School of Computer Sciences, USM, Penang

CMT221/CMM222: Database Organization and Design

Lab 07 – Developing Forms/Pages in Oracle APEX Application

I. Objectives

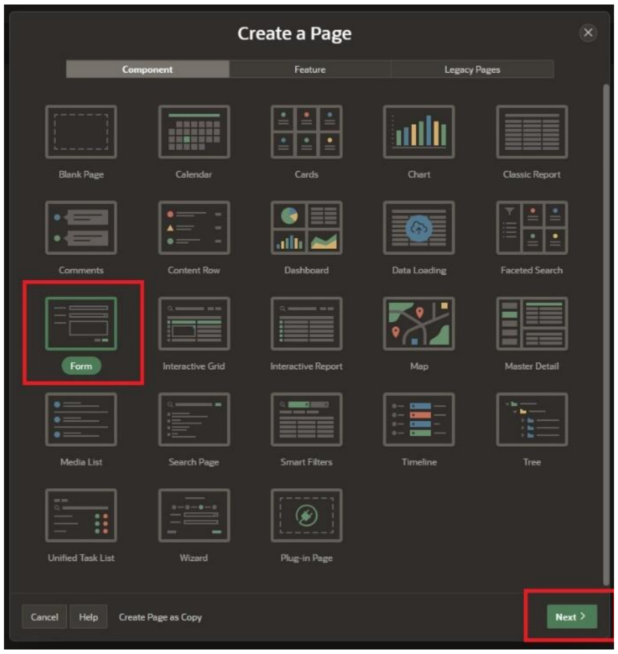
At the end of today's session, you should be able to perform the following:

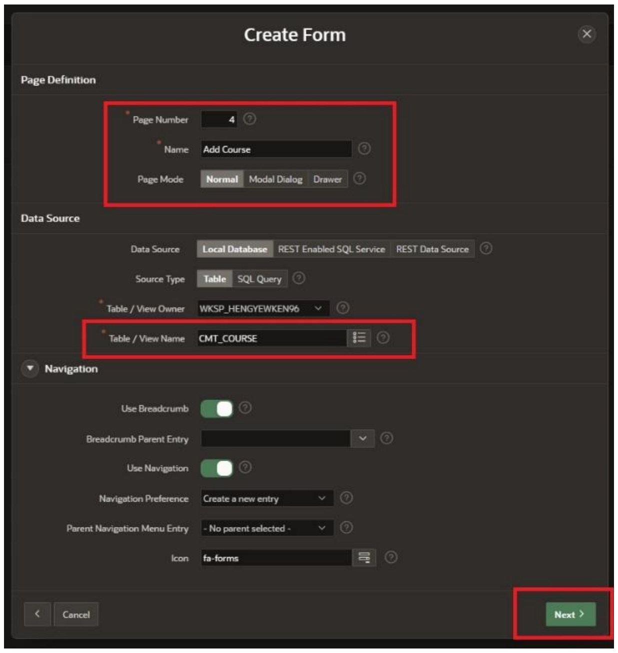
- Define and implement LOVs (list of values).
- Add validations to forms.
- Add view restrictions using an authorization scheme.

II. Setting Up the Application

We have learned about how to create an application with a custom login and sign-up page previously, and we will be further enhancing the application by improving the user experience through various features. Before that, let us create a ***Form*** page to allow users to add new courses to the Oracle database.

1. Set the ***Page Number*** to 4, and name the page as ***Add Course***. Link the ***Table/View Name*** to ***CMT_COURSE***. Click on the ***Next*** button to proceed.




Create Form

Page Definition

Page Number	4
Name	Add Course
Page Mode	[✓ Normal] Modal Dialog Drawer

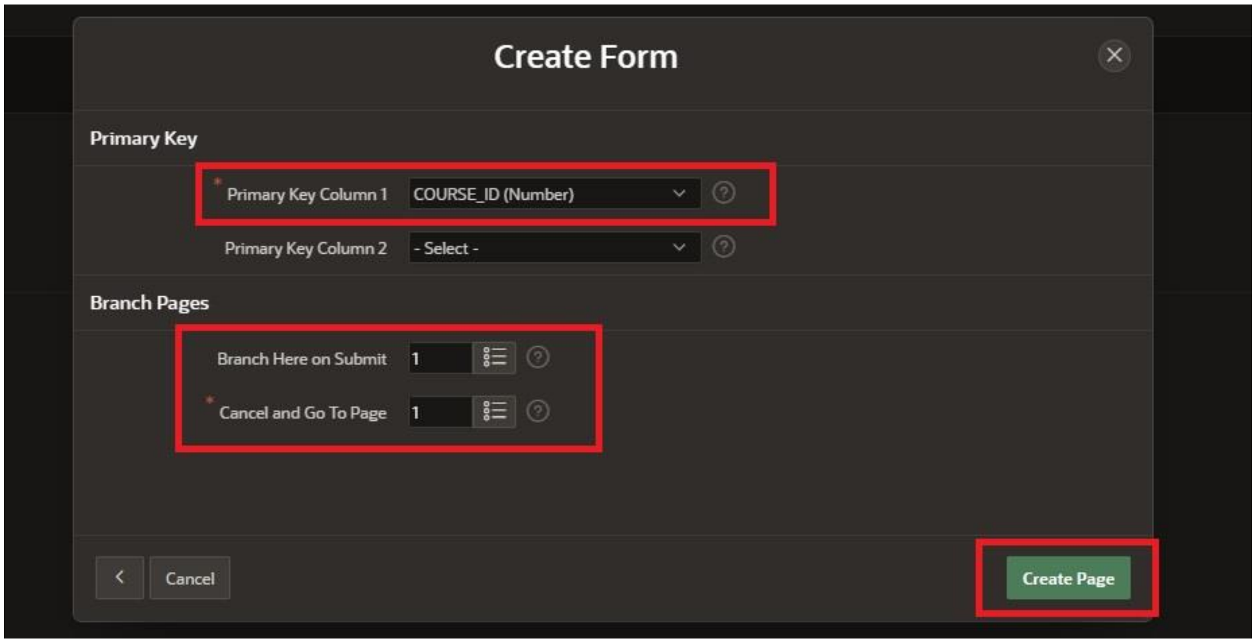
Data Source

Data Source	[✓ Local Database] REST Enabled SQL Service REST Data Source
Source Type	[✓ Table] SQL Query
Table / View Owner	WKSP_HENGYEWKEN96
Table / View Name	CMT_COURSE

Navigation

Use Breadcrumb	✓
Breadcrumb Parent Entry	
Use Navigation	✓
Navigation Preference	Create a new entry
Parent Navigation Menu Entry	- No parent selected -
Icon	fa-forms

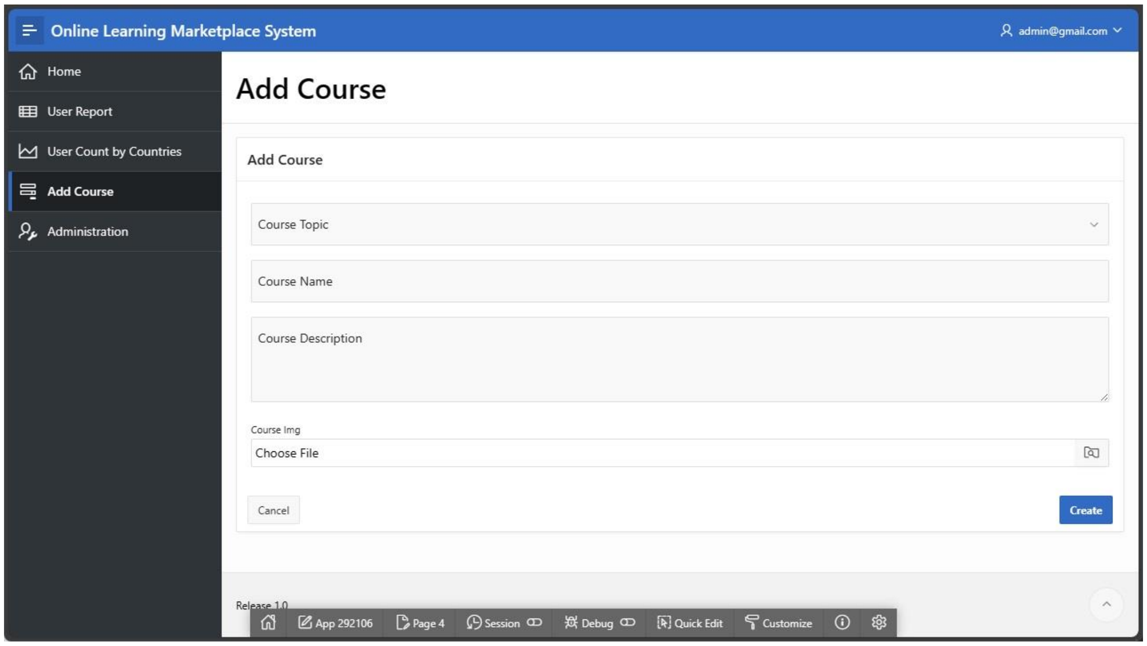
2. Assign ***COURSE_ID*** as ***Primary Key Column 1***. Under the ***Branch Pages*** section, set page 1 for both ***Branch Here on Submit*** and ***Cancel and Go To Page***. Click on the ***Create Page*** button to proceed.



Primary Key Column 1: COURSE_ID (Number)
Primary Key Column 2: - Select -
Branch Here on Submit: 1
Cancel and Go To Page: 1

Q1: Modify the ***Add Course*** page in the ***Page Designer*** according to the instructions below and capture screenshots of the solutions. The updated ***Add Course*** page should look like the figure below.

a) Set the course ID, account user ID, and course uploaded time to hidden.
b) Assign the course sequence as the default value of the course ID.
c) Assign the current logged-in user as the default value of the account user ID.
d) Assign ***SYSDATE*** as the default value of the course uploaded time.



III. Adding LOVs (List of Values)

By default, most of the character-valued attributes, such as CHAR and VARCHAR2, are set to accept input from a text field. It is normal for users to input attributes like the name and description on their own, but it would be a problem for users to input attributes that are used for categorization (i.e., course topic). Under normal circumstances, users are not aware of what kinds of topics are available in the system. An inappropriate input will cause the course to be categorized under an undesired section. At this point, we would expect the course topic to show a drop-down list that displays all available options for selection, and that is a feature available in Oracle APEX, known as LOV (List of Values).

A LOV is a static or dynamic set of values used to display a specific type of page item, such as pop-up lists of values, a select list, a check box, a radio group, or multiple select lists. By creating a list of values at the application-level, you are creating shared components that can be used anywhere within the application. Creating a LOV as a shared component has several advantages:

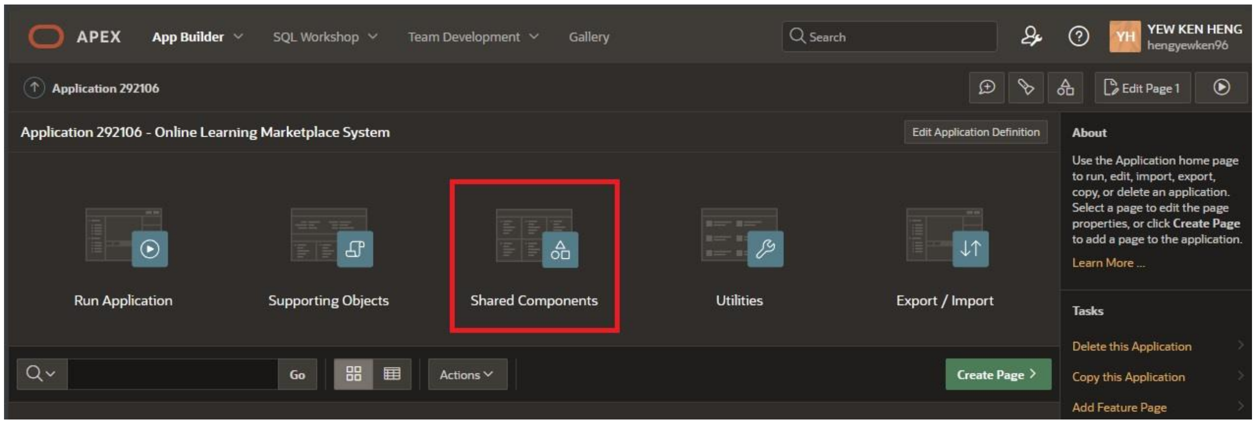
- You can add it to any page within an application.
- All LOV definitions are stored in one location, making them easy to locate and update.

