SW Engineering CSC648/848 Fall 2020

GatorMart

Team 6

Lothar Narins Team Lead

Aaron Colmenares Backend Lead

Allyson Leung Frontend Lead

Wilson Young GitHub Master

Saloni Bhatia Backend

Kevin Nunura Frontend

lnarins@mail.sfsu.edu

Milestone 5

December 17, 2020

Date Submitted: December 18, 2020

Date Revised: TBD

Table of Contents

Product Summary

Product Name: GatorMart

Major Committed Functions:

- All users shall be able to browse items for sale.
- All users shall be able to register with their unique SFSU email address.
- All users shall be able to log in with their registered account to message other sellers.
- All users shall be able to select an item for sale to read more information on the product.
- All users shall be able to browse the website utilizing categorized filters within the search bar.
- Registered users inherit all functionality allowed for unregistered users.
- Registered users shall be able to see and read their messages through their user dashboard.
- Registered users shall be able to privately message a user for an item they are interested in.
- Registered users shall be able to post items for sale or request specific products, and such posts will be reviewed by the website administrator.
- Registered users shall be able to log out of their account.
- The website administrator shall inherit all functionality allowed for non-registered and registered users.
- Administrators shall be required to use MySQLWorkbench to manage the website.
- Administrators shall be able to approve or deny post submissions from registered users.

URL: http://3.134.106.28:3000 (unavailable)

SW Engineering CSC648/848 Fall 2020

GatorMart

Team 6

Lothar Narins Team Lead

Aaron Colmenares Backend Lead

Allyson Leung Frontend Lead

Wilson Young GitHub Master

Saloni Bhatia Backend

Kevin Nunura Frontend

lnarins@mail.sfsu.edu

Milestone 1

September 16, 2020

Date Submitted: 09/24/2020

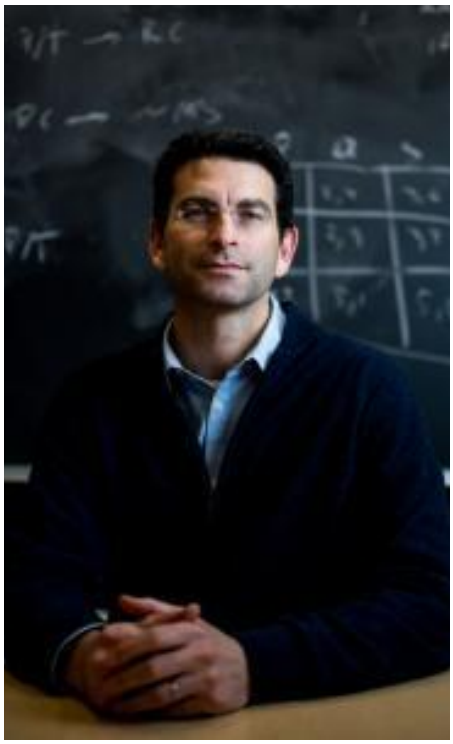
Date Revised: 10/01/2020

Table of Contents

Executive Summary

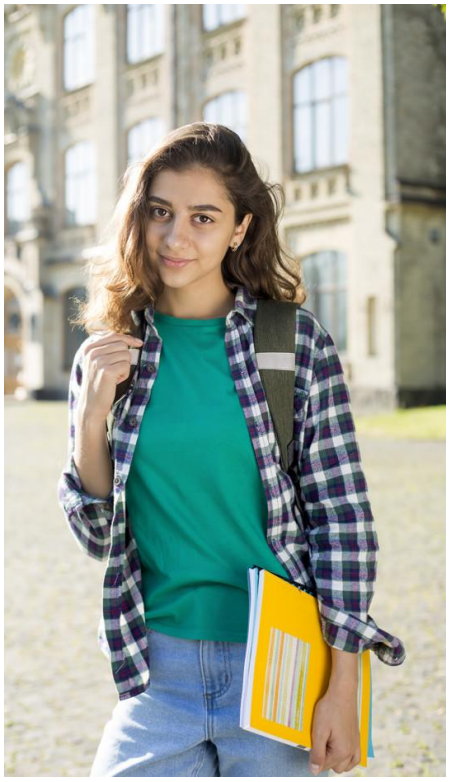
GatorMart is a sleek and functional website where SFSU students, staff, and faculty can go to buy, sell, or exchange select goods and services. By keeping a narrow focus and restricting registration to SFSU students and faculty, we are able to offer more streamlined services to our target audience, as well as greater security through SFSU email verification, which sets us apart from our more generalized competitors. The site allows buyers to contact sellers through a built-in messaging system. Users can browse through categories or search with keywords or for articles related to specific classes, and if they don't find what they're looking for, they may post a request, so that potential sellers can see what is in demand. Of course, all postings must be approved by an admin, so users are protected from anything untoward. We allow our users great flexibility in how they handle purchases by having them use their own payment method rather than through our website. We as a team are very excited to bring these features to the potential users of GatorMart.

Personae and Main Use Cases

Elan Mosk, Ph.D.	
Characteristics	
• First year lecturer at SFSU. • Works as a Senior Engineering Manager. • Has a wife and two children.	
Goals	
• Shape students to be successful post graduation. • Provide free or affordable resources to students.	
Skills	
• Experience as a software developer and engineer. • Thorough knowledge of online buying.	
Pain Points	
• Busy schedule with family and work. • Keep costs low in the Bay Area for his students.	

Ricardo	
Characteristics	
• Majoring in Business Administration. • South American native. • Works part-time at a local cafe.	
Goals	
• Purchase a home in the Bay Area. • Be financially stable.	
Skills	
• Fluent in the Spanish and English language. • Knowledgeable in basic laws and regulations.	
Pain Points	
• Not experienced in using computer systems. • Struggles with time management.	

Liang	
Characteristics	
• Senior in SFSU Bachelor in Nursing program. • Volunteer at local nursing and hospice homes. • Relocating to Minnesota post graduation.	
Goals	
• Save money for medical school. • Volunteer for local healthcare services.	
Skills	
• Familiar with web browsing and shopping. • Great communication skills.	
Pain Points	
• Owns unused and space-consuming furniture. • Tight on money and busy schedule.	

Nadia	
Characteristics	
• SFSU senior in Computer Science program. • Admin of GatorMart site. • Working abroad, originally from India.	
Goals	
• Maintain a safe environment on the website. • Develop the website to assist all student and staff needs.	
Skills	
• Backend web developer experience. • Former intern at Amazon India Software Team.	
Pain Points	
• Time management between school and job. • Managing influx of posts by students.	

Use Cases
Case 1: Registering and Posting
Dr. Mosk has just finished a faculty meeting discussing the emotional and financial impact on current San Francisco State (SFSU) Students as a result of the global pandemic and seasonal local wildfires. As a professor that cares for the wellbeing of his students, Dr. Mosk was able to get a textbook that was being offered for free by a reputable academic institution online approved by the Computer Engineering department for one of his courses. Knowing that the class occupancy was low due to textbook costs, Dr. Mosk decided to motivate future students to enroll in the course by making his cost-free materials known to the university community through GatorMart. He logs on to our website on his laptop and proceeds to search for his course ID. When he finds out that there has been no resources posted for this course, Dr. Mosk proceeds to create a post in GatorMart specific to his course ID with the appropriate information that directs SFSU students to download the free course textbook from the reputable educational website. Upon submission, he is prompted to register and log in to ensure he has the appropriate credentials to post. Dr. Mosk is prompted with an error message when submitting his registration because he accidentally entered the wrong SFSU designated email. He is redirected to register and correctly fills out his information this time around and is able to post his pre-filled post upon admin approval.
Case 2: Buying
Ricardo has just woken up late for his first day as a freshman at San Francisco State University (SFSU). He is excited for this day, but is also exhausted from finishing up an overnight shift the day prior at a local cafe shop and is now running late for school. As Ricardo is traveling on the BART to SFSU, he realizes that he forgot to purchase the required textbook for his first class. He immediately goes to the GatorMart mobile site, logs in with his active account, and seamlessly searches for the title of the book he needs. He notices that there is one post created by a reliable and 5/5 Gator Star seller offering the textbook he required. Ricardo quickly utilizes the simple one-way system from our site to inform the seller of his interest while providing his contact information for their scheduling convenience. Upon successful contact from the seller, both users set up a time to meet up at the university's J. Paul Leonard Library which is on his way to his first class. Ricardo meets his seller at their public appointed place and time and safely purchases his textbook by paying his seller through an external money exchange application. On his way to his course, Ricardo enthusiastically rates the seller another 5 Gator Stars.
Case 3: Selling
Liang is an avid volunteer at her local nursing home and is currently preparing to relocate to Minnesota for a job opportunity. However, due to her constant volunteer jobs, she does not have enough money to purchase a plane ticket to Minnesota. As she is cleaning out her room, Liang realizes that she cannot take multiple furniture pieces from her dorm. Liang decides to go onto the mobile GatorMart site and logs in with her already active account. She creates a

single post and attaches all of the images from her phone of the furniture she has for sale and annotates their respective price point in the post description. After Liang's post is approved by site admin, an incoming freshman at SFSU interested in the item for sale immediately sends a message to Liang with their contact information. Liang and the freshman exchange information through their mutual desired means of communication and meet up to purchase one item from Liang's post. After the sale, Liang returns to the mobile GatorMart site, and because she is already logged in, she goes directly to her post and updates it to only reflect the images of the furniture currently available for sale without needing reapproval.
Case 4: Administering Posts
Nadia is trying to keep up with her last semester set of courses at San Francisco State University (SFSU) while concurrently reviewing an influx of post submissions by faculty, staff, and students for the GatorMart website. As a busy individual, Nadia is glad to have an easy and efficient system set up in our website to quickly review and approve or deny posts. At the beginning of her shift at GatorMart, Nadia travels to our website utilizing a GatorMart employee laptop, logs in with her administrator account, and goes to her work queue. She filters her work queue by posts pending approval and begins reviewing each post. Nadia encounters a post with an inappropriate image and content. She denies the publication of this post and immediately deletes the seller's account for going against GatorMart terms and conditions. The next post Nadia reviews has a buyer requesting materials for a specific course ID which abides by GatorMart's terms and policies, she approves this post. Before finishing her shift, Nadia filters her work queue by reported posts. She encounters a highly reported post that did not comply with the terms and conditions to a certain degree. She deletes the post and informs the poster of their error.

Data Glossary

GatorMart users can be:

1. ***Anonymous users:*** Users that are not logged in.
2. ***Active users:*** Users that are logged in.
3. ***Administrators:*** Active users with special privileges.

Types of posts on GatorMart:

4. ***For-sale:*** An admin approved post created by a registered user about an item that they want to sell. The user provides a description, image, and price of the item. There is an option for other users to message the poster that they are interested in buying the item.
5. ***Requests:*** An admin approved post created by a registered user in search of a specific item. Other users with access to the original request can fulfill this sale.
6. ***Class Resource:*** An admin approved post created by a registered user that offers supplemental tools and resources specific to a course ID.
7. ***Wishlist:*** A compiled list of posts that a registered or unregistered user that wishes to save for future viewing
8. ***Work Queue:*** A queue only accessible to the website administrator. This work queue contains all posts pending approval and posts approved. The administrator will have access to either approve or deny posts from being published on a website from this feature.
9. ***Gator Review:*** Our website's user review system set in place for the safety of our clients. This system includes personal statements written by other users about a specific user as well as a five star rating system called Gator Stars.
10. ***Report:*** A button functionality to flag posts not complying with the website terms and conditions. Administrators will be able to review flagged posts for inappropriate content.
11. ***Messaging:*** A one-way system for users to communicate to another user they are interested in buying an item and provide the seller with their contact information.

Functional Requirements

1. **Anonymous Users**
   1.1 Shall be able to browse and search for items for sale.
   1.2 Shall be able to add posts to their wishlist.
   1.3 Shall be able to log in.
   1.4 Shall be able to register ONLY with an email ending in sfsu.edu.
   1.5 Shall be able to browse the website utilizing categorized filters.

2. **Active Users**
   2.1 Shall inherit all functionality allowed for anonymous users.
   2.2 Shall be able to post items for sale/share or request items, and such posts will be reviewed by the website administrator.
   2.3 Shall be able to privately message a user for an item they are interested in.
   2.4 Shall be able to log out of their account.
   2.5 Shall be able to rate a user utilizing the Gator Review system.
   2.6 Shall be able to flag posts not complying with the website terms and conditions.

3. **Administrator**
   3.1 Shall inherit all functionality allowed for active users.
   3.2 Shall have a special administrator account with access to a work queue.
   3.3. Shall be able to approve or deny post submissions form users.
   3.4 Shall be able to delete posts and users that go against the website policy and procedures.
   3.5 Shall be able to review flagged posts.

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 two major browsers
3. All or selected application functions must render well on mobile devices
4. Data shall be stored in the database 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 (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application should follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
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
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
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).

Competitive Analysis

	SFSU Bookstore	Amazon	eBay	Craigslist	GatorMart
Requires SFSU login	+				+
Post selling and buying offers				+	+
Search by Class ID					+
One-way messaging system					+
Free Class Resources				+	+
Review System		+	+		+
Report System		+	+	+	+
Wishlist		+	+		+

GatorMart provides users with an array of services and features that are unique for San Francisco State (SFSU) students and staff. We require a valid SFSU email to register an account on our website to post selling or buying offers, similar to the SFSU Bookstore. This system is important as it only allows current SFSU students, faculty, and staff to purchase or sell goods within our application. Competing companies such as Amazon and eBay allow users to buy items as a guest, but because GatorMart is specifically designed for SFSU members, requiring a unique SFSU email helps us maintain a safe environment for our clients. Additionally, GatorMart gives our active users freedom to post, whether they are posting goods to sell or are simply looking for offers on goods to purchase. If users are searching for a specific good to purchase but don't see it being offered on our website, they can create a post describing the good they are looking for and how much they are willing to pay for it. The only other website with a similar feature to this is Craigslist, but our system provides a much secure and internal exchange between our clients. Furthermore, our website will provide a filtered search feature that allows users to filter posts by class ID, which provides a more narrowed result than our competitors. Moreover, GatorMart will

provide our active users with a simple but secure one-way messaging system between the users to the original poster through our application. GatorMart will include a report system where users will be able to click on a button to report a post, which notifies the admins to review the post. GatorMart will also include a wishlist which allows users to save to view at a later time. Lastly, GatorMart prides itself in offering the ability for our active users to offer free supplemental class resources and tutoring with admin approval to all users. This is a unique feature our team is eager to offer in our goal to provide assistance to low income and financially struggling students and staff of the SFSU community to increase the overall success of our clients during difficult times.

High-Level System Architecture and Technologies

- Frameworks, APIs, Tools and Systems
    - **Node.js** backend to handle both serving the website as well as providing internal API endpoints. The versatility of Express's middleware/boilerplate allows us to internally implement features such as authentication, SQL access, etc.
    - **Mysql database** for storing users, media items, logs. Database is hosted locally on the same server.
    - **Github** allows the team to keep track of changes made to the code and provides version control.
    - **Bootstrap** is an HTML API that allows for cross browser support with built in features.
- Deployment platform (SW and server)
    - **AWS EC2 Ubuntu** this platform will allow us to host the site and accompanying data in an easily accessible place which has a lot of documentation.
- Supported Browsers
    - Chrome is the most commonly used web browser.
    - Safari for support on Apple devices.
    - Firefox for browser diversity.

Team and Roles

Team Member	Role
Lothar Narins	Team Lead
Aaron Colmenares	Backend Lead
Allyson Leung	Frontend Lead
Wilson Young	GitHub Master
Saloni Bhatia	Backend
Kevin Nunura	Frontend

Checklist

So far all team members are engaged and attending ZOOM sessions when required	**ON TRACK**
Team found a time slot to meet outside of the class	**DONE**
Backend, frontend leads and GitHub master chosen	**DONE**
Team decided and agreed together on using the listed SW tools and deployment server	**DONE**
Team ready and able to use the chosen back- and frontend frameworks and those who need to learn are working on learning and practicing	**ON TRACK**
Team lead ensured that all team members read the final M1 and agree/understand it before submission	**DONE**
GitHub organized as discussed in class (e.g. master branch, development branch, folder for milestone documents, etc.)	**DONE**

SW Engineering CSC648/848 Fall 2020

GatorMart

Team 6

Lothar Narins Team Lead

Aaron Colmenares Backend Lead

Allyson Leung Frontend Lead

Wilson Young GitHub Master

Saloni Bhatia Backend

Kevin Nunura Frontend

lnarins@mail.sfsu.edu

Milestone 2

October 1, 2020

Date Submitted: 10/15/2020

Date Revised: TBD

Table of Contents

Executive Summary

GatorMart is a sleek and functional website where SFSU students, staff, and faculty can go to buy, sell, or exchange select goods and services. By keeping a narrow focus and restricting registration to SFSU students and faculty, we are able to offer more streamlined services to our target audience, as well as greater security through SFSU email verification, which sets us apart from our more generalized competitors. The site allows buyers to contact sellers through a built-in messaging system. Users can browse through categories or search with keywords or for articles related to specific classes, and if they don't find what they're looking for, they may post a request, so that potential sellers can see what is in demand. Of course, all postings must be approved by an admin, so users are protected from anything untoward. We allow our users great flexibility in how they handle purchases by having them use their own payment method rather than through our website.

We as a team are very excited to bring these features to the potential users of GatorMart. Our team consists of aspiring Computer Science majors in the CSC 648 class. We are all very enthusiastic about bringing you a functional and reliable means to sell or buy items from other SFSU members. Our motivation behind this project is to help fellow SFSU members safely sell or buy their goods while also providing free resources to those who need it. We will strive to present the best possible product to help our community.

Main Data Items and Entities

GatorMart users are stored in the '**users**' table. They can be:

1. ***Anonymous users:*** Users that are not logged in. These have no entry in the '**users**' table. Their data is stored in cookies.
2. ***Registered users:*** Users that are logged in. These get an entry in the '**sessions**' table when they log in.
3. ***Administrators:*** Registered users with special privileges. Their entry in the '**users**' table has a usertype of admin.

GatorMart posts are stored in the '**posts**' table. These will be hidden from users until approved by an administrator. They can be:

4. ***For-sale:*** A post created by a registered user about an item that they want to sell. The user provides a description, image, and price of the item. There is an option for other users to message the poster that they are interested in buying the item.
5. ***Requests:*** A post created by a registered user in search of a specific item. Other users with access to the original request can fulfill this sale.
6. ***Class Resource:*** A post created by a registered user that offers supplemental tools and resources specific to a course ID.

Other items:

7. ***Messaging:*** A one-way system for users to communicate to another user they are interested in buying an item and provide the seller with their contact information. These are kept track of in the '**messages**' table.
8. ***Flag:*** A button functionality to flag posts not complying with the website terms and conditions. Administrators will be able to review flagged posts for inappropriate content. These will be stored in the '**flags**' table.

Low-priority items:

9. ***Wishlist:*** A compiled list of posts that an anonymous or registered user wishes to save for future viewing. This will be stored in cookies.
10. ***Gator Review:*** Our website's user review system set in place for the safety of our clients. This system includes personal statements written by other users about a specific user as well as a five star rating system called Gator Stars. This is a priority 3 item which would be stored in the database in a '**reviews**' table (not shown).

Functional Requirements

Priority 1

1. **Anonymous Users**

   1.1 Shall be able to browse and search for items for sale.
   1.2 Shall be able to log in.
   1.3 Shall be able to register ONLY with an email ending in sfsu.edu.
   1.4 Shall be able to browse the website utilizing categorized filters.

2. **Registered Users**

   2.1 Shall inherit all functionality allowed for anonymous users.
   2.2 Shall be able to post items for sale/share or request items, and such posts will be reviewed by the website administrator.
   2.3 Shall be able to privately message a user for an item they are interested in.
   2.4 Shall be able to log out of their account.

3. **Administrator**

   3.1 Shall inherit all functionality allowed for registered users.
   3.2 Shall be required to use MySQLWorkbench to manage the website.
   3.3 Shall be able to approve or deny post submissions from users.

Priority 2

2. **Registered Users**

   2.5 Shall be able to flag posts not complying with the website terms and conditions.

3. **Administrator**

   3.4 Shall be able to delete posts and users that go against the website policy and procedures.
   3.5 Shall be able to review flagged posts.

