Teaching Intro to Web Development with React

CSC497/CSC498
Research Senior Project Final Report
Winter 2022 - Spring 2022
By: Rey Ortiz

Computer Science Department
College of Engineering
California Polytechnic State University San Luis Obispo

Supervised by
Dr. Zoe Wood, Computer Science Department
Abstract:

Throughout the years, the field of computer science has increasingly become in demand and it is projected that in the next decade, job prospects are expected to grow much faster than the average rate for all occupations. With the reliance of technology in our daily lives and technology always evolving, the need for professionals with backgrounds in computer Science has risen dramatically. One of the things that we as educators can do to bridge this gap is to introduce students to web development at the beginning of their college career to get more students interested in the field of computer science. One of the most popular web development JavaScript libraries used is React, which is used at Facebook Netflix, Paypal, and NASA because it gives developers a lot of flexibility during development. The conventional intro programming courses have beginners start off by learning how to print “Hello World” in a language like Python or Java. This may not seem that exciting for students just starting off and can result in lack of motivation. By introducing web development to beginners, they are quickly able to display something on the screen and add their own personal touch. It is more likely that beginners will be more engaged with the material and try to build off on the basics they have acquired. This is why an intro to web development course is key to get students interested in computer science at the beginning of their academic careers and can explore other concentrations in computer science as they progress in their studies.
Introduction:

The field of computer science continues to gain popularity and the demand for positions to be filled out in the industry constantly grows every year. The need for professionals with backgrounds in computer science will continue to rise as industries need professionals who can solve complex and challenging problems. We as educators have the ability to help close this gap by introducing new ways to deliver material or unique courses at the beginning of a student’s college career to get them interested in computer science. One type of course that can motivate or inspire students to pursue a career in computer science, is to introduce them to an intro web development course using React. Students in such a course will be able to display something meaningful that they have created in a short time. The idea behind this course is to show students some of the projects that can be created with computer science principles and hopefully lead them to a career in computer science.

The reason for using a Javascript library such as React in an intro web development course is that it is one of the most popular front-end libraries among developers and it is widely used at companies like Facebook, Instagram, Uber, and many more. An intro web development course would cover the basics of HTML, CSS and Javascript at the beginning of the course to give the student the necessary tools to be successful and understand how to use React. The two main focuses in this type of course would be to make sure that students gain a valuable skill that they can use for personal projects or at an internship and also show students one of the many subfields of computer science.
React continues to be one of the most popular libraries, which was created by Jordan Walke, a Software Engineer at Facebook. React is popular amongst developers for many reasons. One reason is that the modular design of React allows developers to reuse components which cuts down on writing duplicate code for the same features. Each component acts as its own entity which allows you to update one section without having to update everything, this is important when working with more complex applications. Any changes that are made are rendered almost instantly without having to reload the whole page. Lastly, React allows developers to use JSX, an extension of Javascript, in which you can use HTML syntax directly in JavaScript to improve the performance and efficiency of React apps. If we take into account some of these benefits, introducing an intro web development course using React to beginners would be beneficial because they would pick up a very popular library that is widely used. Since React makes use of HTML and Javascript, it is easier for beginners to write code and be able to quickly put something up on the screen. React also has solid documentation and a good community following which are good resources for beginners to use if they get stuck. By learning React, students are able to create both web apps and mobile apps that can run on Android and iOS by using React Native. This flexibility allows them to develop apps without having to learn two different libraries or programming languages. React is widely used at companies such as Netflix, Paypal, NASA, BBC, and Lyft, students who learn React early in their college career are better positioned to get internships with such companies.

In the research paper done by the department of Math and Computer Science at Drexel University, faculty designed an intro web development course using HTML and Javascript to introduce freshman students to computer science principles. The reasoning behind creating this course was because introductory programming courses usually start with Python or C/C++. What they noticed was that introducing freshman students to general purpose languages required more work to create something simple and students lost motivation (Cooper et al. 193).
The basis behind this web development course at Drexel University, was that student motivation was higher because HTML and Javascript produced faster results with less work. The whole idea behind the curriculum that I developed was to engage the students and have them create something meaningful. I believe my curriculum captures the same rationale as the course from Drexel University. The faculty from Drexel University noticed from their experiences that students were more motivated to learn a particular tool if they saw it be useful outside the classroom setting. One major difference between the course from Drexel and the course I developed, is that they spent 2 weeks learning how to use the computers which could be partly due to this research being done in the late 1990s and not a lot of people knew how to use computers like today. One thing that was mentioned was that students did struggle during the middle part of the course when they reached the Javascript part. The course at Drexel did not go over Javascript at the beginning of the course, students were exposed to it during the middle part of the course and then were assigned homework on it. One of the things that I made sure to include in my course was to spend the first few weeks learning HTML, CSS and Javascript by creating labs and projects to make students comfortable using the necessary tools to be successful later on in the course.

