CSE 134B Homework 5

David Lee A09548716
Nikhilesh Aiyer A11247204
Eddy Kim A11690057

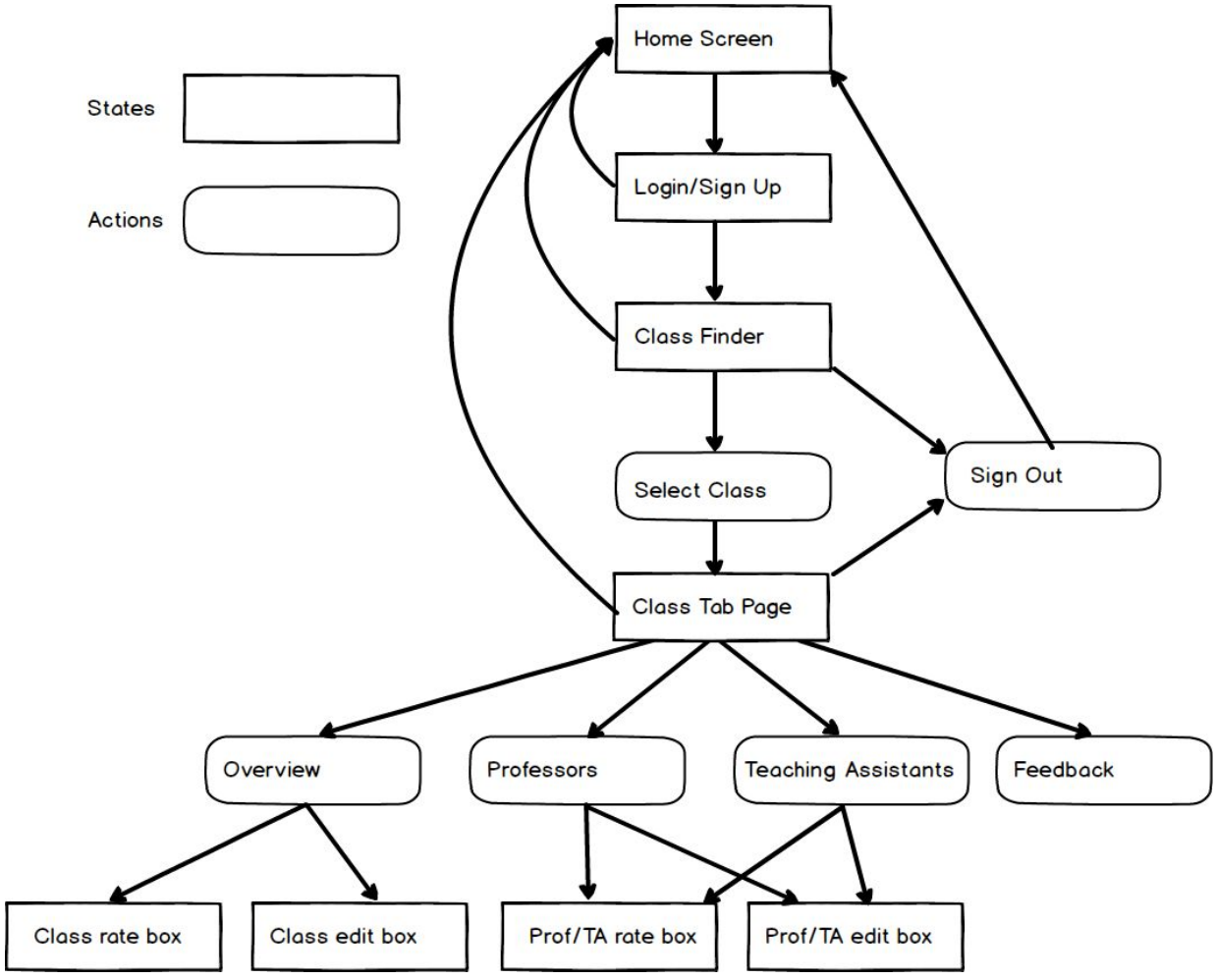
Introduction

This quarter we were tasked with building a web application using different technologies to understand delivery, security, and the medium of the web by creating a *TopicDex* of our choosing. Our team decided to create a Wikipedia-like ClassDex for UCSD CSE classes that allows the user to create, read, update, and delete classes and class related information, using HTML, CSS, and Javascript technologies.