Priority 3

1. **Anonymous Users**

   1.5 Shall be able to add posts to their wishlist.

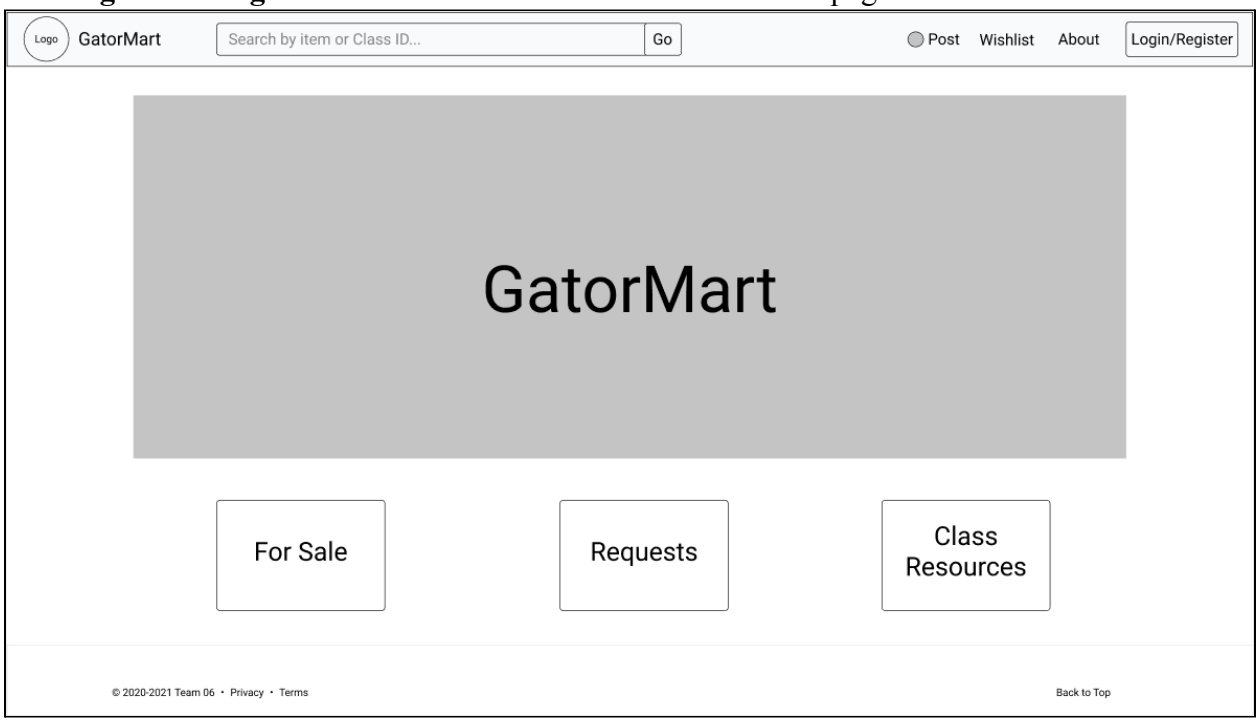
2. **Registered Users**

   2.6 Shall be able to rate a user utilizing the Gator Review system.

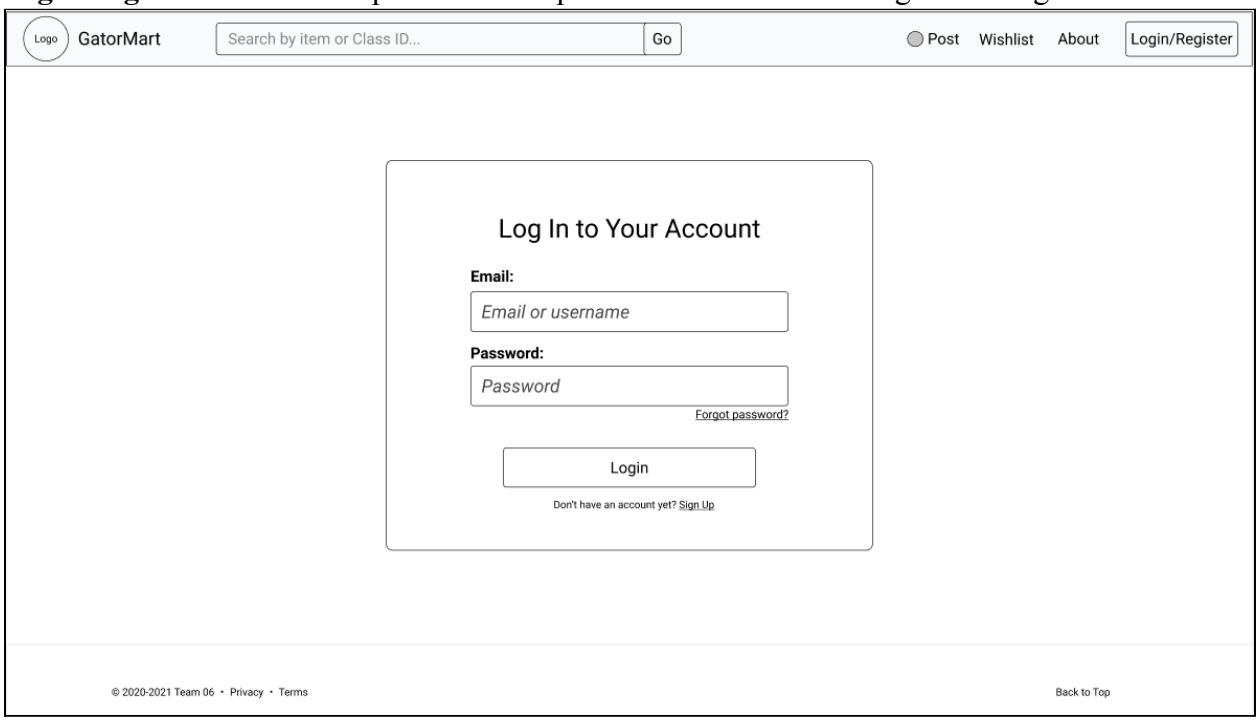
UI Mockups and Storyboards

Use Case 1: Register, Login, and Post
Dr. Mosk has just finished a faculty meeting discussing the emotional and financial impact on current San Francisco State (SFSU) Students as a result of the global pandemic and seasonal local wildfires. As a professor that cares for the wellbeing of his students, Dr. Mosk was able to get a textbook that was being offered for free by a reputable academic institution online approved by the Computer Engineering department for one of his courses. Knowing that the class occupancy was low due to textbook costs, Dr. Mosk decided to motivate future students to enroll in the course by making his cost-free materials known to the university community through GatorMart. He logs in to our website on his laptop and proceeds to create a post in GatorMart specific to his class ID but he is redirected to register or log in with an registered account before doing so. Upon successfully registering with his SFSU email and logging in, he creates a post with the appropriate description that directs SFSU students to download the free course textbook from the reputable educational website. Dr. Mosk fills out all of the fields in his post submission form and is able to submit his post for admin approval.

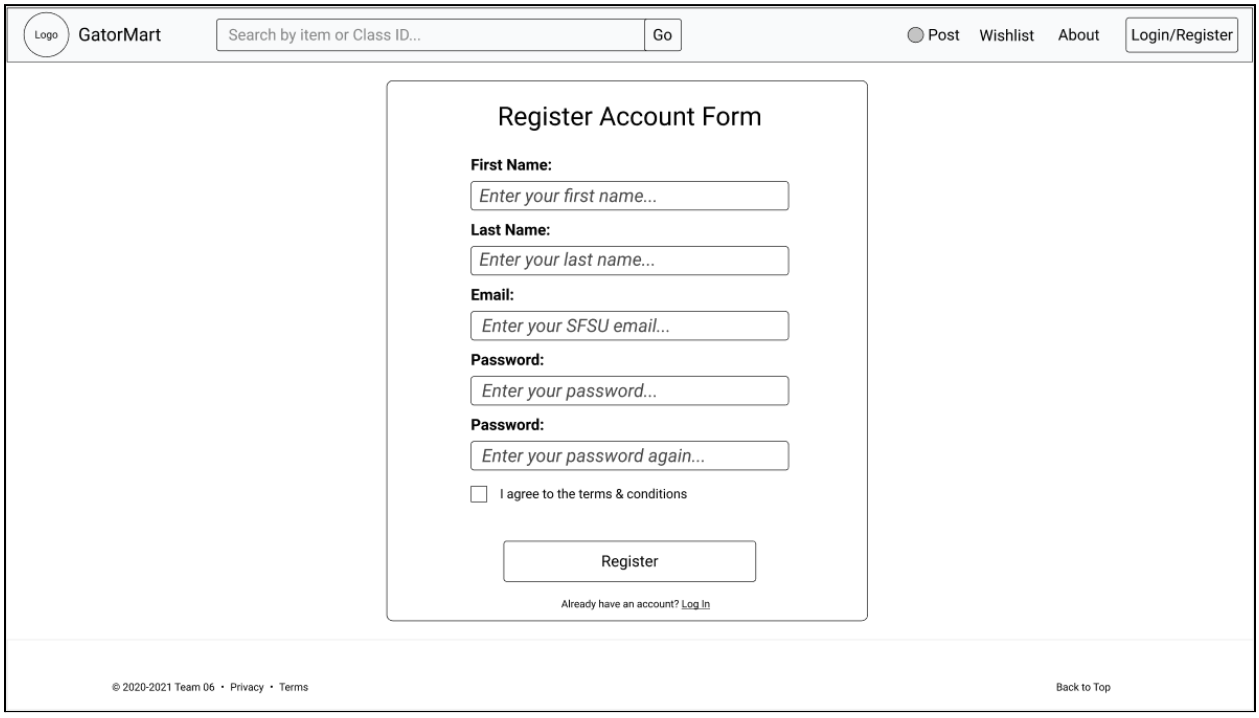
Landing/Home Page: Dr. Mosk travels to the GatorMart home page.



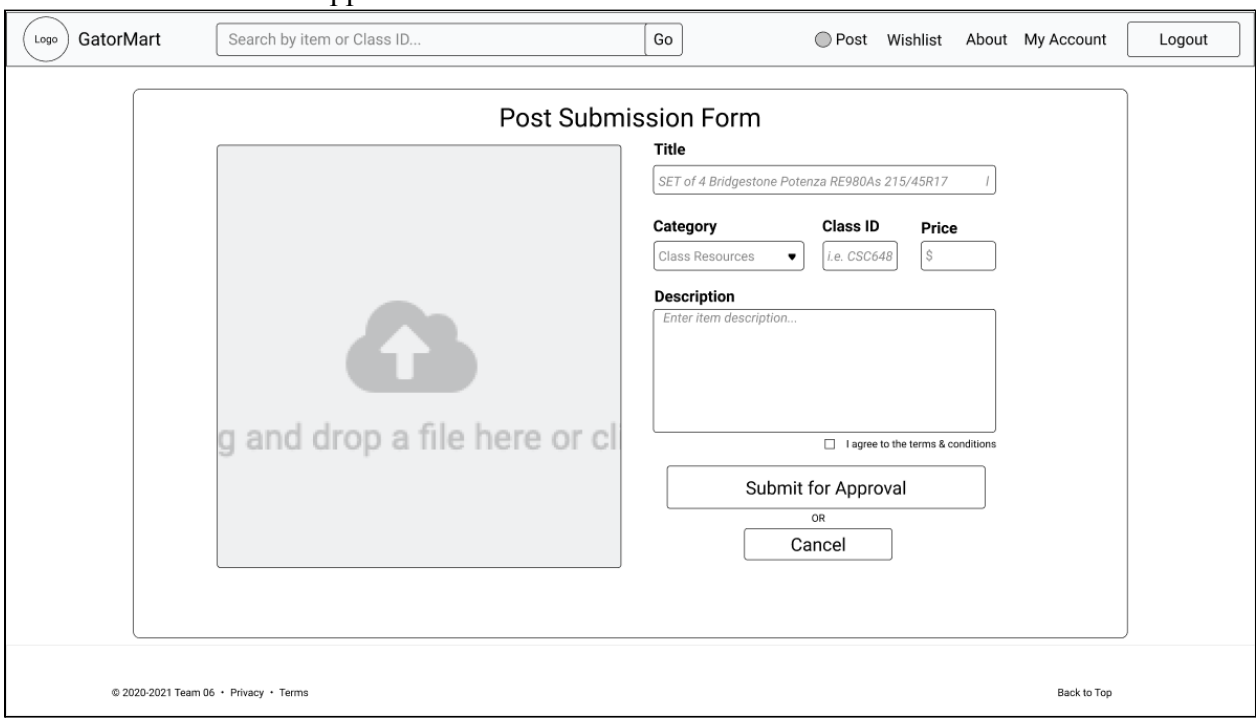
Login Page: Dr. Mosk attempts to create a post and is redirected to register or log in.



Registration Page: Dr. Mosk proceeds to register an account with his SFSU email.

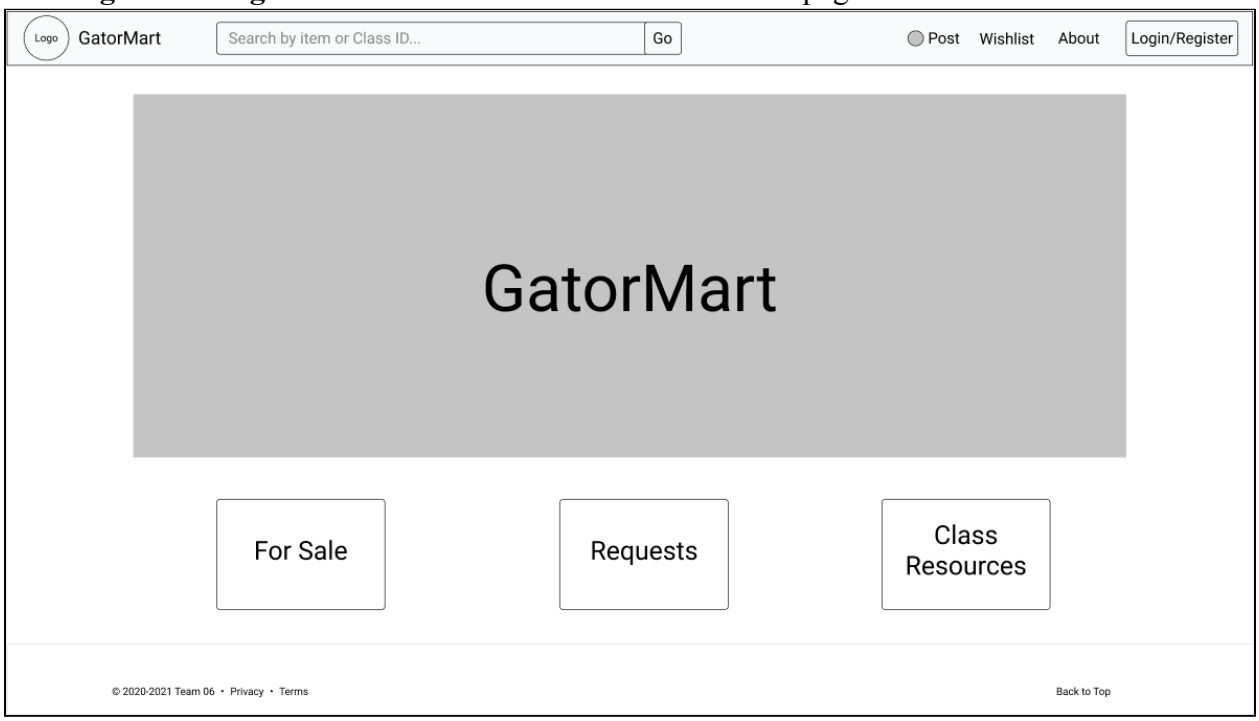


Posts Submission Page: Dr. Mosk clicks “Post” in the navbar then creates and submits a “class resource” for site admin approval.

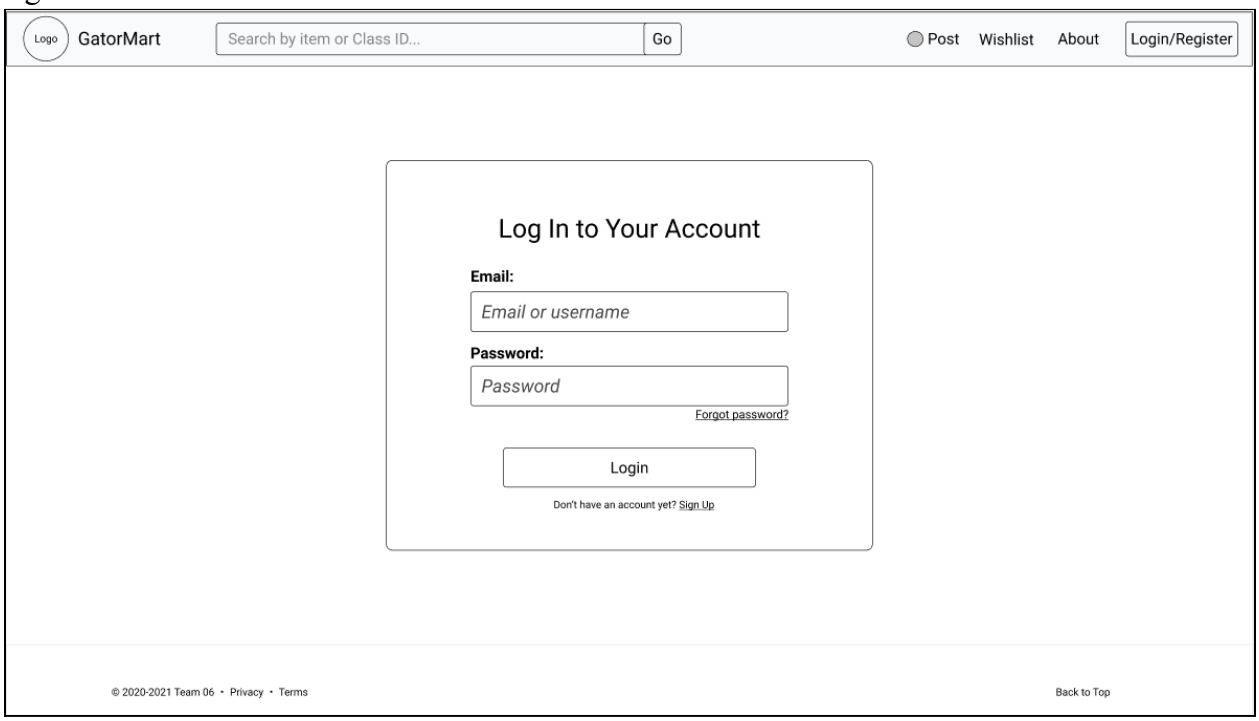


Use Case 2: Buying
Ricardo has just woken up late for his first day as a freshman at San Francisco State University (SFSU). He is excited for this day, but is also exhausted from finishing up an overnight shift the day prior at a local coffee shop and is now running late for school. As Ricardo is traveling on BART to SFSU, he realizes that he forgot to purchase the required textbook for his first class. He immediately goes to the GatorMart mobile site, logs in with his registered account, and seamlessly searches for the title of the book he needs. He notices that there is one “for sale” post created by a reliable registered user offering the textbook he required. Ricardo quickly utilizes the simple one-way messaging system from our site to inform the seller of his interest while providing his contact information for their scheduling convenience. Upon successful contact from the seller, both users set up a time to meet up at the university’s J. Paul Leonard Library which is on his way to his first class. Ricardo meets his seller at their public appointed place and time and safely purchases his textbook by paying his seller through an external money exchange application.

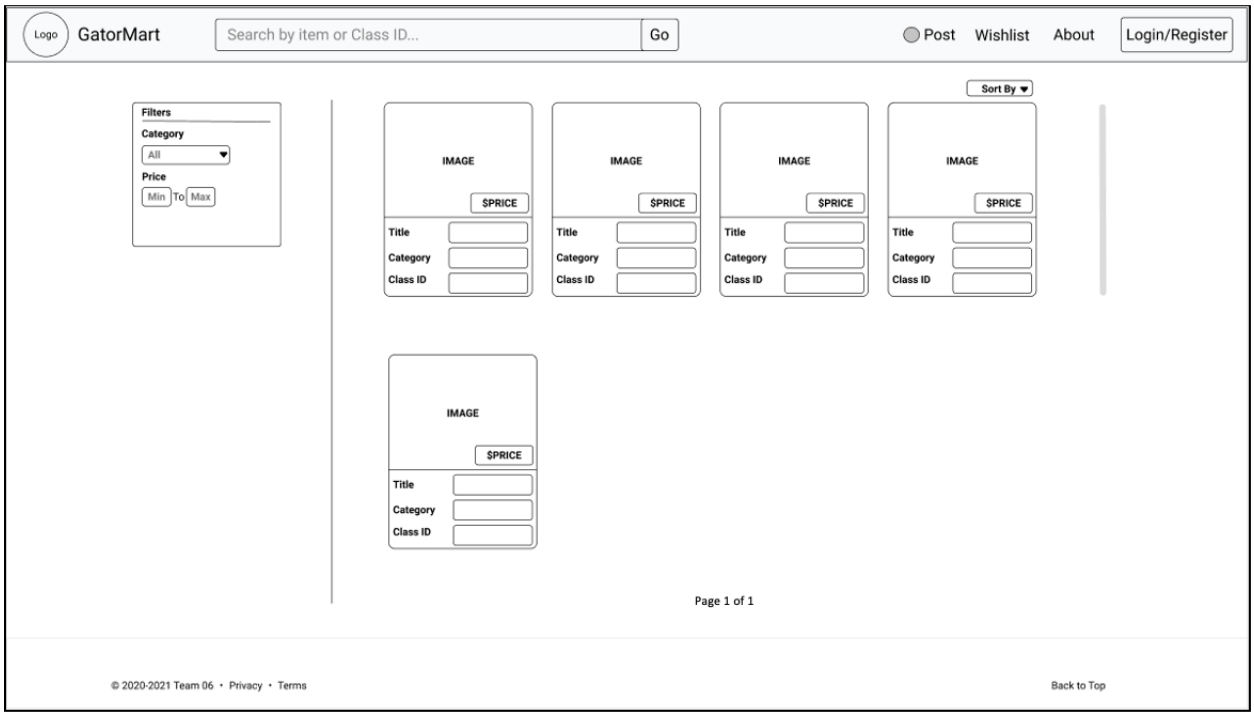
Landing/Home Page: Ricardo travels to the GatorMart home page.