After the recent updates, Oracle APEX will create LOVs automatically and set the type of the corresponding attributes for you, though it will still miss out on some and leave them as text fields. As such, we will discuss about how to create LOVs suited for our use case.

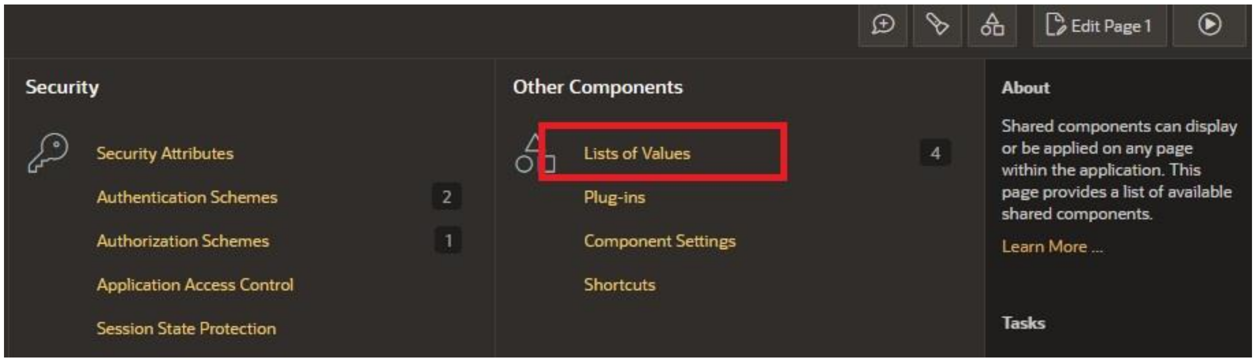
A. Static LOV

Static LOVs are based on a static list of display values and return values you specify when you run the ***Create LOV Wizard***. Static LOVs are not linked to any existing tables in the Oracle database, so users will have to define the values manually. As static LOVs require values hard-coded for the display and return values, it is recommended to use them only for simple selection lists, for example, gender (which typically consists of male, female, and do not show gender). For attributes that need to add new values and scale automatically, it is recommended to use dynamic LOV instead. To create a static LOV:

1. On the application home page, click on ***Shared Components***.

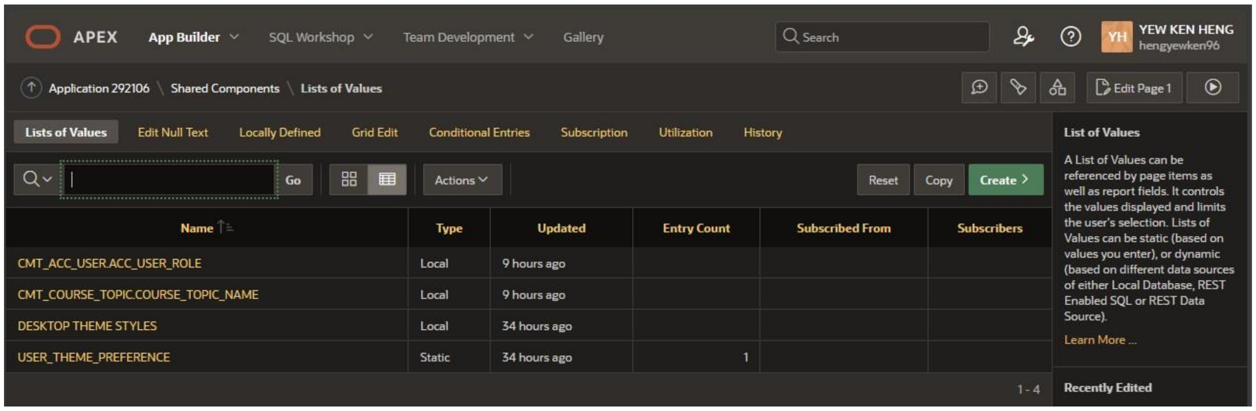


2. Under ***the Other Components*** section, click on ***List of Values***.



Shared components can display or be applied on any page within the application. This page provides a list of available shared components.

3. On the List of Values page, you will see all available LOVs in the list. You can click on the LOV to modify or delete it.
4. Try deleting the ***CMT_ACC_USER.ACC_USER_ROLE*** (skip this if it is not available for you).
5. Click on the ***Create*** button to create a new LOV.



Name	Type	Updated	Entry Count	Subscribed From	Subscribers
CMT_ACC_USER.ACC_USER_ROLE	Local	9 hours ago			
CMT_COURSE_TOPIC.COURSE_TOPIC_NAME	Local	9 hours ago			
DESKTOP THEME STYLES	Local	34 hours ago			
USER_THEME_PREFERENCE	Static	34 hours ago	1		

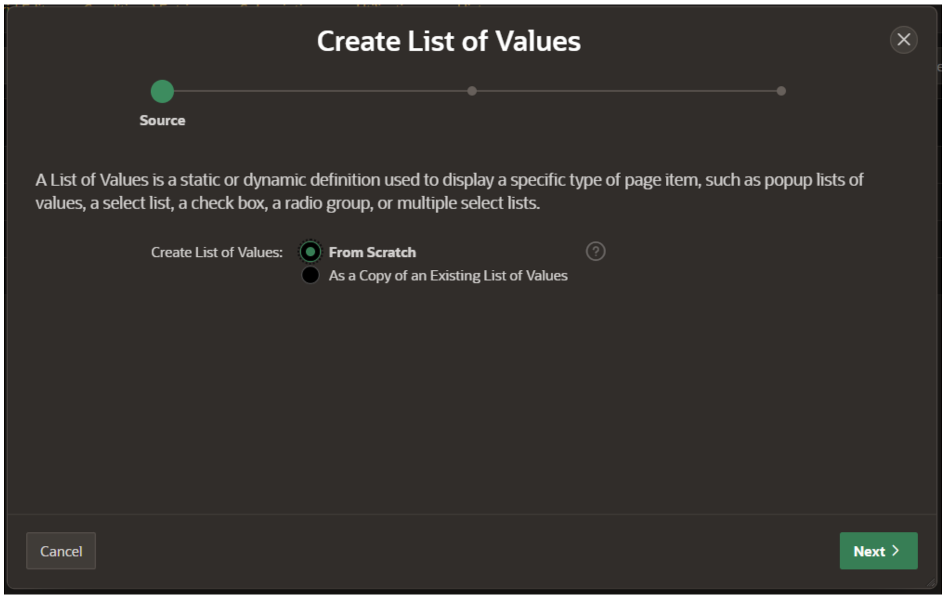
1 - 4

List of Values

A List of Values can be referenced by page items as well as report fields. It controls the values displayed and limits the user's selection. Lists of Values can be static (based on values you enter), or dynamic (based on different data sources of either Local Database, REST Enabled SQL or REST Data Source).

Learn More ...

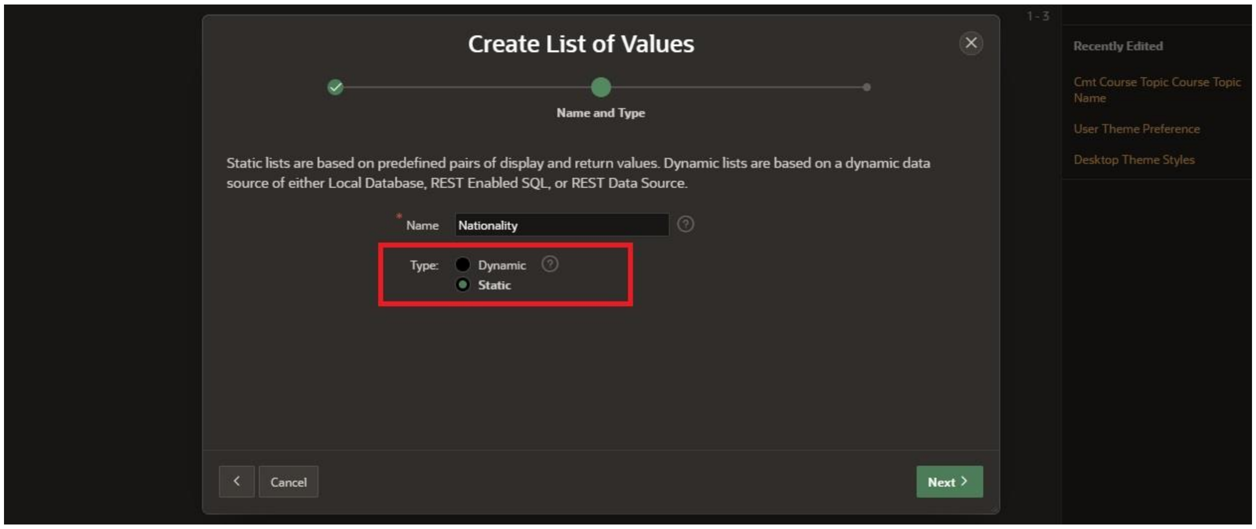
6. In the pop-up window under ***Source***, select ***From Scratch*** in the ***Create List of Values*** radio button option.



A List of Values is a static or dynamic definition used to display a specific type of page item, such as popup lists of values, a select list, a check box, a radio group, or multiple select lists.

Create List of Values: [✓ From Scratch] As a Copy of an Existing List of Values

7. Next, under ***Name and Type***, choose ***Static*** for the ***Type*** and input ***Nationality*** in the ***Name*** field.