Research done by the Applied Computer Science Department at Illinois State University, created a web development course but was very distinctive compared to my course. Their main target was junior, senior and graduate students with prerequisites of having taken two semesters of C/C++ and a semester off system design. This course went over HTML, Javascript, Java and also went over databases during the semester. This course would be comparable to CSC 437 which was taught here at Cal Poly before and required students to have taken CSC 357 and CSC 365. The course at Illinois State was taught in the 1990s when the internet was starting to gain popularity, the purpose for this course was to get students who were graduating soon to be able to learn about the internet and be able to apply these skills once they graduate (Lim, 108). My course on the other hand introduces students about web
development technologies at the beginning of their college career which I feel sets them up to be successful since they will have more experience since the skills they learn in this course can be applied to internships.

The course outlined in the research done by the Department of Computer Science at the University of Villanova was designed to give students the “big picture” of the overall working of a web application (Klassner, 78). The course at Villanova was more of an upper division course because it went over the server-side of a web application and did not actually go over how to build a web application. It tended to focus on how the overall web application works. Villanova is on a semester system so a course like this may be better divided into two quarters. Also when teaching a beginner about web development it would be more beneficial for them to learn how to create something than to learn why a web app works or how it is able to be displayed on the web.

In the research paper “Teaching Objects-first In Introductory Computer Science”, one of the distinct ways they taught the course was by using “Alice”, which is a 3D interactive animation programming environment for building virtual worlds designed for novices (Cooper et al., 192). As a beginner it may be difficult to get a solid understanding of certain topics, so being able to visually see how something works can be very beneficial. From the observation of the research, they found that students had a strong sense of design when working on code as a result of using this platform. The biggest takeaway that researchers found from students who used the animated programming environment in this course, was that they saw students in later classes had a better approach on solving problems and used pseudocode first before typing any code. One of the most important strengths of this course that I saw in the research paper was the students “Appreciation for trial and error”. Researchers found that the students would try out different things until they were able to solve the problem, this was partly due to the interactive platform visually showing the student what effect their change had. This ties in with my course
because in web development any changes that are made to a web app are visible and students have a concrete visual representation of what they are coding.

The same benefits that are found with the Alice platform are really similar with the General Assembly platform that I am using in the course for students to learn HTML, CSS and Javascript. The General Assembly platform displays the code that the student writes on the right side of the screen, just like students who used the Alice platform, students are able to instantly see the code they have written. The CS principles also use this very same method of using visual applications so that the students can see what their code is doing. In traditional intro computing courses it is hard to visualize what is going on because the output is text most of the time and if you do make any changes, the only real aide that students have is the compiler which gives you warnings or errors of the code.

**Course Outline:**

The purpose of this project was to develop an intro to web development curriculum to introduce new incoming students to the field of computer science and show them one of the many subfields of computer science. The curriculum that I developed entailed introducing students to HTML, CSS, Javascript and React which has become and continues to be very popular. The main learning outcomes for this course include the following:

- Learn and have a basic understanding of HTML, CSS, & Javascript.
- Gain experience using an IDE to develop and modify code.
- You will have a basic understanding of Javascript and how it is applied to front-end applications.
- You will be able to build an application using React and understand how to use React component libraries.
- Become familiar with React component libraries (React-Bootstrap, React Router, etc)
- Develop necessary skills to deploy a web application on the internet.
The course starts by introducing students to the basics of HTML, CSS and Javascript while completing exercises in an online platform known as General Assembly, where beginners can start writing code right from the start without having to download or set up anything. This course has two very interesting projects, one of them is that the student will create a basic web application for a fictional nonprofit organization from a template I created, using React. Lastly, the final project of the course would be creating or working with an existing web application for an actual non profit organization.

Labs:

In the course, my idea for having labs was for students to practice and apply the skills that they learned on HTML, CSS & Javascript. To get students up and running quickly without having to download anything and start writing code, I chose to incorporate an online platform called General Assembly which is completely free to use and teaches students about the basics of web development and then they can implement what they have learned by completing the exercises.

