ITMD 465/565

Lab 5

Introduction

In this lab we will be experimenting with some React core concepts like elements, components, and JSX.

Objective

The objective is to build a React class and functional components with and without JSX.

Requirements/Process

- Use the supplied html files as the start of the lab.
- The supplied starter files have React, ReactDOM, and Babel CDN links
- Uses in browser prototyping method for Babel
- Make sure you are looking at your browser's dev tools console for errors if you are having issues.

Part 1

- Use supplied file lab5.1.html for this part.
- **Do not use any JSX for Part 1.**
- Create a stateless functional React component called **Headline**, as shown in class.
- The **Headline** component when rendered on the page should be an h1 tag with the following text inside:
    - ITMD-4/565 Lab 5 Part 1 – *your name here*
- The Headline element should have a prop called **name** that you use to display the text in the element instead of putting your name in the element directly.
- Using ReactDOM render the **Headline** component to the page in the div with id of app. Pass your name in as a prop when you create the element. **No JSX**, you must use the createElement method to create the React elements from the components without JSX syntax.

Part 2

- Use supplied file lab5.2.html for this part.
- **You can use JSX for Part 2. Use stateless functional components for Part 2.**
- You will be creating four stateless functional React components for this part. These components will form a product listing when used together. The basic definitions are done for you in the starter file.
- Here are the component names: **ProductImage, ProductName, ProductDesc, ProductDetail.**

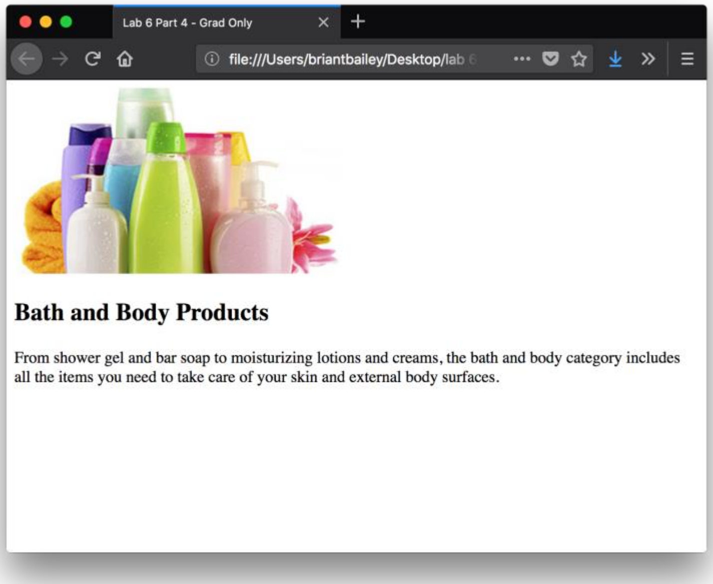
- At the top of the script section a const named **productInfo** is defined for you and it contains a JavaScript object that represents data about a single product. Do not use this object or its values in the individual components directly. It should only be passed to the **ProductDetail** component as a single prop named **product** when you use the **ProductDetail** component.
- The **ProductDetail** component will be a wrapper around the other components. You should return a div element with the three other components nested inside. The order should be image, name, desc.
- The **ProductImage** component should return an img tag. The src attribute on the img tag should be passed into the component as a prop named **imgsrc**.
- The **ProductName** component should return a h2 tag and the h2 tag should have the product name text inside. The product name text should be passed into the component as a prop named **name**.
- The **ProductDesc** component should return a p tag and the p tag should have the product description text inside. The product description text should be passed into the component as a prop named **desc**.
- Pass the **productInfo** object representing the single product to the **ProductDetail** component as a prop and then inside the component use the props object and its individual values to pass props to the other components when you use them inside the **ProductDetail** component. Pass data from parent to child components as props.
- Using ReactDOM render the **ProductDetail** component to the page in the div with id of app.

Part 3

- Use supplied file lab5.3.html for this part.
- **You can use JSX for Part 3. Use a class-based component for Part 3.**
- Create a single class based React component called **ProductDetail**.
- At the top of the script section a const named **productInfo** is defined for you and it contains a JavaScript object that represents data about a single product. Do not use this object or its values in the **ProductDetail** component directly. It should only be passed to the **ProductDetail** component as a single prop named **product** when you use the **ProductDetail** component.
- The **ProductDetail** component should return a div tag with 3 child tags. The child tags should be an img, h2, p.
- Pass the **productInfo** object representing the single product to the **ProductDetail** component as a prop and then inside the component use the props object and its individual values for the img src, product name, and product desc. Do not pass separate props to the **ProductDetail** component, pass in the one object. Do not use the data values from the const directly in the component, it must be passed as a prop called **product**.
- When rendered this should look identical to Part 2, use the same html elements as described in Part 2.

- Using ReactDOM render the **ProductDetail** component to the page in the div with id of app.

It should look like this in both part 2 and 3.



Bath and Body Products

From shower gel and bar soap to moisturizing lotions and creams, the bath and body category includes all the items you need to take care of your skin and external body surfaces.

Graduate Additional Requirements

If you are involved in **any section of 565** you need to complete the additional requirements listed here.

You must do an additional part 4. In part 4 you will do the exact same thing you did in part 3 but **without using any JSX**. Do all the createElement function calls yourself.

README File

No additional readme files other than the ones described in the lab directions.

Due Date / Late Policy

This assignment is due on **Friday April 17, 2020 11:59 PM Chicago Time**. See syllabus for full late policy. **No Extensions**.

Submission Guidelines

You must upload your submission, to the blackboard assignment by the due date. The submission must be in the following format and structure. If you do not submit your assignment exactly as specified, you will receive an immediate 5% deduction.

Submission Format Specification:

Zip your files and upload to the blackboard assignment. Please include the image file in your zip file.

YourUsername_lab5.zip