MTN App Academy Module 2

Creating a custom theme on Android

Step 1: Add our attribute values

Create a file in the resource values folder named "attrs" res/values/attrs.xml and add the following code to define the application attributes

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <attr name="pageBackground" format="reference" />
    <attr name="textLarge" format="reference" />
    <attr name="textSubheader" format="reference" />
    <attr name="textRegular" format="reference" />
    <attr name="whiteBackground" format="reference" />
    <attr name="button" format="reference" />
    <attr name="whiteSmoke" format="color" />
</resources>
```

Step 2: Add dimension attributes and values

Create a file in the resource values folder named "dimens" res/values/dimens.xml and add the following code to define the dimension constants.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="padding_horizontal">16dp</dimen>
    <dimen name="margin_left">8sp</dimen>
    <dimen name="margin_top">40dp</dimen>
    <dimen name="text_size_large">24sp</dimen>
    <dimen name="text_size_subheader">16sp</dimen>
    <dimen name="text_size_regular">15sp</dimen>
    <dimen name="margin">20dp</dimen>
    <dimen name="padding">20dp</dimen>
    <dimen name="margin_top_small">10dp</dimen>
    <dimen name="button_radius">8dp</dimen>
</resources>
```

Step 3: Add Default Colour Attributes and Values

in the drawable/values/colors file, add the following colors to define the application's custom colors

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="color_primary">#F3CB51</color>
    <color name="color_primary_variant">#F3CB51</color>
    <color name="color_secondary">#5A97CF</color>
    <color name="color_secondary_variant">#2F72B9</color>
    <color name="color_on_primary">#20201e</color>
    <color name="color_on_secondary">#303031</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="whiteSmoke">#f7f7f7</color>
</resources>
```

Step 4: Add Page Background Drawable

In the drawable folder, create a file named page_background.xml, drawable/page_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="@color/whiteSmoke"></solid>
</shape>
```

Step 5: Add Button States

Add button states for our application. Define 2 files in the res/, a selected and a default state.

a) Selected State Drawable:

add a file named button_background_selected, drawable/button_background_selected.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="@dimen/button_radius" />
    <solid android:color="?colorSecondaryVariant" ></solid>

</shape>
```

b) Default State Drawable:

add a file named button_background_selected, drawable/button_background_default.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <corners android:radius="@dimen/button_radius" />
    <solid android:color="?colorSecondary" ></solid>

</shape>
```

Step 6: Add Button Background Selector Drawable

Add a file named button_background, drawable/button_background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/button_background_selected" android:state_pressed="true" />
    <item android:drawable="@drawable/button_background_default"></item>
</selector>
```

Step 7: Add theme style that will override default styles

Add a file named styles to the resource values, res/values/styles.xml and add the following. These will be used to override the default set within the application.

a) Page Background:

Add a file named page_background, drawable/page_background.xml

```
<style name="page_background">
        <item name="android:background">@drawable/page_background</item>
        <item name="android:paddingLeft">@dimen/padding_horizontal</item>
        <item name="android:paddingRight">@dimen/padding_horizontal</item>
</style>
```

b) White Background:

Add a file named white_background, drawable/white_background.xml

```
<style name="white_background">
        <item name="android:background">@android:color/white</item>
        <item name="android:layout_marginTop">@dimen/margin_top</item>
        <item name="android:layout_marginBottom">@dimen/margin_top</item>
        <item name="android:padding">@dimen/padding</item>
</style>
```

c) Text Large:

Add a file named text_large, drawable/text_large.xml

```
<style name="text_large">
        <item name="android:textColor">@android:color/background_dark</item>
        <item name="android:textSize">@dimen/text_size_large</item>
```

```
        <item name="android:shadowDy">1.0</item>
        <item name="android:shadowRadius">1</item>
        <item name="android:shadowColor">#888</item>
        <item name="android:textStyle">bold</item>
        <item name="android:layout_marginTop">@dimen/margin_top</item>
 </style>
```

d) Text Sub-header:

Add a file named text_subheader, drawable/text_subheader.xml

```
<style name="text_subheader">
        <item name="android:textColor">#000</item>
        <item name="android:textSize">@dimen/text_size_subheader</item>
        <item name="android:shadowDy">1.0</item>
        <item name="android:shadowRadius">1</item>
        <item name="android:shadowColor">#000</item>
</style>
```

e) Text Regular:

Add a file named text_regular, drawable/text_regular.xml

```
<style name="text_regular">
        <item name="android:textColor">@android:color/background_dark</item>
        <item name="android:textSize">@dimen/text_size_regular</item>
</style>
```

f) Button:

Add a file named button, drawable/button.xml which extends text_large and applies all those styles to our button

```
<style name="button" parent="text_large">
        <item name="android:background">@drawable/button_background</item>
        <item name="android:layout_marginTop">@dimen/margin</item>
        <item name="android:paddingHorizontal">@dimen/padding_horizontal</item>
</style>
```

Step 8: Add Colours to themes.xml

Add the colors that are declared in colors.xml to the application theme. This must be done in res/values/themes/theme.xml

```
<!-- Primary brand color. -->
<item name="colorPrimary">@color/color_primary</item>
<item name="colorPrimaryVariant">@color/color_primary_variant</item>
<item name="colorOnPrimary">@color/white</item>
<!-- Secondary brand color. -->
<item name="colorSecondary">@color/color_secondary</item>
<item name="colorSecondaryVariant">@color/color_secondary_variant</item>
<item name="colorOnSecondary">@color/black</item>
<!-- Status bar color. -->
<item name="android:statusBarColor" tools:targetApi="l">?attr/colorPrimaryVariant</item>
```

Step 9: Set styles to our defined theme attributes

Set all the attributes defined to the individual styles in order to reference them in the application. This must also be done in res/values/themes/theme.xml

```
<item name="android:buttonStyle">@style/button</item>

<item name="pageBackground">@style/page_background</item>
<item name="textSubheader">@style/text_subheader</item>
<item name="textLarge">@style/text_large</item>
<item name="textRegular">@style/text_regular</item>
<item name="whiteBackground">@style/white_background</item>
<item name="button">@style/button</item>
<item name="spinner">@style/spinner</item>
```

Step 10: Use our theme attributes in our layout file

Start using all the attributes declared within the application's layout files.

```
<?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"
    style="?pageBackground"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        style="?textLarge"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="@dimen/margin_top"
        android:text="Accept"
        style="?button"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView"
        app:layout_constraintVertical_bias="0.066" />

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