TRAINING GUIDE

Navigation and Logout

Smart Vendor & Procurement System
redSling UI Builder and Logic Builder

Create reusable Left Navigation and Top Navigation fragments, display the signed-in user's information, and configure page navigation and logout behavior.

1 Create the Left Navigation fragment

Open **Template Fragments** and select **Create New Template Fragment**. Name the fragment **Left Navigation** and open its editor.

2 Build the sidebar structure

Drag a **Div** component onto the blank canvas. Use it as the parent container for the sidebar. Set the required sidebar width and extend the container vertically across the navigation area.

Organize the sidebar into a top branding section and a lower section for navigation menu items.

3 Add application branding

Add an application icon or logo at the top of the sidebar. Add a label displaying the application name. Adjust spacing and alignment to balance the branding area.

4 Create the navigation menu

Add a **Row** containing an icon and a text label. Set the first label to **Dashboard**. Duplicate the row for the remaining menu items, updating each icon and label.

5 Style and save the fragment

Configure the font family, font size, font weight, padding, and spacing between menu items. Align icons and labels consistently. Review the branding, menu hierarchy, and sidebar layout, then save the fragment for reuse across application pages.

Build the Top Navigation

6 Create the Top Navigation fragment

Open **Template Fragments**, select **Create Template Fragment**, and provide a meaningful name such as **Top Navigation**. Open the Builder workspace.

7 Create the header container

Drag a **Row** onto the canvas as the main header container. Configure it to span horizontally across the page, with space for navigation controls and user information.

Adjust its dimensions, alignment, and spacing so that the header sections fit together and remain responsive.

8 Add the action area

Create a container toward the right side of the navigation bar for quick-access actions, such as shortcuts, notifications, or search. Add the required icons horizontally with consistent spacing.

Apply consistent icon sizing and alignment to keep the action area visually balanced.

9 Add the user profile section

Add an **Image** component for the user profile picture or avatar. Beside it, add labels for user information such as the name and designation.

Align the profile image and labels within the header. Review typography, spacing, icon alignment, and the visual hierarchy of the complete navigation bar.

10 Review and save

Check the header layout, action icons, profile section, and spacing. Save the Template Fragment. Insert the saved Top Navigation and Left Navigation fragments into application pages wherever they are required.

Display the Signed In User

Top Navigation logic

11 Open the component event

Open **TopNavigation** in the Builder and switch to **Logic Builder**. Select the component that displays the user information and configure the event that runs when the Top Navigation loads.

12 Retrieve and parse the login data

Property	Setting
Read stored data	Use Get from Local Storage with the key loginData
Store retrieved value	Assign the result to the variable loginData
Parse the string	Use Parse JSON from String with loginData as its input
Store parsed result	Assign the parsed object to parseLoginData

13 Extract and display the user value

Use **Get Property of** to retrieve the required user property from **parseLoginData**. Select the property that contains the information to display.

Use **Set Property** to assign the retrieved value to the selected component's **textContent**.

14 Connect and review the flow

Connect the blocks in this order:

1. Read loginData from Local Storage.
2. Assign the retrieved string to loginData.
3. Parse the string into a JSON object.
4. Store the object in parseLoginData.
5. Extract the required user property.
6. Set the display component's textContent.

Load the navigation and confirm that it shows information associated with the current login session rather than a fixed user name.

Add the Logout Button

Left Navigation layout

15 Add a row for Logout

Open the existing Left Navigation in **UI Builder**. From the **Blocks** panel, add a row below the existing navigation items. Use the **Layers** panel to locate the new row and position it at the bottom of the navigation area.

16 Add the button and label

Place a button inside the new row and change its label to **Logout**. Keep its appearance consistent with the existing navigation controls.

17 Add the sign out icon

In the **Icons** section of the Blocks panel, select the **Font Awesome Icon** component. Search for **out** and choose **fa-sign-out** under **Web App Icons**. Add it alongside the Logout label.

18 Refine the button appearance

Review the Layers panel to confirm that the icon and text are separate elements inside the button. Adjust the icon's typography, including size and weight, so it is proportionate to the label.

Adjust the Logout text size and color so it remains readable against the navigation background. In **Dimension** settings, set the button width and spacing to align with the surrounding layout.

19 Review and save

Confirm that Logout sits at the bottom of the sidebar, includes the sign-out icon and label, and aligns with the navigation. Save the updated fragment before configuring its Click event.

Configure Navigation and Logout Logic

20 Review existing navigation events

Open the Left Navigation in **Logic Builder**. Review the component Click events. Confirm that the **Dashboard** button navigates to the Dashboard page and the **Vendor** button navigates to the Vendor page. Review the destinations of the remaining navigation controls.

21 Configure the Logout Click event

Select the Logout button's **Click** event. Connect the storage cleanup and navigation actions in sequence.

Property	Setting
First action	Use Remove from Local Storage with the key loggedinuser
Second action	Use Remove from Session Storage for the relevant session information
Final action	Navigate to the Login page

Keep all three actions within the Logout Click event so that stored login state is removed before the user returns to Login.

22 Check the storage keys

The Top Navigation reads **loginData** from Local Storage, while the logout flow removes **loggedinuser**. Check these names against the values stored by the application's login flow. Configure logout to clear the relevant stored user and session information used by the application.

Select the relevant Session Storage key in the removal block to match the configured login session.

23 Validate the complete navigation

1. Confirm that the saved fragments appear consistently across pages.
2. Check the sidebar branding, menu alignment, header icons, and profile layout.
3. Verify that user information appears when Top Navigation loads.
4. Check that navigation buttons open their intended pages.
5. Select Logout and verify that the relevant stored login state is cleared before the Login page opens.