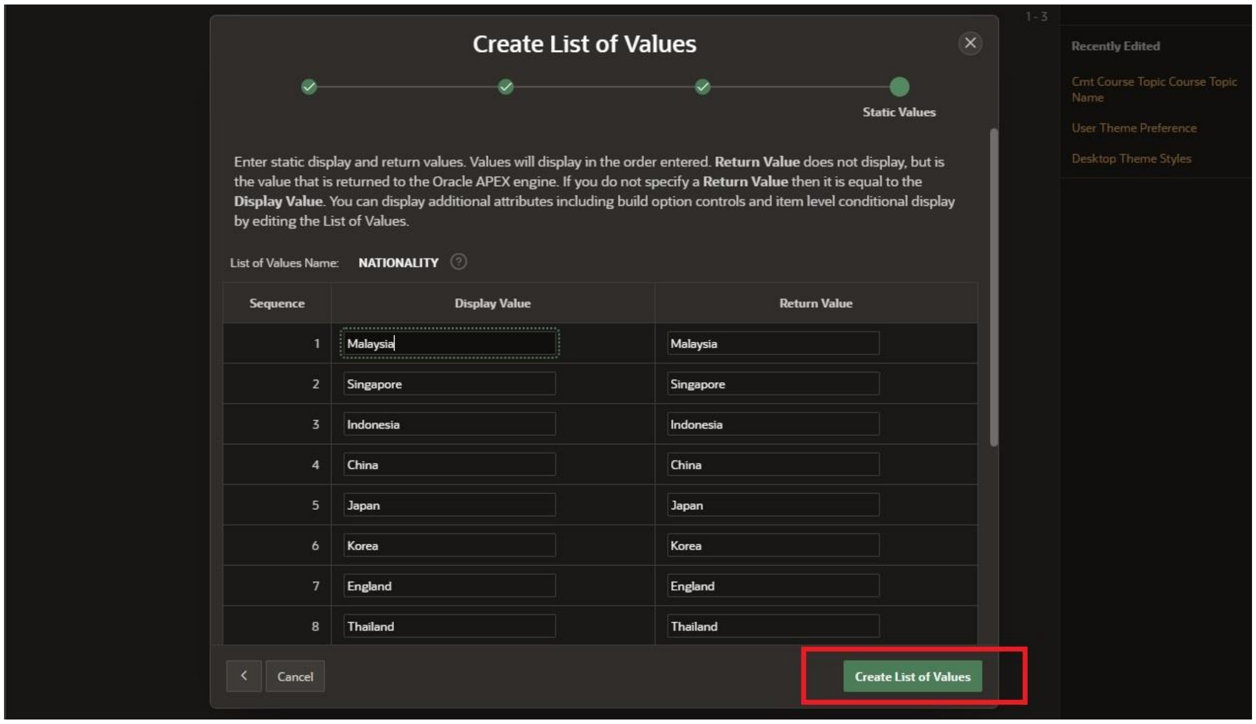


Static lists are based on predefined pairs of display and return values. Dynamic lists are based on a dynamic data source of either Local Database, REST Enabled SQL, or REST Data Source.

Name: Nationality

Type: Dynamic [✓ Static]

8. In the ***Create List of Values*** window, you can define your own values for selection. The display value is the value that will be shown in the selection list, while the return value is the actual value that will be sent to the Oracle database for operations. Once you are done with the settings, click on ***Create List of Values*** button to proceed.

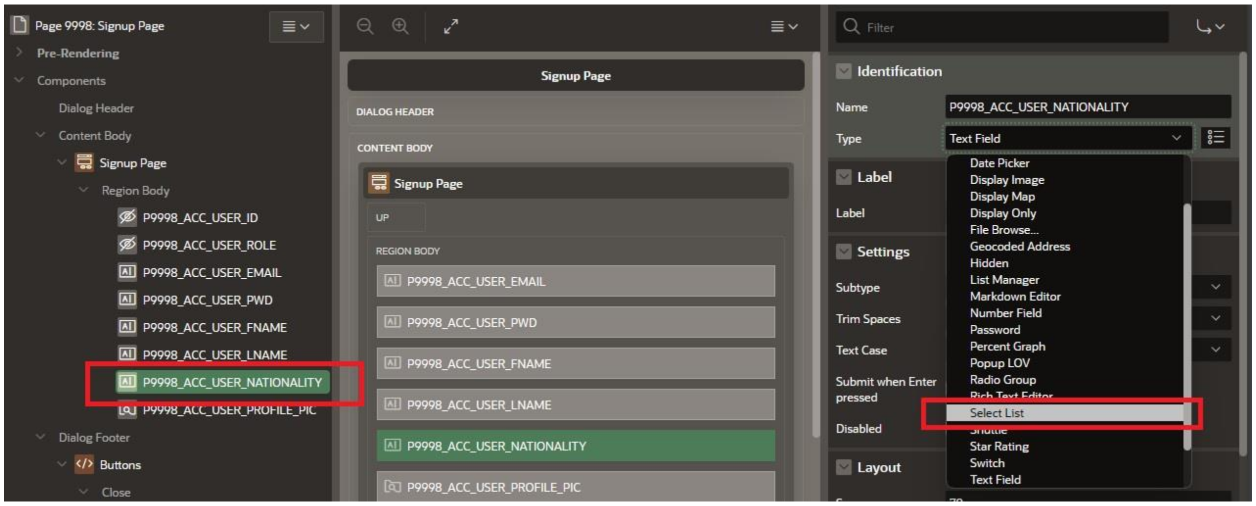


Enter static display and return values. Values will display in the order entered. **Return Value** does not display, but is the value that is returned to the Oracle APEX engine. If you do not specify a **Return Value** then it is equal to the **Display Value**. You can display additional attributes including build option controls and item level conditional display by editing the List of Values.

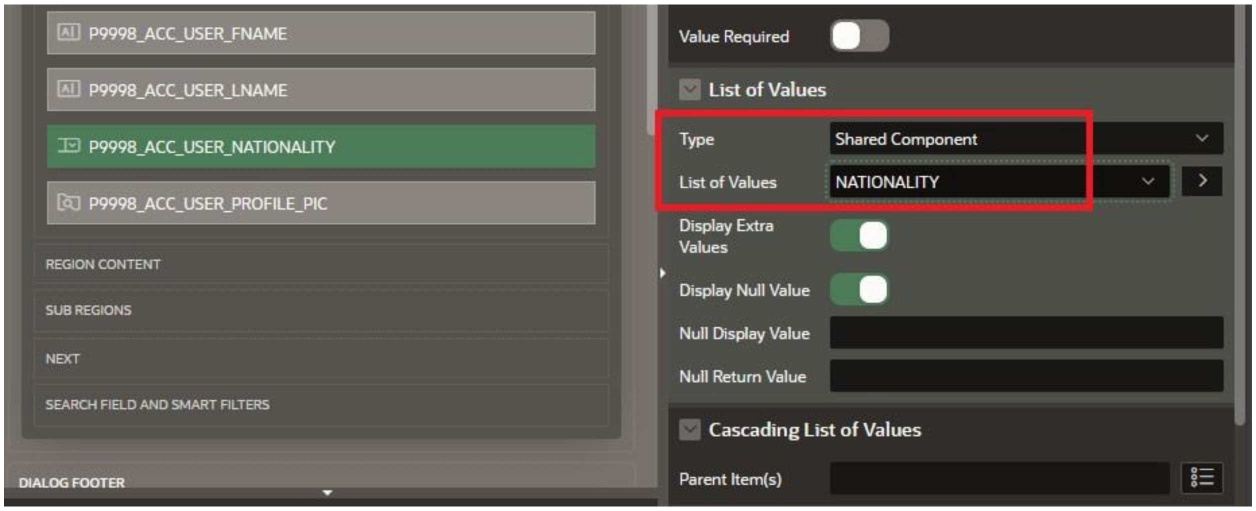
List of Values Name: NATIONALITY

Sequence	Display Value	Return Value
1	Malaysia	Malaysia
2	Singapore	Singapore
3	Indonesia	Indonesia
4	China	China
5	Japan	Japan
6	Korea	Korea
7	England	England
8	Thailand	Thailand

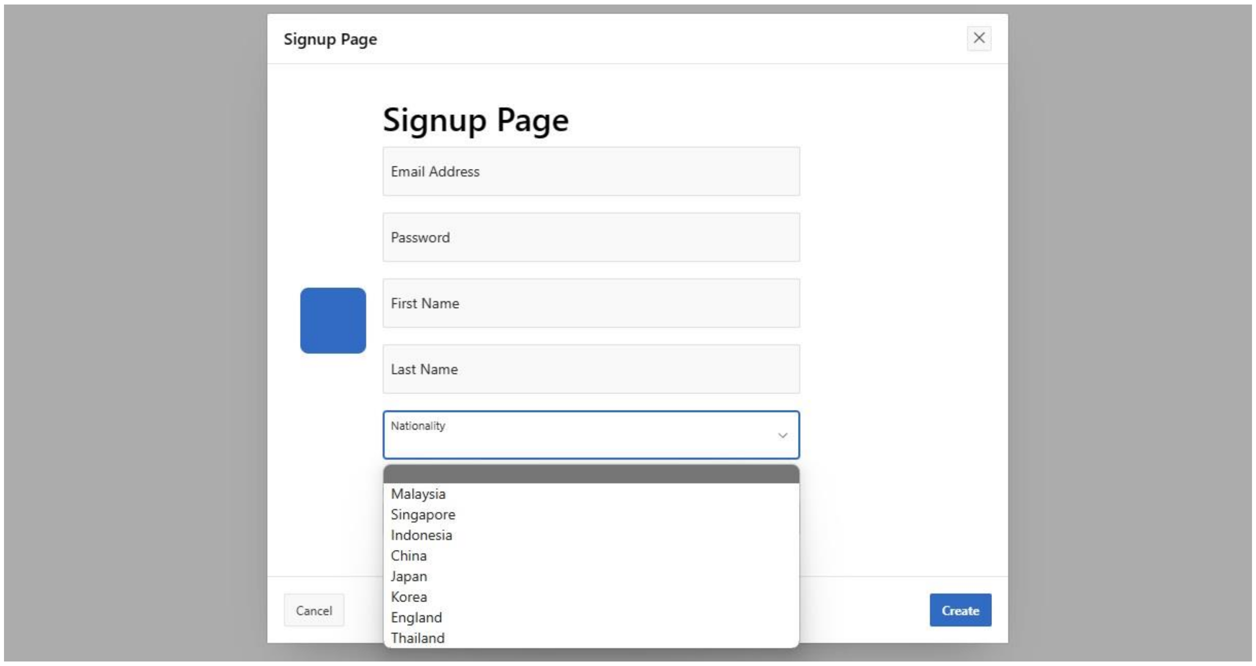
9. To test it out, go to the ***Page Designer*** for ***Page 9998 – Signup Page*** and select the attribute ***P9998_ACC_USER_NATIONALITY*** on the left-hand pane. On the right-hand pane under the ***Identification*** section, change the ***Type*** to ***Select List***.



10. After you have changed the type of the attribute to ***Select List***, you will see a new section (i.e., ***List of Values***) show up on the right-hand pane. Scroll down to the section and select ***Shared Component*** for the ***Type***. For the ***List of Values***, select ***NATIONALITY*** that you have created previously. Save and run the application to see the changes.



11. Go to the sign-up page after running your application and click on the ***Nationality*** field. You will now see a drop-down list showing the values that you have defined in the static LOV.



