React 1 α

New Attempt

Due Oct 11 by 11:59pm **Points** 5 **Submitting** a text entry box
Available after Sep 28 at 4pm

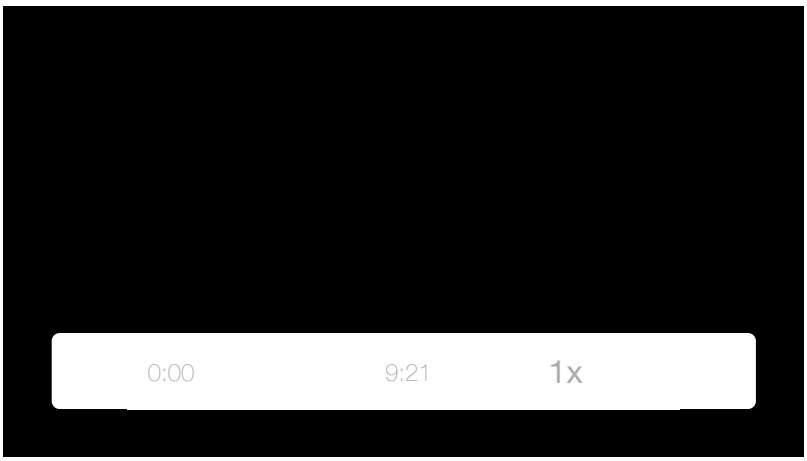
Assignment: React 1 α (5 Points Total)

Starter Code: [GitHub Classroom Starter Code (https://classroom.github.com/a/kcuGpwFP)](https://classroom.github.com/a/kcuGpwFP)

This assignment is meant to introduce you to more features of JavaScript, along with showing how a React project works. Additionally, this will introduce you to the different features of React, e.g. states, props, more modularized components etc. This project will be the first of a two part React project in which you will further extend your course management application. This application uses a limited quantity of modified data from the UW Madison course information database. When your assignment is ready for grading, please **submit your repository name and latest commit hash** from GitHub Classroom.

Setup Tutorial

Watch the [video (https://drive.google.com/file/d/1SeKw7zVTNdvIKyXXlHE5HucbstsKOt1I/view)](https://drive.google.com/file/d/1SeKw7zVTNdvIKyXXlHE5HucbstsKOt1I/view) below to get started with working on your first React project. It will walk you through how to compile and run the React 1 α project, and give you a sneak peek at how you would print course names to the web page.



Also, see [React 1 α Environment Setup (pdf) (https://canvas.wisc.edu/users/146319/files/21970669/download?verifier=t30t0Nh1ZHpQ4Gf8lGumf28SLoEuhcj5wDIAV5cM&download_frd=1)](https://canvas.wisc.edu/users/146319/files/21970669/download?verifier=t30t0Nh1ZHpQ4Gf8lGumf28SLoEuhcj5wDIAV5cM&download_frd=1)

Course Data

Important: *You must be signed into the [UW-Madison VPN (http://uwmadison.vpn.wisc.edu/)](http://uwmadison.vpn.wisc.edu/) or [Computer Sciences VPN (http://compsci.vpn.wisc.edu/)](http://compsci.vpn.wisc.edu/) in order to connect to the API below. See [this Wisc Help Article (https://kb.wisc.edu/page.php?id=90370)](https://kb.wisc.edu/page.php?id=90370) on how to setup GlobalProtect and connect to the VPN. When GlobalProtect asks you to enter portal address, use the following addresses for your preferred VPN service:*

VPN Service	Portal Address
Computer Sciences Departmental VPN	compsci.vpn.wisc.edu
UW-Madison VPN	uwmadison.vpn.wisc.edu

You can watch [our video walkthrough](#) on how to connect to the VPN and access our API endpoints (Please disregard the part that you must be connected to the CS VPN).

The course data is being fetched from `http://cs571.cs.wisc.edu:53706/api/react/classes` and is formatted as follows:

```
[
    {
        "credits": <number of credits for the course>,
        "description": <course description>,
        "keywords": <1D list of string keywords>,
        "name": <course name>,
        "number": <unique course number>,
        "requisites": <2D list of course requisites>,
        "sections": [
            {
                "instructor": <instructor name>,
                "location": <section location>,
                "subsections": [
                    {
                        "location": <subsection location>,
                        "time": {
                            <weekday>: <time range>, ...
                        },
                "number": <subsection number>
                    }
                ],
                "time": {
                <weekday>: <time range>, ...
                },
        "number": <section number>
            }, ...
        ],
        "subject": <course subject>
    }, ...
]
```

- The list of course requisites consists of 1D lists with AND operations between them. Each 1D list has OR operations between elements. For example: `[[A, B], [C, D, E], [F]]` means that the requisites are `(A OR B) AND (C OR D OR E) AND (F)`.
- A course can have any number of sections, and each section can have any number of subsections. If the course contains subsections, the user must schedule exactly one subsection with its parent section.
- Sections and subsections can have any number of times. Each time's key is a weekday in all lowercase ("monday", "tuesday", "wednesday", ...). Each time's value is a string with the following format: `"<12 hour time><am or pm> - <12 hour time><am or pm>"`. An example of this would be `"11:45am - 12:35pm"`.
- Each course has exactly one subject

