SW Engineering CSC 648/848 Fall 2020
SFSU Market

Team # 03

Team Lead: Nicholas Green ngreen2@mail.sfsu.edu
Back end Lead: Steven McHenry
Front end Lead: Ahmad Afghan
Github Master: Lauren Luke
Waqas Hassan

https://648-group3.xyz/

Due Date: December 18, 2020

Product Summary

Name: SFSU Market

List of P1 Functions:

Unregistered users:

- Unregistered users shall be able to create accounts on the application.
- All users shall be able to view listings.
- All users shall be able to search for listings by title and category.
- On load, the site shall display the most recent listings made on the application.

Registered users:

- Registered users shall be able to log in to their preexisting accounts.
- Registered users who have logged in shall be able to log out through the application.
- Registered, logged-in users shall be able to list an item or service for sale on this application and attach images to that listing.
- Registered, logged-in users shall be able to message other users privately.
- Registered users shall have a dashboard page to view their listings and messages.

ADMIN:

- Site admins shall be required to verify all listings before they are made public.
- Site admins shall be able to delete or lock listings made by registered users.
- Site admins shall be able to ban accounts from creating listings or messaging users.

Link: https://648-group3.xyz/

Previous milestones on following pages up to page 49

SW Engineering CSC 648/848 Fall 2020

SFSU Market

Team # 03

Team Lead: Nicholas Green ngreen2@mail.sfsu.edu
Back end Lead: Steven McHenry
Front end Lead: Ahmad Afghan
Github Master: Lauren Luke
Waqas Hassan

“Milestone 1”

Due Date: September 24, 2020

Table of Contents

1. Executive Summary

The goal of this product is to streamline the currently spread-out method by which students, faculty, and staff exchange items and services at SFSU. As of now, there is no real hub to do this, with transactions being spread across many platforms. For instance, a student might post a used textbook on a Facebook group, a tutoring offer on a bulletin board, or an old couch they do not need anymore on Craigslist. Our goal is to consolidate all of this by creating a versatile platform on which all services and goods that could be exchanged in this way can be advertised and sold.

Users will also be able to tag their listings by the classes their items are used in, and search using these tags. This feature is in many places on the internet, and is very useful when it is used. This project will be no exception to that. For example, if a student is buying several textbooks for a class, this prevents students from having to look for each book individually. If they are in a class with five textbooks required, for example, they can simply search using that class name and see all listings that have been marked as related to the class..

Our team consists of five undergraduate students in the computer science department. Several of us have completed general education requirements at SFSU, and all of us have had to get books for a class here. Since this is true, we are all aware of the challenges in this process and have been able to develop a manner in which we would like to address them. The team also contains people who have switched majors from a BA to a BS and therefore are more able to approach this problem from both of these angles.

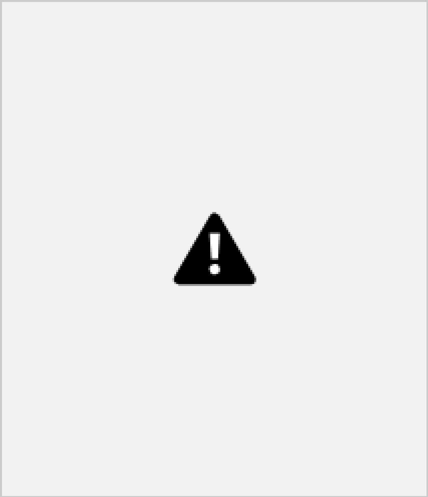
2. Personae and Main Use Cases

Persona: **Jennifer** § First year, majoring in Biology § Lives on-campus in first year dorms § Has one roommate § Originally from SoCal, goes back home during holiday breaks § Has never bought textbooks online before	
After the first day of school, Jennifer knows that she needs a textbook and iClicker for her Bio 100 class. The professor recommends a website to buy a brand new version of both items but it is more expensive than what Jennifer can afford. Professor urges everyone to have the materials in class by next week.	*"I am trying to not take loans this year and have been saving up all summer to pay for my education and housing. I know these materials are important but they are also really expensive. Thankfully, I was able to find used versions from past students for a much reasonable price."*

Persona: **Edgar** § Undergraduate full-time student § He is a transfer student § Lives in San Jose § On campus 4 days a week § Familiar with craigslist and similar sites. § Hates waiting	

It is the beginning of the semester and Edgar needs to buy a textbook for his physics class. He needs it in the next few days so that he can start working on his first assignment that is due next week.	*“I added the class late, so I missed the first few lectures. I did not know that the first assignment is due next week. I found this website that had lots of SFSU students selling their old books, and I could get it on the same day.”*

Persona: **Monica** § Undergraduate part-time student § English major § Works a full-time job § Great with time management, organization and communication § Works mainly remote due to constant travel	
Monica is enrolled in two online classes this semester while recently receiving a promotion at her full-time job. She travels often so she is not able to visit the school to buy her textbooks. Electronic books work best for her but she is also on a budget.	*“I am constantly on the go so I need my textbooks to be easily accessible. Electronic books that I can easily download make my life so much easier. I heard about this website where students sell their links to their online textbooks. This would help me so much to get my books right away and keep me within my budget.”*

Persona: **Samuel** § Philosophy professor at SFSU § On campus three days a week § Has used online learning resources before, but is unfamiliar with e-commerce § Dislikes confusion, both in himself and among his students	
Samuel assigns five books for an undergraduate philosophy course. He understands that this can be a complicated, or at least tedious, process. He wants to find a way to streamline this process for his students.	*"Philosophy has a lot of reading in it, of course, and buying textbooks is difficult to do without breaking the bank. I would like to help my students do this more easily, but I am not willing to change the reading list."*

Persona: **Sasha** § Grad Student, SFSU § Lives off campus in her own house § Originally from France § Technologically minded and strongly involved with student affairs	
She is interested in having a portal for students to easily access school materials and books that is organized by students. She has a deep love for student involvement and is an avid online forum user.	*"I would love for a place where students and teachers can go so they can get affordable books and supplies for their classes. It would be exciting to have it be student run and dynamic enough for many different needs."*

Main Use Cases:

Inexpensive Resources-Jennifer is worried she is going to spend too much on one textbook and iClicker. Her roommate told her about the website where SF State students can sell and buy used school materials. She uses the search toolbar to look up the clicker and her textbook. She finds her specific matches and reaches out to the seller, after which she is prompted to register as a user first. They then decide a meet up place and time through the private chat between Jennifer and the seller. This is very convenient for Jennifer because she lives on campus and ended up finding someone with her book and iClicker that also lives on campus. Now Jennifer is ready for her semester as a first year Bio student.

Quick Resources-Edgar needs a Physics textbook very soon for completing his first assignment that is due next week. He found out that some students are selling their old textbooks on a website. He found the exact textbook and contacted the seller which prompted him to register first before purchasing the textbook. He is very happy and shall get his textbook in the next few days.

Online Textbooks-Monica works eight hour days and travels from city to city for her job so school for her is remote. At the end of her day she needs her electronic textbooks to be easily accessible and ready for download. Monica did some research and found a website through SFSU where students can buy and sell a variety of items. She browsed the website and was able to pick out what she needed very easily by searching by title and tags. Once her browsing was concluded, she was able to message the seller and purchase the link to her online textbook receiving it in minutes. Monica is now ready for the semester and knows at any time she can easily purchase anything she needs from SFSU Market.

Reading List-Samuel assigns several textbooks and recommends SFSU Market to his students. Several users on SFSU Market are selling copies of these books. They can all be found by searching for the tag corresponding to the class name. His students are all able to find the books required for the class within a week using this application and the tag search feature.

Site Administration-It is after a long day of classes and Sasha hops on her admin portal for the website she helped create. She starts by scanning through the requested postings and approved most while denying some that do not fit the criteria of the site. She then notices that a post that displayed a student's address made it past her previous scans. Sasha deletes that post and then goes on to delete the user that made that post as it is not the first time they have posted sensitive information about other students.

3. Main Entities

1. User

a. Three types of possible users
   i. UNREGISTERED
   ii. REGISTERED
   iii. ADMIN

b. Registration Record
   i. Name
   ii. Email (used to verify SFSU member)
   iii. Password (Encrypted)
   iv. Avatar (Possibly upload their own or select from various possibilities)

2. Permission
   a. Contains the list of possible permissions a user can have.
      i. REGISTERED
      ii. ADMIN
3. Message
   a. The messages for each user that they can access.
   b. These messages are used for communication within the app so that users can ask questions pertaining to listings and set up meetings for exchanges
4. Listing
   a. Will contain the following elements
      i. Key: Randomly generated ID
      ii. Title: String
      iii. Description: String
      iv. Category: List of Categories
      v. Price: Int
      vi. Images: List of Images
5. Category
   a. The category of the item for sorting purposes.
6. Image
   a. This entity will contain the following elements
      i. Key: Randomly generated ID
      ii. Name: String
      iii. Size: Int
      iv. Url: String
7. Location
   a. All available safe locations that they can choose from to meet at.
8. Class
   a. Contains all class names and numbers. Used for sorting as well.
9. Book
   a. The book name and ISP for sorting and searching.
10. Tag

a. All possible Tags
    i. List of Classes
    ii. List of Books
    iii. List of Categories

4. Initial List of Functional Requirements

1. On load, the site shall display the most recent posts made on the application.
2. Unregistered users shall be able to create accounts on the application.
3. Unregistered users shall not be prompted to register until the final stages of performing an action on the website. Unregistered users shall be able to begin an action that requires the user to be logged in to an account. They will be unable to finish these actions without registering.
4. Registered users shall be able to log in to their preexisting accounts.
5. Registered users who have logged in shall be able to log out through the application.
6. Registered, logged-in users shall be able to post an item or service for sale on this application and attach images to that post.
7. Registered, logged-in users shall be able to add tags to their posts.
8. Registered, logged-in users shall be able to make public comments on posts.
9. Registered, logged-in users shall be able to message other users privately.
10. Registered, logged-in users shall be able to rate other users they have made transactions with.
11. Registered, logged-in users shall be able to mark an item as sold once sold out.
12. Site admins shall maintain a list of locations that are designated as safe for exchange, which will be publicly available for both registered and non-registered users to access.
13. Site admins shall be required to verify all posts before those posts are made public.
14. Site admins shall be able to delete or lock posts made by registered users.
15. Site admins shall be able to delete comments made on posts.
16. All users shall be able to view posts.
17. All users shall be able to search for posts by title and other criteria such as location, price, and tags.

5. List of Non-Functional Requirements

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team, but all tools and servers have to be approved by class CTO).
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of all major browsers: Mozilla, Safari, Chrome.
3. Selected application functions must render well on mobile devices.

4. Data shall be stored in the team’s chosen database technology on the team’s deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time.
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
7. The language used shall be English.
8. Application shall be very easy to use and intuitive.
9. Google analytics shall be added.
10. No email clients shall be allowed.
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
12. Site security: basic best practices shall be applied (as covered in the class).
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.
14. The website shall prominently display the following exact text on all pages "SFSU Fulda Software Engineering Project CSC 648-848, Fall 2018. For Demonstration Only” at the top of the WWW page. (Important so as to not confuse this with a real application).

6. Competitive analysis

Features	**Craigslist**	**Amazon**	**eBay**	**SFSU Bookstore**	**SFSU Market**
Direct messaging between users	✓✓	✓	✓	✘	✓
Filter search results (price, location, tags)	✓	✓✓	✓	✘	✓✓
Price comparison to new book	✘	✓	✘	✓	✓
Designated safe locations for exchanges	✘	✘	✘	✓	✓
Most recent posts displayed on homepage	✘	✘	✘	✘	✓✓
Teacher-verified reading list for classes	✘	✘	✘	✘	✓✓

✓ = good
✘ = not available
✓✓ = excellent

SFSU Market allows students to easily and safely sell and buy school related materials with each other in a convenient manner. Students from the past can sell their class materials like textbooks, books, iClickers, etc, on our website to other students at SFSU. We have many similar features of Craigslist, Amazon, eBay, and SFSU Bookstore on our website such as private messaging between the buyer and seller and searching by specific filters like price and locations. One benefit that we provide that is not offered by our competitors are designated safe locations for exchanges, which allows our users to feel safe during their meet up with the buyer/seller and relieves the likelihood of danger. Another feature that is only on our website is that the most recent posts get shown on the homepage which takes away the disadvantage of paying for advertisements to get more views on your posts. On our website we also provide teacher-verified readings lists, allowing students to view what materials they need for their respective classes. Although the SFSU Market is a safe place to buy textbooks, you cannot always sell at the bookstore and they are not the cheapest price you can get. On the website, you can compare prices and messages whichever one is available and convenient for the user.

7. High-Level System Architecture

1). Server Host: Amazon Web Services (AWS) 1vCPU 1GB RAM
2). Operating System: Ubuntu
3). Database: MySQL server
4). Web Server:
- Nodejs

5). Server-Side Language:
- Javascript (v 1.8. 5)

6). Additional Technologies:
- Bootstrap 4
- Express (Back-end framework)
- ReactJs (Front-end framework)
- IDE (VScode)

7). Supported Browser:
- Google Chrome and Mozilla Firefox (latest 2 versions for each)

8. Team and Roles

Nick Green- Team Lead
Lauren Luke- Github Master, Document Master
Steven McHenry- Back end Lead
Ahmad Afghan-Front end Lead
Waqas Hassan- Team Member Front end

9. Checklist

· So far all team members are engaged and attending ZOOM sessions when required **DONE/OK**
· Team found a time slot to meet outside of the class **DONE/OK**
· Back end, Front end leads and Github master chosen **DONE/OK**

· Team decided and agreed together on using the listed SW tools and deployment server **DONE/OK**

· Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing **DONE/OK**

· Team lead ensured that all team members read the final M1 and agree/understand it before submission **DONE/OK**

· Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) **DONE/OK**

SW Engineering CSC 648/848 Fall 2020
SFSU Market

Team # 03

Team Lead: Nicholas Green ngreen2@mail.sfsu.edu
Back end Lead: Steven McHenry
Front end Lead: Ahmad Afghan
Github Master: Lauren Luke
Waqas Hassan

"Milestone 2"

Due Date: October 15, 2020

Table of Contents

1. Executive Summary

The goal of SFSU Market is to streamline the currently spread-out method by which students, faculty, and staff exchange items and services at SFSU. As of now, there is no real hub to do this, with transactions being spread across many platforms. For instance, a student might post a used textbook on a Facebook group, a tutoring offer on a bulletin board, or an old couch they do not need anymore on Craigslist. Our goal is to consolidate all of this by creating a versatile platform on which all services and goods that could be exchanged in this way can be advertised and sold.

Users will also be able to place their listings in various categories and use the search function to find listings that are in the categories that are relevant to them. This is a common feature on many platforms, and here it will be used in a similar way to those other platforms, such as Amazon or eBay. The search function will also find listings that resemble the search query, so general information, like a class title or a type of service, will be searchable based on the listing title as well.

Our team consists of five undergraduate students in the computer science department. Several of us have completed general education requirements at SFSU, and all of us have had to get books for a class here. Since this is true, we are all aware of the challenges in this process and have been able to develop a manner in which we would like to address them. The team also contains people who have switched majors from a BA to a BS and therefore are more able to approach this problem from both of these angles.

2. List of Main Data Items and Entities

1. User
    a. Three types of possible users (stored as bool)
        i. UNREGISTERED
        ii. REGISTERED
        iii. ADMIN
    b. Registration Record
        i. Name
        ii. Email (used to verify SFSU member)
        iii. Password (Encrypted)
        iv. Avatar (Possibly upload their own or select from various possibilities)
2. Chat
    a. A collection of messages that can be accessed by either of the two involved users.
    b. Contains sent and read times.
3. Message
    a. The messages for each user that they can access.
    b. These messages are used for communication within the app so that users can ask questions pertaining to listings and set up meetings for exchanges.
4. Listing
    a. Will contain the following elements
        i. Key: Randomly generated ID
        ii. Title: String
        iii. Description: String
        iv. Category: List of Categories
        v. Price: Int
        vi. Images: List of Images
        vii. Class: If there is one
        viii. ISP: If there is one
5. Category
    a. This is category that the listings can fall under and will contain:
        i. Key: Randomly generated ID
        ii. Name: String
6. Image
    a. This entity will contain the following elements
        i. Key: Randomly generated ID
        ii. Name: String
        iii. Url: String (location in S3 bucket)
7. Class
    a. Contains all class names and numbers. Used for sorting as well.

3. Functional Requirements - prioritized

(Priority ratings: (P1) → *must have,* (P2) → *desired,* and (P3) → *opportunistic*)
(Types of users: unregistered, registered, and Admins)

P1 PRIORITY

Unregistered users:

- ❖ Unregistered users shall be able to create accounts on the application.

  Creating an account on the application will secure personal data of buyers and sellers and only allow SFSU students or related facility members to buy and sell.
- ❖ All users shall be able to view listings.

  An important function to allow all registered and unregistered users to see listings made by other users at SFSU.
- ❖ All users shall be able to search for listings by title and other criteria such as location and price.

  This function makes it easy for users to find items. They may search an item by the specific name or they may make a broad search and eventually find something that they were looking for.

Registered users:

- ❖ Registered users shall be able to log in to their preexisting accounts.

  Users will not have to create a new account each time they want to buy or sell, allowing them to easily buy and sell in the future.
- ❖ Registered users who have logged in shall be able to log out through the application.

  Users may log in and out as they desire without any complications.
- ❖ Registered, logged-in users shall be able to list an item or service for sale on this application and attach images to that listing.

  This feature is important because it will give a complete review to users who would want to buy the item. It will only allow registered and approved SFSU students to make listings while unregistered users may view the item but cannot purchase until they register.
- ❖ Registered, logged-in users shall be able to message other users privately.

  This function is very important as there is no shipping the item to the user in our application. Therefore, users must contact each other using a private messaging system, allowing them to discuss the price, meet up location and time, and other questions they might have for each other.
- ❖ Registered users shall have a dashboard page to view their listings and messages.

  This is important because it allows the users to see which items are no longer available, so they do not contact the users for an item that is unavailable.

ADMIN:

- Site admins shall be required to verify all listings before they are made public.

  This is to ensure that our application stays professional where there are no listings upsetting or harming anyone and to reduce the chances of any scams that may occur.

