

Sprint 3 - Manual Testing Document

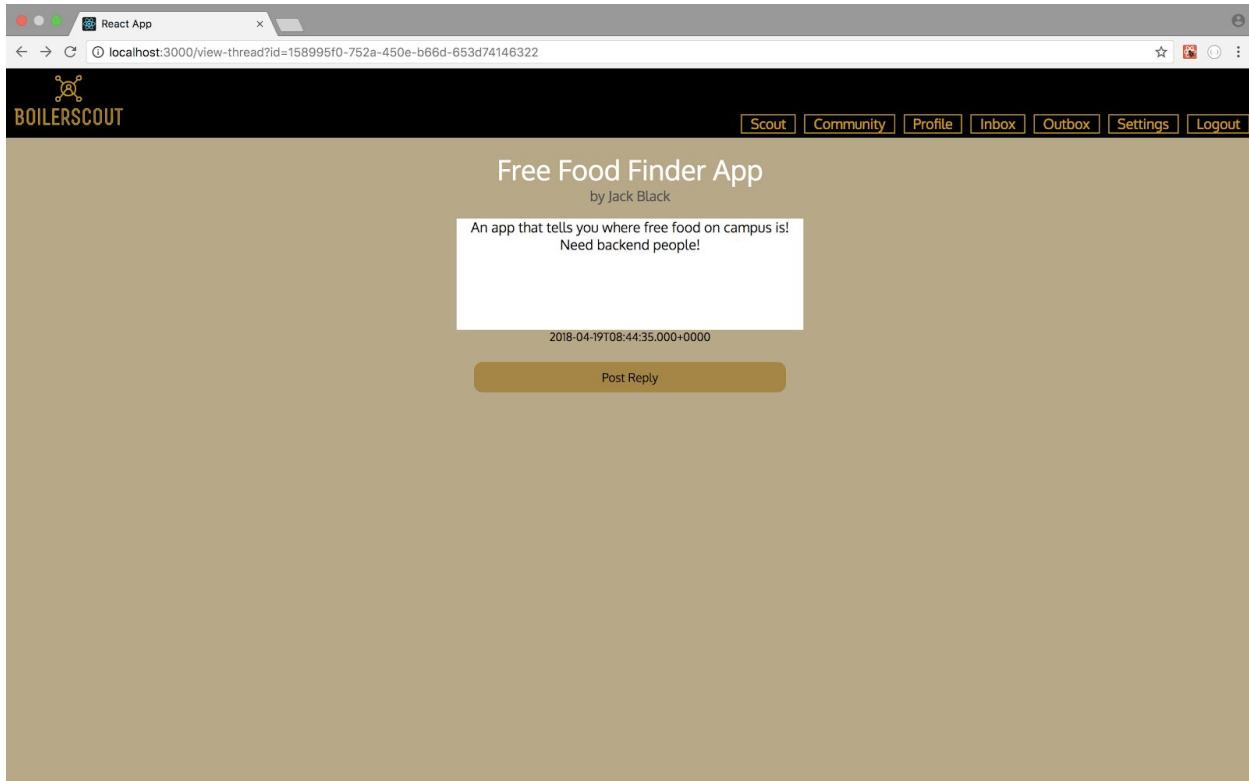
This document will outline the procedure and results of manually testing each story as specified in our sprint planning document.

Testing Set One: Viewing a Thread

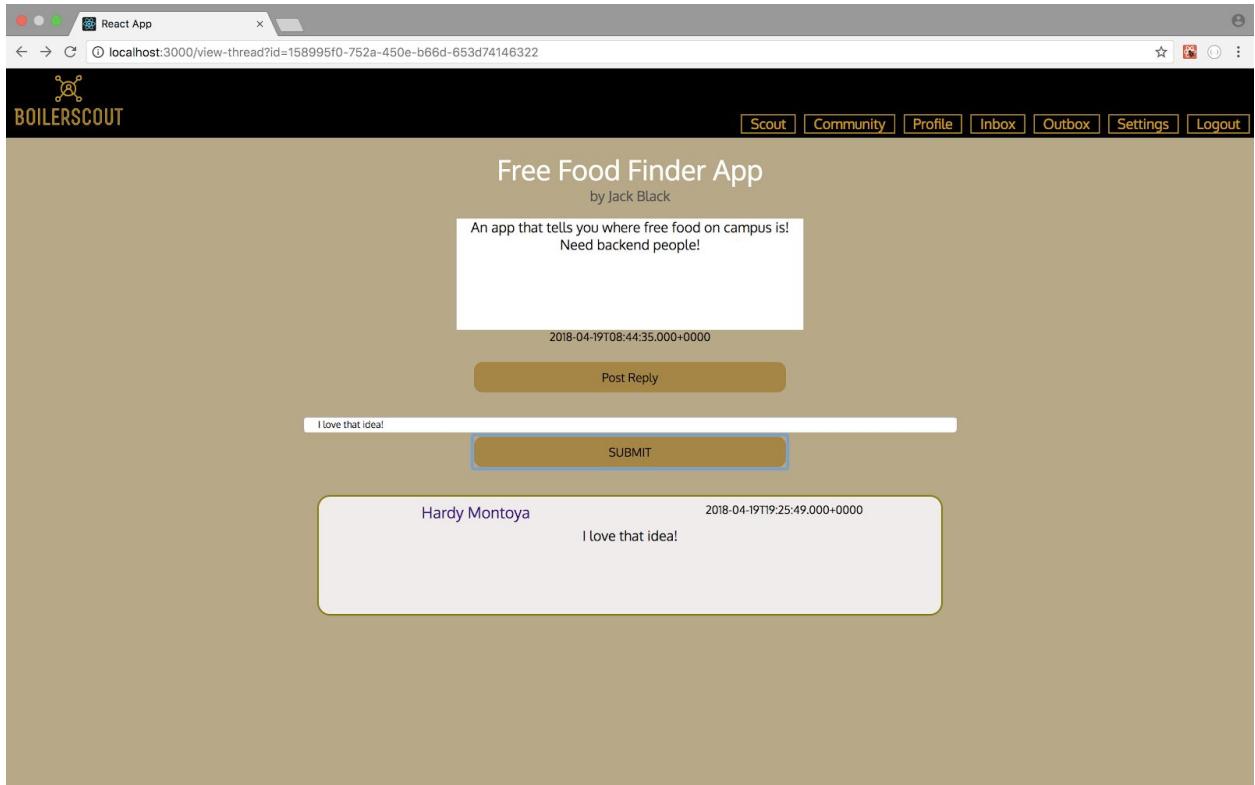
Testing Procedure: This is the protocol used to test viewing a thread.

- When viewing a forum, any thread that is clicked on should be displayed along with all the comments on the thread.
- Users should then be able to create comments of their own.

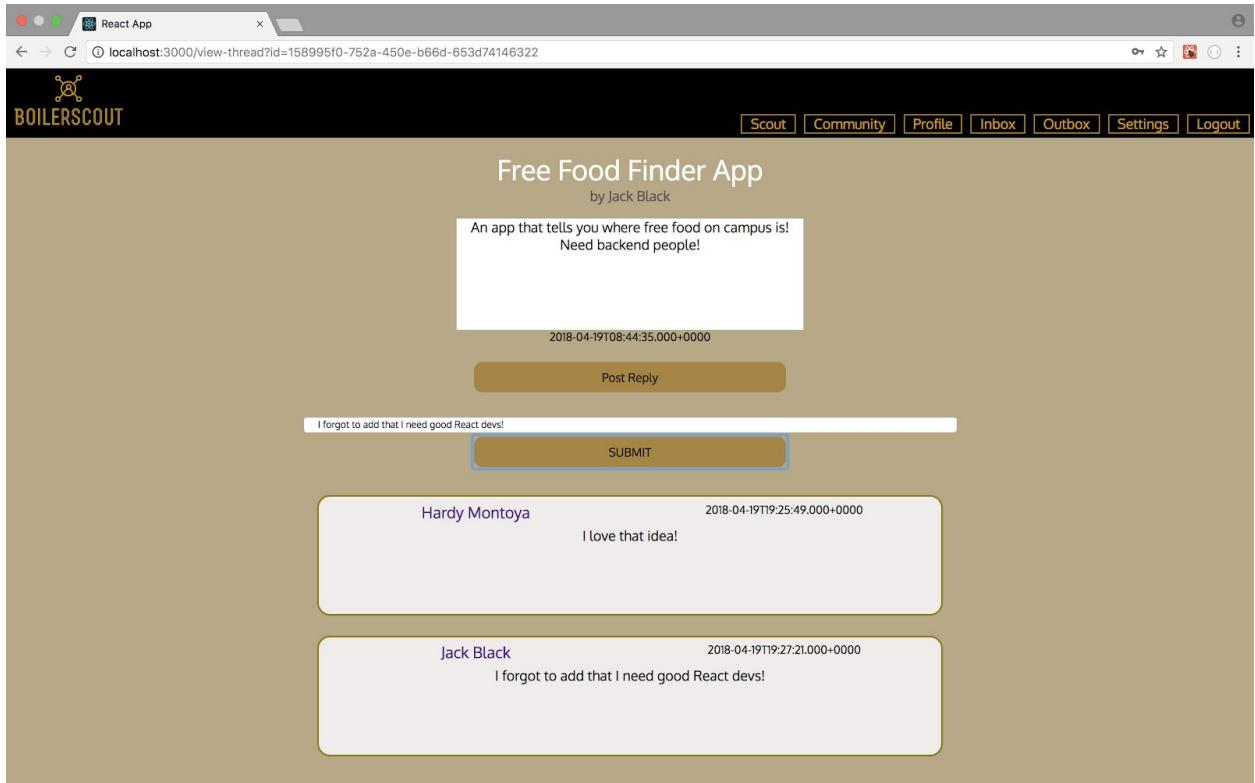
1. A thread is clicked on from Forum. No comments have been posted yet on this thread.



2. A user posts a comment and it is displayed



3. Another user logs in and views the same thread, and makes a comment of his own.

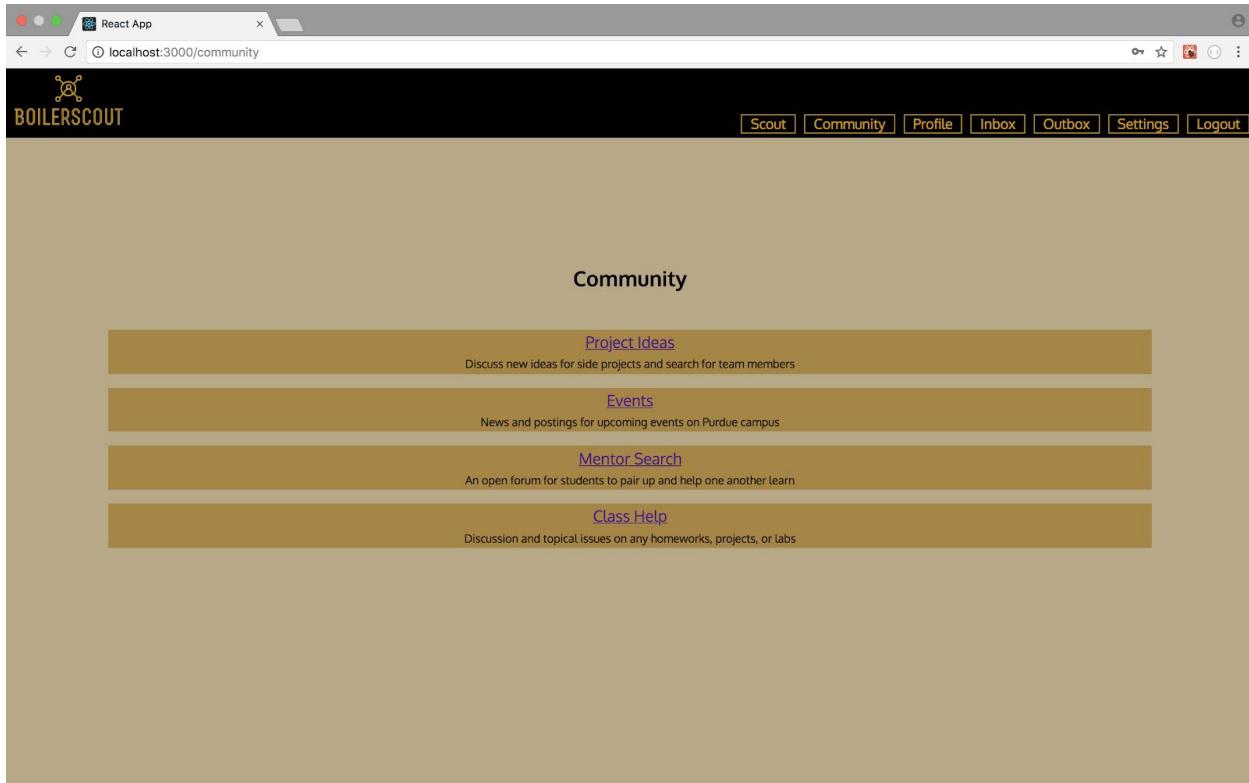


Testing Set Two: Viewing the threads in a forum

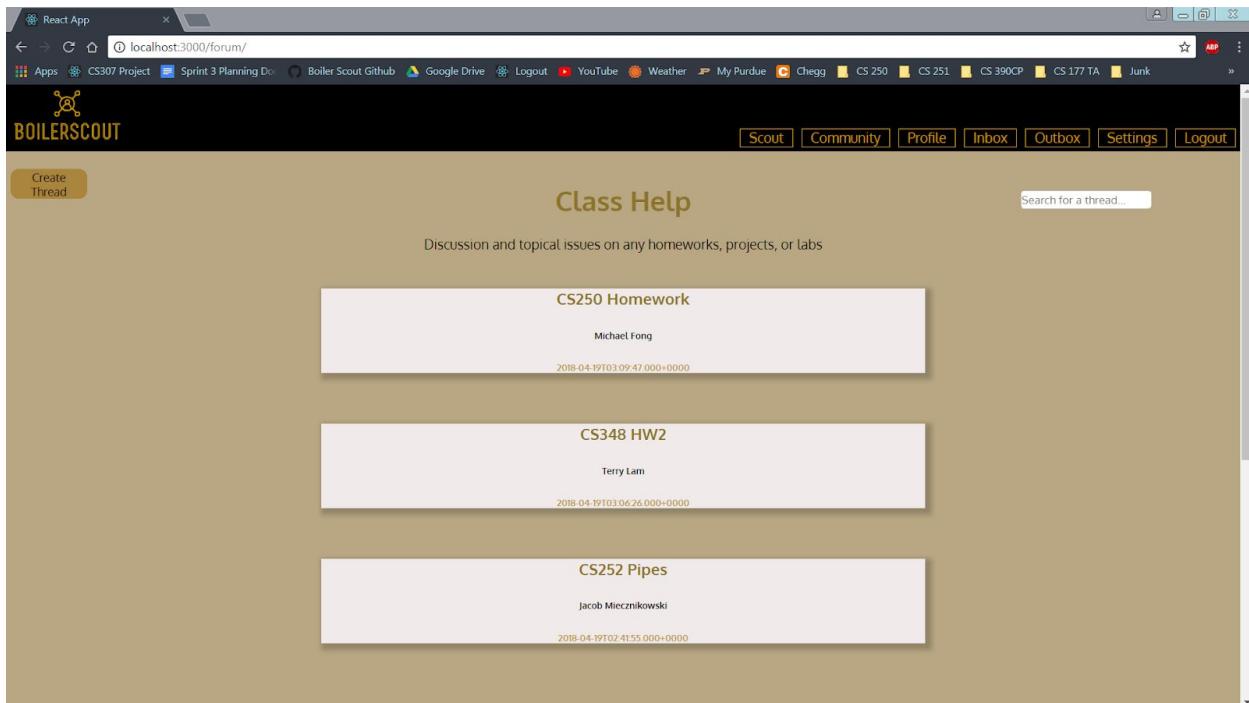
Testing Procedure: This is the protocol used to test viewing a forum.

- Users should be able to view all the available forums
- When a forum is clicked, all the threads of that forum should be displayed for viewing.

1. A user views the community page which displays the different forums



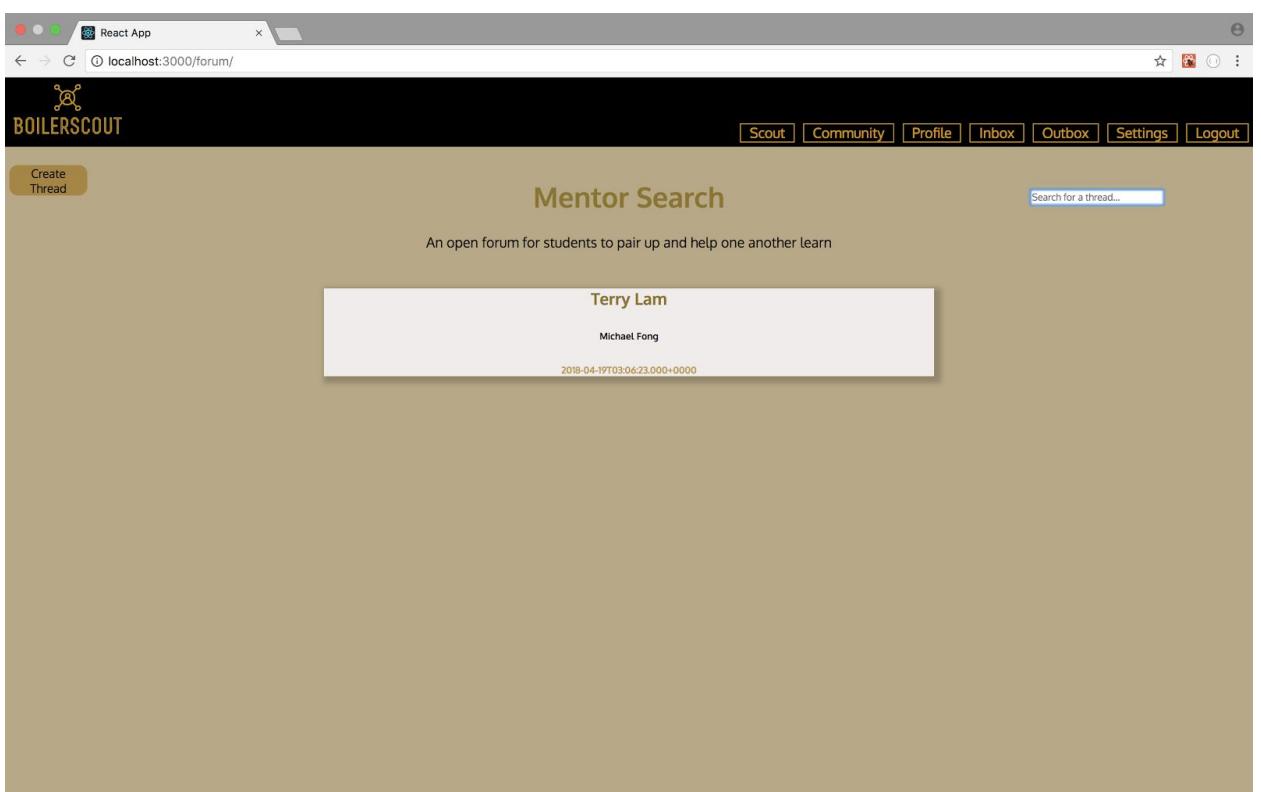
2. “Class Help” is clicked, and the threads posted in that forum are displayed.

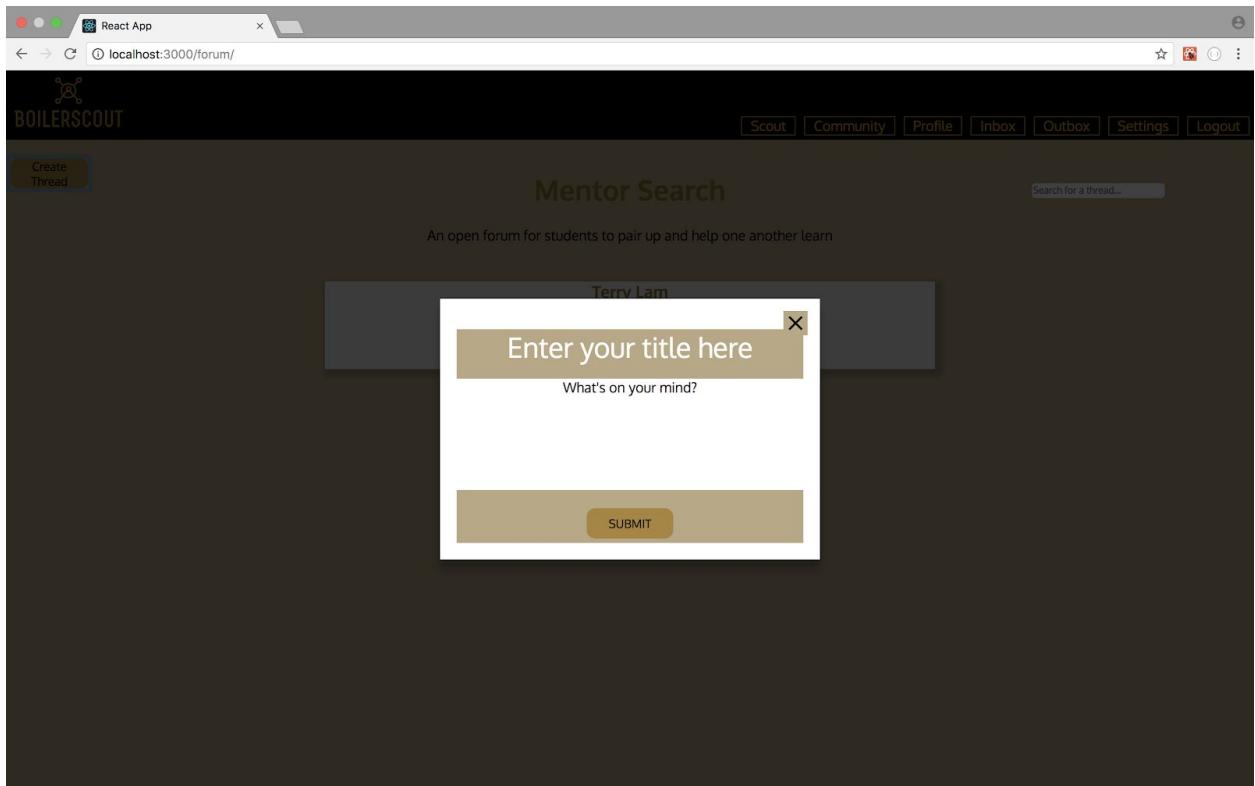


Testing Set Three: Creating a thread

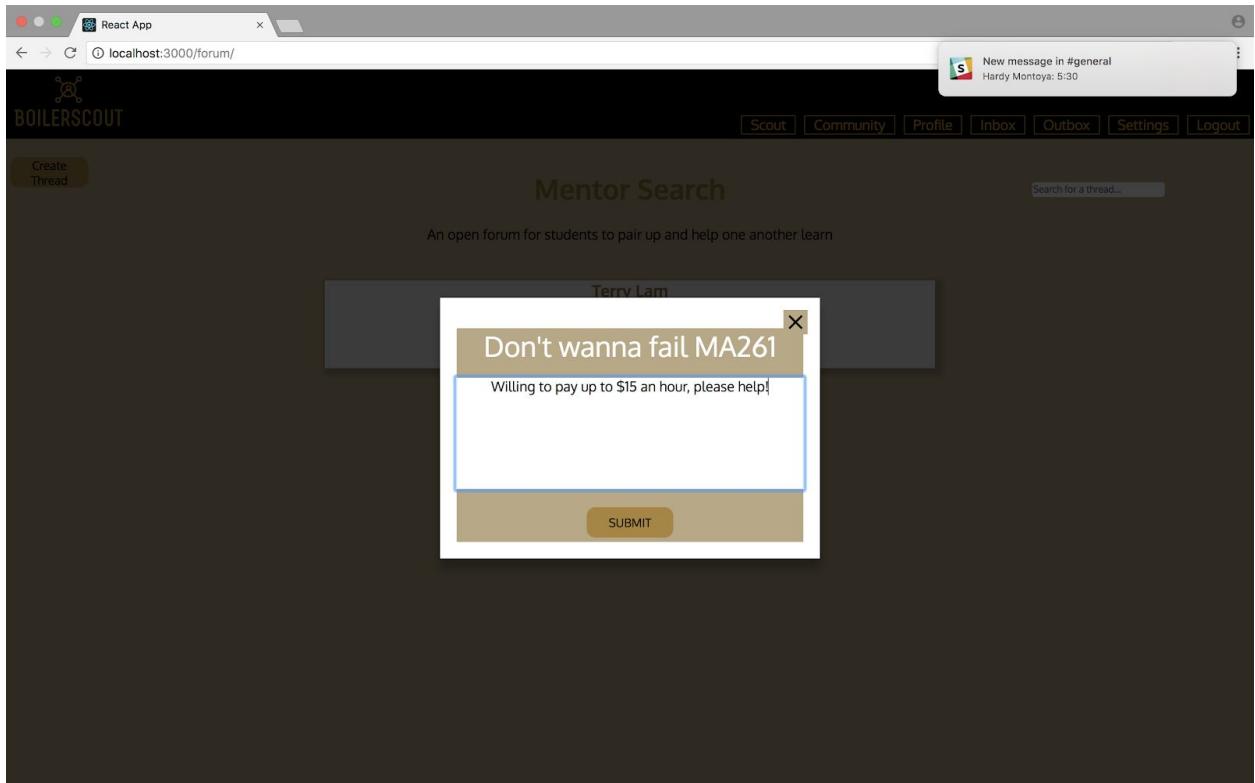
Testing Procedure: This is the protocol used to test searching for threads in a forum.

