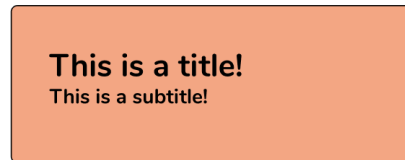


## Problem Statement

Build a UI for creating and configuring *Creatives*. A *Creative* is a container / box that contains a title and a subtitle, with a background color.



## Sample UI

[Mocks.pdf](#)

Font used in mocks: <https://fonts.google.com/specimen/Nunito> (Regular and Bold)

## API Sources

List of background colors: <https://random-flat-colors.vercel.app/api/random?count=5>

## Product Features

### Creative Dashboard

1. The dashboard loads with an empty list of *Creatives*.
2. An “Add Creative” button can be used by the User to open the *Creative Creation Drawer* (details below). **When the drawer is closed, the dashboard should take 100% width.**
3. The UI also provides a means to filter the list of *Creatives* via color **and** text.
4. List can be filtered using color and text (partial and case insensitive) simultaneously.

### Creative Creation Drawer

5. Clicking on the “Add Creative” button changes the layout of the page by opening a drawer. The following are some styling rules when the drawer is open:
  - a. The drawer takes up **35% of the screen width.**
  - b. The “Add Creative” button is **disabled when the drawer is open.**
6. A form needs to be made available in the drawer. A user can enter the Title and Subtitle for a *Creative*, and select a background color. **The list of background colors needs to be fetched via the provided API.** The following are some rules about the form:
  - a. All three fields are required fields. **It is mandatory for the user to fill them.**

- b. The *Done* button on the form is **disabled until all the form fields are entered by the user.**
7. On clicking on *Done*, the drawer automatically closes. The *Creatives Dashboard* updates with the following:
  - a. The *Add Creative* button is **enabled as the drawer is closed.**
  - b. The newly created *Creative* is rendered as a *Preview* on the list. A *Preview* should be a rectangle whose background color is what the user selected in the form, with the title and subtitle information entered by the user inside it.
8. The drawer can also be closed by clicking on a “X” (close) button.

#### Good To Have:

- A limit of five *Creatives* are allowed.
- The Progress bar shows the number of *Creatives* created, both numerically, ie (1/5), and visually via a progress meter **that transitions to show progress.**
- Users should be able to multi-select colors in filters.
- Only the left side scrolls when the creatives preview crosses the viewport.
- Transition animation while opening / closing the drawer.
- Close drawer on esc key press.
- Persist the creatives

#### Points to consider:

- **The solution should be coded in vanilla JavaScript and CSS.** Please refrain from using any plugins / libraries like jQuery / React / Angular or css frameworks like Bootstrap. However, **you can use tooling such as webpack, grunt, gulp,** etc.
- Focus on the modularity of code and design of the solution. Keep the performance of the application in mind.
- The final solution should work without errors.
- Ensure the solution’s look and feel matches the mocks as much as possible.
- Focus on using the right semantic html tags. **Using the right HTML tag may be faster than building your own implementation!**

#### You will be evaluated based on:

1. Correctness and completeness of the solution, and understanding of the problem statement.
2. Code design and quality.
3. Visual aesthetics and your ability to extract styling and UX information from a mock. (the UI should be as close as possible to the given design)