

The Buff Buster

Project Report

CSCI 3308

Team 015-01

Ryan Brace, ryanbrace2003, rybr9677@colorado.edu

Sean Esslinger, seanesslingerr, sees3205@colorado.edu

Boston Abrams, bostonbuff, boab2355@colorado.edu

Sania Joshi, saniajoshi9, sajo1826@colorado.edu

Table of Contents:

Table of Contents:	2
Project Description	2
Project Tracker	2
Video	3
VCS	3
Contributions	4
Boston	4
Sania	4
Ryan	4
Sean	4
Use Case Diagram	5
Wireframes	5
Test Results	8
Deployment	9
ReadMe.md	9

Project Description

Our team has created a helpful tool for CS students to replace the horrible degree audit tool. It allows students to more easily understand what courses they must take and which they have already taken. This helps students plan their degree completion and make sure they can graduate.

First users must register for our website. They must provide their name (first and last) and a password for the website. They will then be shown their “identikey” that is randomly generated to simulate the way that login would work if integrated into CUs system. Then using this “identikey” and password they can login. At this point there is no data about this user in our database so all the fields are blank. To receive the degree audit data we require our user to upload their degree audit to our site. This is done through right clicking on the degree audit and clicking download. Then this .html file can then be uploaded onto our website. We then parse through this information, showing it on multiple pages that are only accessible when the user is logged in. We successfully worked together as a team to design, create, test, and deploy this website.

Project Tracker

We used the Github Project Board to track our progress.

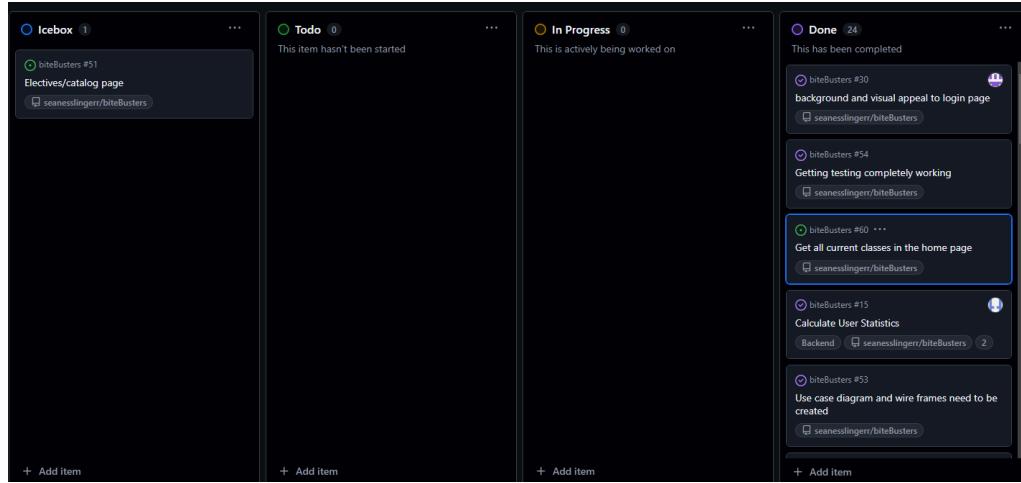


Figure 1: Our Github Project Board, See below for full list of completed tasks

Link: <https://github.com/users/seanesslinger/projects/3/views/1>

Video

Buff Busters has prepared a video of our project!

<https://www.youtube.com/watch?v=d9NdNmHKAZo>

VCS

We used Github as our Version Control System. The link is provided below

<https://github.com/seanesslinger/biteBusters>

We need to add the video to github

Contributions

Boston

The Tools I used were mainly Handlebars and NodeJS.

My contribution to the project was almost solely on the backend/database side. I created a custom javascript function that could read in the user's Degree Audit, store it temporarily, and look through it looking for key lines that are near lines that contain the required information that we want from the degree audit. I then wrote the rest of the route that would store this data in our teams database. Then to get this information on the page I used handlebars to list the required information onto the front end.

Sania

My contribution to the project was primarily on the Frontend. My main goal with the front end was to make sure that it was easy for users to understand and use (unlike the original buff portal). I did a lot of HTML and CSS work on all of the different pages on our site - including the login, home, stats, and logout page of our website. I also used resources such as bootstrap because it enhanced the look of our website overall. I also helped with a lot of last minute editing/changes at the end of the project to make sure it was all working and presentable.

