



FINAL PROJECT FOR SW ENGINEERING CSC648/848

FALL 2019

Gator Barter

Section 01

Team # 04

Aditya Bodi, Team Lead | abodi@mail.sfsu.edu

Alan Ng, Frontend Lead

Philip Yu, Frontend Engineer

Deming Yan, Full Stack Engineer

Akshay Kasar, Git Master- Backend Engineer

Alex Kohanim, Backend Lead, Full Stack Engineer

Tejasvi Belsare, Document Master- Backend Engineer

<https://gatorbarter.store>

Date : 12/16/2019

Table of Contents

Product summary	3
Milestone documents – M1-M4	4
ScreenShots of the actual final product as shown in the demo	43
Screenshots of key DB tables	48
Google analytics plot for your WWW site	49
Team member contributions	51
Post analysis – lessons learned	58

Product summary

Name of our product – Gator Barter

Below is the list of all major committed functions for Gator Barter :

1. Home Page
2. Search results (with search text validation)
3. Filter search by category
4. Sort the search results
5. Item details on the separate page including information like images, description, price, and seller
6. User Login
7. User Registration (including lazy registration)
8. Item posting (with all fields validation)
9. Messaging from a user to other user
10. Item trading (unique feature)
11. Seller dashboard (including seller items and messages)
12. Admin dashboard (to approve or reject the items posted for selling)

Gator Barter is a buying and selling website specially designed for San Francisco State University students. Gator Barter will be marketed as a website created by SFSU students for SFSU students. This website would serve as a platform for students to sell their items, class materials, etc. as well as buy essential items at a cheaper rate compared to the market price that too within the campus. Students can also give away items free of cost on the site or trade their items with other items. It will provide an affordable and safe marketplace for students to exchange their belongings.

Moreover, this website builds direct communication between buyer and seller. This unique approach will eliminate long wait times and no shipping costs would be involved. The basic attraction of our website is its simplicity and user-friendliness. Also, the good design of Gator Barter can be eventually used as a template for other universities looking to design their own buy and sell website.

URL to our website: <https://gatorbarter.store/>

Milestone documents – M1-M4

Milestone 1

Executive Summary

San Francisco State University does not have local buying and selling options unique for SF State students. Our team is a motivated group of students who have gone through a lack of local buying and selling option. We wish to make this easier for everyone that comes after us. Being students of the Computer Science department, we have decided to launch our professional website named “Gator Barter”. We come from many different backgrounds and interests and represent a small portion of the intersectionality of the SFSU student. We will combine these many different points of view to create a site that can cater to as many of these needs as possible. This website would serve as a platform for students to sell their items, old class materials, etc. as well as buy essential items at a cheaper rate compared to the market price. This website also provides an option of trading things or free items where students can either trade the items without money or give away the items for free. Moreover, this website builds direct communication between seller and buyer. This unique approach will eliminate long wait times and shipping cost. It will also help students to build relationships within the university and save their money.

Our website, Gator Barter would allow registered users to post the items they wish to sell on the website. Once posted, it has to be approved by the admin before going live for selling. The post would contain sales item, title, single or multiple images of the item, category, price, brief description. The website can be browsed by any unregistered or registered user for checking the required posts/items. Registered users, interested in buying an item from respective sellers, can message the seller directly and could then set up a meeting point to complete the purchase. It would provide a safe exchange between the seller and the buyer. So, users can buy/sell or trade for items on the Gator Barter website. Also, they can get free items if, someone has posted. Buyers and sellers can contact each other through messages and decide a place to meet for buying or selling.

Our student startup team has seven motivated SF State students from the Computer Science department. Four students are undergraduate and three of them are graduate students. The entire team would work collaboratively towards the development of the website starting from basic requirement gathering, till final product delivery.

Personae and Main Use Cases

Personae:

1. Nikita, a student of SF State pursuing Masters in Embedded Electrical and Computer Systems

- Nikita is an EECS student and works on few electronics related projects
- Nikita requires equipment and materials related to her electronics projects
- Nikita is a very busy person given her schedule; she has less time to spend on other activities
- Nikita's close contacts are her friends and classmates in the school.
- Nikita stays on-campus so that she can work on her projects in the science lab at school.
- She is very interested in online shopping and frequently does it.