Signup Page

Signup Page

Email Address

Password

First Name

Last Name

Nationality

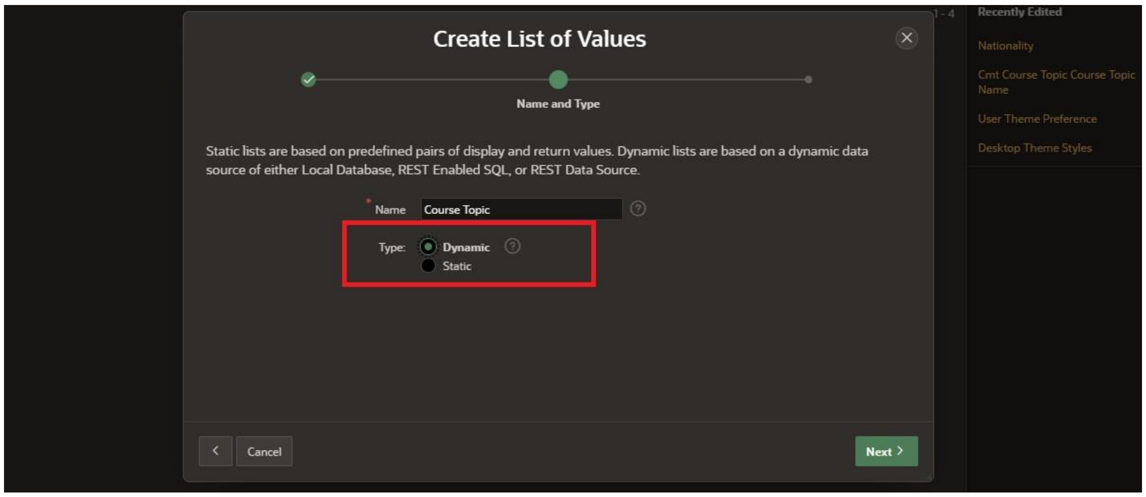
Malaysia
Singapore
Indonesia
China
Japan
Korea
England
Thailand

12. Next, we will create a dynamic LOV, which is another type of supported LOV in Oracle APEX.

B. Dynamic LOV

Different from static LOVs, which require manual value definitions, dynamic LOVs retrieve data automatically from a local data source, a REST-enabled SQL reference, or a REST data source. Dynamic LOVs are commonly used to fetch data that will be in a constant state of change, hence having better scalability and versatility than static LOVs in terms of their usage. To create a dynamic LOV:

1. In the ***Create List of Values*** window, choose ***From Scratch*** for ***Create List of Values*** under ***Source***. Then, under ***Name and Type***, choose ***Dynamic*** for the ***Type*** and input ***Course Topic*** in the ***Name*** field.



Create List of Values

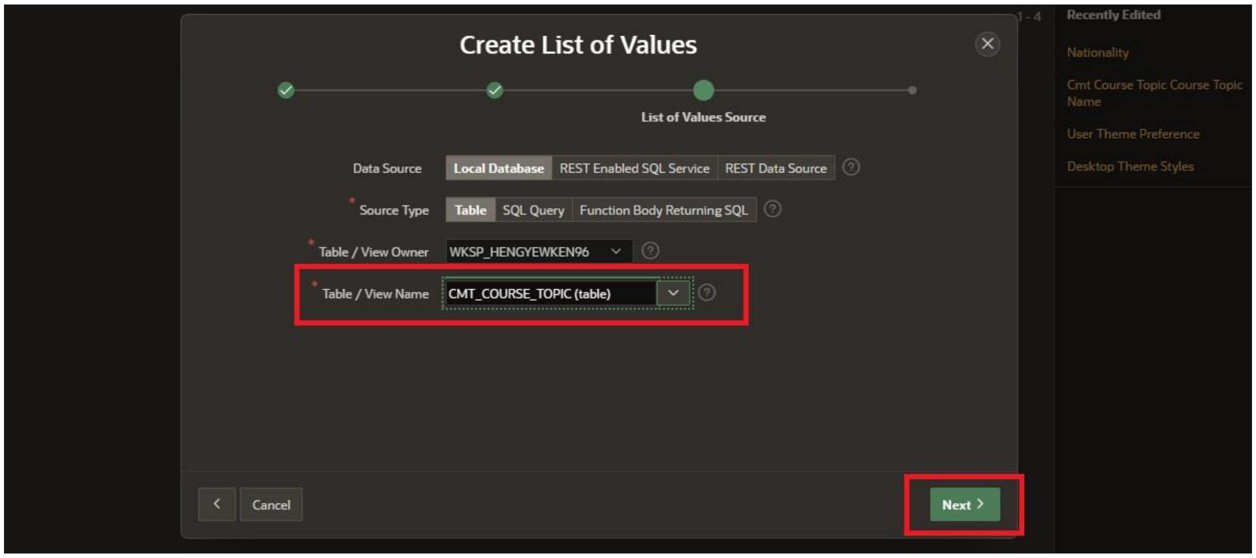
Name and Type

Static lists are based on predefined pairs of display and return values. Dynamic lists are based on a dynamic data source of either Local Database, REST Enabled SQL, or REST Data Source.

* Name: Course Topic

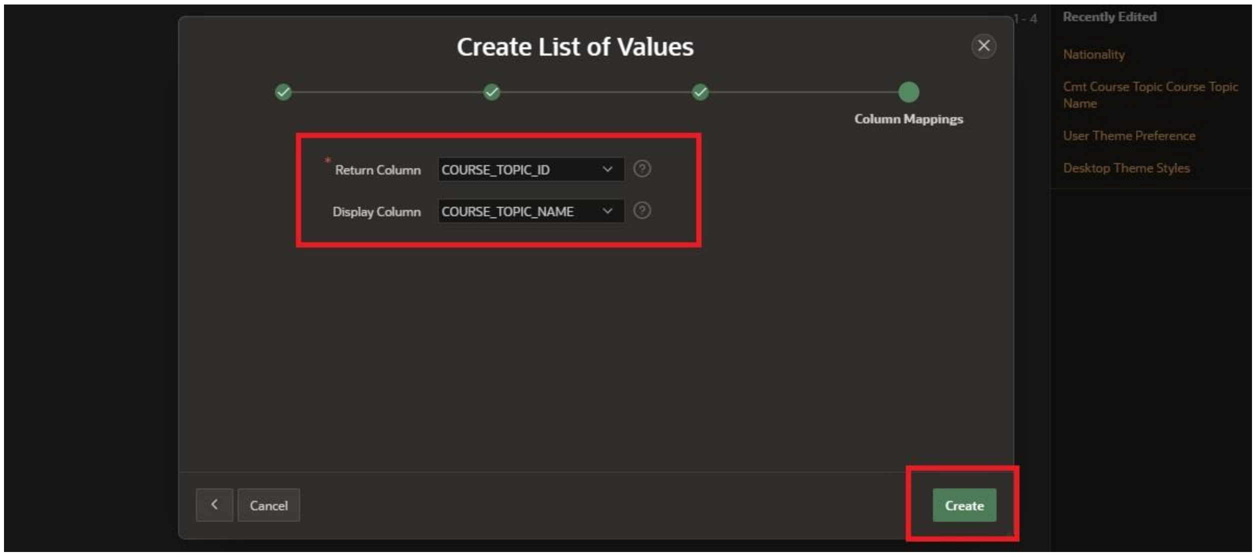
Type: [✓ Dynamic] Static

2. Next, select ***CMT_COURSE_TOPIC*** for the ***Table/View Name***. Click on the ***Next*** button to proceed.



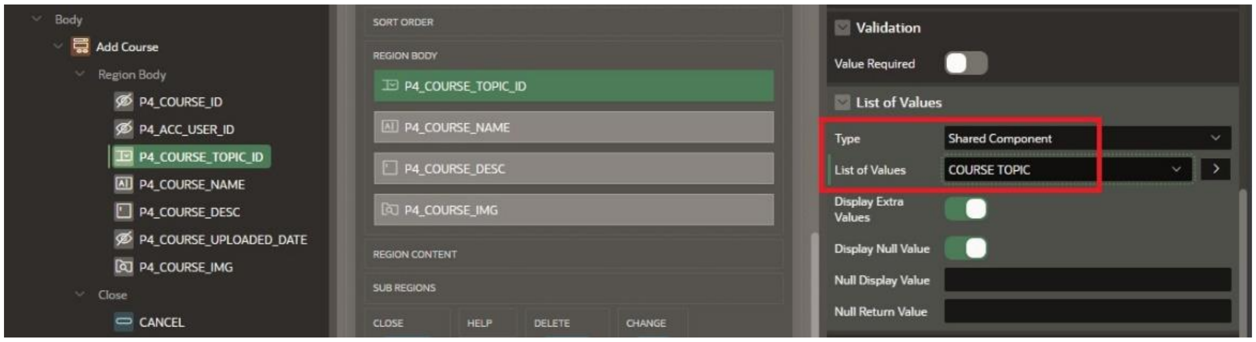
Data Source: Local Database
Source Type: Table
Table / View Owner: WKSP_HENGYEWKEN96
Table / View Name: CMT_COURSE_TOPIC (table)

3. For the column mappings, select ***COURSE_TOPIC_ID*** for the ***Return Column*** and ***COURSE_TOPIC_NAME*** for the ***Display Column***. Click on the ***Create*** button to proceed.



Return Column: COURSE_TOPIC_ID
Display Column: COURSE_TOPIC_NAME

4. Go to the ***Page Designer*** for ***Page 4 – Add Course*** and select the attribute ***P4_COURSE_TOPIC_ID*** on the left-hand pane. On the right-hand pane under the ***List of Values*** section, change the ***List of Values*** to ***COURSE TOPIC***. Save and run the application to see the changes.



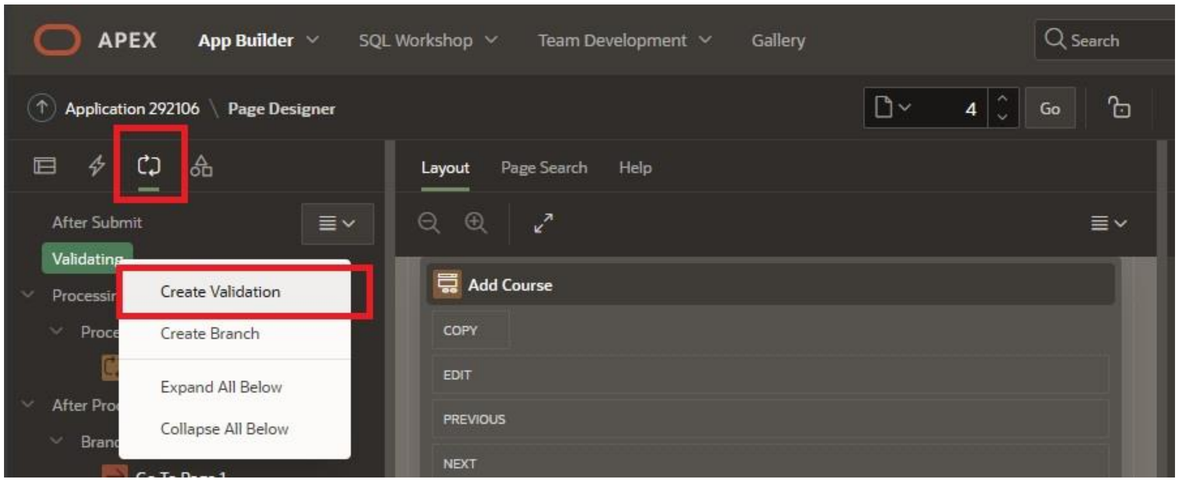
Type: Shared Component
List of Values: COURSE TOPIC