- Site admins shall be able to delete or lock listings made by registered users.

  This allows our admins to have control of what is being shared on our application and ensure that the listings made on our application are appropriate to the intended platform.

- Site admins shall be able to delete comments made on listings.

  If there are comments that are intended to hurt other users, then admins have the authority to delete comments that are not appropriate and offer negative feedback.

- Site admins shall maintain a list of locations that are designated as safe for exchange, which will be publicly available for both registered and non-registered users to access.

  One of our main goals with this application is to create an environment that is both safe and convenient for our SFSU users. This function ensures a level of safety when it comes to the transactions between a buyer and seller. It also takes the question of meet up location since we offer designated safe locations.

P2 PRIORITY

Unregistered users:

- Unregistered users shall not be prompted to register until the final stages of performing an action on the website. Unregistered users shall be able to begin an action that requires the user to be logged in to an account. They will be unable to finish these actions without registering.

  This allows the user to stay on the application without having an account and will prompt them to a registration page once they are ready to buy or sell. This creates an opening inviting doors experience for the users.

- On load, the site shall display the most recent listings made on the application.

  This is to grab the users attention as soon as they open the application.

P3 PRIORITY

Registered users:

- Registered, logged-in users shall be able to make public comments on listings.

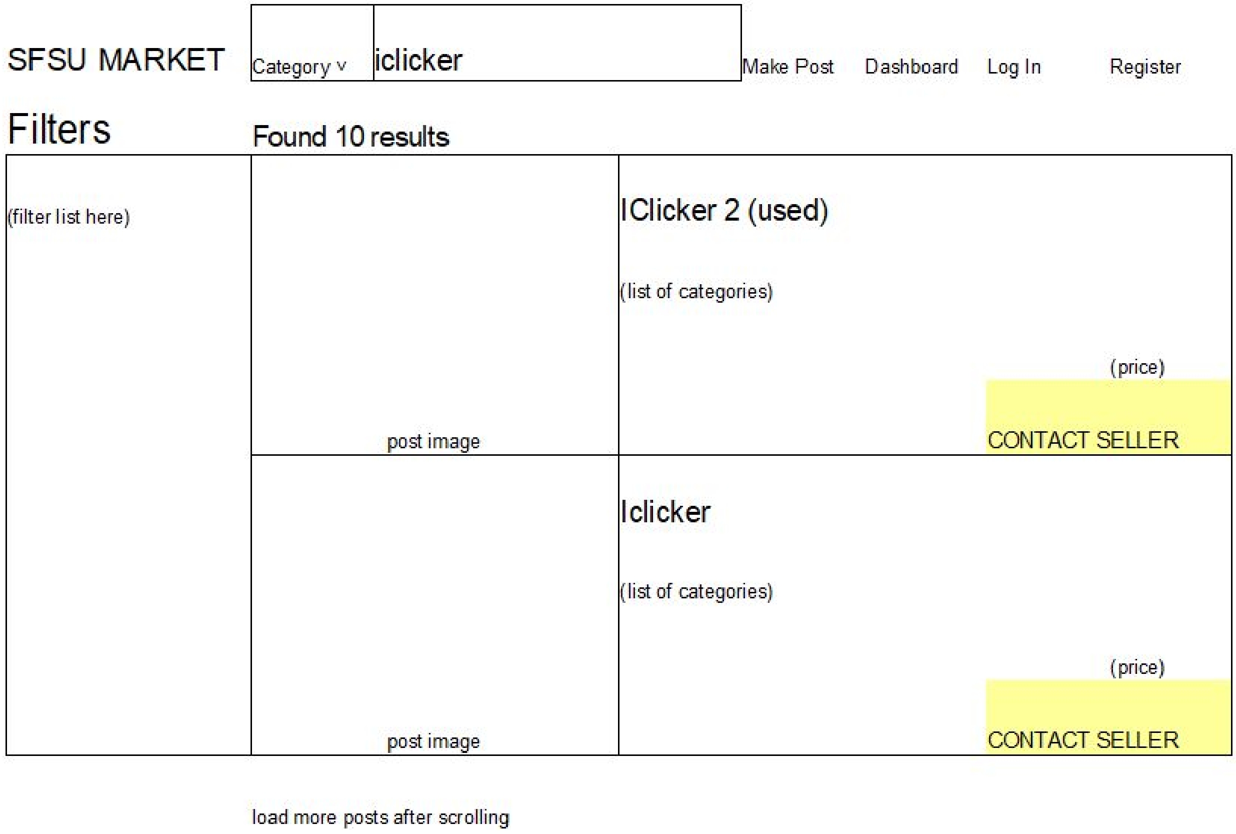
  This function allows the user to post comments as a review of their experience with the seller and their item. This function answers some questions that another user may have for the item and allows users to completely understand the item before they continue with the purchase.

- Registered, logged-in users shall be able to rate other users they have made transactions with.

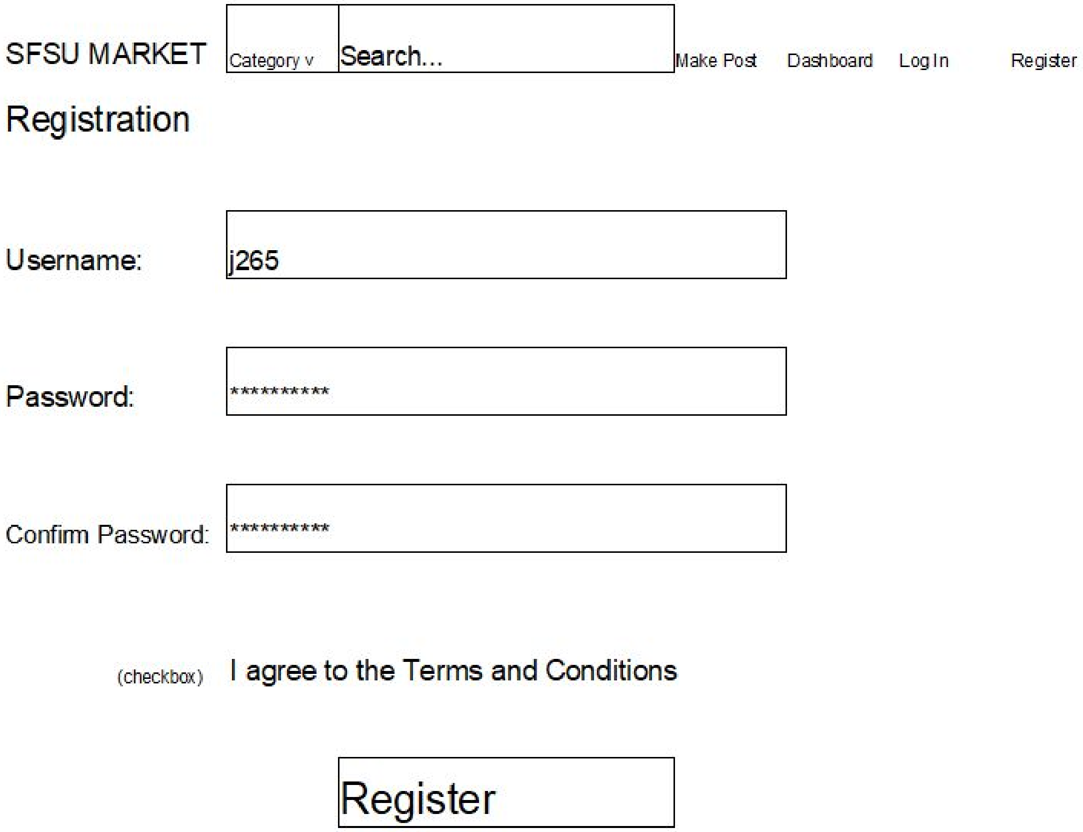
As the function with posting comments on the listings of other users, this function is also public and allows other users to rate their experience with other users they have had business with. Ultimately, this function creates an environment with credibility and decreases the chances of being scammed.

4. UI Mockups and Storyboards

Inexpensive Resources-Jennifer is worried she is going to spend too much on one textbook and iClicker. Her roommate told her about the website where SF State students can sell and buy used school materials. She uses the search toolbar to look up the clicker and her textbook. She finds her specific matches and reaches out to the seller, after which she is prompted to register as a user first. They then decide a meet up place and time through the private chat between Jennifer and the seller. This is very convenient for Jennifer because she lives on campus and ended up finding someone with her book and iClicker that also lives on campus. Now Jennifer is ready for her semester as a first year Bio student.



(click contact seller button)



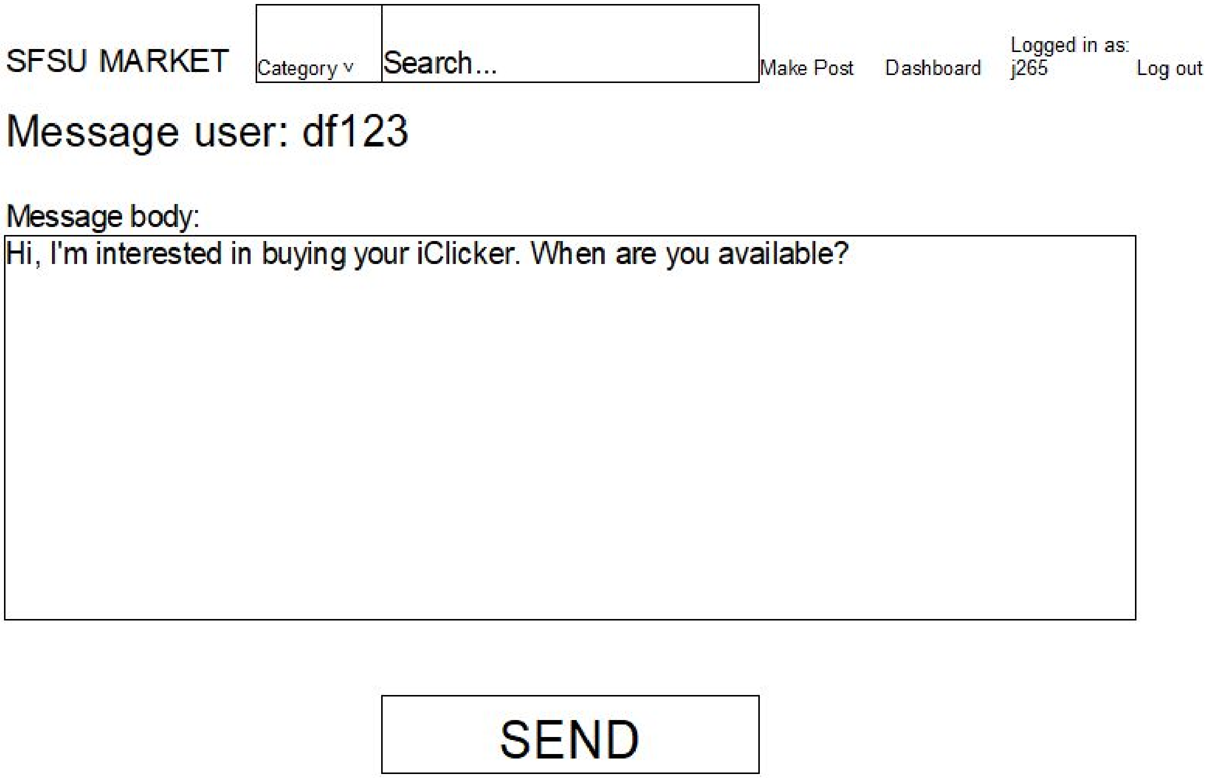
Registration

Username: j265

Password: **********

Confirm Password: **********

(checkbox) I agree to the Terms and Conditions



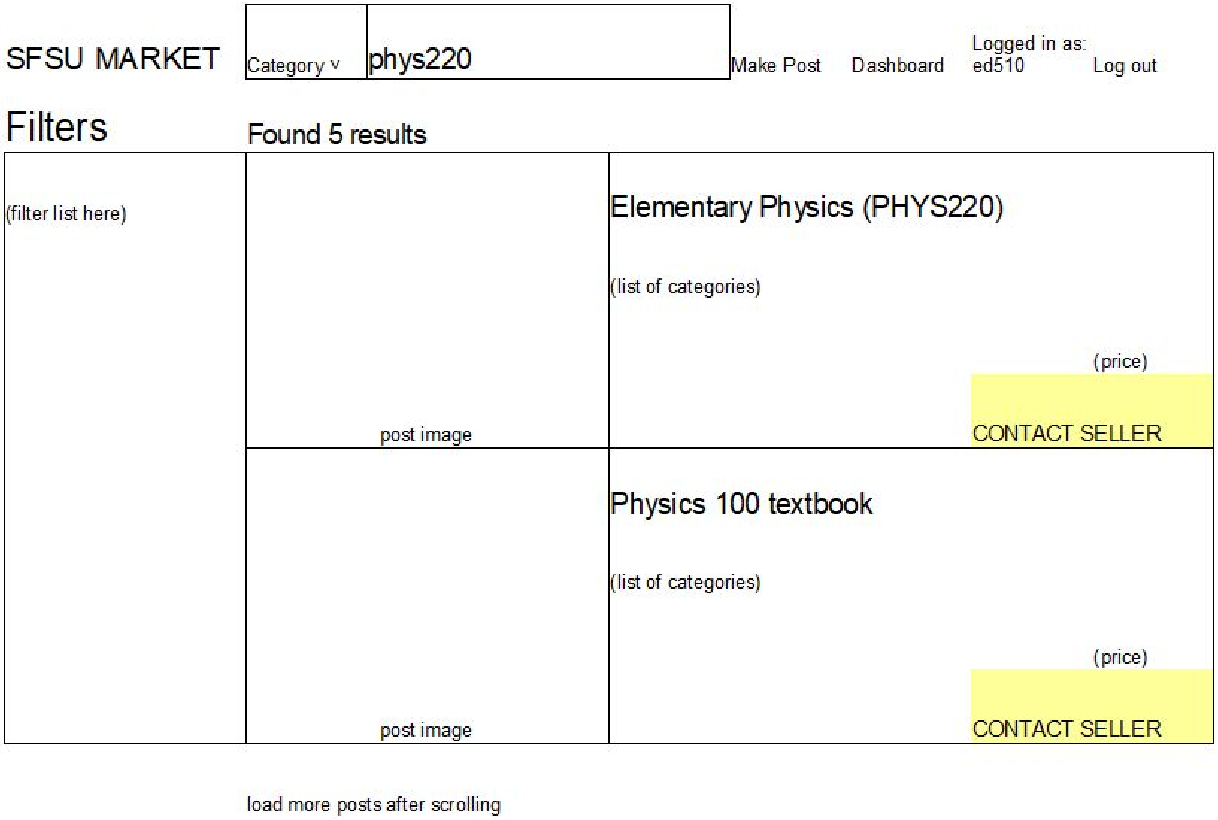
Logged in as: j265

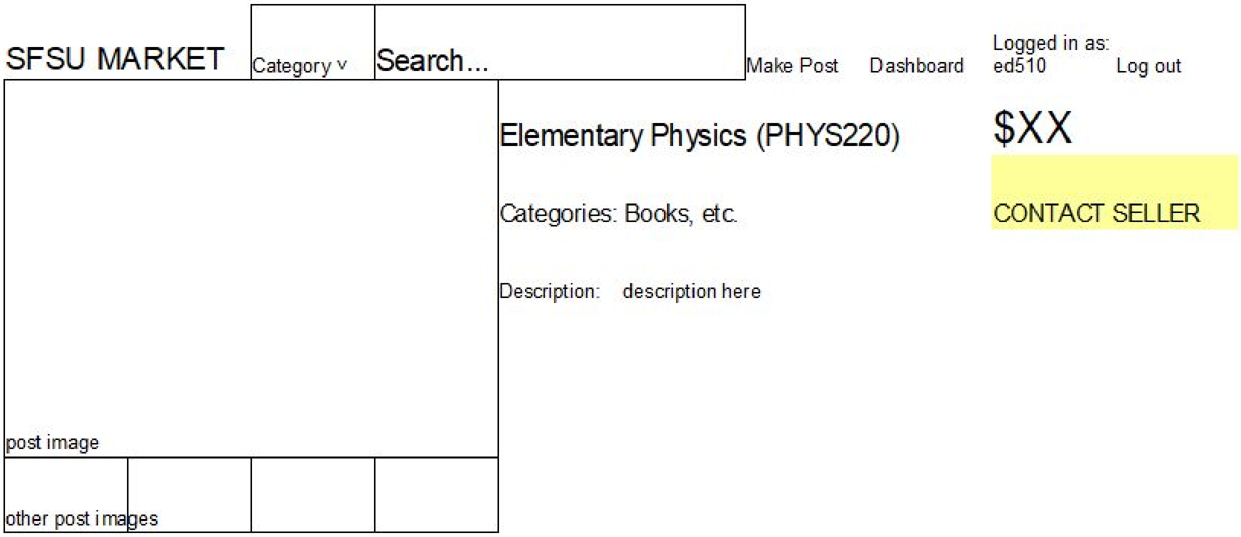
Message user: df123

Message body:

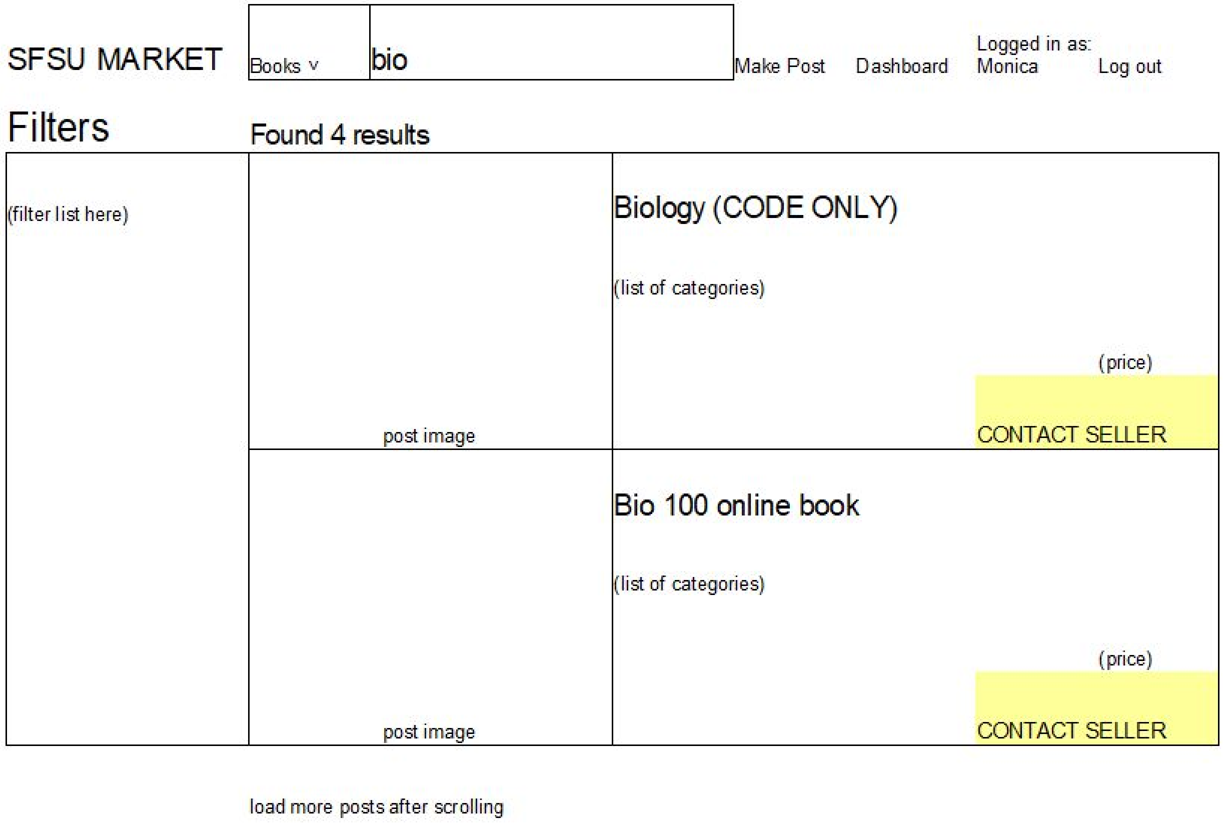
Hi, I'm interested in buying your iClicker. When are you available?