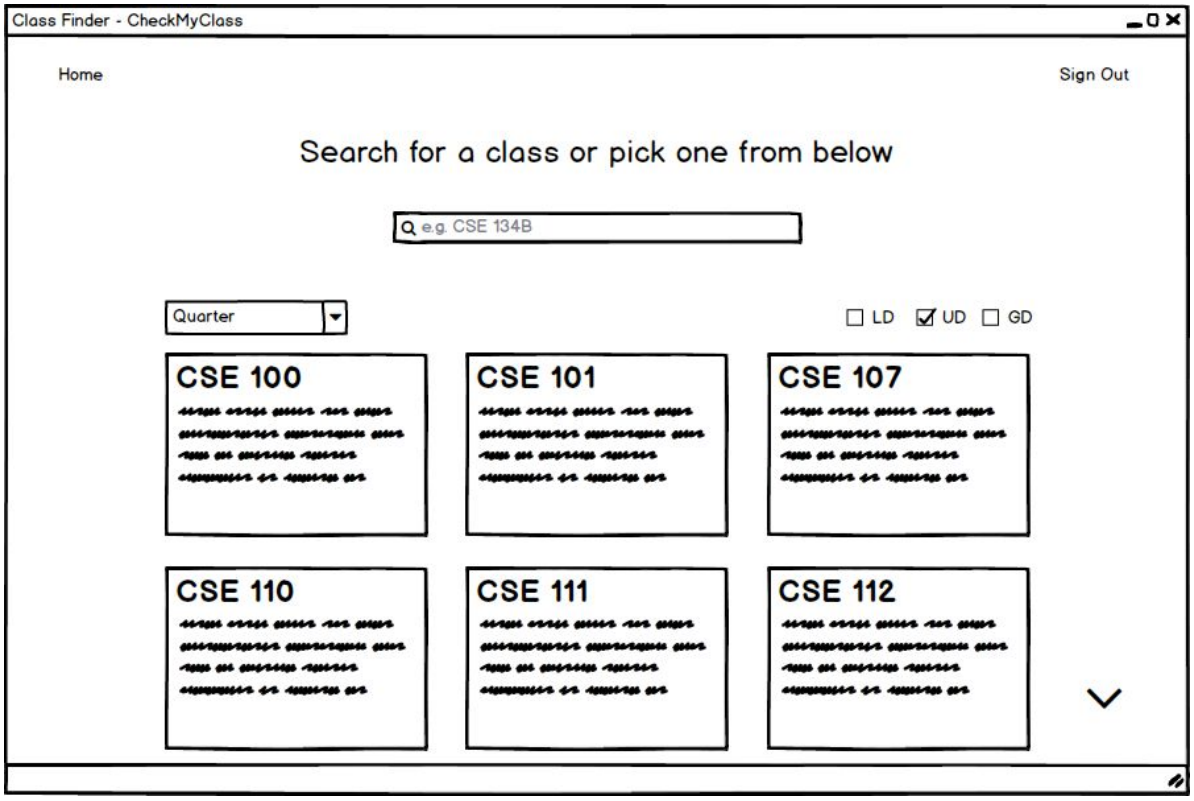
Development

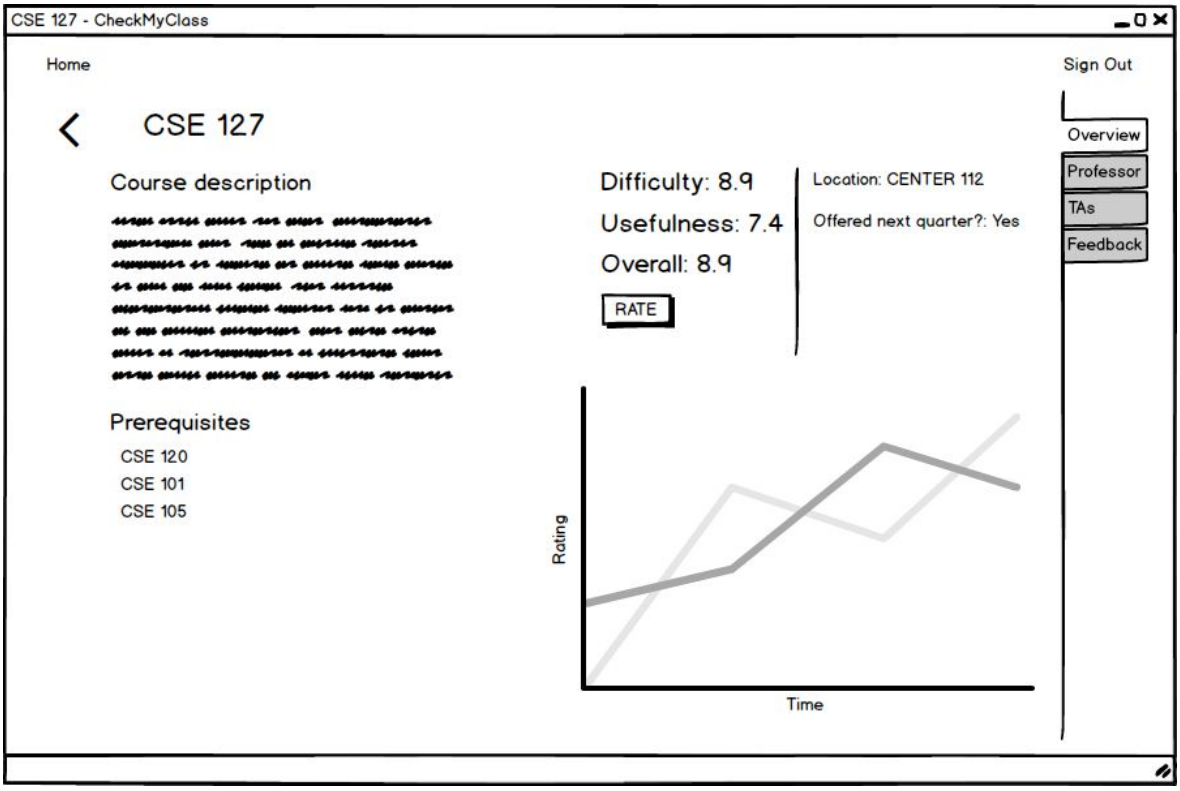
Since we had our idea for what dex we wanted to develop for the quarter, the homework assignments given to us would provide a good development plan for our application. In the first homework, we were tasked with critically thinking and documenting the uses and general goals for the application. In the document, we wrote out the motivations for our idea and features our application will provide, while recognizing our users to be students, professors, and teaching assistants. We also created a flow diagram and wireframes to get a preliminary idea of what we are looking to implement in our application.

