Group Report & Code Documentation

Senior Project II: GradTrak

05/15/23

By: Justin Ulloa, Haylee Allen, Shadi Abdul Razzak, & Tristan Bock

Table of Contents

[Implementation 2](#_Toc135067253)

[Front-End 3](#_Toc135067254)

[Index Page 3](#_Toc135067255)

[Figure 1. *Index Page (PC Layout)* 3](#_Toc135067256)

[Figure 1.2. *First time user* 4](#_Toc135067257)

[Figure 1.3. *NetID Link* 5](#_Toc135067258)

[Figure 1.4. *Index page (Mobile-View)* 6](#_Toc135067259)

[Code (Index Page) 6](#_Toc135067260)

[Figure 1.5 *Code snippet of index page* 7](#_Toc135067261)

[Design (Index Page) 7](#_Toc135067262)

[Figure 1.6. *Code snippet of background design (index Page)* 8](#_Toc135067263)

[Figure 1.7. *Code snippet of header and container* 8](#_Toc135067264)

[Figure 1.8. *Code snippet of buttons an footer* 9](#_Toc135067265)

[Login page 9](#_Toc135067266)

[Figure 1.9. *Login Page* 10](#_Toc135067267)

[Student Page 10](#_Toc135067268)

[Figure 2. *Student Page (PC Layout)* 10](#_Toc135067269)

[Figure 2.1 *Course Requirements* 11](#_Toc135067270)

[Figure 2.2. *Career Resources* 12](#_Toc135067271)

[Code (Student Page) 12](#_Toc135067272)

[Figure 2.3. *Code snippet of Navigation Bar* 13](#_Toc135067273)

[Figure 2.4*. Code snippet of Progress Bar* 13](#_Toc135067274)

[Figure 2.5. *code snippet of student information structure* 14](#_Toc135067275)

[Figure 2.6. *code snippet of accordion and collapsible elements* 15](#_Toc135067276)

[Figure 2.7. *Code Snippet of table structure in panels* 16](#_Toc135067277)

[Figure 2.8. *Code Snippet of Career Resources Section* 17](#_Toc135067278)

[About Page 17](#_Toc135067279)

[Figure 3. *About us* 18](#_Toc135067280)

[Figure 3.1. *Developer Team* 18](#_Toc135067281)

[Figure 3.2. *Featurette Section* 19](#_Toc135067282)

[Code (About Page) 19](#_Toc135067283)

[Figure 3.3. *Code snippet of developer team structure* 20](#_Toc135067284)

[Figure 3.4. *Code snippet of featurette section* 20](#_Toc135067285)

[Works Cited 21](#_Toc135067286)

# 

# **Implementation**

## Front-End

In this section, we will be going into detail about how each page for our application was created in terms of the thought process behind each page, providing what features were used, and describing how those features were applied into our project. It will include relevant images of the final product of our application as well as code snippets being described into detail showing how our product was created.

## Index Page

A screenshot of a web page

Description automatically generated with low confidence

### Figure 1. *Index Page (PC Layout)*

Figure 1 is our completed design of the index page of the application. It was designed using Bootstrap, HTML, & CSS. As you can see, the page displays the name of our application, GradTrak, a section welcoming the users, providing a brief description of the intended purpose of our application, a ready to login in section telling the user what information they need to use our application, as well as a first-time user section that provides the user with a button that navigates them to a page that helps them understand what information is needed and what steps to take if they do know they’re information to access the application that can be seen in Figure 1.2. There is also a secondary link that takes the user to our CSUB website to a specific page where students can look up their NetID, if they are not familiar with it and how to reclaim their password that can be seen in Figure 1.3. The background for the page is an image of the CSUB campus so that the page follows the theme a common university website page. Lastly, the page contains a footer that displays a copyright message.