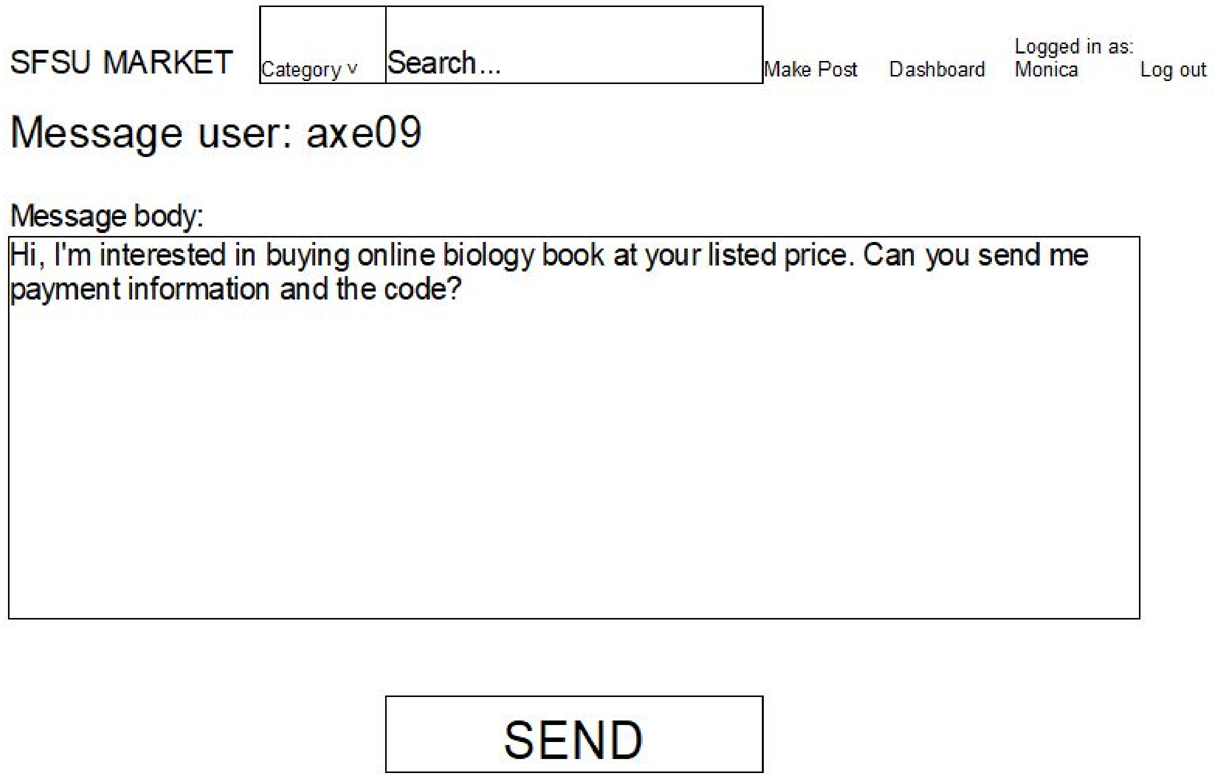
Quick Resources-Edgar needs a Physics textbook very soon for completing his first assignment that is due next week. He found out that some students are selling their old textbooks on a website. He used the search bar on top of the page to search for books he is looking for. He found the exact textbook, and after clicking on the textbook it took him to the description page that he could see the bigger picture of the item, price, category, and item description.





Online Textbooks-Monica works eight hour days and travels from city to city for her job so school for her is remote. At the end of her day she needs her electronic textbooks to be easily accessible and ready for download. Monica did some research and found a website through SFSU where users can buy and sell a variety of items. She browsed the website and was able to pick out what she needed very easily by searching by title. Once her browsing was concluded, she was able to message the seller and purchase the link to her online textbook receiving it in minutes. Monica is now ready for the semester and knows at any time she can easily purchase anything she needs from SFSU Market.

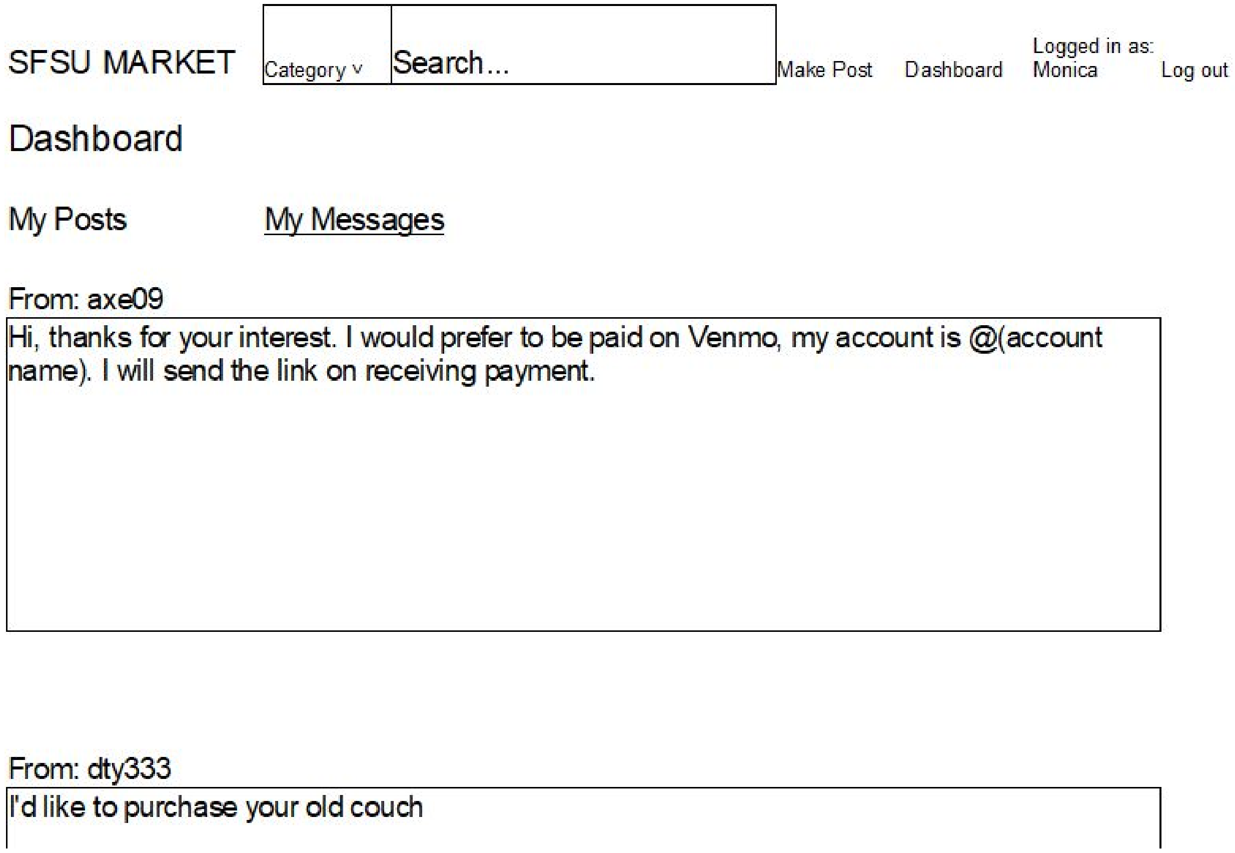




Message user: axe09

Message body:

Hi, I'm interested in buying online biology book at your listed price. Can you send me payment information and the code?



Dashboard

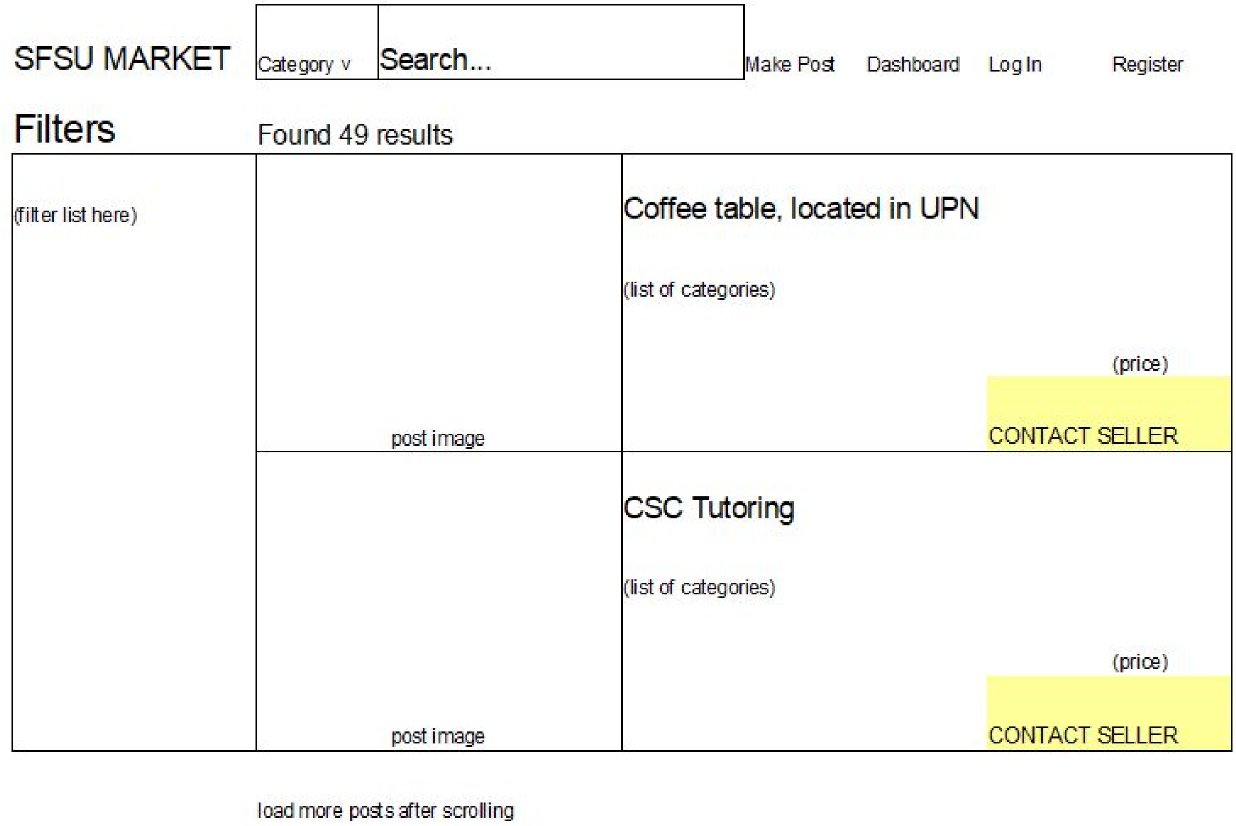
From: axe09

Hi, thanks for your interest. I would prefer to be paid on Venmo, my account is @(account name). I will send the link on receiving payment.

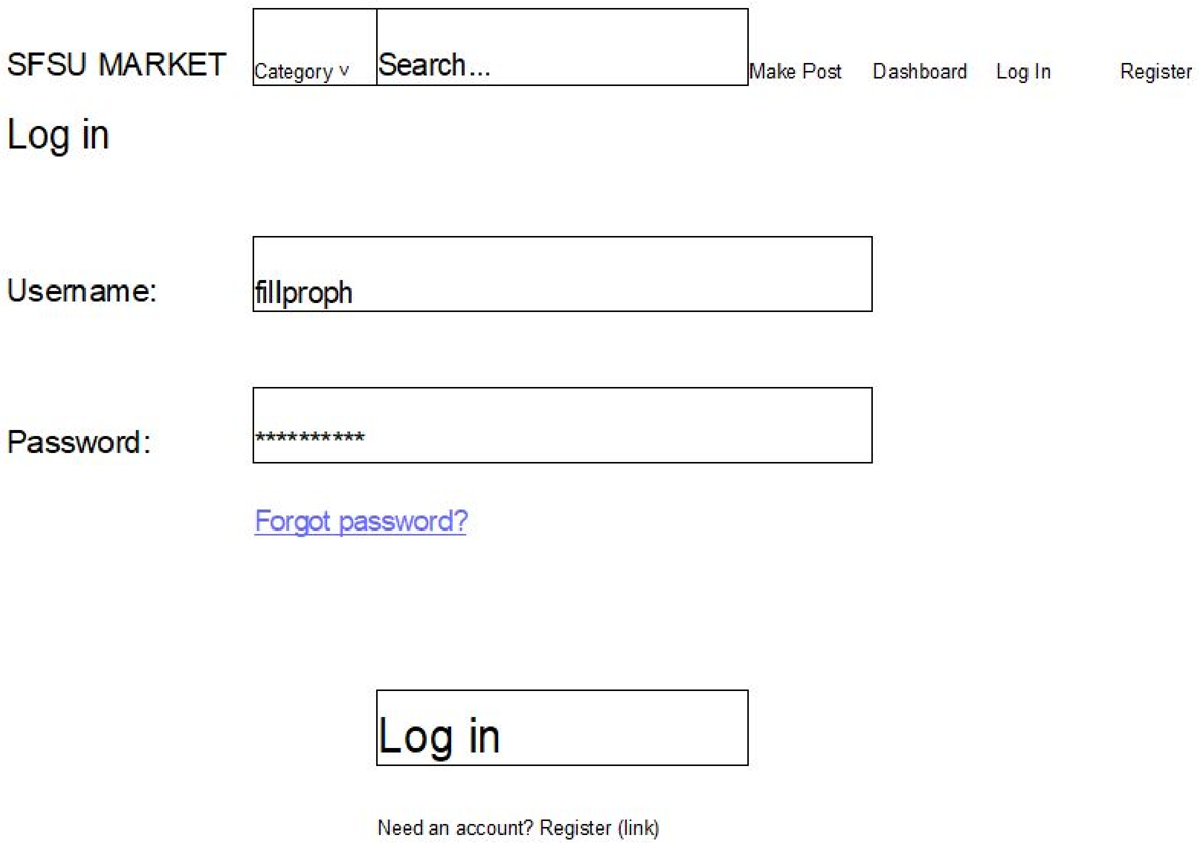
From: dty333

I'd like to purchase your old couch

Reading List-Samuel assigns several textbooks and recommends SFSU Market to his students. Several users on SFSU Market are selling copies of these books. Samuel also owns a few old copies of the books and creates listings for each. His students are all able to find the books required for the class within a week using this application.



(click on Make Post)



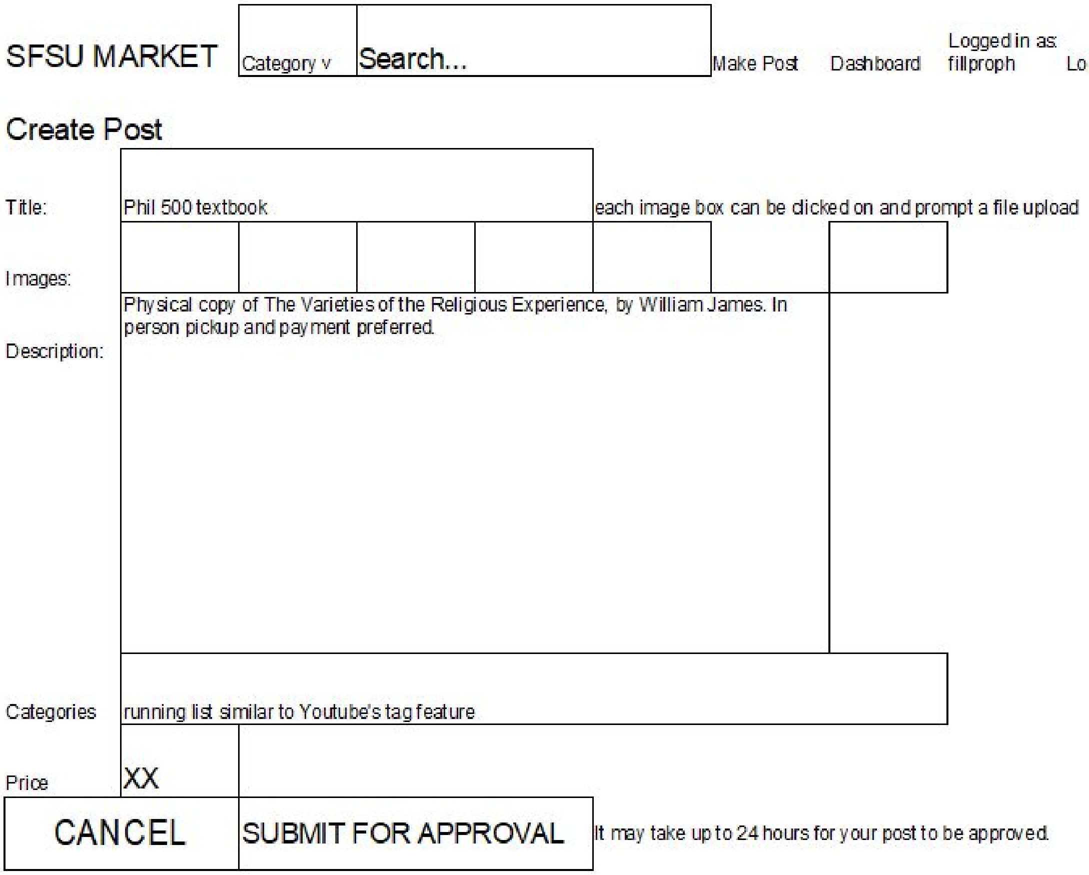
Log in

Username: fillproph

Password: **********

Forgot password?