Flow Chart Diagram



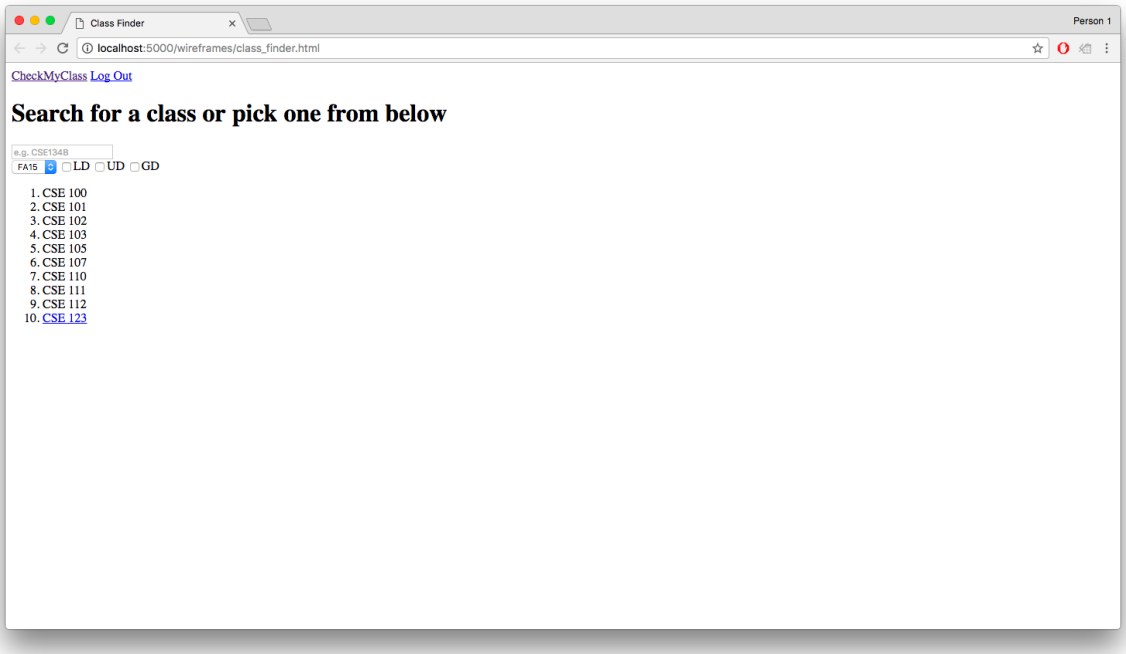
Sample Wireframes

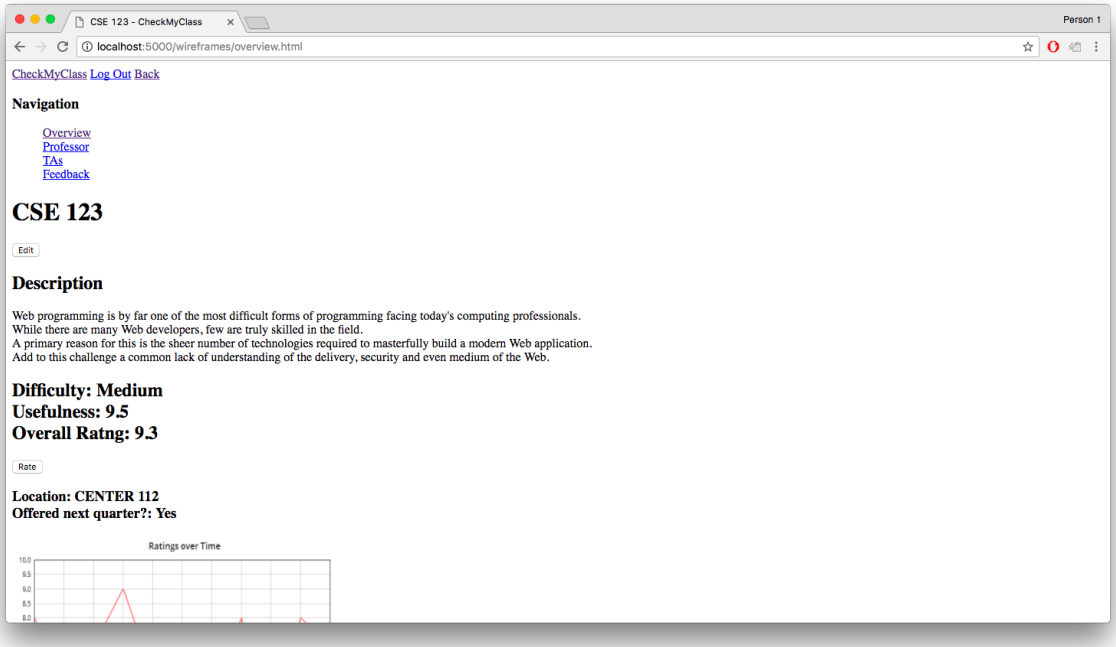




For the second homework, we were tasked with creating our wireframes using only HTML, and to create an HTML document using all of the HTML5 tags we did not use in our wireframe. The purpose of the additional document was to get ourselves more familiar with HTML5 and the capabilities of the tags. We chose to focus on clean HTML code that will be easy to style, as the purpose of this assignment was to set up our HTML code for CSS styling.