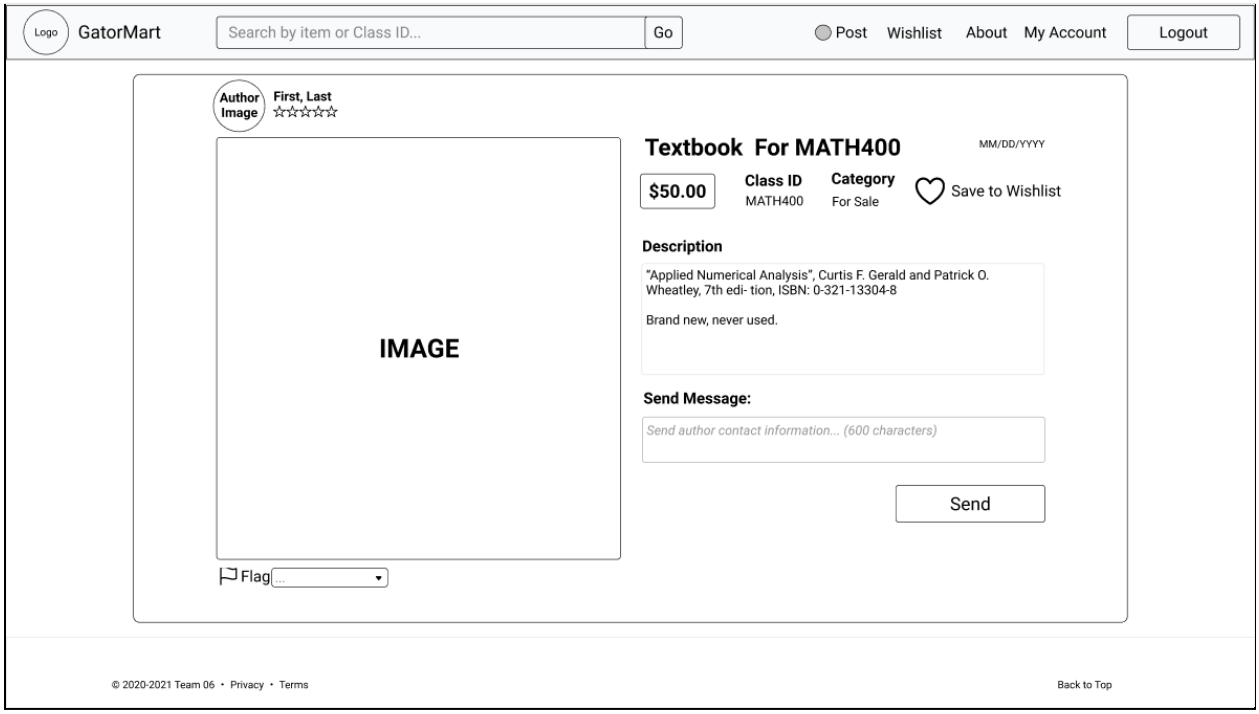
Login Page: Ricardo clicks the "Login/Register" button in the navbar and logs in with his registered account.



Search Page: Ricardo uses the search bar to search for his class specific textbook by filtering the results by class ID.

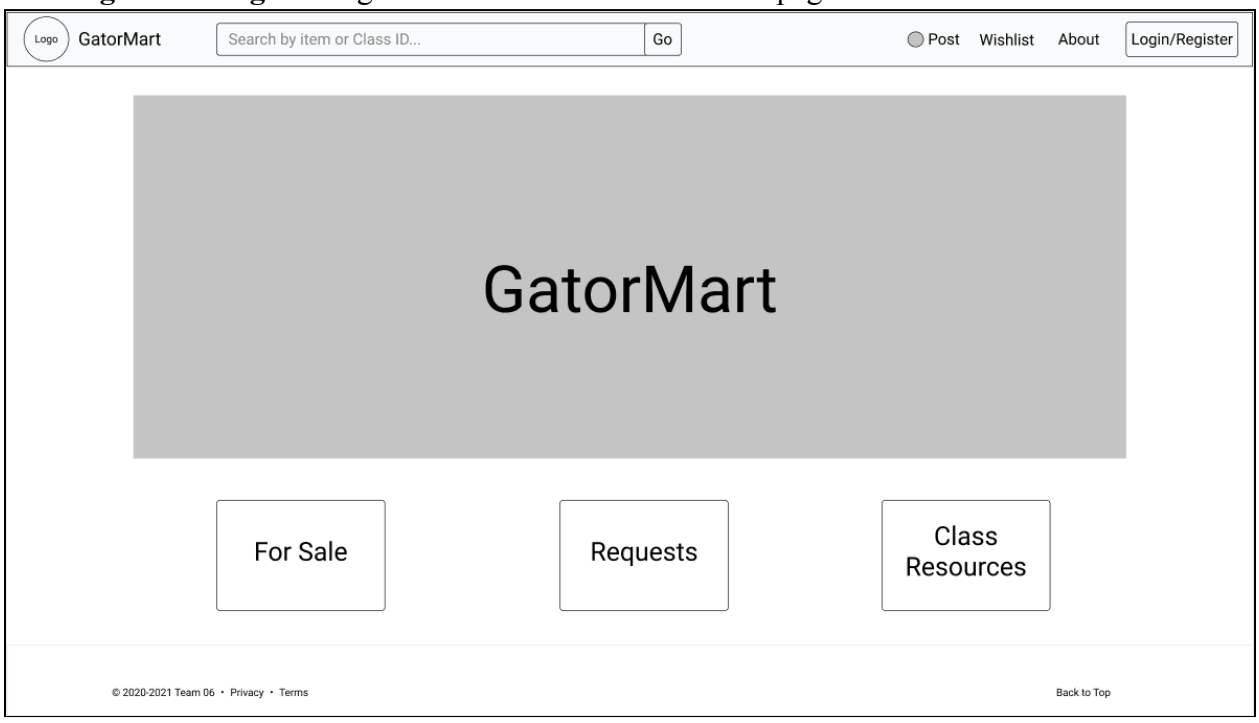


Item Page: Ricardo finds a post and sends a one-way message to the seller then waits for a reply.

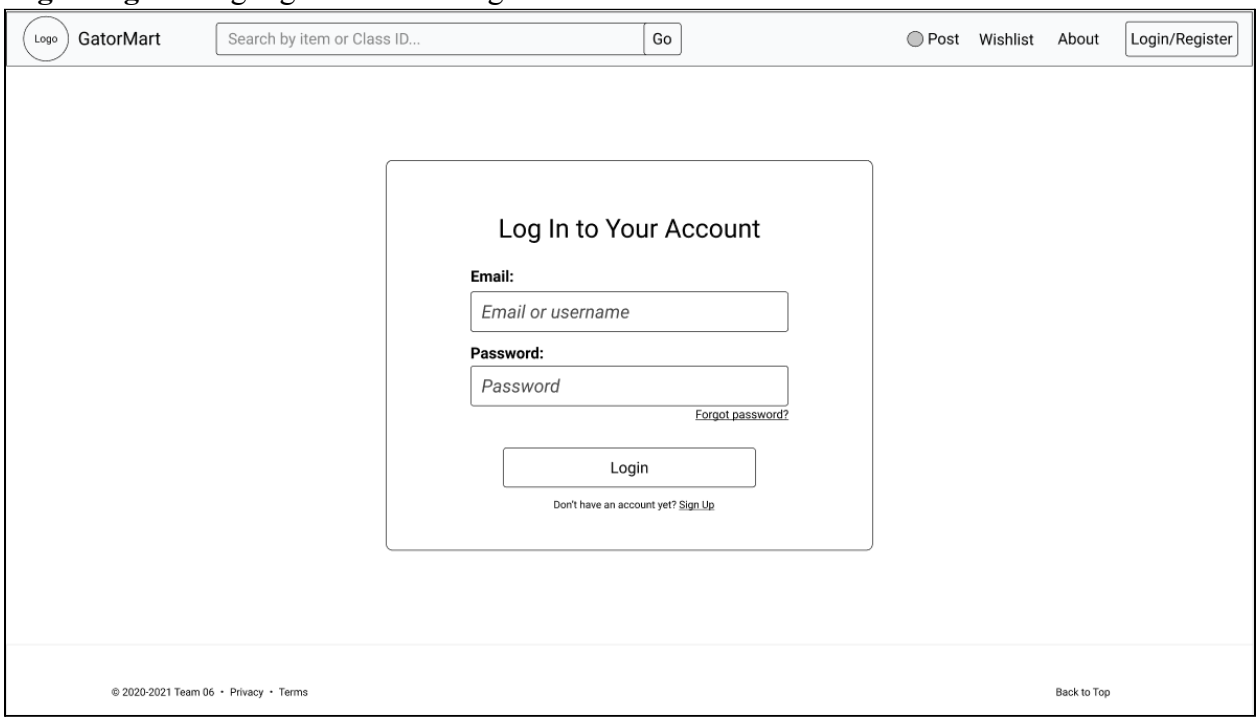


Use Case 3: Selling
Liang is an avid volunteer at her local nursing home and is currently preparing to relocate to Minnesota for a job opportunity. However, due to her constant volunteer roles, she does not have enough money to purchase a plane ticket to Minnesota. As she is cleaning out her room, Liang realizes that she cannot take multiple furniture pieces from her dorm. Liang decides to go onto the desktop GatorMart site and logs in with her already registered account. She creates a "for sale" post for one piece of furniture, attaches an image from her phone of the furniture, and annotates their respective price point in the post description. After Liang's post is approved by site admin, an incoming freshman at SFSU interested in the item for sale immediately sends a message to Liang with their contact information. Liang and the freshman exchange information through their mutual desired means of communication and meet up to purchase one item from Liang's post. After the sale, Liang returns to the mobile GatorMart site, and because she is already logged in, she goes directly to her post and marks it as "sold" to prevent the post from showing up on future searches.

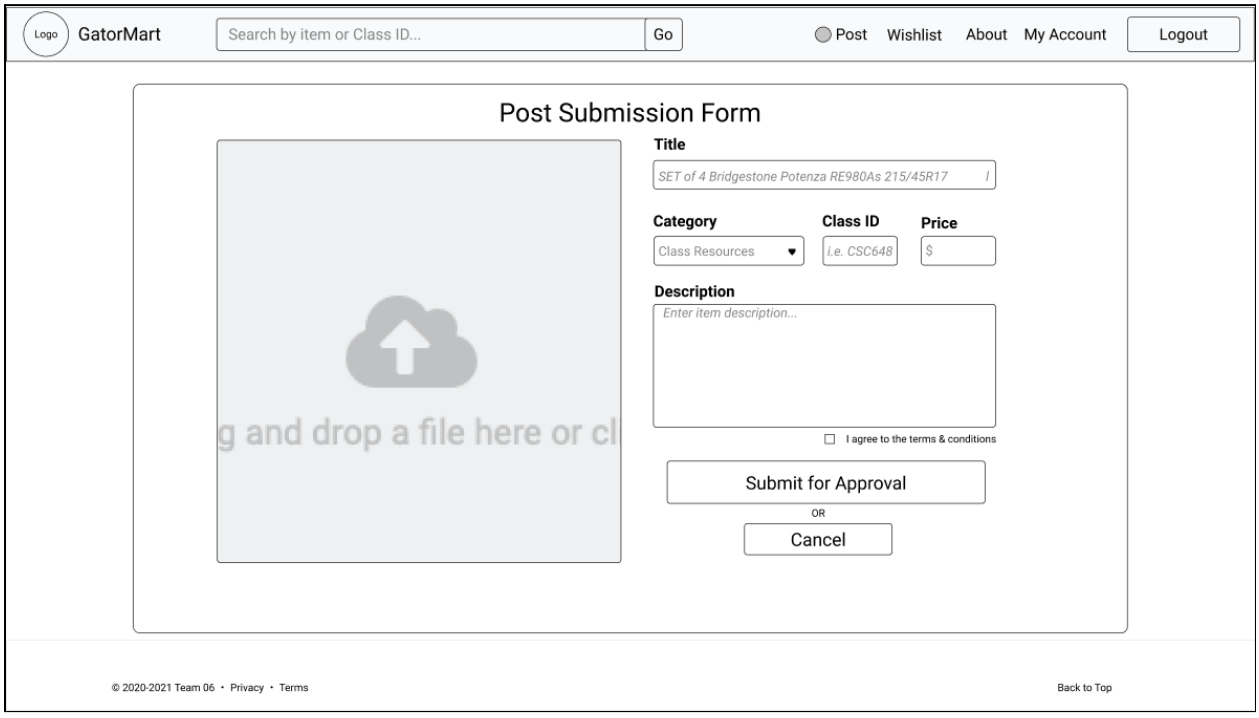
Landing/Home Page: Liang travels to the GatorMart home page.



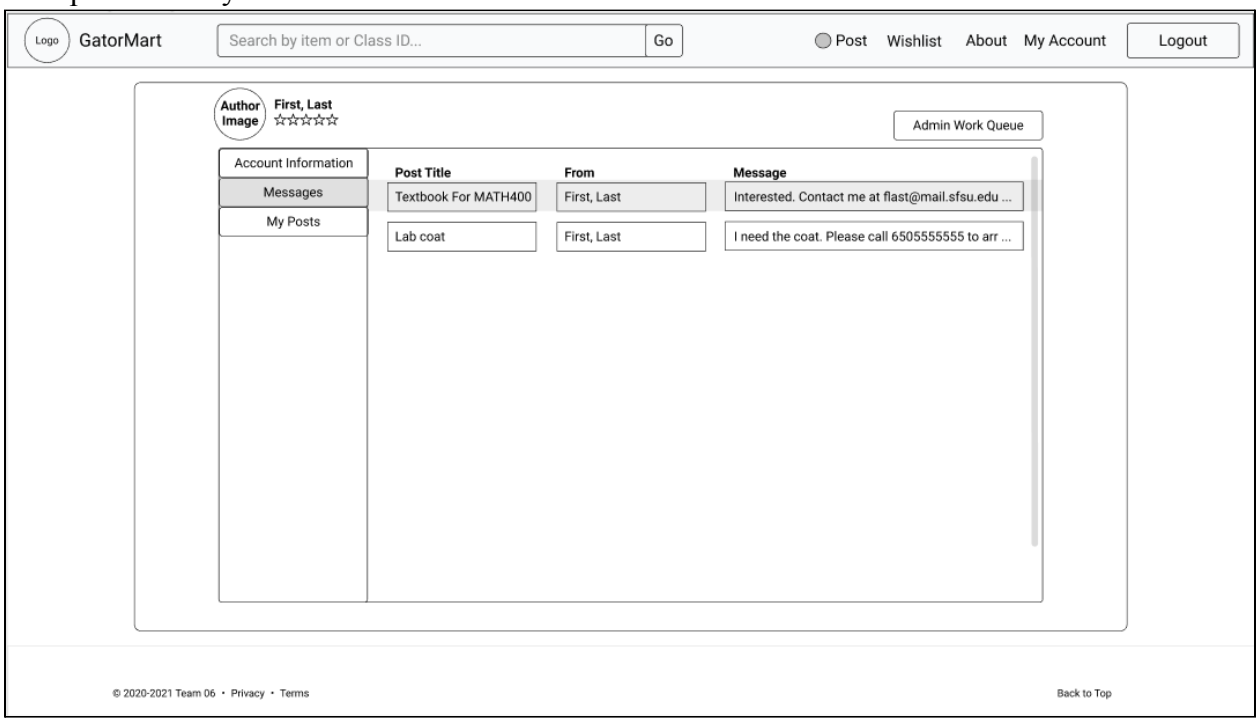
Login Page: Liang logs in with her registered account.



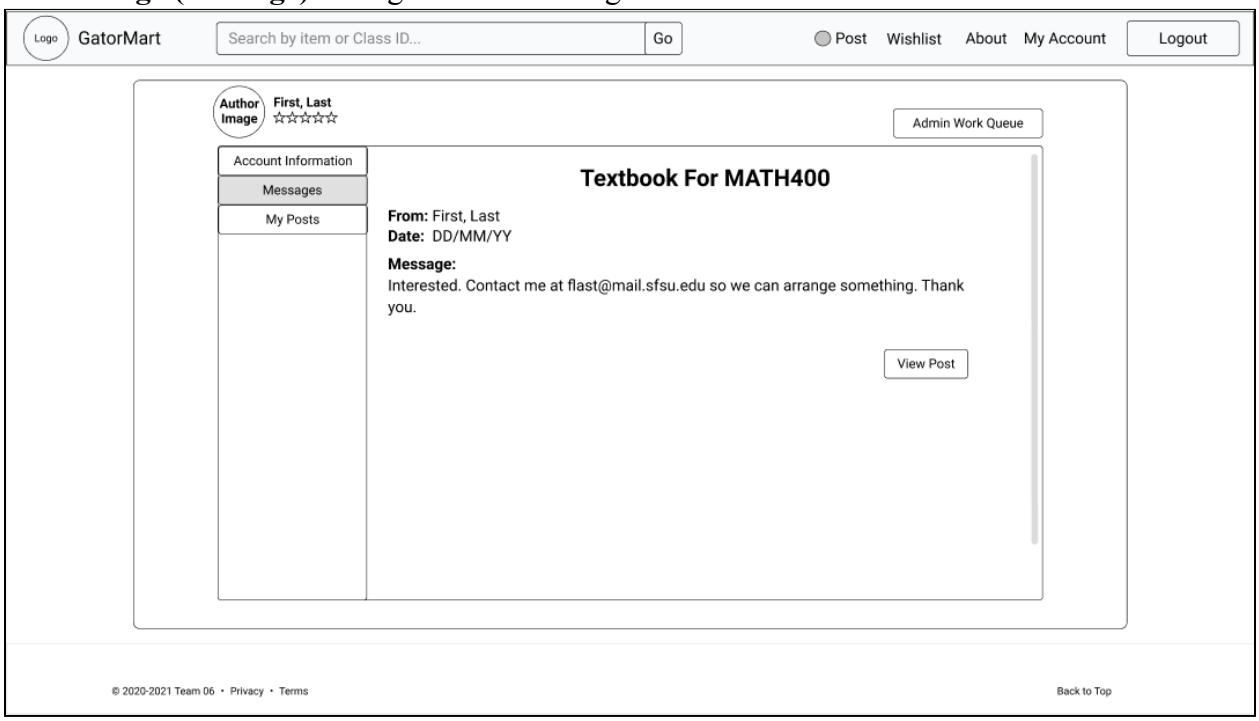
Posts Submission Page: Liang creates and submits a "for sale" post to site admin.