Need an account? Register (link)



SFSU MARKET | Category v | Search... | Make Post | Dashboard | Logged in as: fillproph | Log out

Create Post

Title: Phil 500 textbook

each image box can be clicked on and prompt a file upload

Images:

Description: Physical copy of The Varieties of the Religious Experience, by William James. In person pickup and payment preferred.

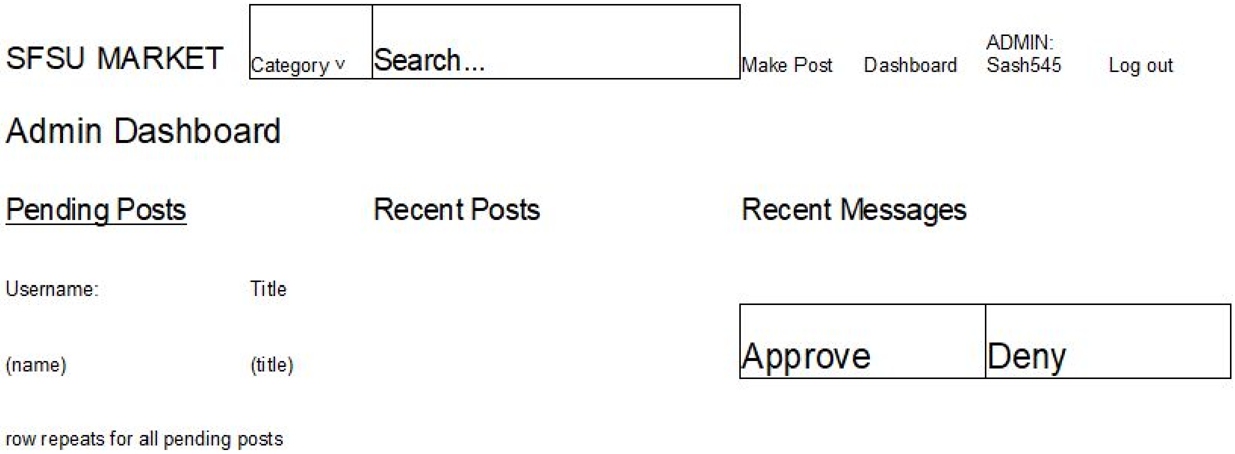
Categories: running list similar to Youtube's tag feature

Price: XX

CANCEL | SUBMIT FOR APPROVAL

It may take up to 24 hours for your post to be approved.

Site Administration-It is after a long day of classes and Sasha hops on her admin portal for the website she helped create. She starts by scanning through the requested listings and approved most while denying some that do not fit the criteria of the site. She then notices a listing that displayed a student's address made it past her previous scans. Sasha deletes that listing and then goes on to delete the user that made that listing as it is not the first time they have posted sensitive information about other students.



SFSU MARKET | Category v | Search... | Make Post | Dashboard | ADMIN: Sash545 | Log out

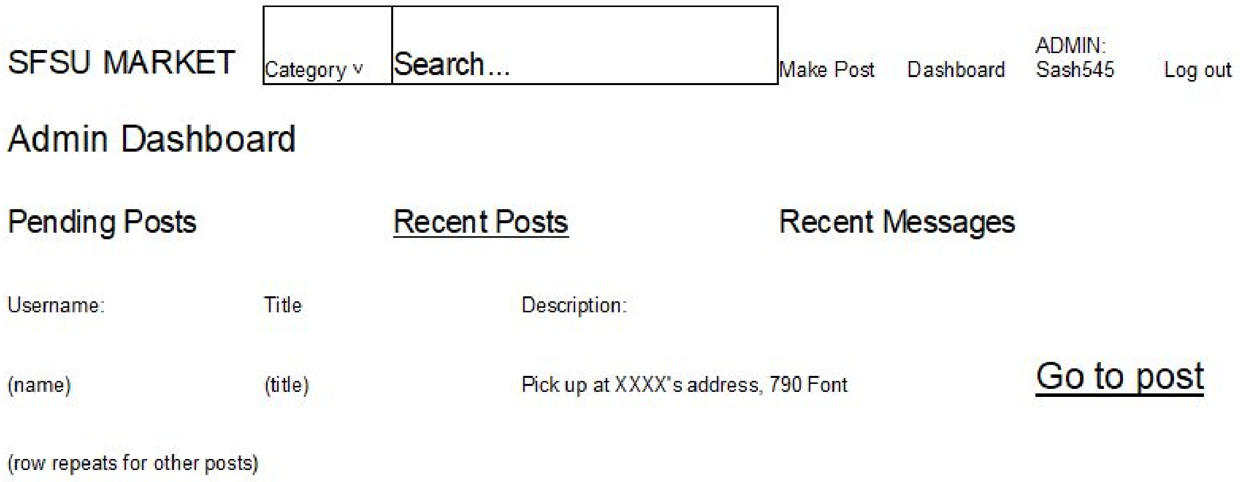
Admin Dashboard

Pending Posts | Recent Posts | Recent Messages

Username: | Title
(name) | (title)

row repeats for all pending posts

Approve | Deny



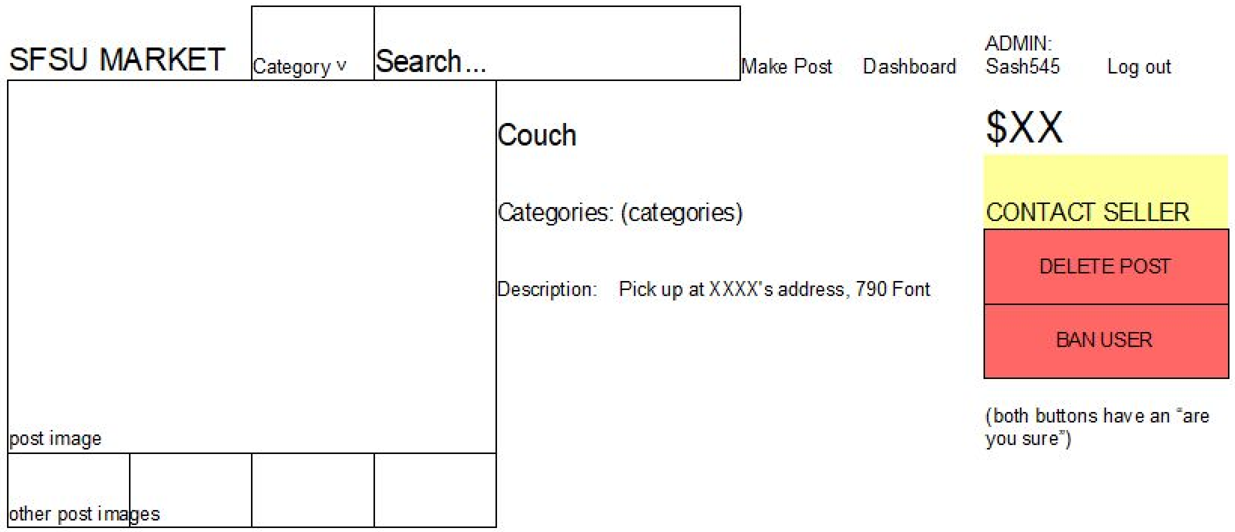
SFSU MARKET | Category v | Search... | Make Post | Dashboard | ADMIN: Sash545 | Log out

Admin Dashboard

Pending Posts | Recent Posts | Recent Messages

Username:	Title	Description:	
(name)	(title)	Pick up at XXXX's address, 790 Font	Go to post

(row repeats for other posts)



SFSU MARKET | Category v | Search... | Make Post | Dashboard | ADMIN: Sash545 | Log out

Couch

Categories: (categories)

Description: Pick up at XXXX's address, 790 Font

$XX

CONTACT SELLER

DELETE POST

BAN USER

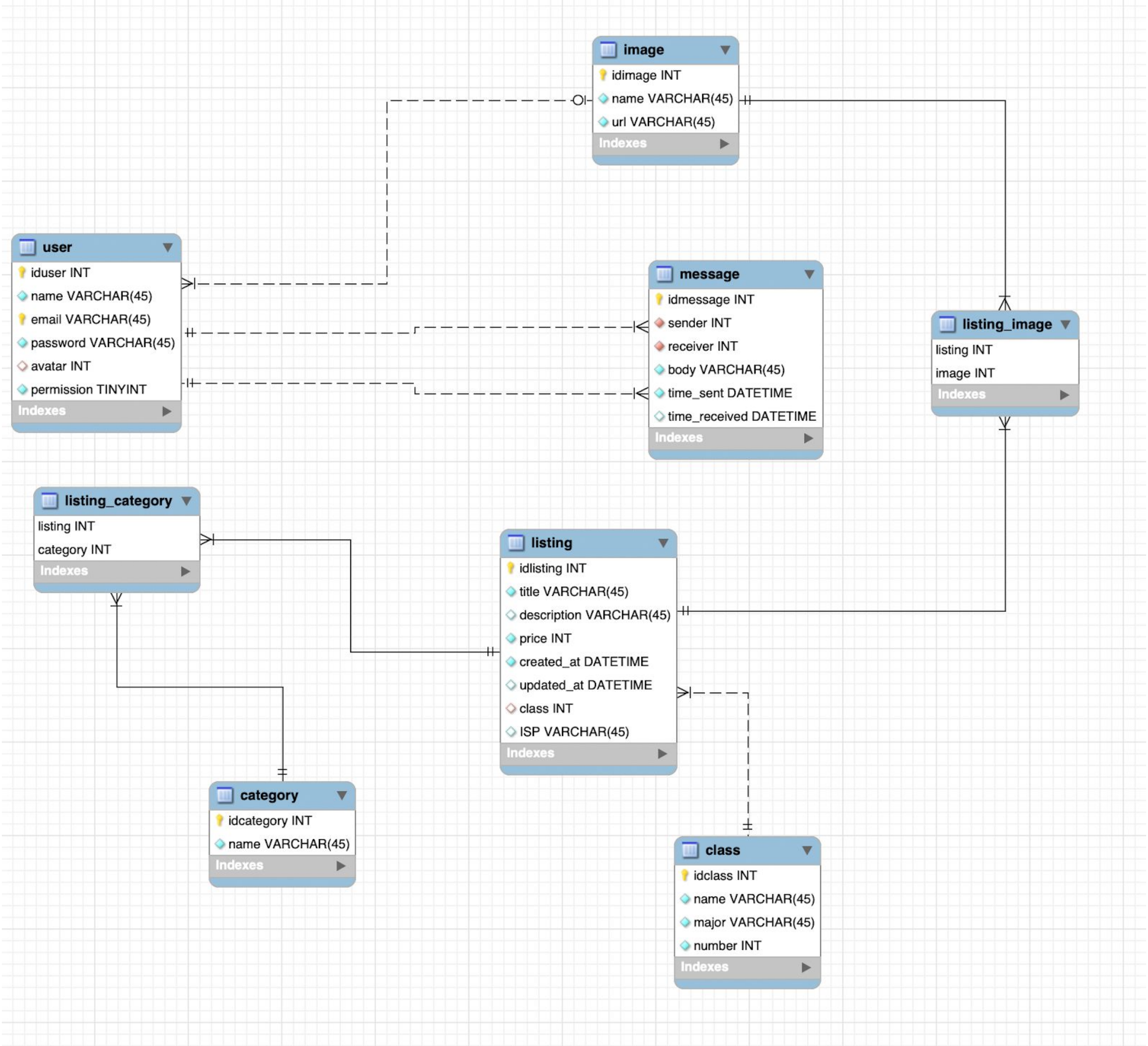
(both buttons have an "are you sure")

post image

other post images

5. High level Architecture, Database Organization summary only

- DB organization:



- The white diamond indicates that they can be null

- Media storage:

- The current plan is to use the AWS javascript sdk functions to allow users to upload images to an s3 bucket and have it auto generate a url that will only be accessible by our app domain.
- Then that URL will be saved within the image -> url entity.

- Search/filter architecture and implementation:

- Will definitely be implementing the SQL %like for “fuzzy” searching terms.

- There will also be filters that the user can categorize their searches by. For instance, if the user just wants to find books, then we query the db to return all instances of a listing_category that includes a book as its category foreign key. It will then return all the listings that are attached to the initial joint table query.
- The user will also have an option to filter their results by price and date-added. These will be simple enough to filter using SQL query params that are built in.

- Your own APIs (if any):
- There will be no other APIs being created other than the one that will be used by the frontend to interface with the backend.

- Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.)
- Prioritizing of items will be done by date added alone.

- If you have changed SW tools and frameworks or added any new one please describe it.

  -The only potential change that we would like to make is to use typescript on the backend and thus something like typeORM as our ORM.

6. Project Risk Assessment

Upon review of the organization of the SFSU Market application and its development team, the following risks have been identified:

● Skills Risks: The majority of the team is unfamiliar with the given development stack (ReactJS/Node/Express) for the project, and may need to accommodate time to gather and learn from various resources.

- In order to resolve the skills risk, while we might not have the depth knowledge necessary for our project, we are all motivated students, we will seek out help either from other team members, the internet, or our CTO, to learn particular skills and continue developing the application.

● Schedule Risks: As the development team is composed of senior level undergraduate students, scheduling conflicts may emerge due online school and pandemic.

- In order to resolve the identified risks, meetings will be held between the team lead and the rest of the team weekly online to discuss any issues with the application development. At scheduled extended meetings, we will periodically review the coding procedures that need to be done and research the methods of accomplishing them within the chosen framework and development stack. Code reviews will be conducted within the smaller teams to resolve any coding issues.

- During the weekly meeting, we discuss availability for the week and plan for any scheduling conflicts that may arise with any of the members. To ensure deadlines are met, tasks are assigned with soft deadlines depending on the availability discussed. An extended weekly meeting is typically scheduled to discuss issues in depth at these meetings.

● Legal/Content Risks: This project requires the use of media (uploaded images) which may potentially infringe upon the rights of its owners.

- For demo purposes, in order to avoid any legal issues, all images must be reviewed by team leads before being pushed to the live project site. The sources will be checked for the distributive rights of the images - only commercial-free (and no attribution required) images will be approved for demo use.

7. Project Management

For both milestone 1 and 2, tasks were assigned to each individual on the team based on their area of expertise. Front and back end leads handled those parts of the milestones that required more technical knowledge. This was done to prevent confusion and make the tasks more efficient. There could in the future be a problem with this, as other team members have not had an opportunity to learn about the technical aspects of this project.

The tasks so far have all been related to documentation. Soon, more tasks will arrive relating to writing code for the application. The goal for these tasks is to have front and back end teams consisting of those leads and one other group member, with the team lead moving back and forth between the two teams to ensure that everyone understands the current task and neither team moves so far away from the other team that the two sides of code cannot work together. The team leader started a Trello page for this group shortly after milestone 2 was assigned. From that point forward, assigned tasks have been kept track of on that platform. All five members have access to it and are able to access and edit the team Trello page.

Milestone 3 review summary and plans - CSC 648-848 Fall 2020 team 3
Date of review: 18 November 2020
Date of document: 18 November 2020

Milestone 3 review

Summary of feedback and tasks to do

- Homepage:
    - Colors are hard to see. Yellow on Blue
    - On pull down. Put a frame
    - Put class Id and info above search bar
    - Put a summarization of what the site does
    - Put below recently posted items (clickable)
    - After about should be create or post
    - Don’t put link to admin
    - Put recently posted on navbar
- Search:
    - Show filters below nav bar
    - Put some space below
    - Show category on every item
    - Maybe buy a button now?
- Login
    - ~~Forgot password? (Dummy link)~~
    - ~~Already have an account?~~
- Register
    - ~~Agree on terms clickable (mandatory)~~
    - ~~Make sure validation is Sfsu~~
    - ~~Black and white is good for contrast~~
- Individual Post
    - Make sure page is a new tab
- Dashboard
    - Have a title for specific user
    - For each message say what the product is with date
    - Don’t need create message
    - Should create message from posting
    - Pre-fill everything when making message
    - Always show the number of items/messages for each section
    - Must be able to delete products
- Create listing

    - Submit for approval on right
    - Cancel on left
    - Keep button colors consistent
    - Maybe use contact seller color for other buttons
    - Next to submit have dummy text that item may take 24 hours to be approved
    - Button should be create listing
    - Category should be filled from db
    - Image upload should be on bottom
    - Then say all other fields above are mandatory
    - Put a star for what is mandatory
- Database
    - Message must be for specific listing
    - Add listing FK
- Github
    - Make sure commits are descriptive
- Code style
    - Put couple lines of header comments for each page
    - Inline comments

List of tasks the team chose to focus on and implement for final delivery

The most important takeaway from this meeting was based on communication. Our code comments and Github commit messages are not descriptive enough. The reasons this happened are easy enough to understand. We are in a small bubble of five people, and we communicate between ourselves very regularly, so any change that is made is contextually understood. However, in an environment where the number of employees and components are very high, communication in these mediums is important. As such, the group will focus on implementing more code comments, including header comments, and making more descriptive Github commit messages.

