



# Assignment 1

## Components Pages & Routing

**Weight: 20%;**

**Due: Check Class Moodle Instance**

### Introduction

In this lab you will be creating the home and the todo pages for the project app. Each page will have a set of components that you will create and style. Check the build requirements for a list of components to build for each page. An example style is provided but you are not permitted to recreate the style in your project.

Once you have created the pages for the todo app you need to add routing between the homepage.js and the toDoPage.js

The app will be viewed and graded at a maximum viewport size of 480px. After the maximum size you can use a center page layout to frame the app for larger viewports.

## Build Requirements

### todos.json

Create a todo.json file this file will contain five different todos each from a unique category. Below you will find an example of a todo object. Take a look at the data structure you will be displaying the category title and isCompleted keys in your todoitem component.

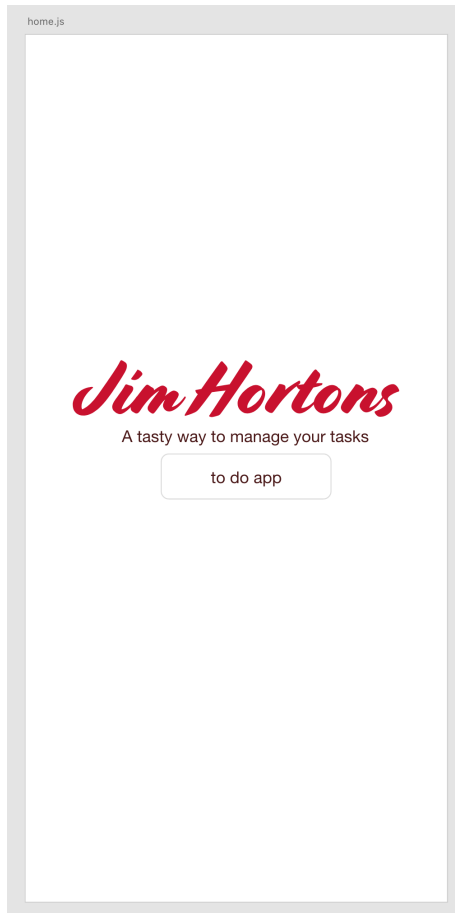
The startDate, startTime, endDate and endTime are just placeholder key value pairs for this assignment. Your component needs to display them. We will be using these four keys in a later build.

```
{
  "id": "aw45f25v",
  "category": "friends",
  "title": "Poss imusa dip is icing",
  "isComplete": true,
  "startDate": "Wednesday September 3rd",
  "startTime": "9:00am",
  "endDate": "Wednesday September 3rd",
  "endTime": "9:00am"
}
```

# Build Requirements

## home.js

An example of the home.js page can be seen below. **You can't use the artwork shown below**, you must create your own branding for the app. The Todo List App button will navigate the user to the todo.js page. To add the navigation add the router developed in class to your project.



## Components

Create the logo heading and link components for the home page. The image provided is an example and you are not to recreate it for this assignment. Your work should be custom and unique.

Add the logo heading and link to a header and add it to the homePage.js file. This is the page what should be displayed by default when your site loads.

### logo.js

Pick an icon from the resources below. Use the icon as the todo app brand.

*icon resources:*

<https://fonts.google.com/icons>  
<https://icons.getbootstrap.com>  
<https://heroicons.com/>  
<https://fontawesome.com/>

### header.js

level one heading component accepts a title and css class parameter. This is the to app name or brand name.

### tagline.js

component will accept a tagline and css class as its parameters. Used to display a catchy tagline for the todo app.

### link.js


generic link element component that accepts a title path and css class parameter. Uses the router to load the todolist.js page into the app container.

# Build Requirements

## todoPage.js

An example of the todoPage.js page can be seen below. **You can not use the design below for your todo app. Your app's artwork must be unique.**

todos.js





You've always got time for jim hortons

**School Category**



Due: Wednesday November 12th

Completed

**Home Category**



Due: Wednesday November 12th

**Social Category**



Due: Wednesday November 12th

Completed:



**Heath Category**

Due: Wednesday November 12th

**Work Category**

Due: Wednesday November 12th

+

### todo.js components

**branding header**  
The branding component created for the home page components logo title and tagline.

**addToDo component**  
This component will contain a button component and any other components you wish to use. This component should be at the bottom of the viewport.

**todolist component**  
This component should be an unordered list that will contain the 5 todo category list items.

**todoitem component**  
This component will be a list item that you will use to present the following data model. You will need 5 variations of this component for five categories. You can make up the categories.

```

    todoItem = {
      id:string,
      title:string,
      category:string
      isCompleted:boolean
    }
  
```

**id property**  
This property will be added as a data attribute to the todo item. Call the data attribute data-key. You will use this attribute to identify the todo item.

