

This is a project to be used by students to build out User Interfaces solely with functional components in ReactJS

Edit

Add topics

23 commits

2 branches

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is 1 commit ahead of LambdaSchool:master.

Pull request

Compare

Erik Kimsey First commit

Latest commit 1082cc1 38 minutes ago

Day-I-Social-Card	First commit	38 minutes ago
Day-II-Calculator	adds necessary instructions and files for calculator	2 months ago
README.md	adds necessary instructions and files for calculator	2 months ago
calculator-component.png	implements some details on calculator	2 months ago
sc-component-outline.png	adds more to the Readme	2 months ago

README.md

React-UI-Components

- This repository is designed to be your first exposure into the world of ReactJS. You will be given 2 days to complete the 2 assignments in this repository. Day 1 is all about implementing a Social Card in ReactJS. On Day 2 you'll be implementing a Calculator. We've given you the file structure and have gone ahead and added all the files you'll need to be set up for success for each project.

Initializing the application.

- This project was put together using `create-react-app` (CRA). **You will not need to install CRA in order to make this project work**
- Each project has its own `package.json` file in it, we'll chat more about this later. So it's already set up for you to install some dependencies that are needed for you to be able to work within the React Ecosystem.
- To start the `Social Card` project, you'll need to `cd` into `Day-1-Social-Card` and then into `social-card` and run `yarn install` to retrieve all the dependencies.
- Inside of `../..social-card` you'll then need to run `yarn start` to open up a React Development Server that can take your `.js` files as components and bundle them up to work in your new environment.
- You'll repeat this last two steps for the calculator project, but you'll need to make sure that you're inside of `../..calculator` directory to make this work.**

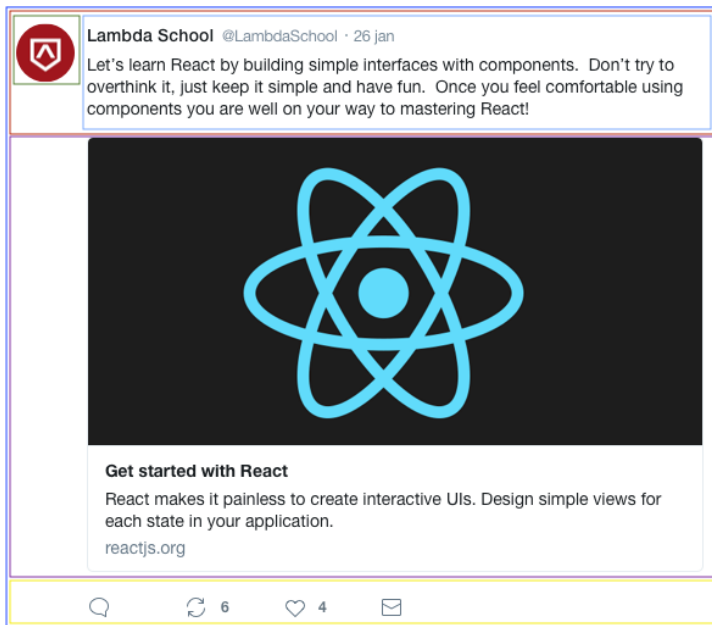
Instructions

- Over the next two days you'll be working on two different projects in this repo.
 - The first day you'll work on the project found in `Day-I-Social-Card`.
 - The second day you'll work on the project found in `Day-II-Calculator`.
- Use the design files to build out your User Interfaces.
- All components can be built out using the provided html files found in their respective directories.
- Each file has been set up to work within a `react.js` environment. Pay attention to the notes found within each project.
- Feel free to include a `cdn` for bootstrap v.4 if you'd like in your `index.html`.
 - This should include bootstraps style sheet globally for all of your components.
 - Don't forget about `className` vs `class` on your `JSX` elements !!

Day I - Social Card

- We're going to break down this assignment in terms of how you should be starting to think in react. Remember that everything is a component.
- All the files you'll need for this project are found in `Day-I-Social-Card/social-card/src/components`. You can find all the component files you should need in their respective directory. i.e. inside of the `HeaderComponents` directory you should see a few `.js` files and a `.css` file.

- Feel free to add any files for any extra components you may feel the need to build.
- Any of the styles you write in your respective `.css` file should be available for your the components where the `.css` file is being imported.
- We have drawn boxes around possible areas that could be components.
 - The outer box will represent the `App.js` file.
 - The red box around the header could represent the `header` directory with a few nested components inside, the thumbnail image and the header content.
 - You could go so far as to break down the header content into a header and body component.
 - The next box around the social card represents the React Banner image and some copy found underneath.
 - And then of course the footer (**Stretch Problem 1**) which contains your icons, could be a container for all of your icon components.