A screenshot of a login page

Description automatically generated with medium confidence

### Figure 1.2. *First time user*

A screenshot of a computer screen

Description automatically generated with low confidence

### Figure 1.3. *NetID Link*

Figure 1.4 showcases our application from a mobile viewpoint. We wanted to ensure that our web application could be useable using any web browser and any devices such as phones, tablets, etc. As is shown, our web page is responsive and fills its’ contents depending on the device being used. This is something we often discussed about and had to learn the best way to approach this task.

### A screenshot of a cellphone Description automatically generated with low confidence

### Figure 1.4. *Index page (Mobile-View)*

## 

## Code (Index Page)

A screen shot of a computer program

Description automatically generated with low confidence

### Figure 1.5 *Code snippet of index page*

As we mentioned earlier, the index page was created using Bootstrap, HTML, and CSS. Justin made sure to include a Bootstrap stylesheet and script, and a jQuery script to format the content to our liking. As is shown, the header of our index page was created defining a div class “jumbotron” that Bootstrap automatically formats to make the header look nice and be positioned at the top of the page. He also defined a separate class “container” that contains 3 separate columns that automatically separates the 3 different sections, “Welcome to GradTrak”, “Ready to Login?”, and “First Time User”. Bootstrap came in handy here, separating out the content for us evenly and formatted to what we were aiming for. Both the log-in button and click here button were created using Bootstrap’s scripts. A separate class was also made for the footer.

## Design (Index Page)

A screen shot of a computer program

Description automatically generated with medium confidence

### Figure 1.6. *Code snippet of background design (index Page)*

Justin also created our own style sheet to design the webpage to look similar to the existing designs of CSUB websites. Figure 1.6 shows highlights the details of how the background of our index page was designed. He created a URL path to set the image in the background, then we made sure to that the image would not repeat itself in the background. He experienced some issues with that earlier in the project due to the size of the image being too small. Then Justin went ahead and made sure the image covered the entire background of the page, positioned it in the center, and blended the image with the color grey to decrease the brightness of the image.

A screen shot of a computer program

Description automatically generated with low confidence

### Figure 1.7. *Code snippet of header and container*

Figure 1.7 showcases the design process of the header and container welcoming the user. As is shown, we wanted to ensure that the colors being used would match CSUB’s color theme. The padding and borders were all adjusted to align the text in an appropriate format as well as adjusting the border edges to make them less sharp and rounder.

A screenshot of a computer program

Description automatically generated with medium confidence

### Figure 1.8. *Code snippet of buttons an footer*

Figure 1.8 displays the design of the buttons, primarily how the colors were implemented. As you can see, the color changes on the button when the cursor is hovered over a button. It’s a very simple technique that Justin has been very experienced in doing and has always found to make the page feel more interactive. The footer follows the same idea as the header in terms of the color scheme and ensure that footer sticks to the bottom of the page because the information on the index page is so little.

## Login page

A screen shot of a login

Description automatically generated with medium confidence

### Figure 1.9. *Login Page*

Initially we wanted to design our own login page and were in the process of doing so, however, after getting feedback from our instructor, it was suggested that we use the already existing CSUB login page. Our application is supposed to be used for CSUB students and made specifically for the university, so after having a couple of discussions amongst ourselves and Dr. Kaur, it was agreed upon to go ahead and implement the existing login page. It looks great and already provides all the functionality required for a traditional login page, so it was a no brainer. As for the code, it was modified to suit our needs, we went ahead and changed the page names to match that of ours so after the user does login, they will be redirected to the correct page. Which we will be discussing in the next section.

## Student Page

A screenshot of a computer

Description automatically generated with medium confidence

### Figure 2. *Student Page (PC Layout)*

The design of the student page was a long and lengthy process. Different ideas were being shared and the layout was constantly being changed. But after spending a significant number of weeks and getting feedback from one another as well as the instructor, it became clearer as to what was expected and what features to implement. As shown in Figure 2, our student page contains a navigation bar that displays the name of our application, a link to the homepage, a contact page, an about us page, and a logout link that signs the user out of the student page and back to the index page. It also displays a message welcoming the user and displaying their name, and a labeled progress bar that shows how close students are to meeting 100 percent of their undergraduate requirements. The progress bar is dynamic and can be explained by other group members that worked on that implementation. It also displays the current user’s information from their student id, name, major, overall GPA, and the number of units that the student has completed thus far. The navigation bar also highlights in yellow the current page that the user is on. So, if they were click on the about us link for example, then that link would be highlighted in yellow instead of the home link.