This platform has different projects with each project going over different areas of either HTML, CSS or Javascript. The image above shows “Project 1” which introduces students with the very basics of HTML. Each lesson goes over different topics where students read through different slides of a particular topic.
Once the student finishes reading through a certain number of slides, the student reaches a checkpoint in which they have to write some code in the text editor as shown in the image below. The output is displayed immediately on the right hand side so the student is able to see in real time the piece of code that they have written. If the student writes the code correctly then they are shown a green check mark and they are prompted to continue with the lesson. Once the lesson is completed they earn a check mark for the lesson that they have completed and can continue to the other lessons in the project. All projects that are in the General Assembly platform follow this same structure which creates an interactive learning experience. This course also has traditional labs that focus on tools that will help the student during the course.

Project(s):

The labs were incorporated as a means for students to understand and learn the basics of the tools required to be successful with React. To make sure that potential students taking this course have an understanding of what they have learned, I decided to have two projects that build off each other in which students would create a web application using React. Students would implement a fictional non-profit organization website, from a template I created of a fictional community park in the city of San Luis Obispo. The students would be following prototypes that I created on Figma as a guide to what I would expect them to implement. The image below is one the guides that the student will be following, this first prototype is the home
page of the web app. The student would implement the navigation bar that links to other pages, an image that is relevant to the non profit organization and just below that a small description of the organization. The student will also implement three images that also link it to other pages. Lastly the home page would also have a bottom footer that has links to other pages and links that redirects a user to the social media pages of the non profit organization.

The other part of the project was to implement the “Future Work” page which has to do with future projects that this organization has planned out or is planning on. In the prototype I created a “Future Work” page, and have the students implement a header that states the name of the page and just below that I want them to add an image that is relevant to the organization. The image below shows the guide the students are following on the “Future Work” page and the contents required. The main aspect of the page is to have an image that is relevant to the organization and have a 2 x 2 image grid that has a short description of the future work on the left hand side of the web page and on the right hand side a picture of future work that they are doing. One thing to note is that the guides that the students are following are simply requirements of the structure that I expect them to implement, the students have the choice to customize the web app however they see fit.
The next page that the student will implement is the “Donate” page which gives a small description on why people should donate to this organization and should include an image of the organization next to the description. Another feature is to have three images with a small description on how a donation would benefit not only the organization but the community as well. The website needs a button to redirect the user to a page where they can donate to the organization, for simplicity the button that the students would implement would simply redirect someone to paypal.com. I mentioned in the project description that Paypal gives you code with the Paypal button that when clicked allows a user to make a donation with their Paypal account or with their credit card and sends the money to the Paypal account of the generated code.
The final two pages that the students have to implement are the “About us” and “Contact us” pages. The “About Us” page has a similar layout to the “Donate” page but one of the differences is that there is a button that has more details about the future work which is linked to the “Future work” page that has been shown previously. Lastly, I have a basic contact page with basic input fields such as the name, email, phone number, and the message text input area. There is also a submit button that once a user is finished typing their message, they are able to submit it. Just above the contact form, there is a section where the student can add any text that they think would be relevant to the contact form. This can include a short message letting the user know that they will get a response in a few days regarding their message.

Final Project:

For the final project, the plan is to be able to work with an actual outside non profit organization and work on their web app to update or make changes according to their needs. Arrangements to secure outside projects would have to be considered at least a few months before the start of this course to make sure there would be enough projects for students to work on. Depending on how many students would be in this course, the idea is to have teams of at
least 5 students per team working on different projects. Through my research and talking with professors in the Department of Computer Science at Cal Poly, I found out that a good resource to look at would be Hack 4 Impact since they already work with outside projects. Another option to consider is Cal Poly SBDC, which provides assistance to startups. By working with SBDC, the students would provide web development services to startups. Working on an outside project not only allows students to apply what they have learned in the classroom but also continue with the philosophy of “Learn by Doing” which is what makes the learning experience at Cal Poly unique compared to other institutions of higher education.