We also received a large number of small recommended UI fixes, and these will also be a focus. All of them are useful and we will implement them in the pages they relate to.

List of final product P1 functions agreed at the meeting

Here are the changes made to the list of P1 requirements from milestone 2:

Moved “Site admins shall maintain a list of locations that are designated as safe for exchange, which will be publicly available for both registered and non-registered users to access” from P1 to P2

Moved “On load, the site shall display the most recent listings made on the application” from P2 to P1

SW Engineering CSC 648/848 Fall 2020
SFSU Market

Team # 03

Team Lead: Nicholas Green ngreen2@mail.sfsu.edu
Back end Lead: Steven McHenry
Front end Lead: Ahmad Afghan
Github Master: Lauren Luke
Waqas Hassan

“Milestone 4”

Due Date: December 8, 2020

Product Summary

Name: SFSU Market

List of P1 Functions:

Unregistered users:

- Unregistered users shall be able to create accounts on the application.
- All users shall be able to view listings.
- All users shall be able to search for listings by title and category.
- On load, the site shall display the most recent listings made on the application.

Registered users:

- Registered users shall be able to log in to their preexisting accounts.
- Registered users who have logged in shall be able to log out through the application.
- Registered, logged-in users shall be able to list an item or service for sale on this application and attach images to that listing.
- Registered, logged-in users shall be able to message other users privately.
- Registered users shall have a dashboard page to view their listings and messages.

ADMIN:

- Site admins shall be required to verify all listings before they are made public.
- Site admins shall be able to delete or lock listings made by registered users.
- Site admins shall be able to ban accounts from creating listings or messaging users.

Link: https://648-group3.xyz/

Usability Test Plan

Test Objective

The test objective is to determine the efficiency and ease-of-use of the search feature. Expose items with the search function by asking the tester to perform the search task with no prior instruction.

The search bar is for general use, not just for account users. Typical online browsers may not want to sign up for SFSU Market but may be interested to view its features and possible listings.

Test Plan

System Setup

- Testers will log into a computer with access to the internet and use the site URL address: https://648-group3.xyz/ with accepted internet browsers. Google chrome, Safari, Firefox.

Starting point

- Testers will begin at the site homepage wherein a search bar will be present.

Task to be accomplished

- Testers will attempt a search for listings and products of their choice.
- Testers will attempt to search for the item by category using the drop down menu.

Intended Users

- Students looking for items and services near San Francisco State University, who desire to buy them as soon as possible.
- Users will be assumed to have no technical expertise.

Completing Criteria

- Tester is satisfied that they have accomplished the search query and found items they were looking for

Feedback Process

- When the student is interested in their desired items that fit their personal tastes, the student may contact the seller via messaging option. The two may communicate with one another through message and may exchange phone numbers, information, etc.

Usability Questionnaire

1. I found the Search function easy to use (check one)

1	2	3	4	5
Totally Disagree	Disagree	Neutral	Agree	Totally Agree

2. I am satisfied with the search results.

1	2	3	4	5
Totally Disagree	Disagree	Neutral	Agree	Totally Agree

3. I understand the individual search parameters in relation to the search category.

1	2	3	4	5

Totally Disagree Disagree Neutral Agree Totally Agree

Comments:

QA Test Plan

- **Test Objective**
  - The test objective is to allow registered and unregistered users to easily find items on the website using the search function.

HW and SW setup

- **Hardware:**
  - Working laptop or computer that have access to web browsers, such as Google Chrome, Safari, or Firefox.

- **System Setup:**
  - The system setup starts off with testers using a computer to run the application on Google Chrome, Safari, or Firefox from which they will be prompted to the homepage. From the homepage, they have access to the search functions which they will use to locate the demo items listed within the application. Testers will

continue to use the search function to locate items within the application using a different web browser each time.

- **Intended users:**
    - San Francisco State University's faculty and students.

- **URL:**
    - https://648-group3.xyz/

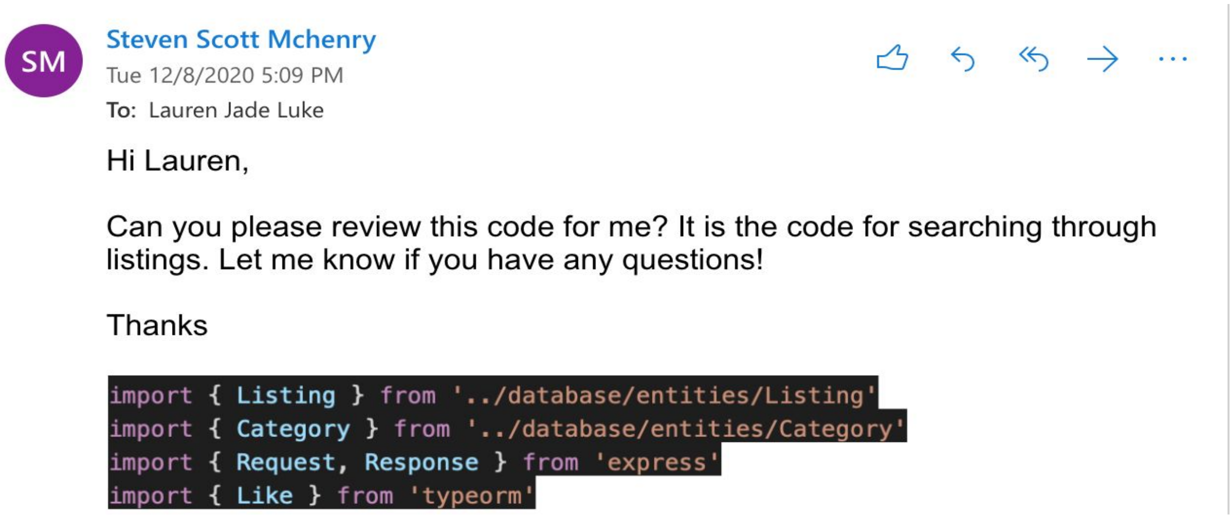
- **Feature to be tested:**
    - Search function.

- **QA Test Plan:**
    - The test plan is to test the functionality of the search functions within the application and by using this table, we can test if the application is capable of working through different web browsers such as Google Chrome, Safari, and Firefox.

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Google Chrome	Safari	Firefox
1	Search functions exist (search bar/button)	Search bar and button should be easily accessible from the homepage.	N/A	If the search bar and button are visible, then test pass. If the search bar and button are not visible, then the test fails.	PASS	PASS	PASS
2	Search functions work.	Enter the search item and click on the search button, the homepage should turn into a results page.	Type "SFSU" in the search bar and click the search button.	If the homepage changes to the results page, then the test passes. If the homepage doesn't change to the results page,	PASS	PASS	PASS

				then the test fails.			
3	Results are relevant to the search item.	The results page should display demo items relevant to the search.	Type "SFSU" in the search bar and click the search button.	If the results page shows available items with the name of "SFSU", then the test passes. If the results page doesn't show items with the name of "SFSU", then the test fails.	PASS	PASS	PASS

Code Review



Steven Scott Mchenry

Tue 12/8/2020 5:09 PM

To: Lauren Jade Luke

Hi Lauren,

Can you please review this code for me? It is the code for searching through listings. Let me know if you have any questions!

Thanks

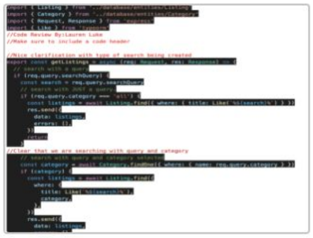
```
import { Listing } from '../database/entities/Listing'
import { Category } from '../database/entities/Category'
import { Request, Response } from 'express'
import { Like } from 'typeorm'
```

LL

Lauren Jade Luke

Thu 12/10/2020 1:00 PM

To: Steven Scott Mchenry



```
import { Listing } from '../database/entities/Listing'
import { Category } from '../database/entities/Category'
import { Request, Response } from 'express'
import { Like } from 'typeorm'
//Code Review By:Lauren Luke
//Make sure to include a code header

//Nice clarification with type of search being created
export const getListings = async (req: Request, res: Response) => {
  // search with a query
  if (req.query.searchQuery) {
    const search = req.query.searchQuery
    // search with JUST a query
    if (req.query.category === 'all') {
      const listings = await Listing.find({ where: { title: Like(`%${search}%`) } })
      res.send({
        data: listings,
        errors: [],
      })
      return
    }
//Clear that we are searching with query and category
    // search with query and category selected
    const category = await Category.findOne({ where: { name: req.query.category } })
    if (category) {
      const listings = await Listing.find({
        where: {
          title: Like(`%${search}%`),
          category,
        },
      })
      res.send({
        data: listings,
```

Hi Steven,
I have reviewed your code and have attached my corrections in this email. My corrections and positive comments you will find in red throughout the code. If you have any questions regarding my review please feel free to contact me.
Have a great day!
-Lauren Luke

...

Reply | Forward

```
import { Listing } from '../database/entities/Listing'
import { Category } from '../database/entities/Category'
import { Request, Response } from 'express'
import { Like } from 'typeorm'
//Code Review By:Lauren Luke
//Make sure to include a code header

//Nice clarification with type of search being created
export const getListings = async (req: Request, res: Response) => {
  // search with a query
  if (req.query.searchQuery) {
    const search = req.query.searchQuery
    // search with JUST a query.
    if (req.query.category === 'all') {
      const listings = await Listing.find({ where: { title: Like(`%${search}%`) } })
      res.send({
        data: listings,
        errors: [],
      })
      return
    }
//Clear that we are searching with query and category
    // search with query and category selected
    const category = await Category.findOne({ where: { name: req.query.category } })
    if (category) {
      const listings = await Listing.find({
        where: {
          title: Like(`%${search}%`),
          category,
        },
      })
      res.send({
        data: listings,
        errors: [],
      })
      return
    }
  }
//Nice in-line comments
  // search with just a category
  if (req.query.category === 'all') {
    res.send({
      data: await Listing.find(),
      errors: [],
    })
  }

  // actually category selected
  const category = await Category.findOne({ where: { name: req.query.category } })
  if (category) {
    const listings = await Listing.find({
      where: {
        category,
      },
    })
    res.send({
      data: listings,
      errors: [],
    })
    return
  }
}
//Overall, code looks good and is easy to follow
```

Self-check on best practices for security

Major assets of protection:

1. User data
2. Api keys
3. Database access

Major threats:

1. User data
    a. Unauthorized user gains access to personal user information
    b. Hacker uses user data to gain access to other websites with financial information
    c. User data gathered and sold on the dark web for scamming purposes
2. Api keys
    a. Hacker gathers api keys to rack up usage costs for their own gain
    b. Hacker uses your api keys and gets admins blocked from platforms
    c. Api keys are stored on github and can be easily viewed
3. Database access
    a. All data can be accessed and analyzed for proprietary gain
    b. Malicious user can delete all of your stored data
    c. Malicious user can overload and shut down your system

How we are protecting:

1. User Data
    a. Gated api routes for authenticated users
    b. Multiple levels of authentication for each step of accessing user information
    c. Encrypted passwords stored on database
2. Api keys
    a. They will be stored as environment variables and those environment variables will be added to the .gitignore
3. Database access
    a. Shielded via ssh and admin password protection
    b. Few people will have direct access to the database to minimize potential security holes

Input Validation:

1. Search bar
    a. Maximum of 40 characters
2. Email
    a. Maximum of 40 characters
    b. Must have sfsu.edu
    c. Cannot be used already
3. Password
    a. Minimum of 6 characters

4. Listing title
    a. Maximum of 40 characters
5. Listing image
    a. Must be .png, .jpg, .jpeg, .heic, of .gif
    b. Maximum file size of 5 mb

Non functional spec check

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). **DONE**

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers **DONE**

3. All or selected application functions must render well on mobile devices **ON TRACK**

4. Data shall be stored in the database on the team's deployment server. **DONE**

5. No more than 50 concurrent users shall be accessing the application at any time **ON TRACK**

6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. **ON TRACK**

7. The language used shall be English (no localization needed) **DONE**

8. Application shall be very easy to use and intuitive **ON TRACK**

9. Application should follow established architecture patterns **DONE**

10. Application code and its repository shall be easy to inspect and maintain **ISSUE**

The group as a whole needs to, and will, get better about code comments and labeling commits. A lack of communication in this way could make the code hard to maintain.

11. Google analytics shall be used **ON TRACK**

12. No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application **DONE**

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**

14. Site security: basic best practices shall be applied (as covered in the class) for main data items **ON TRACK**

15. Media formats shall be standard as used in the market today **DONE**

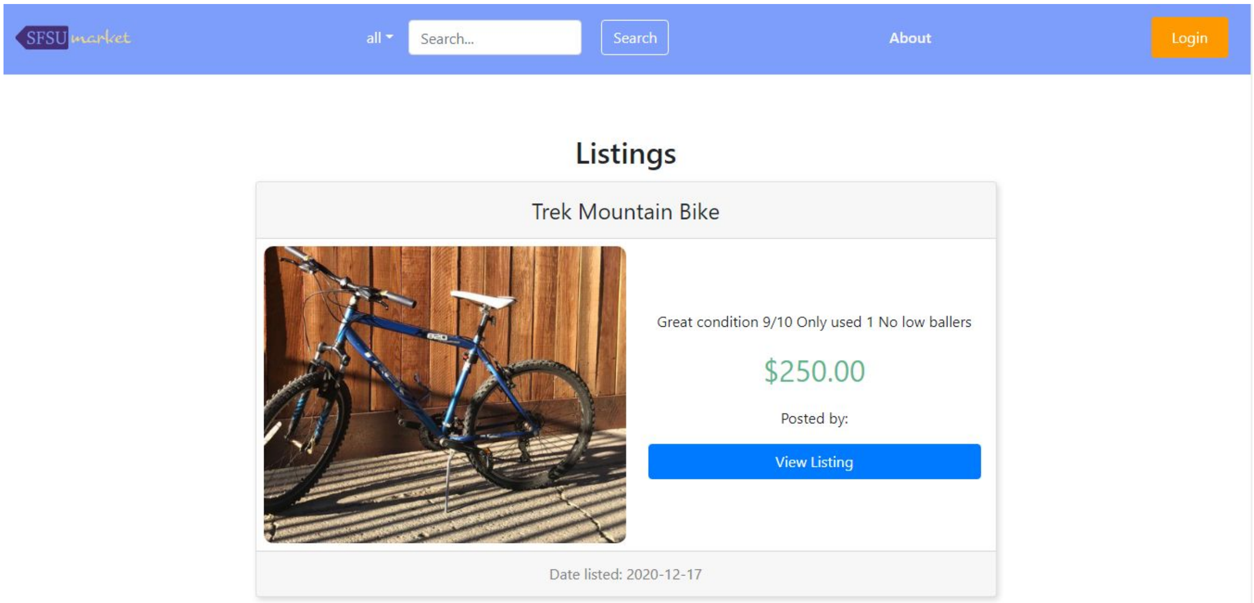
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development **DONE**

17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application). **ON TRACK**

END OF “PREVIOUS MILESTONES” SECTION

Product Screenshots

Launch



Listings

Trek Mountain Bike

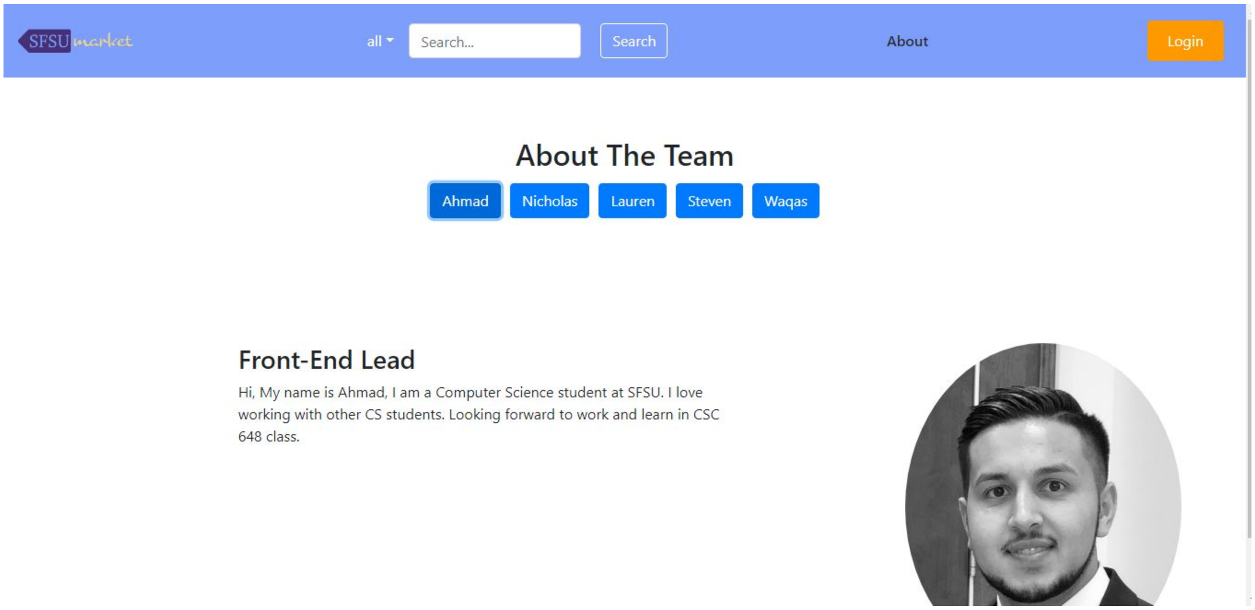
Great condition 9/10 Only used 1 No low ballers

$250.00

Posted by:

Date listed: 2020-12-17

About us page

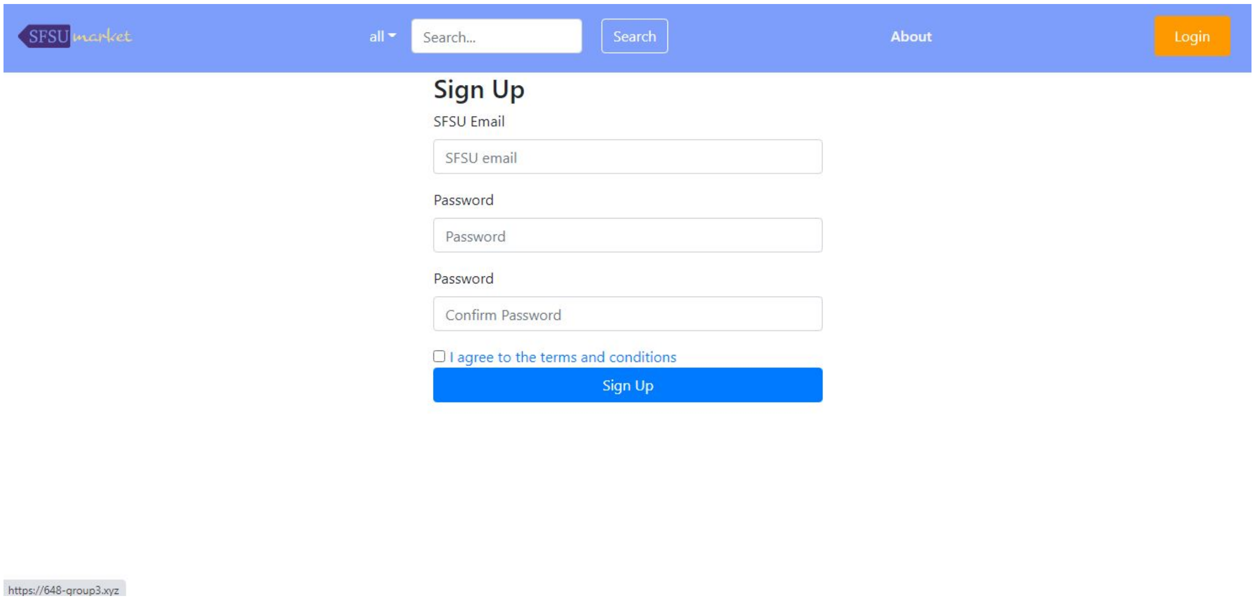


About The Team

Front-End Lead

Hi, My name is Ahmad, I am a Computer Science student at SFSU. I love working with other CS students. Looking forward to work and learn in CSC 648 class.