- Users should be able to go to a Forum and toggle a Create A New Thread view.
 - Users should enter a thread title and thread body.
 - After they submit, user should be displayed the Forum they posted the thread on, with their new thread displayed.
1. User goes to a forum and clicks on Create A Thread button, opening up a small view.





2. User types in the thread and body they would like.



3. After submitting, user is displayed the same forum with their new thread.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/forum/?". The page has a dark brown header with the "BOILERSCOUT" logo and navigation links for "Scout", "Community", "Profile", "Inbox", "Outbox", "Settings", and "Logout". Below the header is a "Create Thread" button. The main content area is titled "Mentor Search" and describes it as "An open forum for students to pair up and help one another learn". There are two visible threads:

- Don't wanna fail MA261** by **Jack Black** posted on **2018-04-19T19:46:17.000+0000**.

Content: Terry Lam
- Terry Lam** by **Michael Fong** posted on **2018-04-19T03:06:23.000+0000**.

A search bar at the top right says "Search for a thread...".

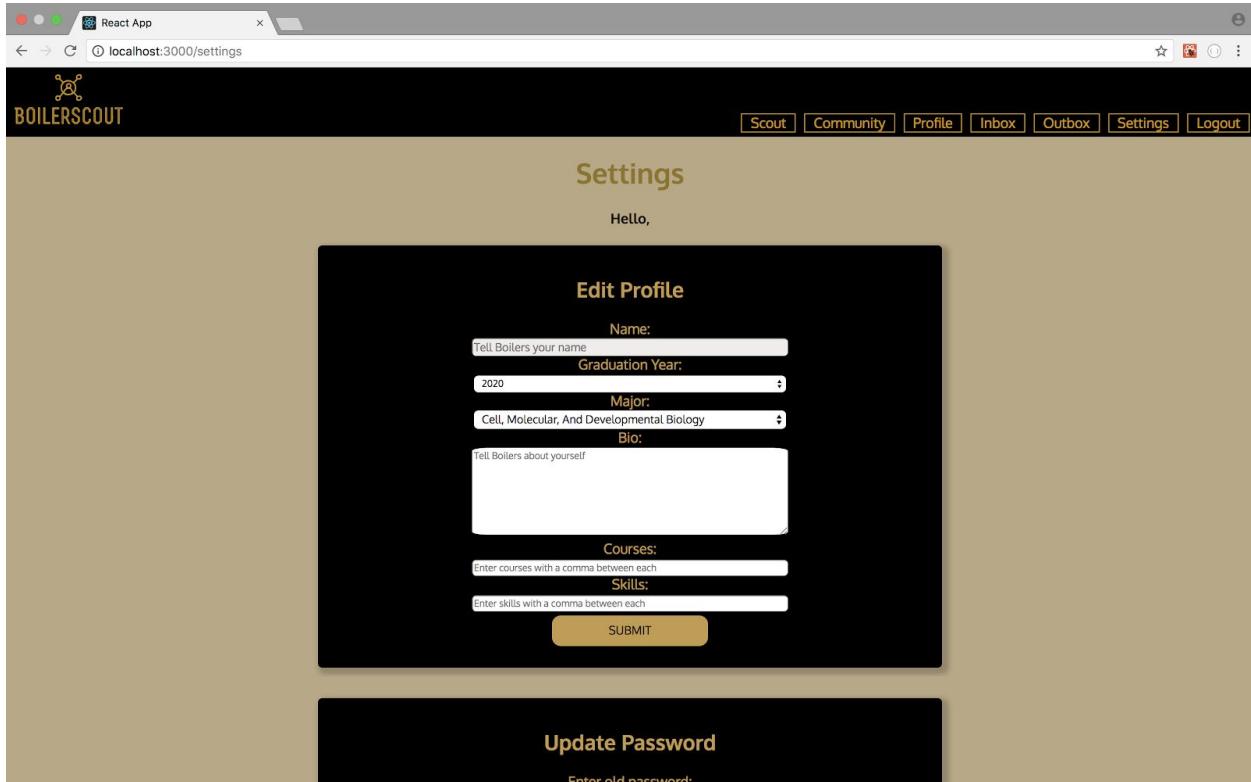
Testing Set Four: Update Profile

Testing Procedure: This is the protocol used to test updating profile.

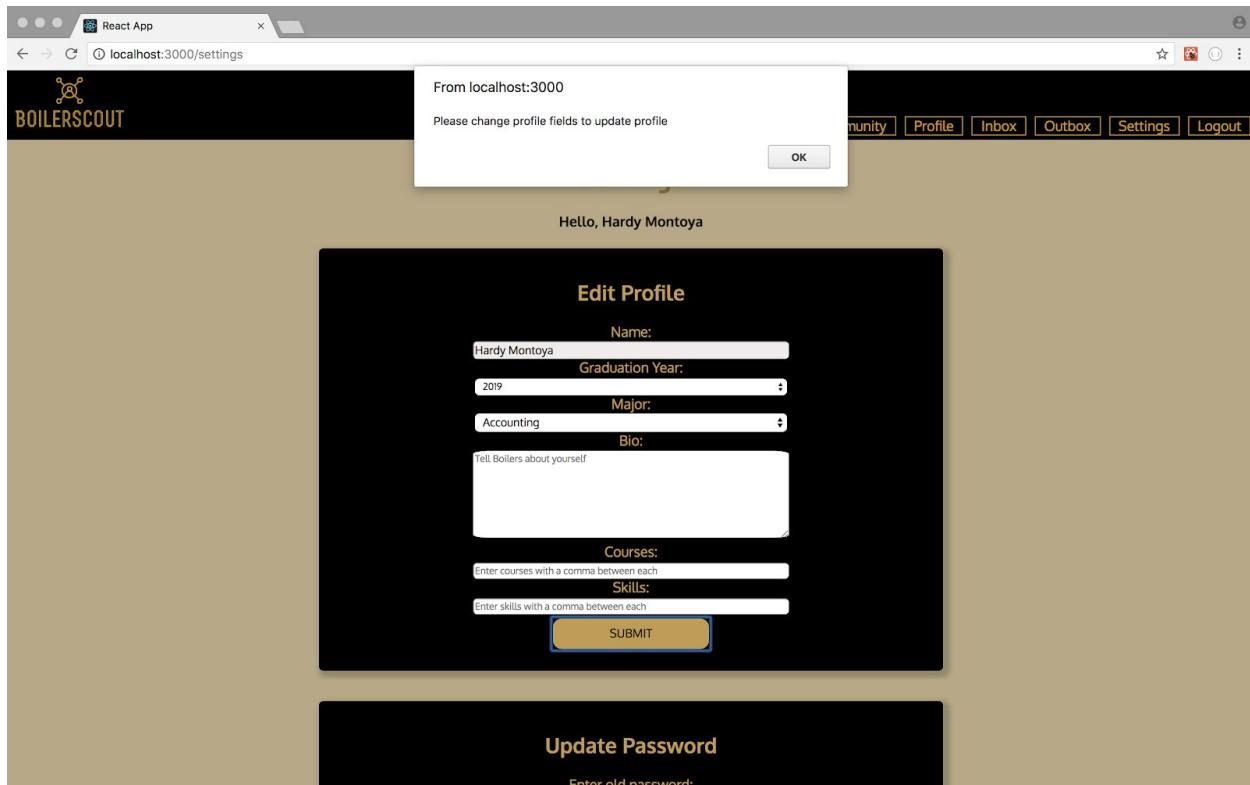
- User goes to Settings to update their profile.
- If user has not made a change and submits, page should alert user and prevent the information from being submitted.
- In the edit profile box, users should be displayed their name, graduation year and major, and bio, courses and skills if they have entered any.
- In the edit profile box, users should be able to change their name and bio freely.
- In the edit profile box, users should be able to change their graduation year and major from a drop down.
- In the edit profile box, users should be able to enter courses they want listed, and have it saved in their profile in uppercase.
- In the edit profile box, users should be able to enter skills they want listed, and have it saved in their profile in with the first letter only uppercase.

- If users adds white spaces when stating a skill or course (such as “j av a, p hp”) white spaces should be removed in frontend logic before being submitted.

1. User can access Edit Profile on their Settings page.

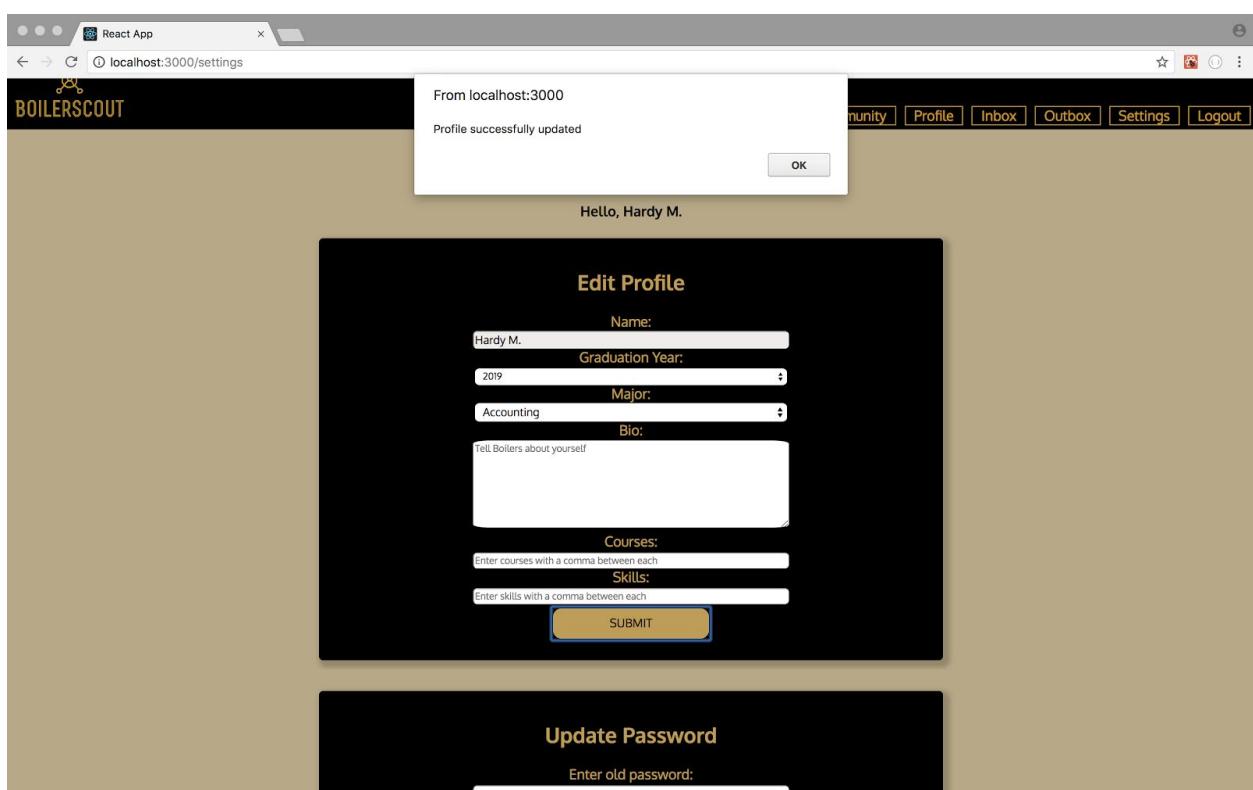


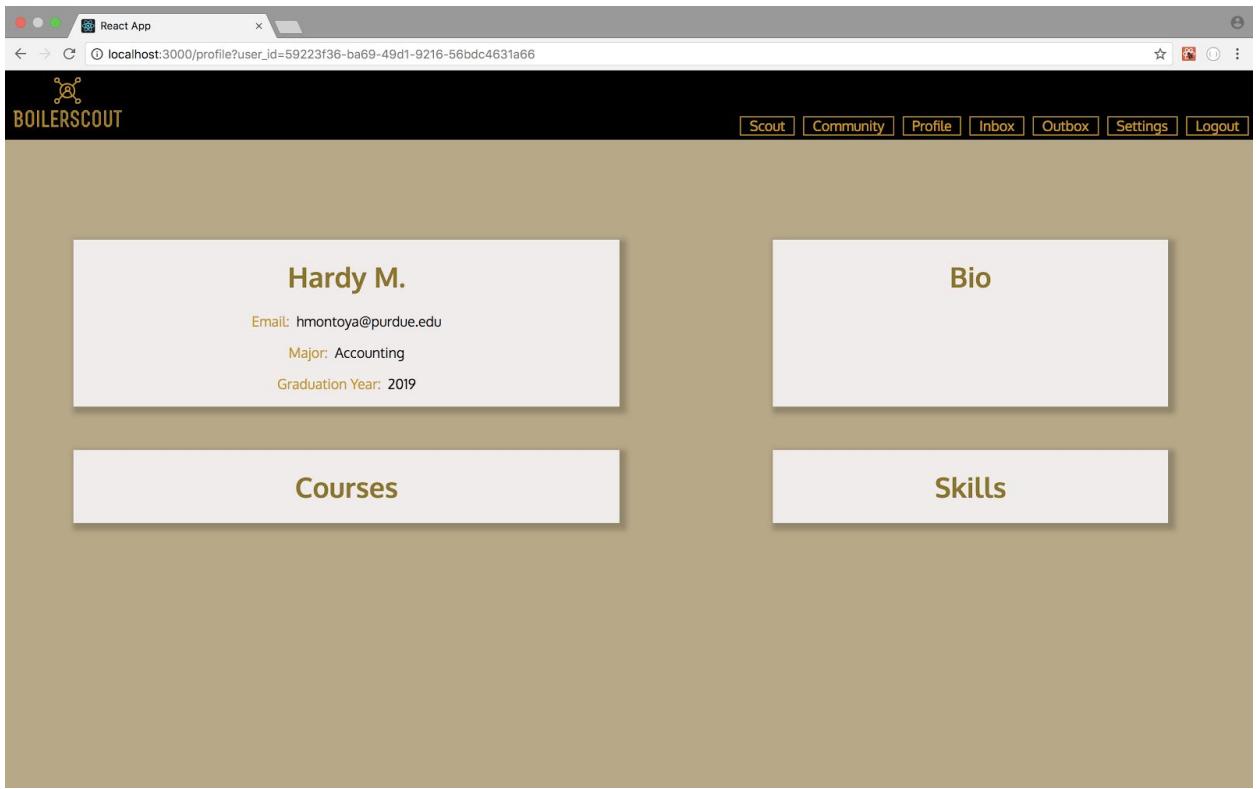
2. If user clicks submit without making a change, user will be alerted and nothing will be submitted.



3. User can edit each field independently and see results on their Profile page.

a. Name:





b. Graduation:

The screenshot displays two browser windows for the Boilerscout application.

Top Window: The URL is `localhost:3000/settings`. The page title is "Edit Profile". It contains fields for Name (Hardy M.), Graduation Year (2021), Major (Accounting), and Bio (Tell Boilers about yourself). Below these are fields for Courses (Enter courses with a comma between each) and Skills (Enter skills with a comma between each). A yellow "SUBMIT" button is at the bottom. A success message "Profile successfully updated" is displayed in a modal window.

Bottom Window: The URL is `localhost:3000/profile?user_id=59223f36-ba69-49d1-9216-56bdc4631a66`. The page title is "Update Password". It has fields for Enter old password and Enter new password.

Profile Summary: The main content area shows a summary of the user's profile: Hardy M., Email: hmontoya@purdue.edu, Major: Accounting, Graduation Year: 2021. To the right, there are sections for Bio, Courses, and Skills, each represented by a large, empty white box.

c. Major:

The screenshots illustrate the Boilerscout application's user interface, specifically the profile management features.

Edit Profile

From localhost:3000
Profile successfully updated

Hello, Hardy M.

Name:
Hardy M.

Graduation Year:
2021

Major:
Studio Arts And Technology

Bio:
Tell Boilers about yourself

Courses:
Enter courses with a comma between each

Skills:
Enter skills with a comma between each

SUBMIT

Update Password

Enter old password:

Hardy M.

Email: hmontoya@purdue.edu

Major: Studio Arts And Technology

Graduation Year: 2021

Bio

Courses

Skills

d. Bio:

The application interface consists of two main components:

- Edit Profile Form:** This form is used to update user information. It includes fields for:
 - Name: Hardy M.
 - Graduation Year: 2021
 - Major: Studio Arts And Technology
 - Bio: Hi! I love this school!
 - Courses: (Input field placeholder: Enter courses with a comma between each)
 - Skills: (Input field placeholder: Enter skills with a comma between each)A yellow "SUBMIT" button is located at the bottom right of the form.
- User Profile Page:** This page displays the user's profile information. It includes:
 - Hardy M.**: Profile name.
 - Email:** hmontoya@purdue.edu
 - Major:** Studio Arts And Technology
 - Graduation Year:** 2021
 - Bio:** Hi! I love this school!
 - Courses:** (Empty input field)
 - Skills:** (Empty input field)

e. Courses:

- Courses are converted to uppercase before being submitted.

From localhost:3000
Profile successfully updated

Hello, Hardy M.

Edit Profile

Name: Hardy M.
Graduation Year: 2021
Major: Studio Arts And Technology
Bio:
Hi! I love this school!

Courses: MA251,ENG891
Skills: Java,C,Scala,Php

Update Password

Enter old password:

Hardy M.

Email: hmontoya@purdue.edu
Major: Studio Arts And Technology
Graduation Year: 2021

Bio

Hi! I love this school!

Courses

MA251
ENG891

Skills

Java
C
Scala
Php

- ii. Whitespaces are removed as well exemplified in last course in this example.

The image consists of two vertically stacked screenshots of a web application interface, likely a React application, showing the process of updating a user's profile and displaying the results.

Screenshot 1: Edit Profile (localhost:3000/settings)

This screenshot shows the "Edit Profile" form. The input fields contain the following data:

- Name: Hardy M.
- Graduation Year: 2021
- Major: Studio Arts And Technology
- Bio: Hi! I love this school!
- Courses: MA251,ENG891,ma718
- Skills: Java,C,Scala,Php

An "OK" button is visible at the bottom right of the modal.

Screenshot 2: Update Password (localhost:3000/profile?user_id=59223f36-ba69-49d1-9216-56bcd4631a66)

This screenshot shows the "Update Password" form with fields for "Enter old password:" and "Enter new password:".

Profile Display (localhost:3000/profile?user_id=59223f36-ba69-49d1-9216-56bcd4631a66)

The user profile is displayed with the following information:

- Hardy M.** (Section title)
- Email: hmontoya@purdue.edu
- Major: Studio Arts And Technology
- Graduation Year: 2021
- Bio** (Section title): Hi! I love this school!
- Courses** (Section title): MA251, MA718, ENG891
- Skills** (Section title): Java, C, Scala, Php

iii. Duplicates are removed as well, as seen with ENG891.

The screenshot shows two browser windows for the BoilerScout application. The top window is titled 'Edit Profile' and displays fields for Name (Hardy M.), Graduation Year (2021), Major (Studio Arts And Technology), and Bio (Hi! I love this school!). It also shows a list of courses: MA251, MA718, ENG891, and ENG891. The bottom window is titled 'Update Password' and has a field for 'Enter old password'. The main application interface shows a profile card for 'Hardy M.' with email (hmontoya@purdue.edu), major (Studio Arts And Technology), and graduation year (2021). To the right is a 'Bio' section with the text 'Hi! I love this school!'. Below the profile are 'Courses' (MA251, MA718, COM217, ENG891) and 'Skills' (Java, C, Scala, PHP).

From localhost:3000
Profile successfully updated

OK

Hello, Hardy M.

Edit Profile

Name: Hardy M.

Graduation Year: 2021

Major: Studio Arts And Technology

Bio: Hi! I love this school!

Courses: MA251, MA718, ENG891, ENG891

Skills: Java, C, Scala, PHP

SUBMIT

Update Password

Enter old password:

Hardy M.

Email: hmontoya@purdue.edu

Major: Studio Arts And Technology

Graduation Year: 2021

Bio

Hi! I love this school!

Courses

MA251
MA718
COM217
ENG891

Skills

Java
C
Scala
PHP

f. Skills

- i. Skills are converted for only first letter to be uppercase before being submitted.

The screenshot displays two browser windows for the BOILERSCOUT application.

Top Window: Edit Profile

- URL: `localhost:3000/settings`
- Content: A modal window titled "Edit Profile" with fields for Name (Hardy M.), Graduation Year (2021), Major (Studio Arts And Technology), and Bio (Hi! I love this school!). It also includes a Courses input field (Enter courses with a comma between each) and a Skills input field (ScAla, JAVA, C). A "SUBMIT" button is at the bottom.
- Message: "Profile successfully updated" is displayed above the modal.

Bottom Window: Update Password

- URL: `localhost:3000/profile?user_id=59223f36-ba69-49d1-9216-56bcd4631a66`
- Content: A modal window titled "Update Password" with fields for Enter old password and Enter new password.

Profile Summary (Bottom Left)

- Header: BOILERSCOUT
- Content:
 - Hardy M.**
 - Email: `hmontoya@purdue.edu`
 - Major: Studio Arts And Technology
 - Graduation Year: 2021

Bio (Bottom Right)

- Content: Hi! I love this school!

Courses (Bottom Left)

- Content: (Empty box)

Skills (Bottom Right)

- Content: Java, C, Scala (Skills listed vertically in a box)

ii. Whitespaces are removed as well, as seen in both Scala and Php.

The image contains two screenshots of a web application interface, likely a React application, showing profile editing and display features.

Screenshot 1: Edit Profile

This screenshot shows the "Edit Profile" form. The "Name:" field contains "Hardy M.". The "Graduation Year:" dropdown is set to "2021". The "Major:" dropdown is set to "Studio Arts And Technology". The "Bio:" text area contains "Hi! I love this school!". The "Courses:" input field has "Enter courses with a comma between each" placeholder. The "Skills:" input field contains "java,C,S cala,P h P". A "SUBMIT" button is at the bottom.

Screenshot 2: Profile Display

This screenshot shows the user's profile information. The header says "From localhost:3000" and "Profile successfully updated". Below it, the user's name is "Hello, Hardy M.". The profile card displays:

- Hardy M.**
- Email: hmontoya@purdue.edu
- Major: Studio Arts And Technology
- Graduation Year: 2021

On the right, there are two boxes:

- Bio**: Hi! I love this school!
- Courses**: (empty)
- Skills**: Java, C, Scala, Php

iii. Duplicates are removed as well, as seen with Java and C

From localhost:3000
Profile successfully updated

Hello, Hardy M.

Edit Profile

Name: Hardy M.
Graduation Year: 2021
Major: Studio Arts And Technology
Bio:
Hi! I love this school!

Courses: MA251,MA718,COM217,ENG891
Skills: Java,C,Scala,Php, c,Java

SUBMIT

Update Password

Enter old password:

Hardy M.

Email: hmontoya@purdue.edu
Major: Studio Arts And Technology
Graduation Year: 2021

Bio

Hi! I love this school!

Courses

MA251
MA718
COM217
ENG891

Skills

Java
C
Scala
Php

Testing Set Five: Search Threads

Testing Procedure: This is the protocol used to test searching for threads in a forum.

- Users should have the displayed threads filtered to display only the one corresponding to the input as the user types.
- If no thread matches the input, then no threads should be displayed.
- Users should still be able to click a thread a be taken to view the comments of that thread.

