**Assignment No: 07**

**Aim:-** Create an android app that demonstrate Screen Navigation Using the App Bar and Tabs.

**Code:-**

## Basics

Typically, an implementation of tabs in Android consists of:

1. Swipe views
2. Tabs UI element

These are two independent navigation patterns, but they can be combined with each other.

In general: swipe views can be combined with tabs, and tabs be combined with swipe views. However, tabs benefit tremendously from being combined with swipe views, as explained below.

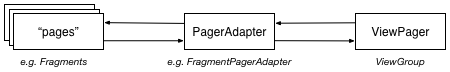
## Swipe Views

Swipe views allow to flip through a set of "pages" by swiping horizontally on the screen.

Combined with tabs, this allows the user to switch to the next or previous tab by just swiping anywhere on the screen, rather than having to click on the tab itself.

* Swipe views are implemented by the [ViewPager](https://developer.android.com/reference/android/support/v4/view/ViewPager.html) ViewGroup (declared in the activity's layout XML)
* A page is typically implemented as a [Fragment](https://developer.android.com/reference/android/support/v4/view/ViewPager.html)
* A [PagerAdapter](https://developer.android.com/reference/android/support/v4/view/PagerAdapter.html) supplies the ViewPager with the pages (fragments) to display. In the case of using fragments as pages, this PagerAdapter is a [FragmentPagerAdapter](https://developer.android.com/reference/android/support/v13/app/FragmentPagerAdapter.html) or [FragmentStatePagerAdapter](https://developer.android.com/reference/android/support/v13/app/FragmentStatePagerAdapter.html)

The swipe views architecture is illustrated below:

[](https://github.com/weibeld/AndroidTabsExample/blob/master/README_res/swipe_views.png)

**TabLayout Tabs**

This is the preferred approach as it's easier to implement than ActionBar tabs and does not rely on an ActionBar.

TabLayout tabs are especially easy to implement if they are used in combination with a formerly implemented ViewPager (see [Swipe Views](https://github.com/weibeld/AndroidTabsExample#swipe-views)), because then the tabs can be automatically populated by the ViewPager (i.e. no need to create and add the individual tabs manually). This is shown in the following:

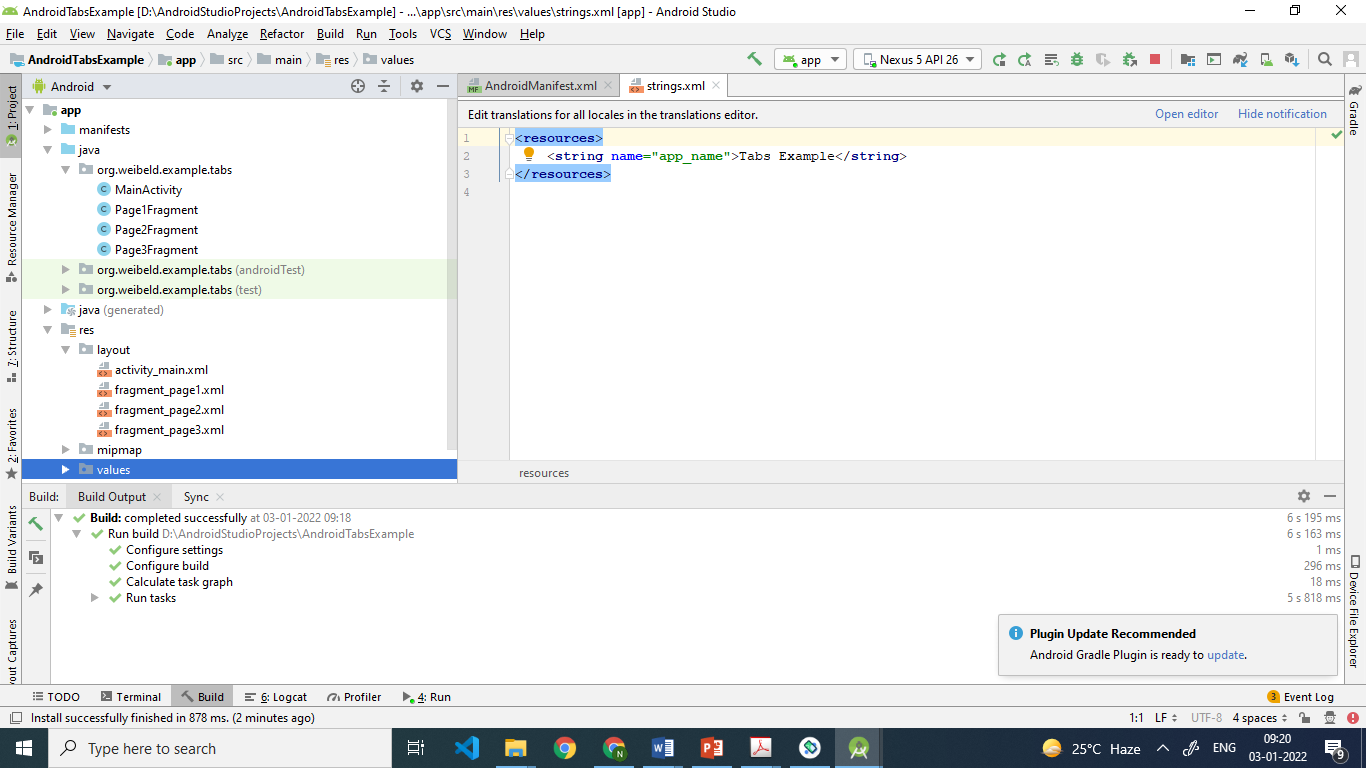
1. In the activity's layout XML, add a [TabLayout](https://developer.android.com/reference/android/support/design/widget/TabLayout.html) element above the ViewPager element (and below the Toolbar element, if a Toolbar is used)
2. In the activity's onCreate method, call [setupWithViewPager(ViewPager)](https://developer.android.com/reference/android/support/design/widget/TabLayout.html" \l "setupWithViewPager(android.support.v4.view.ViewPager)) on the TabLayout to populate and integrate the TabLayout with the ViewPager (requires that [getPageTitle](https://developer.android.com/reference/android/support/v4/view/PagerAdapter.html" \l "getPageTitle(int)) of the PagerAdapter is overriden)

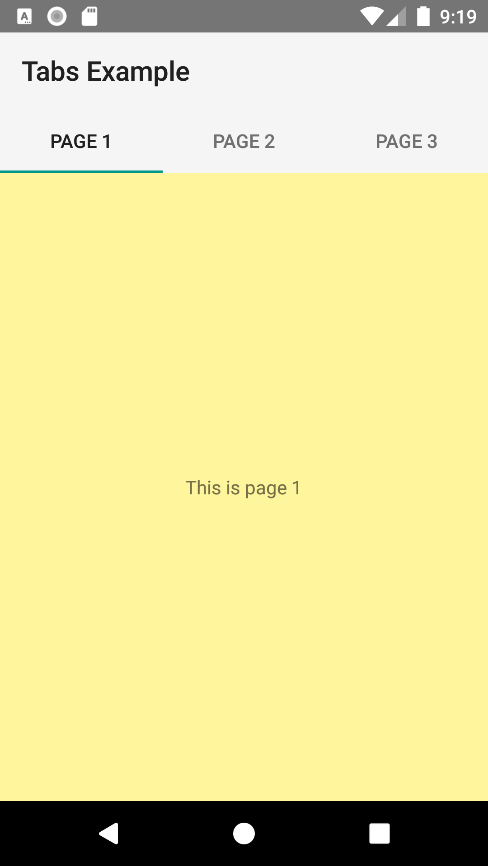
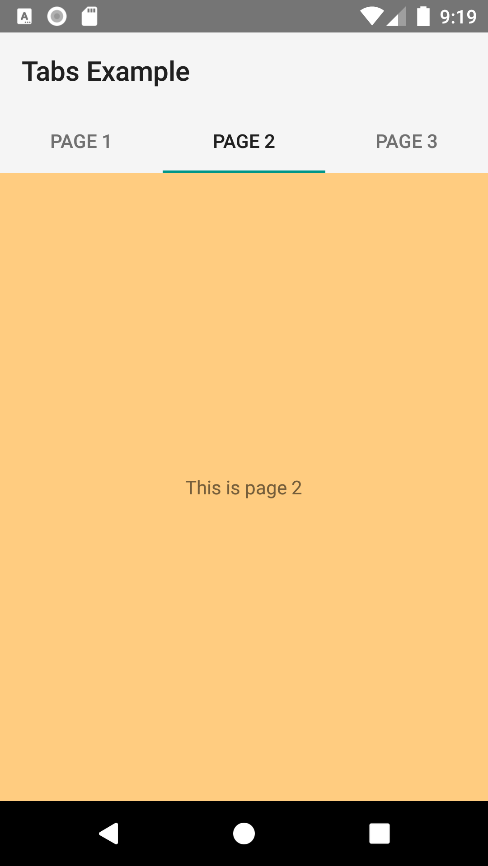
**Note:** to use TabLayout, the *Design Support Library* must be added to the project. In the module build.gradle file, do:

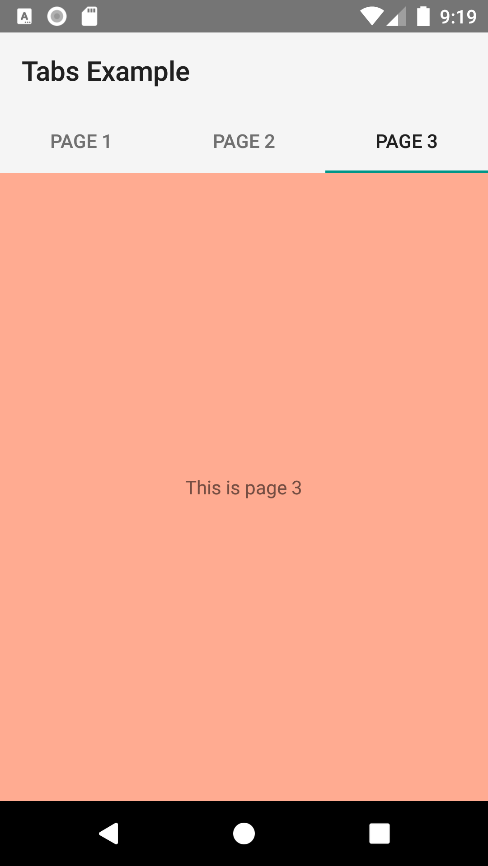
dependencies {

compile 'com.android.support:support-v13:24.2.1'

}





**activity\_main.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:id="@+id/activity\_main"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context="org.weibeld.example.tabs.MainActivity"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:minHeight="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"** />  
  
 <**android.support.design.widget.TabLayout  
 android:id="@+id/tab\_layout"  
 android:layout\_height="wrap\_content"  
 android:layout\_width="match\_parent"  
 android:background="?attr/colorPrimary"** />  
  
 <**android.support.v4.view.ViewPager  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/viewpager"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />  
  
</**LinearLayout**>

**fragment\_page1.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/pageColor1"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="This is page 1"** />  
  
</**FrameLayout**>

**fragment\_page2.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/pageColor2"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="This is page 2"** />  
  
</**FrameLayout**>

**fragment\_page3.xml**

*<?***xml version="1.0" encoding="utf-8"***?>*<**FrameLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@color/pageColor3"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="This is page 3"** />  
  
