TRAINING GUIDE

Designing the Login Screen

Smart Vendor & Procurement System
Part 1 | redSling UI Builder

Customize the generated Login page, then connect its components to the authentication flow in the Logic Builder. The completed screen combines SVPS branding, styled login controls, and success and failure feedback.

1 Open and review the Login page

Open the Login page in the Builder. Review the generated Email and Password fields, Remember Me checkbox, and Sign Up action. Use the **Traits** panel to review placeholders, attributes, and API bindings. Review validation rules, field restrictions, and component security in the **Security** section.

Open the **Logic Builder** to inspect the existing authentication flow, variables, functions, conditions, and reusable logic blocks before changing the page.

2 Remove the default background

Select the main Login container. In **Manage Styles**, remove its existing background color and set the background to **transparent**.

3 Configure the project branding

Replace the project title with **Smart Vendor & Procurement System**. Add **Procurement & Vendor Management Portal** beneath it.

Property	Setting
Project title	Poppins; 32px; weight 700; line height 40px; #FFFFFF; left aligned
Project description	Poppins; 14px; weight 400; line height 22px; #D6E4F0
Left branding container	Background #0B3B5C; padding 40px; border radius 16px

Login Form Styling

4 Update the Login heading

Change the form heading to **Welcome Back**. Apply these typography settings.

Property	Setting
Font family	Poppins
Font size	28px
Font weight	600
Line height	34px
Text color	#1F2937

5 Style the Email and Password fields

Property	Setting
Width	100%
Background	#F5F7FA
Border radius	8px
Padding	12px vertical; 14px horizontal
Border	1px solid #D9E2EC
Font size	14px

6 Customize the primary button

Change the primary button text to **Login**. Set its background to **#0B3B5C**, text color to **#FFFFFF**, border radius to **8px**, height to **44px**, font weight to **600**, and width to **100%**.

7 Save and preview

Save the page and wait for the Builder to finish updating. Open **Preview** and check the branding panel, title and description, field alignment, button styling, and overall layout.

Configuring the Login Logic

Part 2 | redSling Logic Builder

With the interface designed, connect the page events to the authentication API and configure response handling. The logic script refers to the primary action as the **Sign In Button**; this is the action labeled **Login** in the UI guide.

8 Review component events

Switch from UI Builder to **Logic Builder**. Review the Sign Up click event that redirects to the Sign Up page. Confirm that the Email and Password input events capture the entered credentials into variables.

Property	Setting
Email input	Capture the entered email in userEmail
Password input	Capture the entered password in userPassword
Sign Up click	Redirect to the Sign Up page
Sign In Button click	Start the authentication request and response handling

9 Configure the authentication request

Open the primary button Click event. Configure the authentication API request with the following settings, and serialize the credential payload using **JSON.stringify**.

Property	Setting
HTTP method	POST
Endpoint variable	signin_api_url
Request header	Content-Type: application/json
Payload key email	Value from userEmail
Payload key password	Value from userPassword
Response variable	loggedInUser

10 Handle the successful response

In the **Success** branch, review the success logging and returned **loggedInUser** response. Extract **loggedInUser.token** into **generatedToken**. Store **generatedToken** in **Session Storage** using the key **jwt-token**.

Response Handling and Review

11 Prepare supporting authentication data

Use **Create Object with Key-Value Pairs** blocks to group the required base URL and authentication token. Convert the supporting objects using **Stringify JSON Object** and store them as configured in the existing flow for reuse in authenticated API requests.

12 Configure success and failure feedback

Property	Setting
Success notification	Message: User LoggedIn Mode: Success Position: Top Right
Failure branch	Execute Logic on Failure
Failure console message	Failed to login user
Failure notification	Message: Failed to login user Mode: Error Position: Top Right

Keep success and failure handling in their respective branches. On success, store the token, display the success toast, and continue with the configured application flow.

13 Verify the runtime environment

Open the **Runtime** screen and review the active test server, runtime URL, database connection, and available routes. Confirm that the backend environment required by the Login API is available.

14 Review the complete flow

1. The input events capture the email and password.
2. Clicking Login sends the credentials to the authentication API.
3. The response is stored in loggedInUser.
4. On success, the token is extracted and stored, supporting data is prepared, and a success toast appears.
5. On failure, the failure branch logs the error and displays an error toast.

Check the request variable mappings, Session Storage key, supporting object configuration, and toast settings against the configured flow.