Reflections:
I was excited to work on this project and see it evolve to where it is at right now because this project incorporates my interests of web development which started with taking an “Intro to Computing with HTML” course at Allan Hancock College and also teaching since I used to be a math tutor during high school. Starting this project with my previous background experience made it easier to pick the technologies that I chose for the curriculum that I developed. The most difficult part of this project was creating the assignments themselves because there are many factors that can determine whether the assignments are engaging for the students and if the assignments are difficult enough to test the student but at the same time not be overly difficult to the point that the student is struggling to complete them. Looking through various intro computing courses at other universities gave me a good idea on the difficulty and types of assignments that I could create. My main inspiration for the curriculum that I developed was the “Intro to Computing” course that I took at Allan Hancock College which is how I ended up pursuing a degree in computer science. One of things that the course did well was that we were introduced to HTML right away and the project for the course was creating a website for a fictional business that sold products. The professor had a template that we had to follow but gave us the option to customize the site to how we wanted. I followed a similar structure as the
class at Allan Hancock College to deliver the project instructions by creating a template on Figma of the requirements expected which made the requirements easy to understand. Through different research papers that I read on other intro computing courses, some tended to focus on the basics on how to use a computer which seems like a bad use of time since towards the middle part of the semester students struggled since more time could have been spent going over coding concepts. I made sure that my course focused on the basics at the beginning so students are less likely to struggle through the course.

**Future Work:**

Over the course of two quarters my plan was to develop actual lab assignments, projects, and have a well structured course that can be taught to college level students. I can say that I was successful in creating just that and accomplished everything that I had planned out. I was also able to actually code the assignments and make any adjustments to make the requirements more clear to the students. If I would have had more time to work on this project, one thing that I would have liked to have done was to teach students about the basics of web development and then have them work on the assignments that I created and get their feedback on the assignments. Getting actual feedback from students on the assignments would have been great to have a better idea of what can be improved. If this project was to be continued by someone else, the only thing would be to teach this curriculum at Cal Poly to incoming freshmen. This course would be a great addition to the various topics that are taught in CSC 123. This course would give incoming freshmen another option to choose from the current topics that are being offered.
References


Castellanos, Joel, Syllabus for CS-105L Introduction to Computer Programming. Spring 2018, University of New Mexico, Albuquerque.


Donham, Perry, Syllabus for CS101: Introduction to Computing. Fall 2019, U of Massachusetts, Boston.


Wagner, Michael, Syllabus for CS 102: Introduction to Computing with HTML. Fall 2021, Allan Hancock College, Santa Maria.

Appendix A

CSC 1XX: React Web Development - Lab 1

Objectives:

- To be able to apply the basics of HTML in a web application.
- To be able to apply the basics of CSS and add styling.
- To understand the basics of a web application.

Overview:

To learn about React we must first learn the basics of HTML and CSS. In this lab you will become more familiar with HTML and CSS by applying what you have learned and building a small application in increments. For this lab we will be using General Assembly’s Dash, there is no need to set up any IDE; everything for this first lab will be done online through Dash.

Lab Details:

For this lab we will be using an online application known as Dash to write code. Below is all the details of the lab:

1. Head over to https://dash.generalassemb.ly/ and click on the button on the very center of the page that says “Start Learning”.
2. It will then ask you to create and input your information. Once you have inputted your information make sure to confirm your email if required.
3. Once you have made your account you will be redirected to the main where you will see “Project 1”.

4. Start with “Make the headline and inputs”, make sure to read all the slides carefully as it contains valuable information and instruction on how to complete it.
5. Continue until you have finished parts 1,2 and 3 of the project.
**Grading:**

The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

**Demo:**

You will have exactly one week from that date this lab was assigned. You will show me that you completed project 1 on Dash. Come to the lab on the due day with your lab finished!

**Collaboration:**

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand the basics of HTML and CSS as we progress through the quarter and start implementing projects of bigger scope.

**Resources:**

If you are in need of supplemental material to understand the basics of HTML and/or CSS, I have linked two sources that may be helpful to you:

https://www.w3schools.com/html/default.asp

https://www.w3schools.com/css/default.asp
Appendix B

CSC 1XX: React Web Development - Lab 2

Objectives:

- To get an introduction on the basics of Javascript.
- To be able to understand the basics of Javascript
- To understand code structure of Javascript

Overview:

To learn about React we must first learn the basics of HTML, CSS and Javascript. In the previous lab you were introduced to HTML and CSS. In this lab you will be introduced to the basics of Javascript. This lab will serve as your first introduction to the Javascript language. You can use your preferred editor, but can use the following online editor without the need to download anything. [https://www.onlinegdb.com/](https://www.onlinegdb.com/)