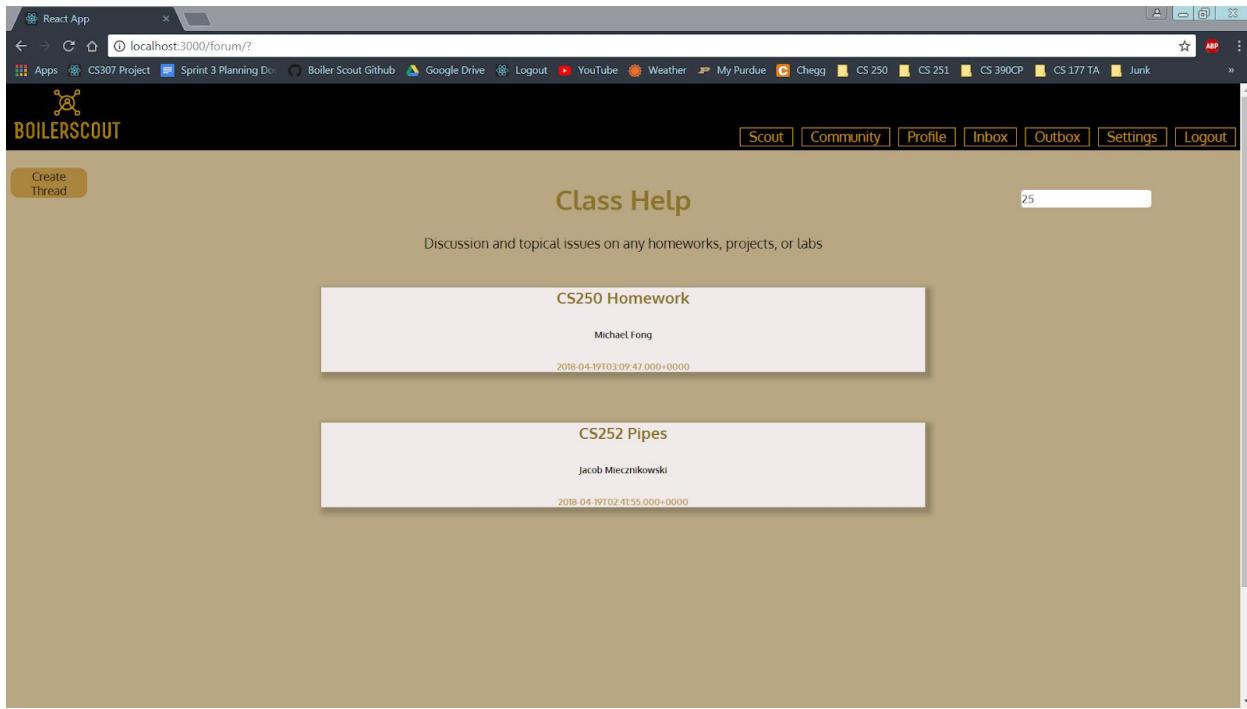
1. User views the threads of the “Class Help” forum.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/forum/?". The page has a dark-themed header with the "BOILERSCOUT" logo and navigation links for "Scout", "Community", "Profile", "Inbox", "Outbox", "Settings", and "Logout". A "Create Thread" button is visible on the left. The main content area is titled "Class Help" and contains a search bar with the placeholder "Search for a thread...". Below the search bar, there is a sub-header "Discussion and topical issues on any homeworks, projects, or labs". Three threads are listed in a grid format:

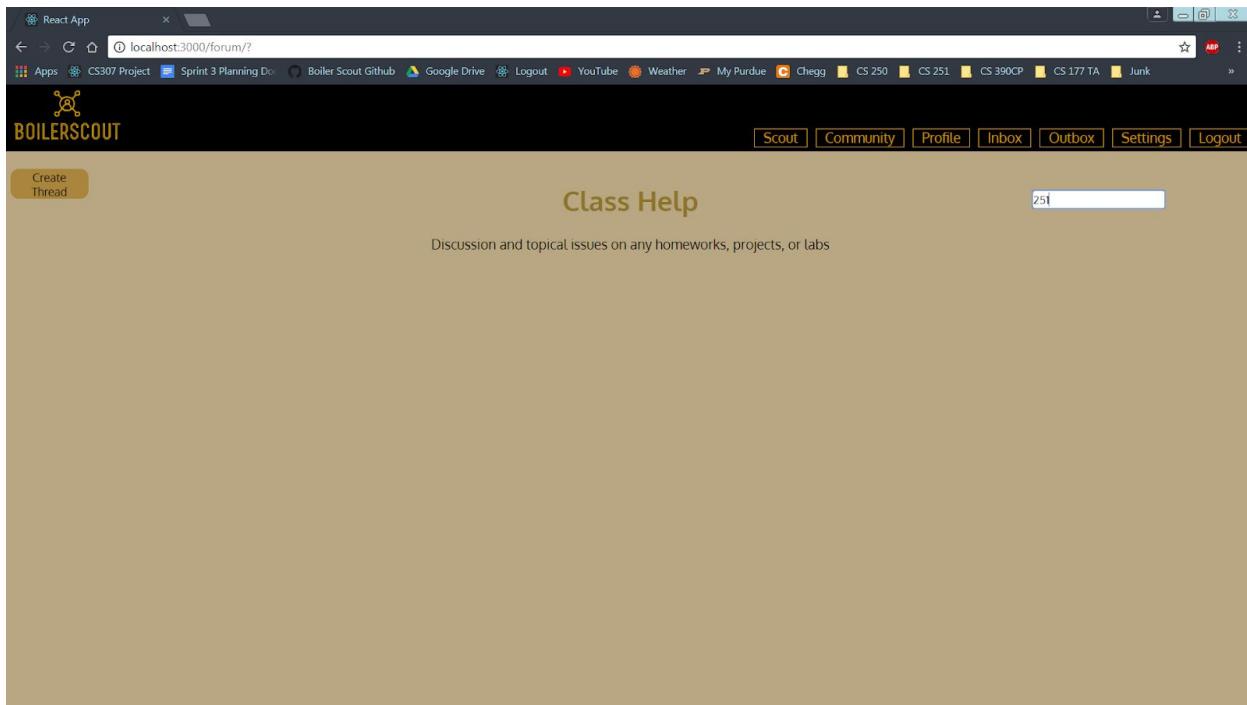
- MA265** by Michael Fong on 2018-04-19T03:18:54.000+0000
- CS250 Homework** by Michael Fong on 2018-04-19T03:09:47.000+0000
- CS348 HW2** by Terry Lam on 2018-04-19T03:06:26.000+0000

A horizontal scrollbar is visible at the bottom of the page.

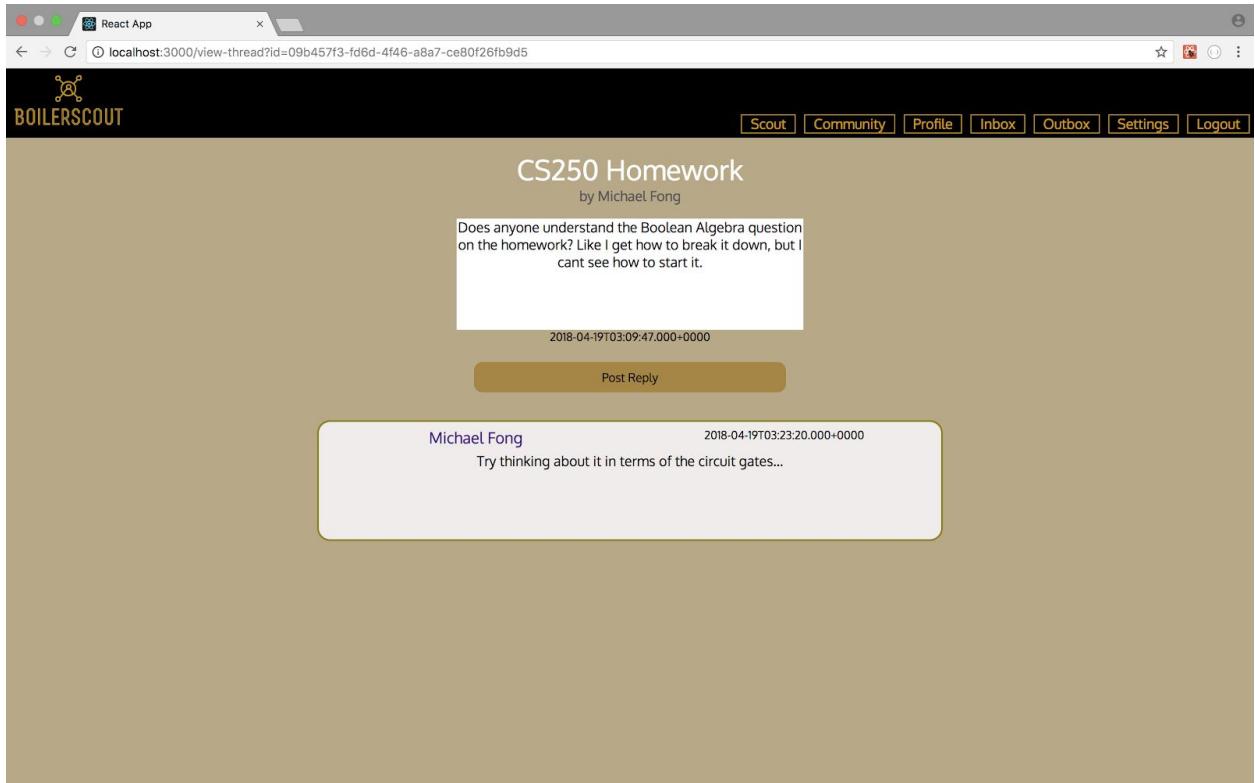
2. User types '25' in the input box. Now only the threads containing '25' are displayed.



2. The user continues to type ("251"), searching for a thread that doesn't exist, and no threads are then displayed.



3. User clicks on one of the search threads and is still able to view it

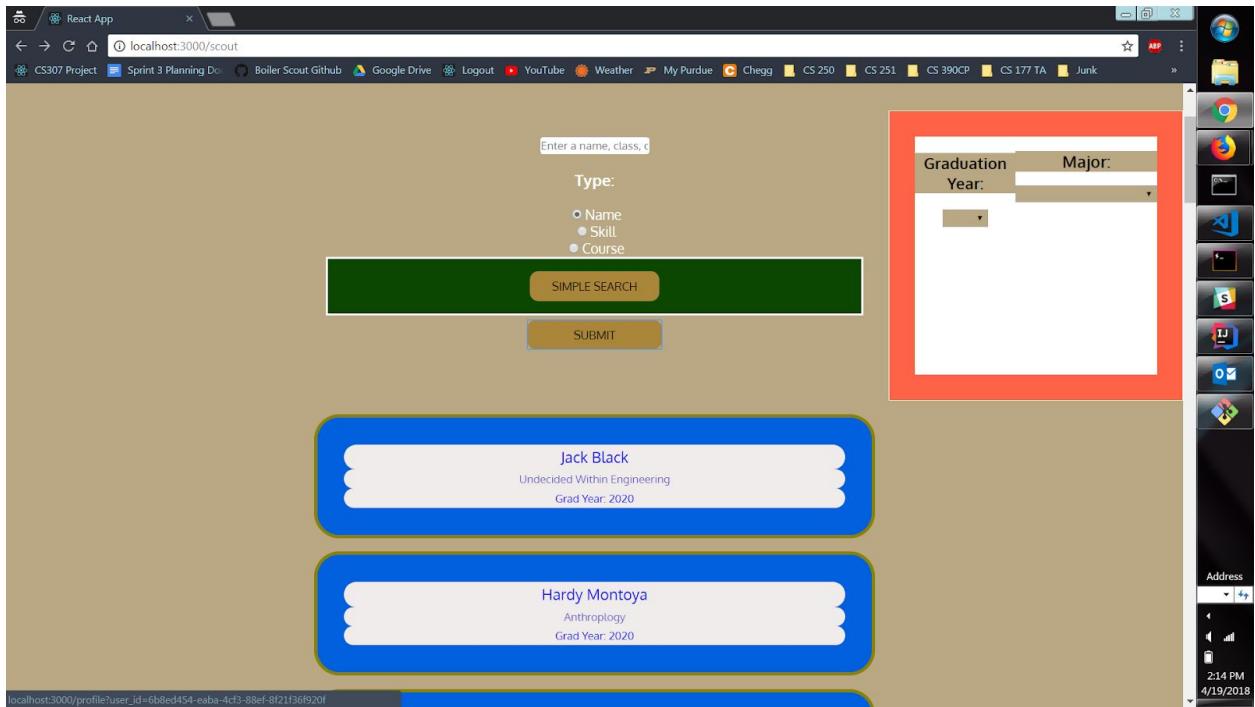


Testing Set Six:

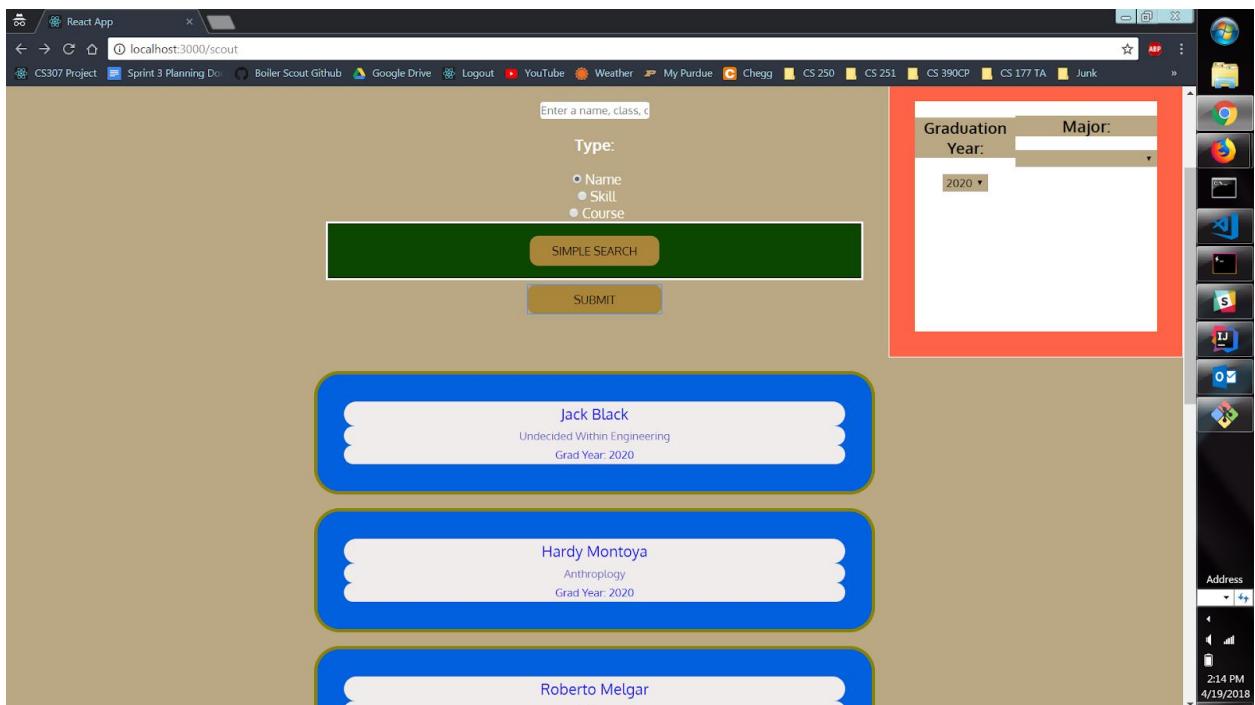
Testing Procedure: This is the protocol used to test the functionality of the Advanced Filters Search.

- Users should have only users displayed that match the criteria of advanced filters, which is graduation year and major.
- Users should have the ability to enable and disable advanced filters.
- No results should be displayed when no users match the advanced filters.

1. Advanced Filters is open, but no extra data is selected. All users are still returned



2. Graduation Year is selected and Submit it clicked. Users with grad year 2020 are returned from all major, since major isn't specified



Hardy Montoya
Anthropology
Grad Year: 2020

Roberto Melgar
Business
Grad Year: 2020

Lyes Infante
Biology
Grad Year: 2020

Nico Bellic
Exploratory Studies
Grad Year: 2020

3. Now Major is selected as Computer Science and Graduation Year is set to 2019. On submit, only users with that major and grad year are shown.

Enter a name, class, c

Type:

Name

Skill

Course

SIMPLE SEARCH

SUBMIT

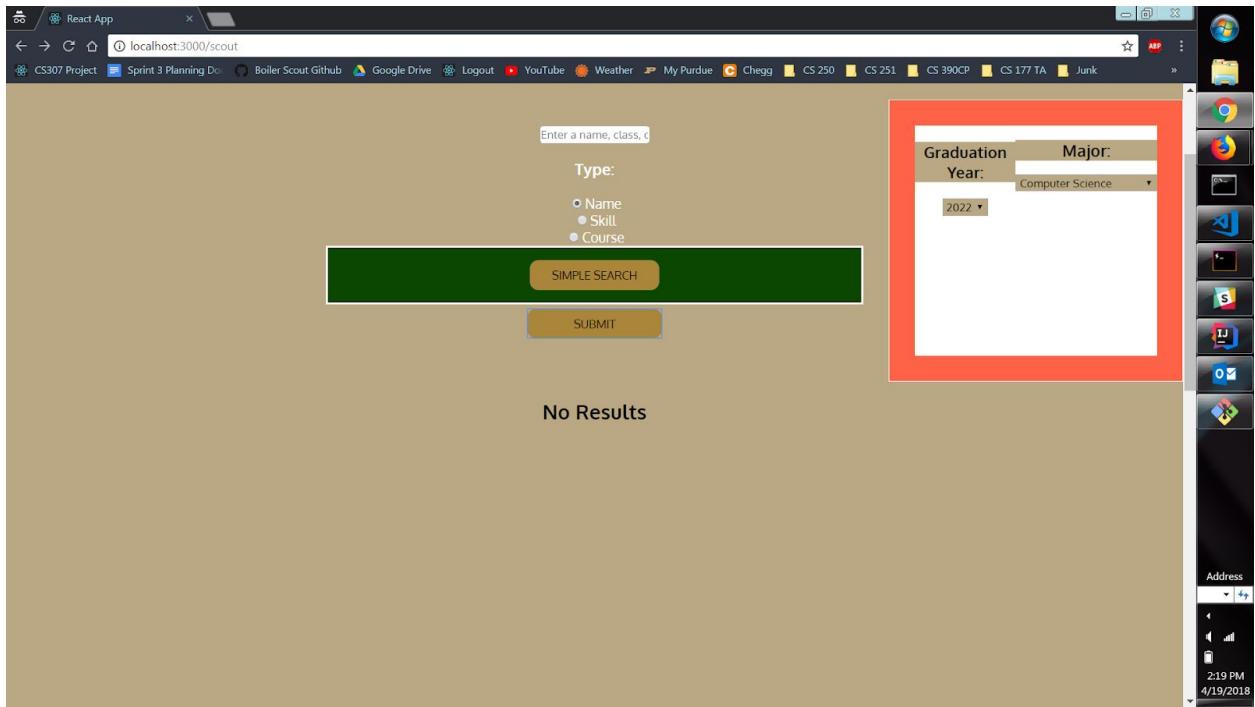
Graduation Year:

Major:

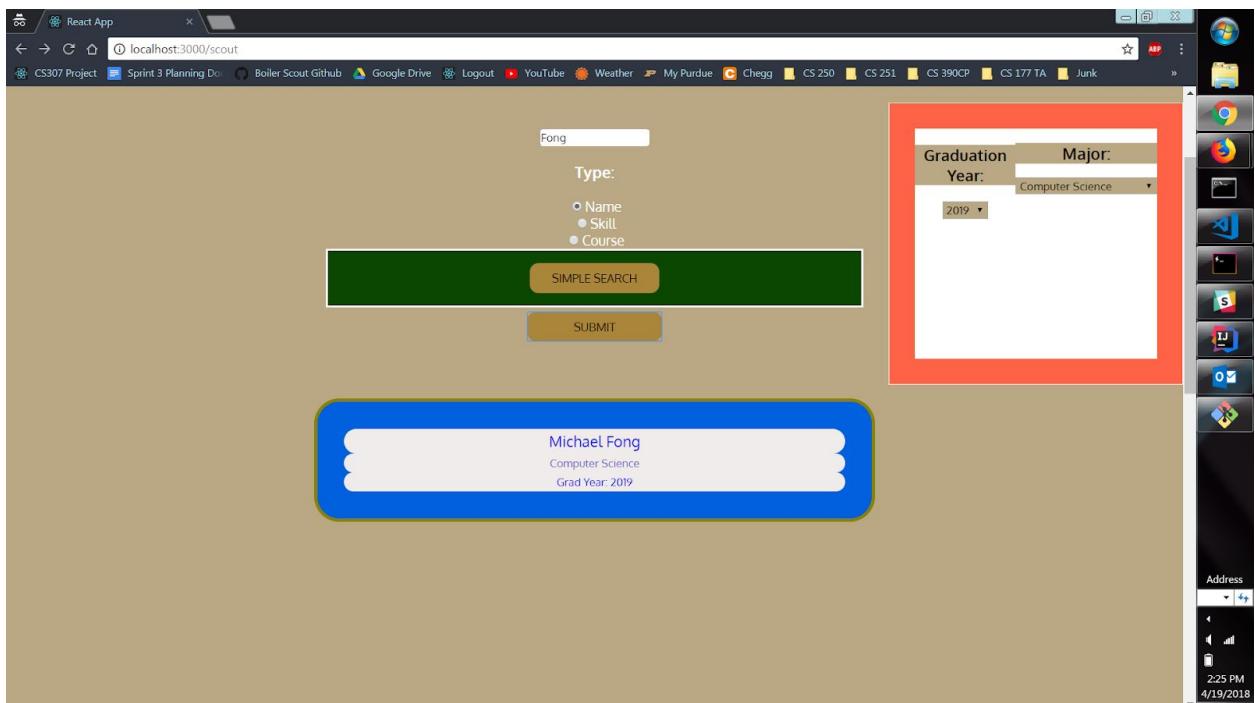
Terry Lam
Computer Science
Grad Year: 2019

Michael Fong
Computer Science
Grad Year: 2019

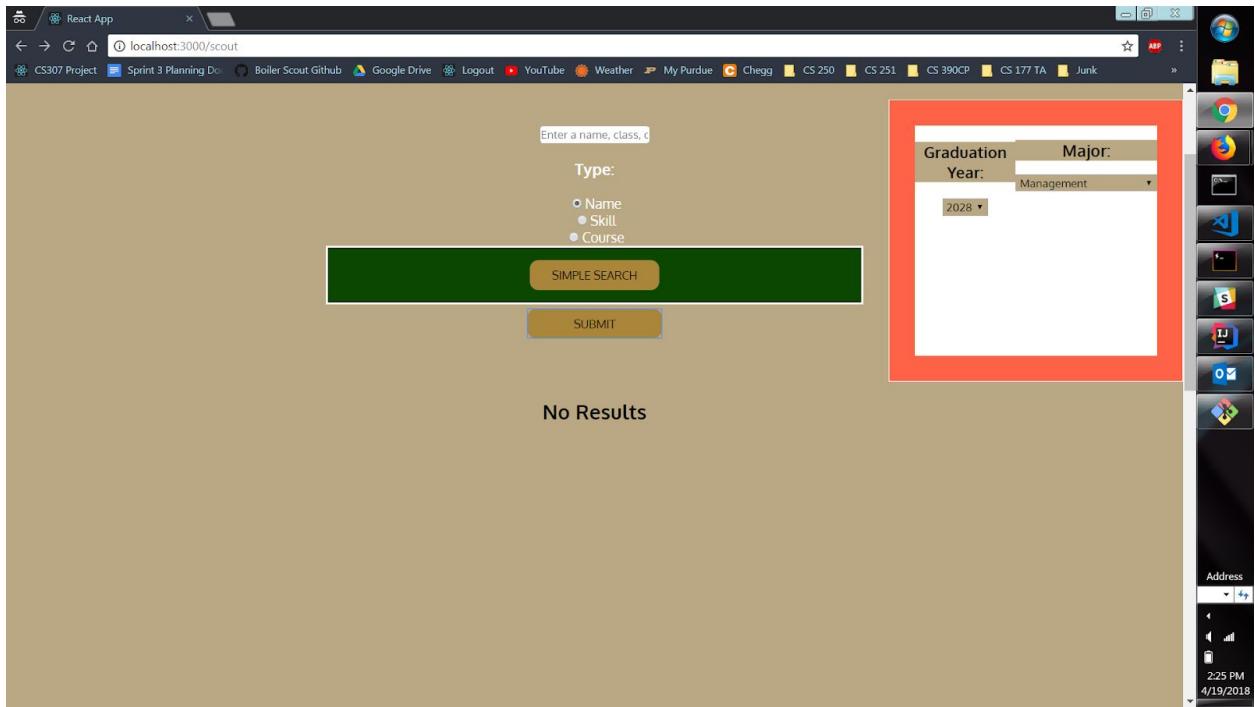
4. Grad year is changed to 2022 and major is left as Computer Science. Now, no users are returned because the only computer science students graduation in 2019



5. Now grad year is set back to 2019, and Fong is entered into the search. There is only one result, which matches the search and advanced filters.



