

# Study Seat: Final Report

*CS 467 Capstone Project: Fall 2019*

**Sean Hinds, Hsiang Lo, John Yoon**

## **I. INTRODUCTION**

After several months of brainstorming, developing, and conferring, the results of our efforts have finally culminated into our “Study Seat” web application. Our team endeavored to create a website that grants a user the functionality and accessibility to search for worthwhile study locations in the user’s vicinity. The process of implementing this project was a rewarding and enriching one, in part due to the complexities and difficulties involved with learning and applying various new tools and technologies, of which there were several. In that same theme, we had to quickly and adeptly figure out a range of technical matters ranging from just setting up accounts for certain API accessibility to adopting new programming paradigms as we coded.

However, at the end of the day, our project is fundamentally a web application. After registering and logging into their account, the user sets the location to where the user is currently at, or even wishes to later travel, and based on that location the nearby study spots are displayed. There are options to choose different types of study environments, ranging from cafes to bookstores to libraries. From there, depending on which study spot the user clicks on, the relevant information regarding the study spot is shown such that the user can make their decision on which study spot to check out.

## **II. SETUP & USAGE**

*There are two options for viewing the functionality of the StudySeat site, by visiting the instance hosted on AWS. The site has been tested on Google Chrome and Safari:*

### **OPTION 1: Build and Run from Source**

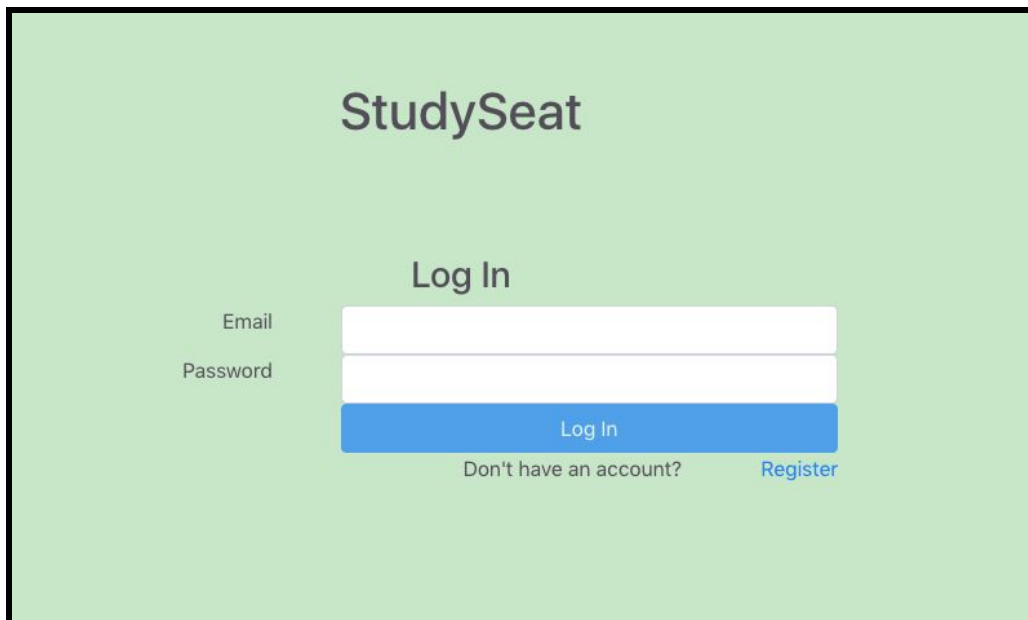
- Will need to have npm and Node.JS installed
- Open a terminal and navigate to StudySeat/
- \$ npm run build
- \$ npm start
- Navigate to <https://localhost:8080>

### **OPTION 2: Hosted Link**

- Input “<https://study-seat.com/>” (or localhost:8080 if running locally) into a web browser like Chrome and press enter.