Ryan

My contribution was all over the place. When we started off deciding jobs I was chosen as the alternate and I did just that. At the beginning I was responsible for most of the frontend formatting including the HTML and CSS work. I helped to create the style for the home, login, and some of the stats page. I also helped my team get all the testing correctly lined up with our API routes so we had something to rely on, and something to help debug with. Once these tests were figured out, I helped edit some of the API routes Sean created in order to get them running and functioning. This included using a lot of Chai and mocha. Finally I helped to get Render up and running. This was mostly a joint effort and Sean deserves major credit for getting it completely running.

Sean

My contribution was mainly in the backend of the website. At the beginning of the project I created all of the API endpoints for the website as well as the original docker.yaml. The original 'get' endpoints to the home, stats, registration, logout, and login pages. I designed the registration page to splice the first and last name and randomly generate four numbers to create an 'indentikey'. Then I created the session authentication so that when users entered it opened into the user session. After Boston created the parser for the degree audit HTML, I created the stats page that created all of the degree requirements for the CS major. I used handlebars and the SQL database to post all of the information on the stats page dynamically. For the cloud hosting I created the working render by rearranging the files and its referencing so that the website could be uploaded publicly.

Use Case Diagram

Here is our final use case diagram (Figure 2)

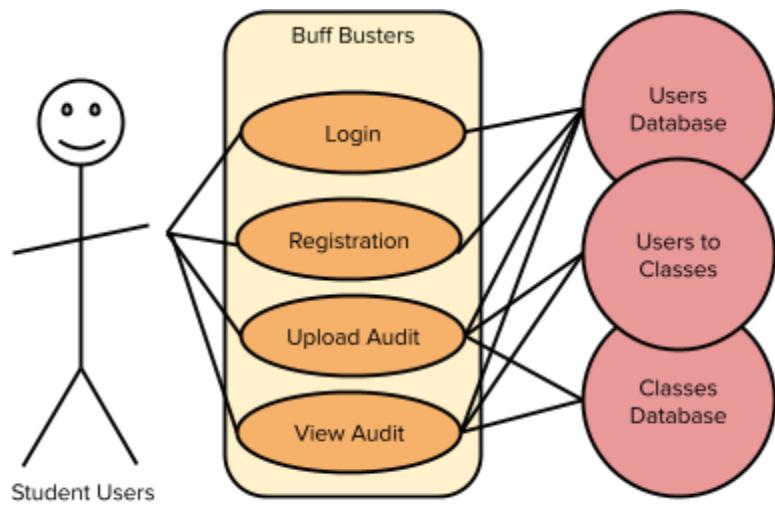


Figure 2: Use Case Diagram

Wireframes

Here are the wireframes of the final version of our website. (Figures 3, 4, & 5)

