# Snap2

# apex-round-128.pdf

# **Oracle Application Express: Developing Database Web Applications**

# **Hands-On-Labs Guide**

# **Unit 12: Using Themes and Theme Styles**

This exercise includes two hands-on-labs.

HOL 12-1 uses the Demo Projects application.  
HOL 12-2 uses the Budget App application.

**HOL 12-1 Updating the Team Members Pages**: In this hands-on lab, you create a Classic Report region and update the report template to use Cards. You also specify template options.

**HOL 12-2 Creating and Using Theme Styles**: In this hands-on lab, you use the Budget App application. You use the Theme Roller to save the changes made to the Theme Style and then enable users to apply this Theme Style in the running application.

## **HOL 12-1: Updating the Team Member Pages**

In this lab, you update the Team Member pages by replacing the interactive report with a card region. This provides a directory of your team members' contact information and roles as though you had a stack of their business cards at hand.

1. In the Demo Projects application runtime environment, click **Team Members** in the navigation menu.  
   In the Developer Toolbar, click **Edit Page 2**.
2. Update the interactive report region title from **Proj Team Members** to **Team Members**.
3. This page is an interactive report. You want to convert the report type to a Classic Report type, so that you can utilize one of the built-in templates. Update the interactive report to a Classic Report, and apply the Cards template to display the team members in a more visually pleasing manner.  
   Click the **Team Members** region and input the following:

* Identification: Type - select **Classic Report**
* Source: Type – select **SQL Query**

1. In the SQL Query attribute, click the **Code Editor** button.  
     
   A screenshot of a cell phone

   Description automatically generated
2. In the Code Editor window, delete the existing SQL Query and copy and paste the following code:

**select id**

**, full\_name card\_title**

**, profile card\_text**

**, nvl(email,' ') card\_subtext**

**, dbms\_lob.getlength('PHOTO\_BLOB') card\_initials**

**, apex\_util.prepare\_url('f?p='||:APP\_ID||':3:'||:APP\_SESSION||'::::P3\_ID:'||id) card\_link**

**, photo\_filename**

**, photo\_mimetype**

**, photo\_charset**

**, photo\_last\_updated**

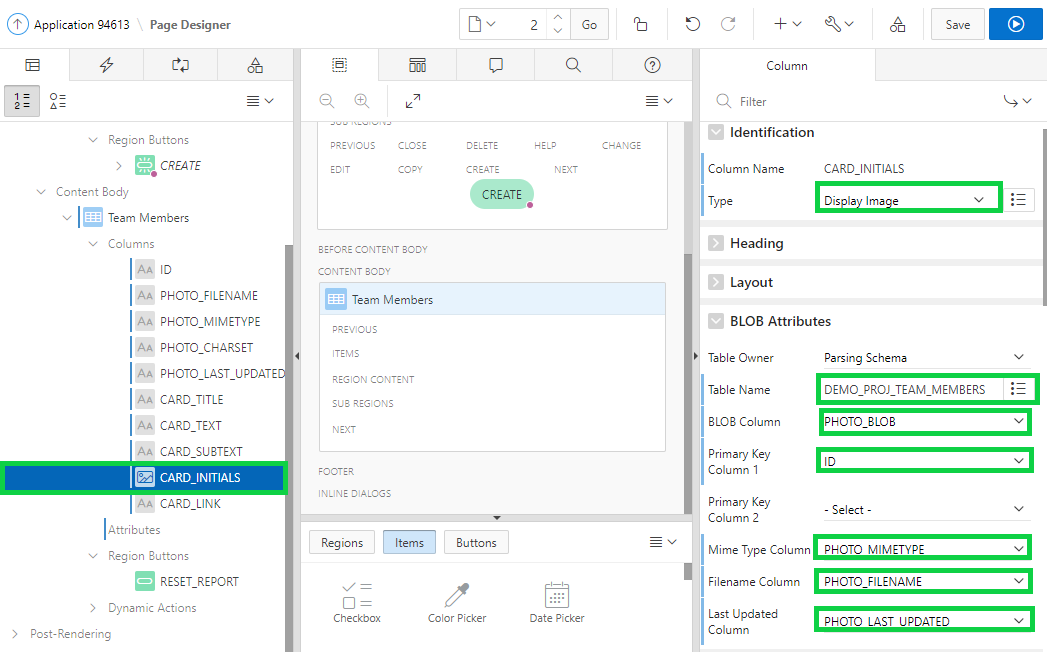
**from demo\_proj\_team\_members  
order by created**