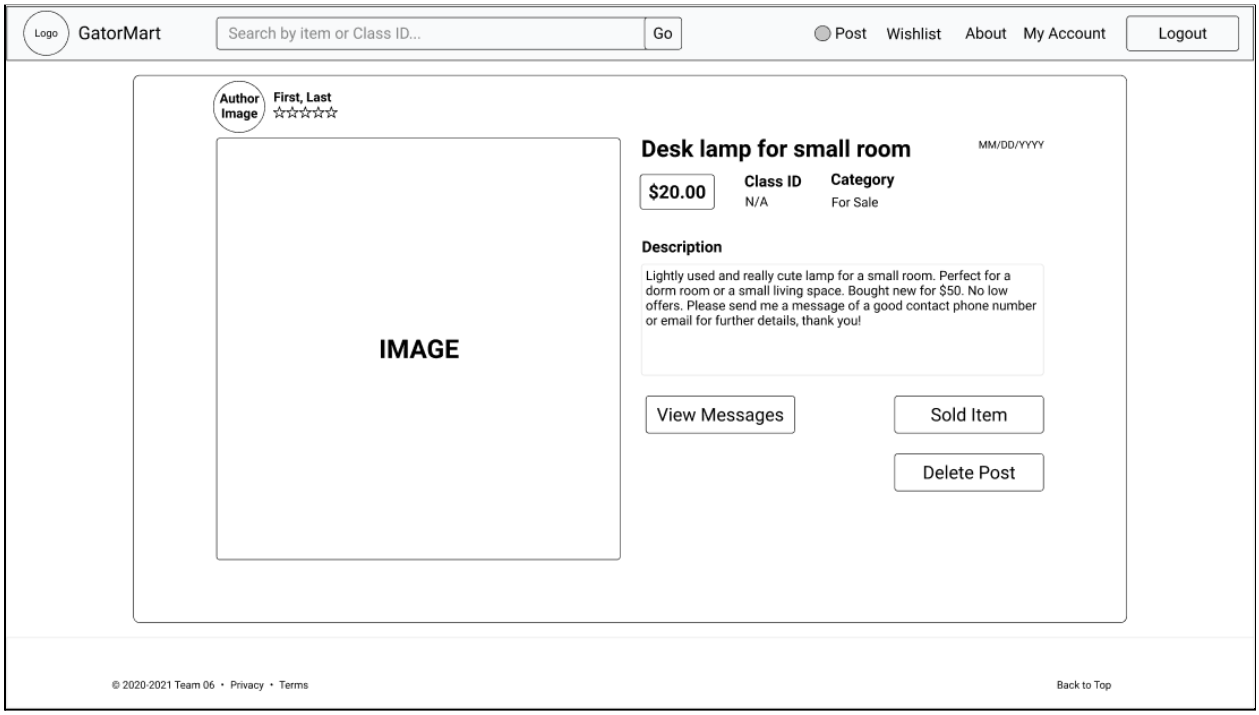
Profile Page (User Dashboard): Upon site admin approval of her post, Liang receives messages from potential buyers.



Profile Page (Message): Liang selects a message to read.

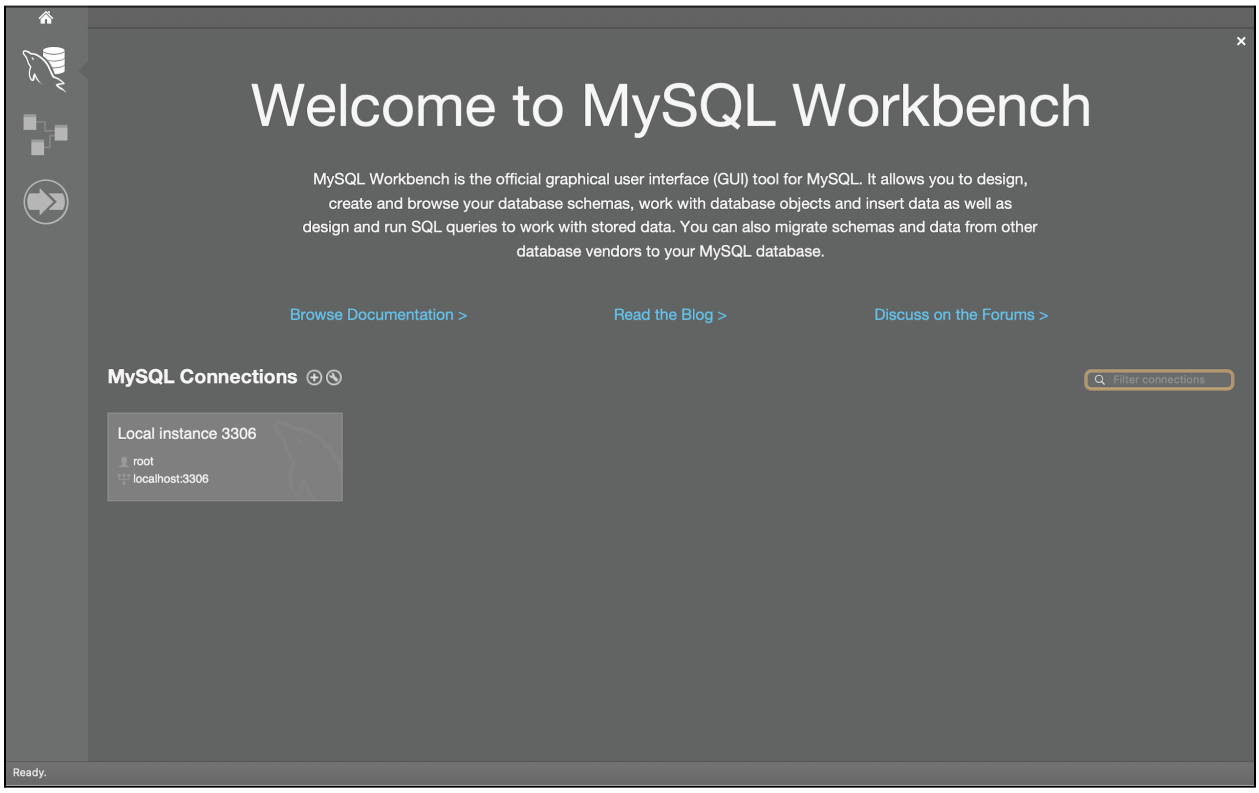


Item Page (Seller Point of View): Upon successful contact with a buyer and sale, Liang proceeds to mark her post as “sold”.



Use Case 4: Administering Posts
Nadia is trying to keep up with her last semester set of courses at San Francisco State University (SFSU) while concurrently reviewing an influx of post submissions by faculty, staff, and students for the GatorMart website. As an experienced programmer, she was hired by GatorMart due to her knowledge in MySQL Workbench which is the main tool used to administer the website. At the beginning of her shift at GatorMart, Nadia logs on to our website’s MySQLWorkbench and begins to search for posts pending approval and begins reviewing the content of each post. Nadia encounters a post with an inappropriate image and content. She denies the publication of this post and immediately deletes the seller’s account for going against GatorMart terms and conditions. The next post Nadia reviews has a buyer requesting materials for a specific course ID which abides by GatorMart’s terms and policies, she approves this post. Before finishing her shift, Nadia inspects posts “flagged” for inappropriate content by registered users. She encounters a highly reported post that did not comply with the terms and conditions to a certain degree. She deletes the post and informs the poster of their error.

MySQLWorkbench (Admin): Nadia will be using this tool to approve or deny posts, inspect flagged posts, and manage users of the site.



High-Level System Architecture and Database

DB Organization:

GatorMart users stored in SQL table labeled '**users**':

id: Each user will have an associated id number with their account for backend uses that is auto generated upon account registration.

firstname: When signing up for the site, users will have to provide thor first name to be displayed on their account alongside posts.

lastname: Upon account creation, users must provide their names to be displayed on accounts and posts.

email: A user's school email must be provided and verified upon registration to be able to use the site.

password: Users need to create a password that will allow them to log into their accounts to post on the site.

rating: There will be a seller rating implemented into the site and the score will be tied to the account.

usertype: This field will show what type of user the account is and can grant admin permissions if it is the correct value.

profilepic: When creating an account on the site, the user will submit a profile picture to identify themselves to other users. This field will have the file path to the image.

active: This will be a field that will determine if the user has been banned from the site or not.

GatorMart posts stored in SQL table labeled '**posts**':

id: When a post is created by a user, it will be automatically assigned an id number for backend purposes.

title: When the user creates a post on the site, they will title their post with relevant information that other users will search for.

description: Upon creation of the post, the user will provide extra information about the item they are posting about. Things like condition and other relevant material.

price: The post will require a price associated for the item being posted about.

type: This field will be an option that the user creating the post will select to determine what kind of item the post will be. The different options include For-sale, Request, or Class Resource.

category: This will allow for posts to be sorted by broad categories to narrow search results.

class: This will be a field for storing what class this item could potentially be useful for.

created: This will be the date associated with when the post was made for other users to determine relevance.

photopath: The user will be allowed to attach a picture to their post to display what the post is about. This will be the path to that image.
thumbnail: When in the main search page, users see a preview of the items being sold. The path to this lower resolution image will be stored here.
approved: The field will indicate if an admin has approved the post for display on the site.
fk_userid: When the user creates a post, their account id will be tied to the post to be displayed with the post and for backend purposes.

GatorMart messages stored in SQL table labeled '**messages**':
fk_sender: This field will store the user id of the person sending the message.
fk_receive: The field will store the user id of the person the message is being sent to.
created: This will be the date that the message will be sent on.
message: This will be the field that stores the message being sent between users.
fk_post: This will store the post id that the message will be about.

GatorMart sessions stored in SQL table labeled '**sessions**':
session_id: This will be the auto generated id for the session when a user logs in.
expires: This will be the value for when the session will no longer be valid.
data: This will store the cookie and information used to track the user session.

GatorMart flagged posts stored in SQL table labeled '**flags**':
id: This will be the identifier for the specific message
reason: The reason that the post was flagged, an option from a provided list
fk_postid: The post id of the post that was flagged to be used in the backend.
date: The date that the post was flagged.

Media Storage:
Data will be kept in a file system rather than in DB BLOBs. This means file paths will be in the database leading to the images to be served to the user. There won't be any video, audio, or other form of media needed to be stored for the site.

Search/filter:
The search will work by a user submitting text to be compared to titles of posts on the site. This will be done with the SQL %like statement to try and find the closest item that the user is looking for. For filters, these will also be accomplished by using SQL statements that search for posts with desired fields.

Key Risks

1. **Skill Risks**
   Some team members have little to no experience in web development, and are concurrently taking CSC 317 for the first time this semester. This creates a skill risk, as some knowledge that is assumed may be missing or may be learned later than would be convenient for the project. To address this risk, we will look to anticipate what missing knowledge will be necessary to have, so that those lacking it can teach themselves as necessary.

2. **Schedule Risks**
   Many team members have very busy schedules and only have limited time to devote to the project. In order to manage this risk, we will keep each other informed via slack and during our team meetings about times when we are extra busy or when we have more time, so that we can plan accordingly to make sure things can get done on time.

3. **Technical Risks**
   Among our planned features is a wishlist that can be populated even by anonymous users, but that can then be stored in the database when a user registers. This presents a technical risk, as the implementation of this feature could be complicated. A further technical risk is that we would prefer to have a drag and drop style file upload for images, but are unsure as to how to implement that within our framework.

4. **Teamwork Risks**
   With any team it is possible for the members to have a difference of opinion that can lead to friction or even resentment. To mitigate this risk, it is important to ensure that every team member gets an opportunity to voice their opinions, and is listened to with respect and willingness to compromise.

Project Management

Our team has set up a slack channel for asynchronous communication between team members, and created a Trello board to keep track of which tasks are left to do, and which tasks are being worked on, and who is assigned to each task. We hold regular Zoom meetings on Thursdays in addition to the one on Wednesday during class. During these meetings we discuss what we want the project to look like, and what tasks need to get done by what deadline. We also use the meetings to work on tasks that need the whole team’s input, such as finalizing milestone documents, and reviewing each other's work.

Milestone 3 review summary and plans - CSC 648-848 Fall 2020 Team 6

Team 6

Date of Milestone 3 review with Prof. D. Petkovic: 11/16/2020

Date of this summary and plan document: 11/19/2020

Summary of feedback and tasks to do

General Notes
- Put posts and requests under separate tabs
- Keep in mind how how much work the users have to do
- Get website onto deployment server
- Include Item requests in pull down bar and have it open a new page.
- Pictures in a post should be optional in the scenario of “Item Request”
- Header comments on code, who did what and what is this for?
- Try to separate files for project, frontend and backend should be separated.
- UI is okay, need to clean up home page
- Biggest issue is request, it is complicated to him, don’t touch for now until we can get it to run on deployment server
- Put Warning banner above the search bar
- Combine everything in the search bar, maybe remove the whole price filter since price is optional?

Index
- Category pull down smaller, it is too wide
- Missing what the site is about on the home page, 3 lines at the top, why shop with GatorMart?
- Call recently posted or recently sold, favor of him moving items in carousel, not us.
- Doesn’t like Item Request and Class Resources, might be too much effort to implement
- Pull down on index should be books, electronics, and whatnot, shows all the available items
- Have a separate page with item requests page?
- Remove for sale, item request could be an option under search
- Class resource should be dropped, too much work to admin/manage
- Item request is now a P2

Post
- Post should be condensed into one column, post image should be at the bottom.
- Mark mandatory fields
- Class ID should pop up

- Text next to submit for approval “Might take 24hrs to approve”

Dashboard
- Make sure “View Posts” is opened into a new tab
- Should have a delete next to “View Posts”

Login
- Underline “Sign Up”

Registration
- Show “All fields mandatory”?

Search
- Sort and filter is too much, sorting should be good enough?
- Get rid of sorting
- Details for posts look perfect
- Psyche, make clicking onto a post a new tab
- Need to also align cards so they’re all the same size
- “Showing 7 results” instead of “7 results”

Database
- Have a category table, not hard coded
- Password needs to be encrypted for users table

GitHub
- Be more specific for commit messages in GitHub

CHANGES TO BE MADE(24hr)
- Move alert to the top of Navbar (check)
- Delete button next to posts
- Text next to submit button “May take up to 24hr”
- Purple rectangle on index (check)
- Pull down too wide
- Purple square above recent items
- Actually implement recent items
- Take up to 24 hours to approve
- Need delete button
- Category its own sql table
- Password encrypted

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

Implement login and registration, connecting with the backend.
Connect user dashboard with backend.
Save posted items to the database.
Get categories from sql table.

List of final product P1 functions agreed at the meeting

1. **Anonymous Users**
   1.1 Shall be able to browse and search for items for sale. (They shall have full search available to them as any other user).
   1.3 Shall be able to register ONLY with an email ending in sfsu.edu.
   1.4 Shall be able to browse the website utilizing categorized filters.

2. **Registered Users**
   2.1 Shall inherit all functionality allowed for anonymous users.
   2.2 Shall be able to post items for sale/share or request items, and such posts will be reviewed by the website administrator.
   2.3 Shall be able to privately message a user for an item they are interested in.
   2.4 Shall be able to log out of their account.

3. **Administrator**
   3.1 Shall inherit all functionality allowed for registered users.
   3.2 Shall be required to approve or deny post submissions from users.
   3.3 Shall be able to remove users.

SW Engineering CSC648/848 Fall 2020

GatorMart

Team 6

Lothar Narins Team Lead

Aaron Colmenares Backend Lead

Allyson Leung Frontend Lead

Wilson Young GitHub Master

Saloni Bhatia Backend

Kevin Nunura Frontend

lnarins@mail.sfsu.edu

Milestone 4

December 2, 2020

Date Submitted: December 8, 2020

Date Revised: December 10, 2020

Table of Contents

Product Summary

Product Name: GatorMart

Major Committed Functions: (In Progress)

- All users shall be able to browse items for sale.
- All users shall be able to register with their unique SFSU email address.
- All users shall be able to log in with their registered account to message other sellers.
- All users shall be able to select an item for sale to read more information on the product.
- All users shall be able to browse the website utilizing categorized filters within the search bar.
- Registered users inherit all functionality allowed for unregistered users.
- Registered users shall be able to see and read their messages through their user dashboard.
- Registered users shall be able to privately message a user for an item they are interested in.
- Registered users shall be able to post items for sale or request specific products, and such posts will be reviewed by the website administrator.
- Registered users shall be able to log out of their account.
- The website administrator shall inherit all functionality allowed for non-registered and registered users.
- Administrators shall be required to use MySQLWorkbench to manage the website.
- Administrators shall be able to approve or deny post submissions from registered users.

URL: http://3.134.106.28:3000

Usability Test Plan

Test Objective

The main objective of this usability test is to assess and evaluate the usability of our website GatorMart. The major function that is being tested for usability is our search function. We are choosing the search function because we feel it is a major function that is crucial for navigating our website. The usability test is used to determine if our website is user friendly and fast. The goal of this test is to evaluate the user's ability to search for items using the search function and to filter their results through our category filter. We are expecting feedback from testing to improve our website's user friendliness.

Test Background and Setup

The test is designed to evaluate GatorMart's ability to direct users to their desired product using the search function. The website must be able to run in the most commonly used browsers, such as Firefox, Internet Explorer, and Google Chrome. The intended users are primarily San Francisco State University students and staff or faculty.

To start the usability test, we first open up Google Chrome and navigate to the homepage of GatorMart. We then give the user a list of tasks to complete without giving them any information on how to accomplish the task. After the user completes the tasks, we will use the criterias in the Usability Task Description table and the Likert test to determine if the function passed the test.

The URL of the website is http://3.134.106.28:3000/ which will take you to the homepage of the website. The user can input the valid search query into the search bar which includes the class ID or the title of the post. Users can also filter through results by the category of the item utilizing the category drop down list or sort the items by price. The usability test is successful if the user is able to use the search function to find the specified item by searching the proper term in the search bar. When the correct items load onto the page according to the user's search criteria, we can consider the test a success.

Usability Test Description

Task	Description
Task	Search for items in MATH325
Machine State	Search listings loaded
Successful Completion Criteria	Shows related search results according to the search query
Benchmark	Complete within 12 seconds

Effectiveness : We would measure effectiveness by looking at the percentage of users who completed the task within the time limit. We would also take into account any errors that the user encountered while navigating to the search bar and entering their query, while also listening to the feedback given.

Efficiency : We would measure efficiency by the average time it took for users who completed the test and users who attempted the test while eliminating outliers. Efficiency could also be measured by the amount of clicks it took to complete the task.

("**X**" applicable column)

	Strongly Disagree	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
I was able to find the search bar easily.					
I was able to search for the specified item.					
I found a related search result.					

Comments and Feedback (Optional):

QA Test Plan

Test objectives:
Assessing if the search function performs per specs.
HW and SW setup:
URL: http://3.134.106.28:3000/
Feature to be tested:
Search function

QA Test Plan:
Browsers used in tests: Chrome and Firefox

Test #	Test Title	Description	Input	Expected output	Chrome	Firefox
1	searching by class id	Test whether search by class id works	Enter “MATH325” into search bar and press search	Should show 1 result: “Linear Algebra Textbook”	PASS	PASS
2	searching by category	using category dropdown menu	Select “Supplies” in dropdown menu next to search bar, leave search bar empty, press search	Should show 2 results: “50 index cards”, “Slides for microscope”	PASS	PASS
3	searching by title/ item name	Test if text field search works for title/item name	Enter “book” into search bar and press search	Should show 2 results: “Howl on Trial Book”, “Linear Algebra Textbook”	PASS	PASS

Code Review

For the code review, the marketplace.html page was sent to another team member. This page has the code for the feature used for the QA and usability test, which is the search function. The Html file includes not only the structure for the entire page but also the javascript code for the search function itself. Comments explaining the code are embedded and below are the comments made by the team member who reviewed the file.

