

```

<android.support.v7.widget.AppCompatImageView
    android:layout_width="match_parent"
    android:layout_height="200dp"
    app:srcCompat="@drawable/rahiyan"/>
<Button
    android:id="@+id/btnHome"
    android:layout_width="match_parent"
    android:layout_height="72dp"
    android:text="@string/home"
    android:paddingRight="20dp"
    android:gravity="right|center_vertical"
    android:textSize="20sp"
    android:theme="@style/MyButton"
    style="@style/Widget.AppCompat.Button.Borderless"/>
</LinearLayout>

```

آموزش استفاده از SnackBar

در ابتدا باید کتابخانه design رو به gradle اضافه کنیم :

```
compile 'com.android.support:design:23.3.0'
```

حالا باید به تغییر توی لایوت ها بدیم و به جای LinearLayout و RelativeLayout از CoordinatorLayout استفاده کنیم.

حالا جایی که نیاز داریم SnackBar نمایش داده بشه به شکل زیر عمل می کنیم :

```
Snackbar snackbar = Snackbar.make(coordinatorLayout,
    "Welcome ", Snackbar.LENGTH_LONG );
```

```
snackbar.show();
```

اضافه کردن دکمه اکشن در SnackBar :

```
Snackbar snackbar = Snackbar.make(coordinatorLayout, "Connect
Faild !", Snackbar.LENGTH_LONG).setAction("RETRY", new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
```

آموزش استفاده از NavigationDrawer

در ابتدا کتابخانه design رو به gradle اضافه می کنیم :

```
compile 'com.android.support:design:23.3.0'
```

سپس لایوت مورد نظرمون رو باز می کنیم و کد های زیر رو قرار میدیم :

```

<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:clickable="true"
    android:focusableInTouchMode="true">

```

```

<include layout="@layout/content"
    android:id="@+id/content">

```

```
</include>
```

```

<android.support.design.widget.NavigationView
    android:id="@+id/nvView"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="end"
    android:background="@android:color/white"
    app:headerLayout="@layout/nav"

```

```
/>
```

```
</android.support.v4.widget.DrawerLayout>
```

حالا باید لایوت nav رو بسازیم :

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

```

```

FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab_main);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Click action
    }
});

```

آموزش استفاده از Floating Action Menu (FAM) :

در ابتدا کتابخانه design و کتابخانه زیر رو به gradle اضافه می کنیم :

```
compile 'com.github.clans:fab:1.6.2'
```

حالا باید از CoordinatorLayout به جای لایوت های دیگر استفاده کنیم. در ادامه یک FAM

میسازیم و بعدش ۳ تا FAB کوچک میسازیم.

```

<com.github.clans.fab.FloatingActionMenu
    android:id="@+id/floating_action_menu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_marginBottom="11dp"
    android:layout_marginLeft="11dp"
    android:layout_marginRight="11dp"
    fab:menu_animationDelayPerItem="55"
    fab:menu_backgroundColor="@android:color/transparent"
    fab:menu_buttonSpacing="0dp"
    fab:menu_colorNormal="#da3c2f"
    fab:menu_colorPressed="#dc4b3f"
    fab:menu_colorRipple="#99d4d4d4"
    fab:menu_fab_label="Floating Action Menu"
    fab:menu_fab_size="normal"
    fab:menu_icon="@drawable/fab_add"
    fab:menu_labels_colorNormal="#3333"

```

```

    }
});

```

```
snackbar.show();
```

(ص ۲) سفارشی سازی Snackbar :

عوض کردن رنگ دکمه اکشن :

```
snackbar.setActionTextColor(Color.RED);
```

عوض کردن رنگ پیغام :

```

View sbView = snackbar.getView();
TextView textView = (TextView)
sbView.findViewById(android.support.design.R.id.snackbar_text
);
textView.setTextColor(Color.YELLOW);
snackbar.show();

```

آموزش استفاده از Floating Action Button (FAB) :

در ابتدا باید کتابخانه design رو به gradle اضافه کنیم. بعد در لایوتی که میخواهیم از FAB استفاده کنیم LinearLayout و RelativeLayout را به CoordinatorLayout تغییر میدیم.

در فایل dims.xml کد زیر رو قرار میدیم:

```
<dimen name="fab_margin">16dp</dimen>
```

حالا می تونیم FAB رو به شکل زیر استفاده کنیم :

```

<android.support.design.widget.FloatingActionButton
    android:id="@+id/fab_main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:fabSize="normal"
    app:elevation="10dp"
    android:layout_gravity="bottom|end"
    android:src="@drawable/phone"
    android:layout_margin="@dimen/fab_margin"/>

```

برای اینکه از رویداد کلیک روی FAB استفاده کنیم در اکتیوییتی مربوط به لایوت بالا از کد های زیر

استفاده می کنیم :

```

        android:layout_height="wrap_content"
        android:hint="@string/hint_email" />
</android.support.design.widget.TextInputLayout>

```

TextInputLayout دارای ۲ متد setError() و setErrorEnabled() هست که به شکل زیر از آن ها استفاده می کنیم :