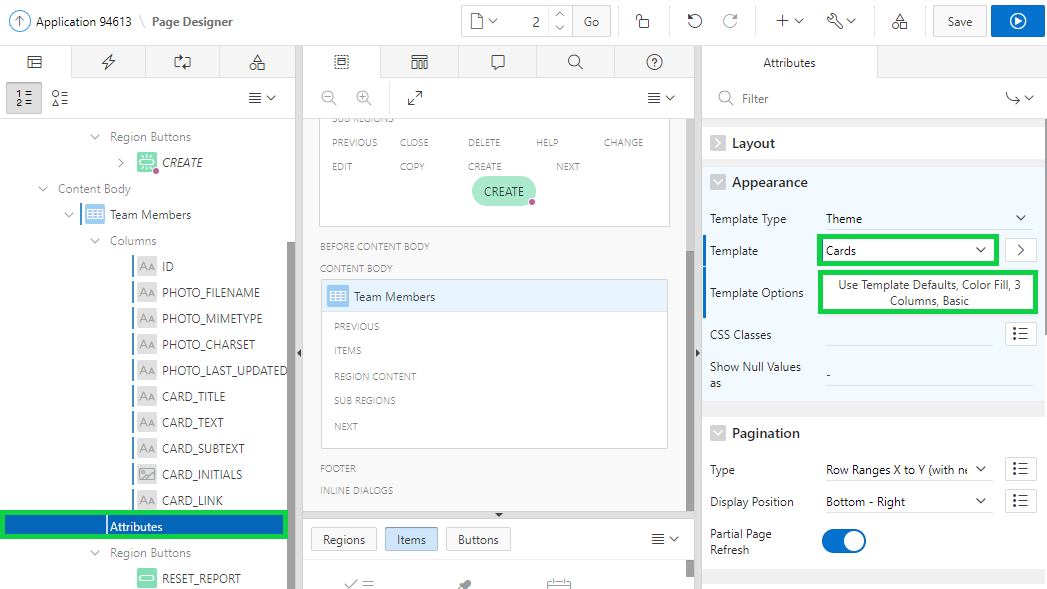
Then, click **OK**.  
  
 **Note**: This SQL query uses specific column names, such as card\_title, which correspond to the   
 columns required for the Card template.

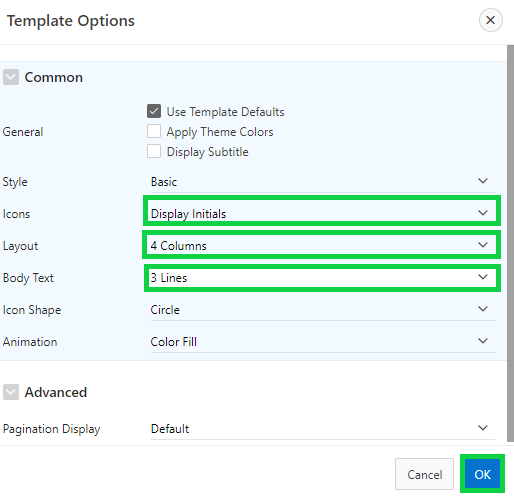
1. The report region has the required Type and SQL Source, now the report needs to be enhanced to display an image and utilize the correct report template.  
   The DEMO\_PROJ\_TEAM\_MEMBERS table includes columns which store an image and the image's associated properties. You can use this information to show the team Member's stored image, instead of the member's initials in the cards report.