2. John, a recent graduate and working at Texas

- John recently graduated and got a great job offer in Texas.
- John has got busy with his new life.
- John stayed at SF state dorm.
- John wants to keep things simple.
- John wants the moving out process to be quick.
- John is familiar with internet surfing.
- John prefers communicating with trusted users



3. Bucky, a senior student at SF State and an intern

- Bucky is a senior in CS at SFSU.
- Bucky is also doing internship at San Jose.
- Bucky moved to San Jose and comes to the campus 4 days a week.
- Bucky has bought a few CS-related course textbooks in the previous semesters which are not required now.
- Bucky is in need of some stuff required for his new room.



4. Richard, a very busy working employee at IBM

- Richard is a Consultant for IBM.
- He lives in Seattle
- Richard is very busy and cannot waste time being trained to use a new platform.
- Richard wishes to have tools to increase his workflow
- Richard is willing to work as an Admin for a website.



User Cases:

1. Nikita is a graduate student from EECS department at SF State. She lives near campus. She needs a PCB board for her Embedded course project. As project deadline is closer, she doesn't have enough time to buy the new board. Buying online would be a slow process considering the shipping time. Also, used PCB board would serve her perfectly from this project's point of view. She needs it within the next 2 days, so that she can cover up her project work and complete milestones on time. Our website serves as a great opportunity to this student. There will be students who are willing to sell their used/new pcb boards on the gator barter website. She can access the Gator Barter website to find the post for a PCB board and buy it. She can get it from the seller in very less time as the they can meet on campus.
2. John is a student from SF State Mathematics department. Now, he is moving out of SF State dorm and wants to sell his room items quickly before moving. Selling online would be a slow process. It would save a lot of time if he can sell those items to someone from the dorms or SF State itself. He can access our gator barter website to and create posts which he wants to sell. He can choose a relevant category to sell his items through the website. As this website is accessed by SF state students, John can easily find potential buyers.
3. Bucky is a senior student from the Computer Science Department. He lives in San Jose and is on campus 4 days a week. Bucky has few CS course-related textbooks, which he doesn't need once he graduates. He wants to trade those readers/textbooks with some other stuff required for his new room. As course material is the same for courses, it would be good if someone from SF State CS department can trade those textbooks. As Bucky comes frequently to the school, he can trade things with the students with an ease. Gator Barter website would be the best choice for Bucky to get his work done. As our website is used by the SF State students, he can easily find students who are willing to buy/trade textbooks. He has to create a post on the website with relevant course related category and choose an option to trade. This post will then be available to the students as a tradeable option.
4. Richard is a Consultant for IBM. He lives in Seattle and has experience in managing subreddits. Richard is very busy and cannot waste time being trained to use a new platform. He will be one of the admins of our Gator Barter website. Richard wishes to have tools to efficiently moderate posts, approve/reject posts, add/remove users and communicate with users if their posts are inappropriate. Richard prefers a clean and simple workflow that allows him to quickly execute a task and get back to his other work. As the workflow of our website will be simple and smooth, it will be a good place for him to be the admin of our Gator Barter website.

List of Main Data Items and Entities

1. User: Any person who uses Gator Barter is a user.

a) **Unregistered User:** These users have below access –

- Unregistered users can browse and search the website to check for the items being sold/tradeable by category

- Unregistered users are not allowed to post or buy items

b) **Registered User:**

- Users who have registered and created an account with Gator Barter. User account contain first name, last name, SF State email id and password.

- Registered users can log in the website, to either sell their items or buy available items on the website.

- Registered users can be both sellers as well as buyers. They can check their transaction history for the items they sold through the website.

c) **Administrator:**

- Administrator is a user having special privileges to access all data and content as well as modify the database.

- Administrator is capable of monitoring Gator Barter transactions

- Administrator can perform tasks such as remove illegal posts, removing suspicious users, review and approve pending posts from sellers in order to be posted on the website, etc.

2. Item:

- It is a product listed on the website by a registered user

- Item consists of Name/Title of the item, Price, Image(s), Description about the item, Category, Name of user who posted.

- Buyers can browse by different categories to check for items as per their need

- Sellers need to select a category for posting their item for sale

3. Messages:

- The communication medium between a buyer and a seller used for in-site messaging in Gator Barter
- Buyer can contact the seller by sending them a message through Messages
- The seller can check for the messages received from buyers for their posts

4. Categories:

- These are different types of items listed for buying or selling
- Categories would help to sort and group posts into different sections

Initial List of Functional Requirements

1. Unregistered users –

1. Unregistered users shall be able to browse the Gator Barter website.
2. Unregistered users shall be able to search any item as per their need on the Gator Barter website.
3. Unregistered users shall be able to search as per Category on the Gator Barter website.
4. Unregistered users shall be able to see search results in an organized manner.
5. Unregistered users shall be able to filter search results.
6. Unregistered users shall be able to create an account on Gator Barter by using credentials.
7. Unregistered users shall be able to search for class materials using abbreviated course name and course number.
8. Unregistered users shall be able to initiate pre-posting listings onto the website which shall require registration to complete the listing.