**title property**  
Add a field to the todoitem component for the title.

**category property**  
The todoitem category should be distinguished by a unique color, icon or styling.

**isCompleted property**  
Add a field to the todolist component to display the isCompleted state of the component.

**Date Placeholder**  
Add a text placeholder for the date. This could be a feature you could add later.

**Edit Button Component**  
Icon button for editing the component.

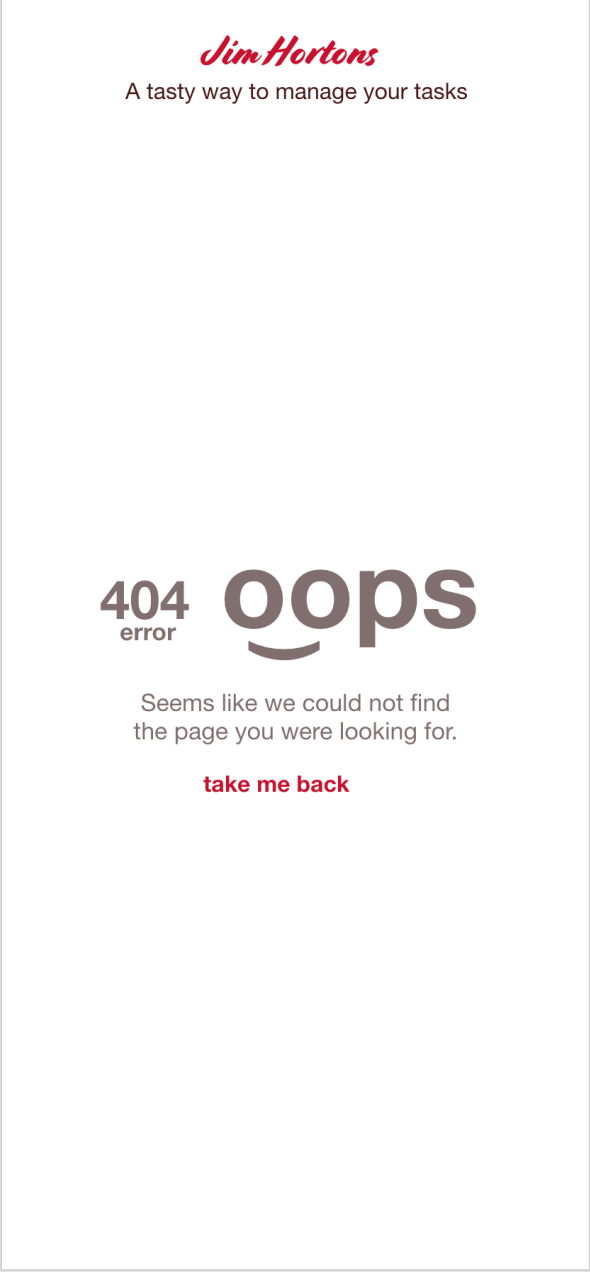
**Delete Button Component**  
Icon button for deleting the component.

**App Bar Component**  
Containing a button to add a new todo list item.

# Build Requirements

## pageNotFound.js

An example of the pageNotFound.js page can be seen below. **You can not use the design below for your todo app. Your app's artwork must be unique.**

<div data-bbox="198 525 331 543">pageNotFound.js</div> <div data-bbox="198 554 784 1818"></div>	<h3>pageNotFound.js components</h3> <p><b>branding header .</b> The branding component created for the home page components logo title and tagline.</p> <p><b>404 error message</b> Custom 404 error message, can contain artwork icons or anything else you would like to see on the page.</p> <p><b>take me back.</b> Link component.</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



## Submission

You are required to submit the url to the deployed app on Netlify and the url for the GitHub repository that contains the files for your todo app.

### **Submit To Moodle**

Netlify todo app url

GitHub repository url

**Late submission will not be graded.**



## Marking Key

Task	Mark	Task Value
<b>homePage.js</b>  <b>page components</b> logo title and tagline and any other components that you wish to add to customize your app.  <b>page styles</b> ui styles  <b>page routes</b> route to toDoPage.js		3
<b>toDoPage.js</b>  <b>page components</b> A logo title and tagline for the page header. A component for the toDoList that contains 5 toDoItems components. one from each category you have assigned. Control strip with an add new todo button.  <b>page styles</b> ui styles for components		12
<b>pageNotFound.js</b>  <b>page components</b> logo title ui styles homePage route		3
<b>GitHub Repository</b> contains a readme.md file with instructions on how to start and run the project.		1
<b>Deployment</b> Deployed to netlify using a github site deploy.		1
		<b>20</b>