Note: The peer reviewer marked their comments with a “@Todo” prefix

```
<body class="body-color">
    <div class="border alert-warning d-flex justify-content-center" role="alert" style="font-weight: bold; padding-bottom
        SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
    </div>
    @TODO(Lothar): Fix spelling on [background theme]
    <!-- [navbar] [theme] [backrgound theme]-->
    <nav class="navbar navbar-expand-sm navbar-color">
        <!-- Logo -->
        <a class="navbar-brand" href="/">
            @TODO(Lothar): This old code could probably go
            <!-- <i class="fa fa-cube wid" aria-hidden="true"></i>
            GatorMart -->
            <!--Below is for GatorMart actual logo when available-->
            <!-- <img src="/docs/4.0/assets/brand/bootstrap-solid.svg" width="30" height="30"
                class="d-inline-block align-top" alt="">
            GatorMart -->
            <img src="../Images/Logo/gatormart_logo.svg" height="40" class="d-inline-block align-top" alt="GatorMart" />
        </a>
```

```
        <div class="collapse navbar-collapse" id="navbarMenu">
            @TODO(Lothar): Make the categories come from the database as in the other pages
            <!-- Search bar code -->
            <form class="mx-5 d-inline w-100" id="search-parts" method="GET" action="/HTML/marketplace.html">
                <div class="input-group">
                    <select class="form-control search-slt col-sm-2" id="category" name="category" onchange="filteredSearch();">
                        <option value="All">All</option>
                        <option value="Books">Books</option>
                        <option value="Furniture">Furniture</option>
                        <option value="Supplies">Supplies</option>
                        <option value="Textbooks">Textbooks</option>
                        <option value="Tutoring">Tutoring</option>
                        <option value="Item Request">Item Request</option>
                    </select>
```

```
            @TODO(Lothar): This is fine to remove. It is unlikely to come back, and we can dig it out of old commits on GitHub if necessary for reference.
            <!--Filters COL div-->
            <!--<div class="col-sm-2">
                <div class="border filters-card shadow-sm form-color">
                    <h6>Filters</h6>
                    <hr class="featurette-divider my-0">
                    <div class="form-horizontal my-2">
                        <div class="form-group" id="search-type">
                            <label for="typeLabel">Category</label>
                            <select class="form-control" id="category" name="category" onchange="filteredSearch();">
                                <option value="All">All</option>
                                <option value="Books">Books</option>
                                <option value="Furniture">Furniture</option>
                                <option value="Supplies">Supplies</option>
                                <option value="Textbooks">Textbooks</option>
                                <option value="Tutoring">Tutoring</option>
                            </select>
                        </div>
                        <div class="form-group">
                            <label for="classLabel">Class ID</label>
                            <input type="text" class="form-control" id="classLabel" onchange="filteredSearch();" placeholder="e.g. CSC648">
                        </div>
                    </div>
                </div>
            </div>-->
            <!--Results COL div-->
```

```
@TODO(Lothar): Remove old code
<!--<button class="border btn-sm btn-light dropdown-toggle float-right" id="sortby" type="button"
    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">Sort By</button>-->
```

```
@TODO(Lothar): This function could have a more specific name, e.g. parseURLQuery
//This function processes the important information that is grabbed from the URL and then calls actual search function
function parse() {
    var url = document.location.href,
        params = url.split('?')[1].split('&'),
        data = {}, tmp;
    for (var i = 0, l = params.length; i < l; i++) {
        tmp = params[i].split('=');
        data[tmp[0]] = tmp[1];
        tmp[1] = tmp[1].replace(/\+/g, " ");
        //console.log(tmp[0]);
        setStuff(tmp[0], tmp[1]);
    }
    filteredSearch();
}
```

```
@TODO(Lothar): This function could have a better name, and why does it have an if statement if it does the same thing in both cases? Could be cleaned up.
//This function is used to set values to their correct elements, can be expanded if more filters are needed
function setStuff(name, val) {
    if (name == "category") {
        document.getElementById("category").value = val;
    }
    else {
        var holder = document.getElementById(name);
        holder.value = val;
    }
}
```

```
@TODO(Lothar): Feel free to remove the commented out declarations of type and classID. Perhaps display some error message to the user on error, instead of just console.log?
/*
    This function processes all of the information for the search
    it grabs info from the page and creates the result text that is displayed with the results
*/
function filteredSearch() {
    let searchTerm = document.getElementById('searchtext').value || '__NO_VALUE__';
    let category = document.getElementById('category').value;
    // let type = document.getElementById('typeSelect').value;
    // let classID = document.getElementById('classLabel').value || '__NO_VALUE__';
    let searchOrder = document.getElementById('sortBy').value;
    // console.log(category + ', ' + type + ', ' + classID + '.');
```

Self-check on Best Practices for Security

Major Assets	Threats	Protection
User records	Unauthorized user gains access to confidential data	1. Require users to authenticate themselves 2. Track system usage 3. Needs access to the database by connecting it to the server by SSH 4. Requires input validation
Posts	Unauthorized user makes feature unavailable, scrambles post information or posts inappropriate content	1. Require users to authenticate themselves 2. Track system usage 3. Needs access to the database by connecting it to the server by SSH 4. Requires input validation
Messages	Unauthorized user makes feature unavailable, distorts messages and can also gain access to personal information shared between buyer and seller	1. Require users to authenticate themselves 2. Track system usage 3. Needs access to the database by connecting it to the server by SSH 4. Requires input validation
Images	Unauthorized user makes feature unavailable, or can post incorrect or inappropriate images	1. Require users to authenticate themselves 2. Track system usage 3. Needs access to the database by connecting it to the server by SSH 4. Requires input validation 5. File system inaccessible without credentials

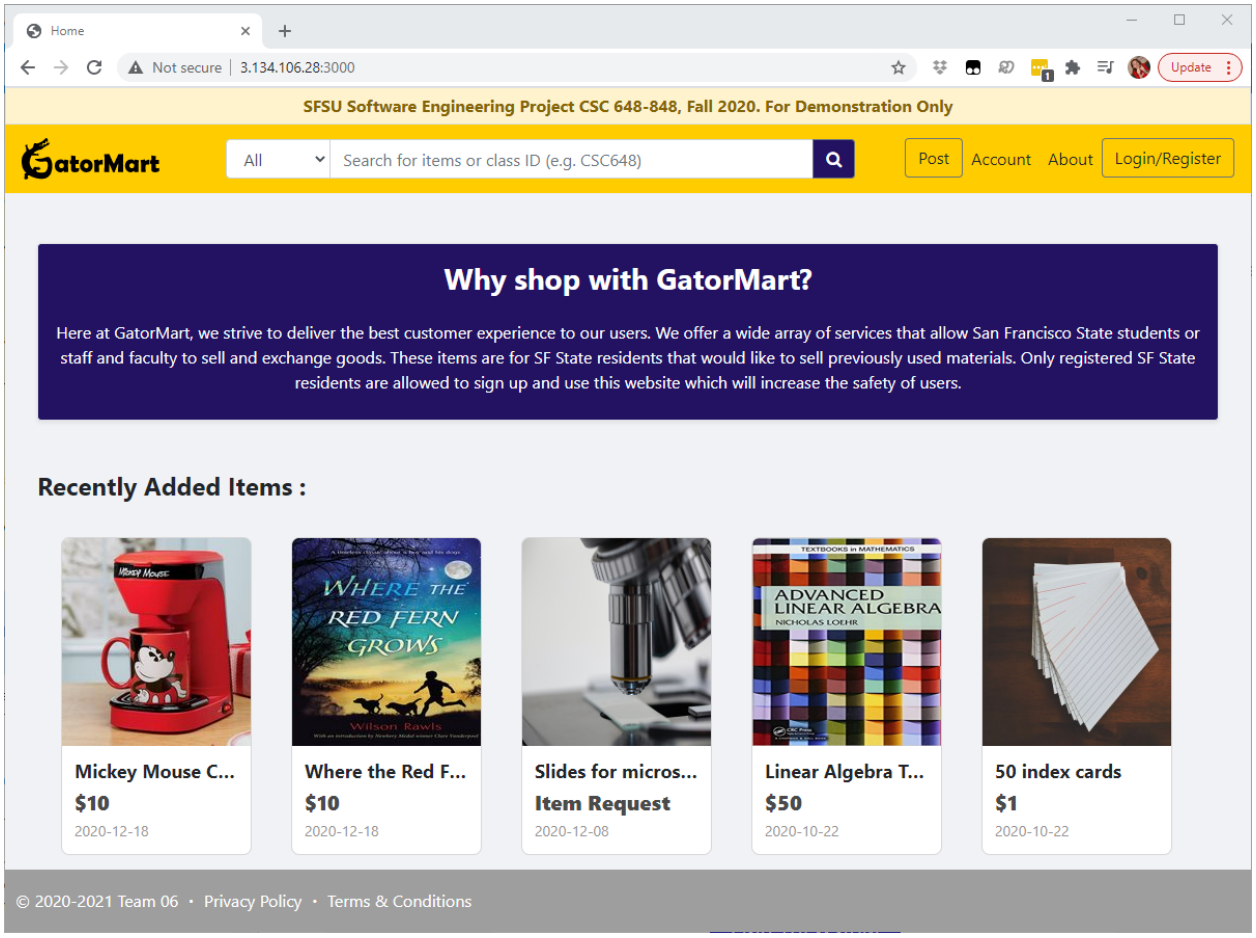
PW encryption in the DB: DONE

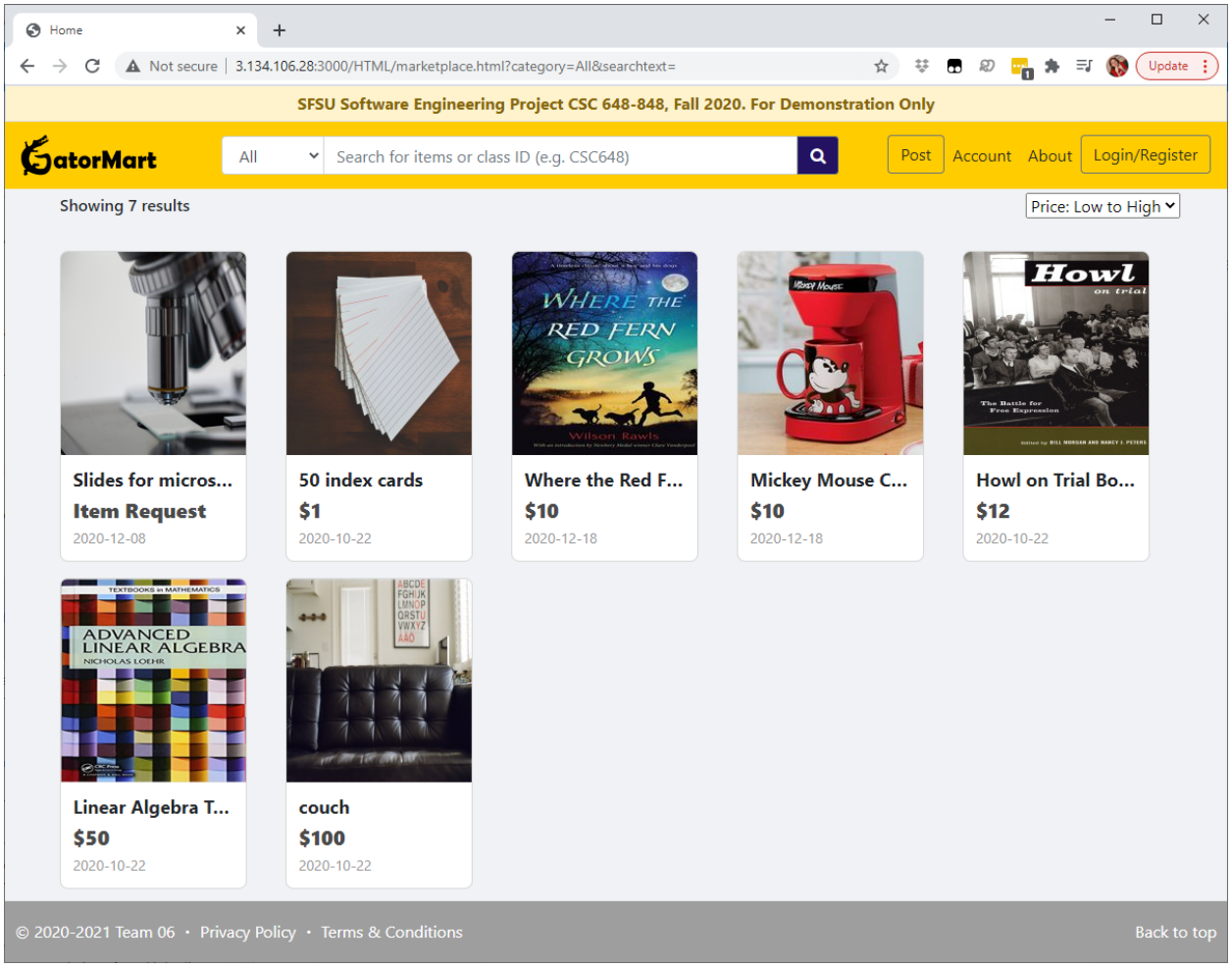
Confirm Input data validation; ON TRACK

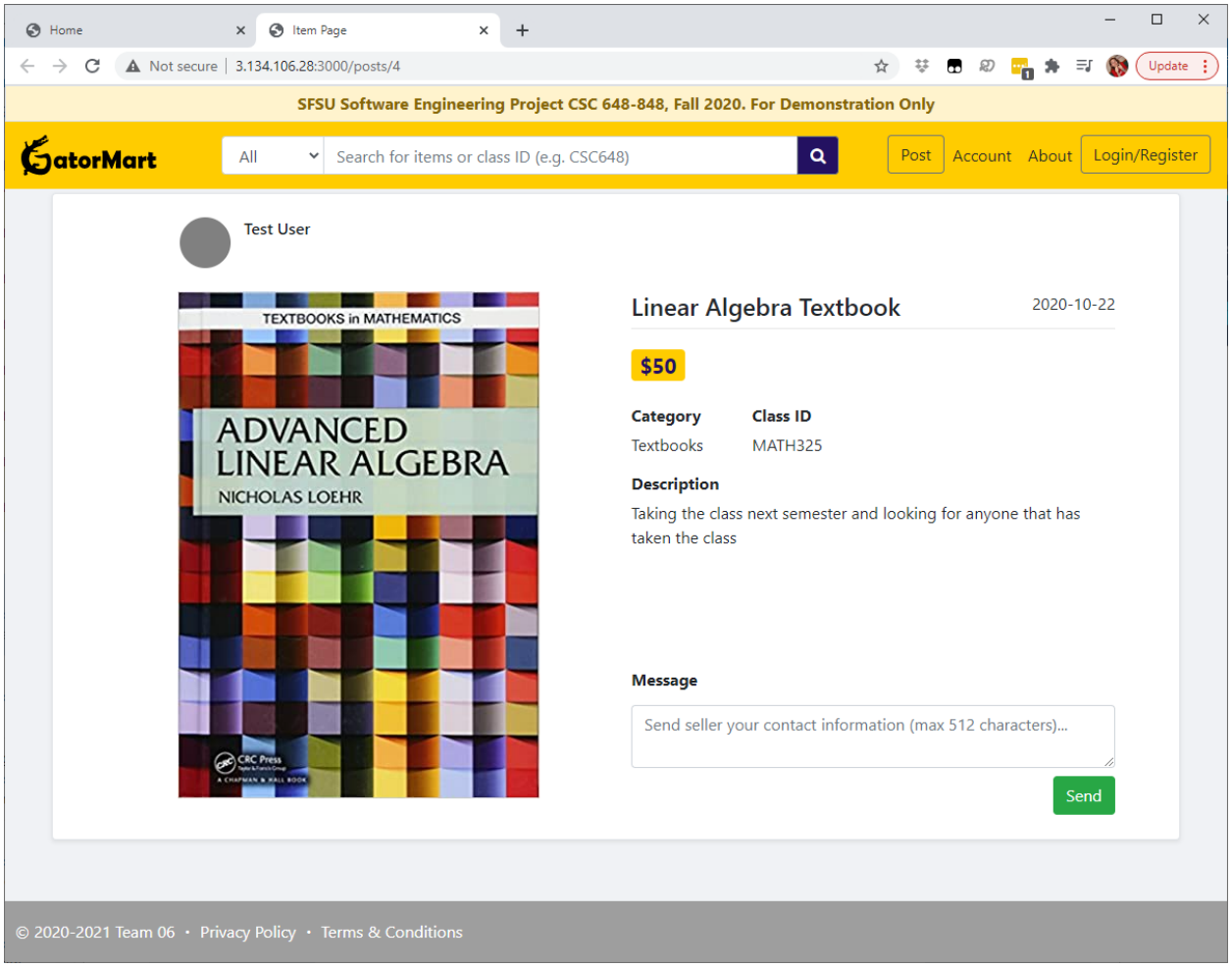
Self-check: Adherence to original Non-functional specs

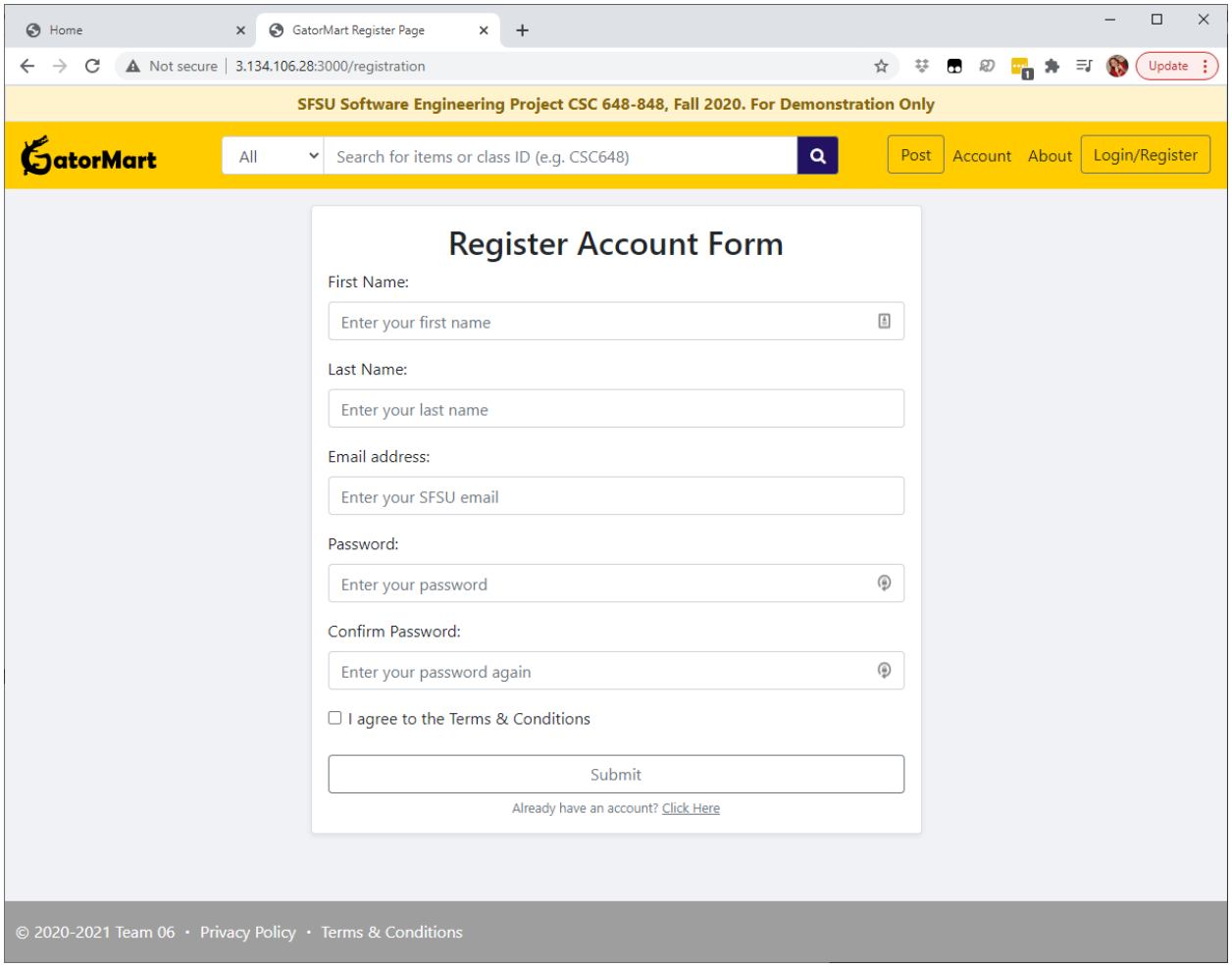
1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO). DONE
2. Application shall be optimized for standard 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 DONE
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 DONE
9. Application should follow established architecture patterns DONE
10. Application code and its repository shall be easy to inspect and maintain DONE
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 ON TRACK
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). DONE