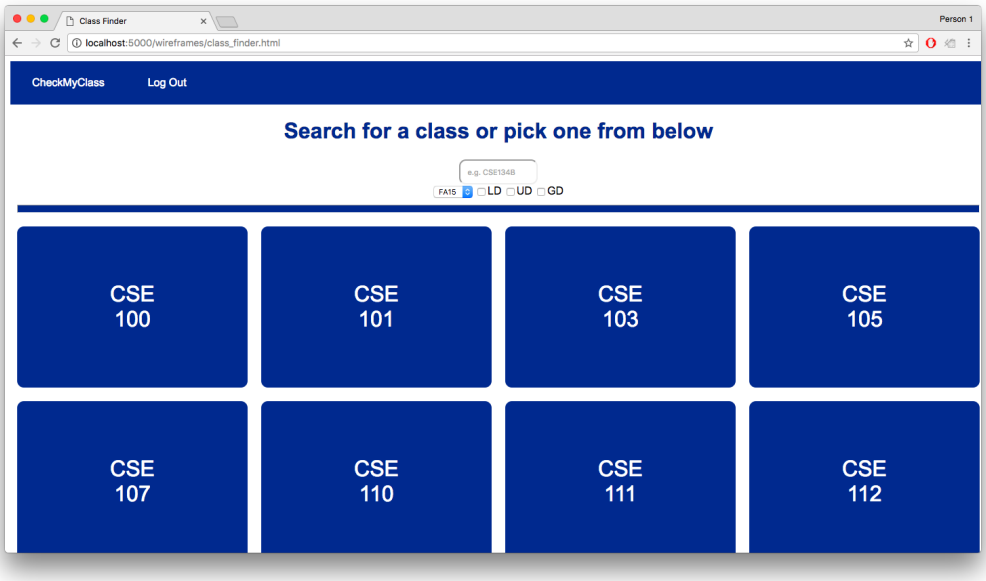
HTML Only Wireframes

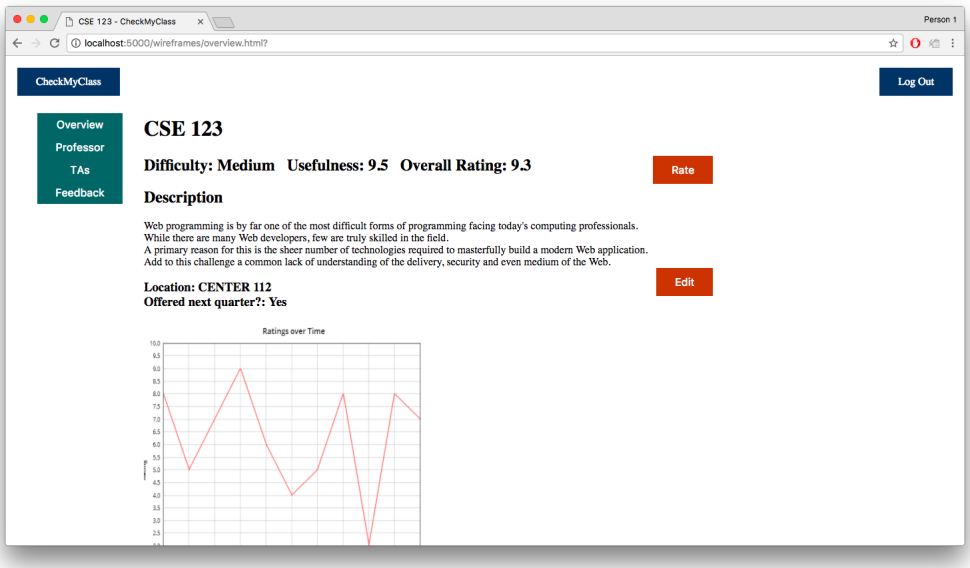




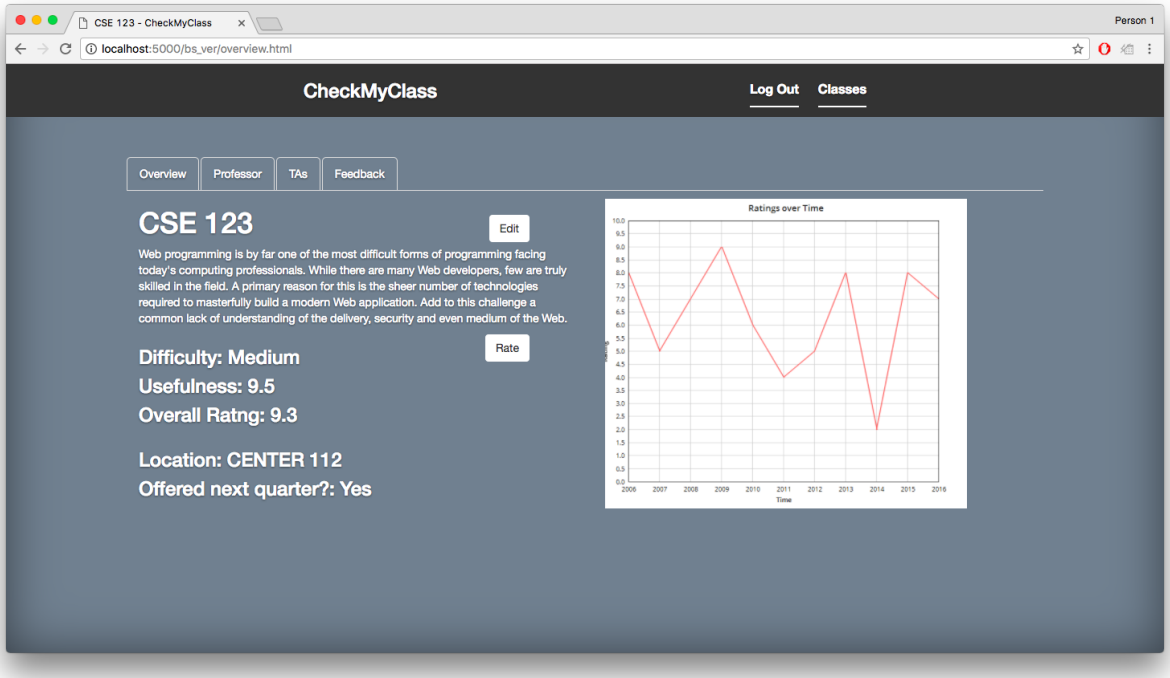
In the third homework, we had to take our HTML only wireframes we made for homework #2, and make two different versions of our web application, using strictly vanilla CSS for one and a framework for the other. After we completed making both versions, we analyzed the differences between vanilla CSS and framework CSS, and concluded that using Bootstrap was more beneficial in terms of aesthetics and ease of use for resource heavy pages. We also conducted some performance testing on a mobile device for the two versions, and found that the bootstrap did take longer to load than the vanilla CSS version, as expected. For this assignment, our code was inconsistent, and can be seen in our vanilla CSS version. We didn't pay too much attention to the vanilla CSS version, as we knew that we would be using the bootstrap version moving forward.