Create user



Sign Up

SFSU Email

SFSU email

Password

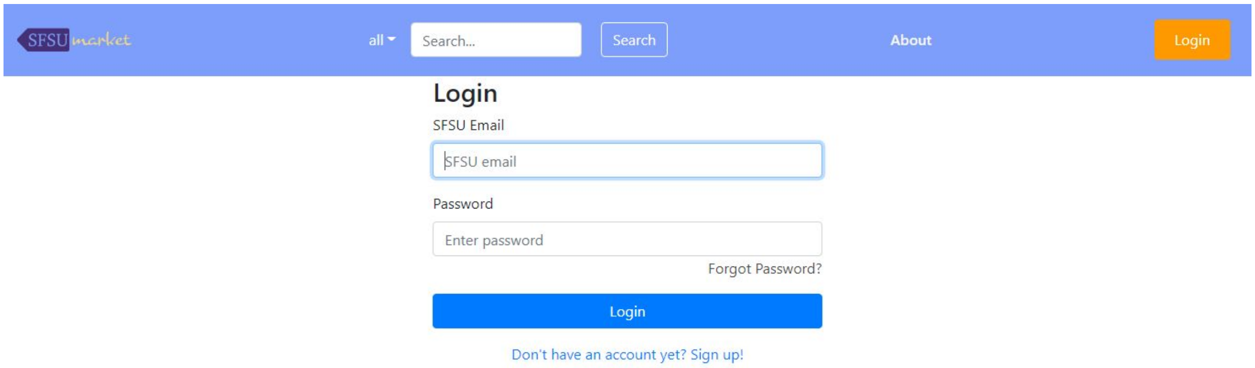
Password

Password

Confirm Password

□ I agree to the terms and conditions

Login



Login

SFSU Email

SFSU email

Password

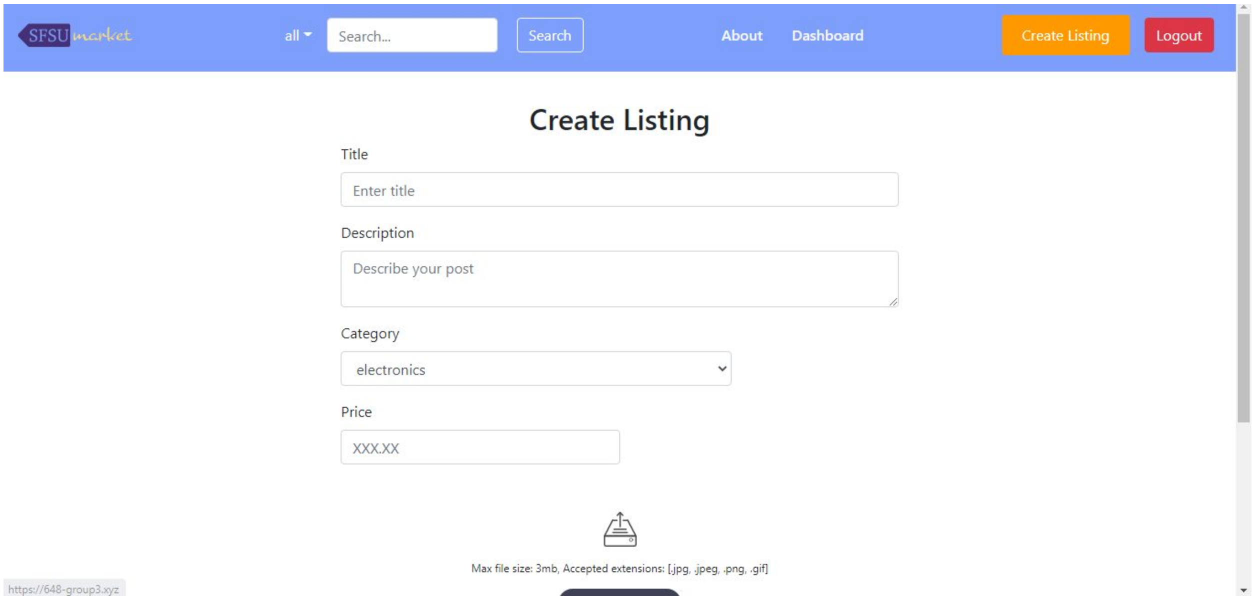
Enter password

Forgot Password?

Login

Don't have an account yet? Sign up!

Create Listing



Create Listing

Title

Enter title

Description

Describe your post

Category

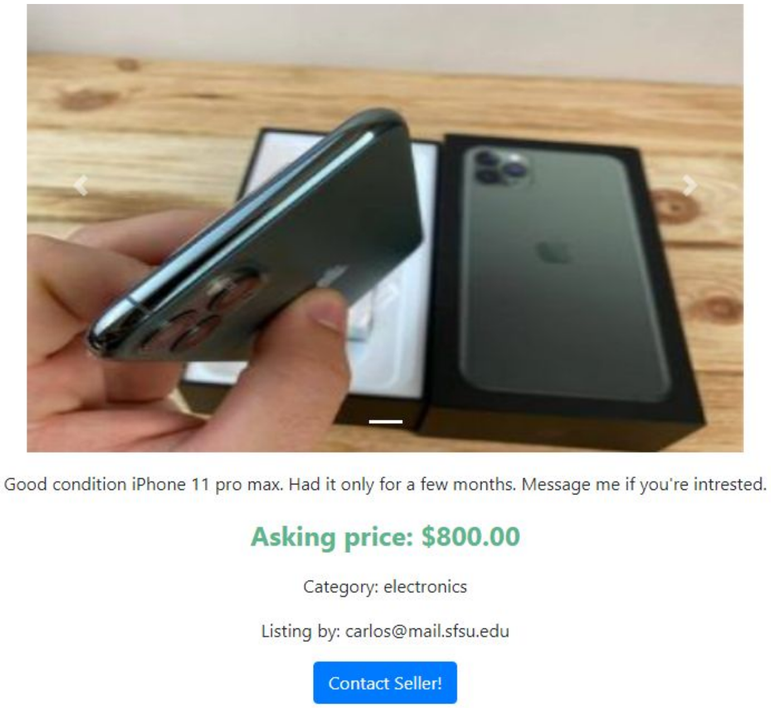
electronics

Price

XXX.XX

Max file size: 3mb, Accepted extensions: [.jpg, .jpeg, .png, .gif]

View Listing



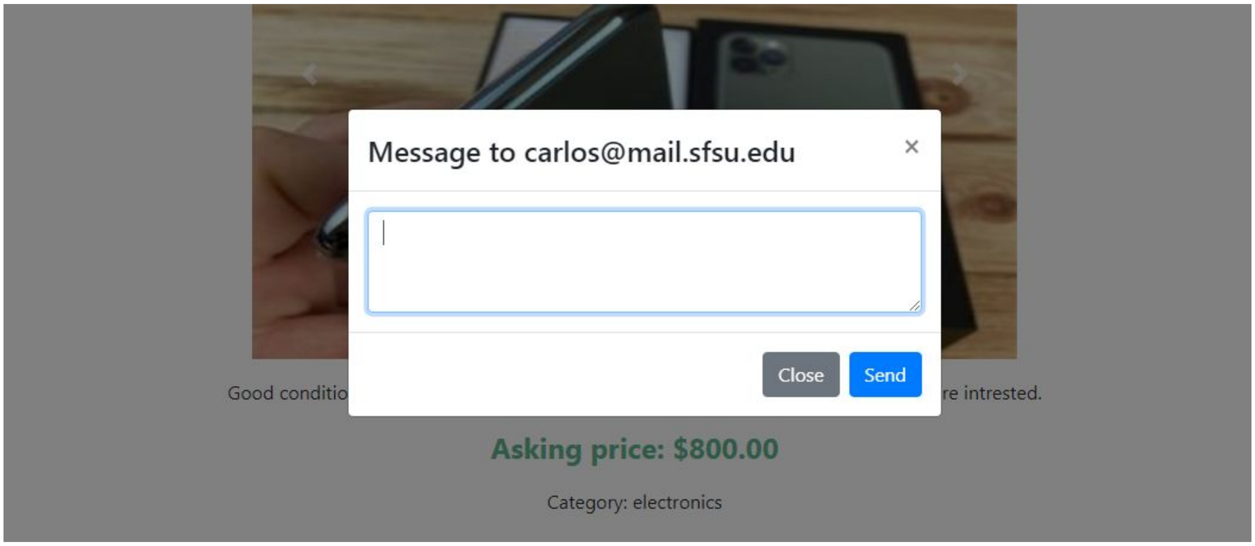
Good condition iPhone 11 pro max. Had it only for a few months. Message me if you're intrested.

Asking price: $800.00

Category: electronics

Listing by: carlos@mail.sfsu.edu

Message Prompt



Message to carlos@mail.sfsu.edu

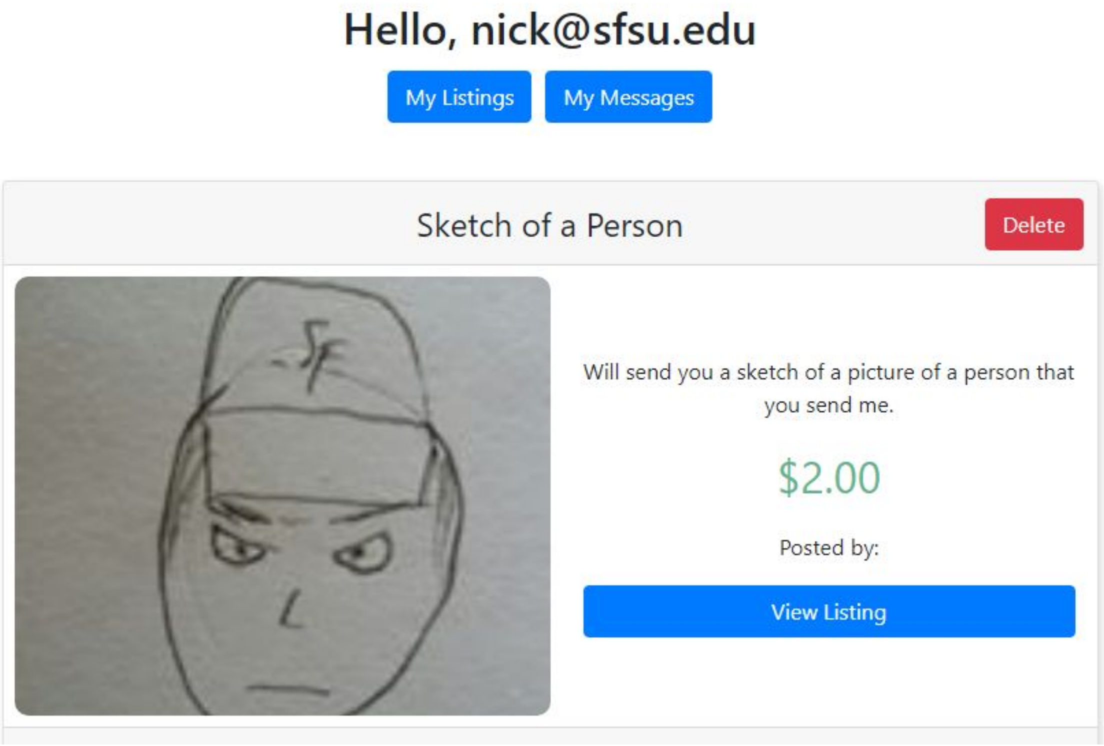
Good conditio

re intrested.

Asking price: $800.00

Category: electronics

User Dashboard



Hello, nick@sfsu.edu

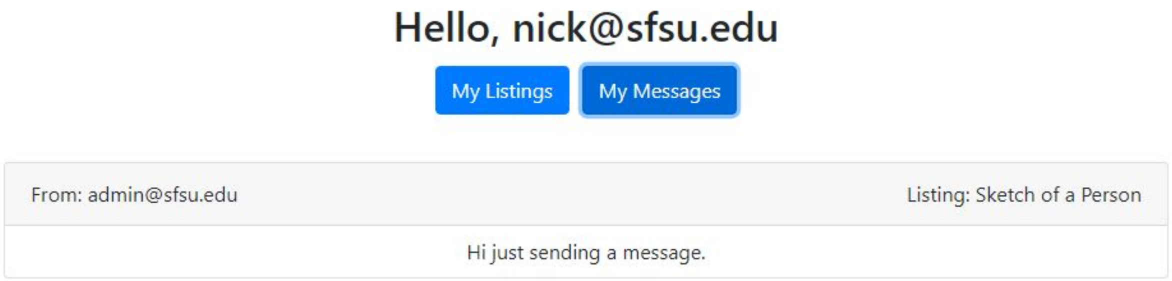
Sketch of a Person

Will send you a sketch of a picture of a person that you send me.

$2.00

Posted by:

Note: this listing does not appear on search until it is verified by an admin.



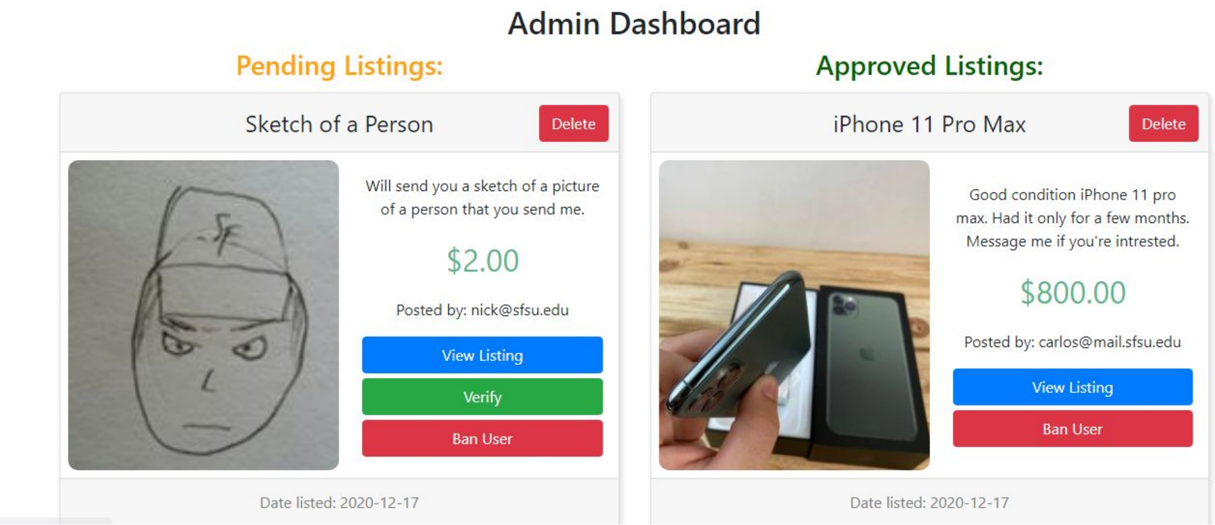
Hello, nick@sfsu.edu

From: admin@sfsu.edu

Listing: Sketch of a Person

Hi just sending a message.

Admin



Admin Dashboard

Pending Listings:

Sketch of a Person

Will send you a sketch of a picture of a person that you send me.

$2.00

Posted by: nick@sfsu.edu

Date listed: 2020-12-17

Approved Listings:

iPhone 11 Pro Max

Good condition iPhone 11 pro max. Had it only for a few months. Message me if you're intrested.

