

Assignment 8 — React

Due Nov 22 by 11:59pm **Points** 11 **Submitting** a file upload

Available Nov 14 at 8am - Nov 24 at 11:59pm

This assignment was locked Nov 24 at 11:59pm.

Introduction



In this assignment, you'll write a client-side application using React.

Be sure to periodically review [Assignment 8: Tips, FAQs, Corrections thread](https://edstem.org/us/courses/29198/discussion/2155438)  **(<https://edstem.org/us/courses/29198/discussion/2155438>) in the Ed discussion board (pinned at the top)**

Learning Outcomes


- What is the React framework? (Module 8 MLO 1)
- What are destructuring expressions in JavaScript and how do we use them? (Module 8 MLO 2)
- What is JSX and how do we define components using JSX? (Module 8 MLO 3)
- What is the State Hook and what are its uses? (Module 8 MLO 4)
- How can we write Single Page Applications (SPAs) using React Router? (Module 8 MLO 5)
- How can we define and use forms in React apps? (Module 8 MLO 6)

Instructions

Develop a web app as a Single Page Application (SPA) using React and using React Router for routing between pages. The website will have a navigation section and will include the following 3 pages:

1. Homepage.
2. Order Page.
3. Registration Page.

Data for the App

- We have provided you a zip file with starter code [assignment8.zip](https://canvas.oregonstate.edu/courses/1890443/files/96065190?wrap=1) (<https://canvas.oregonstate.edu/courses/1890443/files/96065190?wrap=1>)  (https://canvas.oregonstate.edu/courses/1890443/files/96065190/download?download_frd=1) .
- Instead of creating a new React app, download the zip file, unzip it into a directory and then run `npm install` in the directory where you have unzipped the file.
- The zip file has the boiler-plate code for a React app with the following changes and additions
 - The `package.json` file already includes the dependencies `react-router-dom` and `react-icons` .

- The `App.js` file has been modified to import data from the data file that we have added.
- The data your app will use is in the file `items.js`.
 - This file is in the directory `src/data`.
 - Don't change this file.
- The `App.js` file imports data from this file into the variable `items`.
 - Your app must get its data from this variable.
 - Do not change the part of `App.js` that imports this file.
- During testing, we may replace the data file with a different test file with the same name.
 - The properties of the objects and the type of their values in the test file will match the properties of the objects in the file provided to you.
 - However, the values of these properties as well as the number of objects in the test file can be different from the file provided to you.

Create Additional Files

- You must create a separate file for each *page* and *React component* you define.
- Inside the `/src` folder, create a `/pages` folder and in this folder create three files for the 3 pages of your app (you can choose different names for your components, and hence the files)
 - `HomePage.js`
 - `OrderPage.js`, and
 - `RegistrationPage.js`
- Inside the `/src` folder, create a `/components` folder and in this folder create one file for each of the React components (you can choose different names for your components, and hence the files)
 - `OrderTable.js`
 - `OrderRow.js`
 - `SelectQuantity.js`
 - `RegistrationInfo.js`
 - `Navigation.js`
- Note:
 - You are not required to use the same names for the React components as we have listed above. The names have been given as an example.
 - You can also define additional React components if you want.

1. Homepage

- This page renders when the app starts up.
- This page must include:
 - A heading level 2 `<h2>` to introduce the page.
 - A paragraph `<p>` to tell users what to do.

2. Order Page

This page must include:

- A heading level 2 `<h2>` to introduce the page.
- A paragraph `<p>` to tell users what to do.
- A `<table>` that displays data from the variable `items`. This HTML table will be created by a React component (`OrderTable` in our example) and will include:
 - A `<thead>` row that specifies these three columns for an item's:
 - Name
 - Price
 - Quantity
 - Rows of data that render in the `<tbody>`.
 - Each row of data must display the name, price and quantity of the item, and must be created by another React component (`OrderRow` in our example).
 - Use the map function to create these row components. There should be one row component for each item in the variable `items`.
 - Within each row, the quantity value must use another React component (`SelectQuantity` in our example) that provides a control via 2 icons to increment and decrement the quantity of the item in the row.
 - The initial value of the quantity must be zero.
 - The user must not be able to set the quantity to less than 0 or greater than 10.
 - You can use **Exercise Counter Component** in [Exploration - State and React Hooks](https://canvas.oregonstate.edu/courses/1890443/pages/exploration-state-and-react-hooks) (<https://canvas.oregonstate.edu/courses/1890443/pages/exploration-state-and-react-hooks>) as a starting point for this component.

3. Registration Page

This page allows the users to register to get promotions by email. This page must include:

- A heading level 2 `<h2>` to introduce the page.
- A paragraph `<p>` to tell users what to do.
- A React component (`RegistrationInfo` in our example) that creates a `<form>` with `<fieldset>`, `<legend>`, and controls for user to input data (described below), and a `<button>` to submit the data.
- The input controls must include
 - A `name` input control
 - An `email` input control
- When the user hits the submit button, don't submit the form. Instead show an **alert** (or other dynamic message) that displays the response message, which must include the data entered by the user.

Navigation Section Using a Navigation Component

Add a navigation section by creating a React component called `Navigation` that supports navigation to all 3 pages using the `Link` component.

CSS

Optionally you can update and add rules to the existing `App.css` file that resides in the `/src` folder.

Technical Requirements

- Your React components must be function-based. You are not allowed to define class-based components.
 - In the explorations we studied function-based components. We are adding this requirement in case anyone has prior experience of React using class-based components and may have ended up using them for the assignment.
- You cannot use React based frameworks, such as Next.js. In general, don't use any React functionality that we haven't covered in this module without the instructor's permission.

Hints

- **Exploration — React Components & JSX**
(<https://canvas.oregonstate.edu/courses/1890443/modules/items/22518226>) shows an example of a React component to display an array of data. Think about how you can adapt that example to create an HTML table.

What to Turn In?

- Submit a single zip file with all of the React folders and files (**except for the `node_modules` folder**). Include the `package.json` file.
- Name the file: `youronid_assignment8.zip` where `youronid` should be replaced by your own ONID.
 - Don't worry about a filename change if Canvas adds a version number when resubmitting.
- The grader will unzip your file, go to the root directory, run `npm install` and then run `npm start` to start your application and test it.

Video of a Sample Solution

Here is a video of a sample solution for Assignment 8.