IV. Adding Validations

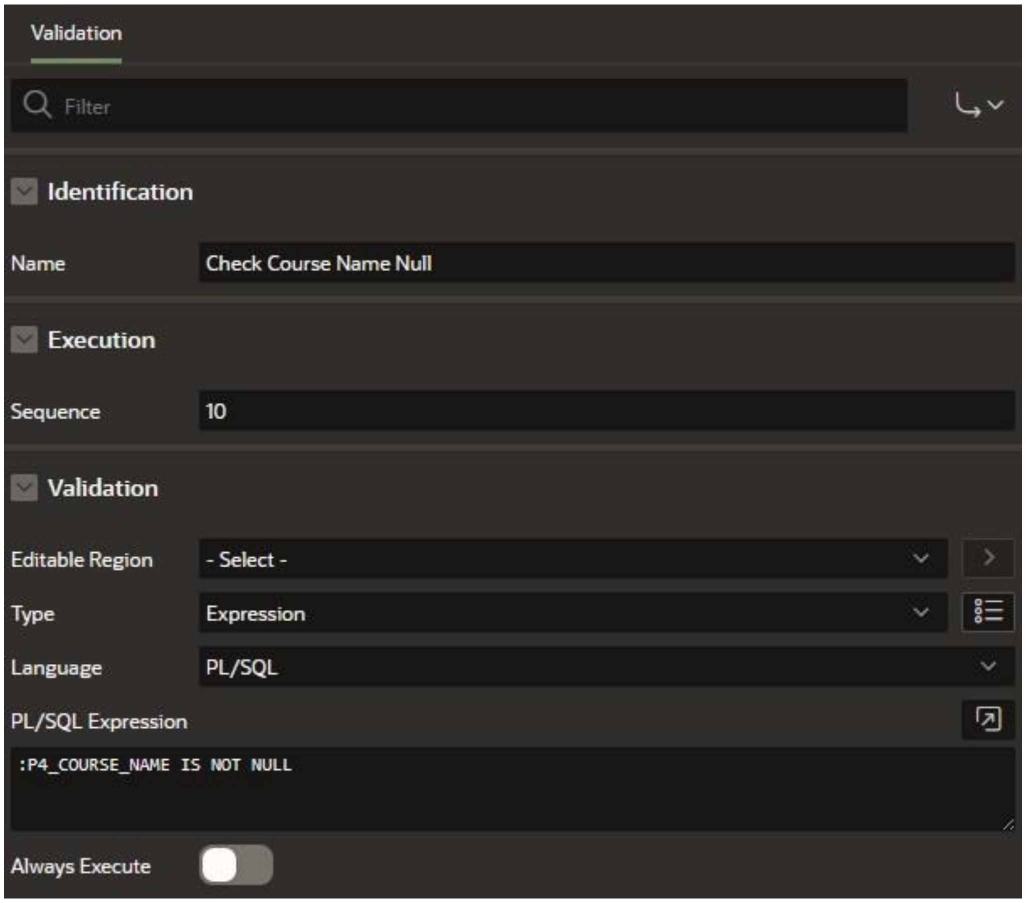
We have learned about how to create LOVs and integrate them into forms. Next, we will be adding validations to the forms. Validations are edit checks that developers can create to check the data a user enters before processing. Once you create a validation and the associated error message, you can associate it with a specific item. You can choose to have validation error messages displayed inline (that is, on the page where the validation is performed) or on a separate error page. Meanwhile, inline error messages can be displayed in a notification area or within the field label.

To create a validation:

1. Go to the ***Page Designer*** for ***Page 4 – Add Course*** and select the ***Processing*** tab on the left-hand pane. Right click on the ***Validating*** section and select ***Create Validation***.



2. Set the validation name to ***Check Course Name Null***. Make sure to select the ***Editable Region*** (if applicable) and set the ***Type*** to ***Expression***. We will be using PL/SQL to write the validation expression. Input the following expression to check whether the course name field is empty.
   :P4_COURSE_NAME IS NOT NULL

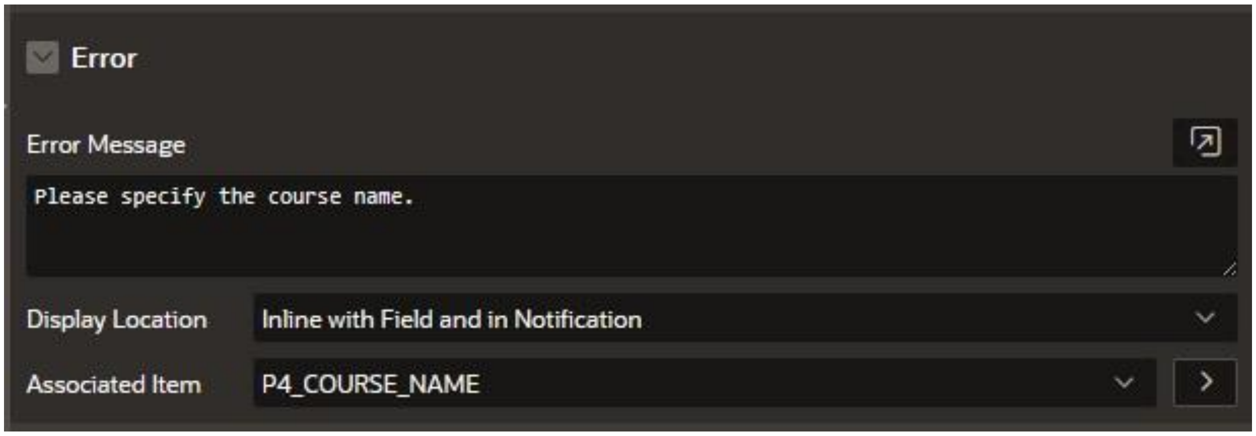


Name: Check Course Name Null
Sequence: 10
Editable Region: - Select -
Type: Expression
Language: PL/SQL
PL/SQL Expression:

```
:P4_COURSE_NAME IS NOT NULL
```

Always Execute

3. Under the ***Error*** section, specify the error message that you would like to display. Please make sure that the message is easy to understand and relatable to the error itself. Set the Display Location to ***Inline with Field and in Notification*** and ***Associated Item*** to ***P4_COURSE_NAME***. Save and run the application to see the changes.

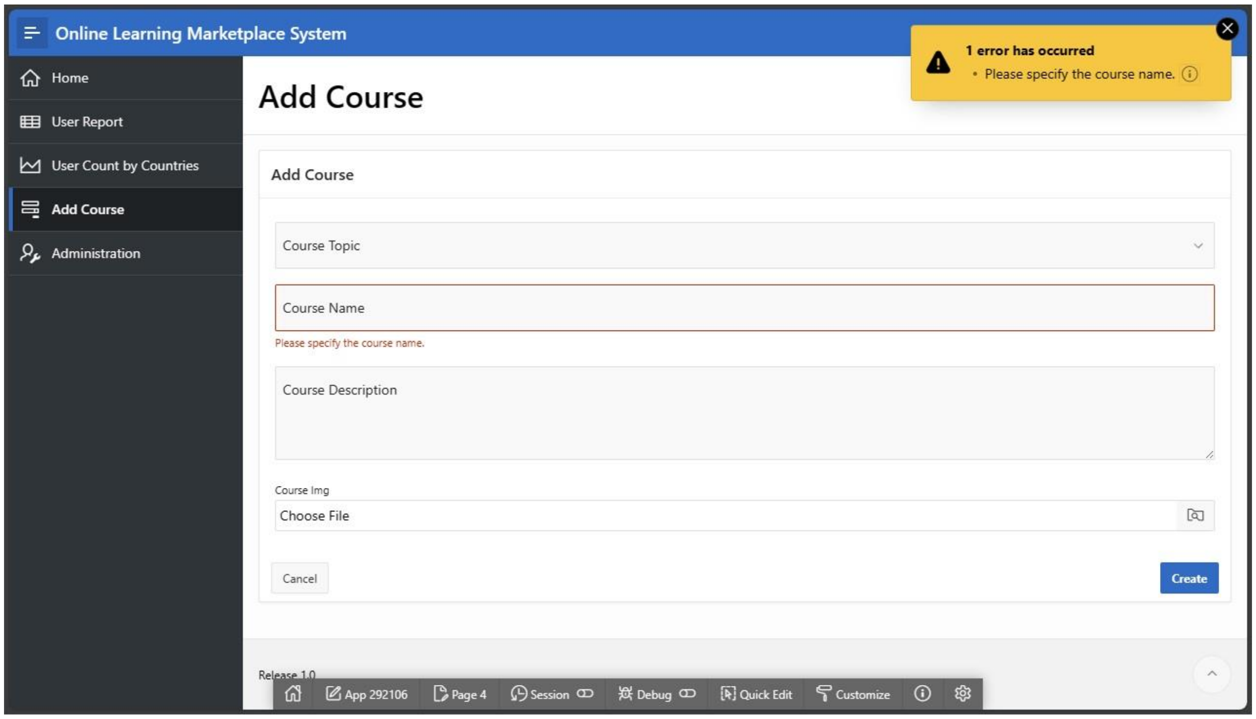


Error Message:

```
Please specify the course name.
```

Display Location: Inline with Field and in Notification
Associated Item: P4_COURSE_NAME

4. Go to the ***Add Course*** page in your application and test out the validation. You should see the error message displayed on the top right corner as well as below the ***Course Name*** field whenever you leave an empty string for the field.



1 error has occurred
- Please specify the course name.

Please specify the course name.

<u>Q2:</u> Explain the similarities and differences between validations and SQL constraints.

V. Adding View Restrictions Using Authorization Scheme

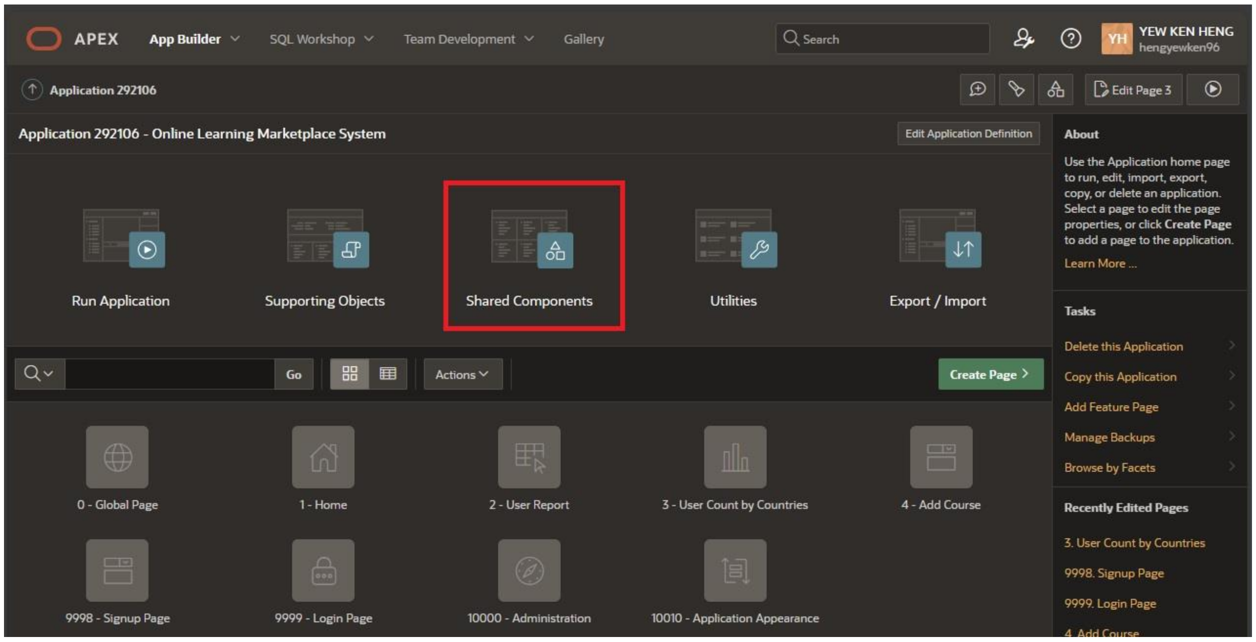
Now that we have learnt to create validations that can be used to perform checking on user input in forms, we can further enhance the application by adding view restrictions. Basically, we want to restrict certain pages to only being visible to users with authorization. For example, we want to restrict the application so that only admins are authorized to navigate to ***Add Course*** and ***User Report*** pages. By limiting access to certain pages or functionalities based on user roles, you can reduce the risk of unauthorized access to sensitive information or critical system functions. This helps protect user data, business data, and the overall integrity of your application. At the same time, role-based navigation can streamline workflows by providing users with the information and tools they need to perform their tasks efficiently. This can boost productivity and reduce the time users spend searching for relevant features or content. This can be done in Oracle APEX by setting up an authorization scheme.