Your project must be able to accept any data with the same format as above and the data located at `http://cs571.cs.wisc.edu:53706/api/react/classes`

Problem 1 (1 point)

Required: `src/Course.js`
Recommended: `src/Section.js` `src/Subsection.js`

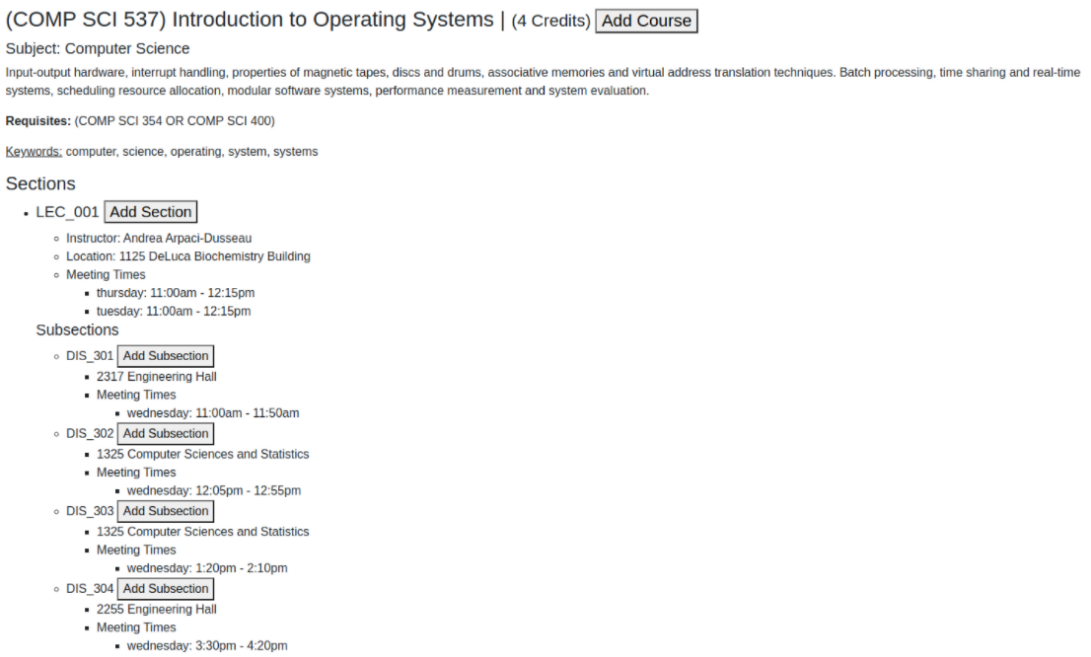
For this problem, you will be creating and organizing a way to represent the course data to the user.

There is an empty div currently being displayed for each course. You will display ALL information associated with each course instead of this empty div.

You will be graded on successfully displaying ALL content for each course. Note that you can use a modal (popup) or accordion (collapsible) view, as long as there is a way to access all of the specified data about a given course from its component.

- **(0.1 points)** Name
- **(0.1 points)** Credits
- **(0.1 points)** Course Number
- **(0.1 points)** Course Description
- **(0.1 points)** Requisites
  - *Note: Must be displayed as described above e.g.* `(A OR B) AND (C OR D OR E) AND (F)` *for relevant courses, for no requisites display* `None`.
- **(0.1 points)** Keywords
- **(0.05 points)** Subject
- **(0.05 points)** Section Number
- **(0.05 points)** Section Instructor
- **(0.05 points)** Section Location
- **(0.05 points)** Section Meeting Times (Weekdays + Times)
- **(0.05 points)** Subsection Number
- **(0.05 points)** Subsection Location
- **(0.05 points)** Subsection Meeting Times (Weekdays + Times)

Example Course Display



Problem 2 (1.5 points)

Required: `src/SearchAndFilter.js`

For this problem, you will be designing a search and filter method to decide which courses to display given a variety of inputs:

- **(0.5 points) Credits**: only display courses that have an amount of credits within the selected credit range *(inclusive)*
- **(0.5 points) Subject**: only display courses that match the selected subject
- **(0.5 points) Search**: only display courses that have a keyword that contains (or is) the user input from the search bar.

Providing multiple fields (e.g. both credits and subject) will return the intersection of the sets.

Problem 3 (2.5 points)

Recommended: Modifications to `App.js` `CourseArea.js` `Course.js` `Section.js` `Subsection.js` and/or additional Component(s)

For this problem, you will be creating a cart that users can **add** courses to and **remove** courses from.

Add to cart (1 point)

The user should be able to add 3 slight variations of course information into the cart:

1. A course with **all sections and subsections**
2. A course with **one specific section** of a course with **all subsections**
3. A course with **one specific section** that contains **one specific subsection**.

