

Dashboard App

A single page app in JavaScript, HTML and CSS.

No frameworks, no libraries should be used in the development of this project.

Cross browser compatibility is not requirement of this project.

Data

Use HTTP GET method to retrieve the data from the provided API.

Assets

Fonts

Font family and weights: Roboto Regular and Bold. Use google fonts.

Font sizes: Do not override the browser's default font size. Use relative font size units.

Icons

1. User icon in header: Material UI "person" icon. Download this icon in SVG format from <https://material.io/resources/icons/>. Use it inline in your code.
2. Navigation toggle button burger icon for use in mobile viewport: To create this icon use HTML and CSS only.

Images

All images are provided in the project folder.

Color codes

Color codes are provided in the **app.css** file.

UI sections and components

The main sections and components are:

- Header
- Main
- Nav
- Footer

The application should cover the full width of the browser window.

The application should be responsive and mobile friendly.

In all viewports the header and navigation components should be fixed to their positions as the user scrolls the page.

Screens:

1. Welcome screen
2. Colors screen
3. Users screen

Navigation:

By clicking the logo, the **Welcome** screen should appear.

By clicking the **Colors** option in the navigation component, the Colors screen should appear.

By clicking the **Users** option in the navigation component, the Users screen should appear.

The active navigation item should be highlighted in a lighter shade (provided).

App loading state

Till the page fully loads a spinner is present with a loading message in a white background covering the whole page at the length of 100vh and 100vw.

To create the spinner use HTML and CSS only.



Figure 1 Spinner at loading

The spinner will be also in the main section of the **Colors** and **Users** screens till the contents of each screen are rendered.

Welcome screen

When the page fully loads the spinner is removed and the welcome page appears.

