



React To-do Planning Assignment

Deadline: Monday, JUL 27 2020, 09:00 AM

GitHub Classroom Link: <https://classroom.github.com/a/PnUvnLHF>

REMEMBER TO SET REPO TO PRIVATE UNTIL AFTER DUE DATE

Introduction

This assignment is meant to prepare you to think in terms of component based designs. Your goal in this assignment is to come up with a **detailed diagram** that breaks down all the components necessary to create a to-do app in React.

Requirements

- ☐ Components are broken down into reusable segments
- ☐ Components have a detailed description of how each component will function (pseudocode)
- ☐ Arrows are used to show how each component is linked to one another to create a fully functioning application
- ☐ Image of diagram is uploaded to your repository for this assignment

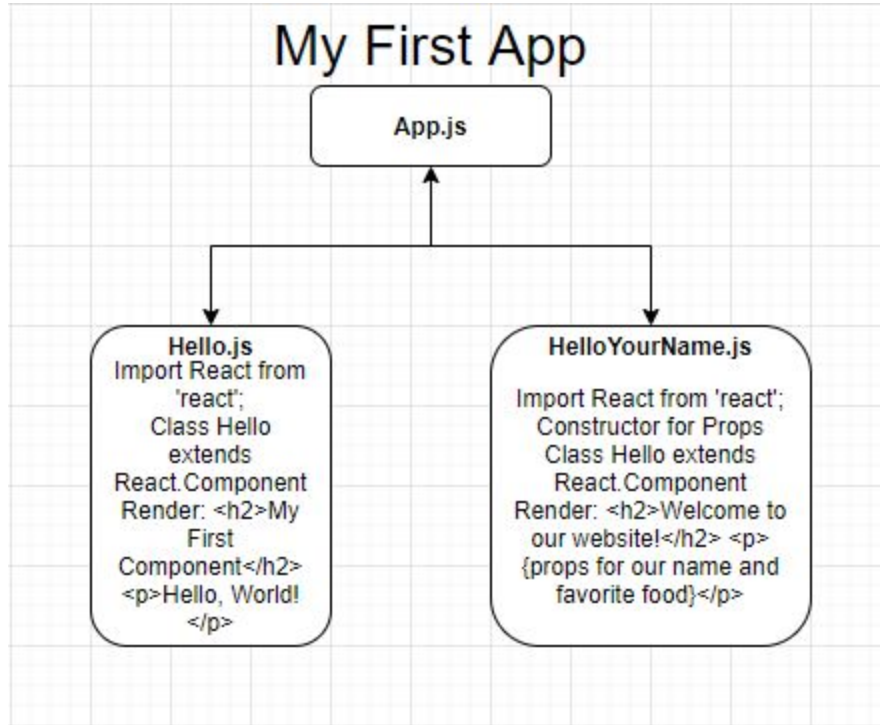
Challenges

- ☐ Add potential additional features
- ☐ Use JSX over pseudocode in your diagram (or partial JSX)

Hints

- Tools like draw.io can be useful in helping you create neat diagrams
- Be creative! Add features that would make sense in a to-do list
- Focus on the requirements before moving on to the challenges! Challenges are extra

Sample Screenshot



Rubric

You will be evaluated on the following points. You must get all 4 of the Mandatory points to pass:

Requirement	Points
MANDATORY: <ul style="list-style-type: none">• Components are broken down into reusable segments• Components have a detailed description of how each component will function (pseudocode)• Arrows are used to show how each component is linked to one another to create a fully functioning application• Image of diagram is uploaded to your repository for this assignment	4
CHALLENGE: <ul style="list-style-type: none">• Add potential additional features• Use JSX over pseudocode in your diagram (or partial JSX)	2
Total:	4

Citation Guide

Whenever you borrow code, the following information must be included:

- ❑ Comments to indicate both where the borrowed code begins and ends.
- ❑ A source linking to where you found the code.
- ❑ Your reason for adding the code to your assignment/project instead of writing it out yourself
- ❑ How it works. Explain to us how the code is supposed to work, include links to documentation/articles you read to help you understand.
- ❑ A small demonstration to prove you understand how the code works.

```
1  const inputArr = [5,1,3,4,2];
2
3  /*Borrowed code for bubbleSort starts*/
4  let bubbleSort = (inputArr) => {
5      let len = inputArr.length;
6      for (let i = 0; i < len; i++) {
7          for (let j = 0; j < len; j++) {
8              if (inputArr[j] > inputArr[j + 1]) {
9                  let tmp = inputArr[j];
10                 inputArr[j] = inputArr[j + 1];
11                 inputArr[j + 1] = tmp;
12             }
13         }
14     }
15     return inputArr;
16 };
17
18 /*Borrowed code from bubbleSort ends*/
19
20 //Source: bubbleSort function obtained from https://medium.com/javascript-algorithms/javascript-algorithms-bubble-sort-3d27f285c3b2
21 //Reason to add: implementing bubble sort can be tedious and bug prone, it would be better to use a proven version than to write my own
22 //How it works: I read the following article to understand how bubble sorts work (http://www.pkirs.utep.edu/CIS3355/Tutorials/chapter9)
23 //Demonstration of understanding:
24 //Example array: [3,1,2]
25 //Step 1: Compare 3 and 1. Since 1 is smaller, swap places.
26 //Array: [1,3,2]
27 //Step 2: Compare 3 and 2. Since 2 is smaller, swap places.
28 //Array: [1,2,3]
29 //Step 3: Compare 1 and 2. No need to swap.
30 //Array: [1,2,3]
31 //Step 4: Compare 2 and 3. No need to swap.
32 //Array: [1,2,3]
33 //Function complete.
34 console.log(bubbleSort(inputArr));
```