2. Registered users –

9. Satisfy all functional requirements of unregistered users
10. Registered users shall be able to login into their Gator Barter account by using credentials.
11. Registered users shall be able to post their item(s) for sale on Gator Barter website.
12. Registered users shall be able to check the list of their transaction history

13. Registered users shall be able to communicate through messages with other registered users for buying or selling or trading.
 14. Registered users shall be able to purchase/trade products on Gator Barter website.
 15. Registered users shall be able to communicate with the Administrator.
 16. Registered users shall be able to remove the items posted if they no longer wish to sell the item.
 17. Registered users shall be able to mark the item as sold once sold out.
 18. Registered users shall be able to report other users to the Administrator for violating website policies and terms of service
- Administrator –
 19. Administrator shall be able to approve or reject the pending item(s) in the selling queue.
 20. Administrator shall be able to remove inappropriate or illegal item(s).
 21. Administrator shall be able to block suspicious registered users.
 22. Administrator shall be able to contact registered users if required.

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 the 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. 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 in UI.
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 Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application).

Competitive Analysis

Features	Ebay	Amazon	SF State Bookstore	Gator Barter
Search Text / Filter / Sort /	+	+	+	+
Messages	+	+	-	+
Reduced cost for resell items	+	+	-	++
Internal Organizational Use	-	-	++	++
Local Seller	-	-	++	++
Barter System	-	-	-	++
Free Items	-	-	-	+

+ feature exists; ++ superior; - does not exist

Feature Definitions:

- Search Text/ Filter/ Sort: Searching by entering text in search bar, filter the results, sort the results
- Messages: Seller receiving messages from the buyer.
- Reduced cost for resell items: Cost is reduced while reselling the items in comparison with new item's market price
- Internal Organizational Use: Can be privately used inside an organization. The items can be categorized as per the organization, e.g., by 'class'.
- Local Sellers: Sellers who are geographically near the buyer (results in reduced shipping time and no shipping cost)
- Barter System: Provision to exchange items for another item, just like the traditional barter system. This works like a win-win situation by saving money for both the seller & buyer, as well as reduces the cost of utilizing an online payment gateway.
- Free Items: As users of this website are mostly students, who have to vacate their places every semester or every year. Thus, they have a lot of opportunities to donate or post giveaways, and Gator Barter can be used as a platform for the same.

Gator Barter's main goals are to provide SFSU students a platform which provides a local, student-friendly, direct market for specific products geared towards students. The main key feature of our website includes local selling, internal organizational use, barter system and free items. This system is designed for SFSU student use and can be similarly deployed in any other

school or organization campus. Thus, providing the school or organization a secure website for local trade. We believe that having a good organization based on categories would help to cater the buyer or seller, making it easier for students to find what they are looking for. Also, students would benefit from local options available around them to save time and money.

High-Level System Architecture

- Server Host: AWS 1GB RAM, 1vCPU
- Operating System: Ubuntu 18.04 LTS Server
- Database: mysql Ver 14.14 Distrib 5.7.27, for Linux (x86_64) using EditLine wrapper
- Web Server: uWSGI 2.0.18-debian and nginx/1.15.9 (Ubuntu)
- Server-Side Language: Python3.6
- Web Framework: Flask 1.1.1
- Development Environments: PyCharm, VSCode, Sublime, Atom
- SSL Cert: Lets Encrypt (Cert Bot)
- BootStrap: Version 3
- Google Analytics

Team

Name	Role
Aditya Bodi	Team Lead
Alan Ng	Frontend Lead
Alex Kohamin	Backend lead
Akshay Kasar	Git Master, Backend Engineer
Philip Yu	Frontend Engineer
Deming Yan	Full Stack Engineer
Tejasvi Belsare	Document master, Backend Engineer

Checklist

- Team found a time slot to meet outside of class - DONE
- 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