Lab Details:

Below is all the problems that you will need to complete for this lab:

1. For the first part you will initialize 3 variables X, Y, and Z. X and Y will contain an integer and Z will be the addition of X and Y. You will now print the result of the variable Z.
2. Create a while loop that prints the numbers from 1 to 10.
3. The second task involves creating an array of integers and creating a loop that iterates through the array and prints the content of the array. Don’t use any built in libraries for this!
4. Write a function named myGrades that:
   a. takes 1 argument, a number score.
   b. returns a grade for the score, either "A", "B", "C", "D", or "F".
   c. Call that function for a few different scores and log the result to make sure it works
5. Print a triangle using only loops, the output show look like this:

```
  *
 ***
 *****
  *
 **
  *```

Grading:

The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

Demo:

You will have exactly one week from that date this lab was assigned. You will show me that you completed lab 2. Come to the lab on the due day with your lab finished!

Collaboration:

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand the basics of Javascript. As we progress through the quarter and start implementing projects of bigger scope you will be using it to implement various features.

Resources:

If you are in need of supplemental material to understand Javascript I have linked two sources that may be helpful to you:

https://www.w3schools.com/js/
https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basic
Appendix C

CSC 1XX: React Web Development - Lab 3a

Objectives:

- To be able to setup Visual Studio Code for React Development
- To be able to create your very first React project
- To have a understanding of a basics of React

Overview:

The purpose of this lab is for you to download Visual Studio Code and familiarize yourself with Visual Studio Code and be able to create your very first project. For assignments we will be using Visual Studio Code as our development working environment. We will also set up the environment plugins to get up and running.

Lab Details:

For this lab we will download Visual Studio Code and Node.js to get our environment up and running. Once we have our environment up you will display “Hello, World!” on the web browser using npm.

1. Our first step will be to download Visual Studio Code by clicking on the following link: https://code.visualstudio.com/download
2. Pick your OS from the list and follow the instructions to download Visual Studio Code.
3. Next we will check if we have node.js installed on our local machine. Open up Visual Studio Code and go over to “Terminal” then click on “New Terminal”. You will then see the terminal open at the very bottom.
4. Next you will type “node --version” and then “npm --version” without the quotation marks. If you have node.js preinstalled it will display the version.

![Node and NPM Version]

5. If you do not have node.js installed click the following link and follow the download instructions: [https://nodejs.org/en/](https://nodejs.org/en/). Repeat steps 3 & 4 to make sure you have node.js installed.

6. Once we have node.js installed, open up the terminal in Visual Studio Code. We want to create a file in Desktop so we type “cd Desktop” to go to the Desktop directory then type “mkdir Lab3” to make a folder called Lab3. *note (Don’t add the quotation marks)

![Directory Creation]

7. In the same terminal we will type “cd Lab3” which will take us to the Lab 3 folder then we will type “npx create-react-app my-app” to start downloading the dependencies for React.

![React App Creation]
8. Once downloading is complete go to the directory you named Lab3 and type “cd my-app”. Once you have that, type “npm start” in the command line and a separate browser window will display the default React Logo.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\rey4p> cd Desktop
PS C:\Users\rey4p\Desktop> cd Lab3
PS C:\Users\rey4p\Desktop\Lab3> cd my-app
PS C:\Users\rey4p\Desktop\Lab3\my-app> npm start

> my-app@0.1.0 start
> react-scripts start
```

9. If you did everything correctly the following window will display:

```
Edit src/App.js and save to reload.
Learn React
```

10. Display “Hello,World!” as an <h1> header by modifying the App.js file.
Grading:

The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

Demo:

You will have exactly one week from that date this lab was assigned. You will show me that you were able to download and get Visual Studio Code running by displaying “Hello, World!” in the browser. Come to the lab on the due day with your lab finished!

Collaboration:

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand how to navigate Visual Studio Code and be able to run your application.

Resources:
If you want learn more about React and Visual Studio Code check out the following links:
https://reactjs.org/tutorial/tutorial.html
https://code.visualstudio.com/docs
CSC 1XX: React Web Development - Lab 3b

Objectives:

- To be able to setup GitHub Desktop
- To be able to push code to your GitHub repo
- Become familiar with Github and its capabilities

Overview:

The purpose of this lab is for you to download and familiarize yourself with GitHub Desktop. You will create a Github account and download Git and GitHub Desktop. Once you have downloaded everything you will push your code from the first part of lab 3.

