Involved technologies

- Node.js
- Express.js
- React
- Redux
- SLDS (Recommended material: https://trailhead.salesforce.com/en/modules/lightning_design_system)

Project requirements

- Implement a simple server using Node and Express.
- Must include a basic task runner setup (Gulp or Webpack)
- The client side must be built using React and SLDS.
- The navigation on the page must be implemented using React Router.
- All code artifacts must be under version control (Git)

The homepage should look like this.

1. The "Welcome Willard" and the Trailhead logo images should be copied from [Trailhead Website](#) (you need to be logged in to see the first one).
2. The header must have 2 tabs, but you only need to implement the "Content" one.
3. The user name and image URL should be passed as props.
4. No functionality required on the "Trailhead/Logout" dropdown, just the UI.
5. No functionality required on the "Edit/preview/export" dropdown, just the UI.
6. On the Add button dropdown, mock as the image and only implement the "Add trail" linking to the *Trail Edit page* and creating a trail into the store.
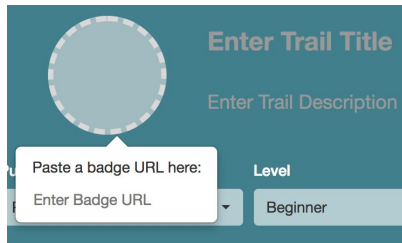
Trail Edit page.

Image 1. Trail edit page. Empty trail.

1. The header looks like the one on the homepage, except for the tabs, they are not displayed here and the "Trailmaker" text that's added (ignore the differences on the avatar image).
2. There are 3 inputs here:
   a. Title
   b. Description
   c. The circle next to the title and description.
3. When the user enters data on the inputs, the page should looks like the image 3.
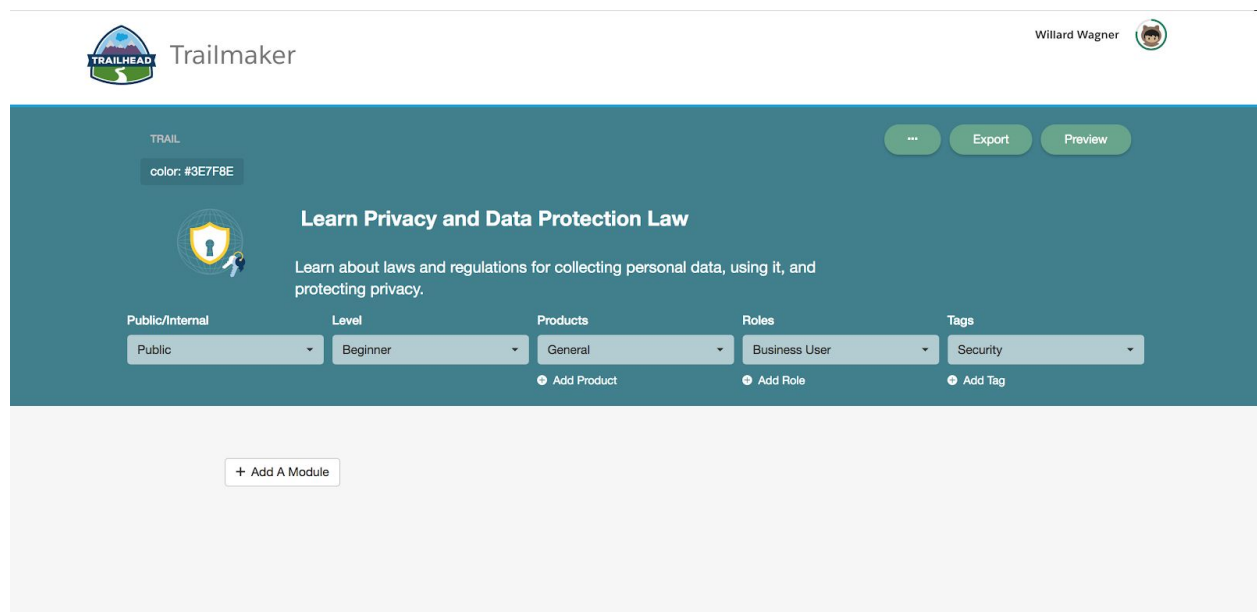4. When clicking on the circle, a popover should appear to allow the user to paste a url to an image. The popover should disappear when the user clicks anywhere.

Image 2. Badge URL input.

5. The dropdowns show information passed as props (1 array of strings for each dropdown).
6. No functionality needed for the buttons, only UI.

Image 3. Trail edit page. Full trail.

The initial state must be like this:

```
{
    content: [],
    subcontent:[],
    trails:[]
}
```