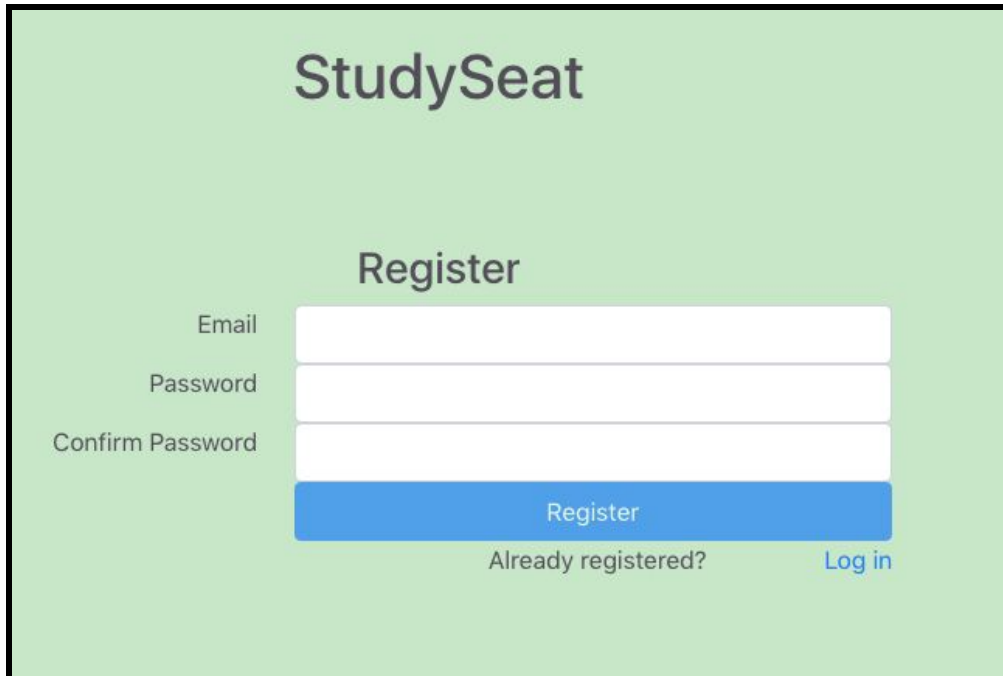


- You will be directed to a login page. If you have not already registered for a user account, then select the register button.

A screenshot of the StudySeat login page. The page has a light green background. At the top center, the text "StudySeat" is displayed in a dark grey font. Below it, the text "Log In" is centered. There are two input fields: the first is labeled "Email" and the second is labeled "Password". Below these fields is a blue button with the text "Log In" in white. At the bottom, there is a link that says "Don't have an account?" followed by a blue link that says "Register".

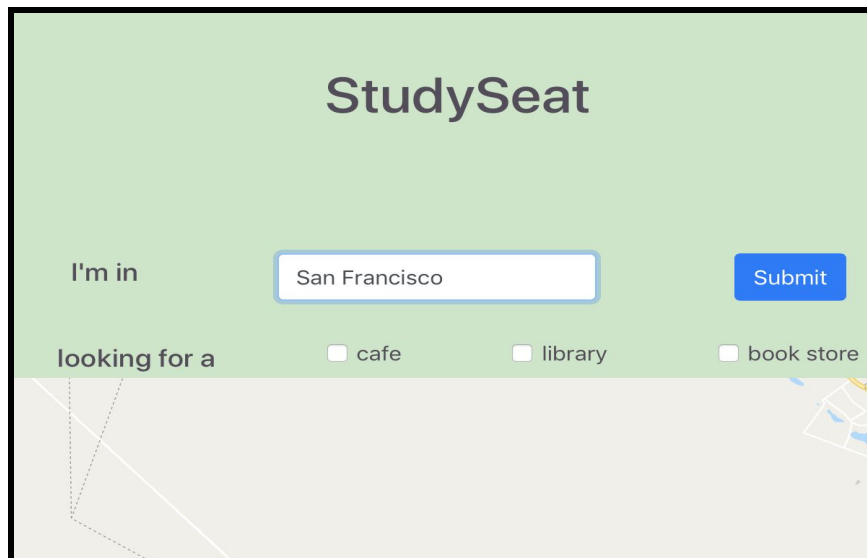
- On the registration page, you will enter your email address and password that will be connected with your user account (IMPORTANT\*\*\* Please make sure that your password contains at least one special character such as an exclamation point “!”).

Notice that the register button does not become enabled until an email and password have been entered. Upon registering, you should be automatically logged in.