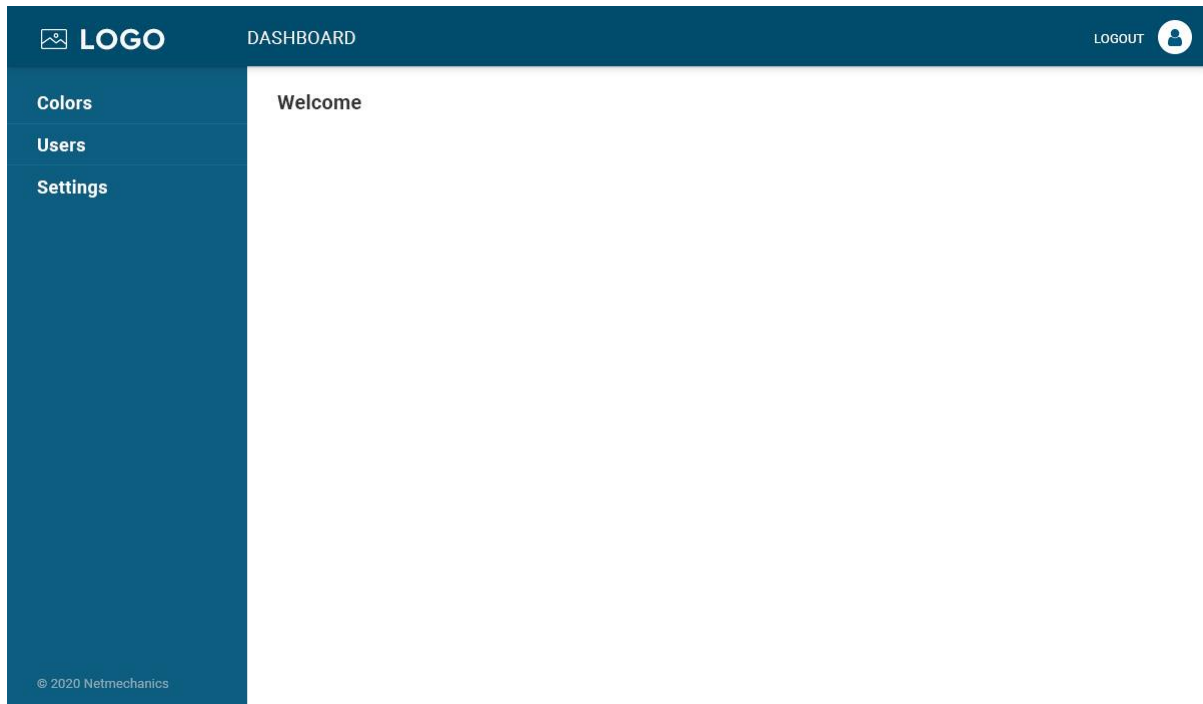


Figure 2 Welcome screen

Colors screen

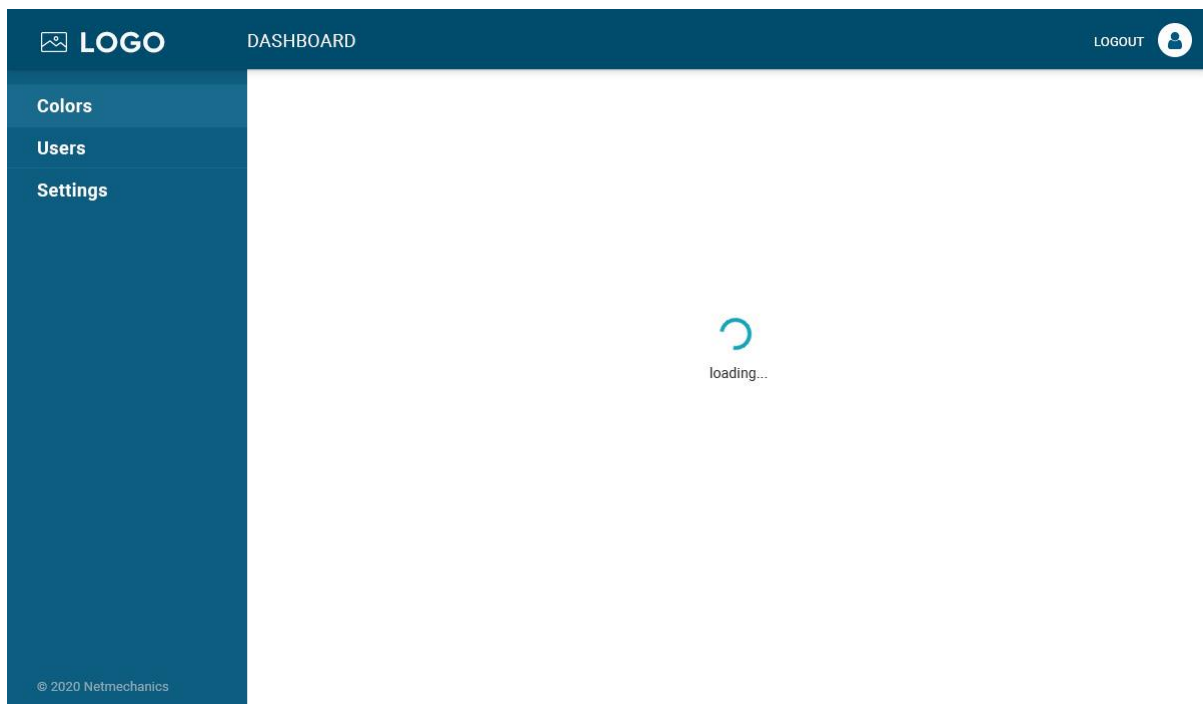


Figure 3 Colors screen loading state

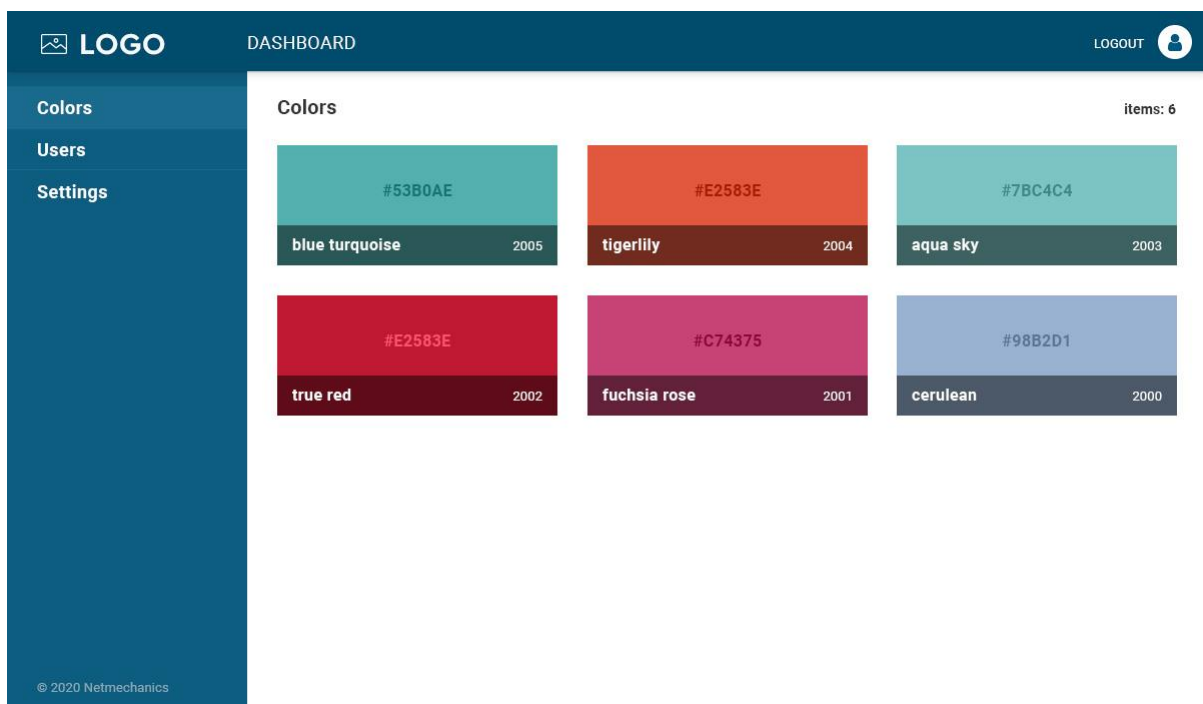


Figure 4 Colors screen rendered state

API request: <https://reqres.in/api/products/>

An items counter should be present in the top right corner of the main section.

The items should be sorted by year, most recent first.

Colors HEX code text should be lighter or darker by 60% of the original color.

All grid items are displayed in three columns with equals widths and a gutter of 30px.

Column widths should be adjusted at certain breakpoints.

The vertical distance between items is also 30px.

The color title has a black background with opacity of 50%. Opacity changes to a 100% when the user hovers over a color item.