6. When a user doesn't match the advanced filters, nothing is displayed and the user is informed that there are no results

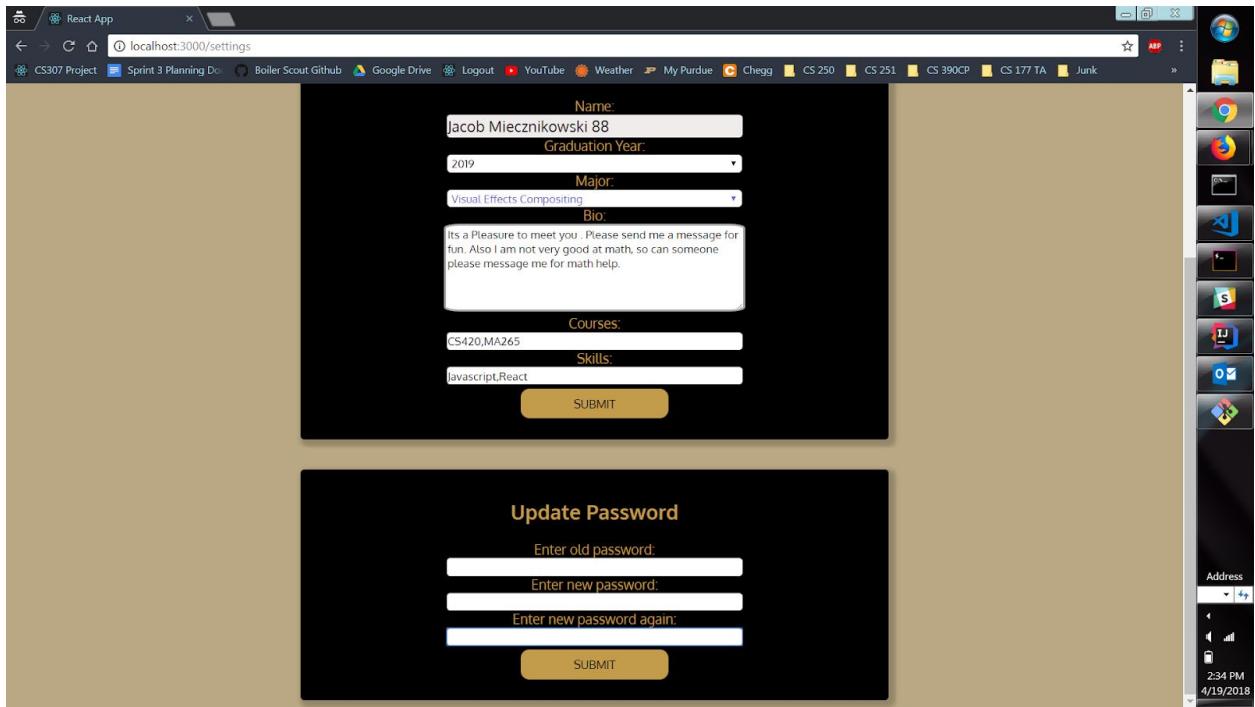


Testing Set Seven:

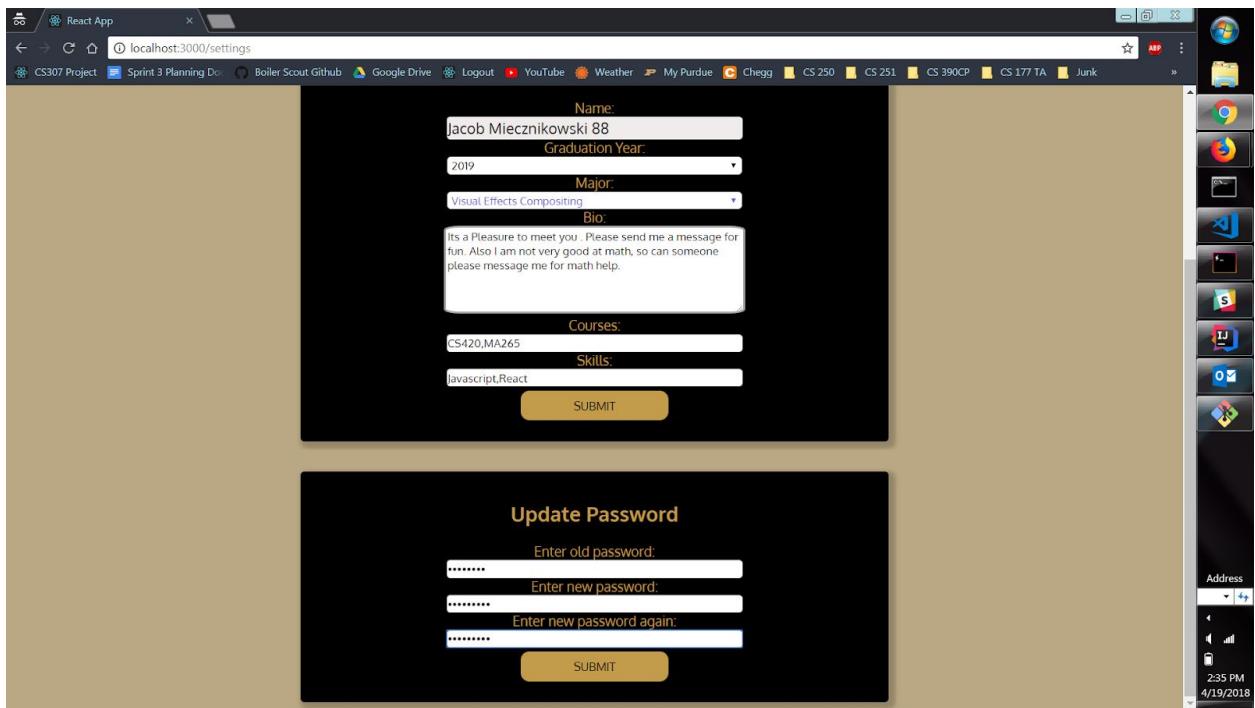
Testing Procedure: This is the protocol used to test the functionality of updating the password inside the settings page

- Users should be able to update their password inside settings once they are logged in.
- Input a correct old password and matching new password. The user should be informed when the password updating is successful.
- Users should be informed when their old password is incorrect when the old input password is incorrect.
- Input new passwords that do not match along with a correct old password. The user should be informed that the new passwords do not match.

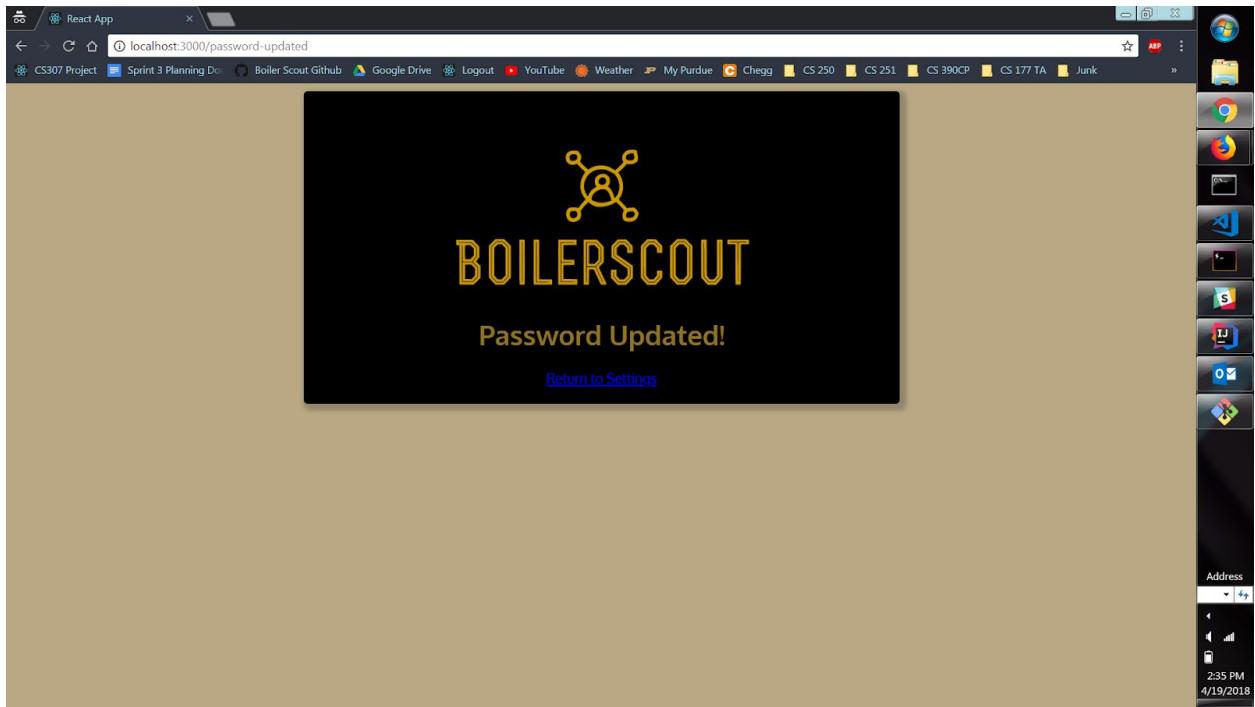
1. Navigate to the Update Password form in settings after logging in



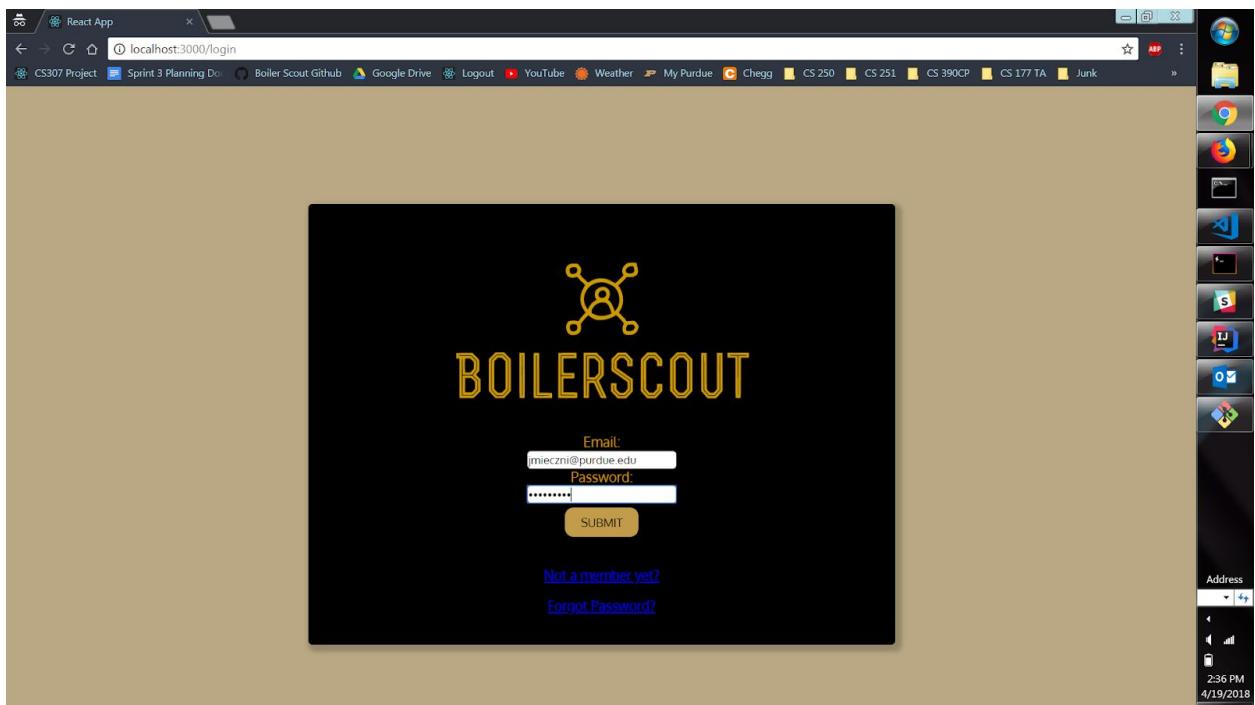
2. A correct old password is entered along with matching new passwords



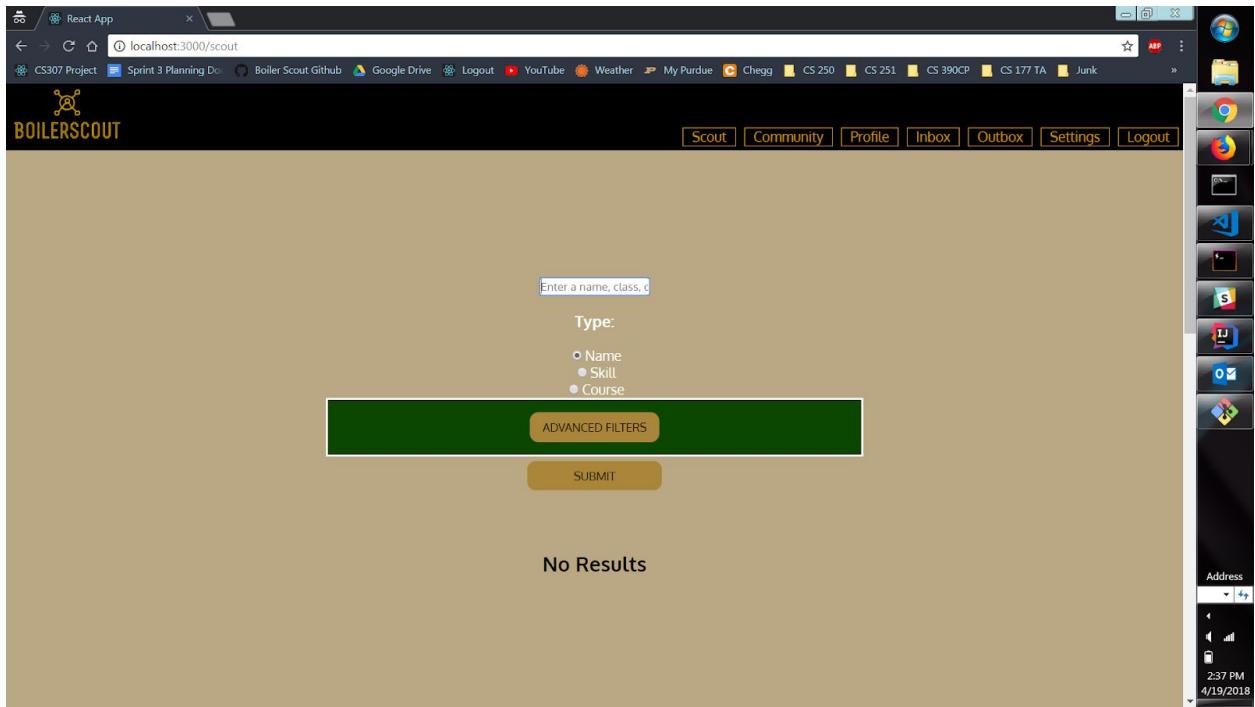
3. The user is notified that the updating of the password is successful



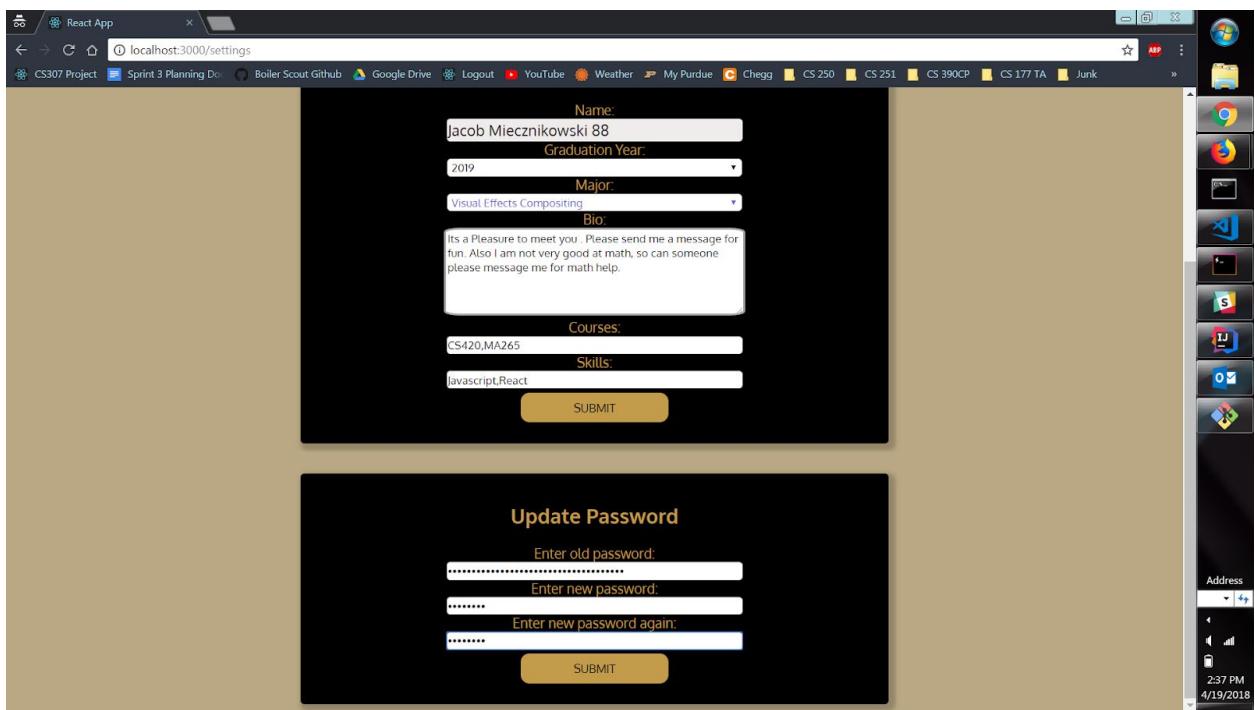
4. Now, we logout and go back to login with the new password to verify that it indeed did updated



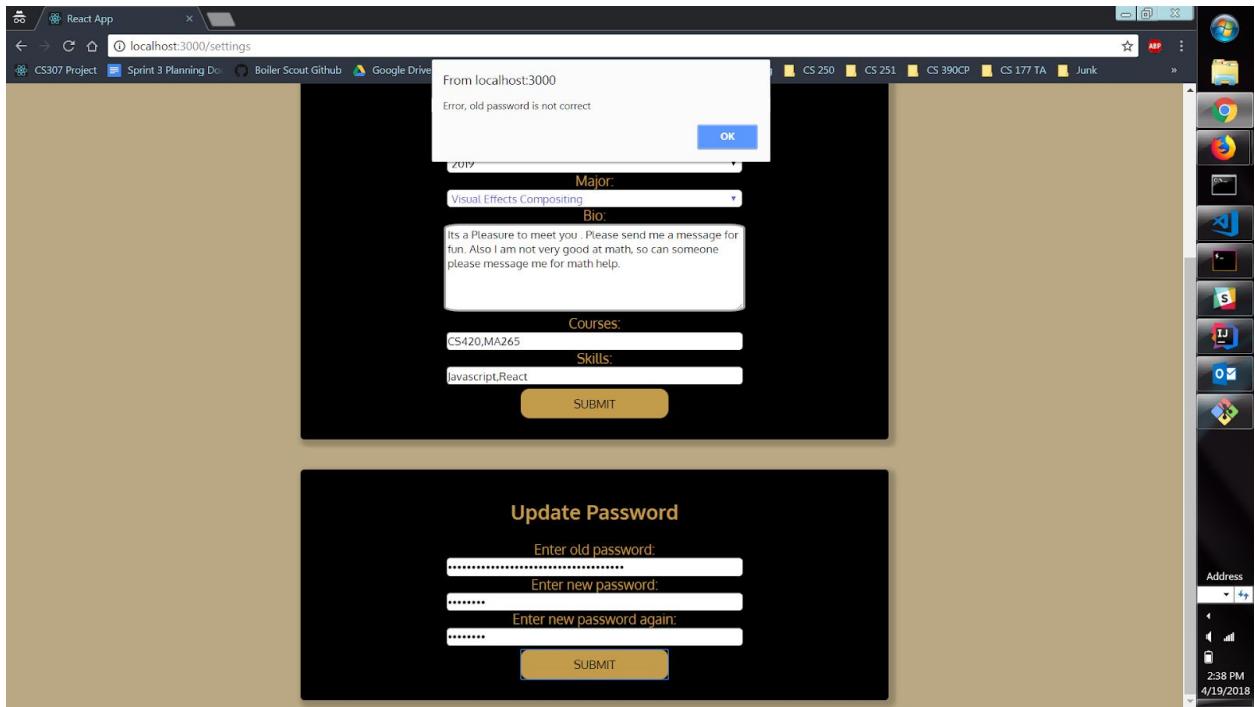
5. Login is successful as the user is redirected to Scout.



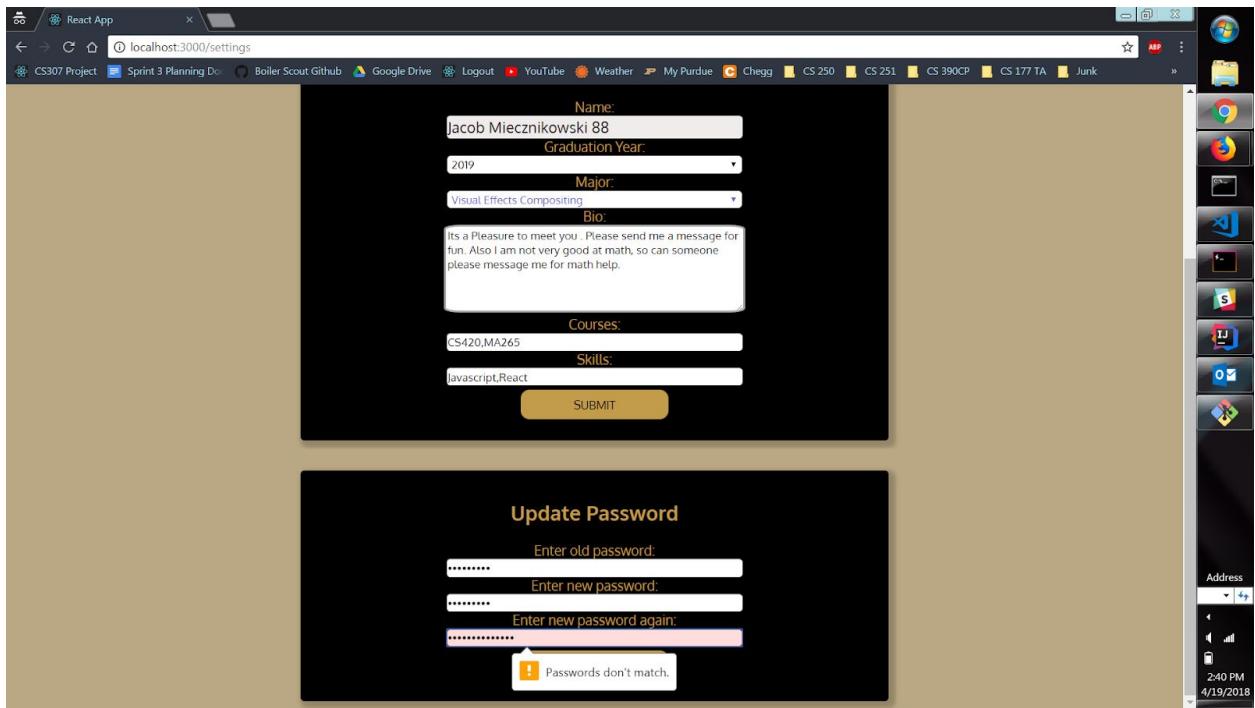
6. An incorrect old password is entered.



7. And the user is informed that the old password is incorrect.



8. The correct old password is entered, but now the new passwords do not match. The new password fields are both now red, indicating that they do not match. User is informed that the passwords do not match when Submit is clicked.