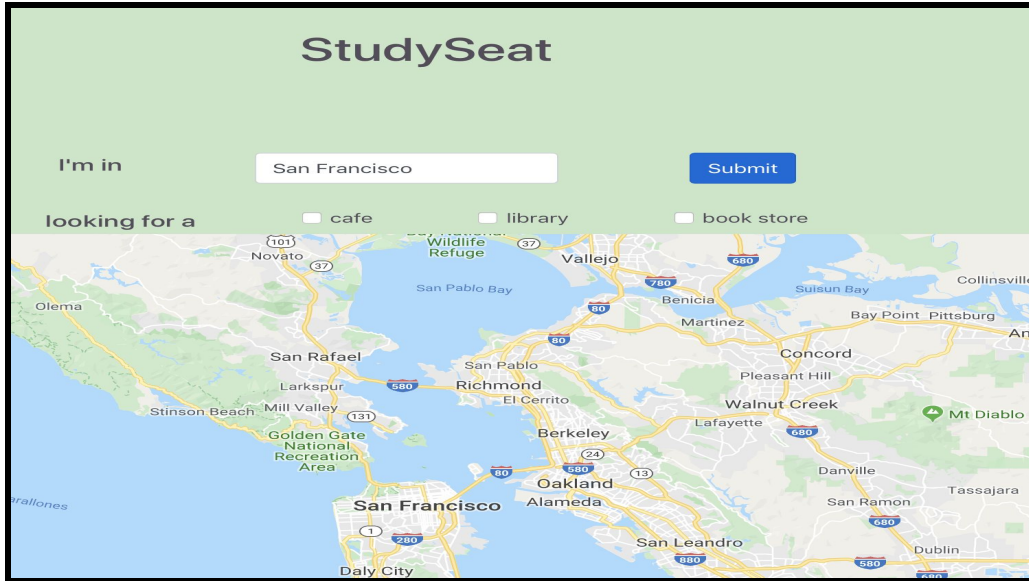
The image shows a registration form for StudySeat. The form is titled "Register" and is set against a light green background. It contains three input fields for "Email", "Password", and "Confirm Password". Below these fields is a blue "Register" button. At the bottom of the form, there is a link "Already registered?" and a link "Log in".

- Input a location such as “San Francisco” in the blank text box following the “I’m in” and click the Submit button.

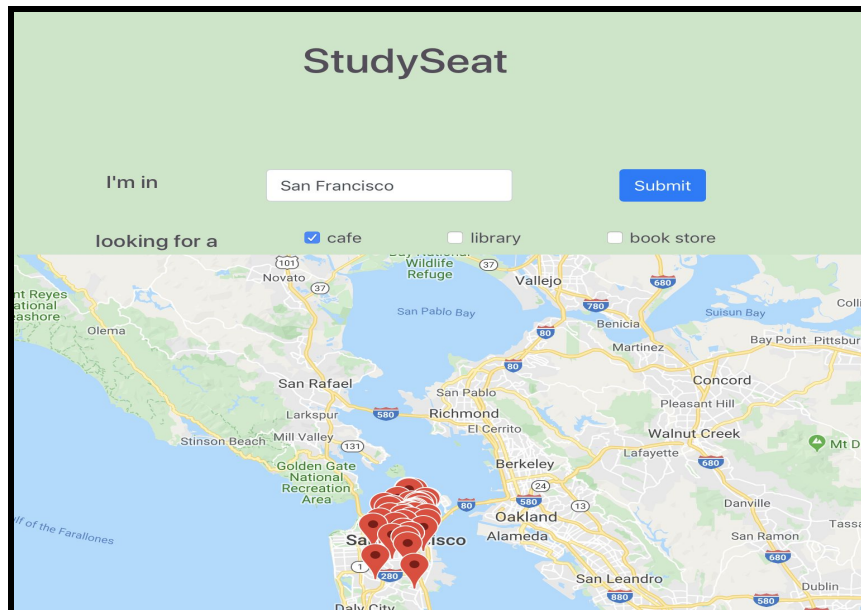


The image shows a location selection form for StudySeat. The form is titled "StudySeat" and is set against a light green background. It contains a text box labeled "I'm in" with the value "San Francisco" entered. To the right of the text box is a blue "Submit" button. Below the text box, there is a section labeled "looking for a" with three radio button options: "cafe", "library", and "book store". At the bottom of the form, there is a map showing a street grid.