Under the **Rendering > Regions**, locate **Team Members**, and expand the **Columns** node.  
Select the **CARD\_INITIALS** column, and input the following:

* Identification: Type - select **Display Image**
* BLOB Attributes: Table Name - select **DEMO\_PROJ\_TEAM\_MEMBERS**
* BLOB Attributes: BLOB Column - select **PHOTO\_BLOB**
* BLOB Attributes: Primary Key Column 1 - select **ID**
* BLOB Attributes: Mime Type Column - select **PHOTO\_MIMETYPE**
* BLOB Attributes: Filename Column - select **PHOTO\_FILENAME**
* BLOB Attributes: Last Updated Column - select **PHOTO\_LAST\_UPDATED**



1. Under Rendering, expand the **Team Members** region. Select **Attributes**.  
   In the property editor, locate **Appearance**. For Template, select **Cards**.  
   Click **Template Options**.  
     
   
2. In the Template Options dialog:

* Icons - select **Display Initials**
* Layout - select **4 Columns**
* Body Text - select **3 Lines**  
  Click **OK**.  
  

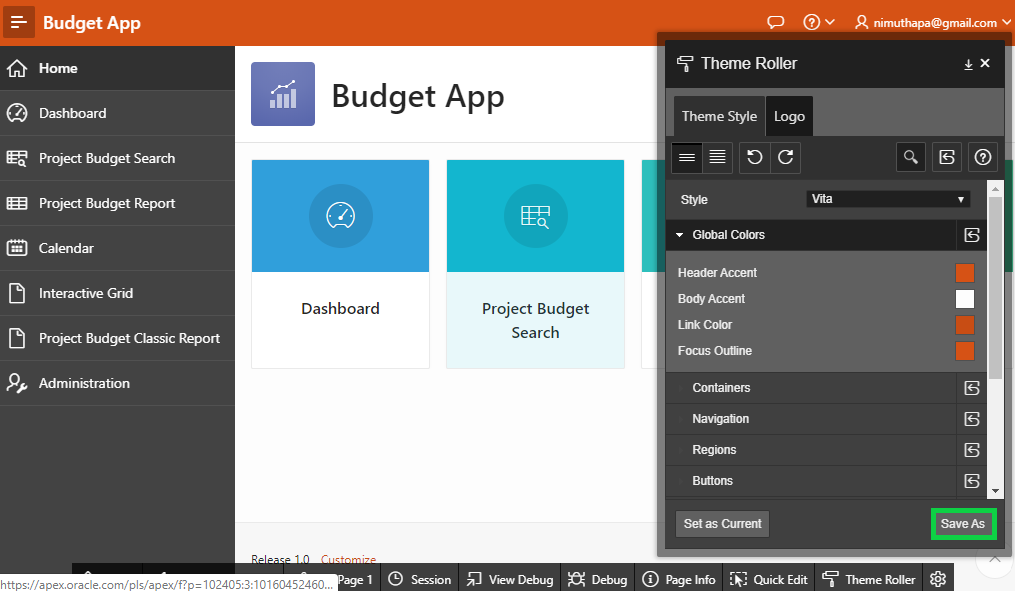
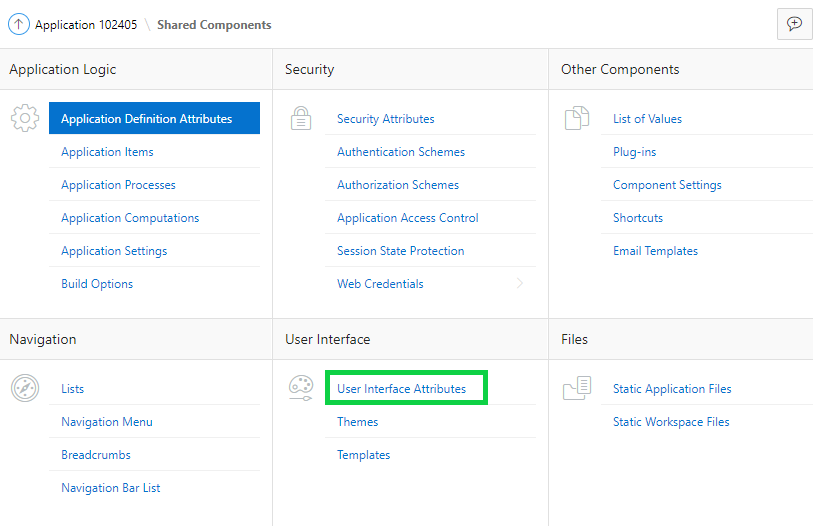
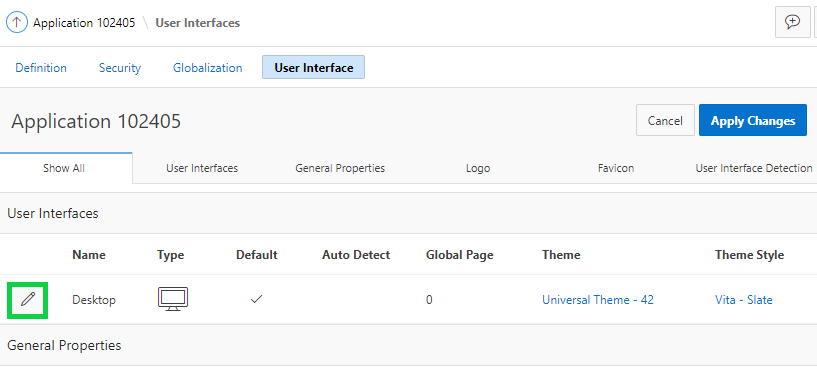
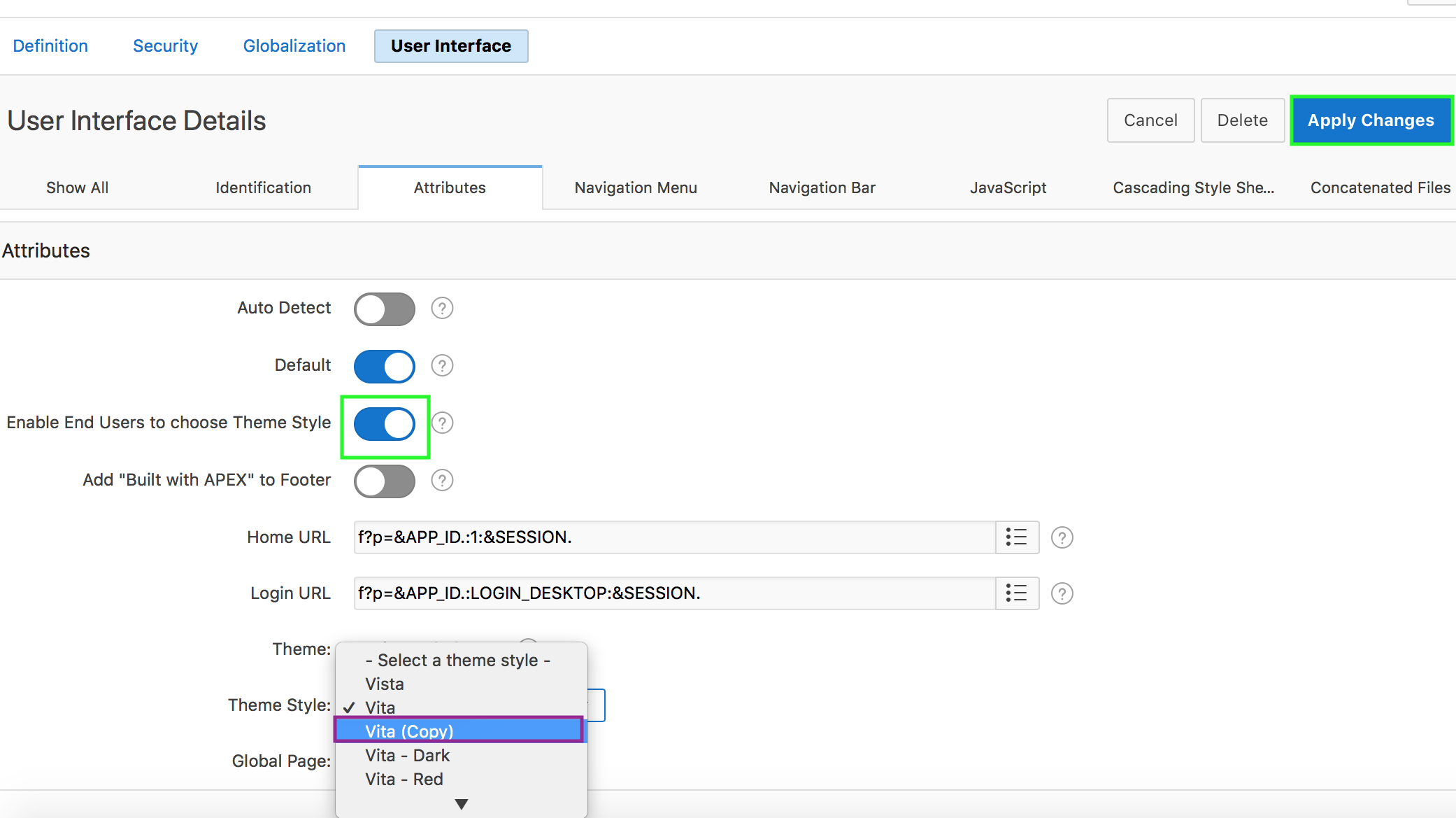
1. Click **Save and Run** **Page** to see the new cards.  
     
