Colors Challenge

Preface

The intent of this challenge is to explore general system design, along with approaches to code extensibility. While a simple challenge, we wish to stress that extensibility is important, please think about how easy it would be for others to take your work and add new elements to it.

Short Description

- Build a back-end to generate color swatches via an API in various color spaces.
- Build a front-end to consume the back-end API and render the color swatches.
- Maintain a focus on extensibility in your code.

Long Description

You've been working on a number of projects within a company across a few teams and you've found that the teams have a shared requirement for easily viewing "color swatches" using different color spaces.

We'd like to produce a service that could be easily used, and also easily extended, by all the teams in the company. The service should be able to render a "swatch" of five colors, each color being potentially represented in a different color space (RGB, HSL, etc).

The back-end should provide an API capable of returning to the front-end a set of five different colors for each call.

Each color should be in a random color space; e.g. if the back-end supports RGB and HSL, then in one call the back-end may return something like *(note that your payload structure may be different)*

```
[
  {
    "type": "rgb",
    "red": 255,
    "green": 255,
    "blue": 255
  },
  {
    "type": "hsl",
    "hue": "360",
    "saturation": "100",
    "lightness": "100"
  },
  // Three more colors in random combinations of RGB and HSL*
]
```

The front-end should consume this API, rendering each of the five colors in a swatch to the user. The user should be able to regenerate the swatches by clicking a button.

Stage 1

We know for sure that we will be using RGB and HSL color spaces. Please implement your service supporting RGB and HSL color spaces.

Stage 2

After we delivered the solution to stage 1 to the teams, we found that one of the teams uses a strange representation of the RGB color space, BRGB. BRGB stores the red, green, and blue components as values between 0 and 10000 (inclusive). The team has asked about how they can easily extend your implementation to include the BRGB color space.

Please write a how-to for any team to be able to add new color spaces themselves. Have you designed the system in such a way to make it easy and straightforward to extend?

General notes

- Ideally, you can develop the solution with Nextjs entirely both API and UI, but feel free to use any other technologies you're more comfortable with. Only Frontend must be React.js
- Please don't spend more than 3-4 hours on it. If you can't get to do everything, please write down how you would approach adding any missing functionality.