

React Dashboard Portal

Create a dashboard portal with the following pages.

- 1. Login Page
 - a. Ask the user to enter data like email address and password
 - b. Use all developer-side test cases and validations
- 2. Dashboard Landing Page
 - a. Upon successful login, user should get redirected to this page.
 - b. This page should have a Sidebar and a Content Pane (See wireframe)
 - c. The sidebar will consist of two options.
 - i. User List
 - ii. Photos
 - d. When the user selects User List, the Content Pane should be populated with a table of records taken from the users JSON (See JSON source below)
 - e. When the user selects Photos, the Content Pane should be populated with a list of photo thumbnails from the Photos JSON (See JSON source below). The title of the photo should be printed under the photo. The Content Pane should be paginated so that no more than 10 thumbnails are loaded at once in the Content Pane.

For both the Login Page and Dashboard, use the following links to fetch dummy JSON data from:

Users: <https://jsonplaceholder.typicode.com/users>

Photos: <https://jsonplaceholder.typicode.com/photos>

Wireframes

The wireframes illustrate some example of what the Dashboard Landing Page might look like.

Dashboard Landing Page General Design

Sidebar	Content Pane
---------	--------------

Dashboard Landing Page When User List Is Selected

User List											
Photos	id	Name	Username	Email	Street	Suite	City	Zip	Phone	Website	Company
	1	Leanne Graham	Bret	Sincere@april.biz	Kulas Light	Apt. 556	Gwenborough	92998-3874	1-777-736-8031 x56442	hildegard.org	Romaguera-Crona
	2	Ervin Howell	Antonette	Shanna@melissa.tv	Victor Plains	Suite 879	Wisokyburgh	90566-7771	010-692-6593 x09125	anastasia.net	Deckow-Crist
	...										

Dashboard Landing Page When Photos Is Selected

User List	
Photos	<div>150 x 150</div> <p>qui eius qui autem sed</p> <div>150 x 150</div>

	beatae et provident et ut vel	1 2 3 4 5 6 7 8 9 10
--	-------------------------------	----------------------

Additional Requirements or Information

1. You should use redux for all API and central state management.
2. You can use functional or class components.
3. You should apply a decent user-friendly styling for all pages.
4. You should deploy and share the link of the assignment with the link of the repository for code too.
5. If you have any questions regarding the assignment or any doubts, please reach out to us.