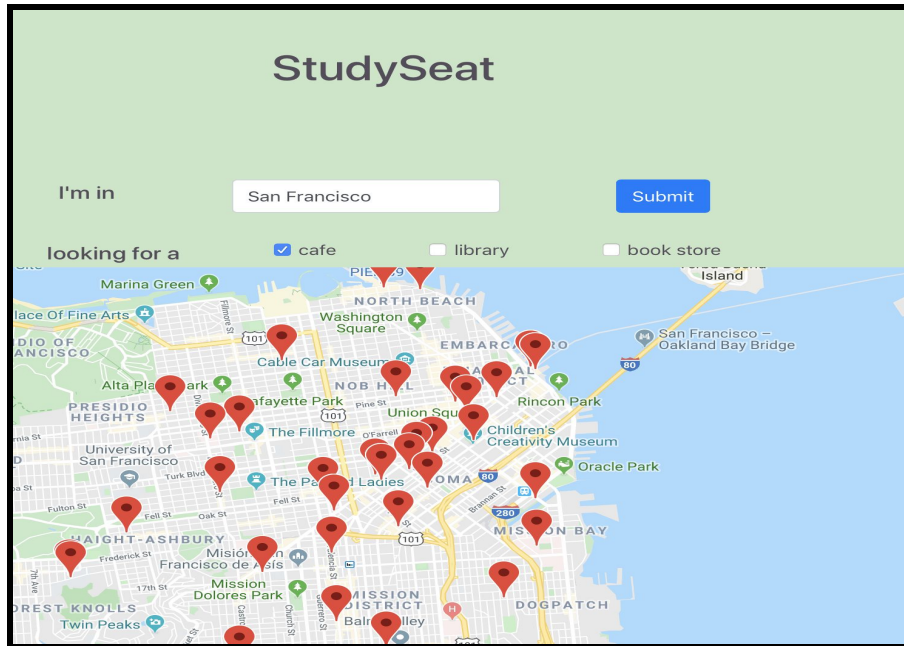
- You will now be navigated to the location that you have just inputted, and will have the option to select from three check boxes, including: cafe, library, and bookstore.



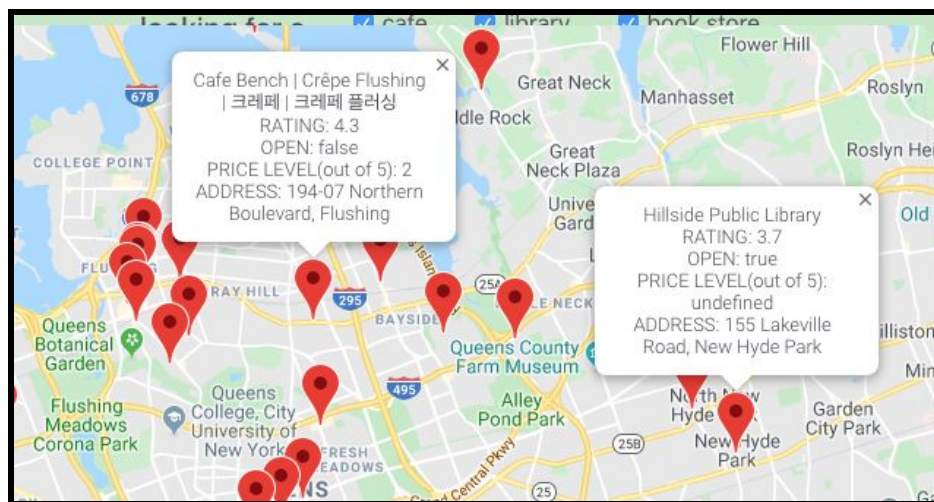
- After selecting from any number of the check boxes, the map will automatically populate or depopulate red markers that correspond with cafes, libraries, or bookstores.



- Toggling the checkboxes should cause the markers for a given category to appear and disappear.
- You can zoom in closer into a specific area or red flag for further inspection, and similarly, you can zoom out for a broader view of an area. This can be done by double-clicking the map with your mouse.