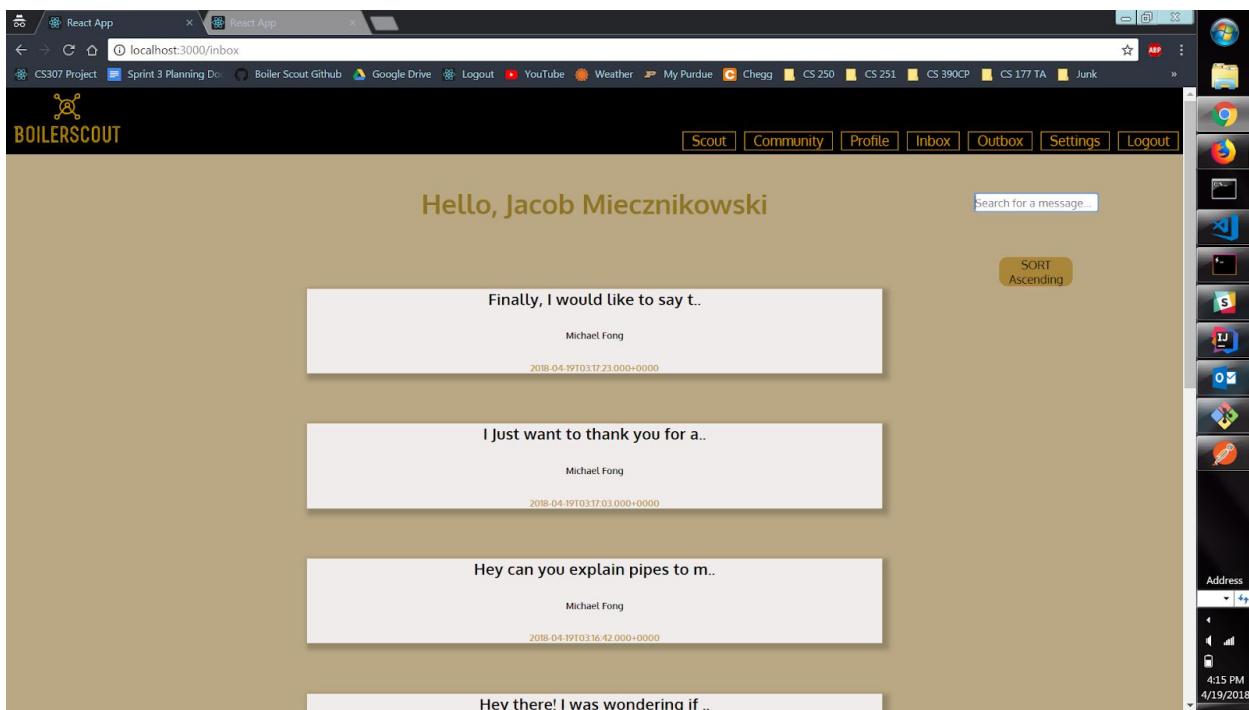


Testing Set Nine:

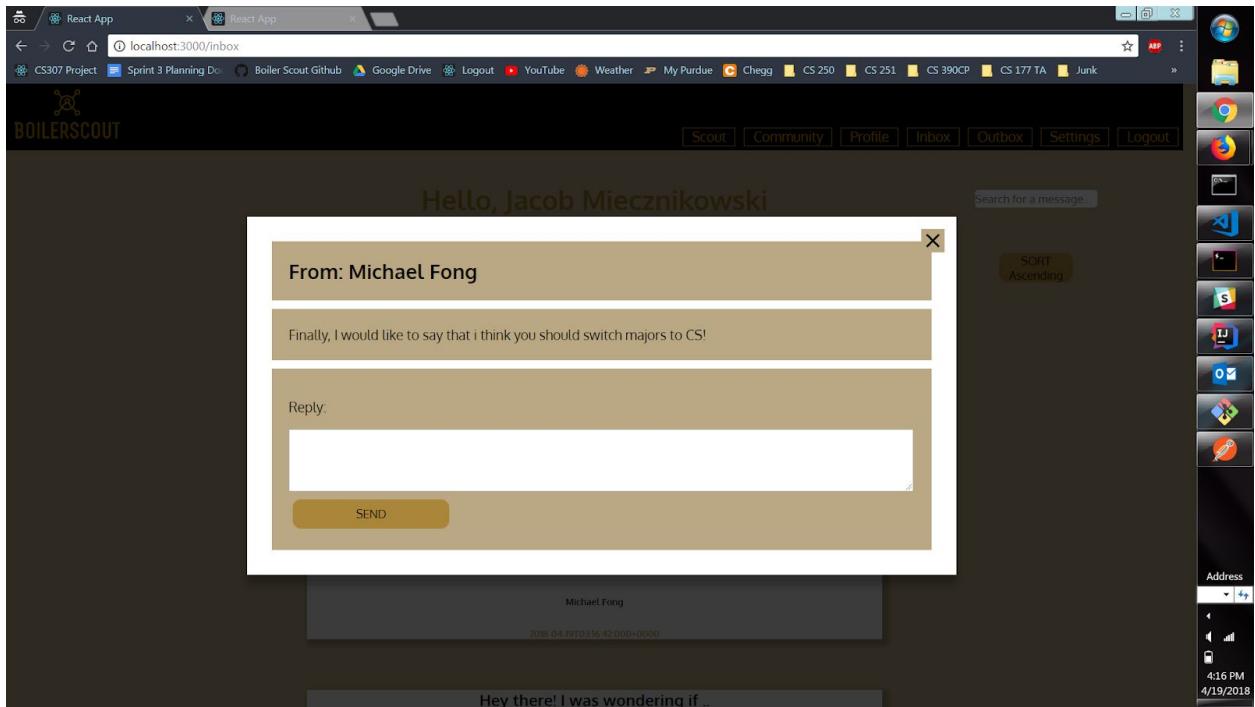
Testing Procedure: This is the protocol used to test the functionality of the inbox and outbox

- Login, and navigate to the Inbox view. Messages that have been received by the user will be displayed.
- Click on a message. The message should popup and show the user who it is from and the content of the message.
- There should be a box to reply, which when clicked sends a message to the user who the message is from.
- Navigate to the outbox. All the messages the user has sent should be displayed.
- When a message is clicked on, the full message should be displayed along with who it is to.

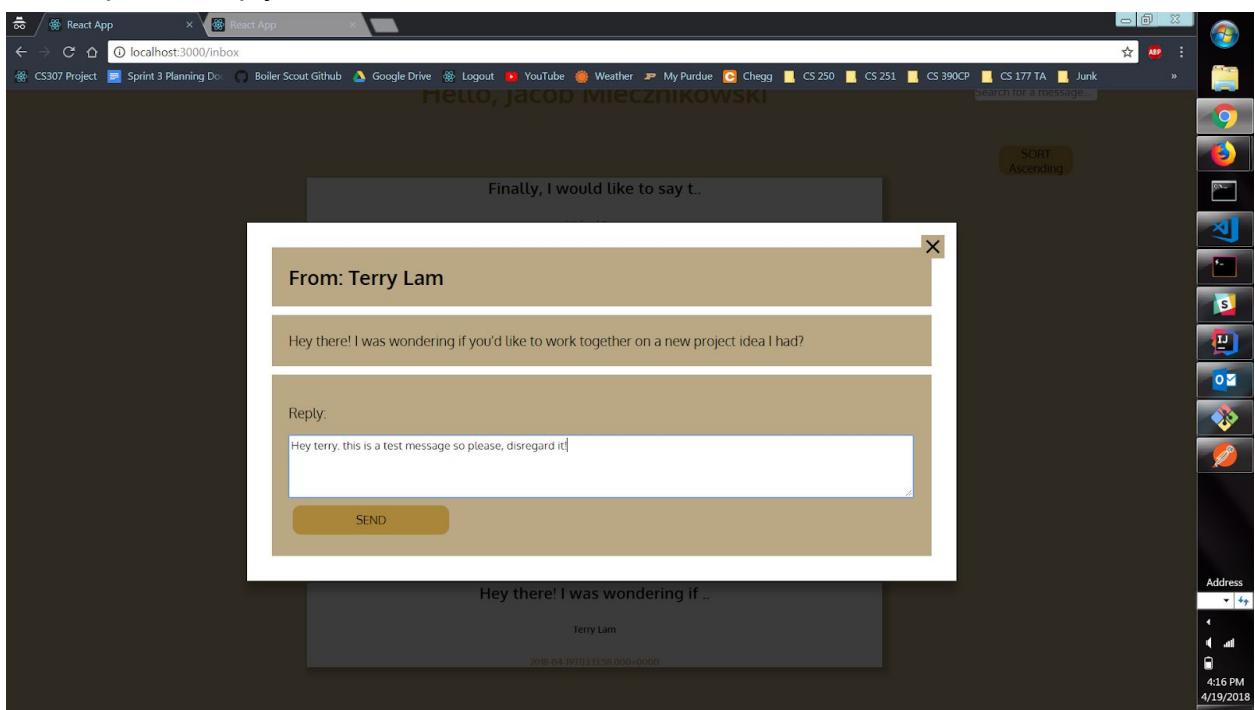
1. Navigate to Inbox. All the messages received are shown, along with who it is from.



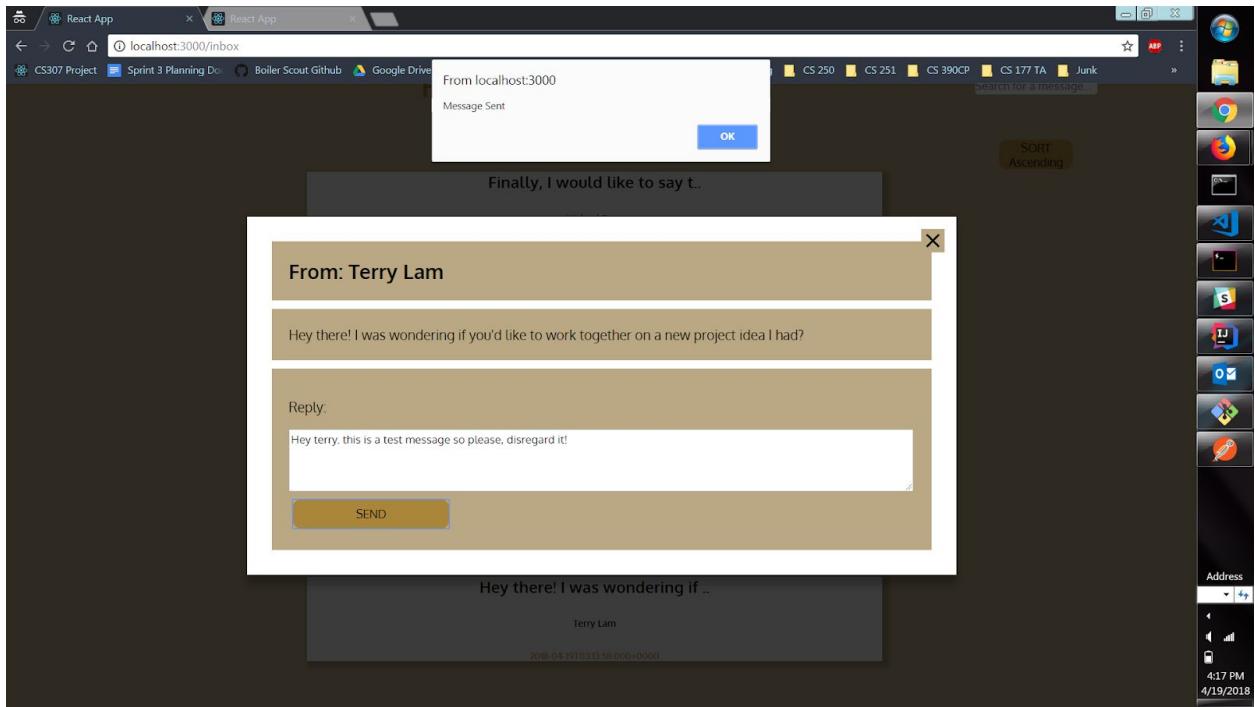
2. Click a message to see the full message.



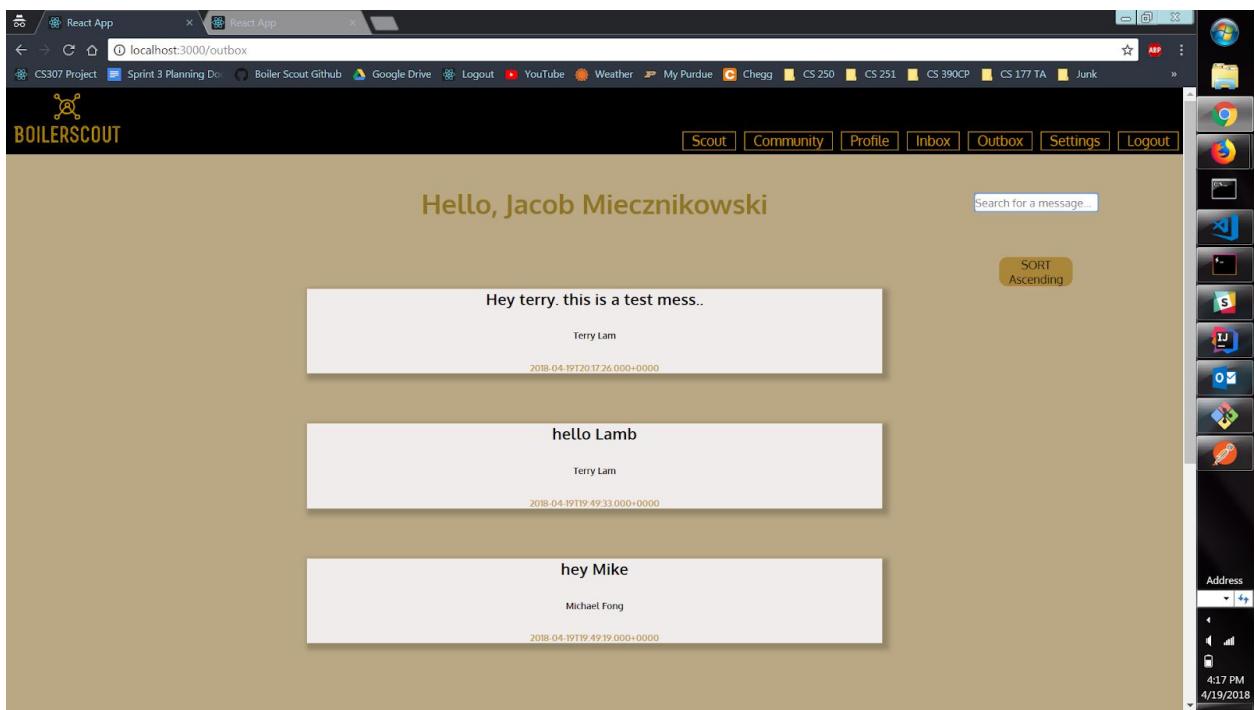
3. Compose a reply to the user.



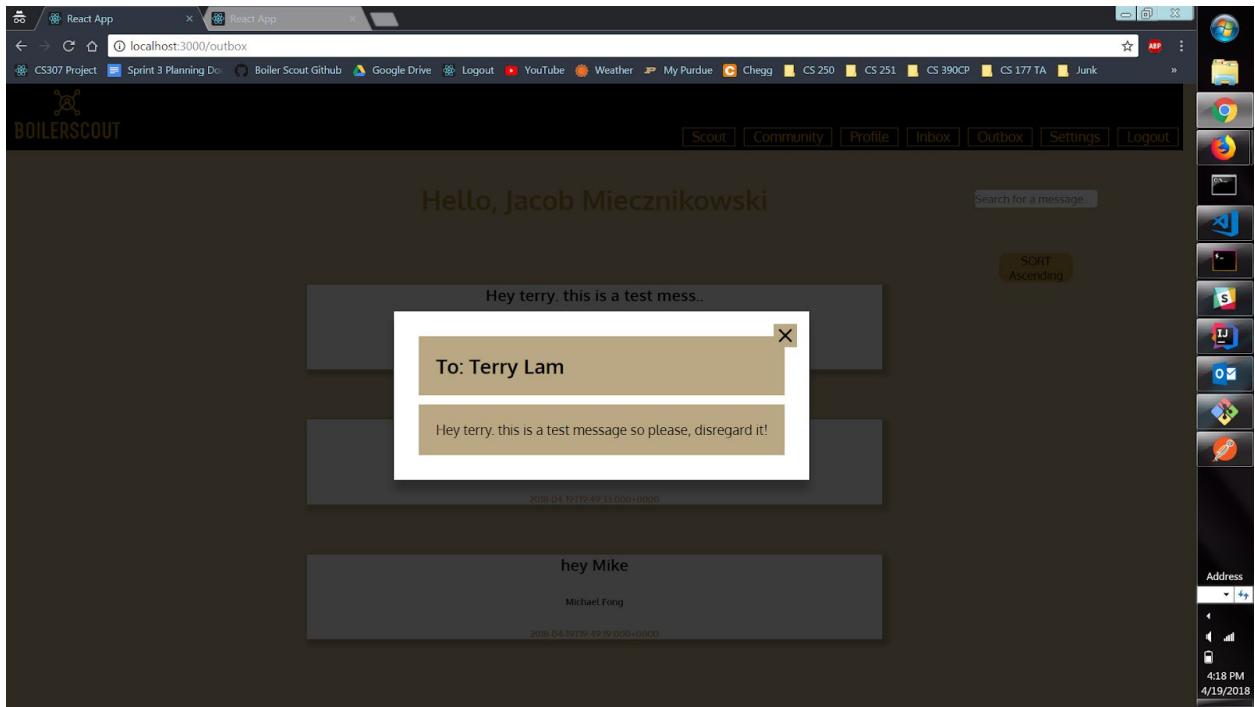
4. Send is clicked. The user is informed that the message has been sent.



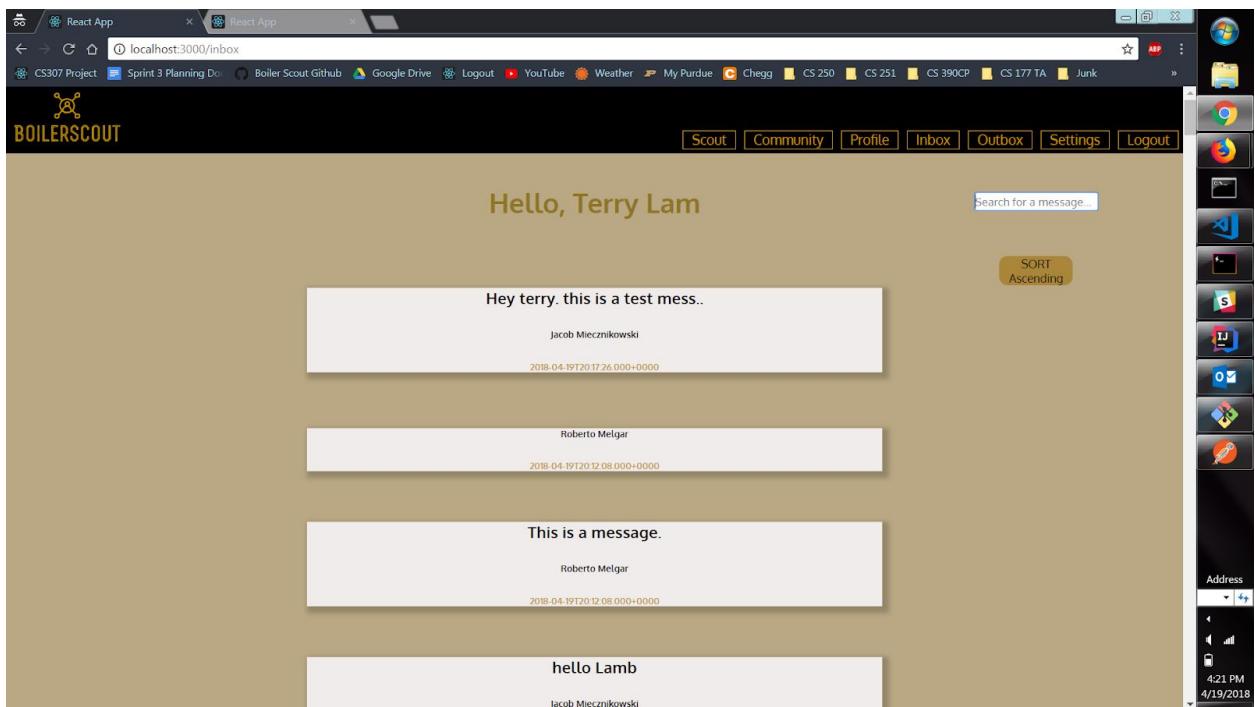
5. Navigate to Outbox. The message that was just sent is displayed.



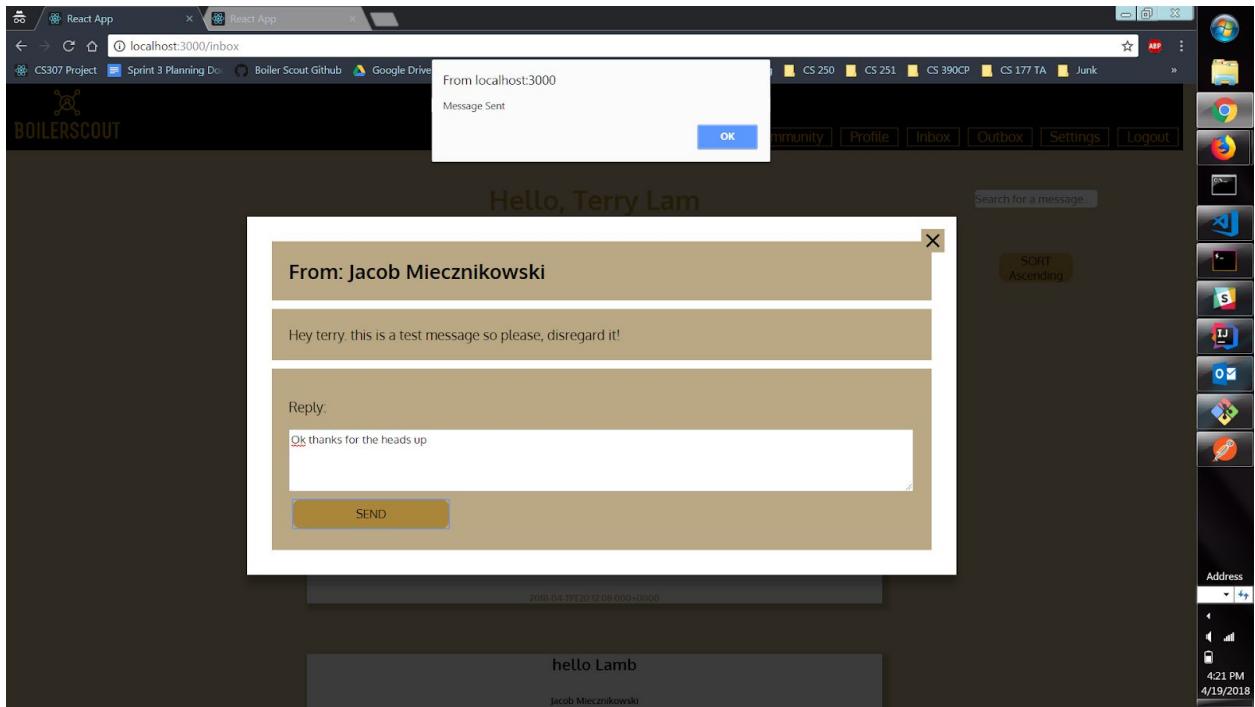
6. That message can be clicked and the entire message can be seen.



7. Now, logout and log back in as Terry Lam. Navigate to Inbox to confirm that the message from Jacob Miecznikowski that was just sent, is indeed there.



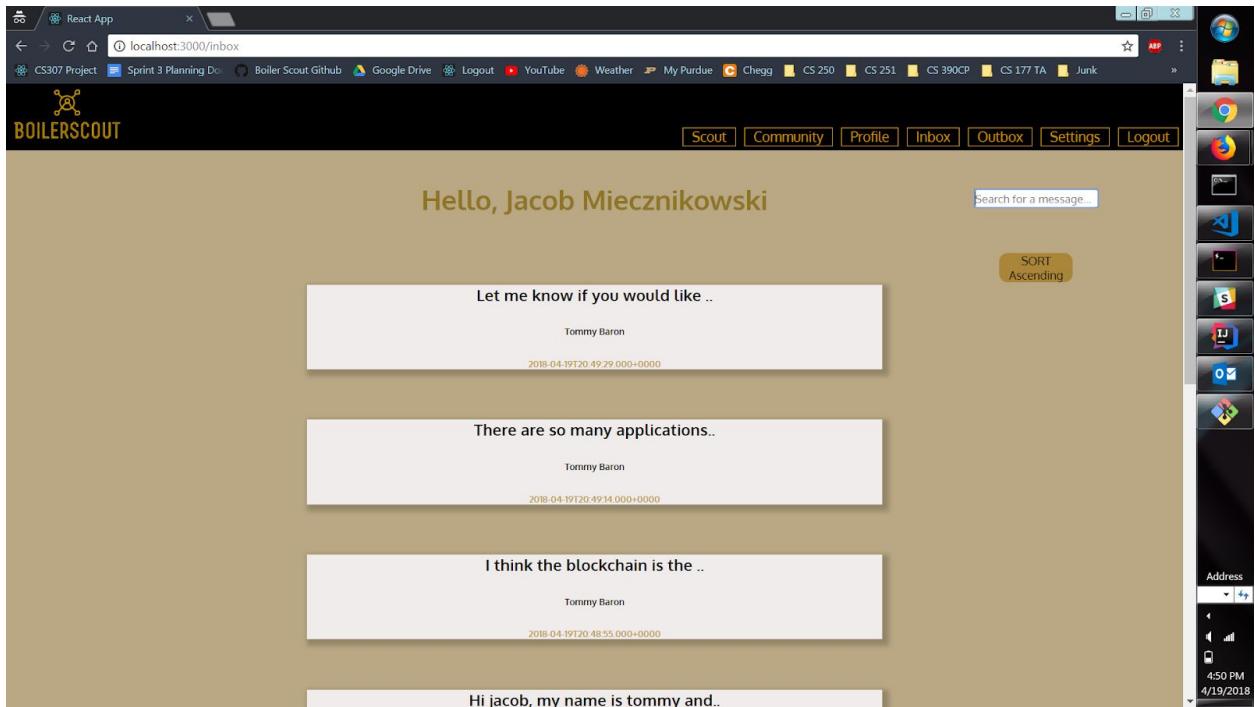
8. We can still click this message to view it, and send a message back!