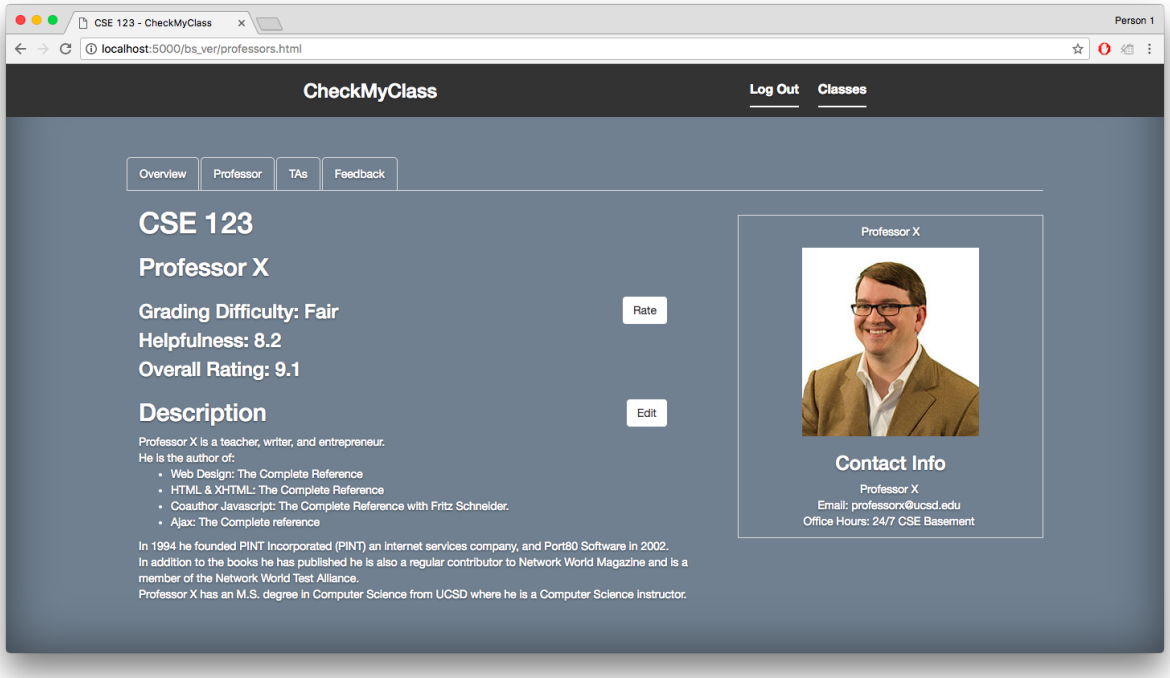
Vanilla CSS





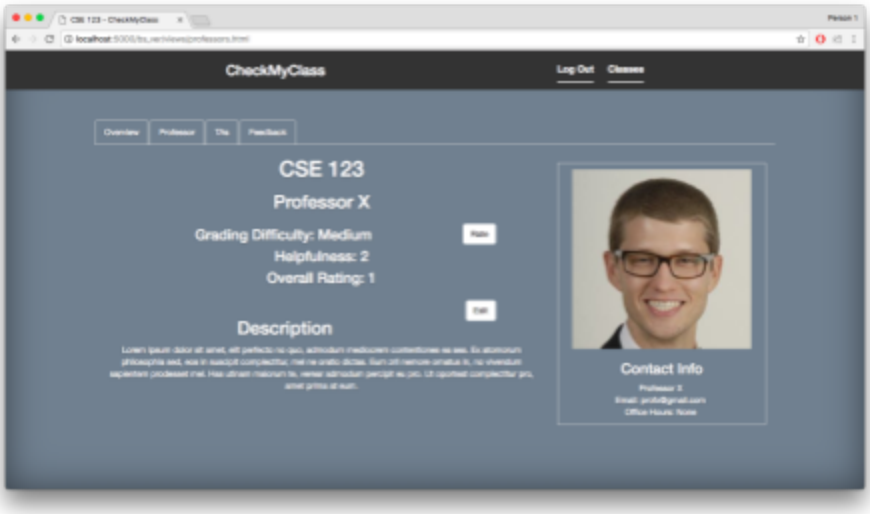
Bootstrap CSS



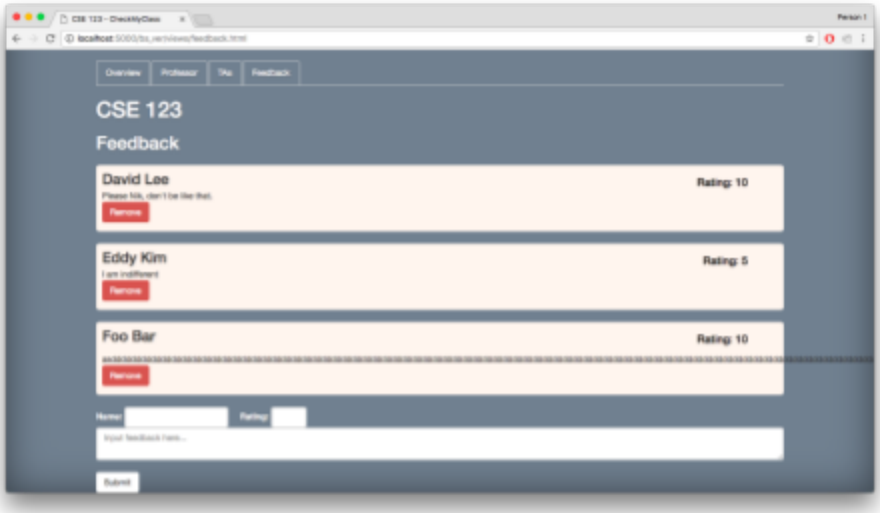


For the fourth homework, we needed to implement create, read, update, and delete functionality to our web application along with authentication and asset management. We were able to implement authentication using the Firebase API and TA sample to allow users to sign up using their email or Google account information. For the CRUD functionality, we used VueJS to handle our database requests. In our application, the user is able to edit and rate the class, professor, and TA information, and is also able to leave feedback about that class. We also did some performance testing to see how our web application performance differs once we added CRUD functionality. The results showed that logging in was the longest process for our web application.