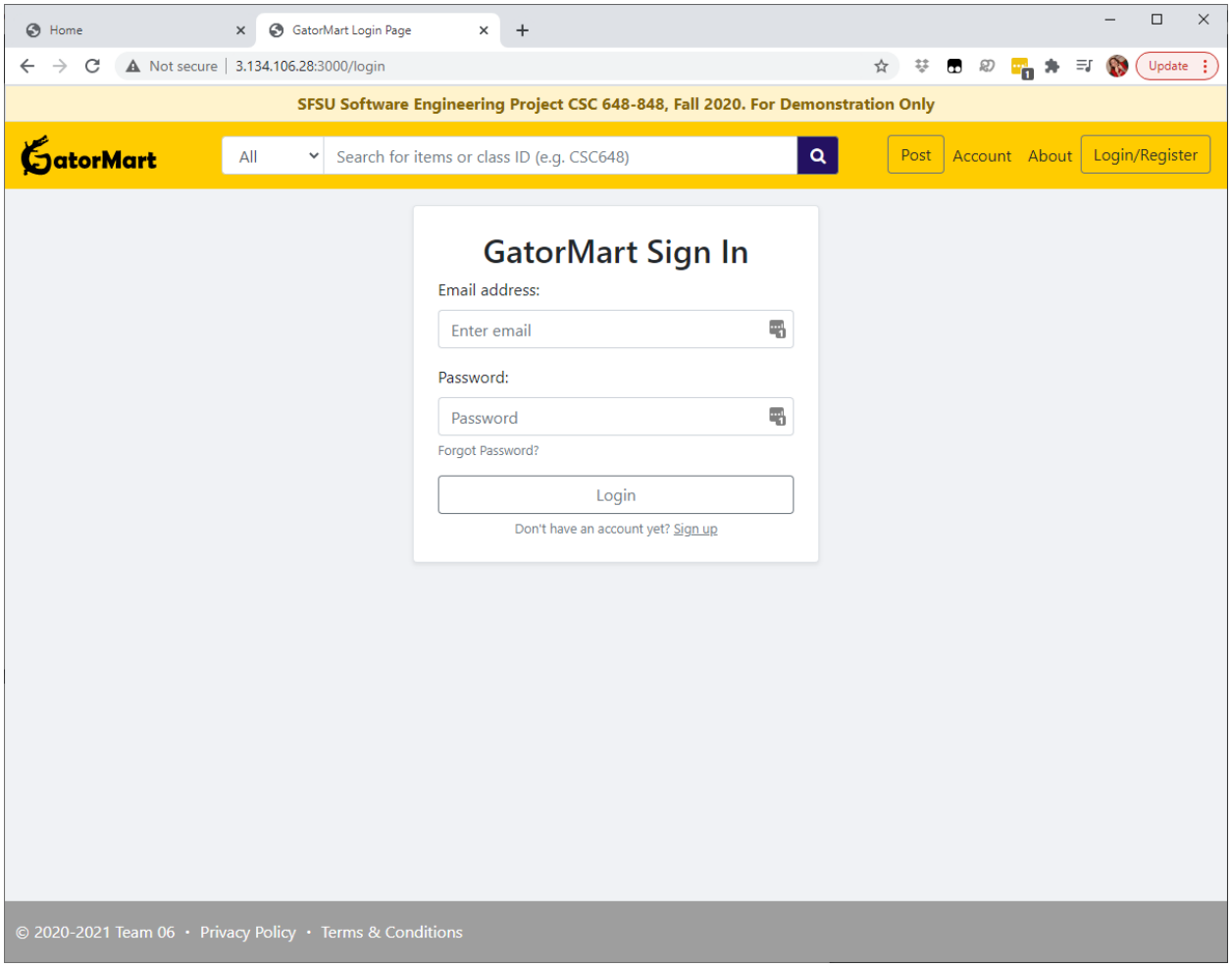
Product Screen Shots

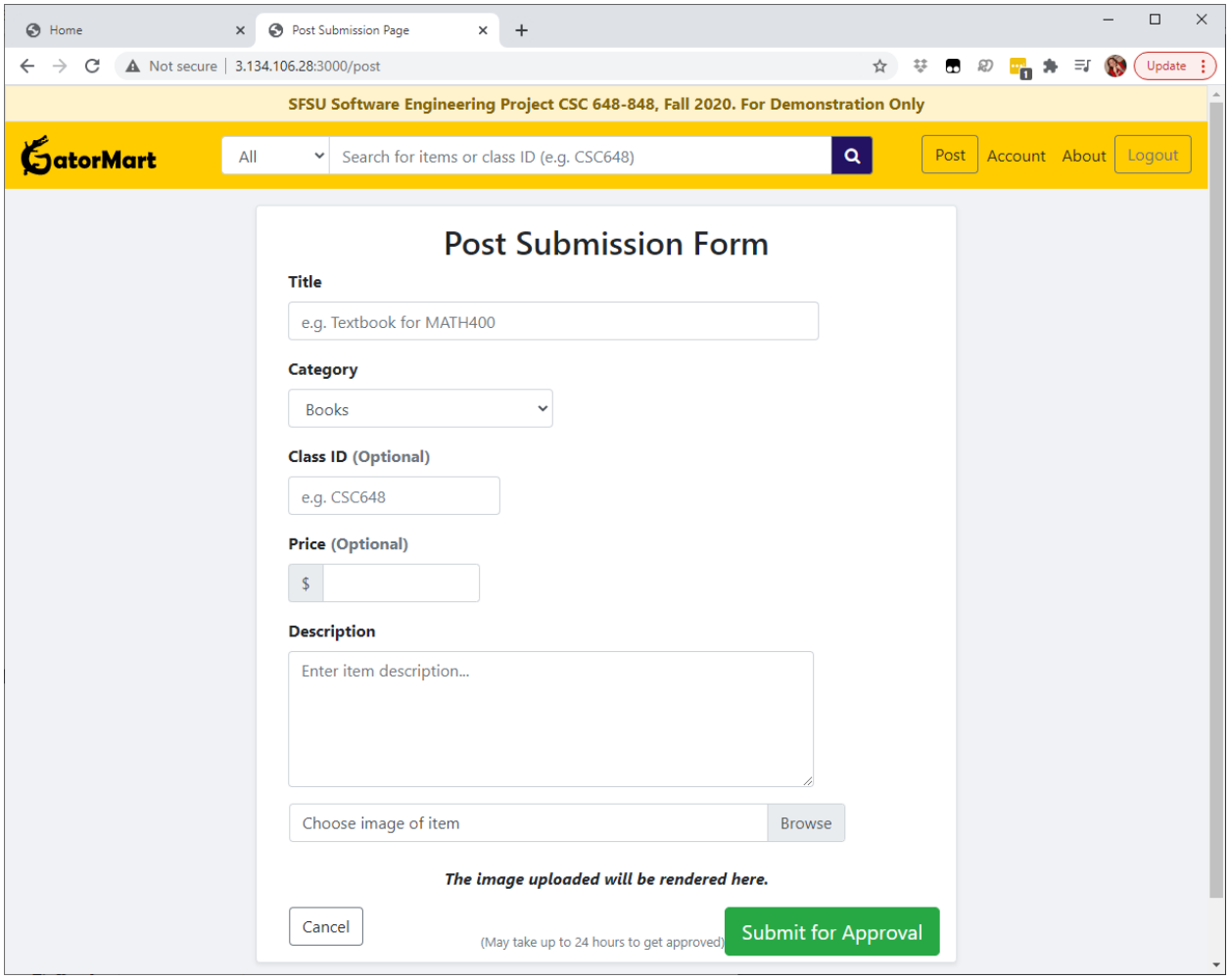


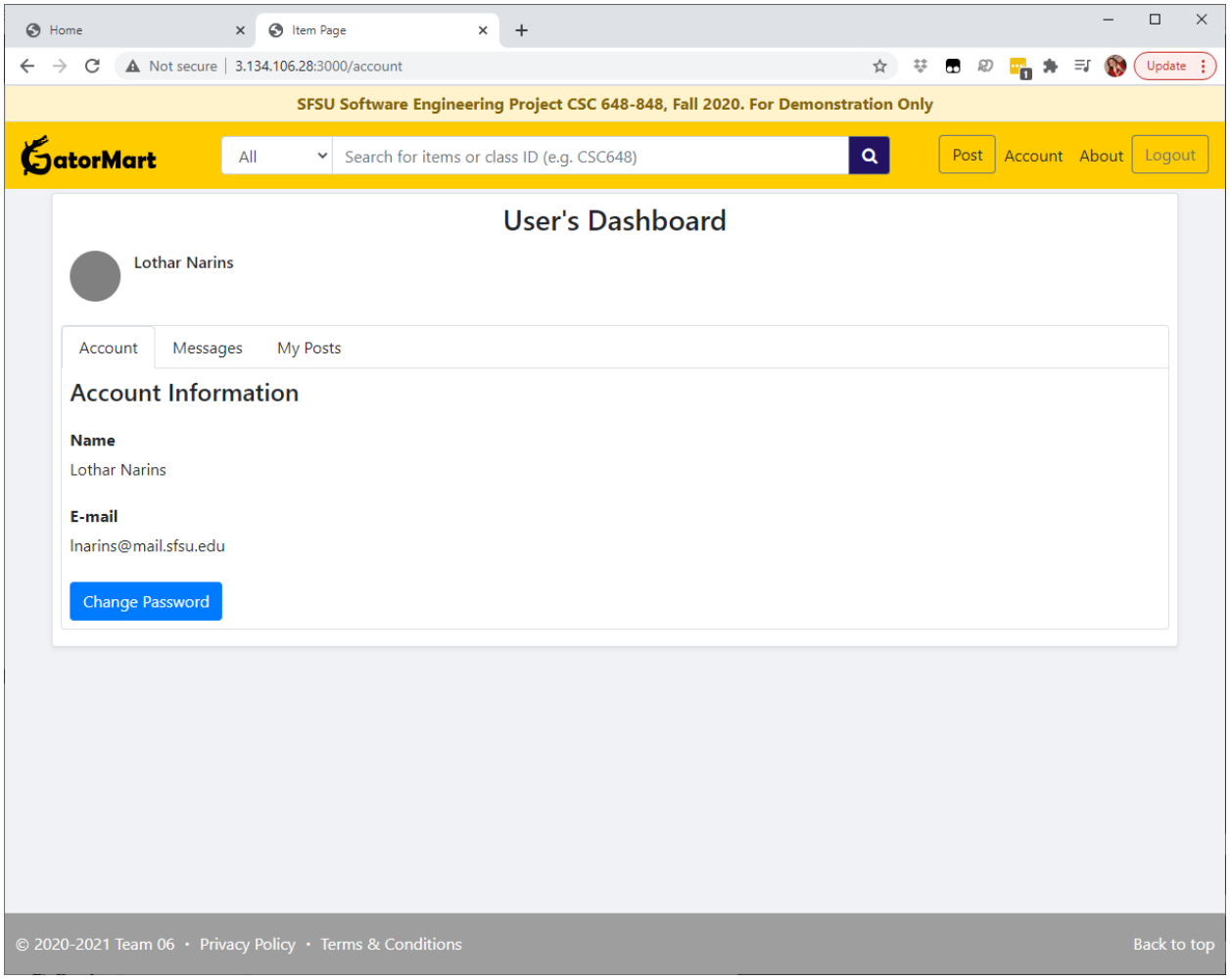

Home
Not secure | 3.134.106.28:3000/HTML/marketplace.html?category=All&searchtext=
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Login/Register
Showing 7 results
Price: Low to High
Slides for micros...
Item Request
2020-12-08
50 index cards
$1
2020-10-22
Where the Red F...
$10
2020-12-18
Mickey Mouse C...
$10
2020-12-18
Howl on Trial Bo...
$12
2020-10-22
Linear Algebra T...
$50
2020-10-22
couch
$100
2020-10-22
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions
Back to top

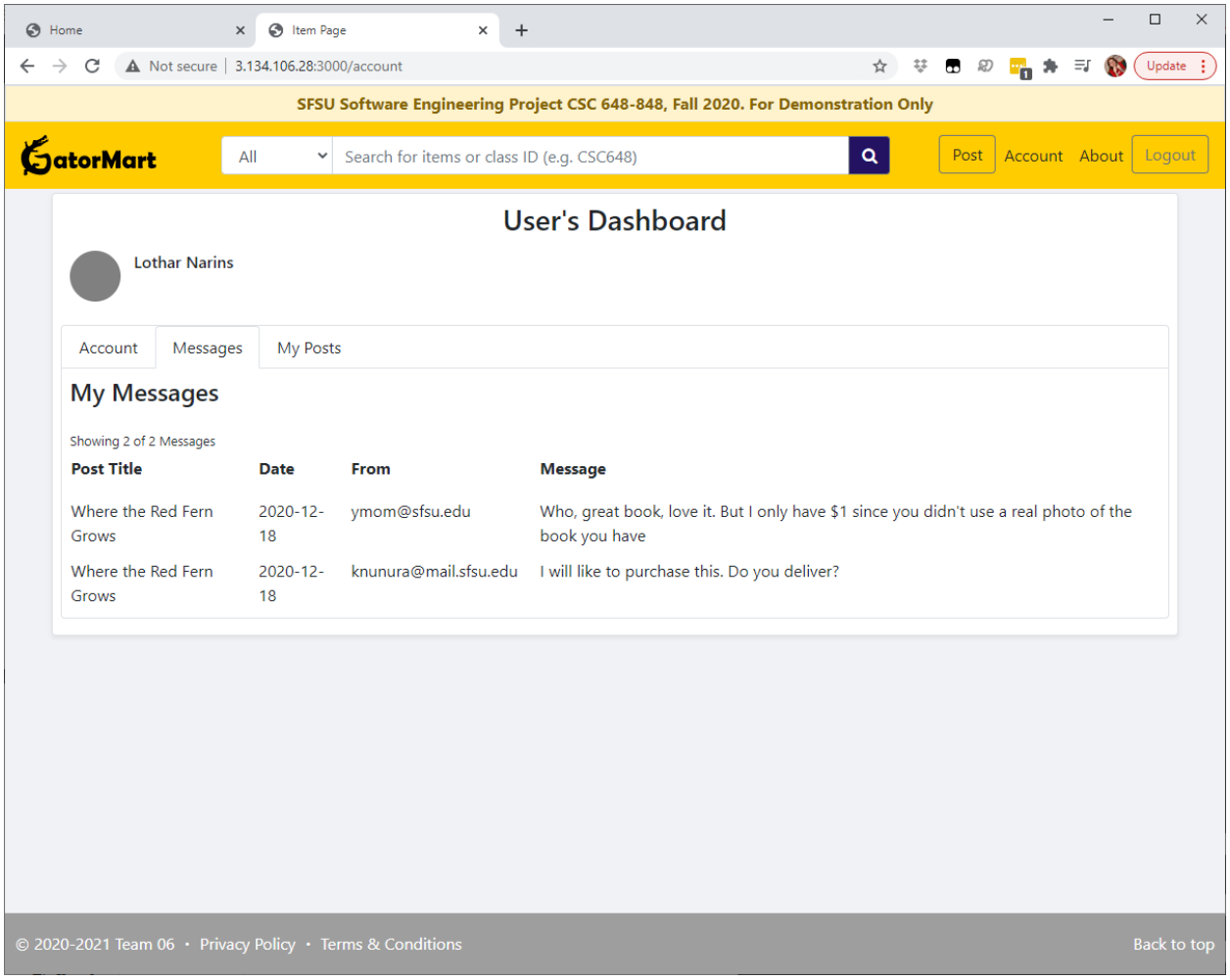

Home
Item Page
Not secure | 3.134.106.28:3000/posts/4
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Login/Register
Test User
TEXTBOOKS in MATHEMATICS
ADVANCED LINEAR ALGEBRA
NICHOLAS LOEHR
CRC Press
Linear Algebra Textbook
2020-10-22
$50
Category
Textbooks
Class ID
MATH325
Description
Taking the class next semester and looking for anyone that has taken the class
Message
Send seller your contact information (max 512 characters)...
Send
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions

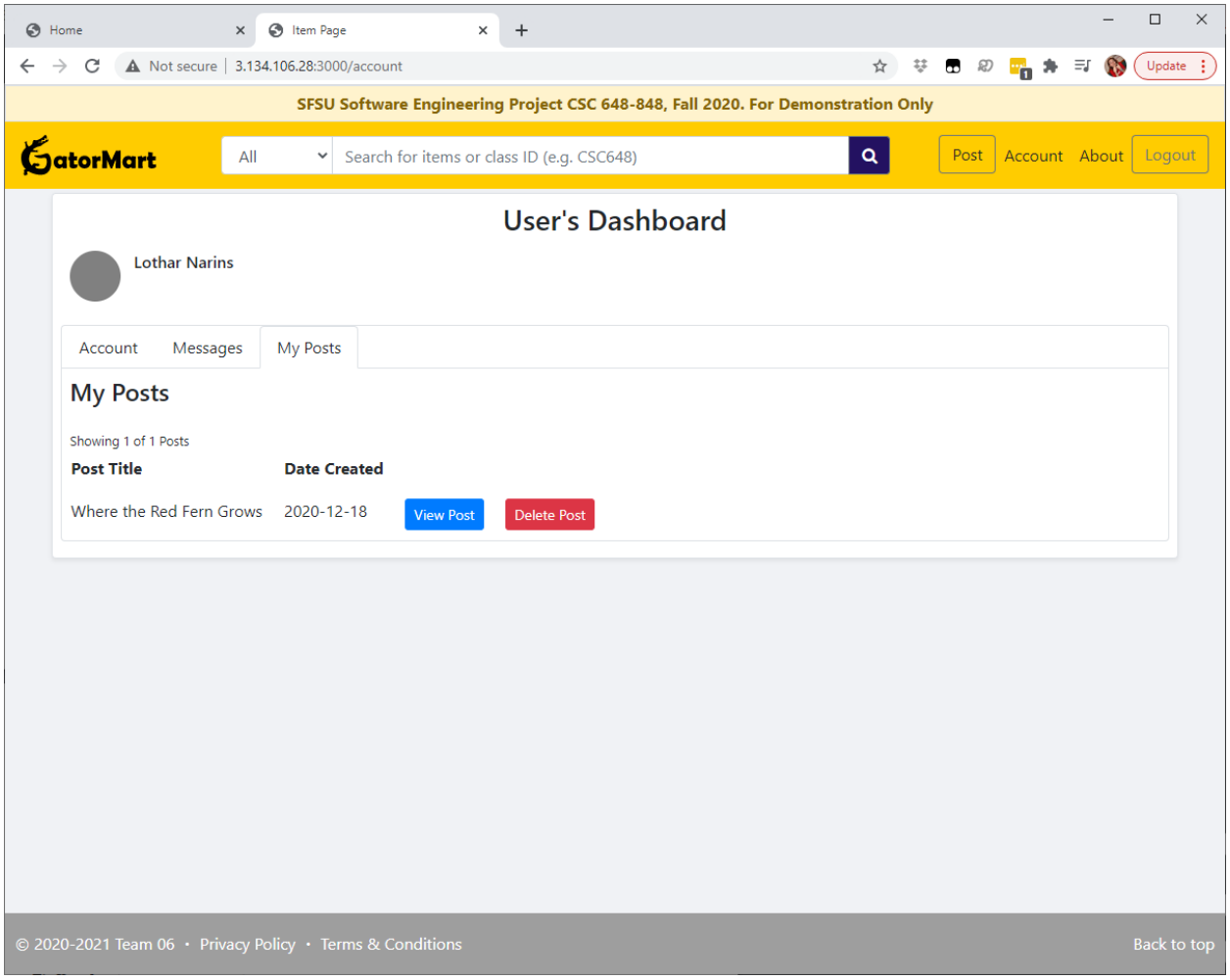

Home
GatorMart Register Page
Not secure | 3.134.106.28:3000/registration
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Login/Register
Register Account Form
First Name:
Enter your first name
Last Name:
Enter your last name
Email address:
Enter your SFSU email
Password:
Enter your password
Confirm Password:
Enter your password again
I agree to the Terms & Conditions
Submit
Already have an account? Click Here
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions


Home
GatorMart Login Page
Not secure | 3.134.106.28:3000/login
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Login/Register
GatorMart Sign In
Email address:
Enter email
Password:
Password
Forgot Password?
Login
Don't have an account yet? Sign up
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions


Home
Post Submission Page
Not secure | 3.134.106.28:3000/post
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Logout
Post Submission Form
Title
e.g. Textbook for MATH400
Category
Books
Class ID (Optional)
e.g. CSC648
Price (Optional)
$
Description
Enter item description...
Choose image of item
Browse
The image uploaded will be rendered here.
Cancel
(May take up to 24 hours to get approved)
Submit for Approval


Home
Item Page
Not secure | 3.134.106.28:3000/account
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Logout
User's Dashboard
Lothar Narins
Account
Messages
My Posts
Account Information
Name
Lothar Narins
E-mail
lnarins@mail.sfsu.edu
Change Password
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions
Back to top


Home
Item Page
Not secure | 3.134.106.28:3000/account
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Logout
User's Dashboard
Lothar Narins
Account
Messages
My Posts
My Messages
Showing 2 of 2 Messages
Post Title
Date
From
Message
Where the Red Fern Grows
2020-12-18
ymom@sfsu.edu
Who, great book, love it. But I only have $1 since you didn't use a real photo of the book you have
Where the Red Fern Grows
2020-12-18
knunura@mail.sfsu.edu
I will like to purchase this. Do you deliver?
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions
Back to top


Home
Item Page
Not secure | 3.134.106.28:3000/account
Update
SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only
GatorMart
All
Search for items or class ID (e.g. CSC648)
Post
Account
About
Logout
User's Dashboard
Lothar Narins
Account
Messages
My Posts
My Posts
Showing 1 of 1 Posts
Post Title
Date Created
Where the Red Fern Grows
2020-12-18
View Post
Delete Post
© 2020-2021 Team 06 · Privacy Policy · Terms & Conditions
Back to top