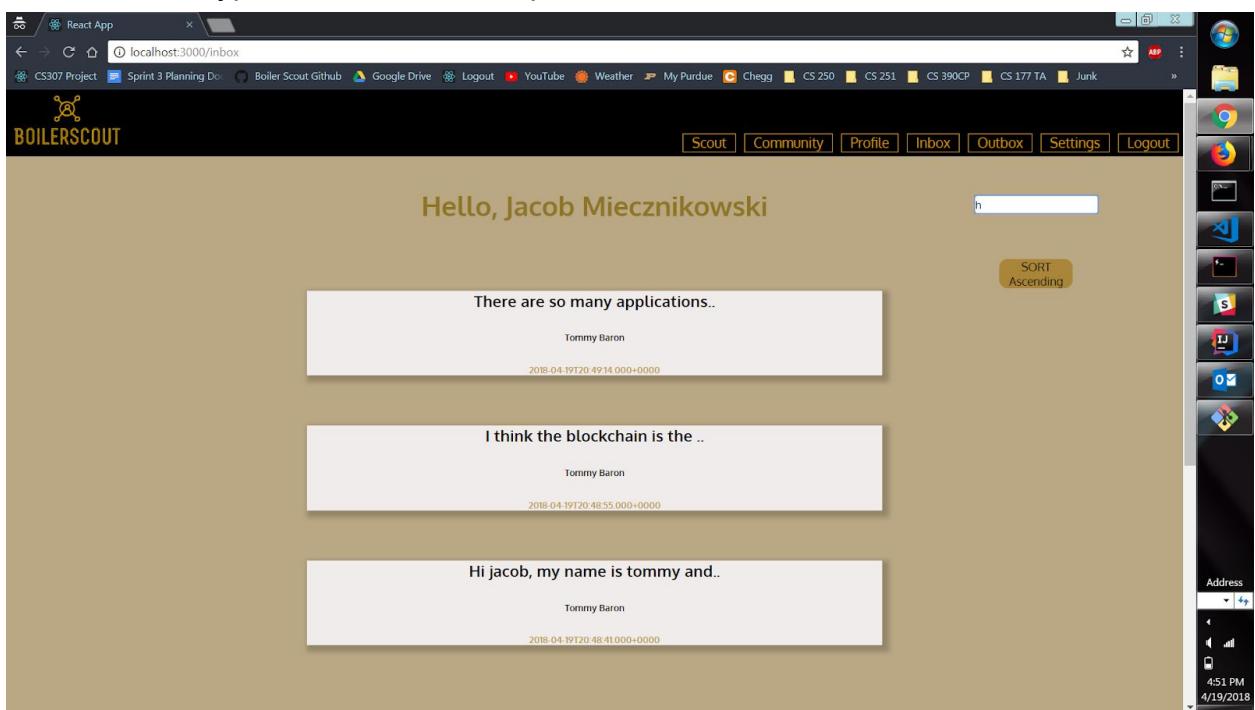
Testing Set Eleven:

Testing Procedure: This is the protocol used to test the functionality of the inbox/outbox search.

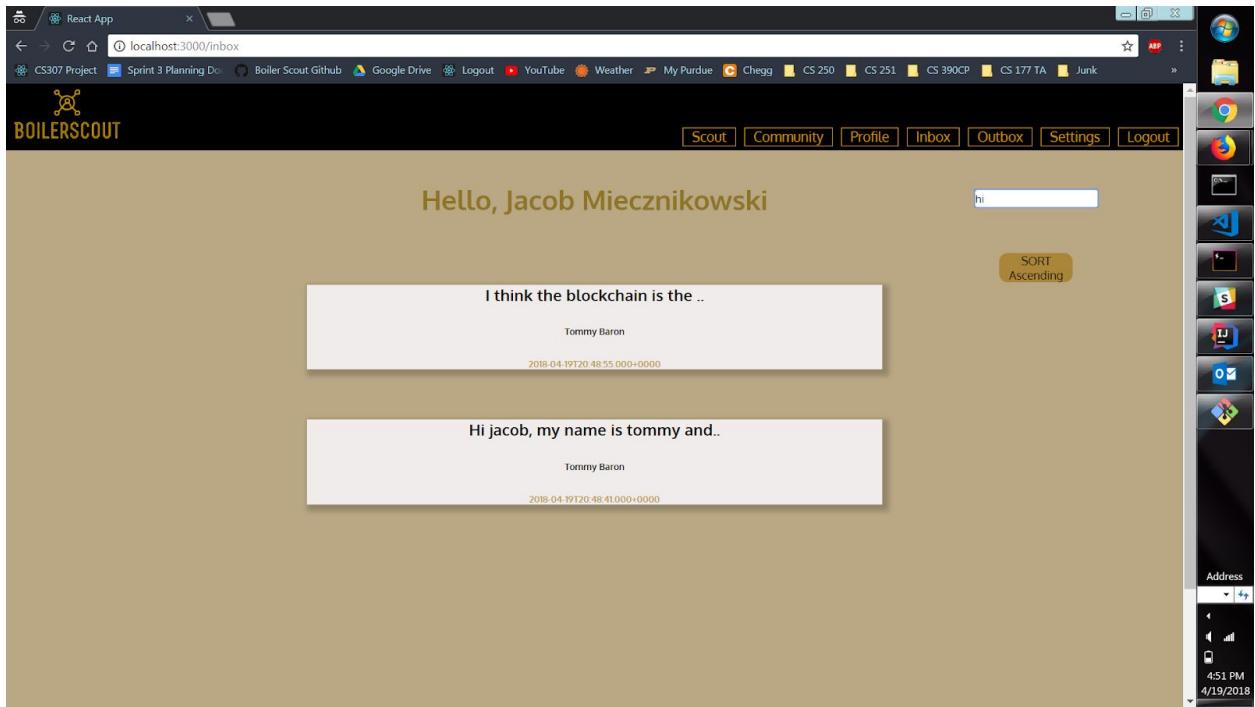
- Login, and navigate to the Inbox view. Messages will be displayed as they are when nothing is in the search box.
 - As the user types, the messages displayed will be updated in real time to reflect what is in the search box
 - If input that does not match any message, none will be displayed.
1. Login, and navigate to the Inbox view. Messages will be displayed as they are when nothing is in the search box.



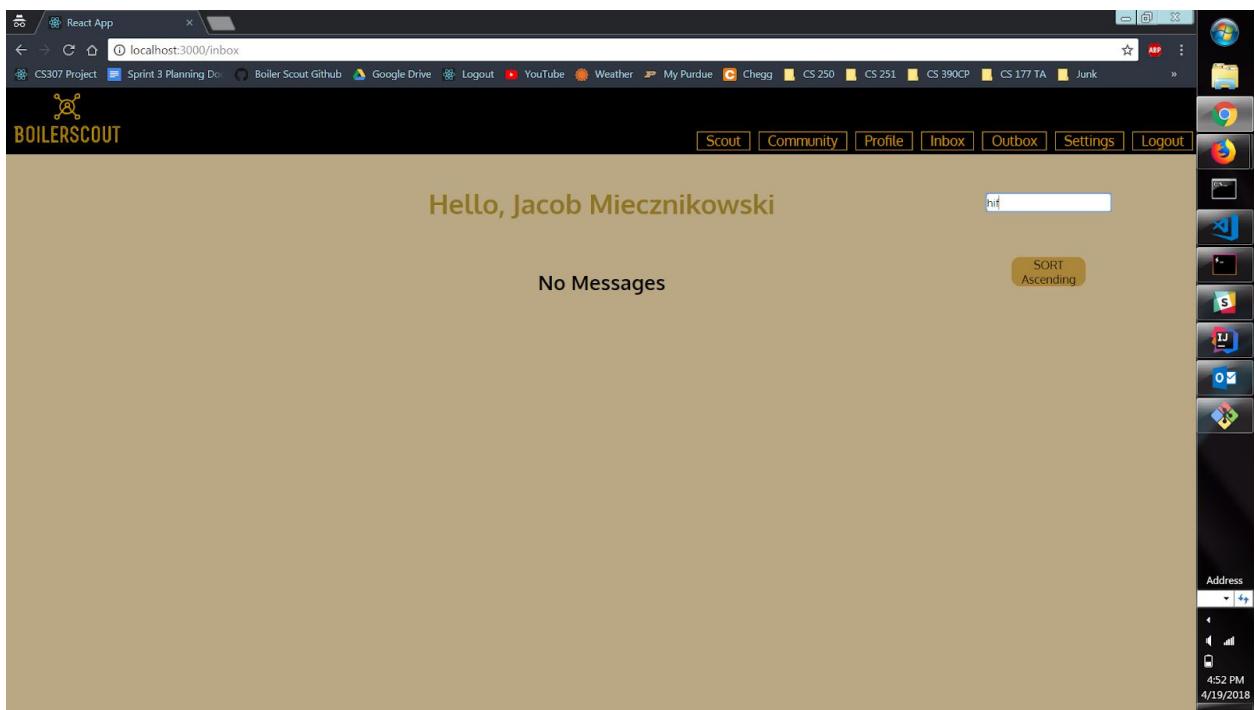
2. As the user types, the results will update in real time.



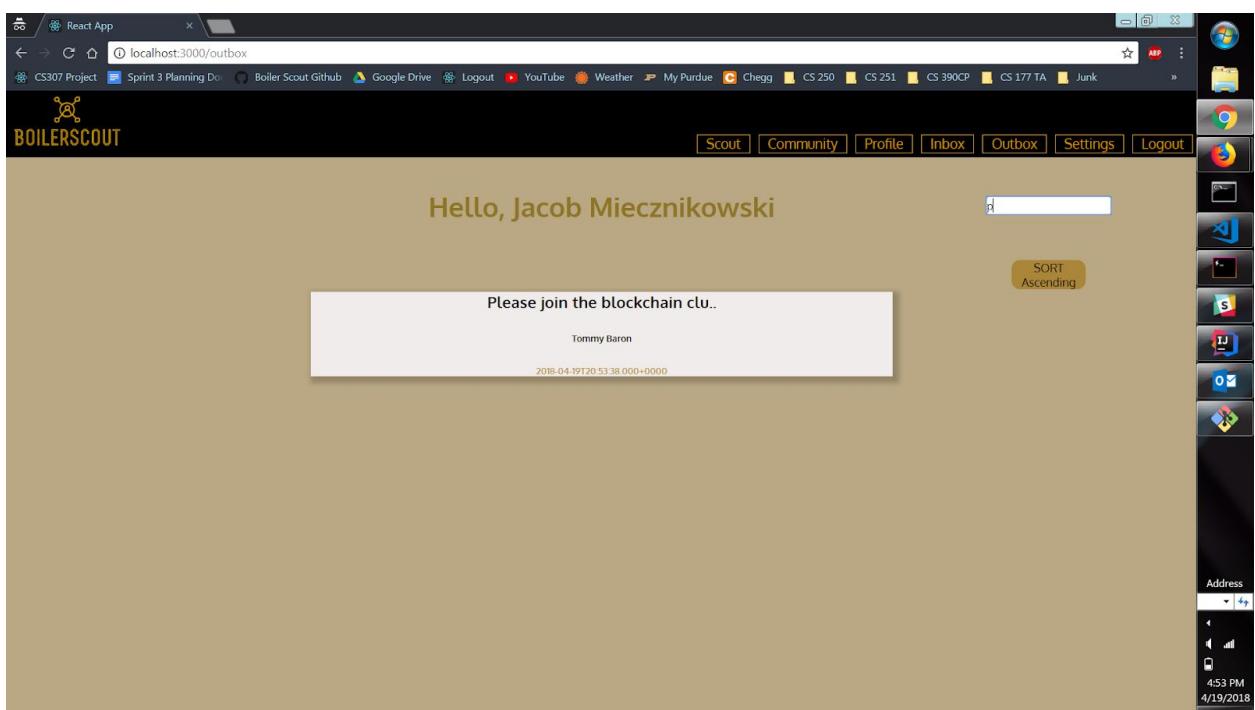
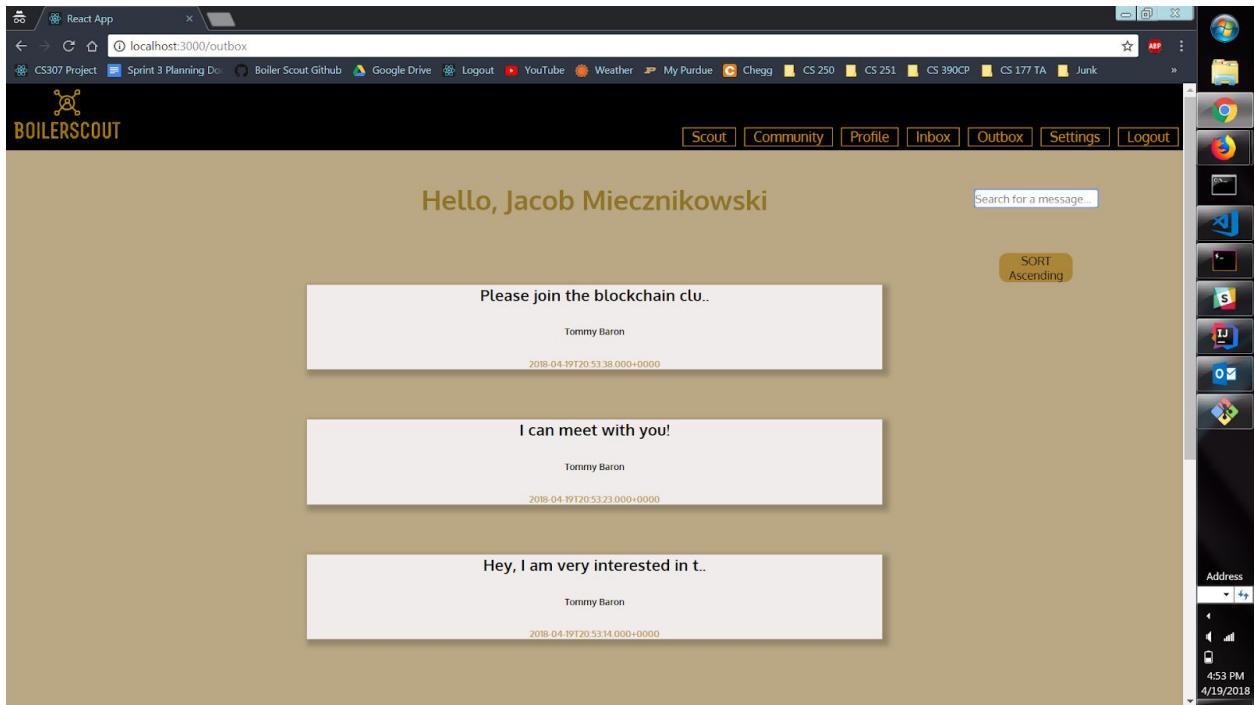
3. As more letters are typed, the results get filtered down more.

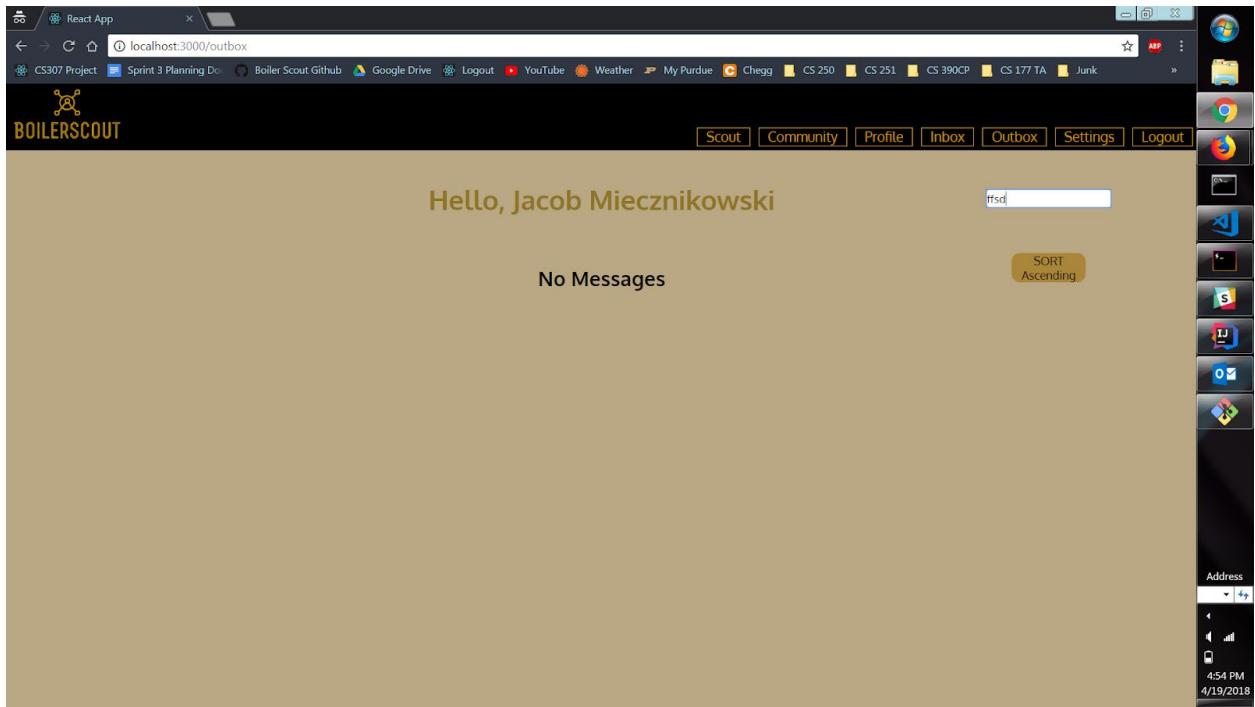


4. If the search doesn't match a message, then nothing is displayed.



5. The same exact tests can be shown in the outbox view.





Testing Set Twelve:

Testing Procedure: This is the protocol used to test the functionality of the inbox/outbox sort.

- Login and go to the inbox view. Messages should be sorted from most recent to least recent.
- When the sort button is clicked, the messages will reverse order.
- When clicked again, they will be back the most recent first order.
- These same tests will be used for the outbox view too.

1. Go to inbox view and make sure that messages are in order of most recent first.

The screenshot shows the Boilerscout inbox interface. At the top, there's a navigation bar with links for Scout, Community, Profile, Inbox, Outbox, Settings, and Logout. Below the navigation bar, the title "Hello, Jacob Miecznikowski" is displayed, along with a search bar labeled "Search for a message...". A "SORT Ascending" button is visible. The main content area contains several message cards:

- A message from Terry Lam: "Hey Jacob! How are you?" dated 2018-04-19T20:58:59.000+0000.
- A message from Tommy Baron: "Let me know if you would like .." dated 2018-04-19T20:49:29.000+0000.
- A message from Tommy Baron: "There are so many applications.." dated 2018-04-19T20:49:14.000+0000.
- A message from Tommy Baron: "I think the blockchain is the ..." which is partially cut off.

On the right side of the screen, there's a vertical sidebar with icons for various applications like Google Drive, YouTube, and My Purdue. The system tray at the bottom right shows the date as 4/19/2018 and the time as 5:16 PM.

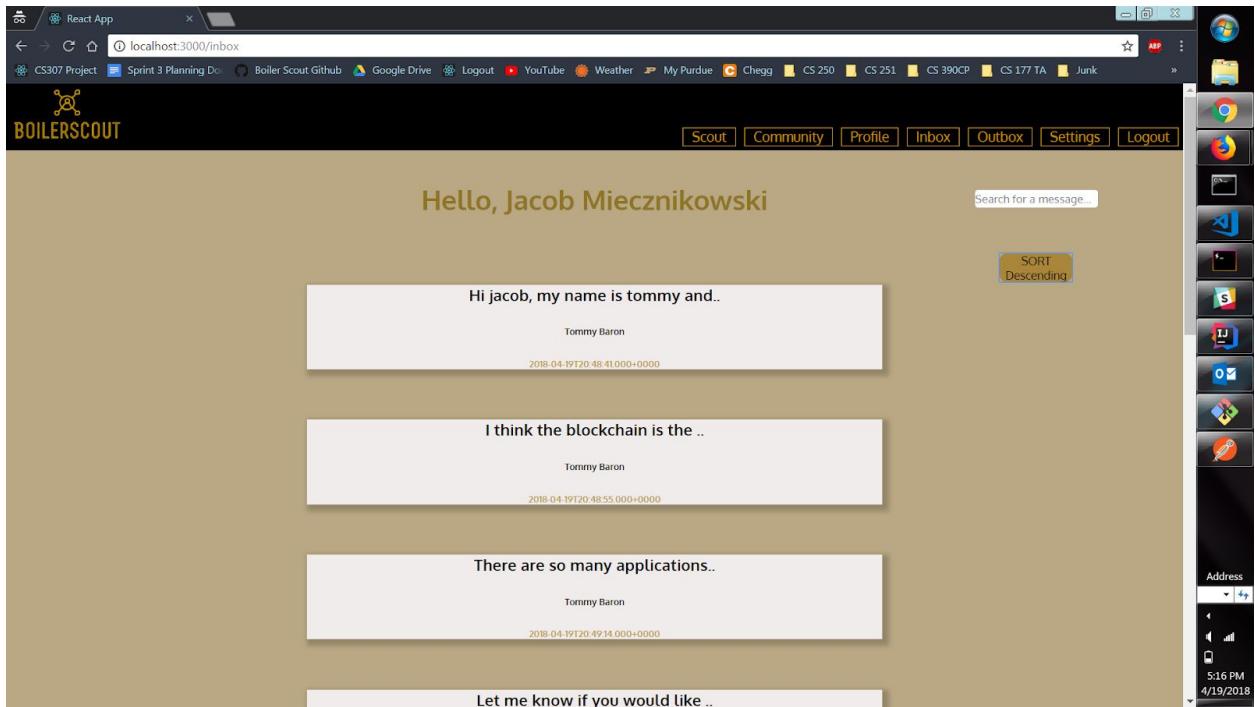
2. Go to outbox view and make sure that messages are in order of most recent first.

The screenshot shows the Boilerscout outbox interface. The layout is similar to the inbox, with a navigation bar at the top and a title "Hello, Jacob Miecznikowski" followed by a search bar. A "SORT Ascending" button is present. The message list is as follows:

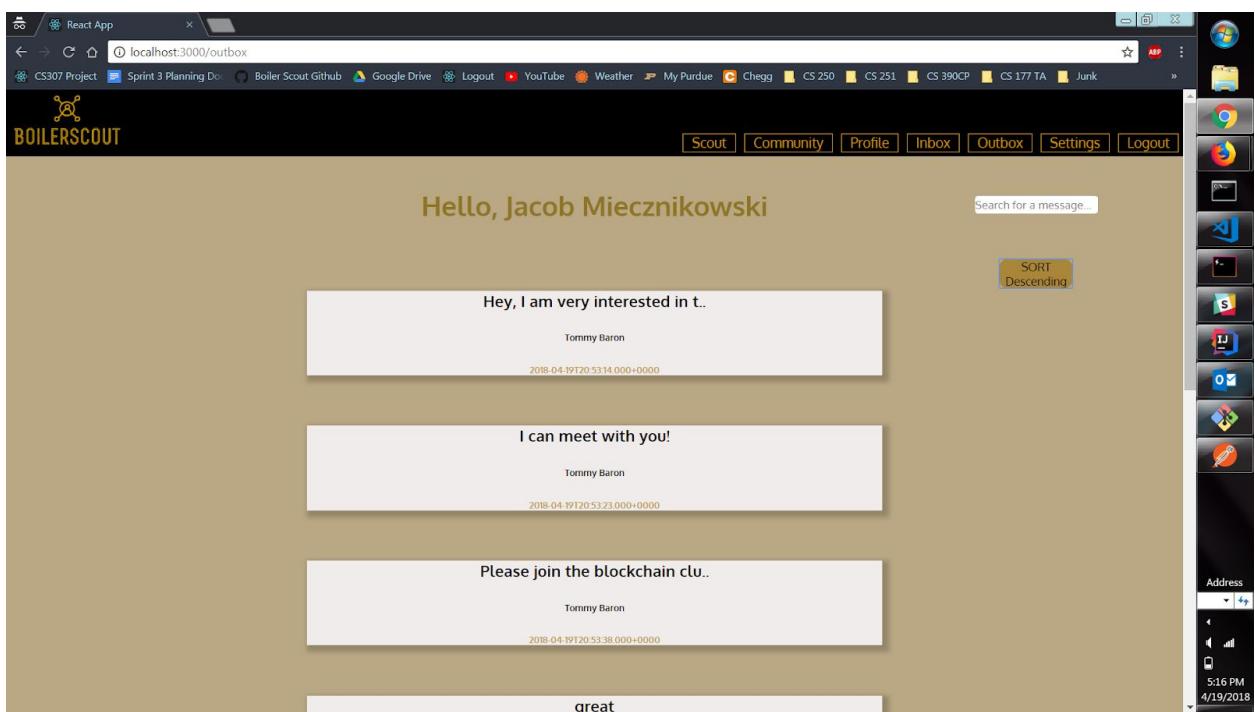
- A message to Terry Lam: "great" dated 2018-04-19T20:59:11.000+0000.
- A message to Tommy Baron: "Please join the blockchain clu.." dated 2018-04-19T20:53:38.000+0000.
- A message to Tommy Baron: "I can meet with you!" dated 2018-04-19T20:53:23.000+0000.
- A message to Tommy Baron: "Hey, I am very interested in t..." which is partially cut off.

The right sidebar and system tray are identical to the inbox screenshot.