```

private void isValid(){
    if (edtFirst.getText().toString().trim().length() == 0){
        tilFirst.setError("NOT CHAR !!");
        tilFirst.setErrorEnabled(true);
    }
    else{
        tilFirst.setErrorEnabled(false);
    }
}

```

آموزش استفاده از Toolbar :

ابتدا کتابخانه gradle رو به اضافه می کنیم. سپس می توی فایل styles.xml و تم رو به NoActionBar تغییر میدیم. حالا یه لایوت جدید میسازیم به نام tool_bar.xml که توش تولبارمون رو قرار میدیم :

```

<android.support.v7.widget.Toolbar
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@color/colorPrimary"
    android:theme="@style/Toolbar.Theme"
    android:gravity="right"
    app:layout_scrollFlags="scroll|enterAlways">
</android.support.v7.widget.Toolbar>

```

حالا میریم توی لایوتی که میخوایم تولبار داشته باشه و لایوت اون رو به CoordinatorLayout تغییر میدیم. به شکل زیر از تولبار استفاده می کنیم :

```

fab:menu_labels_colorPressed="#444"
fab:menu_labels_colorRipple="#66efec"
fab:menu_labels_cornerRadius="3dp"
fab:menu_labels_ellipsize="none"
fab:menu_labels_hideAnimation="@anim/out_to_right"
fab:menu_labels_margin="0dp"
fab:menu_labels_maxLines="-1"
fab:menu_labels_padding="8dp"
fab:menu_labels_position="left"
fab:menu_labels_showAnimation="@anim/in_from_right"
fab:menu_labels_showShadow="true"
fab:menu_labels_singleLine="false"
fab:menu_labels_textColor="#f2f1f1"
fab:menu_labels_textSize="15sp"
fab:menu_openDirection="up"
fab:menu_shadowColor="#66aff198"
fab:menu_shadowRadius="4dp"
fab:menu_shadowXOffset="1dp"
fab:menu_shadowYOffset="4dp"
fab:menu_showShadow="true">
<FAB .... />
<FAB .... />
<FAB .... />

```

```
</com.github.clans.fab.FloatingActionMenu>
```

(۳ ص)

آموزش استفاده از Floating Label :

ابتدا کتابخانه design رو به اضافه می کنیم. سپس لایوت های دیگر رو به CoordinatorLayout تبدیل می کنیم. به شکل زیر آن را اضافه می کنیم :

```

<android.support.design.widget.TextInputLayout
    android:id="@+id/input_layout_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/input_password"
        android:layout_width="match_parent"

```

```

tb = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(tb);
tb.setOnMenuItemClickListener(new
Toolbar.OnMenuItemClickListener() {
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
});

```

برای اینکه بتوانیم دکمه Back رو به تولبار اضافه کنیم از کد زیر استفاده می کنیم: <- خانه
 getSupportActionBar().**setDisplayHomeAsUpEnabled(false)**;
 کنار onCreate() اکتیوییتی یک متد دیگر را override می کنیم :

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.m, menu);
    return true;
}

```

۴ تا امکان از پیش تعیین شده در تولبار داریم که به شکل های زیر از آن ها استفاده می کنیم :
 getSupportActionBar().**setSubtitle("hi")**;
 getSupportActionBar().**setTitle("HI")**;
 getSupportActionBar().**setLogo(R.drawable.map)**;
 tb.setNavigationIcon(R.drawable.menu);

آموزش استفاده از Fragment – ViewPager – TabLayout

ابتدا کتابخانه design رو به gradle اضافه می کنیم. حالا باید چند Fragment بسازیم که با راست کلیک و انتخاب (New --> Fragment(Blank)) امکان پذیر هست. حالا در فایل `dimens.xml` مقادیر زیر را وارد می کنیم :

```

<dimen name="tab_max_width">264dp</dimen>
<dimen name="tab_padding_bottom">16dp</dimen>
<dimen name="tab_label">14sp</dimen>

```

در نهایت لایوت main رو باز می کنیم و لایوت آن را به CoordinatorLayout تغییر میدیم. حالا TabLayout و ViewPager رو به شکل زیر اضافه می کنیم :

```

<android.support.design.widget.CoordinatorLayout
    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:id="@+id/coordLmain"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="ir.ucgain.app.practice.MainActivity">

```

```

<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

```

```

    <include layout="@layout/toolbar"
        android:id="@+id/toolB">
    </include>

```

```

</android.support.design.widget.AppBarLayout>

```

```

</android.support.design.widget.CoordinatorLayout>

```

(ص ۴) برای اینکه بتوانیم آیکن به تولبارمون اضافه کنیم توی دایرکتوری `res` به پوشه جدید به اسم `menu` میسازیم و توش یه فایل بنام `m.xml` میسازیم و کد های زیر رو قرار میدیم :

```

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

```

```

    <item
        android:id="@+id/action_favorite"
        android:icon="@drawable/menu"
        android:title="منو"
        app:showAsAction="ifRoom"/>