For example, if course `CS 571` has section `Section 1` with subsections `Subsection 1` and `Subsection 2`, the user should be able to add either of the following with the format of: course -> sections ->

subsections with one action (such as a button click)

1. `CS 571` -> `All` -> `All`
2. `CS 571` -> `Section 1` -> `All`
3. `CS 571` -> `Section 1` -> `Subsection 1`

Remove from cart (1 point)

The user should be able to remove 3 slight variations of course information from the cart:

1. A course with **all sections and subsections**
2. A course with **one specific section** of a course with **all subsections**
3. A course with **one specific section** that contains **one specific subsection**.

For example, if course `CS 571` has section `Section 1` with subsections `Subsection 1` and `Subsection 2`, the user should be able to remove either of the following with the format of: course -> sections -> subsections with one action (such as a button click)

1. `CS 571` -> `All` -> `All`
2. `CS 571` -> `Section 1` -> `All`
3. `CS 571` -> `Section 1` -> `Subsection 1`

View courses in cart (0.5 points)

The user should be able to view which courses are in the cart. From the cart, users should be able to remove courses as described above. For a course that only has some sections and/or subsections added to the cart, you can choose to display only these sections/subsections, or the data for the entire course while making it clear which sections/subsections the user has and has not added to the cart. When a course has been removed from the course, it should disapear immediately from the cart without any additional action from the user.

Run `npm install` in the terminal after cloning to automatically install needed npm packages

This project was bootstrapped with [Create React App (https://github.com/facebook/create-react-app)](https://github.com/facebook/create-react-app).

Available Scripts

In the project directory, you can run:

`npm start`

Runs the app in the development mode.
Open [http://localhost:3000 (http://localhost:3000/)](http://localhost:3000/) to view it in the browser.

The page will reload if you make edits.
You will also see any lint errors in the console.

Potentially Helpful Links

- Getting an Object's values (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/values)
- Getting an Object's keys (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/keys)
- An Array's forEach() method (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach)
- forEach() with Arrays of Objects (https://stackoverflow.com/questions/16626735/how-to-loop-through-an-array-containing-objects-and-access-their-properties)
- Pushing elements onto an Array (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push)
- Joining Array elements into a String (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/join)
- String to Int (parseInt()) (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/parseInt)
- React Bootstrap Modal View (https://react-bootstrap.github.io/components/modal/)
- React Bootstrap Accordion view (https://react-bootstrap.github.io/components/accordion/)

React 1 α

Criteria	Ratings				Pts
Course Data is displayed properly for each course.	**1 to >0.9 pts** **Full Marks** All course data is displayed correctly, in a legible format.	**0.9 to >0.5 pts** **Partial Marks** Most course data is displayed correctly, in a legible format.	**0.5 to >0.0 pts** **Partial Marks** Some course data is displayed correctly, in a legible format.	**0 pts** **Incorrect** Little to no course data is displayed in a legible format.	1 pts
Course filtering by subject works correctly	**0.5 pts** **Full Marks** Course filtering by subject works correctly.	**0.5 to >0.0 pts** **Partial Marks** A partial solution is present, but it is not fully correct.	**0 pts** **Incorrect** Not implemented, or minimal attempt at implementation.		0.5 pts
Course filtering by credit hours works correctly	**0.5 to >0.25 pts** **Full Marks** Course filtering by credit hours works correctly.	**0.25 to >0.0 pts** **Partial Marks** A partial solution is present, but it is not fully correct.	**0 pts** **Incorrect** Not implemented, or minimal attempt at implementation.		0.5 pts
Course search works correctly	**0.5 to >0.25 pts** **Full Marks** Course search works correctly.	**0.25 to >0.0 pts** **Partial Marks** A partial solution is present, but it is not fully correct.	**0 pts** **Incorrect** Not implemented, or minimal attempt at implementation.		0.5 pts

Criteria	Ratings			Pts
Courses can be added to the cart	**1 to >0.5 pts** **Full Marks** Courses can be added to the cart by course, section, or subsection. Adding a section automatically adds the course it is part of, and adding subsection automatically adds the section and course it is part of.	**0.5 to >0.0 pts** **Partial Marks** Courses can be added, but some features are missing regarding sections or subsections.	**0 pts** **No Marks** Adding courses to the cart is missing or minimally implemented.	1 pts
Courses can be removed from the cart	**1 to >0.5 pts** **Full Marks** Courses can be removed from the cart by course, section or subsection. Removing a section removes all subsections within that section, and removing a course removes all sections within that course, and all subsections within those sections.	**0.5 to >0.0 pts** **Partial Marks** Courses can be removed, but some features are missing regarding sections or subsections.	**0 pts** **No Marks** Removing courses form the cart is missing or minimally implemented.	1 pts
Courses in cart are displayed in dedicated view	**0.5 to >0.25 pts** **Full Marks** A view exists within the application for viewing courses in the cart, which clearly displays which sections and subsections the student is currently enrolled in.	**0.25 to >0.0 pts** **Partial Marks** A view exists within the application for viewing courses in the cart, but not all information about section and subsection enrollment is visible from this view.	**0 pts** **No Marks** A cart view is missing or minimally implemented.	0.5 pts
				Total Points: 5