Users screen

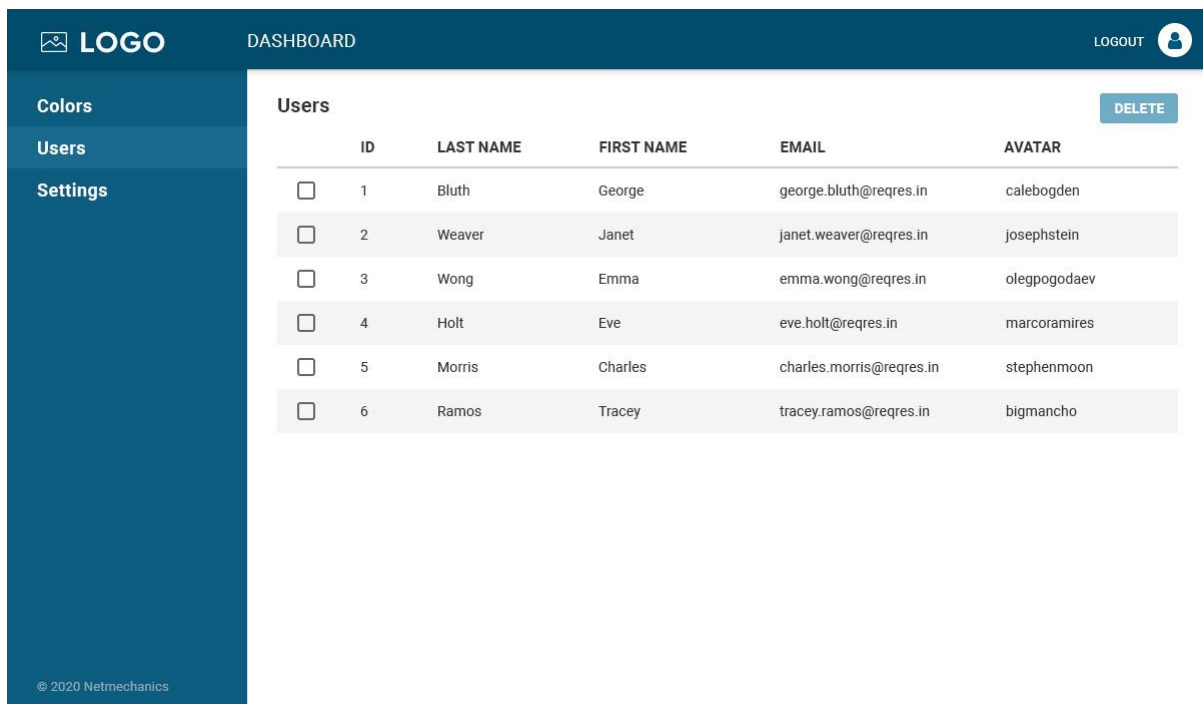


Figure 5 Users screen rendered state

API request: <https://reqres.in/api/users>

Users data should be rendered in table format, sorted by id. The table should be mobile friendly.

For table headers use the data item keys in capital letter as presented in figure 5.

For user avatars extract and display the avatar filename.

For every record, a checkbox should be added at the beginning of each table row.

At the right of the header a DELETE button is displayed in disabled state.

The button is enabled only when a table record is checked.

The user of the application should be able to delete a record from the table. The delete action should only be performed after user confirmation.

A deleted record should no longer be rendered while the user navigates the app (session). To achieve that use the Web Storage API.

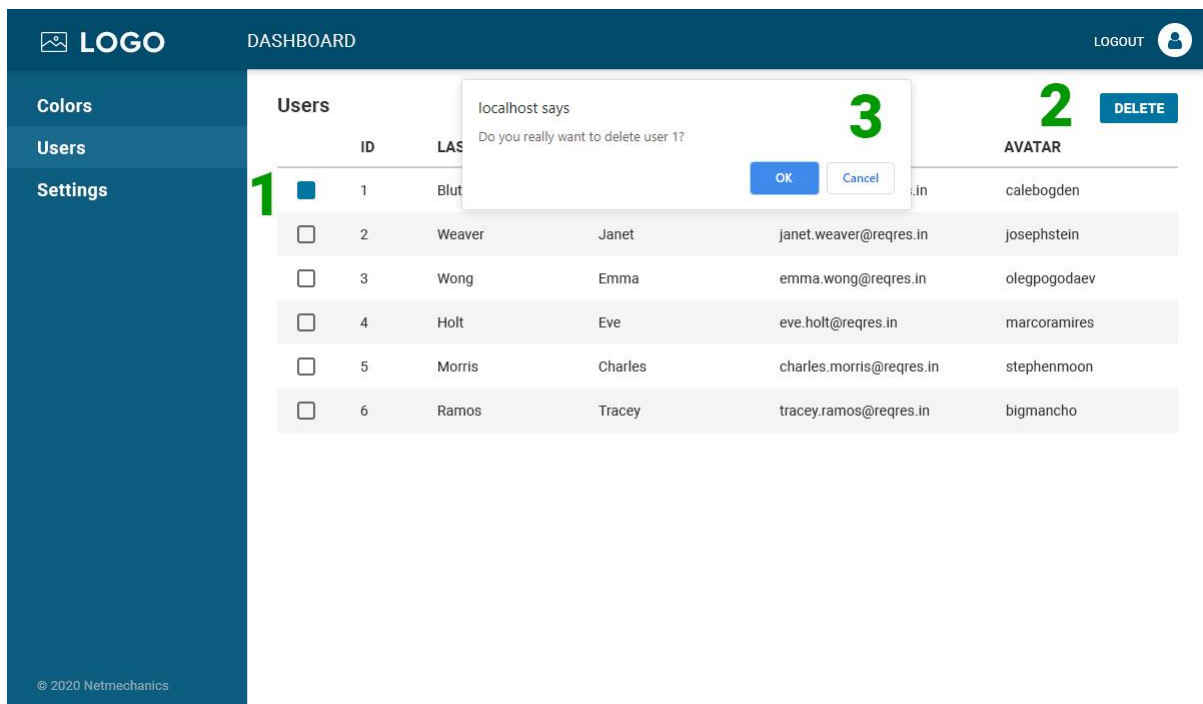


Figure 6 User delete actions sequence