You can specify an authorization scheme for an entire application, page, or specific control such as a region, item, or button. For example, you could use an authorization scheme to selectively determine which tabs, regions, or navigation bars a user sees. An authorization scheme will either succeed or fail, depending on the condition. Common authorization scheme types include Exists, Not Exists SQL Queries, and PL/SQL Function Returning Boolean. If a component or control-level authorization scheme succeeds, the user can view the component or gain control over it, and vice versa.

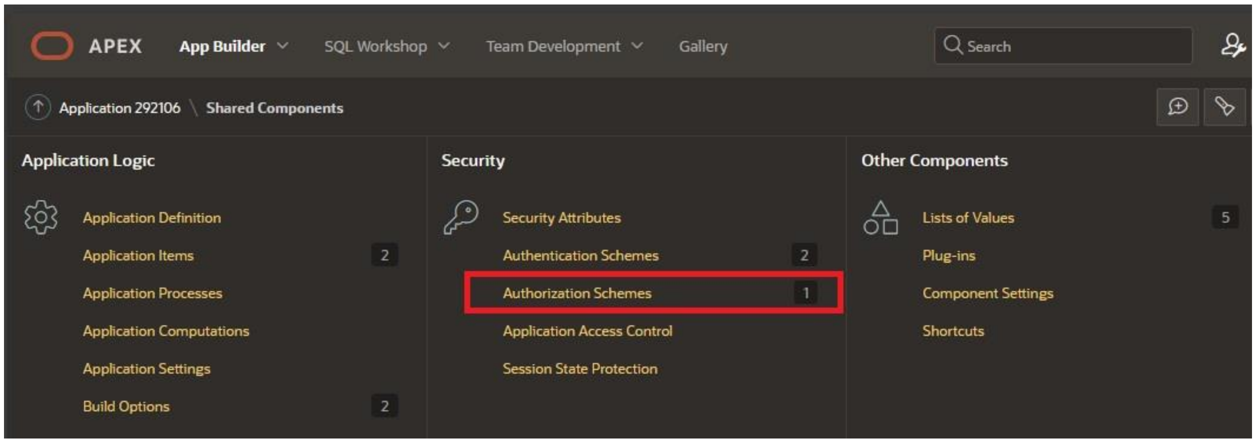
A. Adding the Authorization Scheme

Before we can restrict a page view or component view, we need to define an authorization scheme first. To create an authorization scheme:

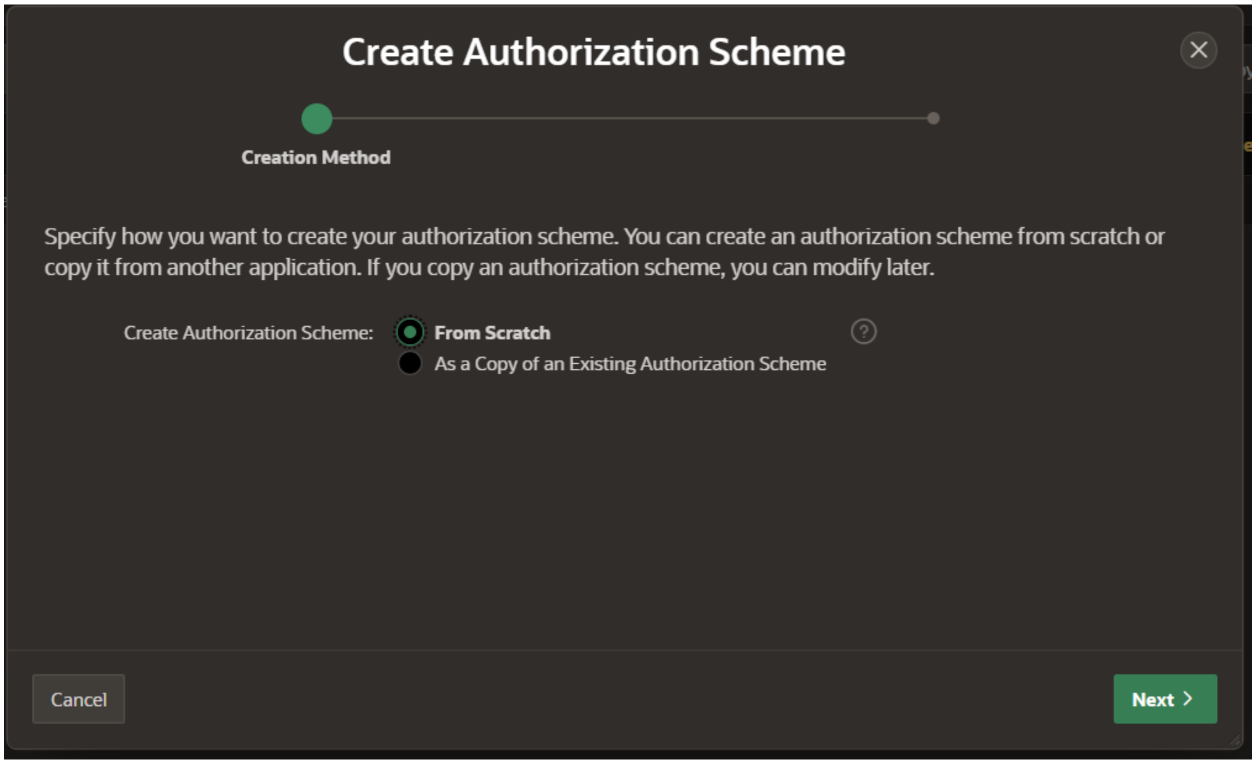
1. On the application home page, click on ***Shared Components***.



2. Under the ***Security*** section, click on ***Authorization Schemes***.



3. Click on the ***Create*** button to create a new authorization scheme.
4. In the ***Create Authorization Scheme*** window, under ***Creation Method***, select the default ***Create Authorization Scheme*** option: ***From Scratch***. Click ***Next***.



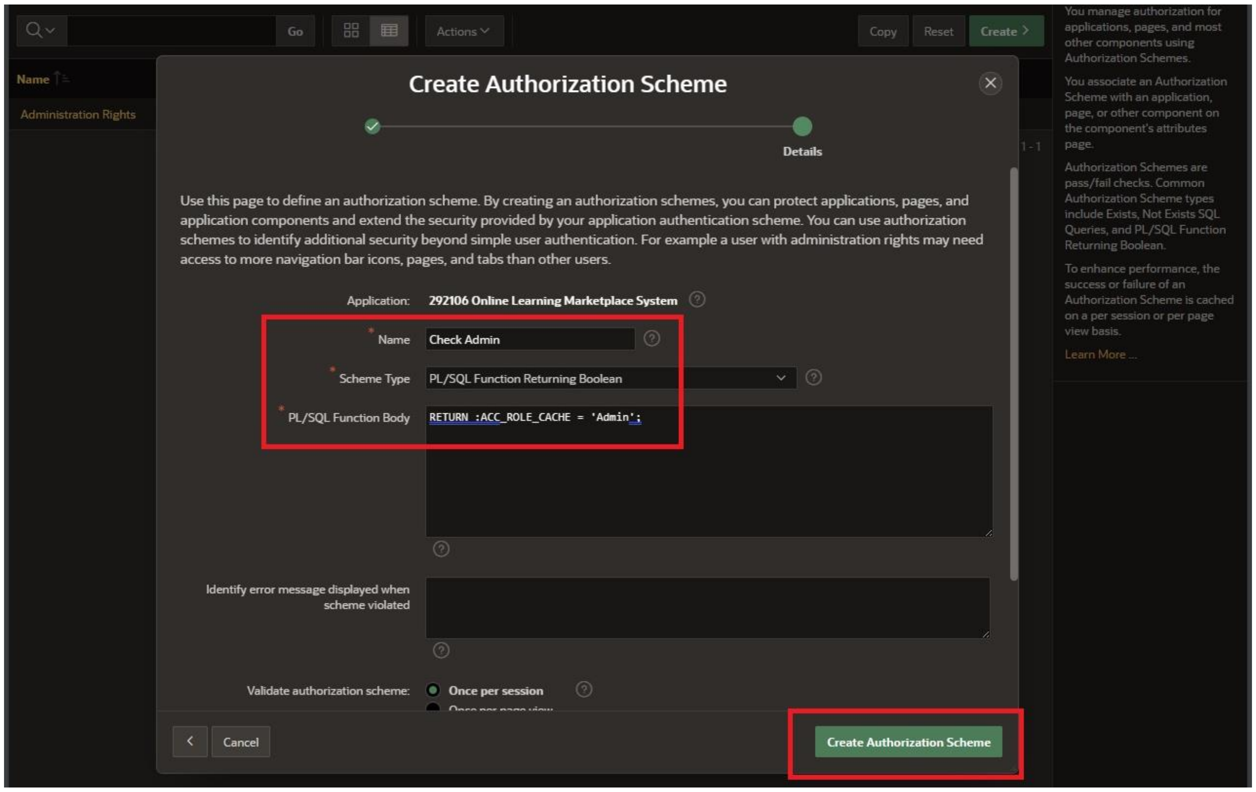
Specify how you want to create your authorization scheme. You can create an authorization scheme from scratch or copy it from another application. If you copy an authorization scheme, you can modify later.

Create Authorization Scheme: [✓ From Scratch] As a Copy of an Existing Authorization Scheme

5. Under ***Details***, type ***Check Admin*** for the ***Name*** field. Select ***PL/SQL Function Returning Boolean*** as the ***Scheme Type***. In the ***PL/SQL Function Body*** field, insert the following statement:
   RETURN :ACC_ROLE_CACHE = 'Admin';
6. Basically, you want to verify whether the user is an admin. If you recall back to the previous lab, we created an application item to store the user role (i.e., ***ACC_ROLE_CACHE***). The PL/SQL statement will return a Boolean value by comparing the application item value with

the phrase '***Admin***'. If the values are equal, then the authorization scheme will return true, and vice versa.

7. Click on the ***Create Authorization Scheme*** button to complete the setup.



Create Authorization Scheme

Use this page to define an authorization scheme. By creating an authorization schemes, you can protect applications, pages, and application components and extend the security provided by your application authentication scheme. You can use authorization schemes to identify additional security beyond simple user authentication. For example a user with administration rights may need access to more navigation bar icons, pages, and tabs than other users.

Application: 292106 Online Learning Marketplace System

Name: Check Admin

Scheme Type: PL/SQL Function Returning Boolean

PL/SQL Function Body:

```
RETURN :ACC_ROLE_CACHE = 'Admin';
```

Identify error message displayed when scheme violated:

Validate authorization scheme: Once per session

You manage authorization for applications, pages, and most other components using Authorization Schemes.

You associate an Authorization Scheme with an application, page, or other component on the component's attributes page.

