
            fragment = new Fragment2();
            FragmentManager fm = getSupportFragmentManager();
            FragmentTransaction ft = fm.beginTransaction();
            ft.replace(R.id.fragment_view, fragment);
            ft.commit();
        }
    }
}

//Fragment1.java
package com.example.administrator.prac_8;

import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment1 extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_fragment1, container, false);
    }
}


//Fragement2.java
package com.example.administrator.prac_8;

import android.content.Context;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

public class Fragment2 extends Fragment {

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate(R.layout.fragment_fragment2, container, false);
    }
}
```

OUTPUT:





CONCLUSION: We have successfully created android application which calls different fragment on click of associated button.