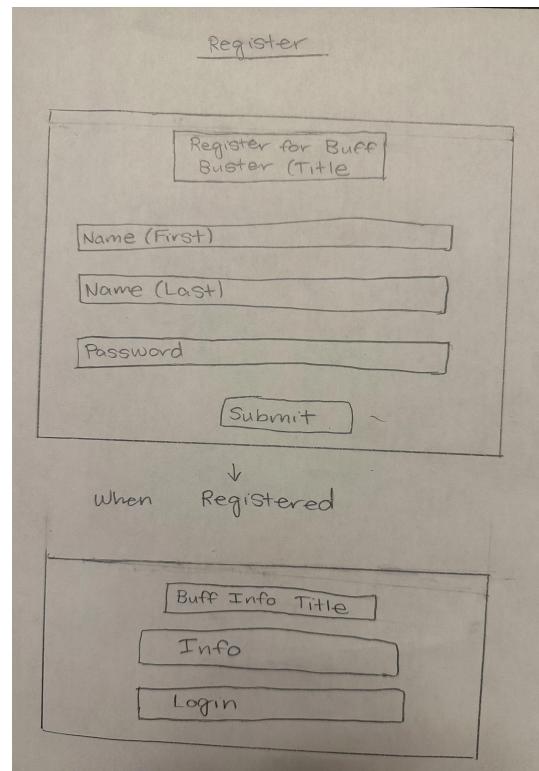


Figure 3: Wireframe of the Register and Registration information pages



Figure 4: Wireframe of the Login Page

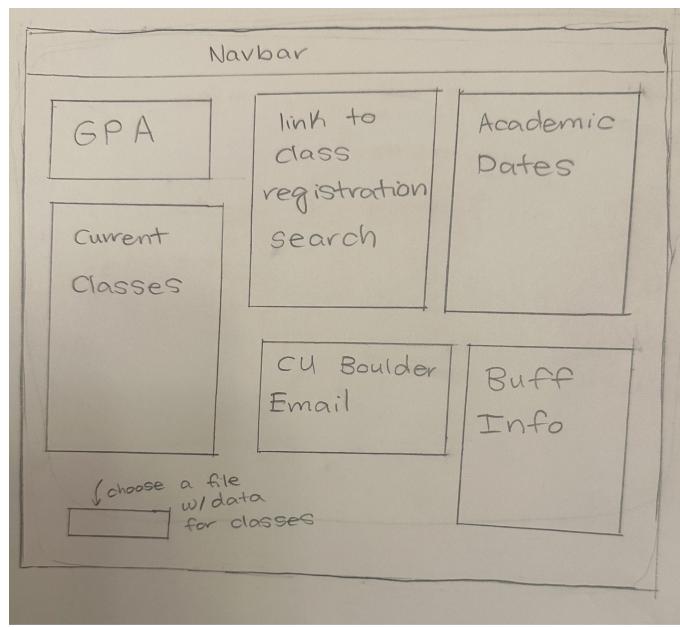


Figure 5: Wireframe of the Home page

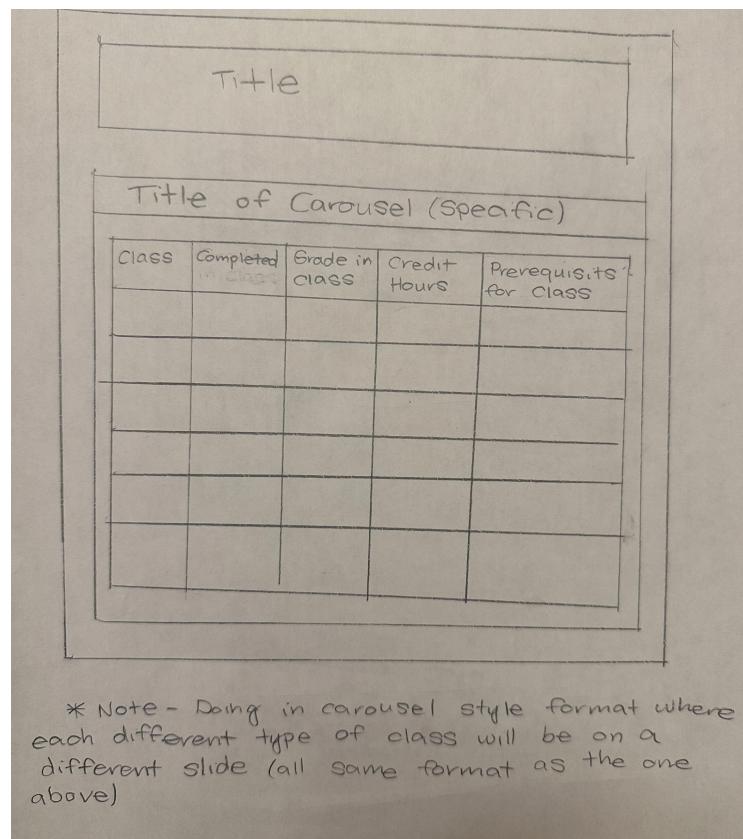


Figure 6: Wireframe of the Stats page

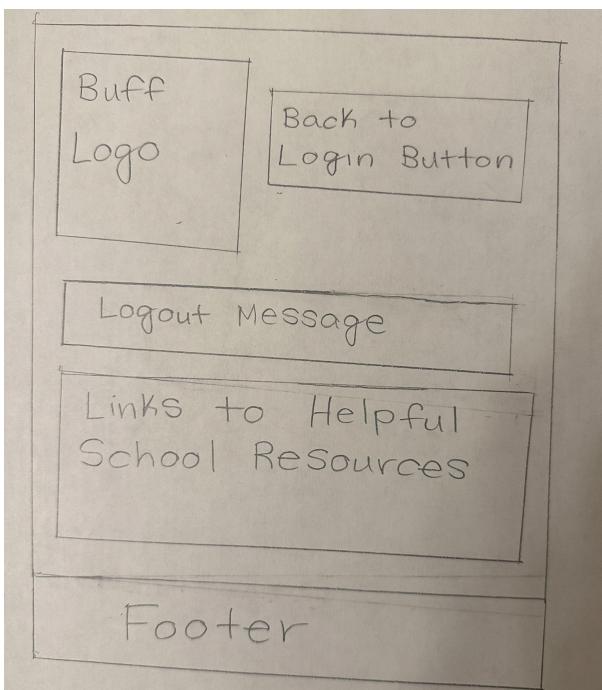


Figure 6: Wireframe of logout page

Test Results

We chose to have each of our users test these four questions:

1. Logging in
2. Uploading their audit
3. Reviewing their information on the two pages
4. Logging out

User #1: Tinius Klaveness

This user was quite enamored with our website, he first clicked on the footer links and checked to see how the website functioned. The user was using his phone which was relatively new as most of the tests I have been doing were on the computer. The user was impressed with the compatibility on the phone. After looking around the login he registered and made his way to the home page. My friend is an engineering/computer science student so he was a much more ideal candidate for this test than most. The one problem he ran into on his phone was downloading the degree audit. Downloading this proved to be a difficult task, and was the most time consuming part about this test. Finally once done, he uploaded the degree audit and was impressed! He was originally confused about his GPA, thinking it was only an engineering GPA, as I had to explain it was a cumulative GPA. After looking at the organization of the classes he had and reading some of the other informational cards, he navigated to the stats page. At first he didn't see the arrow for the stats page and deviated a little from expected behavior by waiting and seeing if the carousel would move, which it did. He scrolled through, and enjoyed it, as he finally logged out. Overall, with small deviation Tinius stayed near the expected behavior, and from him we learned we needed to make the carousel arrows on the stats page more visual. Overall, an ideal tester!

User #2: Hayden Thomas

The user got right to business, they immediately registered and went to login. He found it very useful for the page that showed up about the identikey and allowed him to login. Sadly, my friend was a business major, so he had trouble seeing the full scope of this website. He ended up clicking around and trying every one of the links posted and found they were effective. He did run into an issue clicking on the academic calendar link, because it asked him to fill out his true identikey, and he tried with the one made up, which obviously did not work. Overall despite the user not having a computer science degree he had an orderly fashion to moving through the website as one would expect, and he did not appear to deviate from the original structure. The changes we made were trying to set up a larger database for all majors, but this proved difficult as I tested Hayden too late in the process.

User #3: Brayden Bacon

The user originally tried to use their real identikey on the login page which obviously did not work but made sense because it is a degree audit page associated with CU Boulder. But then

they used the register page to create an account and login. From there it was sent to the page with the generated indentkey and email. The user then logged in and went to the home page. On the home page there is no information about classes since no file was uploaded so I had to help upload one of the html files. Then the user viewed the stats and home page. The user was consistent with the use cases because I think the site is a little trivial, however the site worked well in the visual applications of a more simplified degree audit.

User #4: Finn White

The user went straight to the register page, and properly created an indentkey on the website, but then skipped past the information page and did not remember the user information, so they needed to look into the database to remember it. Once the user was on the page they uploaded the file and viewed the page. The user navigated through the links and the stats page and looked at the information. The user used the website as intended and enjoyed the display of information. The point of the website is somewhat trivial since it is just an audit, but the website was used as intended by the user and was successful in its intent of a simplified audit.

Overall from all of these users we realized we need to include instructions on how to use our website or make the website more clear on how to use it. If we continue this project we will definitely make these changes so new users will be able to understand how to use it.

Deployment

We used Render to host our application for public view.

Link: <https://bitebusters-ktmr.onrender.com>

(Be patient it is slow to load)

ReadMe.md

(Also included on Github)

Our team has created a helpful tool for CS students to replace the horrible degree audit tool. It allows students to more easily understand what courses they must take and which they have already taken. This helps students plan their degree completion and ensure when they reach the end of their senior year that they can successfully graduate.

Directory Structure:

Milestone Submissions

Project Source Code

Src

 Init_data

 Create & insert

 Views

 All our .hbs views

 index.js

Test

Test JS

Team Meeting Logs

Test

Test JS (deprecated)

How To Run:

1. Clone the github repository.
2. Run: “docker compose up” in /ProjectSourceCode in (in WSL or a Unix Terminal)
3. Enter “localhost:3000/login” into you web browser to see the website (Google Chrome Preferred)

Link to Deployed Application: <https://bitebusters-ktmr.onrender.com>