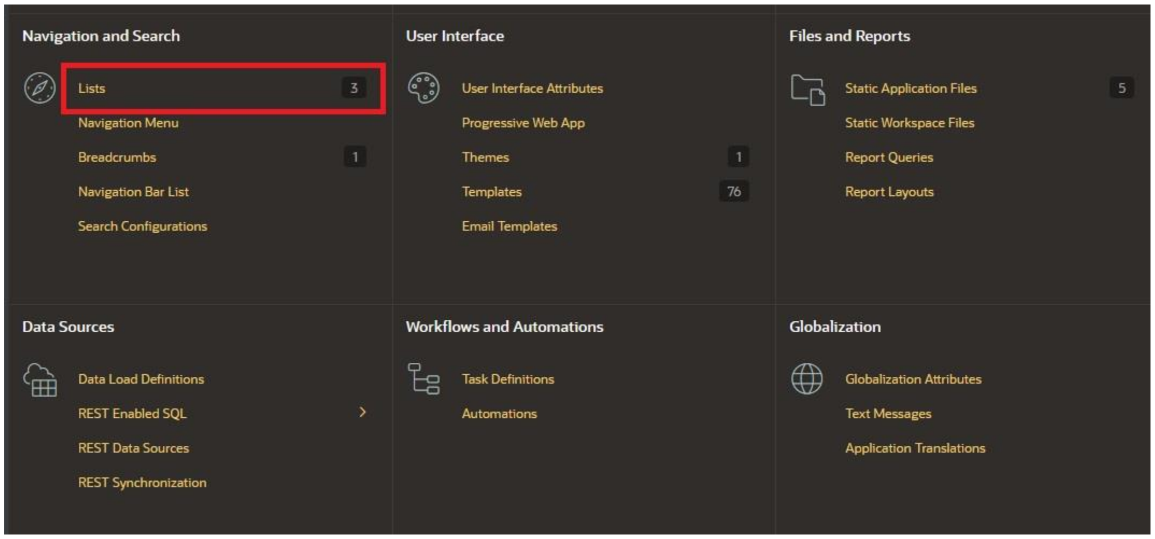
Authorization Schemes are pass/fail checks. Common Authorization Scheme types include Exists, Not Exists SQL Queries, and PL/SQL Function Returning Boolean.

To enhance performance, the success or failure of an Authorization Scheme is cached on a per session or per page view basis.

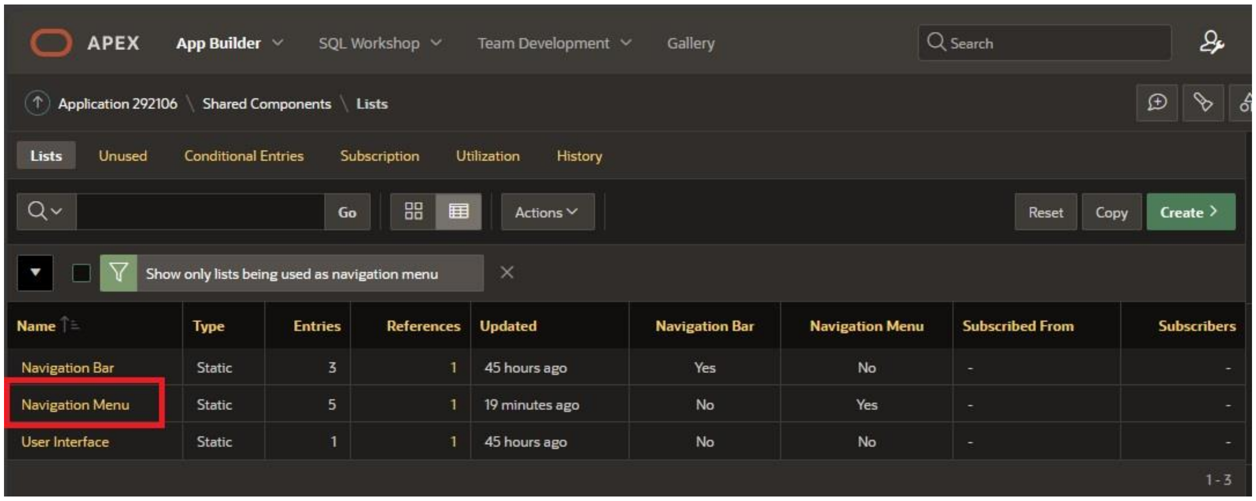
B. Adding the View Restriction on Page Level

Now that we have created the ***Check Admin*** authorization scheme, we will now apply it to the page navigation. To do so:

1. On the ***Shared Component*** page, go to the ***Navigation and Search*** section and click on ***Lists***.



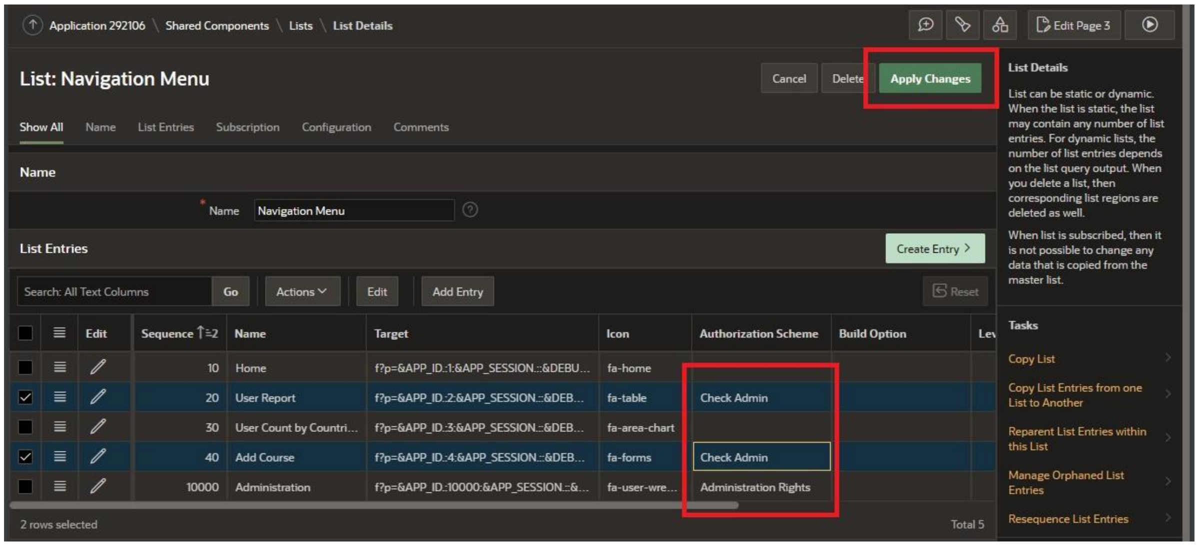
2. On the ***Lists*** page, click on ***Navigation Menu*** to view its details.



Name	Type	Entries	References	Updated	Navigation Bar	Navigation Menu	Subscribed From	Subscribers
Navigation Bar	Static	3	1	45 hours ago	Yes	No	-	-
Navigation Menu	Static	5	1	19 minutes ago	No	Yes	-	-
User Interface	Static	1	1	45 hours ago	No	No	-	-

1 - 3

3. From the ***List Entries***, change the ***Authorization Scheme*** for both ***User Report*** and ***Add Course*** to ***Check Admin*** by double-clicking on the empty space. Click on the ***Apply Changes*** to proceed.



List: Navigation Menu

Name

Name: Navigation Menu

List Entries

		Edit	Sequence	Name	Target	Icon	Authorization Scheme	Build Option	Lev
	≡	✎	10	Home	f?p=&APP_ID.:1:&APP_SESSION.::&DEBU...	fa-home			
✓	≡	✎	20	User Report	f?p=&APP_ID.:2:&APP_SESSION.::&DEB...	fa-table	Check Admin		
	≡	✎	30	User Count by Countri...	f?p=&APP_ID.:3:&APP_SESSION.::&DEB...	fa-area-chart			
✓	≡	✎	40	Add Course	f?p=&APP_ID.:4:&APP_SESSION.::&DEB...	fa-forms	Check Admin		
	≡	✎	10000	Administration	f?p=&APP_ID.:10000:&APP_SESSION.::&...	fa-user-wre...	Administration Rights		

2 rows selected   Total 5

List Details

List can be static or dynamic. When the list is static, the list may contain any number of list entries. For dynamic lists, the number of list entries depends on the list query output. When you delete a list, then corresponding list regions are deleted as well.

When list is subscribed, then it is not possible to change any data that is copied from the master list.

Tasks

Copy List

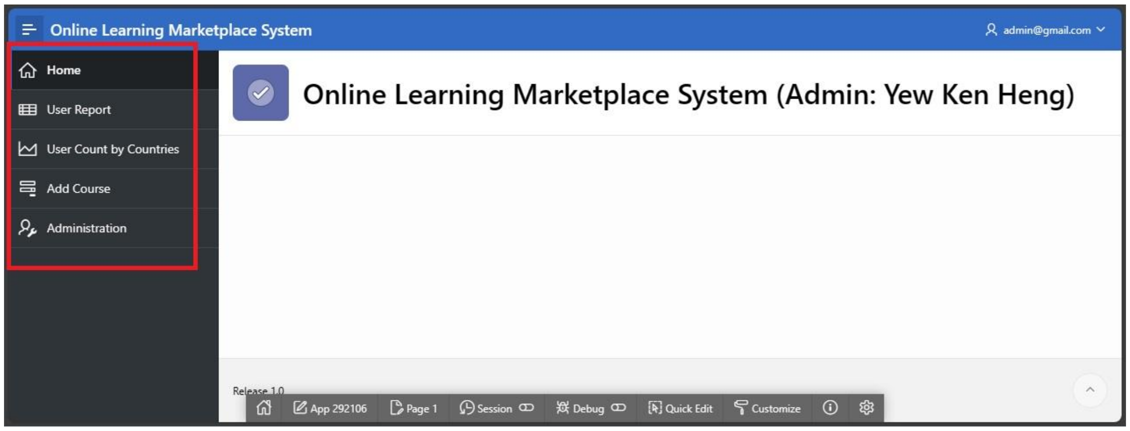
Copy List Entries from one List to Another

Reparent List Entries within this List

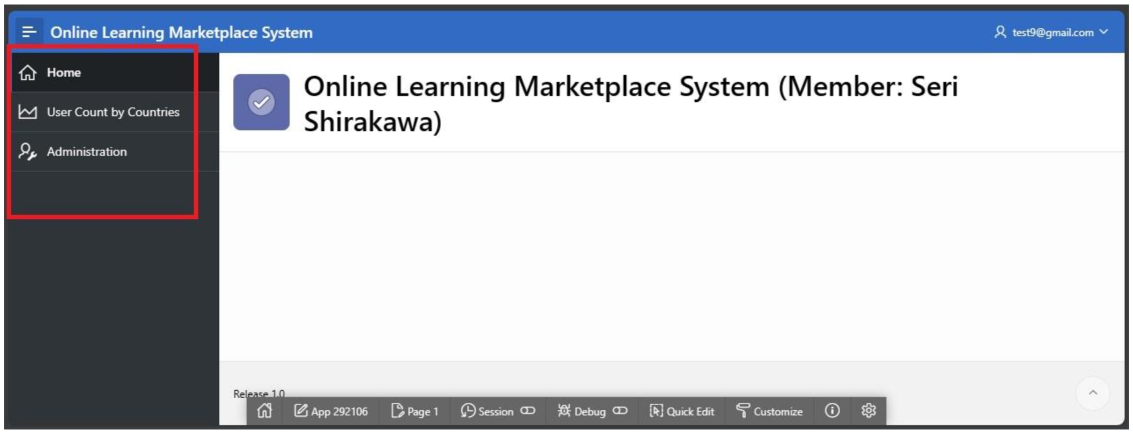
Manage Orphaned List Entries

Resequence List Entries

4. Run your application and log in with an admin account. You will see that both ***User Report*** and ***Add Course*** are still available in the navigation menu. However, if you log in with a member account, you will see that the navigation options for both pages are no longer available.