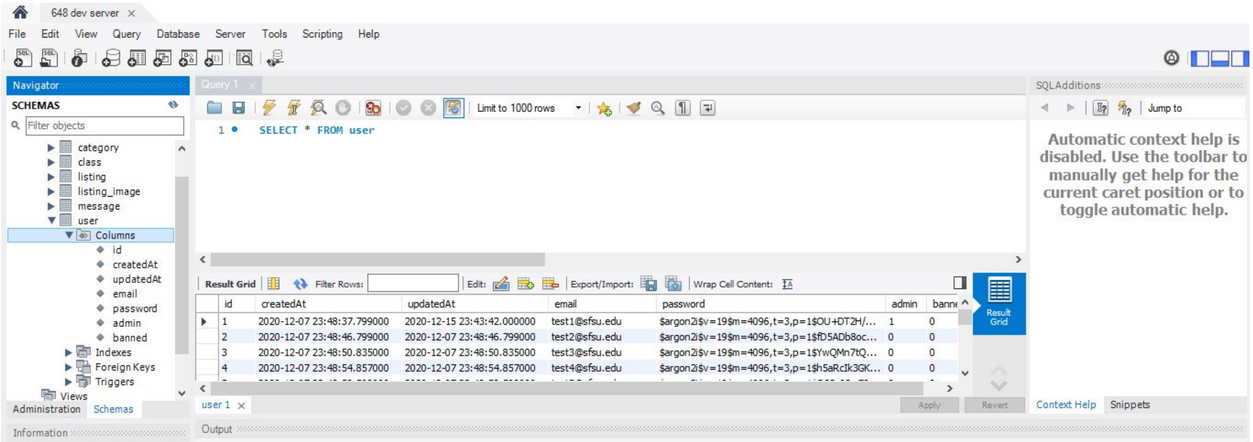
$800.00

Posted by: carlos@mail.sfsu.edu

Date listed: 2020-12-17

Database Organization

User table

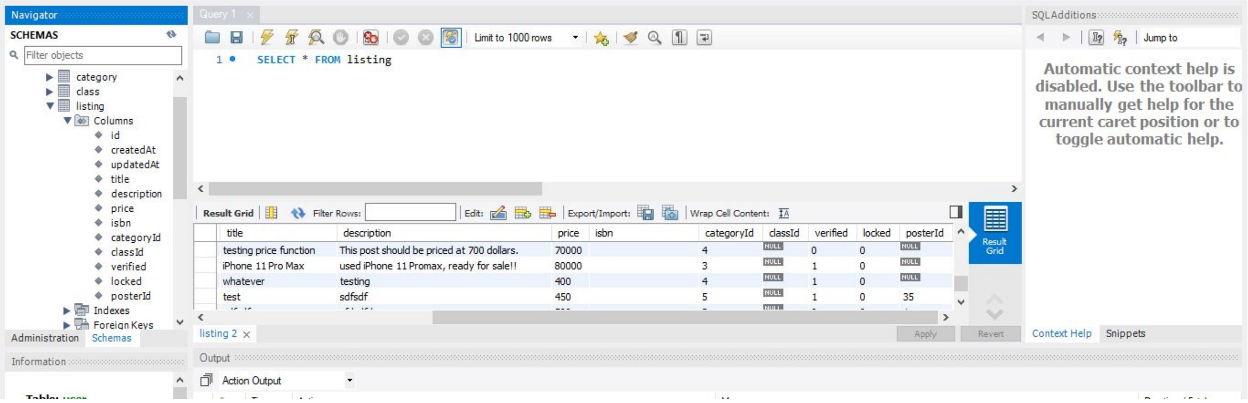


```
SELECT * FROM user
```

id	createdAt	updatedAt	email	password	admin	banne
1	2020-12-07 23:48:37.799000	2020-12-15 23:43:42.000000	test1@sfsu.edu	$argon2i$v=19$m=4096,t=3,p=1$OU+DT2H/...	1	0
2	2020-12-07 23:48:46.799000	2020-12-07 23:48:46.799000	test2@sfsu.edu	$argon2i$v=19$m=4096,t=3,p=1$fD5ADb8oc...	0	0
3	2020-12-07 23:48:50.835000	2020-12-07 23:48:50.835000	test3@sfsu.edu	$argon2i$v=19$m=4096,t=3,p=1$YwQMn7tQ...	0	0
4	2020-12-07 23:48:54.857000	2020-12-07 23:48:54.857000	test4@sfsu.edu	$argon2i$v=19$m=4096,t=3,p=1$h5aRcIk3GK...	0	0

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Listing table

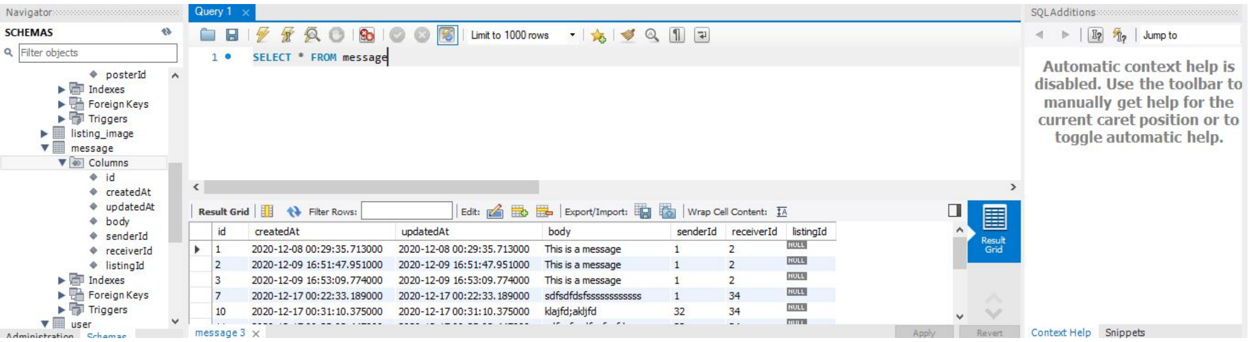


```
SELECT * FROM listing
```

title	description	price	isbn	categoryId	classId	verified	locked	posterId
testing price function	This post should be priced at 700 dollars.	70000		4	NULL	0	0	NULL
iPhone 11 Pro Max	used iPhone 11 Promax, ready for sale!!	80000		3	NULL	1	0	NULL
whatever	testing	400		4	NULL	1	0	NULL
test	sdfsdf	450		5	NULL	1	0	35

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Messages



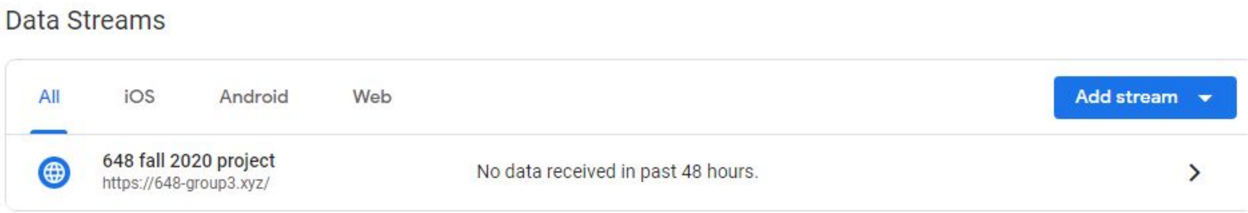
```
SELECT * FROM message
```

id	createdAt	updatedAt	body	senderId	receiverId	listingId
1	2020-12-08 00:29:35.713000	2020-12-08 00:29:35.713000	This is a message	1	2	NULL
2	2020-12-09 16:51:47.951000	2020-12-09 16:51:47.951000	This is a message	1	2	NULL
3	2020-12-09 16:53:09.774000	2020-12-09 16:53:09.774000	This is a message	1	2	NULL
7	2020-12-17 00:22:33.189000	2020-12-17 00:22:33.189000	sdfsdfdsfsssssssssss	1	34	NULL
10	2020-12-17 00:31:10.375000	2020-12-17 00:31:10.375000	klajfd;akljfd	32	34	NULL

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Google Analytics Plot

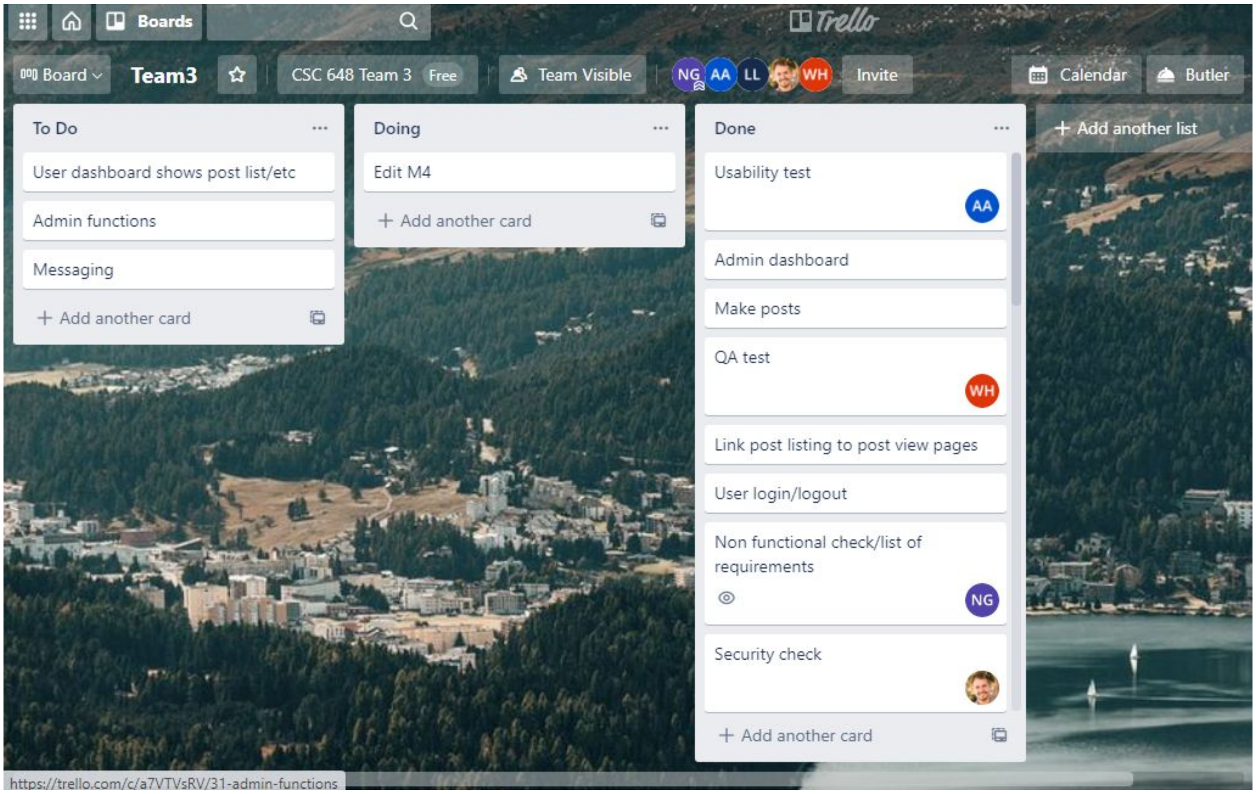
An attempt to install Google Analytics using React was unsuccessful. No data was sent to the page.



648 fall 2020 project
https://648-group3.xyz/

No data received in past 48 hours.

Project Management



To Do

- User dashboard shows post list/etc
- Admin functions
- Messaging
- + Add another card

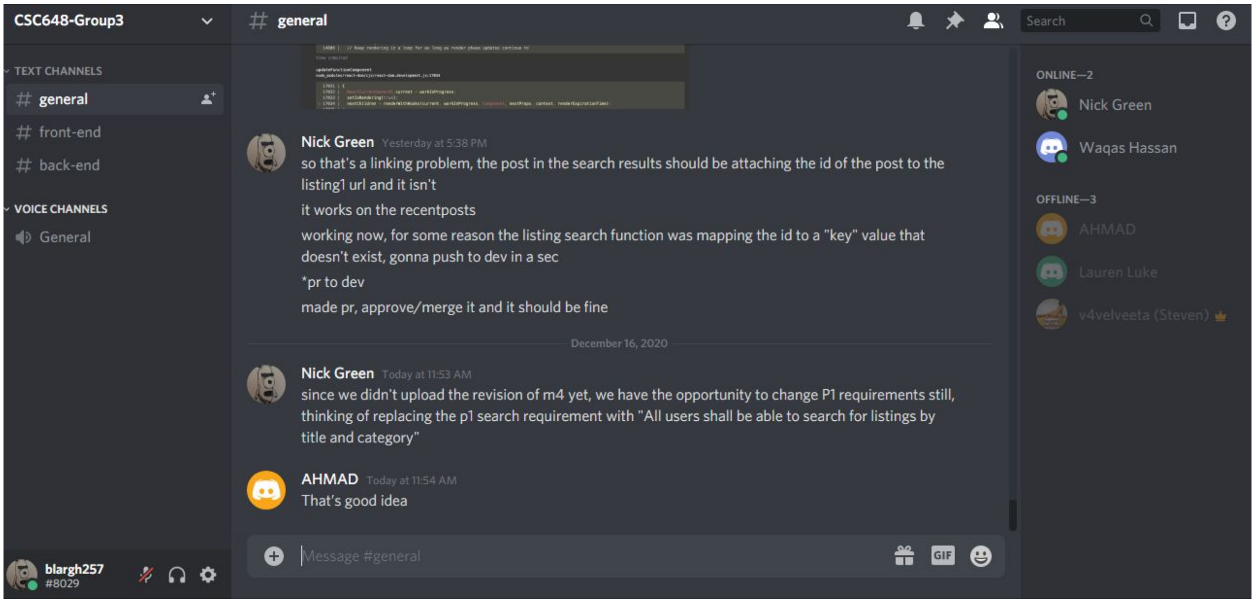
Doing

- Edit M4
- + Add another card

Done

- Usability test AA
- Admin dashboard
- Make posts
- QA test WH
- Link post listing to post view pages
- User login/logout
- Non functional check/list of requirements NG
- Security check
- + Add another card

https://trello.com/c/a7VTVsRV/31-admin-functions



```
14505 |   // Keep rendering in a loop for as long as render phase updates continue to
View compiled

updateFunctionComponent
node_modules/react-dom/cjs/react-dom.development.js:17034

  17031 | {
  17032 |   ReactCurrentOwner$1.current = workInProgress;
  17033 |   setIsRendering(true);
> 17034 |   nextChildren = renderWithHooks(current, workInProgress, Component, nextProps, context, renderExpirationTime);
```

Nick Green Yesterday at 5:38 PM

so that's a linking problem, the post in the search results should be attaching the id of the post to the listing1 url and it isn't

it works on the recentposts

working now, for some reason the listing search function was mapping the id to a "key" value that doesn't exist, gonna push to dev in a sec

*pr to dev

made pr, approve/merge it and it should be fine

December 16, 2020

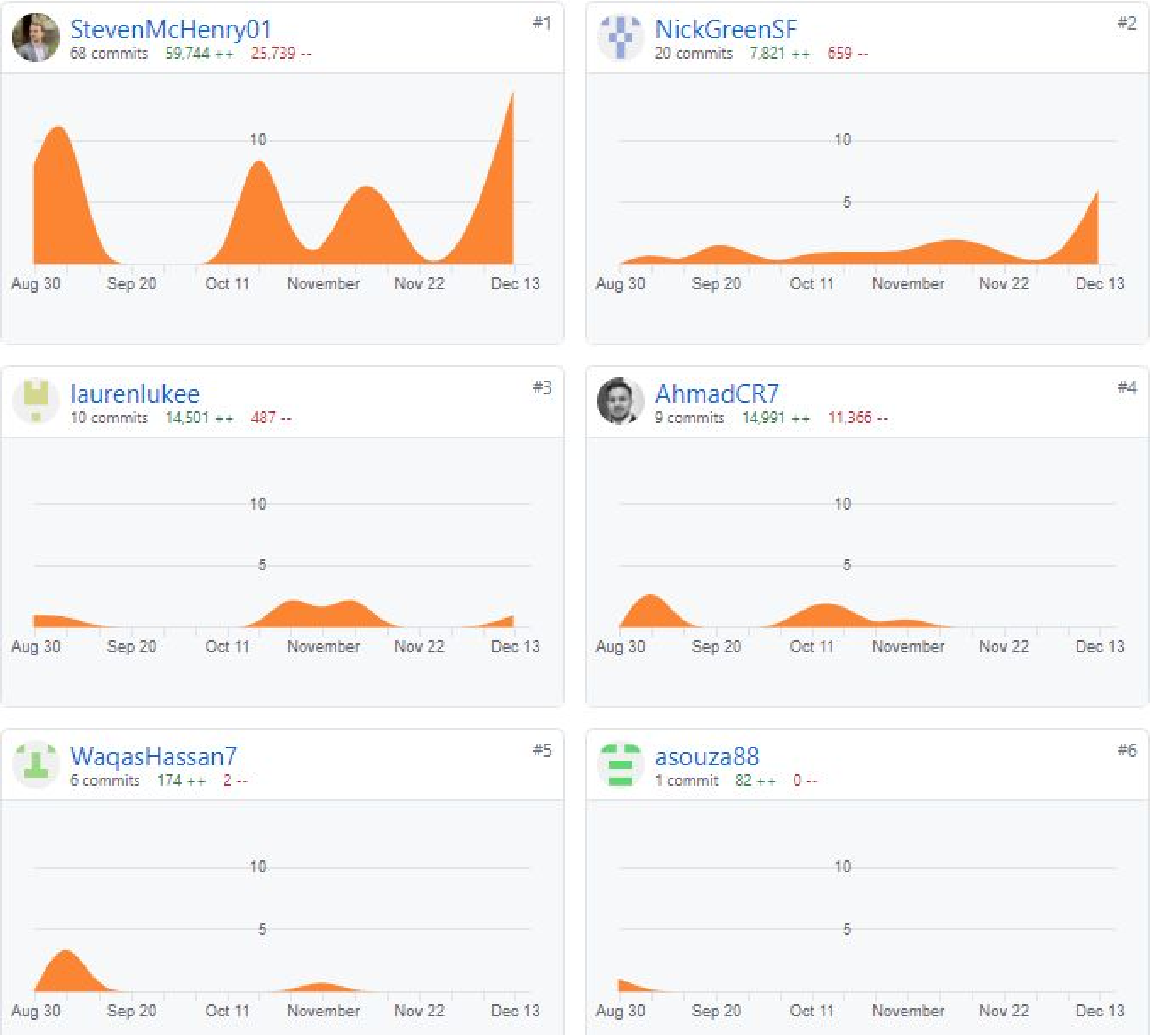
Nick Green Today at 11:53 AM

since we didn't upload the revision of m4 yet, we have the opportunity to change P1 requirements still, thinking of replacing the p1 search requirement with "All users shall be able to search for listings by title and category"