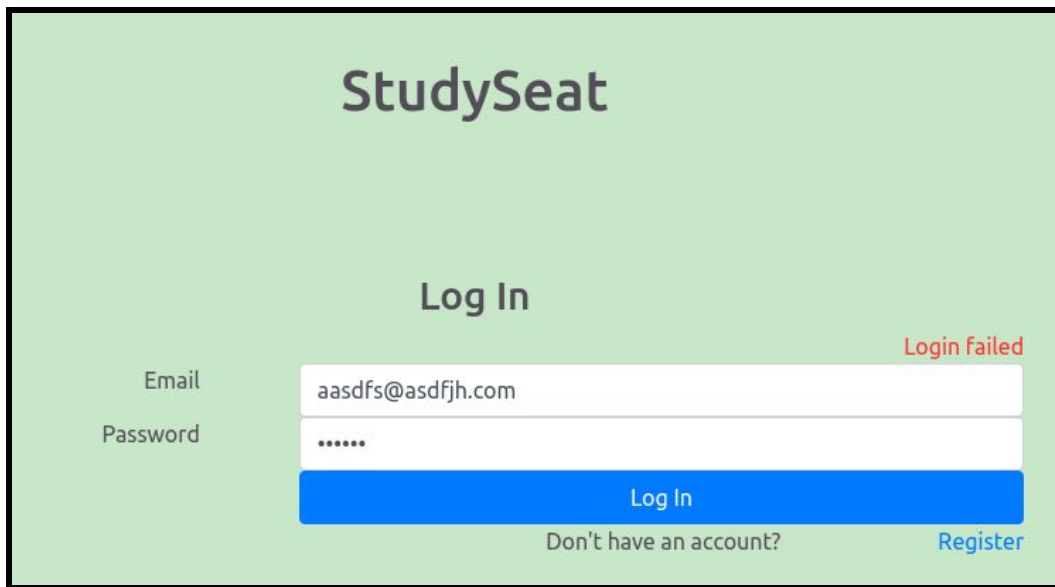


- To see further information regarding a specific study spot, just click the matching red flag and an information box will appear providing it



- Try logging out by pressing the log out button

- Try logging in with a random email and password (which should fail). After a few seconds (auth0 is slow for some reason), you should see this red “login failed” text.



The screenshot shows a web page for 'StudySeat' with a light green background. In the center, there is a 'Log In' section. It contains two input fields: 'Email' with the value 'aasdfs@asdfjh.com' and 'Password' with masked characters '.....'. Below these fields is a blue 'Log In' button. To the right of the email field, the text 'Login failed' is displayed in red. At the bottom of the login section, there is a link 'Don't have an account?' and a blue link 'Register'.

- Try logging in with the same username and password you registered above, you should be redirected to the map.
- To exit the application, simply close your web browser
- If running locally, Ctrl+C in the terminal will shut down the server

### III. SOFTWARE SYSTEMS

#### SERVER

We hosted our load-balanced Node.JS server instance on AWS Elastic Beanstalk. The server is written in TypeScript (described below) and compiled to JavaScript. The server serves the client bundle when the browser makes a GET request to the port the server is listening on.

#### CLIENT

Our React client is a nested project within our main project repository. The client project contains all of the TypeScript, HTML, and SCSS which is compiled into the client bundle to be served by the server to the browser.

## **GOOGLE MAPS API**

We relied heavily on Google's APIs and systems. Our client interacts with the Google Maps and Google Places APIs to manage the map object and populate it with info windows. We had intended to allow data from Google Places to be stored in a database and associated with user accounts, but encountered unforeseen complexities in setting up the database and ran out of time.

## **AUTH0**

We used Auth0 for authentication. Our client builds requests through the login and registration forms, which are passed to the server. The server then creates new HTTP requests from this information to send to Auth0. The server uses the response from Auth0 to determine whether, for example, the user has successfully logged in or registered.

## **IV. TOOLS, APIs, LIBRARIES, ETC.**

### **GOOGLE MAPS API**

The Google Maps Platform consists of a series of various APIs that allowed our team to leverage Google Maps' numerous functionalities and data. Specifically, we predominantly focused on the subsets of the Google Maps Platform called Google Maps' Places API and Google Maps' Javascript API. This is the interface that granted us the ability to display to the user the various potential study spots available to a user in their nearby area.