```

```

</menu>

```

حالا فایل اکتیوییتی مربوطه رو باز می کنیم و به شکل زیر عمل می کنیم :

```

@Override
public Fragment getItem(int position) {
    return mFragmentManager.get(position);
}
@Override
public int getCount() {
    return mFragmentManager.size();
}

public void addFragment(Fragment fragment, String title)
{
    mFragmentManager.add(fragment);
    mFragmentManager.add(title);
}
@Override
public CharSequence getPageTitle(int position) {
    return mFragmentManager.get(position);
}
}

```

حالا اکتیویتی که در لایوتش و ViewPager و TabLayout رو اضافه کردیم رو باز می کنیم و این چنین می نویسیم :

```

private TabLayout tabLayout;
private ViewPager viewPager;

viewPager = (ViewPager) findViewById(R.id.viewpager);
tabLayout = (TabLayout) findViewById(R.id.tabs);

setupViewPager(viewPager);
tabLayout.setupWithViewPager(viewPager);
    تابع setupViewPager به شکل زیر پیاده سازی میشه :
private void setupViewPager(ViewPager viewPager) {
    ViewPagerAdapter adapter = new
ViewPagerAdapter(getSupportFragmentManager());
    adapter.addFragment(new FragmentA(), "A");
    adapter.addFragment(new FragmentB(), "B");
    adapter.addFragment(new FragmentC(), "C");
    viewPager.setAdapter(adapter);
}

```

```

<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <include layout="@layout/toolbar"
        android:id="@+id/toolB">

    </include>

<android.support.design.widget.TabLayout
    android:id="@+id/tabs"
    android:layout_below="@+id/toolB"
    android:layout_width="match_parent"
    android:layout_height="96dp"
    app:tabMode="fixed"// scrollable
    app:tabGravity="fill" // center
    app:tabTextAppearance="@style/MyTabLayoutTextAppearance"
    app:tabTextColor="@color/colorTabText"
    app:tabSelectedTextColor="@color/colorTabTextSelected"
    app:tabIndicatorHeight="5dp"
    />
</android.support.design.widget.AppBarLayout>

<android.support.v4.view.ViewPager
    android:id="@+id/viewpager"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    />

    class ViewPagerAdapter extends FragmentPagerAdapter {
        private final List<Fragment> mFragmentManager = new
ArrayList<>();
        private final List<String> mFragmentManagerTitleList = new
ArrayList<>();

        public ViewPagerAdapter(FragmentManager manager) {
            super(manager);
        }
    }

```

(ص ۵) حالا باید کلاس ViewPagerAdapter رو بسازیم :

حالا اگه بخواهیم که فقط آیکون برای تب نمایش داده بشه و متن نباشه تابع `getPageTitle()` در کلاس `ViewPagerAdapter` رو باید به شکل زیر بنویسیم :

```
@Override
public CharSequence getPageTitle(int position) {
    return null;
}
```

آموزش استفاده از `TabLayout + Toolbar + NavigationDraw` از `ViewPager`

نکته مهم در اینجا نوع چینش این View هاست :

```
<android.support.v4.widget.DrawerLayout >
    <android.support.design.widget.CoordinatorLayout>
        <android.support.design.widget.AppBarLayout>
            <include layout="@layout/toolbar">
            </include>
            <android.support.design.widget.TabLayout/>
        </android.support.design.widget.AppBarLayout>
        <android.support.v4.view.ViewPager/>
        <android.support.design.widget.FloatingActionButton/>
    </android.support.design.widget.CoordinatorLayout>
</android.support.design.widget.NavigationView/>
</android.support.v4.widget.DrawerLayout>
```

آموزش قراردادن FAB بین ۲ لایوت :

با ویژگی `app:layout_anchor` تعیین می کنیم که زیر کدام View این FAB ما قرار بگیره و با ویژگی `app:layout_anchorGravity` تعیین می کنیم FAB در سمت راست باشه یا چپ یا وسط.:

```
app:layout_anchor="@id/viewA"
app:layout_anchorGravity="bottom|right|end"
```

(ص ۶) حالا فرض کنیم که یک FAB داریم و توی فرگمنت اول نمایش داده بشه و توی بقیه دیده نشه . برای پیاده سازی این قضیه از کد زیر استفاده می کنیم :

```
viewPager.addOnPageChangeListener(new
ViewPager.OnPageChangeListener() {

    @Override
    public void onPageScrolled(int position, float
positionOffset, int positionOffsetPixels) {

    }

    @Override
    public void onPageSelected(int position) {
        switch (position) {
            case 0:
                fab.show();
                break;
            case 1:
                fab.hide();
                break;
            case 3:
                fab.hide();
                break;
        }
    }

    @Override
    public void onPageScrollStateChanged(int state) {

    }
});

    setupTabIcons();

private void setupTabIcons() {
    tabLayout.getTabAt(0).setIcon(R.drawable.ic_tab0);
    tabLayout.getTabAt(1).setIcon(R.drawable.ic_tab1);
    tabLayout.getTabAt(2).setIcon(R.drawable.ic_tab3);
}
```