Moving on to Figure 2.1, this image shows 2 different panels, the General Education Requirements, and the Major Requirements. When either panel is clicked on, it will display different subsections that exist such as foundational skills and capstone requirements, or lower and upper division requirements needed to be satisfied. As shown in Figure 2.1, when one of these subsections is clicked on, it will display a table that will display the Course ID, Course Name, the number units specific to that course, a grade if they have taken the course, and whether that status of the course has been satisfied or not. There are tables for each and every subsection that will display different information.

A screenshot of a computer

Description automatically generated with medium confidence

### Figure 2.1 *Course Requirements*

After implementing the most important features that our group had discussed in Senior Project I, we felt that it our student page was missing something. It did not feel complete is the best way we could describe it and we felt it could use more user interactivity. So, after spending a lot of time thinking and doing research, Justin went ahead and added in a “Career Resources” section. As shown in Figure 2.2, this section contains various different things. The first being the “Job Sites” section that provides the user with different websites that they can navigate to such as popular options like Indeed, LinkedIn, Glassdoor, etc. It even includes a website called Dice for people searching for jobs specifically in tech as well as another website, USAjobs, that is for students looking to work for the government. The next section, “Internship Opportunities”, that allows the user to visit different websites with different interests to search for various internship opportunities, regardless if the student has no prior work experience. The final section, “Interview & Resume Resources”, provides students free and paid resources such as the CSUB CECE or Big Interview, to help students build professional resumes to help them land interviews and help them clean up their interview skills. The career resources section is well researched to provide students with the best possible tools to guarantee them an internship or job. It saves the students time, resources, and helps them start thinking about that next big leap after graduating from college.

A screenshot of a computer

Description automatically generated with medium confidence

### Figure 2.2. *Career Resources*

## Code (Student Page)

A screen shot of a computer program

Description automatically generated with low confidence

### Figure 2.3. *Code snippet of Navigation Bar*

The implementation of the navigation bar was very basic and simple. Figure 2.3 shows how Justin simply defined a nav class using HTML and Bootstrap, put the navbar in a fluid container, created an unordered list and added relative paths to different pages within the application and separate links. Justin also used Bootstrap to add the logout icon that can be seen in Figure 2.

A screen shot of a computer

Description automatically generated with low confidence

### Figure 2.4*. Code snippet of Progress Bar*

Creating the progress bar was a very simple process, probably the easiest thing to implement out of everything in this section. Again, HTML and Bootstrap were used to create the progress bar, all Justin did was create a class called progress, and another class to define a striped progress bar to make it look more dynamic and less basic. Justin set the total value to 100 and the current value to 70 to test out how the progress bar would like once our project was completed.

A screen shot of a computer

Description automatically generated with low confidence

### Figure 2.5. *code snippet of student information structure*

Figure 2.5 Displays the structure of the student information section. It is put into a separate container, used Bootstrap to create the table formatting and HTML to fill the rows with information. As mentioned earlier, this all this information was hardcoded to get a general idea as to what our application would like once it was completed. The back-end will be explained later in this report.