3. When the sort button is clicked in inbox, the messages will be sorted from least recent to most recent.



4. When the sort button is clicked in outbox, the messages will be sorted from least recent to most recent.

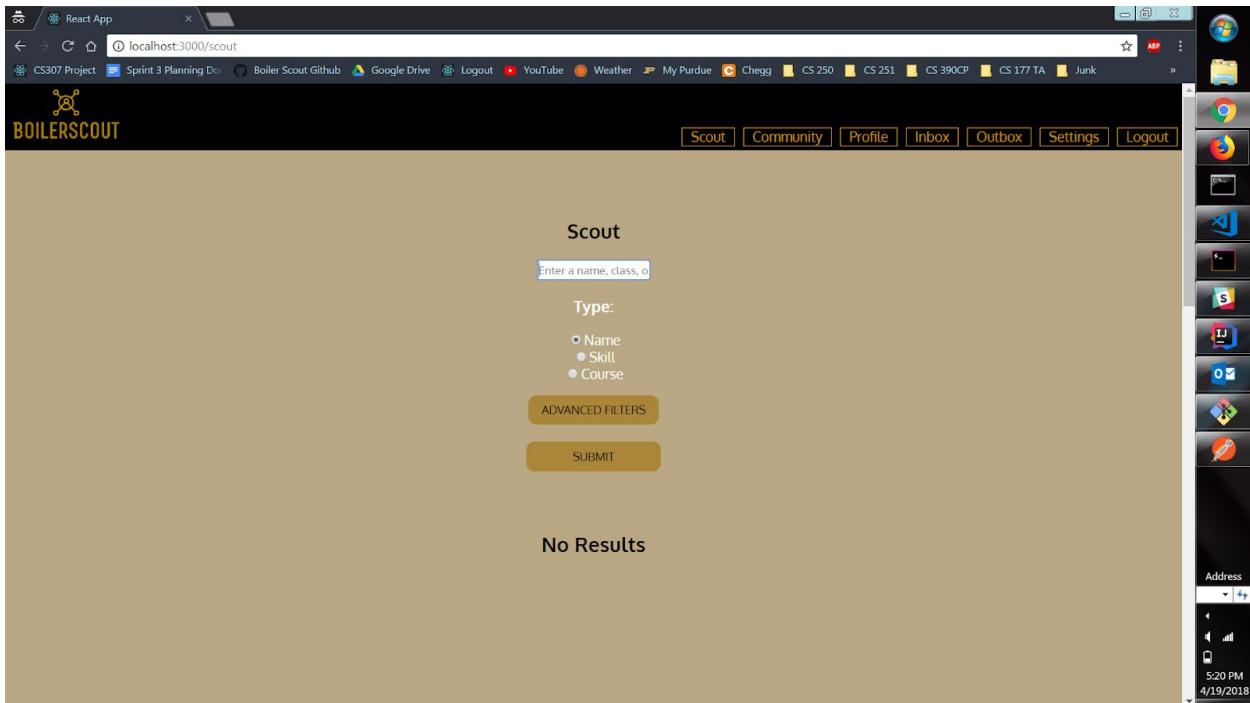


Testing Set Thirteen:

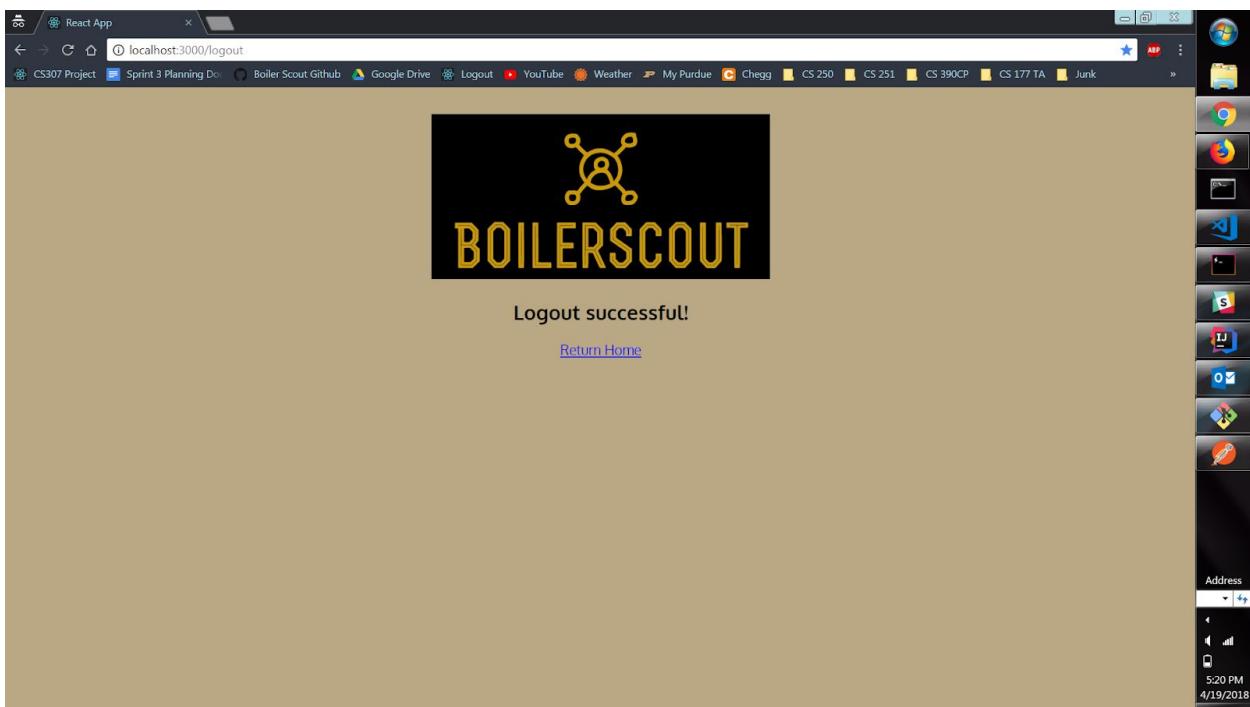
Testing Procedure: This is the protocol used to test the functionality logging out.

- Login . Then hit logout button. You should be notified that you were logged out.
- Try navigating to arbitrary pages within the app. You should be kicked out to the home view because you are not logged in.

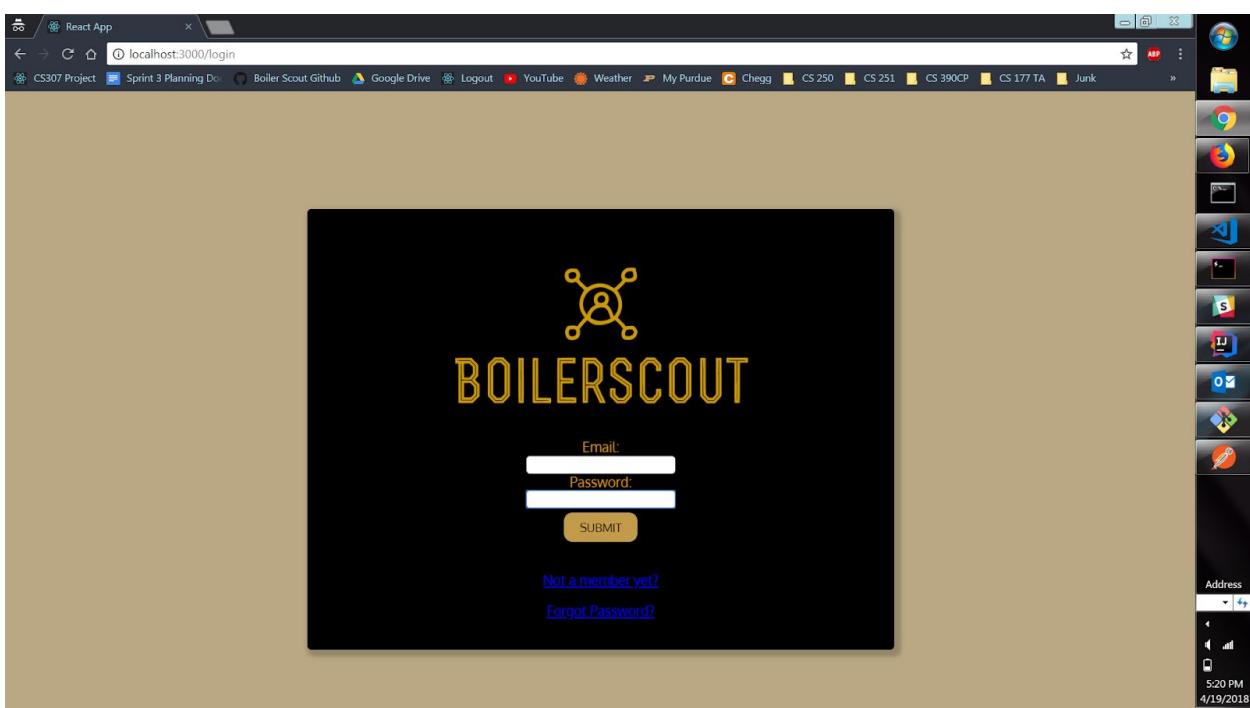
1. User logs in



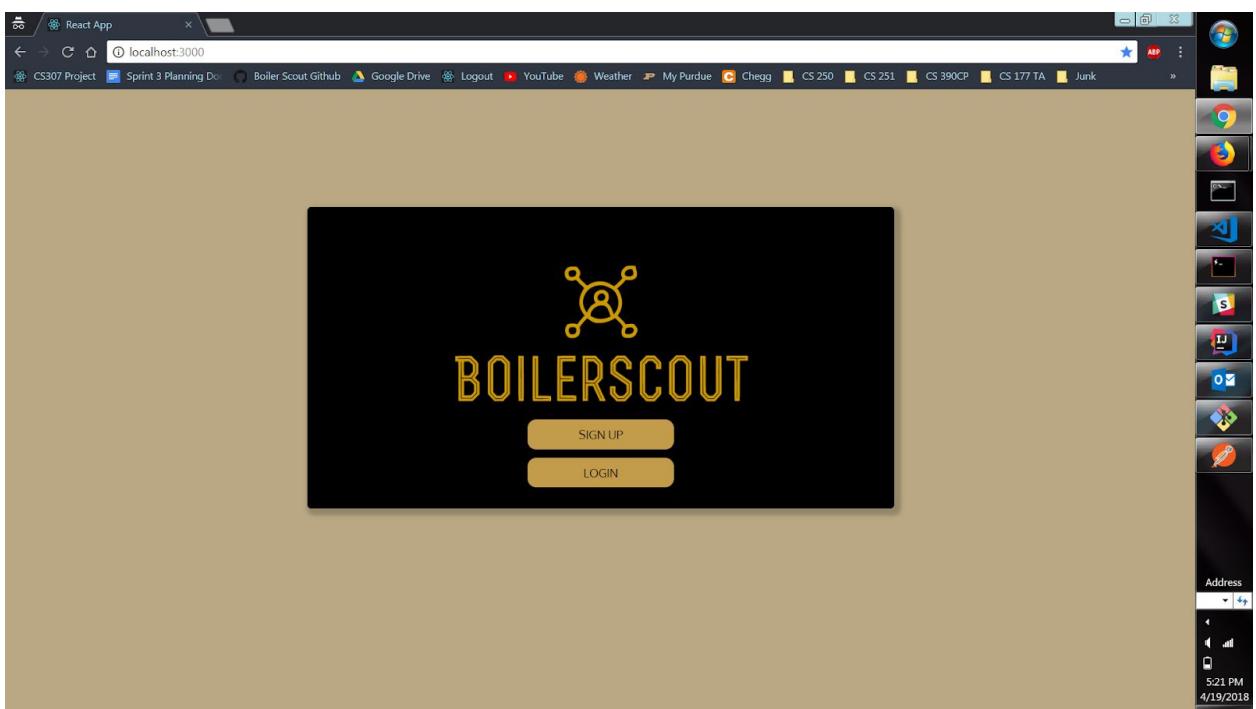
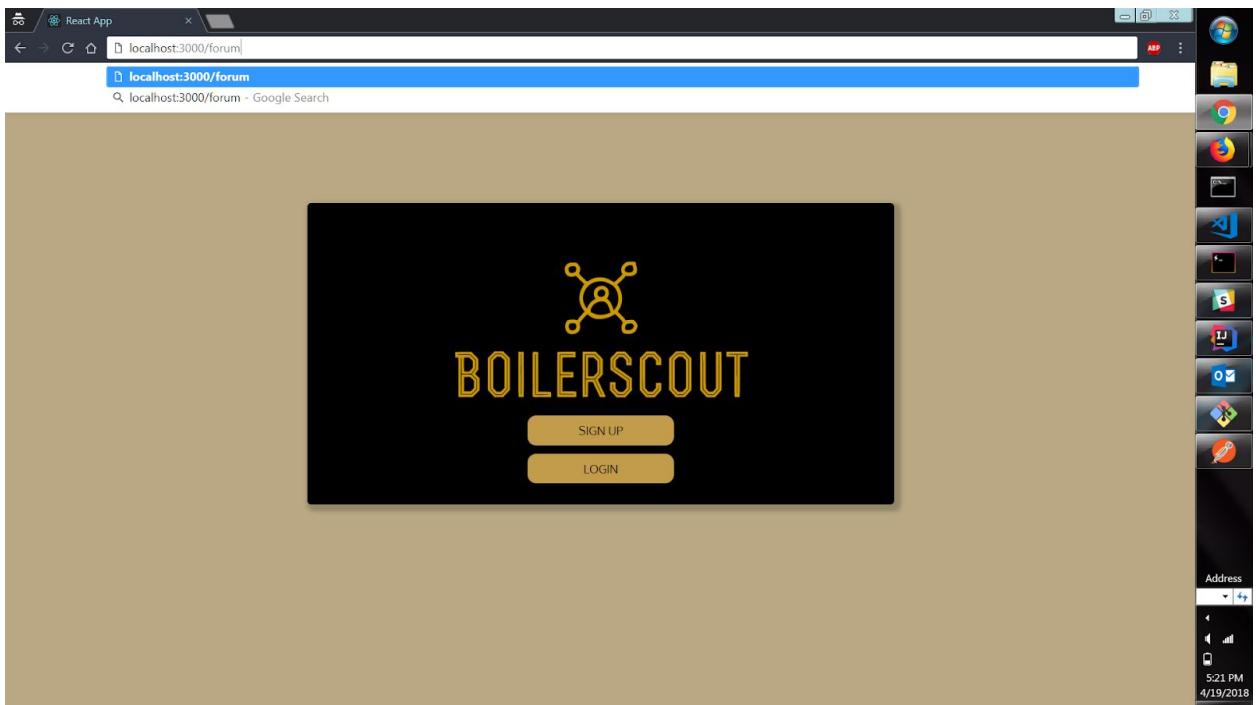
2. User clicks log out



3. Try to navigate to /scout
4. User is kicked back to home



- Type in /forum after the base url. You should be kicked to home



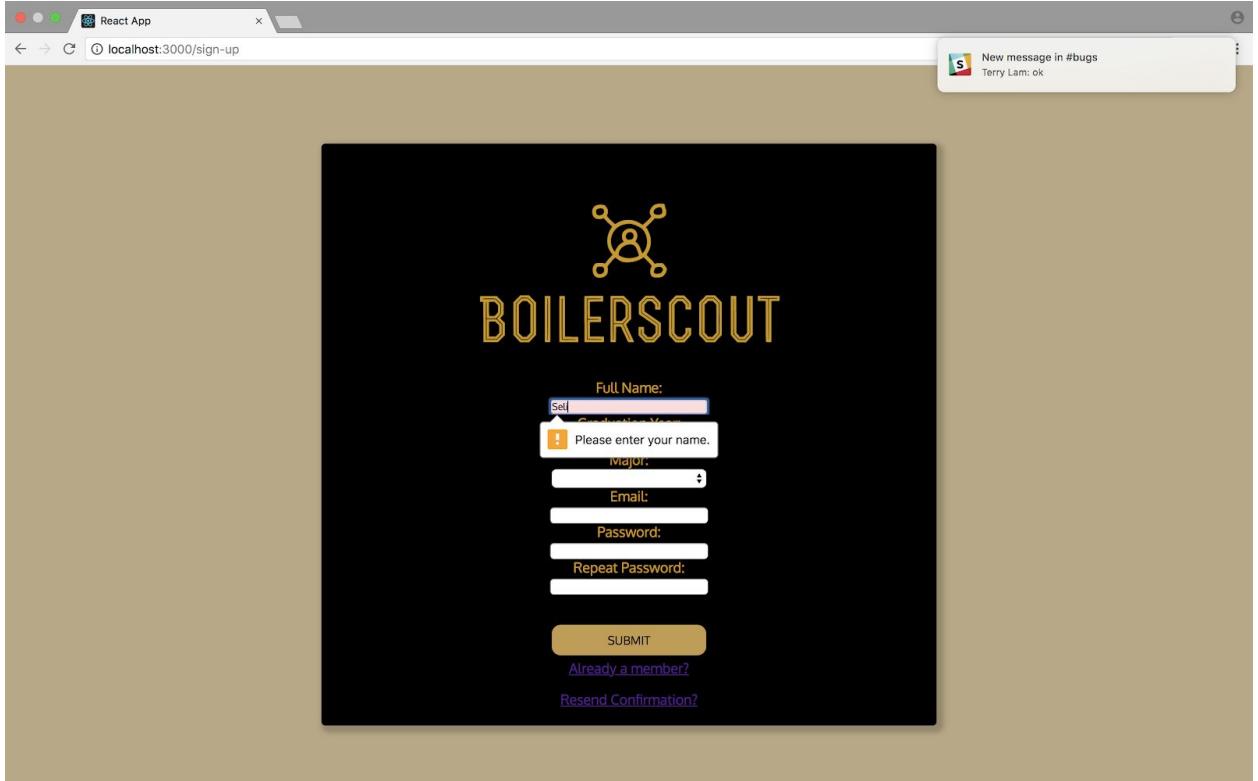
Testing Set 13

Testing Procedure: This is the protocol used to test the input validation on Sign Up.

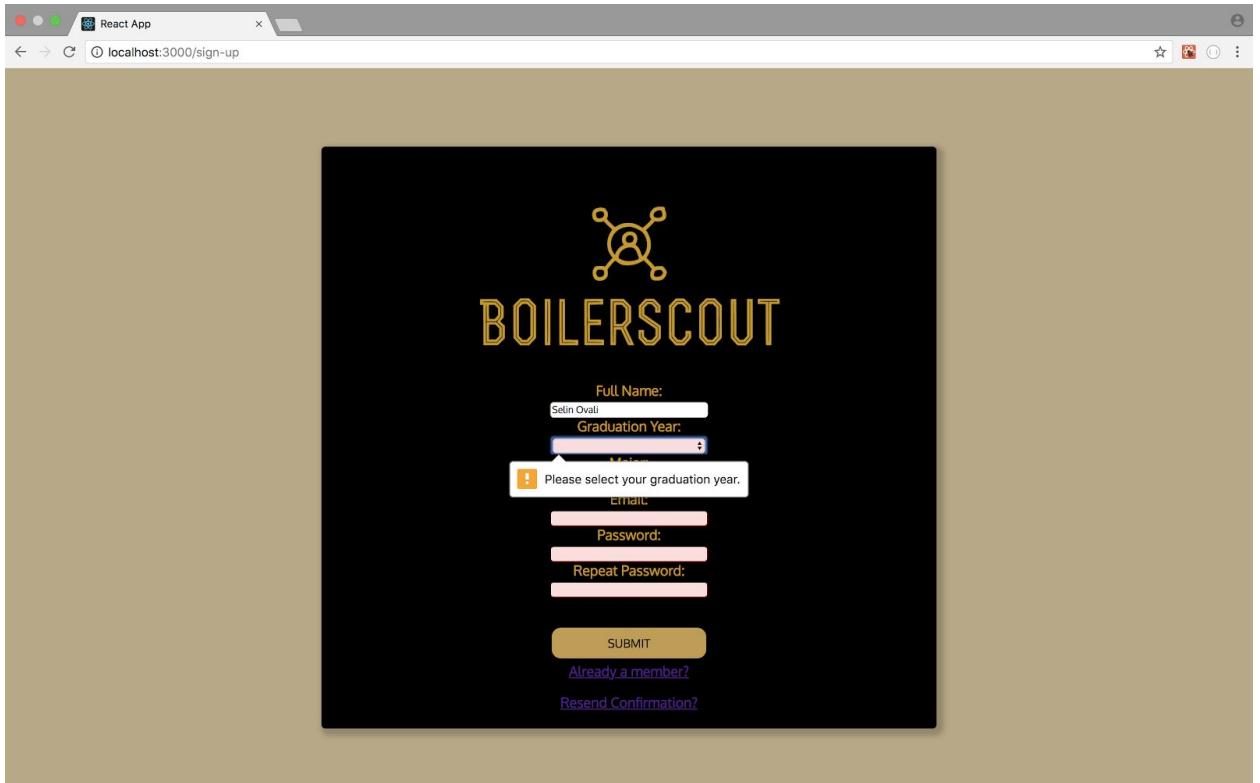
- Users should enter a full name with space in between or get a visual error.

- Users should select a graduation year or get a visual error.
- Users should select a major or get a visual error.
- Users should enter a Purdue email or get a visual error.
- Users should enter a password that has one digit, one uppercase, one special character and be 8-12 length.

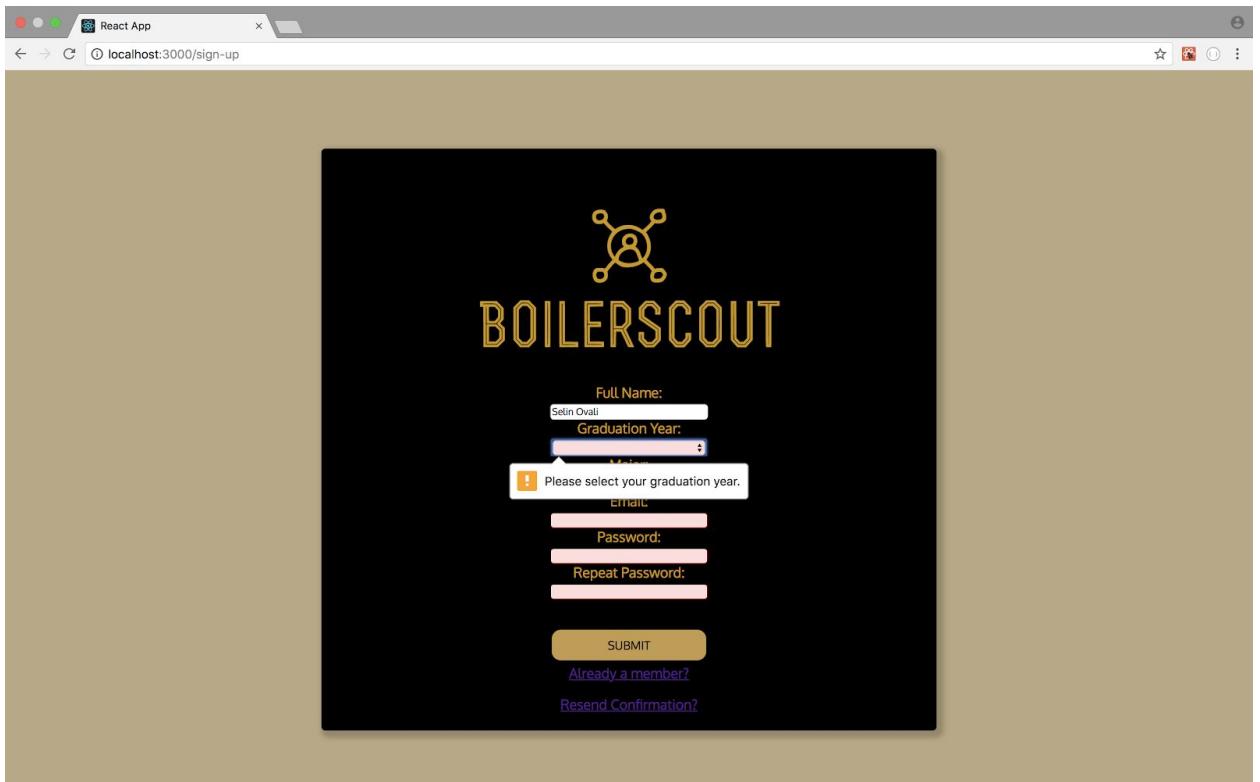
1. Users should enter a full name with space in between or get a visual error.