Lab Details:

1. Create an account on GitHub by following the link: https://github.com/.
2. Once you have created your account you will need to download GitHub Desktop by clicking on the following link: https://desktop.github.com/.
3. After you have finished downloading GitHub Desktop we need to download Git by clicking on this link: https://git-scm.com/downloads.
4. Once you have everything downloaded follow this link to configure GitHub Desktop: https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/configuring-and-customizing-github-desktop/configuring-git-for-github-desktop
5. Open up GitHub Desktop, go to File > Add local repository.
6. Add the path of the Lab3 project as shown below:

7. Next we will commit our changes from the lab. Make sure to add a description of the changes and then click “Commit to master”.

8. Next we need to publish our repo so click on “Publish”: 
9. Next name your repo, then click “Publish” at the very bottom:

10. You should now see your repo published on Github.com:

**Grading:**
The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.
Demo:

You will have exactly one week from that date this lab was assigned. You will show me that you were able to push your code on to GitHub.

Collaboration:

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand how to commit code to your repo on GitHub.
Appendix D

CSC 1XX: React Web Development - Lab 4

Objectives:

- To be able to apply the basics of CSS in a web application.
- To continue learning about Javascript and see how it works in web development.
- To understand how to make a responsive web application.

Overview:

We will continue to learn more about HTML, CSS and Javascript. In this lab you will continue developing your HTML and CSS skills by applying it to a small scale application and be introduced to Javascript. For this lab we will be using General Assembly’s Dash, there is no need to set up any IDE; everything for this lab will be done online through Dash.

Lab Details:

For this lab we will be using the online application known as Dash to write code. Below is all the details of the lab:

1. Head over to https://dash.generalassemb.ly/ and enter your login credentials.
2. For this lab you will be working on “Project 2”.

3. Start with “Make the header and navigation HTML”, make sure to read all the slides carefully as it contains valuable information and instruction on how to complete it.
4. Continue until you have finished everything up to number 3. You do not have to do number 4.
**Grading:**

The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

**Demo:**

You will have exactly one week from that date this lab was assigned. You will show me that you completed project 2 on Dash. Come to the lab on the due day with your lab finished!

**Collaboration:**

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand the basics of HTML, CSS and now Javascript. As we progress through the quarter and start implementing projects of bigger scope you will be using all three to implement various features.

**Resources:**

If you are in need of supplemental material to understand the basics of HTML, CSS, or Javascript I have linked two sources that may be helpful to you:

https://www.w3schools.com/html/default.asp

https://www.w3schools.com/css/default.asp

Appendix E

CSC 1XX: React Web Development - Lab 5

Objectives:

- To be able to apply advanced CSS code in a web application.
- To learn some of the basics on how to make a web look aesthetically pleasing.
- To understand how to make a responsive web application.

Overview:

We will continue to learn more about HTML, CSS and Javascript. In this lab you will continue developing your HTML, CSS, and Javascript skills by applying it to a small scale application. For this lab we will be using General Assembly’s Dash, there is no need to set up any IDE; everything for this lab will be done online through Dash.

Lab Details:

For this lab we will be using the online application known as Dash to write code. Below is all the details of the lab:

1. Head over to https://dash.generalassemb.ly/ and enter your login credentials.
2. For this lab you will be working on “Project 2”.
3. Start with “Position images & text”, make sure to read all the slides carefully as it contains valuable information and instruction on how to complete it.
4. Continue until you have finished everything up to number 2. You do not have to do numbers 3 & 4.
Grading:

The lab will be graded depending if the lab is complete or not. If you complete everything you will earn full marks, if you fail to complete all of it you will fail to earn any marks. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

Demo:

You will have exactly one week from that date this lab was assigned. You will show me that you completed project 3 on Dash. Come to the lab on the due day with your lab finished!

Collaboration:

This lab is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this lab individually since you will need to understand the basics of HTML, and CSS. As we progress through the quarter and start implementing projects of bigger scope you will be using all three to implement various features.

Resources:

If you are in need of supplemental material to understand the basics of HTML, CSS, or Javascript I have linked two sources that may be helpful to you:

https://www.w3schools.com/html/default.asp

https://www.w3schools.com/css/default.asp

https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basic
Appendix F
CSC 1XX: React Web Development - Milestone 1

Objectives:

- To be able to setup React development environment for the course project
- To be able to implement a basic landing page
- To be able to implement and add features to the landing page.
- To have an understanding of the React structure in web development

Overview:

The assignment for this course will be broken up into two parts where you will be adding new features during each milestone. For the first milestone our focus will be on creating the landing page (home page) for our web app. We will start by getting a top nav bar implemented on the landing page and then add some features to the landing page.