### **AMAZON WEB SERVICES**

Amazon Web Services is how we were able to host our web application. Specifically we used the service that they offer called, Elastic Beanstalk. Out of the vast series of Amazon web services available, it is the most concise, quick, and simple to use for basic web applications, and thus it suited our purposes perfectly. There were a few issues regarding the deployment of our application, but once we resolved that, it was an ideal service for our requirements.

Amazon Web Service also provided two essential tools that ensures client's privacy. Route 53 enabled us to transfer the domain nameservers from GoDaddy to Amazon. It is then certified for HyperText Transfer Protocol Secure or HTTPS. THis added security protocol protects the integrity of the website.

## **AUTH0**

Auth0 is a free authentication service which allows the registration of users and handles storing of passwords.

## **HTML & CSS & JAVASCRIPT**

Fundamental, basic, and yet essential tools for web development. We took the javascript to a somewhat different paradigm, which is further delved into in the next section.

## **REACT & TYPESCRIPT**

React is a Javascript library originally developed by Facebook that is known for effectively managing the creation and reuse of user interface components, particularly for single page web applications. Therefore, since our project was planned with a general single page application in mind, React met our needs very well.

Similarly, Typescript is a superset of the vanilla plain Javascript, which offers a series of advantages and benefits relative to the latter, such as static type analysis, with features like modules and classes. It has compile time type checking, and is also great for larger projects with multiple developers, as was the case with our team.

## **REDUX**

Redux is a state management library for front-end JavaScript. Redux provides the ability to create a globally accessible datastore whose properties are modified by dispatching action objects. Action objects are passed to a series of functions called reducers to generate an updated state. Redux integrates very well with React, and allows all of our React components to access the same data



## **GIT & GITHUB**

Over the past several months, our group has only used Github for our primary means of collaboration and version control. We each had a separate branch that allowed us to implement and merge various versions of our features to the broader project as we iterated updates each week.

## **TRELLO**

Trello is a visual project management tool that uses boards as a means to show the flow of work and when and how tasks should be completed. This was instrumental in our group having an organizational handle on our workload and deadlines.

## **SLACK**

Our team's primary means of communication, through both text threads and conference calls. Was key in managing our discussions and any ideas or issues we had quickly resolved.

# **V. TEAMMATE CONTRIBUTIONS**

## **SEAN HINDS**

Our most web-development experienced team member, Sean was crucial in guiding our team to overcoming many roadblocks, as well as suggesting better tools and methodologies for our project, such as his recommendation of using React and Typescript, as well as Trello. With that in mind, Sean also largely oversaw our Trello project management, such that we always clearly had which task to do when, and this was crucial in the efficient time management and execution of our project development. Sean also lead the implementation for our application's feature on geotag location and the manual input of other locations for the user to search study spots; he was also instrumental in creating the functionality of sorting by type of study spots. He also was indispensable in the creation of the user account login and registration abilities of our project.

## **HSIANG LO**

Hsiang was key in our ability to have an up and running website in several ways, not least of all which included the setting up, configuration, and resolving of Amazon Web Server's Elastic Beanstalk. His efforts in untangling several infrastructure issues regarding that service were essential to our team's ability to have a properly running website. Similarly, Hsiang navigated the establishing of Google Map API functionalities, which again was indispensable for our web application's most important capabilities as it relates to accessing Google Map's data regarding any potential study location. He also lead the implementation for the feature to allow the user to click on any aforementioned study spot and have an accessible information box with the name of the study spot appear. Hsiang also acquired a domain for the project through GoDaddy. He then have the nameserver transferred to AWS through AWS's Route 53. The website is then certified and secured using AWS's Certification Manager which in turn allowed the website to run using HTTPS. This action was crucial in ensuring privacy and security for the clients and is essential in allowing Google's function to locate the user's current location. Furthermore, he spent extensive time and effort leading the process for implementing a user database portion of the application, which were only curtailed due to time limitations with the end of the course. This in part, was related to the thorough testing and debugging processes that he completed on our project.