</**FrameLayout**>

**Page1Fragment.java**

**package** org.weibeld.example.tabs;  
  
**import** android.app.Fragment;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
  
**import** org.weibeld.example.R;  
  
*/\* Fragment used as page 1 \*/***public class** Page1Fragment **extends** Fragment {  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View rootView = inflater.inflate(R.layout.***fragment\_page1***, container, **false**);  
 **return** rootView;  
 }  
}

**Page2Fragment.java**

**package** org.weibeld.example.tabs;  
  
**import** android.app.Fragment;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
  
**import** org.weibeld.example.R;  
  
*/\* Fragment used as page 2 \*/***public class** Page2Fragment **extends** Fragment {  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View rootView = inflater.inflate(R.layout.***fragment\_page2***, container, **false**);  
 **return** rootView;  
 }  
  
}

**Page3Fragment.java**

**package** org.weibeld.example.tabs;  
  
**import** android.app.Fragment;  
**import** android.os.Bundle;  
**import** android.view.LayoutInflater;  
**import** android.view.View;  
**import** android.view.ViewGroup;  
  
**import** org.weibeld.example.R;  
  
*/\* Fragment used as page 3 \*/***public class** Page3Fragment **extends** Fragment {  
  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
 View rootView = inflater.inflate(R.layout.***fragment\_page3***, container, **false**);  
 **return** rootView;  
 }  
}

**MainActivity.java**

**package** org.weibeld.example.tabs;  
  
**import** android.app.Fragment;  
**import** android.app.FragmentManager;  
**import** android.os.Bundle;  
**import** android.support.design.widget.TabLayout;  
**import** android.support.v13.app.FragmentPagerAdapter;  
**import** android.support.v4.view.ViewPager;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
  
**import** org.weibeld.example.R;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private final** String **LOG\_TAG** = MainActivity.**class**.getSimpleName();  
  
 *// Titles of the individual pages (displayed in tabs)* **private final** String[] **PAGE\_TITLES** = **new** String[] {  
 **"Page 1"**,  
 **"Page 2"**,  
 **"Page 3"** };  
  
 *// The fragments that are used as the individual pages* **private final** Fragment[] **PAGES** = **new** Fragment[] {  
 **new** Page1Fragment(),  
 **new** Page2Fragment(),  
 **new** Page3Fragment()  
 };  
  
 *// The ViewPager is responsible for sliding pages (fragments) in and out upon user input* **private** ViewPager **mViewPager**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 *// Set the Toolbar as the activity's app bar (instead of the default ActionBar)* Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 *// Connect the ViewPager to our custom PagerAdapter. The PagerAdapter supplies the pages  
 // (fragments) to the ViewPager, which the ViewPager needs to display.* **mViewPager** = (ViewPager) findViewById(R.id.***viewpager***);  
 **mViewPager**.setAdapter(**new** MyPagerAdapter(getFragmentManager()));  
  
 *// Connect the tabs with the ViewPager (the setupWithViewPager method does this for us in  
 // both directions, i.e. when a new tab is selected, the ViewPager switches to this page,  
 // and when the ViewPager switches to a new page, the corresponding tab is selected)* TabLayout tabLayout = (TabLayout) findViewById(R.id.***tab\_layout***);  
 tabLayout.setupWithViewPager(**mViewPager**);  
 }  
  
  
 */\* PagerAdapter for supplying the ViewPager with the pages (fragments) to display. \*/* **public class** MyPagerAdapter **extends** FragmentPagerAdapter {  
  
 **public** MyPagerAdapter(FragmentManager fragmentManager) {  
 **super**(fragmentManager);  
 }  
  
 @Override  
 **public** Fragment getItem(**int** position) {  
 **return PAGES**[position];  
 }  
  
 @Override  
 **public int** getCount() {  
 **return PAGES**.**length**;  
 }  
  
 @Override  
 **public** CharSequence getPageTitle(**int** position) {  
 **return PAGE\_TITLES**[position];  
 }  
  
 }  
}