Day 1 MVP requirements

- Create a `<HeaderContainer />` container component that will hold your header components.
 - Create an `<ImageThumbnail />` component using this image url <https://ibin.co/3whrpKSBbZ81.png> as it's img src.
 - Create a `<HeaderTitle />` component that displays the Lambda School header text, @LambdaSchool handle and timestamp.
 - Create a `<HeaderContent />` component that displays the copy provided in the headers content.
- Create a `<CardContainer />` container component that will hold your card components.
 - Create a `<CardBanner />` component that will display this image as it's background: <https://ibin.co/3wnC6SgIOJud.png>
 - Create a `<CardContent />` component that displays the card copy provided.
 - The entire `<CardContainer />` should take a user to <https://www.reactjs.org> when clicked.

Day 1 Stretch Problems

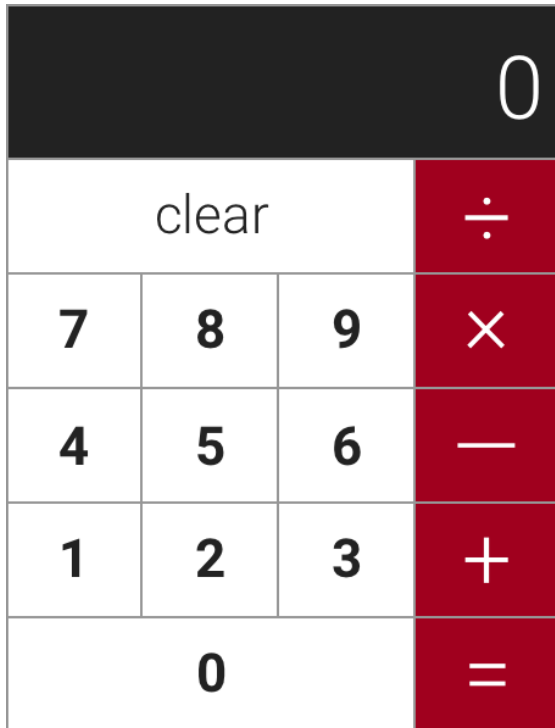
- Create a `<Footer />` component that pulls in the icons and displays them properly.
 - Build out the functionality so that a user can click on the icons and have them react to events.
- Ensure that your product is as pixel perfect as possible using any tools that you were introduced to in previous weeks.
- Look up the `moment.js` library and figure out how to format your `time-stamp` in your header to be today's date.

Day II - React Calculator

- For this project you're not going to be given any tips on how to break down the image file into components. Now that you've had some practice with the social card, this should be something you can start doing on your own. Just think about the image, and what potentially looks like a component.
- For the MVP the calculator just needs to be displayed properly. Functionality will be a part of the stretch requirement.
- All the files you'll need for this project are found in `Day-II-Calculator/calculator/src/components`. You can find all the component files you should need in their respective directory. i.e. inside of the `ButtonComponents` directory you

should see a couple `.js` files and a `.css` file.

- Feel free to add any files for any extra components you may feel the need to build.
- Here is what your calculator should look like:



- **BEFORE YOU DO ANYTHING ELSE READ [This](#)**
- This is the time to stop and break down this image in terms of components. Each container on the screen should give you an idea of what your components should be. You're now about to begin your journey into learning how to think in React.

Day 2 MVP requirements

- Create a `<NumberButton />` component that can accept `props` and display any number/symbol passed down as `text`.
 - Example your component should be able to render a dynamic prop called `text`:
 - Your button should also be able to accept dynamic props `buttonStyle` for styling

```
<button className={props.buttonStyle}>{props.text}</button>
```

- Create an `<ActionButton />` component that will be used for the `zero` character and the `clear` button.
- Create a `<CalculatorDisplay />` component that will be used as the `calculator display`

Day 2 Stretch Problems

- Re-factor your `App.js` file to be a classical component that can hold state on its constructor. (There is some documentation in training kit on how this works).
 - On your state object be sure to include a property for the `total` that can be passed down to your `<CalculatorDisplay />` component.
- Create some `handler functions` that can take in some information from an `onClick` and use that information to update the `total` on the App state.
 - `this.setState` will be your best friend here :)

GOOD LUCK!