## **JOHN YOON**

John was vital in establishing the structural skeleton map of the web application, which leveraged the Google Maps API, and upon which the rest of the features would be implemented on. In addition to setting up the initial Git repository, he also lead the implementation for the features that allowed the user to access within the information box, all relevant data regarding the study spot's viability for selection, such as the overall rating of the establishment, the price level, the address, and whether or not the study spot was open at the time of the user's utilization of the application. Furthermore, he spent extensive time and effort leading the process for implementing a directions functionality of the application that would show a path from the user to a study spot, which were only curtailed due to time limitations with the end of the course. This in part, was related to the thorough testing of the application, as well as the completion of the final report. Relatedly, John also was key in outlining the various reports throughout the term.

## VI. DEVIATION FROM ORIGINAL PLAN

Our original plan included the consideration of having a more manual style filter system in which the user could select certain parameters with which to affect what types of study spots would be displayed on the map interface's results. However, this specific approach was modified somewhat by and large due to a conversation that revolved around several use case scenarios; for instance, in situations where the user might not have many options available, perhaps because they live in a rural region, then the quality may have to be sacrificed at the cost of quantity, or vice versa. In other words, either many mediocre study spots would be shown, or very few quality ones. Rather than the latter case, it was decided that the user could benefit more from having all the relevant available options displayed, and then the user could decide for themselves which study locations to disregard, by applying the various information data points available in the information box for each study spot. Relatedly, we had previously thought to include a visual media in the information box but as a design choice it was not included because the box was already cluttered with just text, and would be exponentially even more so if a picture was added.

In addition, although we implemented a landing page that handled registration and was also configured with the functionality of user account login and logout capabilities, ideally this would have progressed at least one more iteration. In this further stage, which we could not finalize due to time constraints and database complications, the user would have had the option to also save some of the favorite study spots that they had frequented, and maybe even study locations that they would have liked to go in the future. Similarly for the same reasons, the feature which would have allowed a user to have the directions mapped out from their location to the study spot of their choice was not finalized; in retrospect, even if it had been done so, in terms of use case probability, it would be much more likely that after selecting a study spot with our application, a user would then choose the GPS directional application of their usual preference such as Google Maps and instead proceed from there.

## VII. CONCLUSION

The overall process of creating our web application was a rigorous yet immensely rewarding one. Even though our group members had a wide range of website development

experience, we all learned and grew a substantial amount, particularly because we used and applied an extensive deal of technologies that none of us had ever used prior to this course. The new ways of thinking and developing applications that we fostered over the course of this class had profoundly shaped and grew our appreciation for all that goes into the successful creation and maintenance of a website.

Another noteworthy consideration developed during this process, was the deeper realization that our website reflected a larger theme and trend, in which a substantial amount of the application's effectiveness derived from standing upon the shoulders of giants. Whether it was leveraging the Amazon Web Services' Elastic Beanstalk capabilities, or Google Map's API functionalities, the overall integration and functionality of our website came from understanding what these services were and how best to utilize them into our project to create optimal results.

## REFERENCES

- Bryce St. Pierre “Google Maps API v.3.36 Tutorial” , *Youtube*, February 24th, 2019 <https://www.youtube.com/watch?v=oVr6unKZbg4&feature=youtu.be>
- Jonny Buchanan, Ean Platter, Max Stoiber, “Create-React-App”, *Github*, May 29th, 2013 <https://github.com/facebook/create-react-app>
- Jonny Buchanan, Ean Platter, Max Stoiber, “Create-React-App”, *Github*, <https://facebook.github.io/create-react-app/docs/running-tests>
- Jonny Buchanan, Ean Platter, Max Stoiber, “Create-React-App”, *Github*, <https://facebook.github.io/create-react-app/docs/deployments>
- Jonny Buchanan, Ean Platter, Max Stoiber, “Create-React-App”, *Github*, <https://facebook.github.io/create-react-app/docs/getting-started>
- Jordan Walke, React- A Javascript library for building user interfaces, May 29th, 2013 <https://reactjs.org/>
- Google “Google Maps API Bundles”, *Google Maps API* , June 2015, <https://developers.google.com/maps/documentation>