   A screenshot of a cell phone

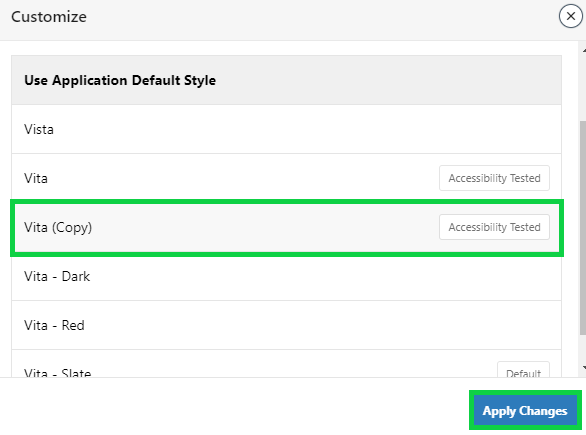
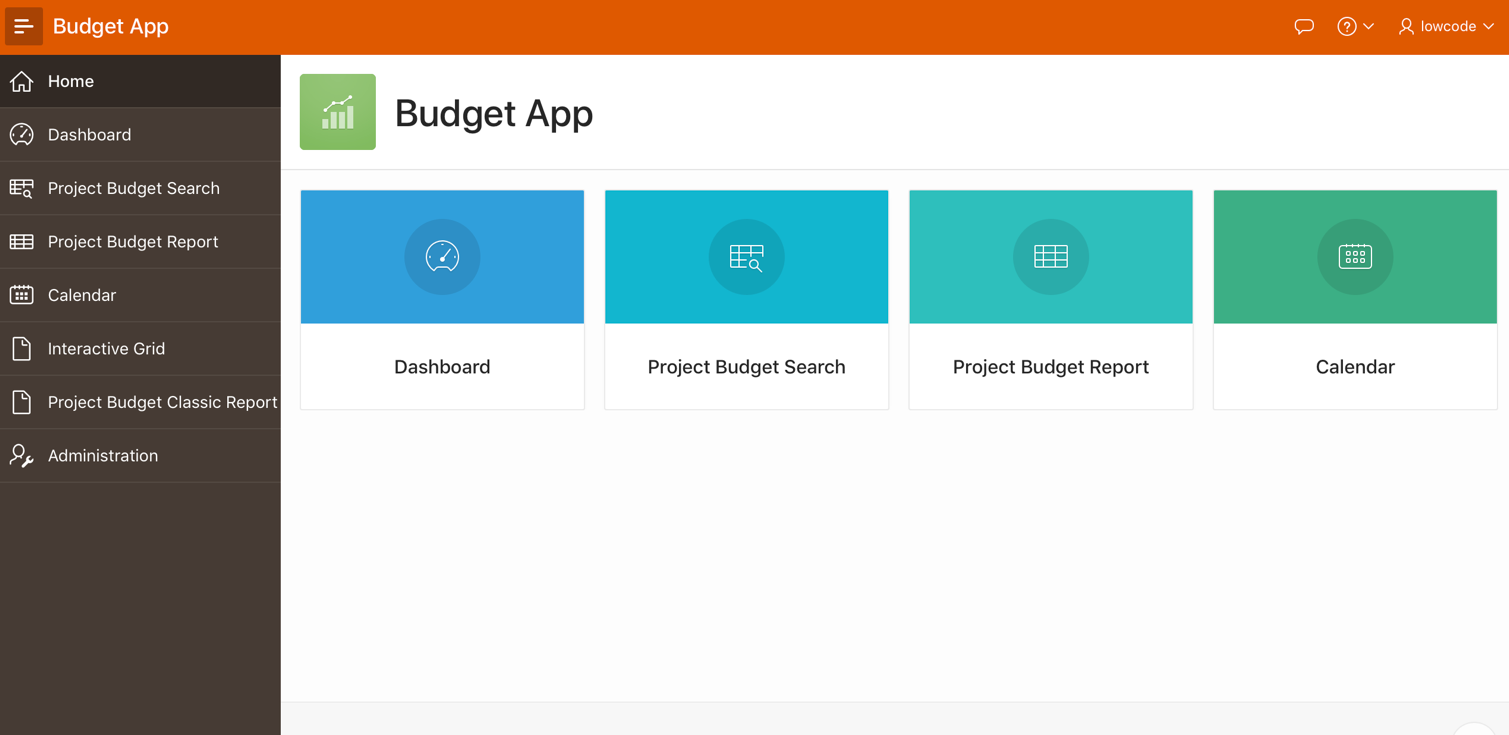
   Description automatically generated

**HOL 12-2: Using Themes and Theme Styles**

In this hands-on lab, you use the Theme Roller to save a Theme Style. Then, you enable end users to apply this style while running the application.

1. Navigate to **App Builder** and then run the **Budget App** application.  
   In the Developer Toolbar, click **Theme Roller**.  
     
   A screenshot of a cell phone

   Description automatically generated
2. The current Theme Style is Vita.  
   Under Global Colors, click the color swatch to select new colors.  
   To edit a specific component, expand a group and select new colors or styling of the component you wish to edit.  
   Modify the look by specifying colors of your choice and preview the changes. Once you are done, click **Save As**.  
     
   
3. In the Save As dialog, enter **Vita (Copy)** for Style Name and click **Save**.
4. Click **OK**.
5. Click **Application < n >** in the Developer Toolbar.  
   In the application home page, click **Shared Components**.  
   Under User Interface, click **User Interface Attributes**.  
     
   
6. Under User Interfaces, click the **Edit** icon for Desktop.  
     
   
7. Under Attributes, click **Enable End Users to Choose Theme Style**. Make sure this attribute is set to On.  
   Notice that the new Theme Style you saved is displayed under Theme Style list.  
   Click **Apply Changes**.  
     
   
8. Now, run the application. At the bottom of the page, click the **Customize** link.  
     
   A screenshot of a cell phone

   Description automatically generated
9. Using the Customize dialog, users can specify their Theme Style while running the application. Select the Theme Style that you saved in step 3 above and click **Apply Changes**.  
     
   
10. The Theme Style is applied, and you see a message that says **Preferences changed for user < username >**.  
      
      
      
      
    A screenshot of a cell phone

    Description automatically generated