CRUD Functionality



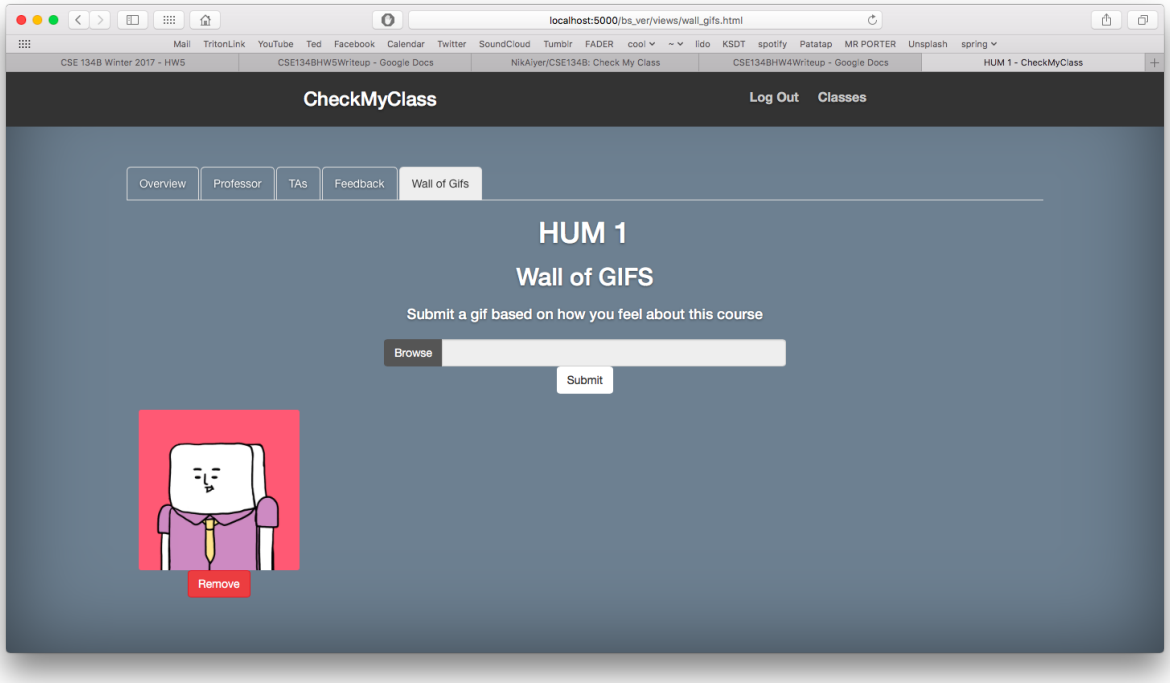
Users are able to rate the professor and edit information (Read, Update)



Users can leave feedback about the class (Create, Read, Update, Delete)

For our last homework assignment, the main focus was to polish our application and it's functionality. Our asset management CRUD was not functioning properly at the time of turn in, but it was an easy fix that got lost between commits. We polished our UI by adding additional styling and animations, and also extending our CRUD functionality to allow the user to add and delete classes. In our previous assignment, our application did not fully include asset management, so in this homework, we added "Wall of GIFs" for users to add GIFs about their feelings for the class.

Wall of GIFs



Performance

File (including any .js and .css)	Time (average in milliseconds)
index.html	416ms
login_signup.html	488ms
class_finder.html (w/ log in)	2.81s
class_finder.html (w/o log in)	514ms
overview.html	415ms

class_edit.html	422ms
class_rate.html	434ms
professor.html	421ms
prof_edit.html	387ms
prof_rate.html	544ms
ta.html	407ms
ta_edit.html	525ms
ta_rate.html	584ms
feedback.html	377ms

Overall performance of our application has improved compared to Homework #4. Due to the minified and bundled files, our application was able to load a little bit faster, even though we had more resources to load. If we had more time, our team would've researched more on performance optimization in order to make our web application faster.

Final Thoughts

Though we are proud of our final project, there are still many features we would have liked to implement. One feature would have been a way to average out all the ratings given to classes, ta's and professors. This way each user can post their own ratings, and have all the ratings averaged out and displayed on the class page. We also could have added more CRUD functionality, specifically for images and other data. Some other features we would have added are analytics, more filtering options, as well as some overall UI improvements.

Some parts of our code still have some bugs that need to be fixed, such as our edit boxes. Our edit boxes have had some problems with image uploading, as well as problems with uploading single pieces of data at a time. To use our edit boxes, you must fill out all the info and overwrite the existing data. If we had more time to work on this project, we would try to find a solution towards this problem.

All in all, we have accomplished a great deal with this project, as well as learned much about JavaScript, VueJS, and Firebase. These practical skills, as well as the conceptual knowledge learned in class and through trying to improve our project, will provide much support for us in our future as software engineers.