Deliverables:

For milestone 1 we will focus on creating a landing page for a nonprofit organization of your choice. For milestone 1, I am looking to make sure that your project has a top navigation bar, a picture in the center of the landing page, a bottom footer and a description of the business or what makes them unique. Below is an example of the structure that I am looking for, follow the image below as your guide buy you can customize it as you wish.
1. The first step is to create the project (see lab 3 if you forgot) and get everything up and running. Before we start with the implementation head over to the App.js file and delete everything inside the div. You can also comment out the code in the App.css file for now. Add an <h1> tag displaying “Hello world!” like the image below to make sure everything is good.

![Image of App.js and App.css files]

2. Now you will create the top navigation bar for the web site. You can create the pages to pages listed on the navbar and put placeholder text. We will be utilizing them in Milestone 2. Before you begin, you must install the React Router library by typing the following in the terminal: npm install react-router-dom

*Hint* When adding the routes to the navbar use Routes instead of Switch. Version 6.0.0 and higher of React Router does not use Switch anymore. See the resources listed below for more information.

3. You will need to add css to the navigation to get it to look like the example on page 1. You can pick any color you want and logo image depending on your choice nonprofit organization. Since you will be creating different components and css files, create a Folder that contains your components and css files.

4. Once the navigation is displayed, we want to add an image of the organization such as the image on the guide on page 1. To make it easier it would be best to create a Home.js file that houses all the components used in the main landing page. Make sure to add CSS to the <div> that contains the image to get it properly aligned.
5. We are going to have a top and a bottom `<div>`, first focus on the top `<div>` that contains the mission statement of the organization on the left and right side, and an image that pertains to the organization. We want to create a grid system that contains 1 row and three columns that will contain the required information.

6. For the bottom `<div>` that redirects the user to either to the donate, future plans, or events pages uses a similar structure as the top `<div>` You can reuse the code from the top container and modify it to make it work with the bottom container. If you have already created the redirect pages you can link them to the images cards on the bottom container.

7. The “Events” cards will be directed to an events calendar, Outlook Calendar is easy to share since it gives you a link which can be added to the “Events Card” To get your link, you will click on the settings in the upper right hand corner. It will then take you to this page where you can get the html that you will be using. Please note that you can use any other calendar that you wish.

8. The last part of this first assignment is to create a simple footer that will be used throughout all the pages of the web page. On the bottom left of the footer we will have a quicklinks header and below we will have the pages of our site. On the right side we will have the social media links. For this assignment just set the link to the home page of that particular social media platform.

Grading:

The lab will be graded depending if you implemented everything that was required. If you complete everything correctly you will earn full marks, if you fail to complete all of it you will earn a lower score. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

Demo:
You will have exactly two weeks from that date this milestone was assigned. You will submit a zip file of all your code on Canvas.

Collaboration:

This milestone is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this milestone individually since you will be adding more features throughout the quarter.

Resources:

If you want learn more about React, HTML, and/or CSS click on the links below:
https://reactjs.org/tutorial/tutorial.html
https://www.w3schools.com/html/
https://www.w3schools.com/css/default.asp
https://reacttraining.com/blog/react-router-v6-pre/
Appendix F
CSC 1XX: React Web Development - Milestone 2

Objectives:

- To be able to add new features to the current web app
- To be able to implement a basic landing pages
- To be able to implement and add features to the landing pages
- To have an understanding of the React structure in web development

Overview:

For the second milestone our focus will be on creating the landing pages that are listed in the navbar for our web app. We will start by implementing each of the pages listed on the navbar.

Deliverables:

For milestone 2 we will focus on creating a landing page for the “Donate”, “About Us” and “Contact” landing pages. For milestone 2, I am looking to make sure that your project has each of the previous landing pages implemented from the navigation bar. A picture of the structure of each of the pages will be shown below. Each page must be able to switch from page to page using the navigation bar. On some of the landing pages you will have to create external links.
1. For the About Us page you will notice (see above picture) that we will be using the same image as the home page. One of the key features of React is that we can reuse components in various parts of our application.