Online Learning Marketplace System (Admin: Yew Ken Heng)

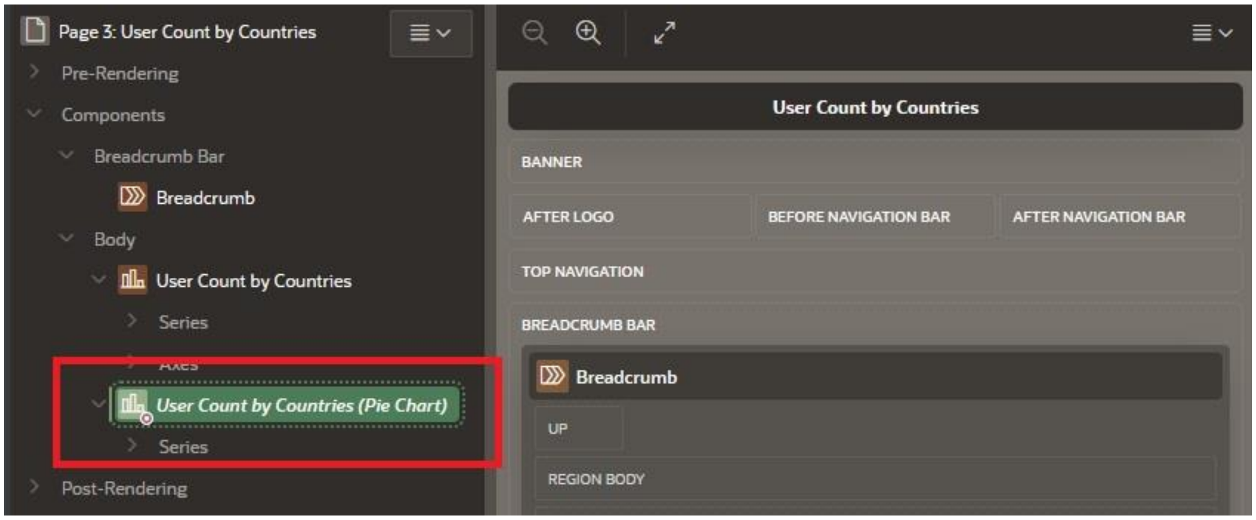


Online Learning Marketplace System (Member: Seri Shirakawa)

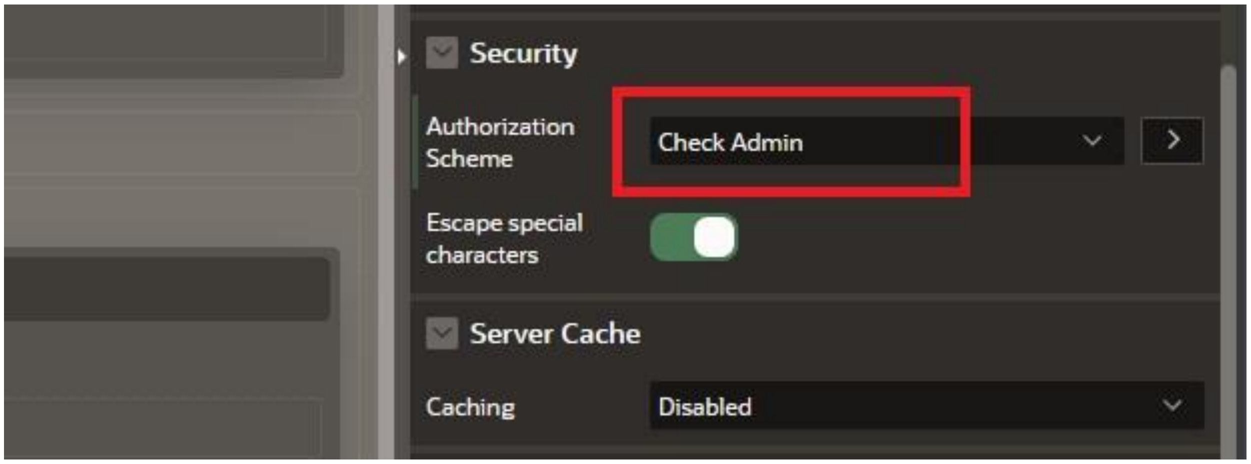
C. Adding the View Restriction on Component Level

Next, we will be using the ***Check Admin*** authorization scheme to apply a view restriction to a component. In our case, we will apply it on ***Page 3 - User Count by Countries***. To do so:

1. Go to the ***Page Designer*** of ***Page 3 - User Count by Countries*** and select the Pie chart (Lab 5 Exercise).



2. On the right-hand pane (***Property Editor***), scroll down until you find the Security section. Set the ***Authorization Scheme*** to ***Check Admin***. Save and run to apply changes to your application.



Security

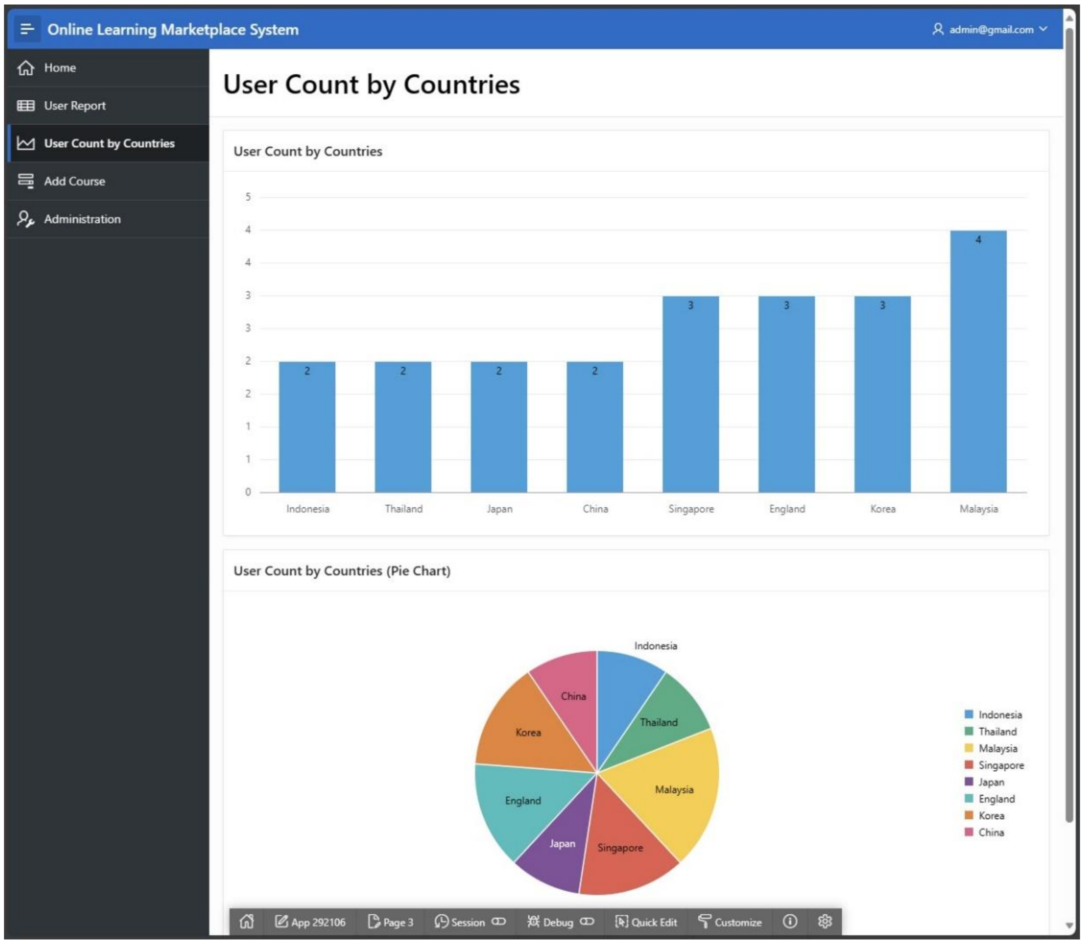
Authorization Scheme: Check Admin

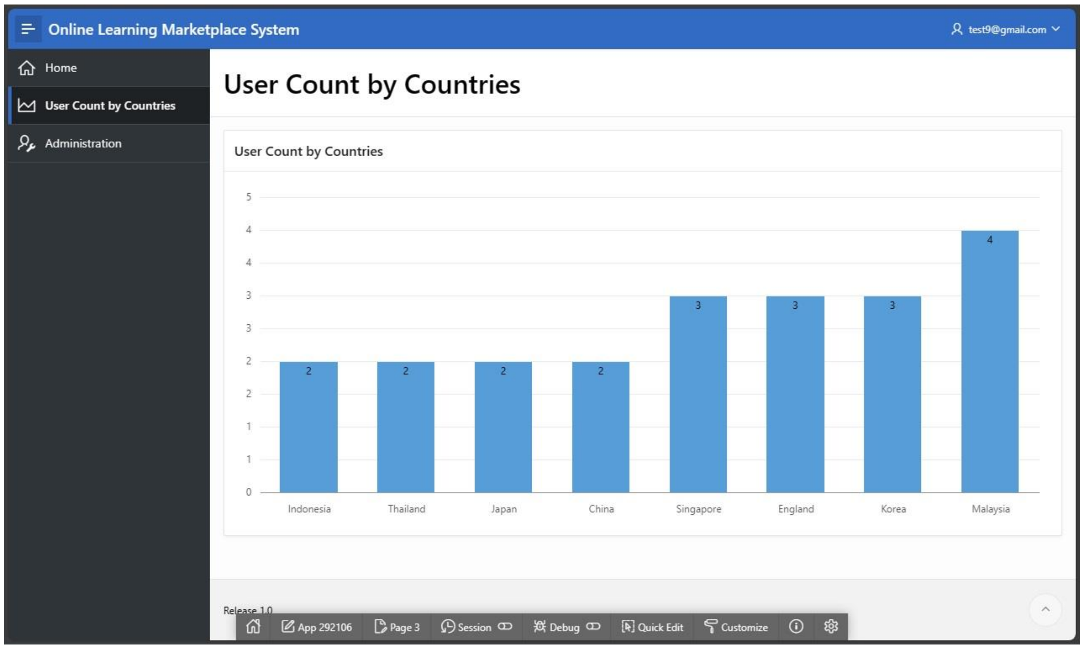
Escape special characters: On

Server Cache

Caching: Disabled

3. Log in to both admin and member accounts to see the difference. You will find that the Pie chart is only shown on the admin account instead of the member account.





VI. Exercise

Add validations for all attributes in the sign-up page. The instructions are as follows:

- Names should be alphabetical only.
- The email address should end with ***@gmail.com*** or ***@hotmail.com***.
- The length of the password must be 8 or above.
- Both nationality and profile picture cannot be left blank.

Capture screenshots of your solutions along with the error message output on the application's sign-up page. You are required to show sample inputs for both accepted and rejected cases.

VII. Lab Report Submission

In addition to the student's name and matrix number, the lab report must contain the following:

- solution for Q1 in Section II.
- solution for Q2 in Section IV.
- solution for Exercise in Section VI.

Submit your lab report on elearn before the due date. Lab submissions will be counted towards your coursework grade.

~~ END OF LAB 7~~