A screen shot of a computer

Description automatically generated with medium confidence

### Figure 2.6. *code snippet of accordion and collapsible elements*

Figure 2.6 displays the general structuring for the collapsible panels that Justin used to make the General Requirements section as well as the Major Requirements section in Figure 2.1. It was a very simple process that only required him to use collapsible panels and an accordion that only displays 1 collapsed item at a time. He also created and labeled separate ID’s for both the requirements sections so that they could be their own separate entities, and so that he could make the page follow the mock image that we had envisioned. This was arguably the most important part of our application simply because this is the main feature that our targeted users would be using the most. Justin had to test it many times to ensure that there were no problems occurring when a user was engaging with the collapsible elements. As shown in Figure 2.7, Justin went ahead and created tables within each collapsible element so that user’s course history could be displayed in well-structured manner. All the data seen here was all hardcoded just so that he could adjust the content effectively for any device. The tables are responsive and does change in size if a user does access the website using a smaller screen. This was all applied using a media query to apply different styles for different screens, which can be seen in our CSS file on our GitHub. Without applying this technique, the formatting of the tables on smaller screen would have not adjusted properly and caused a huge issue.

A screen shot of a computer program

Description automatically generated with low confidence

### Figure 2.7. *Code Snippet of table structure in panels*

Finally, our Career Resources section was implemented by taking advantage of HTML’s section tags and creating a separate container so that all the related content within this section would remain in one spot. Figure 2.8 displays the structuring of this specific section. As is shown, Justin went ahead and created even columns for each separate section such as the jobs, internships, and resume/interview resources so that they would be displayed evenly. Justin also structured the links to specifically open a new tab with the link that was selected by the user so that they did not have leave the webpage and could simply have multiple different tabs up and be ready for them to access while remaining on the page. Justin made sure to put the links within lists so that he could add brief descriptions to each individual resource. He also edited the links and changed the colors to let the user know what links were accessed and which ones have not. As was mentioned earlier, our overall goal for the front-end design of the application was to make it as simple as possible to navigate to ensure a great quality experience.

A picture containing text, screenshot

Description automatically generated

### Figure 2.8. *Code Snippet of Career Resources Section*

## About Page

After finishing with the design of the student page, we also decided to create an about us page (Figure 2.9). This page essentially provides a brief description of the goal of our project, lists the names of everyone involved with the creation of our project, a contact section with the images, names, and roles of each team member. We wanted this page to be a place where the user can meet the team behind the creation of this wonderful application, get a better understanding of how the website was created and to be able to directly contact any team member via email (Figure 3). We also decided to add in a featurette section to the page to not only enhance the quality of the page, but to show the user, especially new users, the main features of our application. We also made sure to add in images in this section so that our website could feel more visually appealing and to add some “life” to our webpage. We figured what better place to add finally add some images to our application than here (Figure 3.1).

A screenshot of a computer

Description automatically generated with low confidence

### Figure 3. *About us*

A screenshot of a computer

Description automatically generated with medium confidence

### Figure 3.1. *Developer Team*

A picture containing screenshot, text, computer, software

Description automatically generated

### Figure 3.2. *Featurette Section*

## Code (About Page)

Figure 3.3 showcases the implementation of the Developer team section in the about us page. Just put the content in a container, split the content into even columns using Bootstrap and the overall design of the individual sections of each group member was created setting a div class as a “carousel-item”. HE also made a separate div class, “contact”, so that separate the link from the other content and fill it with the color yellow to follow the theme our application. Justin also used Bootstrap to add an envelope icon next the link so that design could look more modern. As for the featurette implementation, he defined a class with a container that specifically creates the formatting of the featurettes. He also defined a separate span class, “text-muted” (Figure 3.4), that allowed him to have two different color schemes for the heading (Figure 3.2). Finally, Justin simply made 2 separate columns dividing the wording and image within each featurette. That is the end of our final implantation of the front-end of our application.

A picture containing text, screenshot, font

Description automatically generated

### Figure 3.3. *Code snippet of developer team structure*

A screen shot of a computer program

Description automatically generated with low confidence

### Figure 3.4. *Code snippet of featurette section*

# Works Cited

*Bootstrap 3 Tutorial*. (n.d.). W3schools. Retrieved May 15, 2023, from https://www.w3schools.com/bootstrap/