2. For the “Our Mission” header you can either keep the header that says “Our Mission” or you can add “About Us”. You will want to center the header so that it looks appropriate on the web page. *Hint* When you want to style something you can add styling in the tag itself instead of having to write code in the CSS file. Check this site on how to accomplish this. https://www.w3schools.com/react/react_css.asp

3. Just below the “Our Mission” header you will be adding a mission statement of your particular non-profit organization with a picture that is relevant to the organization. To separate the mission statement with the “Future Work” add a <hr> tag and style it accordingly

4. Lastly, you will implement the div that contains the “Future Work” statement with two relevant images to the organization. For this part of the implementation a grid would be better as each part of this can be divided into 1 row and 3 columns. Once that is completed you will implement a button that redirects you to the future page. Make sure to center the button accordingly as shown in the image above.
1. For the contact form page you will notice that we will be using the same image as the home page. As stated previously, we can reuse components in various parts of our application. You may want to reuse previous code in this case. The “Contact Us” header is very similar to the “About Us” header you previously implemented. You may also want to reuse that code.

2. Just below the header we will create a div that will house a brief statement if someone wants to contact the organizations. 1-3 sentences should be fine in this case, a brief example could thanking them for showing interest in the organizations and that they will get a response in a few days.

3. Just like you did previously add a line that will separate the actual contact form with the contact form statement.

4. You will now be implementing a simple form and a submit button that will allow users to submit a question. Some key tags to consider when creating the form is the <form> tag which will house the actual form and the <label> and <input> tags. The last two tags will create the fields as shown in the image above. To get the form and the contents of the form you will be using CSS to adjust it.
1. Just like the previous pages you can reuse the background component as you have previously done. The “Why Donate?” header is also very similar to the other headers you have implemented, so you can reuse the same code again.

2. This page follows a similar structure as the above page, just below the “Why Donate?” header you will have a statement why someone should donate and a picture relevant to the organization.

3. Lastly you will be using another 1 x 3 grid that will house the three reasons why someone should donate to your particular organization. The images can be anything relevant to the small statement as shown in the image above. Once you have that done you will implement the button that redirects them to the donate page.

4. Since we are not actually getting money, if you click on the button it will redirect you to the Paypal or Venmo homepage. If we were to actually have a button that can receive donations, Paypal generates a unique id that if someone donates it will send you money to your Paypal account. For this project you can create your own donate button with a redirect of your choice. You can also use the Paypal button which this code is shown below.

```html
<form action="https://www.paypal.com/">
    <input type="hidden" name="business" value="H5G4Z4YCVD554" />
    <input type="hidden" name="no_recurring" value="0" />
    <input type="hidden" name="currency_code" value="USD" />
    <input type="image" src="https://www.paypalobjects.com/en_US/i/btn/btn_donateCC_LG.gif" border="0" name="submit" title="PayPal - The safer, easier way to pay online!" alt="Donate with PayPal button" />
    <img alt="" border="0" src="https://www.paypal.com/en_US/i/scr/pixel.gif" width="1" height="1" />
</form>
</div>
```
1. Lastly you will implement the “Future” page, you will add the top header as shown in the image below. It follows the same header styling and format as the other pages you have already implemented.

2. For this page we want an image to be in the background while the main content is above it. In the image above, we have an aerial image of San Luis Obipso. Your background image can be anything related to your organizations.
3.
You will then create another grid that will have the future work that your organization has planned on the left hand side of the web page. On the right hand side you will add an image of the project that your organization has planned or will plan. See example above.

**Grading:**

The lab will be graded depending if you implemented everything that was required. If you complete everything correctly you will earn full marks, if you fail to complete all of it you will earn a lower score. There will be no late work accepted, if you fail to turn the lab in on time you will earn a zero for this lab.

**Demo:**

You will have exactly two weeks from that date this milestone was assigned. You will submit a zip file of all your code on Canvas.

**Collaboration:**

This milestone is to be completed individually, group collaboration is not allowed. It is in your best interest to complete this milestone individually since you will be adding more features throughout the quarter.

**Resources:**

If you want learn more about React, HTML, and/or CSS click on the links below:

- [https://reactjs.org/tutorial/tutorial.html](https://reactjs.org/tutorial/tutorial.html)
- [https://www.w3schools.com/html/](https://www.w3schools.com/html/)
- [https://www.w3schools.com/css/default.asp](https://www.w3schools.com/css/default.asp)