Database Organization

Users Table on server

mysql> SELECT * FROM users;

id	firstname	lastname	email	password	usertype	active	profilepic
1	Test	User	tuser@mail.sfsu.edu	$2b$10$ZwLlwg5lAV8zvH3QCNj5.eYJt8sOEjFMSB8Pnw5/qgKHMDEmPBfle	0	1	null
2	Another	User	another@mail.sfsu.edu	$2b$10$IibyjlM5w5xjlYHHlTc79eF.gBhaKELaa6TwssSvaRP/QfG563c/a	0	1	null
3	Lothar	Narins	lnarins@mail.sfsu.edu	$2b$10$Hd3xTpDC9.8MlKcGUfBpuOljkWKoBwbuMREbZXbJyrdGu/HSp7aZS	0	1	null
4	Allyson	Leung	aleung4@mail.sfsu.edu	$2b$10$l7vhiipdN0daNYkj2uwpcerM7hBUHnKc41BlMGtKaPlOyl4ZUIHd2	0	1	null
5	Your	Mom	ymom@sfsu.edu	$2b$10$plmElXtyU3M8Ua9sqhGzCOHoAyCs.6Onc6OXEmHurbNd4tEVB7yWO	0	1	null
6	Wilson	Young	wilson@mail.sfsu.edu	$2b$10$0ihBjZ9t0/P7vlNU7Eq8SuMtPSp4wNI3L89VFqlZkNGQpbic0YfqS	0	1	null
7	Kevin	Doe	knunura@mail.sfsu.edu	$2b$10$cqNC4NkKLsVsiw/SISlp9O4DpGtoypmeanexaMslwwOC.iQTLuaDW	0	1	null

Posts Table on server

mysql> SELECT * FROM posts;

id	title	description	price	class	created	photopath	thumbnail	approved	fk_userid	category
1	couch	Lightly used couch, in great condition	100	NULL	2020-10-22 16:42:39	../Images/uploads/temp.jpg	../Images/uploads/tempThumb.jpg	1	1	Furniture
2	Howl on Trial Book	Minimal wear, great condition	12	HUM376	2020-10-22 16:49:18	../Images/uploads/temp2.jpg	../Images/uploads/temp2Thumb.jpg	1	1	Books
3	50 index cards	Had some extra that I dont need anymore	1	NULL	2020-10-22 16:51:19	../Images/uploads/temp3.jpg	../Images/uploads/temp3Thumb.jpg	1	1	Supplies
4	Linear Algebra Textbook	Taking the class next semester and looking for anyone that has taken the class	50	MATH325	2020-10-22 16:57:08	../Images/uploads/temp4.jpg	../Images/uploads/temp4Thumb.jpg	1	1	Textbooks
6	Slides for microscope	I know that you can buy a pack, but was wondering if anyone had any extra left over from when they took the course	NULL	NULL	2020-12-08 17:55:59	../Images/uploads/temp6.jpg	../Images/uploads/temp6Thumb.jpg	1	2	Item Request
10	Where the Red Fern Grows	A great children's book for ages 9-14. In mint condition with no visible wear and no writing on the inside.	10	NULL	2020-12-18 02:51:51	../Images/uploads/6a9268f4647de60dcdf82ffe54a bccd85233aaf501e1.jpeg	../Images/uploads/thumbnail-6a9268f4647de60dcdf82ffe54abccd85233aaf501e1.jpeg	1	3	Books
11	Mickey Mouse Coffee Maker	Served me well in my dorm. Lightly used and ready for a new owner.	10	NULL	2020-12-18 02:56:58	../Images/uploads/73fcd89b5860bd7fd4d6d57b051 03d2d01e899579a0a.jpeg	../Images/uploads/thumbnail-73fcd89b5860bd7fd4d6d57b05103d2d01e899579a0a.jpeg	0	7	Furniture

Same Table in workbench for an alternative view

id	title	description	price	class	created	photopath	thumbnail	approved	fk_userid	category
1	couch	Lightly used couch, in great condition	100	NULL	2020-10-22 16:42:39	../Images/uploads/temp.jpg	../Images/uploads/tempThumb.jpg	1	1	Furniture
2	Howl on Trial Book	Minimal wear, great condition	12	HUM376	2020-10-22 16:49:18	../Images/uploads/temp2.jpg	../Images/uploads/temp2Thumb.jpg	1	1	Books
3	50 index cards	Had some extra that I dont need anymore	1	NULL	2020-10-22 16:51:19	../Images/uploads/temp3.jpg	../Images/uploads/temp3Thumb.jpg	1	1	Supplies
4	Linear Algebra Textbook	Taking the class next semester and looking for ...	50	MATH325	2020-10-22 16:57:08	../Images/uploads/temp4.jpg	../Images/uploads/temp4Thumb.jpg	1	1	Textbooks
6	Slides for microscope	I know that you can buy a pack, but was wond...	NULL	NULL	2020-12-08 09:50:21	../Images/uploads/temp6.jpg	../Images/uploads/temp6Thumb.jpg	1	2	Item Request
8	Where the Red Fern Grows	A great children's book for ages 9-14. In mint c...	10.51	NULL	2020-12-13 17:51:01	..\Images\uploads\2bd897cee327bea3bd8a653...	../Images/uploads/thumbnail-2bd897cee327be...	1	8	Books

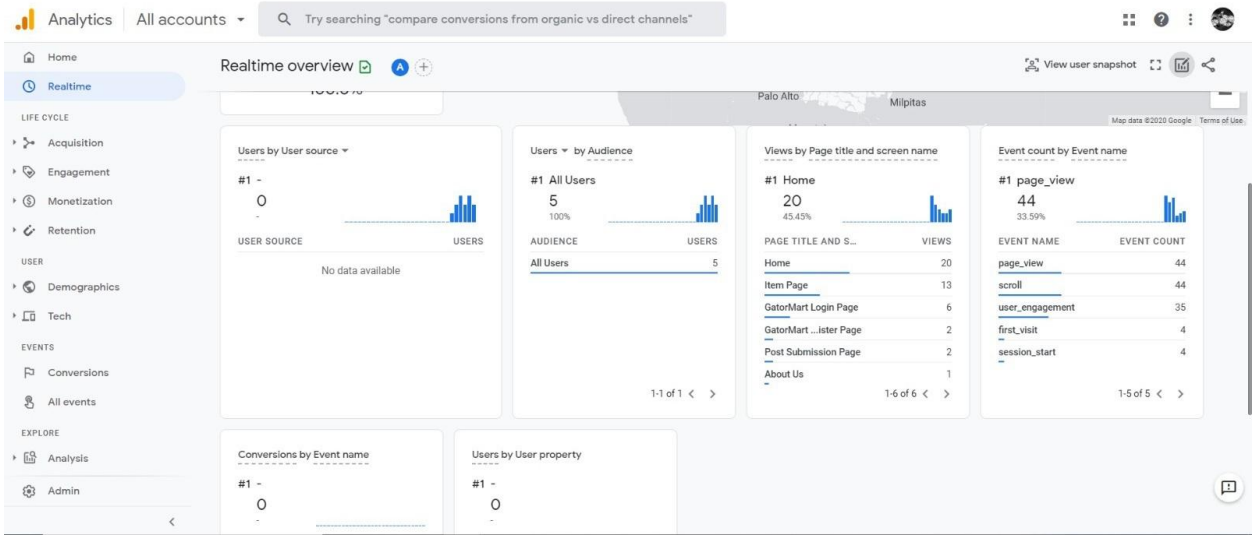
Sessions Table on server

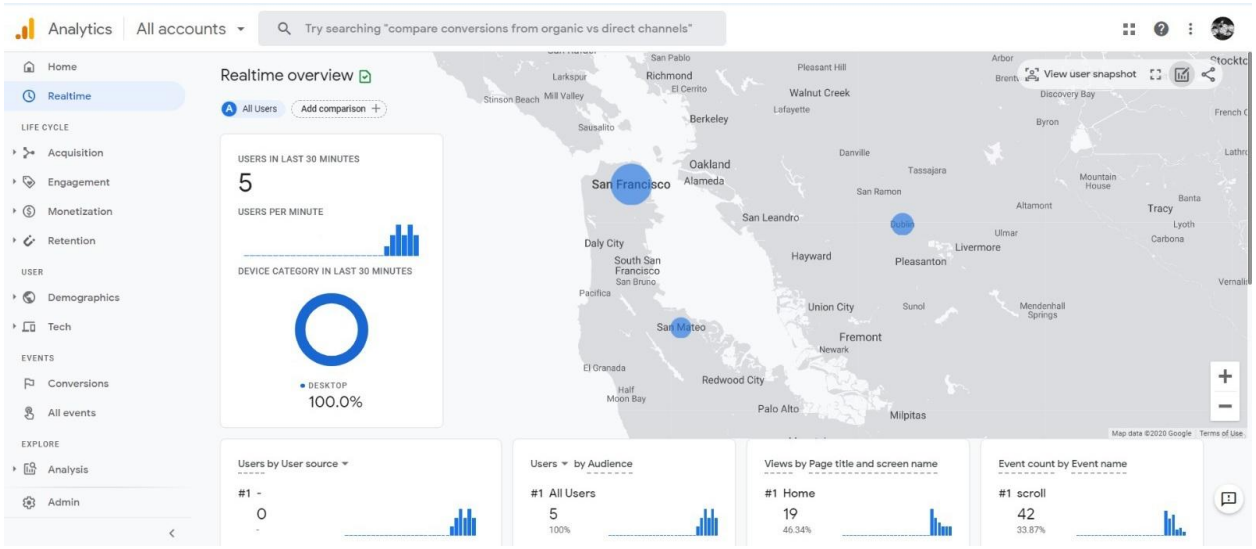
```
mysql> SELECT * FROM sessions;
+----------------------------------+------------+----------------------------------------------------
| session_id                       | expires    | data
+----------------------------------+------------+----------------------------------------------------
| 4kCIsemXJy-3wBVh6On_NjBaACfjuPMZ | 1608263109 | {"cookie":{"originalMaxAge":900000,"expires":"202
+----------------------------------+------------+----------------------------------------------------
```

Messages Table

```
mysql> SELECT * FROM messages;
+----+-----------+-------------+---------------------+-----------------------------------------------------------------------------------------------------+---------+
| id | fk_sender | fk_receiver | created             | message                                                                                             | fk_post |
+----+-----------+-------------+---------------------+-----------------------------------------------------------------------------------------------------+---------+
|  9 |         5 |           3 | 2020-12-18 02:56:19 | Who, great book, love it. But I only have $1 since you didn't use a real photo of the book you have |      10 |
| 10 |         7 |           3 | 2020-12-18 02:58:09 | I will like to purchase this. Do you deliver?                                                       |      10 |
| 11 |         5 |           7 | 2020-12-18 03:30:13 | Would look great in my kitchen, I'll meet you in front of the library for it                        |      11 |
+----+-----------+-------------+---------------------+-----------------------------------------------------------------------------------------------------+---------+
```