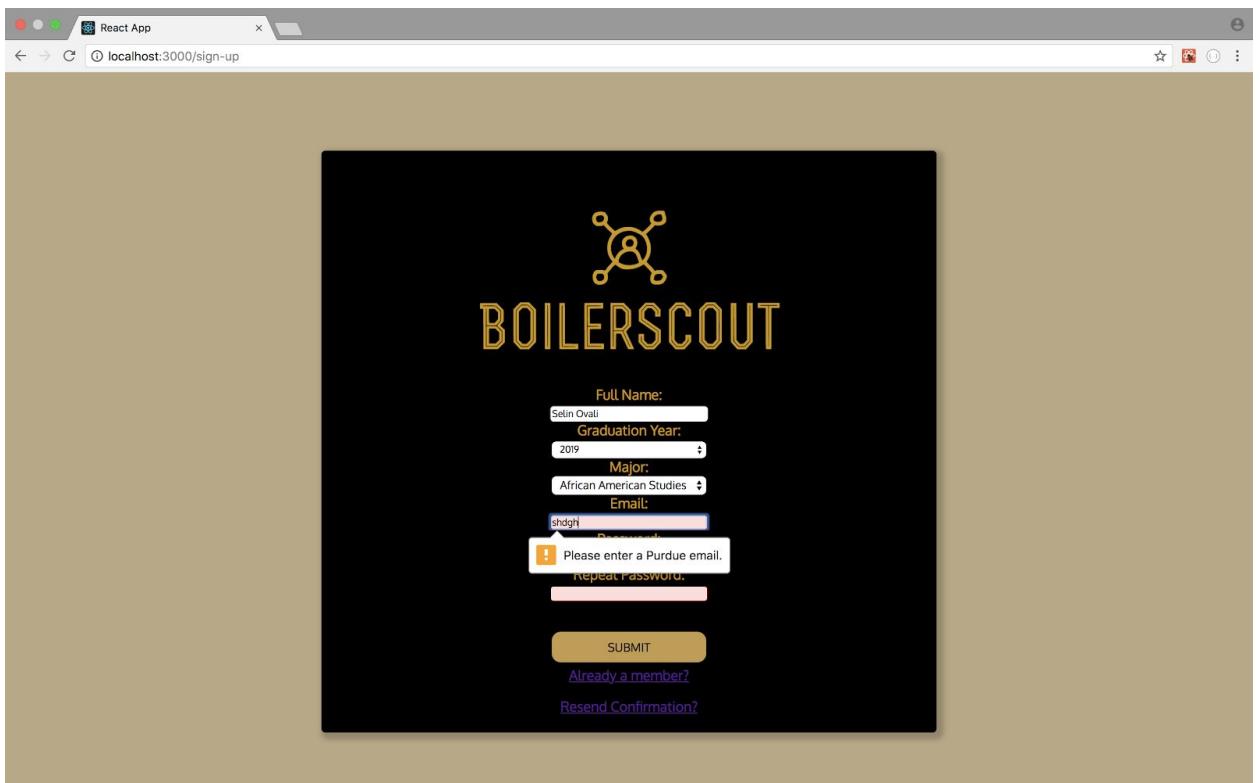
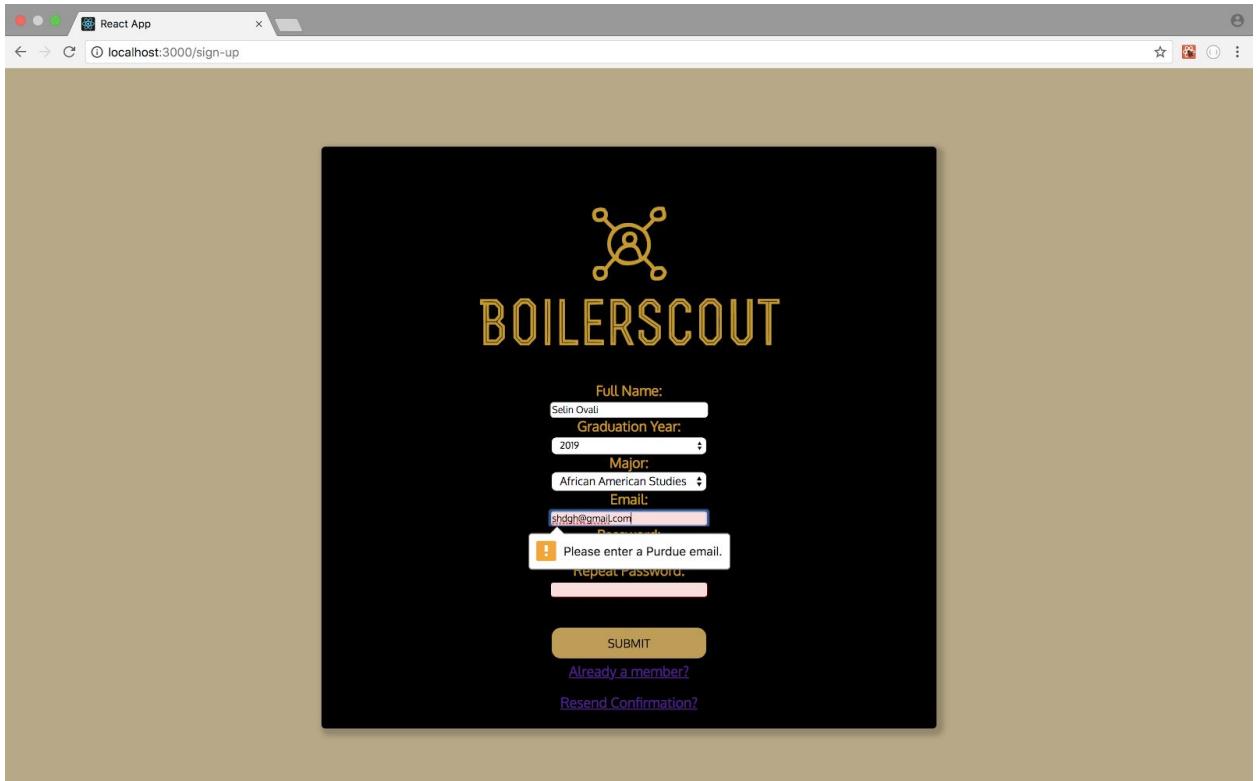
2. Users should select a graduation year or get a visual error.



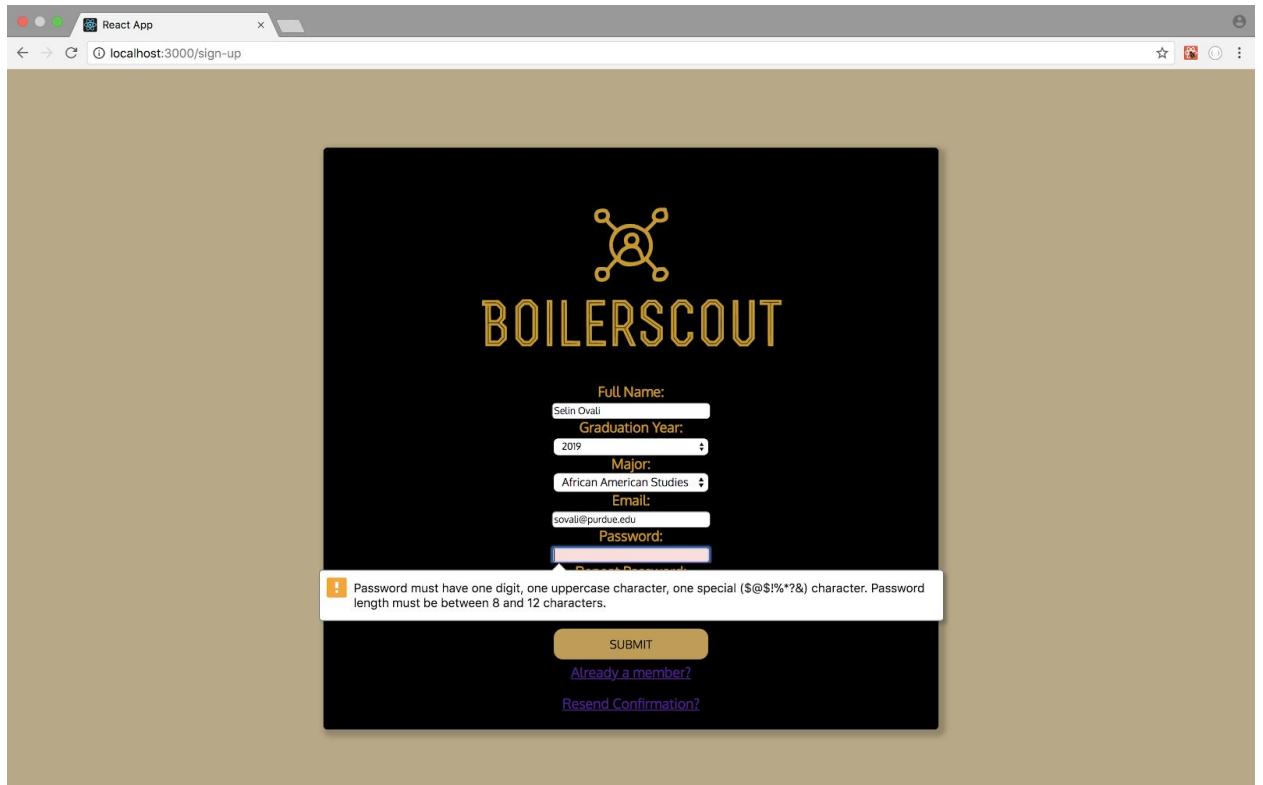
3. Users should select a major or get a visual error.



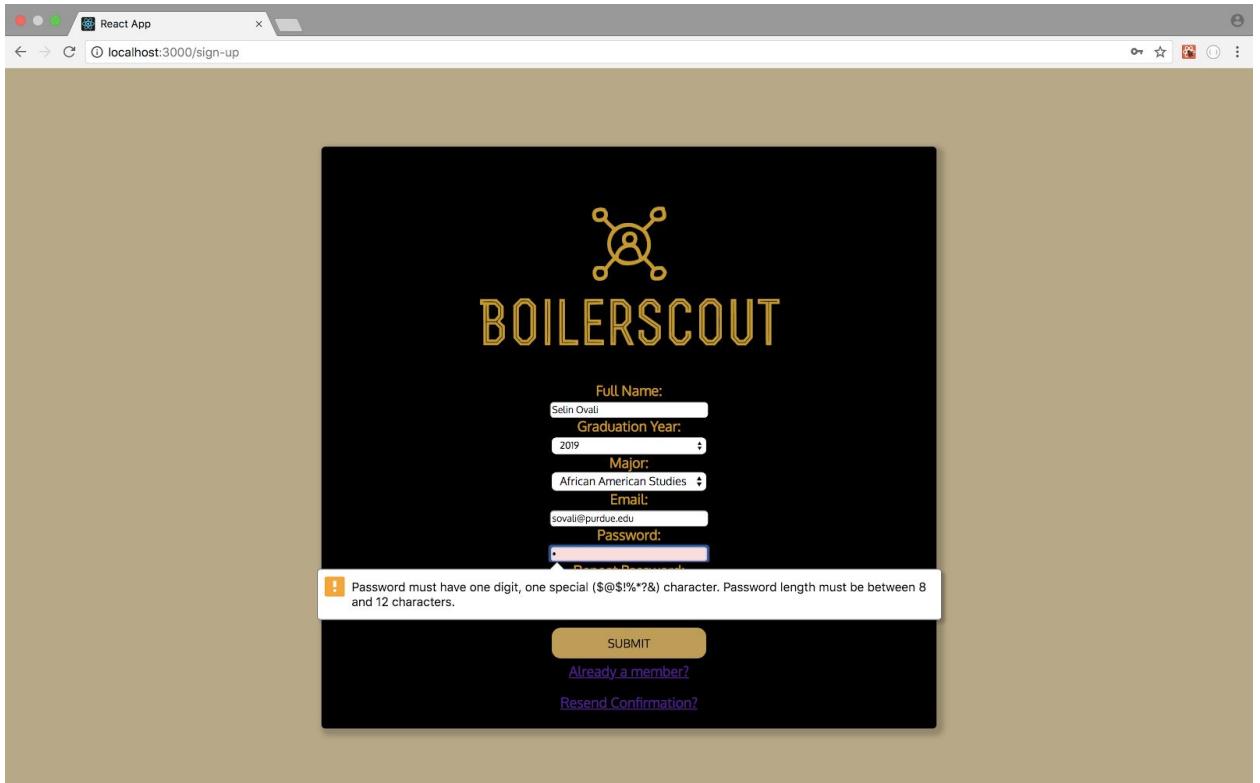
4. Users should enter a Purdue email or get a visual error.



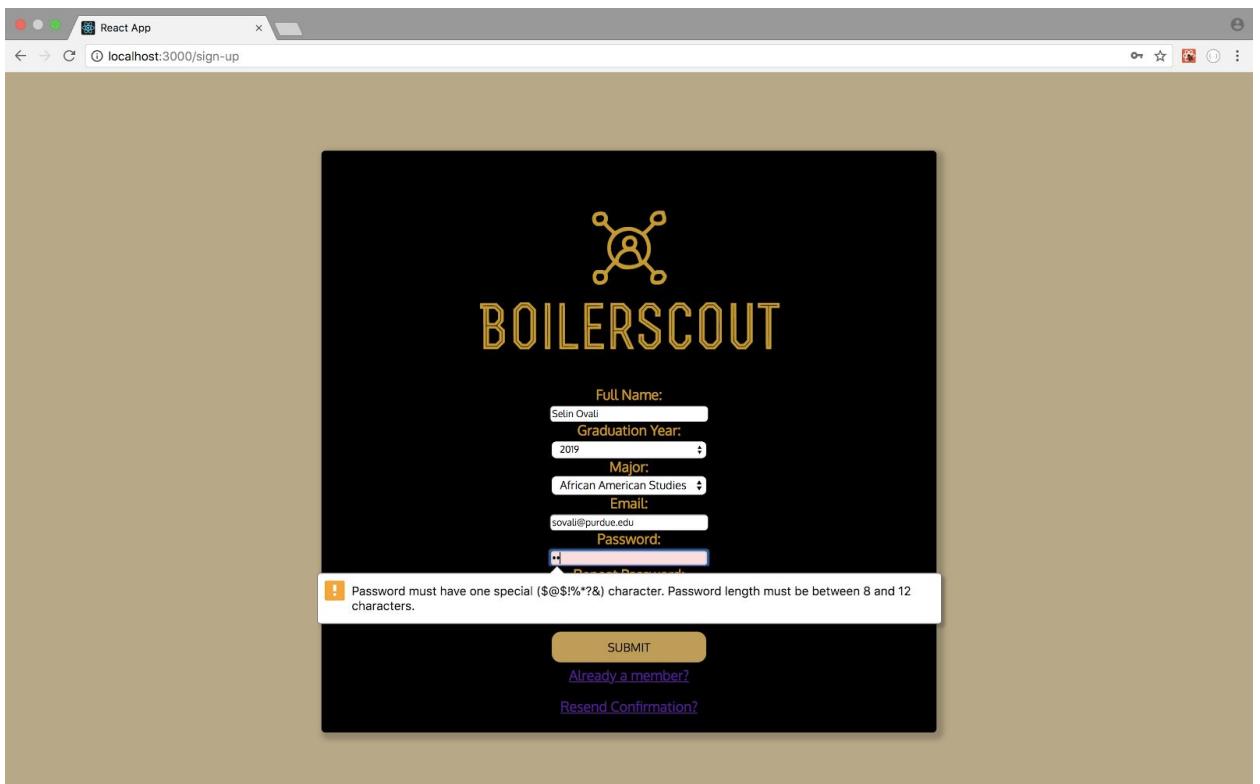
5. Users should enter a password that has one digit, one uppercase, one special character and be 8-12 length. Adding each of the necessary items changes the error message.



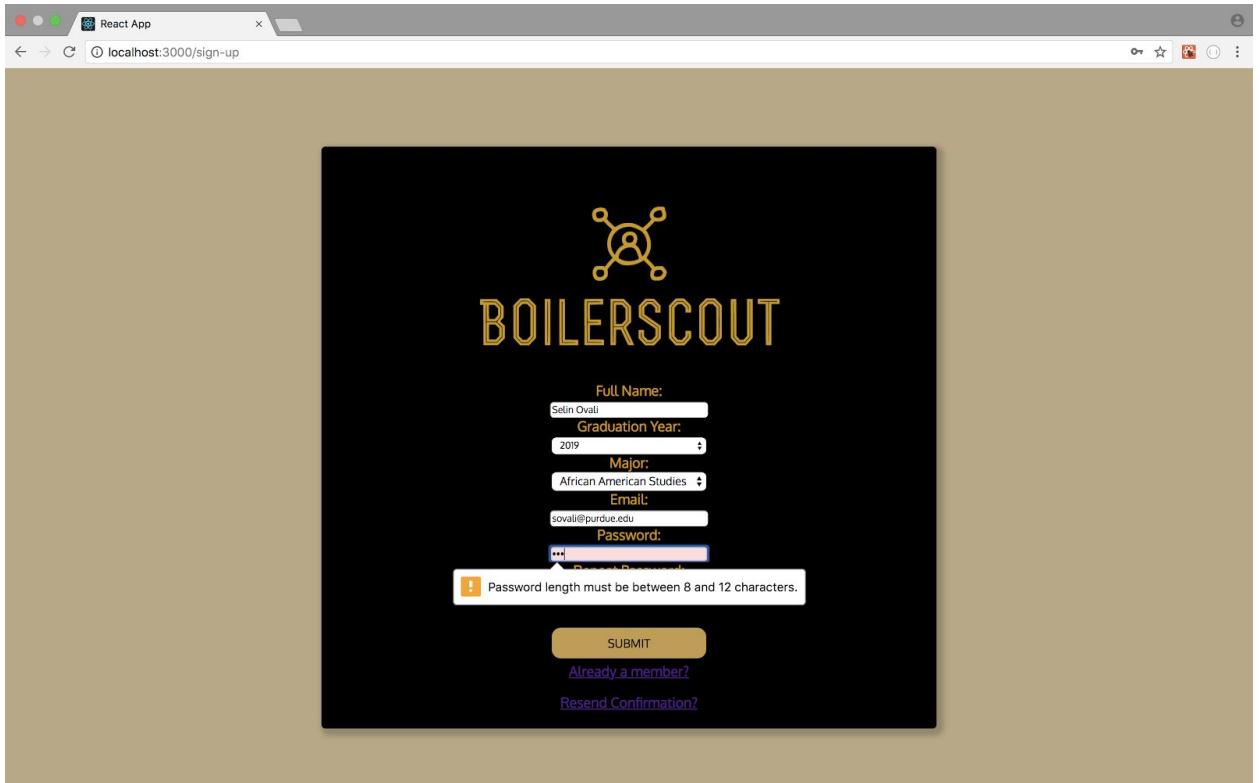
Adding an uppercase character removes that part from the error message.



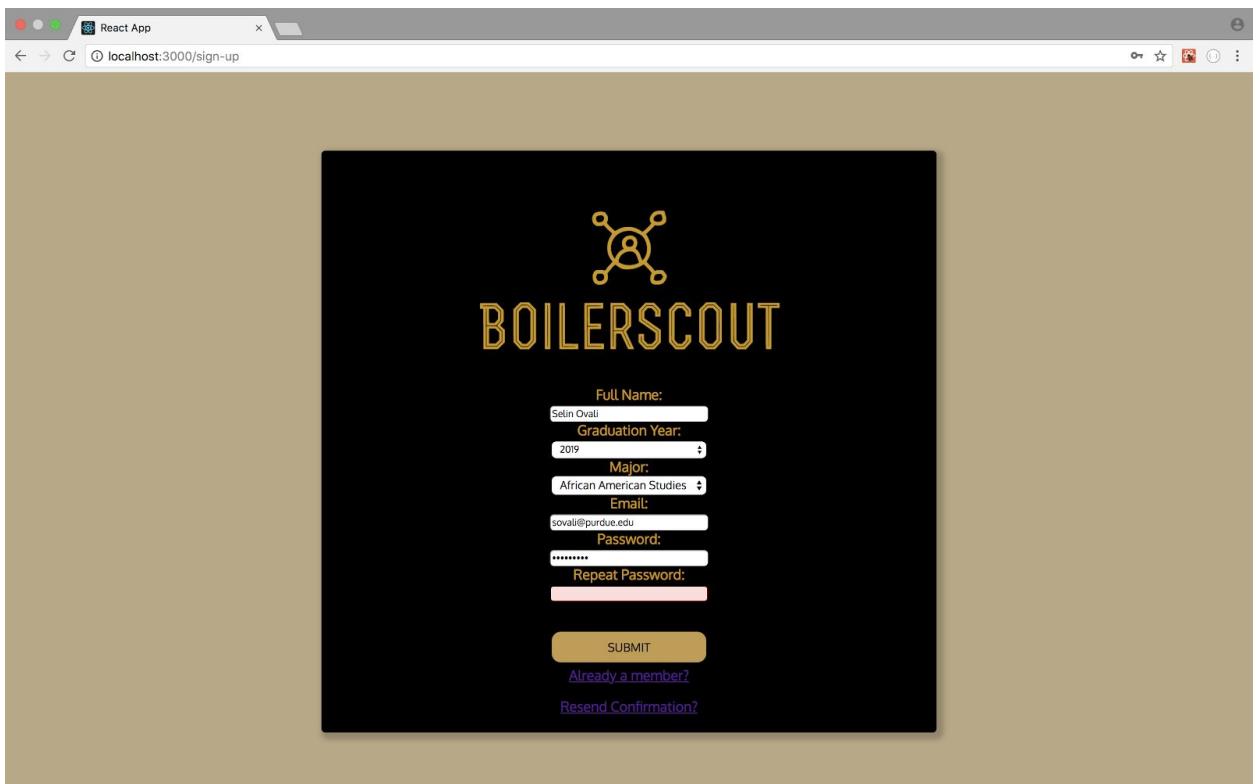
Adding a digit removes that part from the error message.



Adding a special character removes that part from the error message.



Having the right length removes error message.



6. Users should enter a repeat password that matches password

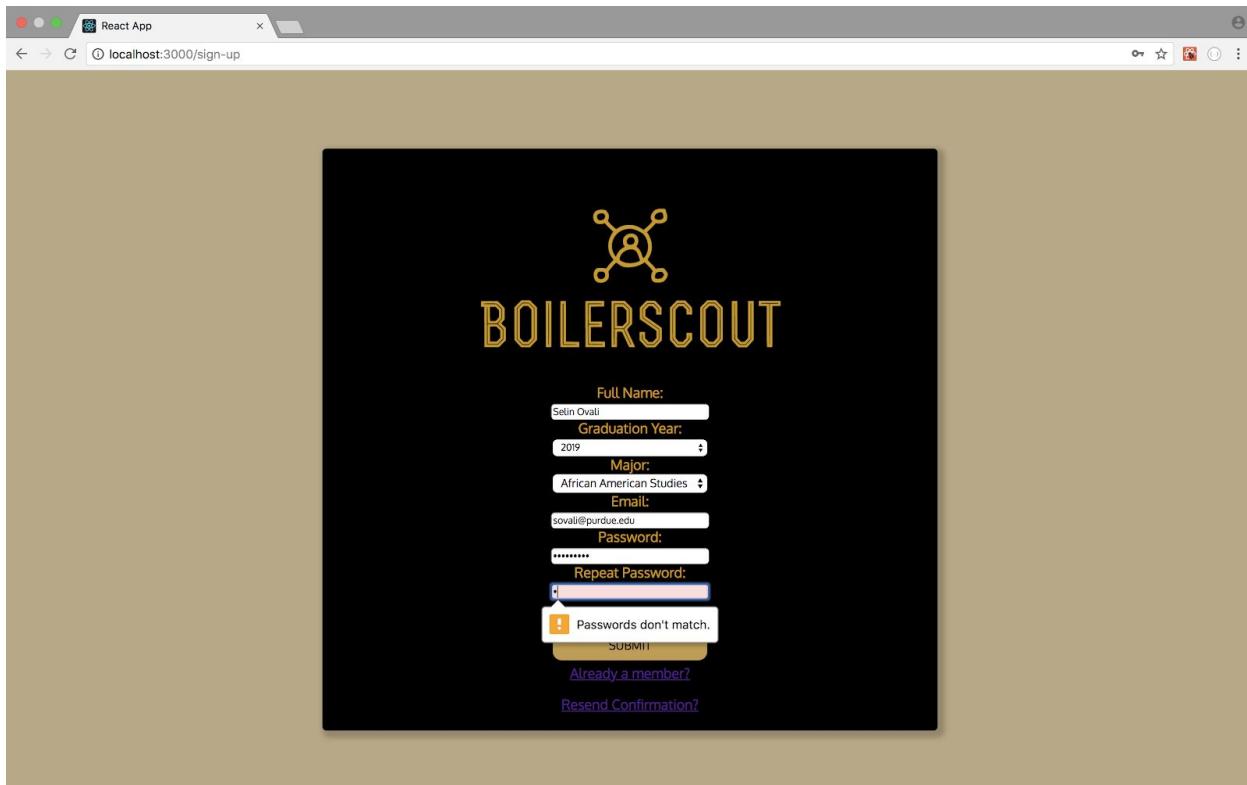
When there's no user input in repeat password, an appropriate error message is displayed.

The screenshot shows a web browser window titled "React App" with the URL "localhost:3000/sign-up". The page has a dark background with gold text and icons. At the top center is a logo consisting of three circles connected by lines, resembling a molecular structure or a gear. Below it is the word "BOILERSCOUT" in large, bold, gold capital letters. The form fields are as follows:

- Full Name: Selin Ovali
- Graduation Year: 2019
- Major: African American Studies
- Email: sovali@purdue.edu
- Password: (represented by five asterisks)
- Repeat Password: (represented by a blue placeholder bar)

A yellow warning box is positioned between the "Password" and "Repeat Password" fields, containing the text "Please enter your password again." Below the form is a brown "SUBMIT" button. At the bottom of the page, there are two links: "Already a member?" and "Resend Confirmation?".

If there's input but it doesn't match the password, another message is displayed.



When repeat password matches password, errors are removed and user is able to sign up.