AHMAD Today at 11:54 AM

That's good idea

Commits to master



#1 StevenMcHenry01 68 commits 59,744 ++ 25,739 --

#2 NickGreenSF 20 commits 7,821 ++ 659 --

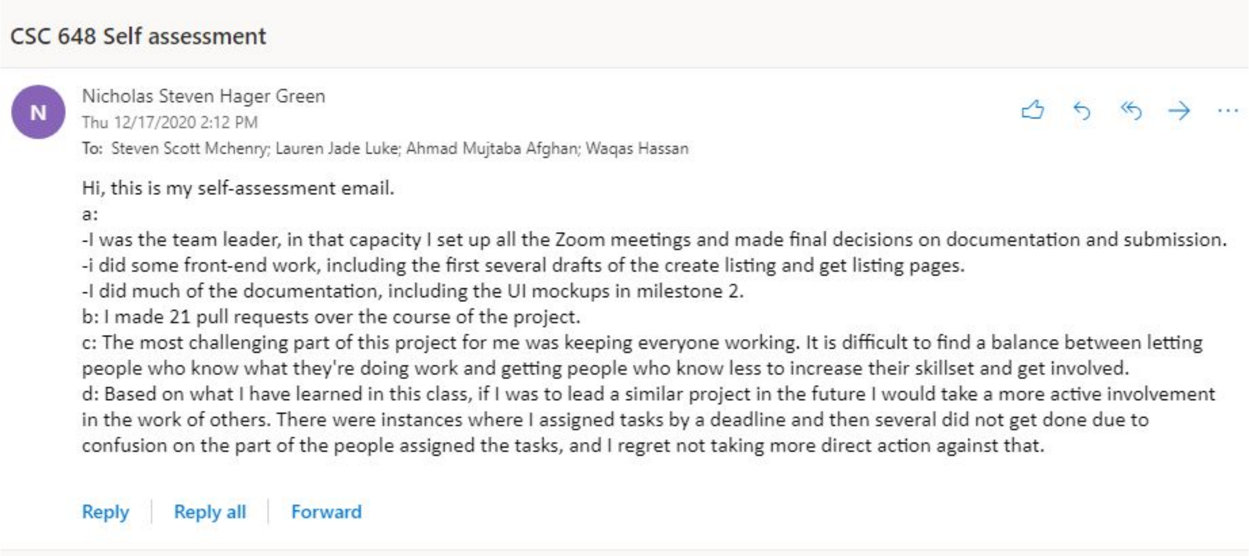
#3 laurenlukee 10 commits 14,501 ++ 487 --

#4 AhmadCR7 9 commits 14,991 ++ 11,366 --

#5 WaqasHassan7 6 commits 174 ++ 2 --

#6 asouza88 1 commit 82 ++ 0 --

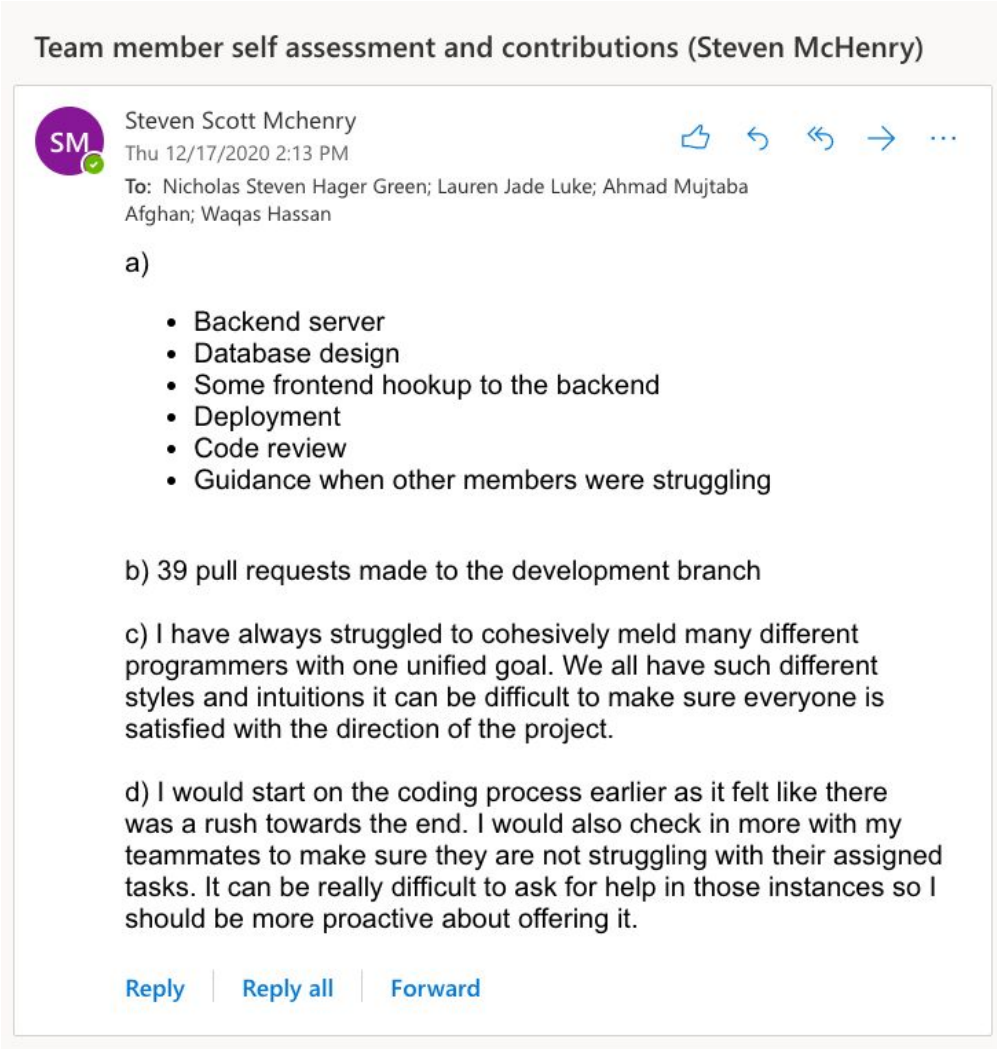
Team self-assessment



CSC 648 Self assessment

Nicholas Steven Hager Green
Thu 12/17/2020 2:12 PM
To: Steven Scott Mchenry; Lauren Jade Luke; Ahmad Mujtaba Afghan; Waqas Hassan

Hi, this is my self-assessment email.
a:
-I was the team leader, in that capacity I set up all the Zoom meetings and made final decisions on documentation and submission.
-i did some front-end work, including the first several drafts of the create listing and get listing pages.
-I did much of the documentation, including the UI mockups in milestone 2.
b: I made 21 pull requests over the course of the project.
c: The most challenging part of this project for me was keeping everyone working. It is difficult to find a balance between letting people who know what they're doing work and getting people who know less to increase their skillset and get involved.
d: Based on what I have learned in this class, if I was to lead a similar project in the future I would take a more active involvement in the work of others. There were instances where I assigned tasks by a deadline and then several did not get done due to confusion on the part of the people assigned the tasks, and I regret not taking more direct action against that.



Team member self assessment and contributions (Steven McHenry)

Steven Scott Mchenry
Thu 12/17/2020 2:13 PM
To: Nicholas Steven Hager Green; Lauren Jade Luke; Ahmad Mujtaba Afghan; Waqas Hassan

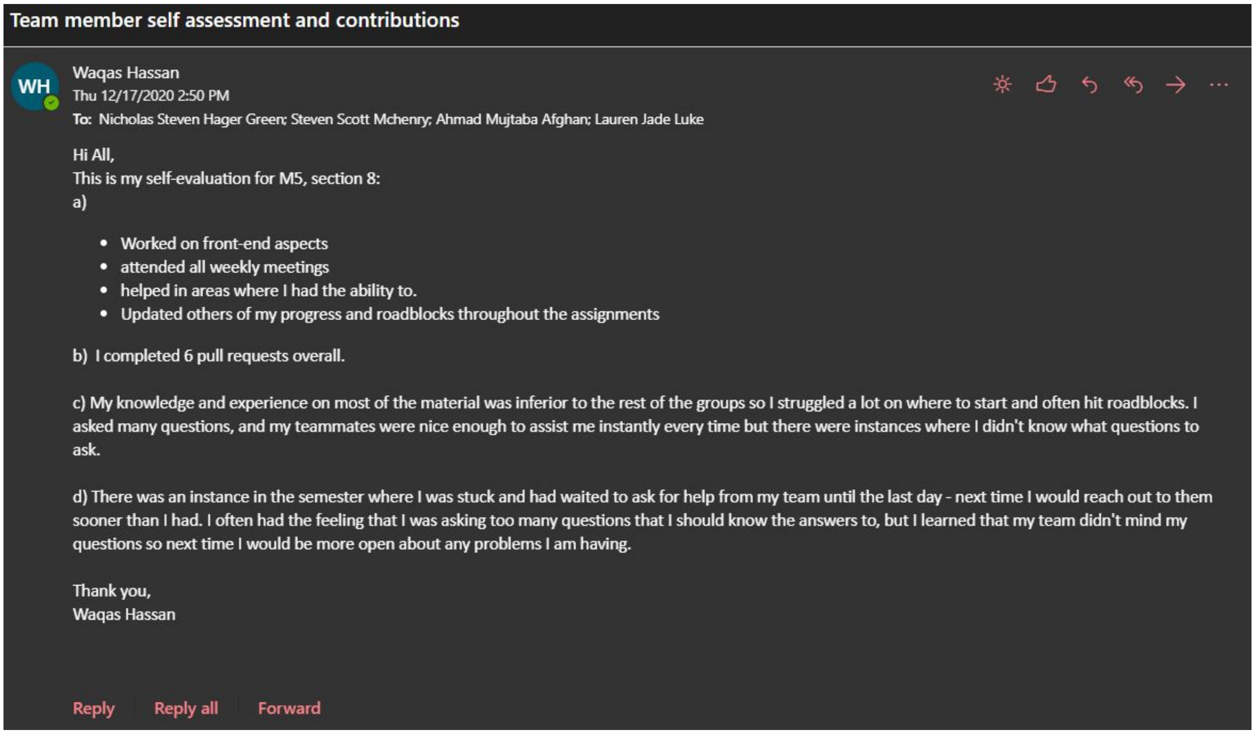
a)

- Backend server
- Database design
- Some frontend hookup to the backend
- Deployment
- Code review
- Guidance when other members were struggling

b) 39 pull requests made to the development branch

c) I have always struggled to cohesively meld many different programmers with one unified goal. We all have such different styles and intuitions it can be difficult to make sure everyone is satisfied with the direction of the project.

d) I would start on the coding process earlier as it felt like there was a rush towards the end. I would also check in more with my teammates to make sure they are not struggling with their assigned tasks. It can be really difficult to ask for help in those instances so I should be more proactive about offering it.



Team member self assessment and contributions

Waqas Hassan
Thu 12/17/2020 2:50 PM
To: Nicholas Steven Hager Green; Steven Scott Mchenry; Ahmad Mujtaba Afghan; Lauren Jade Luke

Hi All,
This is my self-evaluation for M5, section 8:
a)

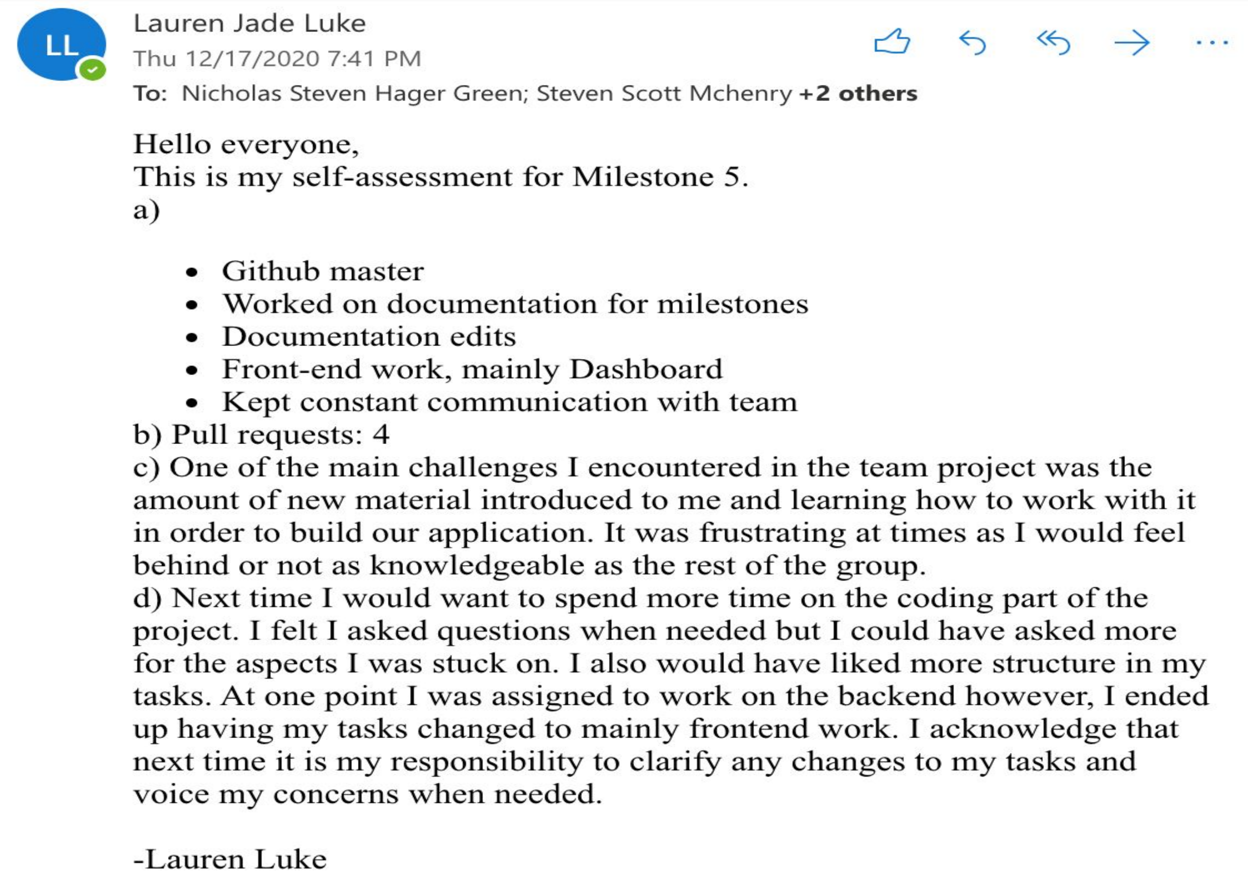
- Worked on front-end aspects
- attended all weekly meetings
- helped in areas where I had the ability to.
- Updated others of my progress and roadblocks throughout the assignments

b) I completed 6 pull requests overall.

c) My knowledge and experience on most of the material was inferior to the rest of the groups so I struggled a lot on where to start and often hit roadblocks. I asked many questions, and my teammates were nice enough to assist me instantly every time but there were instances where I didn't know what questions to ask.

d) There was an instance in the semester where I was stuck and had waited to ask for help from my team until the last day - next time I would reach out to them sooner than I had. I often had the feeling that I was asking too many questions that I should know the answers to, but I learned that my team didn't mind my questions so next time I would be more open about any problems I am having.

Thank you,
Waqas Hassan



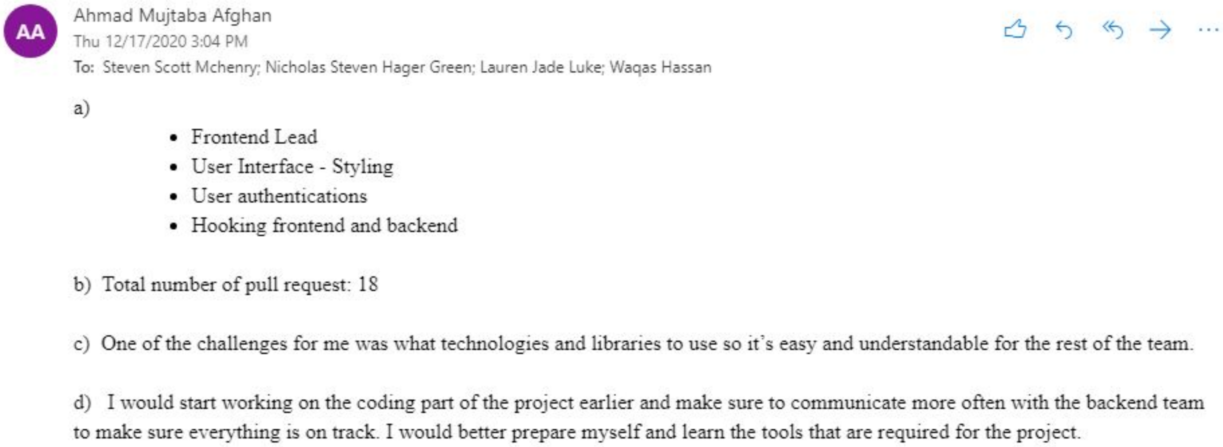
Lauren Jade Luke
Thu 12/17/2020 7:41 PM
To: Nicholas Steven Hager Green; Steven Scott Mchenry **+2 others**

Hello everyone,
This is my self-assessment for Milestone 5.
a)

- Github master
- Worked on documentation for milestones
- Documentation edits
- Front-end work, mainly Dashboard
- Kept constant communication with team

b) Pull requests: 4
c) One of the main challenges I encountered in the team project was the amount of new material introduced to me and learning how to work with it in order to build our application. It was frustrating at times as I would feel behind or not as knowledgeable as the rest of the group.
d) Next time I would want to spend more time on the coding part of the project. I felt I asked questions when needed but I could have asked more for the aspects I was stuck on. I also would have liked more structure in my tasks. At one point I was assigned to work on the backend however, I ended up having my tasks changed to mainly frontend work. I acknowledge that next time it is my responsibility to clarify any changes to my tasks and voice my concerns when needed.

-Lauren Luke



Ahmad Mujtaba Afghan
Thu 12/17/2020 3:04 PM
To: Steven Scott Mchenry; Nicholas Steven Hager Green; Lauren Jade Luke; Waqas Hassan

a)

- Frontend Lead
- User Interface - Styling
- User authentications
- Hooking frontend and backend

b) Total number of pull request: 18

c) One of the challenges for me was what technologies and libraries to use so it's easy and understandable for the rest of the team.

d) I would start working on the coding part of the project earlier and make sure to communicate more often with the backend team to make sure everything is on track. I would better prepare myself and learn the tools that are required for the project.