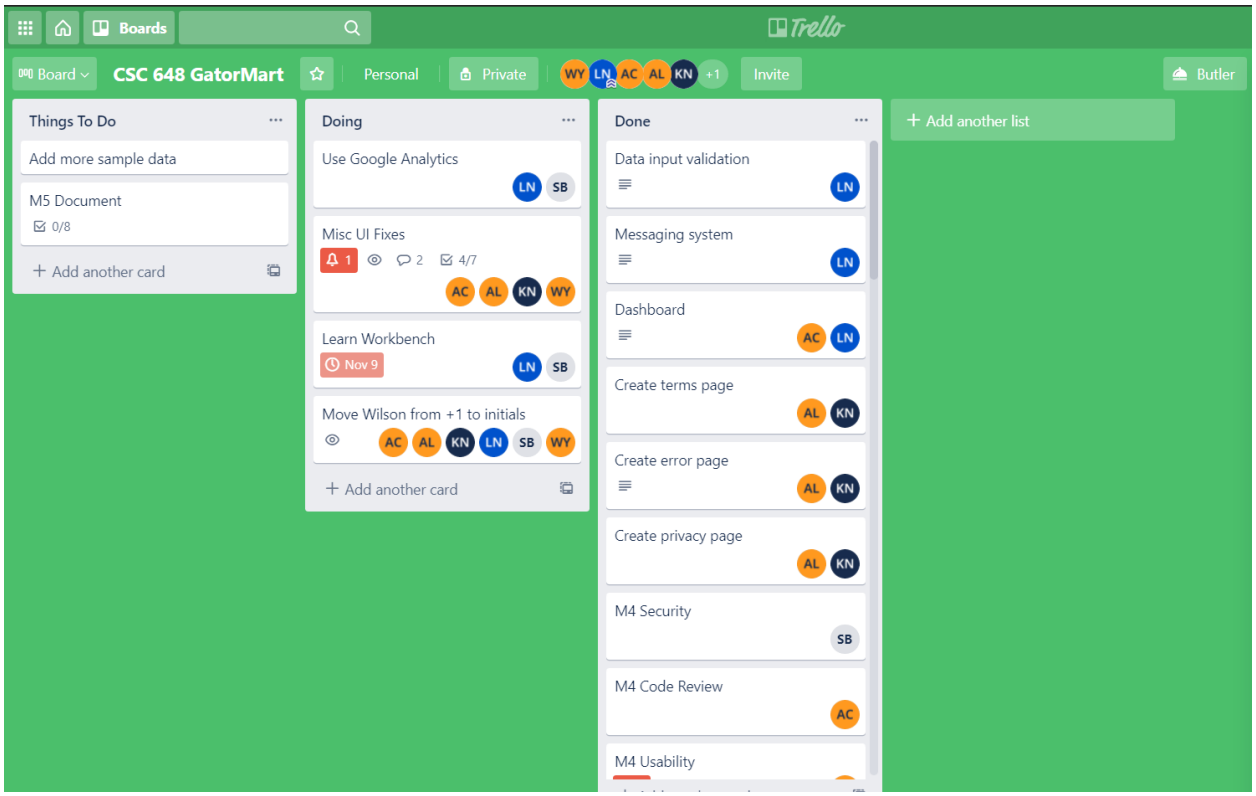
Google Analytics Plot

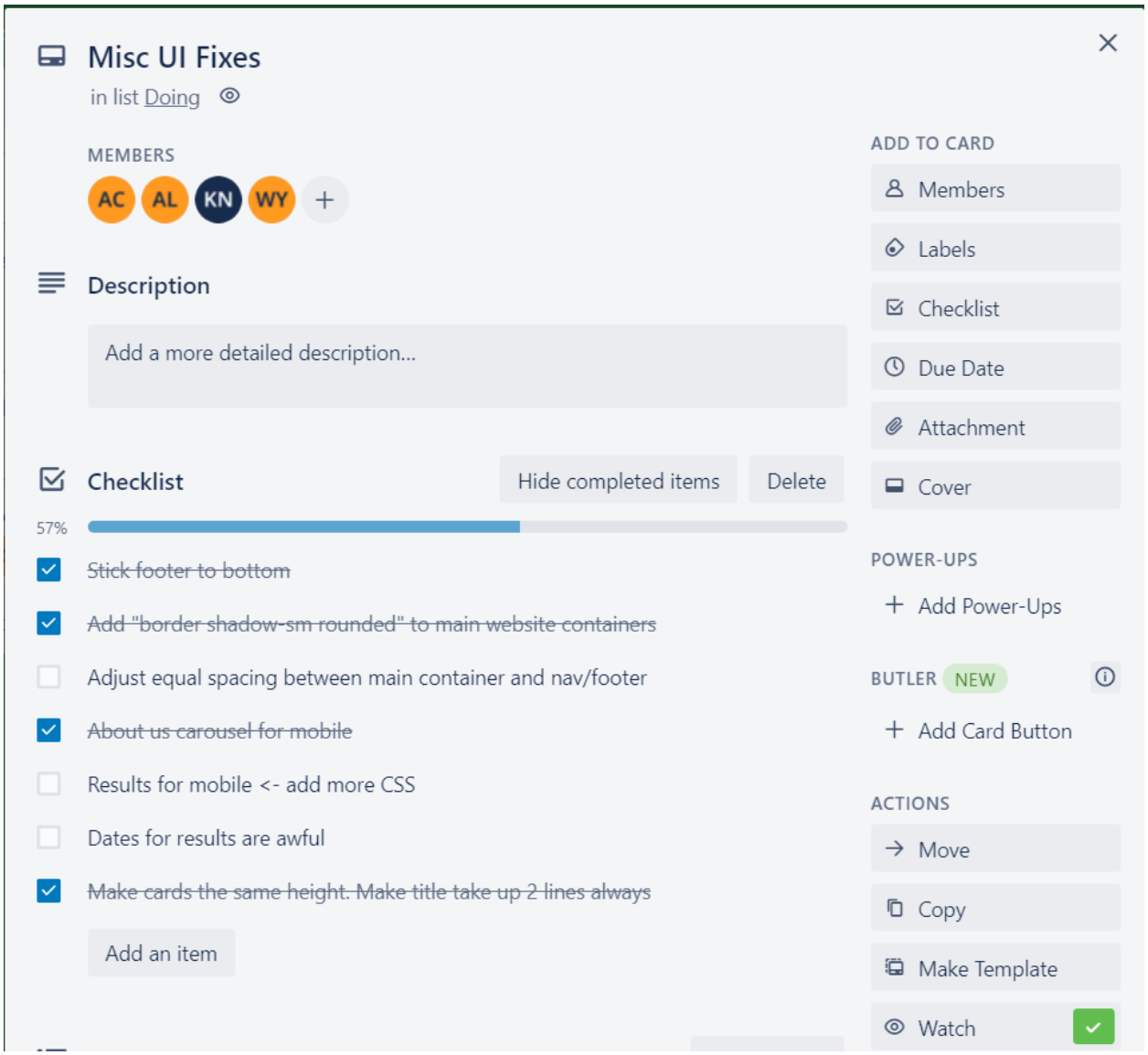


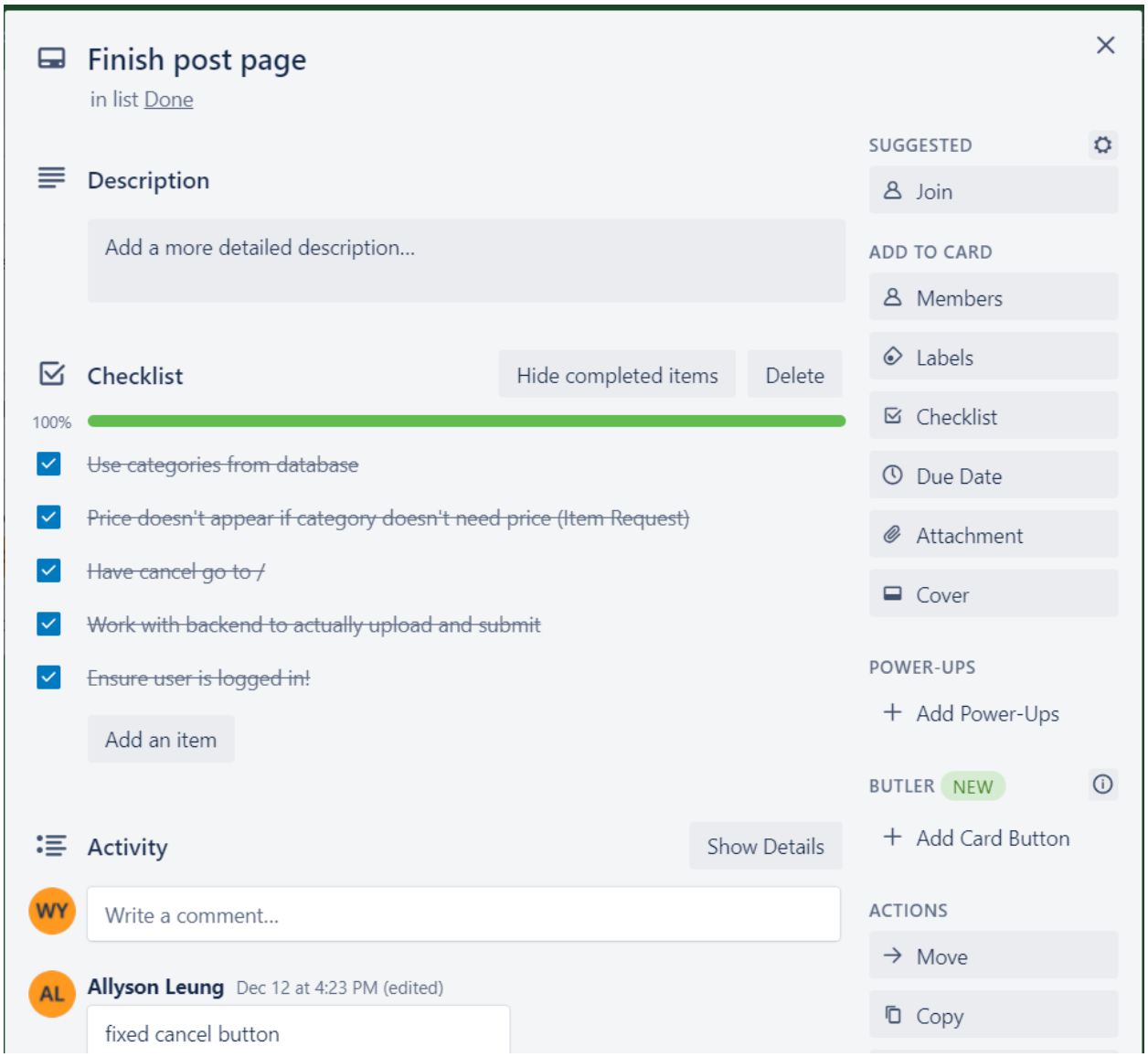


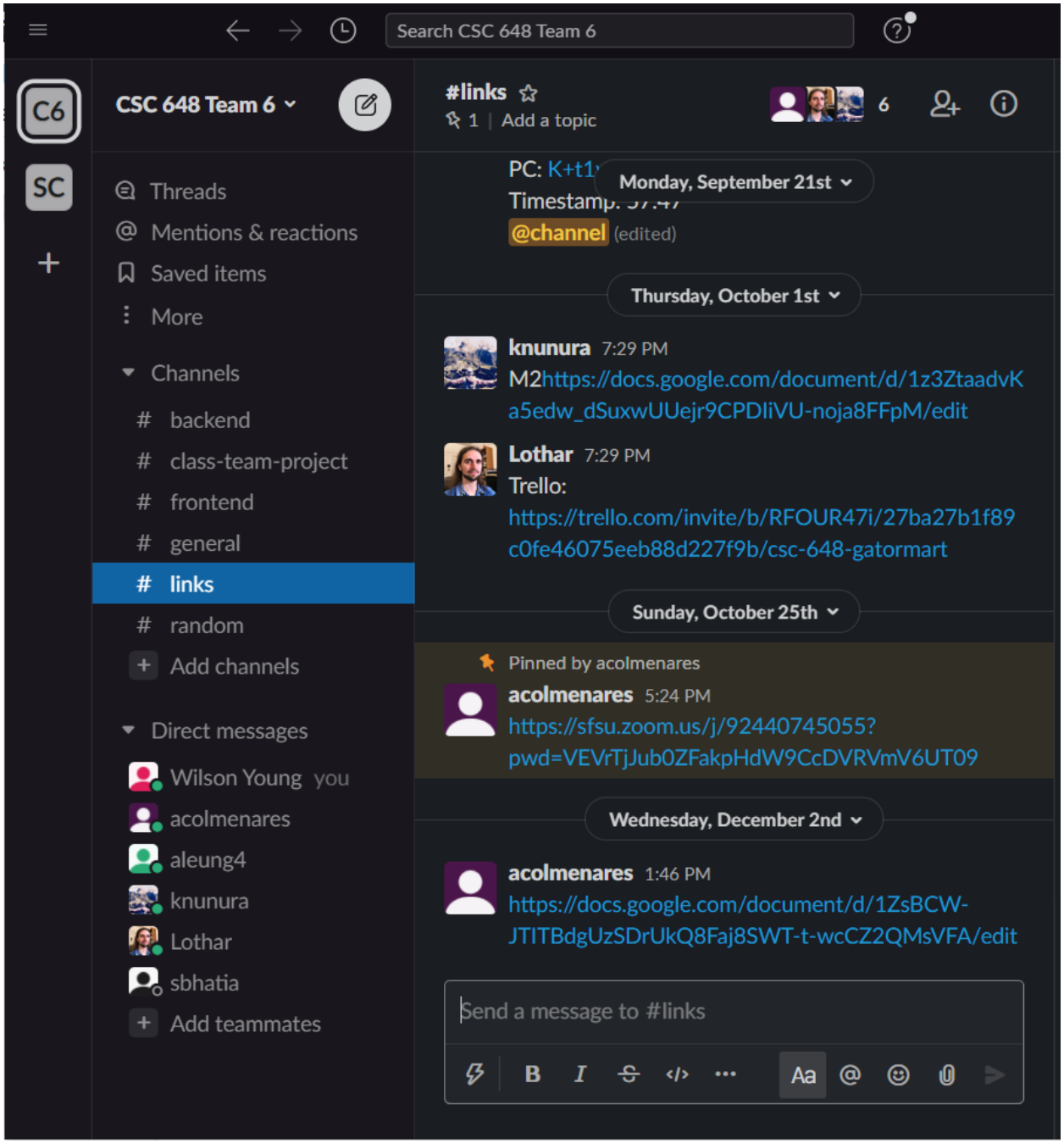
Project Management

Project Management Systems:




Misc UI Fixes
in list Doing
MEMBERS
AC AL KN WY
Description
Add a more detailed description...
Checklist
Hide completed items
Delete
57%
Stick footer to bottom
Add "border shadow-sm rounded" to main website containers
Adjust equal spacing between main container and nav/footer
About us carousel for mobile
Results for mobile <- add more CSS
Dates for results are awful
Make cards the same height. Make title take up 2 lines always
Add an item
ADD TO CARD
Members
Labels
Checklist
Due Date
Attachment
Cover
POWER-UPS
Add Power-Ups
BUTLER NEW
Add Card Button
ACTIONS
Move
Copy
Make Template
Watch


Finish post page
in list Done
Description
Add a more detailed description...
Checklist
Hide completed items
Delete
100%
Use categories from database
Price doesn't appear if category doesn't need price (Item Request)
Have cancel go to /
Work with backend to actually upload and submit
Ensure user is logged in!
Add an item
Activity
Show Details
WY
Write a comment...
AL
Allyson Leung Dec 12 at 4:23 PM (edited)
fixed cancel button
SUGGESTED
Join
ADD TO CARD
Members
Labels
Checklist
Due Date
Attachment
Cover
POWER-UPS
Add Power-Ups
BUTLER NEW
Add Card Button
ACTIONS
Move
Copy

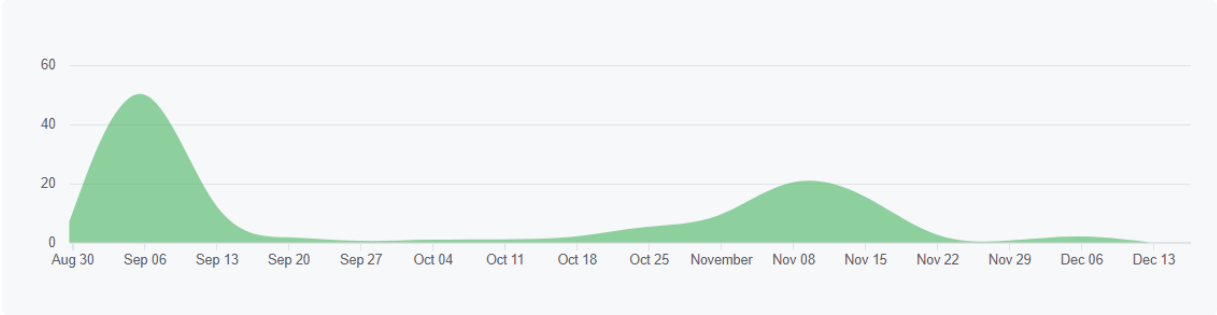


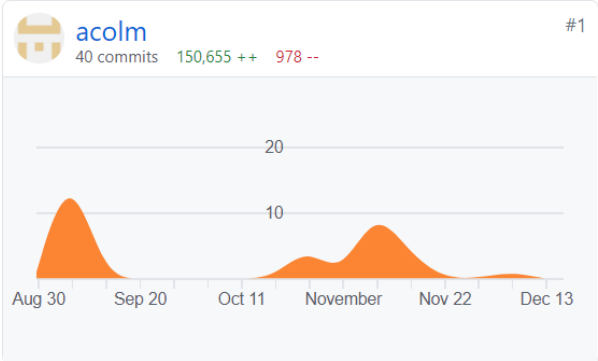
Github commits:

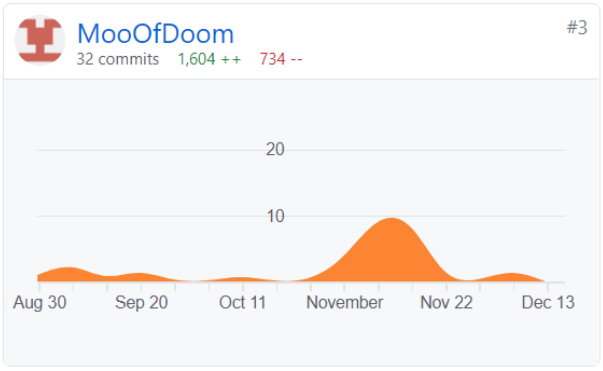
Aug 30, 2020 – Dec 17, 2020

Contributions: **Commits**

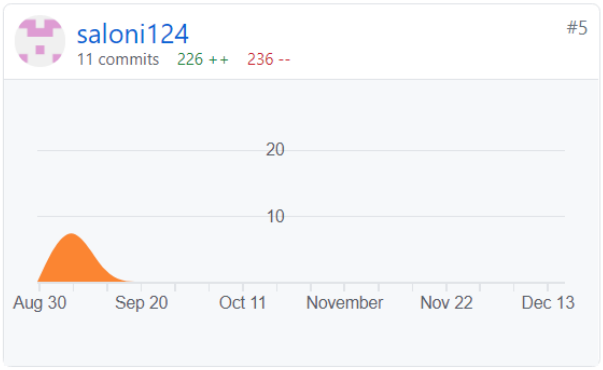
Contributions to master, excluding merge commits






MooOfDoom
#3
32 commits 1,604 ++ 734 --
20
10
Aug 30
Sep 20
Oct 11
November
Nov 22
Dec 13


knunura
#4
19 commits 3,276 ++ 1,636 --
20
10
Aug 30
Sep 20
Oct 11
November
Nov 22
Dec 13


saloni124
#5
11 commits 226 ++ 236 --
20
10
Aug 30
Sep 20
Oct 11
November
Nov 22
Dec 13


alleung
#6
4 commits 213 ++ 213 --
20
10
Aug 30
Sep 20
Oct 11
November
Nov 22
Dec 13

Team Member Self Assessment and Contributions

Team member self assessment and contributions

Lothar Narins <lnarins@mail.sfsu.edu>

Thu 12/17/2020 10:40 PM

To:

- Aaron Colmenares <acolmenares@mail.sfsu.edu>;
- Allyson May Leung <aleung4@mail.sfsu.edu>;
- Kevin Johan Nunura <knunura@mail.sfsu.edu>;
- Saloni Bhatia <sbhatia@mail.sfsu.edu>;
- Wilson Kai Young <wyoung@mail.sfsu.edu>

Hi Team,

a) As a team lead, I led the weekly meetings, created a Slack channel for communication, and managed a Trello board to keep track of tasks. I created outlines for all milestone documents and assigned tasks and sections to my teammates as well as contributed writing and coding of my own. On the technical side, I contributed to the search features, making the marketplace responsive to filter changes, and adding sort-by-price functionality. I also connected the item pages to the backend so that they load the content dynamically from the database, and I got the user dashboard and messaging systems working. I also did considerable testing on all the functionality.
b) According to GitHub, I've had 32 commits to the repository.
c) It was quite a challenge figuring out how to split the workload among the team members. For some milestones it seemed that the frontend team would have a lot more to do than the backend team, or vice versa. Planning out schedules for when things should be ready was also hard, and sometimes we ended up having to scramble to get things done on time when my estimates were poor. It was also a challenge for me to contribute to the codebase in the beginning because I was concurrently taking CSC 317 and had very little web development experience prior to this class. Only once we had a working example of how the backend was going to be linked to the frontend did I feel comfortable adding my own code.
d) I would try to figure out how to break down the tasks better and have more checkpoints. There seemed to be quite an uneven spread of workload throughout the semester, where some weeks there was hardly anything to do, while in others we were scrambling to barely complete things on time. I would also want to have more code reviews to get everybody more familiar with the codebase.

Best,
Lothar

Re: Team Member Self Assessment and Contributions

Kevin Johan Nunura <knunura@mail.sfsu.edu>

Thu 12/17/2020 9:46 PM

To:

- Wilson Kai Young <wyoung@mail.sfsu.edu>;
- Lothar Narins <lnarins@mail.sfsu.edu>;
- Aaron Colmenares <acolmenares@mail.sfsu.edu>;
- Allyson May Leung <aleung4@mail.sfsu.edu>;
- Saloni Bhatia <sbhatia@mail.sfsu.edu>

Good evening Team,

a) I primarily contributed to the frontend development of our team's website UI. The peak of my contribution was in the early state of the project beginning with the creation of the "About Us" template, use cases, and UI mockups utilizing Figma. After our initial UI mockups, I worked directly with the frontend and backend lead to create our vertical and horizontal prototypes in relation to our approved UI mockup which includes the overall structure of our HTML pages utilizing Bootstrap v4. Additionally, I assisted the team in fixing small UI imperfections and creating assigned HTML pages for our website.

b) I made a total of 19 commits to the GitHub team Dev. branch.

c) My main challenge throughout this project was getting familiar with the stack which was primarily focused on Web Software Development. My lack of adequate experience in this field made it difficult to assist my team as much as I desired. However, I was able to improve in my frontend skills throughout this project to meet our team goals and milestones. However, my biggest challenge was being able to assist the backend due to my lack of experience in server-side knowledge and languages such as JavaScript. Fortunately, our team had an incredible backend and team lead to guide us in this field.

d) In the next opportunity that I get, I would be more involved in the frontend side of a project and heavily focus on implementing JavaScript from the frontend side. I feel like this process would help me get better at frontend development and help prepare me for backend development. I would also dwell more into the backend side of the project as there is still a lot to learn including source code refactoring. Finally, this experience with the entire Software Development Cycle and processes has reinforced my believe that communication is key in a team setting, therefore, I would make sure to be even more communicative with the team and do my best to involve all team members in the project as much as I can.

Respectfully,
Kevin Nunura

From: Wilson Kai Young <wyoung@mail.sfsu.edu>
Sent: Thursday, December 17, 2020 7:44 PM
To: Lothar Narins <lnarins@mail.sfsu.edu>; Aaron Colmenares <acolmenares@mail.sfsu.edu>; Kevin Johan Nunura <knunura@mail.sfsu.edu>; Allyson May Leung <aleung4@mail.sfsu.edu>; Saloni Bhatia <sbhatia@mail.sfsu.edu>
Subject: Team Member Self Assessment and Contributions

a. For the team project, I contributed to the team project by making sure our Github repository was properly organized and the branches were used for their intended purposes. I also ensured no commits to the main branch were made without ensuring there were no bugs with the branch we were merging into the main. I helped contribute to the front end of the project by fixing some UI bugs and helped with styling and arranging components in other pages.
b. I made 37 commits to the development branch of our team project.
c. The main challenge I faced was learning how to use Github as a team. I was primarily used to using Github to upload my own work, but this was my first time managing a team project. I didn't have much experience merging branches and looking at commit conflicts before this so it was a nice learning experience. I also learned that you should make branches for new features of our project, which I had never done before. This was a challenge to manage all the new branches and make sure everything was up to date. Lastly, another challenge I had was working with HTML in general because I only had one class worth of experience prior to this.
d. I would utilize branches more, I felt as Github master, I could've organized the project in a more clear and concise manner with how our branches were organized. We had branches for certain features but not for every new feature we added. In a professional environment, this could've caused our project to have bugs in features that weren't part of the right branch and lead to confusion. In a real life scenario I would've also liked to have more testing on the project since we weren't able to have any actual user tesing for the website.

Team 6: Member Self Assessment and Contributions

Aaron Colmenares <acolmenares@mail.sfsu.edu>
Thu 12/17/2020 9:01 PM
To:

- Lothar Narins <lnarins@mail.sfsu.edu>;
- Wilson Kai Young <wyoung@mail.sfsu.edu>;
- Allyson May Leung <aleung4@mail.sfsu.edu>;
- Kevin Johan Nunura <knunura@mail.sfsu.edu>;
- Saloni Bhatia <sbhatia@mail.sfsu.edu>

Hello team,

a. For this semester's project, I was the backend lead, which included setting up the MySQL table schemes and figuring out how data would be passed from the user to an eventual server. I set up our server on AWS and even ended up contributing to the front end. A more technical summation of my contribution includes setting up routes for pages, writing javascript that would take user information,

and having the site interact with that user data; this not including the front-end work of creating CSS classes and organizing Html.
b. I made 41 commits to the development of our project.
c. The main challenge I faced when working on this project came from deploying the site on a server. I had never done this before, and configuring AWS separate from logging in was a multistep process. Having to install different software and learn their commands on the command line had me visiting Google often. Another area of difficulty was setting up MySQL on the server since it was all done through a shell. Transferring over a database from the workbench to the server also had me searching for what ended up being a unicode difference.
d. Next time I have a project like this, I will try to put as many javascript functions in their own files instead of in the Html. This sentiment extends to trying out different ways of managing the site like with handlebars. I would also try to emphasize the use of multiple files for one function in a team so that there would be fewer merge conflicts.

Team Member Self Assessment and Contributions

Allyson May Leung <aleung4@mail.sfsu.edu>
Thu 12/17/2020 10:12 PM
To:

- Lothar Narins <lnarins@mail.sfsu.edu>;
- Saloni Bhatia <sbhatia@mail.sfsu.edu>;
- Kevin Johan Nunura <knunura@mail.sfsu.edu>;
- Wilson Kai Young <wyoung@mail.sfsu.edu>;
- Aaron Colmenares <acolmenares@mail.sfsu.edu>

a) I mainly contributed to this project by working on the frontend aspect of the website. I helped design and put together initial mockups for each page in Figma. Using HTML and Bootstrap, I wrote the initial code for login, register and post submission pages. On those pages, I mostly worked on the form validation such as making sure that the user's email was an SFSU email or that the passwords matched and writing little instructions for which fields were missing information.
b) I made 4 commits
c) I don't think I had very many technical challenges, but I think I was challenged in terms of the working on a team. I didn't really feel like I fulfilled the role as frontend leader that I volunteered for. I sort of just let the other team members decide and argue about how we designed our website and just went with whatever they decided. I don't think I was really working with the backend as much as I should have and because of that I didn't always understand how it worked so I couldn't really help them as much as I wanted to. I also feel like I could have spoken up more in our team meetings.
d) Something I would want to do next time is be more involved in the backend part of the project and not just limit myself to frontend. I feel like I was a little too much in my comfort zone on the frontend and would really like to at least understand how the backend stuff works because I think that would also help me improve in designing the frontend. I feel like this way I might feel a little more comfortable speaking up in meetings and overall communicating with the rest of the team because I basically have to.

Team Member Self Assesment

Saloni Bhatia <sbhatia@mail.sfsu.edu>

Thu 12/17/2020 11:50 PM

To:

- Lothar Narins <lnarins@mail.sfsu.edu>;
- Kevin Johan Nunura <knunura@mail.sfsu.edu>;
- Allyson May Leung <aleung4@mail.sfsu.edu>;
- Aaron Colmenares <acolmenares@mail.sfsu.edu>;
- Wilson Kai Young <wyoung@mail.sfsu.edu>

a) I was a part of the initial planning of the site along with the rest of the team, coming up with ideas such as a wish list that could enhance our website. During the term of this project, I was a part of the backend team for this project, which included looking into the incorporation and understanding of workbench as well as assisting the backend lead. I also aided in incorporating Google Analytics into our website.

b) As per GitHub, I have done 11 commits.

c) Being a part of the backend team was a bit of a challenge for me as I was only experienced with front-end work. I was also intimidated by the back end, not quite understanding what I was doing at first. This had initially hindered me and also resulted in me having an issue with finding my footing with the team. I was, however, able to overcome it and came out with a better understanding of MySQL and backend development.

d) I would like to be able to reach out to my team members more next time. It may be due to our virtual situation, but I didn't feel comfortable reaching out to my teammates to follow up with other parts of the project and only stuck to the part that I was assigned. Due to this, I fell into a pattern of following what was needed and mainly communicated with either my team lead or backend lead. Had this been a physical environment, I feel that I would have also been able to learn from my team members more. As such, I also want to work on my adaptability.