Milestone 2

1. Functional Requirements – prioritized

1.1 Priority 1

- Unregistered Users
 1. Unregistered users shall be able to browse the Gator Barter website.
 2. Unregistered users shall be able to search any item as per their need on the Gator Barter website.
 3. Unregistered users shall be able to search as per Category on the Gator Barter website.
 4. Unregistered users shall be able to see search results in an organized manner.
 5. Unregistered users shall be able to filter search results.
 6. Unregistered users shall be able to create an account on Gator Barter by using credentials.
 8. Unregistered users shall be able to initiate a post for selling an item onto the website which shall require registration to submit the post (lazy registration).
- Registered Users
 9. Satisfy all priority 1 functional requirements of unregistered users.
 10. Registered users shall be able to login into their Gator Barter account by using credentials.
 11. Registered users shall be able to post their item(s) for sale on Gator Barter website and see the item(s) they posted.
 13. Registered users shall be able to send messages for buying/trading.
 14. Registered users shall be able to trade the items with other items on Gator Barter website also can trade items for free.
- Administrator
 19. Administrator shall be required to approve or reject the pending item(s) in the selling queue.
 20. Administrator shall be able to remove inappropriate or illegal item(s).
 21. Administrator shall be able to block suspicious registered users.

1.2 Priority 2

- Unregistered Users
 7. Unregistered users shall be able to search for class materials using abbreviated course name and course number.
- Registered Users

9. Satisfy all priority 2 functional requirements of unregistered users.
15. Registered users shall be able to communicate with the Administrator.
16. Registered users shall be able to remove the items posted if they no longer wish to sell the item.
17. Registered users shall be able to mark the item as sold once sold out.

1.3 Priority 3

- Registered Users
 9. Satisfy all priority 2 functional requirements of registered users
 12. Registered users shall be able to check the list of their transaction history.
 18. Registered users shall be able to report other users to the Administrator for violating website policies and terms of service.
- Administrator
 22. Administrator shall be able to contact registered users if required.

2. UI Mockups and Storyboards (high level only)

2.1 UI Mockups

1. Home Page - This is the default home page for Gator Barter website where any user can browse and search by entering specific name or by category.

The image shows a high-level UI mockup for the Gator Barter website's home page. The header is dark purple with the "Gator Barter" logo in yellow, followed by navigation links: Home, About, FAQ, and Sell. On the right are "First Last" and "Logout" buttons. Below the header is a search bar with a yellow placeholder "Category" and a magnifying glass icon. The main content area has a white background and features three identical item cards. Each card has a large empty box labeled "Image Not Available". Below this is a row with "Item" and "@username" on the left, "\$9.99" and "Tradeable" on the right. A small description follows. At the bottom of each card are two buttons: "More Details" (yellow) and "Contact Seller" (green). A downward arrow is centered below the cards.

"SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only"

Image Not Available	Item @username	\$9.99 Tradeable
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...		
More Details		Contact Seller

Image Not Available	Item @username	\$9.99 Tradeable
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...		
More Details		Contact Seller

Image Not Available	Item @username	\$9.99 Tradeable
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...		
More Details		Contact Seller

2. Search Page - This is the search page returned upon searching for a specific keyword (e.g. iPhone).

The screenshot shows the Gator Barter website's search results for the term "iphone". The top navigation bar includes links for Home, About, FAQ, Sell, and a user profile for Alex Kohanim. A search bar at the top right contains the query "iphone". Below the search bar, a message indicates 2,147,483,647 results found. Three search results are displayed in a grid:

- iPhone 11** by @ajsouza25, \$699.99, Tradeable. Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua... Buttons: More Details (yellow), Contact Seller (green).
- iPhone 11 Pro** by @apuder, \$999.99, Tradeable. Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua... Buttons: More Details (yellow), Contact Seller (green).
- iPhone 11 Pro Max** by @dpetkovic, \$1,099.99, Tradeable. Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua... Buttons: More Details (yellow), Contact Seller (green).

3. View Item Details Page - This is the Item detail page that opens in a new tab upon clicking View Details, where users can look up the details of the item.

The screenshot shows the Gator Barter item details page for a "Limited Edition Batarang" by @brucewayne. The top navigation bar includes links for Home, About, FAQ, Sell, and a user profile for Alex Kohanim. A search bar at the top right contains the query "iphone". The item details include:

- Limited Edition Batarang** by @brucewayne, \$999.99, Tradeable.
- Description: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sit amet mattis vulputate enim nulla aliquet porttitor lacus. Est ullamcorper eget nulla facilisi etiam dignissim diam. Sit amet risus nullam eget felis eget nunc lobortis. Et odio pellentesque diam volutpat commodo sed egestas egestas. Erat velit scelerisque in dictum non consectetur a erat. Nulla facilisi cras fermentum odio eu feugiat pretium. Et egestas quis ipsum suspendisse. Aenean sed adipiscing diam donec adipiscing tristique risus nec. Tincidunt praesent semper feugiat nibh. Cum sociis natoque penatibus et magnis dis parturient montes. Mauris cursus mattis molestie a. Id donec ultrices tincidunt arcu non sodales neque.
- Risus quis varius quam quisque id. Morbi tincidunt ornare massa eget egestas. Tortor vitae purus faucibus ornare suspendisse sed nisi lacus. Pellentesque dignissim enim sit amet. Ac auctor augue mauris augue neque gravida in fermentum.
- Buttons: Cancel (red), Interested (yellow).
- A sidebar on the right shows a "Wishlist" for @brucewayne, listing four items.

4. Post an Item Page - This page allows a registered user to post an item on the Gator Barter website by entering information in the fields such as Title, Price, Category, etc.

The screenshot shows the 'Sell' item listing form. At the top, there's a navigation bar with 'Gator Barter' logo, 'Home', 'About', 'FAQ', 'Sell', and user account links ('First Last', profile icon, 'Logout'). Below the navigation is a search bar with 'Category' dropdown and a placeholder 'Search for books, electronics, furnitures, etc...'. The main form area has fields for 'Item name*', 'Item category*', and a large text area for 'Add a description for the item you are listing...*'. Below that is a photo upload section with a placeholder 'Drag your item photos here' and an 'Or' link to 'Select a photo from your computer'. At the bottom, there's a question 'Would you like to trade items from your wish list for this item?*' with 'Yes' and 'No' radio buttons, followed by 'Cancel' and 'Submit' buttons.

5. Login Page - This is the page for logging in with the credentials for registered users and administrators.

The screenshot shows the 'Login' page. At the top, it has the 'Gator Barter' logo and navigation links. The central login form contains fields for 'SFSU Email Address' and 'Password', a 'Remember Me' checkbox, and a green 'Login' button. Below the login button are links for 'Forgot Password?' and 'Don't have an account? Register'.

6. Registration Page - This is the registration page where unregistered users can create a Gator Barter account by entering details such as email and password.

The registration form is located on the right side of the page. It consists of several input fields and a reCAPTCHA verification. The fields include:

- First Name
- Last Name
- SFSU Email Address
- Password
- Confirm Password

Below the fields are two checkboxes:

- I agree to the [Terms and Conditions](#)
- I'm not a robot reCAPTCHA

A large green "Register" button is centered at the bottom of the form. Below the button, a link says "Already have an account? [Login](#)".

7. Transaction History Page - This is the transaction history page for a registered user which shows all their items posted for sale, previously sold items or previously bought items.

Gater Barter		Home	About	FAQ	Sell	First Last		Logout
Category ▾		Search for books, electronics, furnitures, etc...						
Open	Closed	Dashboard				3 of 3 results found		
		Item Name	Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.					
		Item Name	Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.					
		Item Name	Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.					

8. Admin Dashboard Page - This page is only displayed to administrators who has the privilege to approve or deny items posted by a registered user.

The Admin Dashboard wireframe consists of a header bar and a main content area. The header bar includes the 'Gater Barter' logo, navigation links for Home, About, FAQ, and Sell, and user account information for First Last and Logout. Below the header is a search bar with a dropdown for Category and a magnifying glass icon. The main content area features three cards labeled 'Waitlist', 'Approved', and 'Users'. Each card displays a placeholder image, listing details, and three action buttons: 'Approve' (green), 'More Details' (yellow), and 'Reject' (red). The 'Approved' and 'Users' cards also include a 'Tradeable' badge.

2.2 Storyboards

1. User Case 1:

Nikita is a graduate student from EECS department at SF State. She lives near campus. She needs a PCB board for her Embedded course project. As project deadline is closer, she doesn't have enough time to buy the new board. Buying online would be a slow process considering the shipping time. Also, used PCB board would serve her perfectly from this project's point of view. She needs it within the next 2 days, so that she can cover up her project work and 4 complete milestones on time.

Gater Barter Home About FAQ Sell

Nikita  Logout 

Category  PCB Board 

2,147,483,647 results found for "PCB Board"



DIY PCB Circuit Board \$9.99

@ajsouza25 Tradeable

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...

[More Details](#)

[Contact Seller](#)



PCB Universal Bread \$5.95

@apuder

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...

[More Details](#)

[Contact Seller](#)



Prototype PCB \$19.99

@dpetkovic Tradeable

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...

[More Details](#)

[Contact Seller](#)

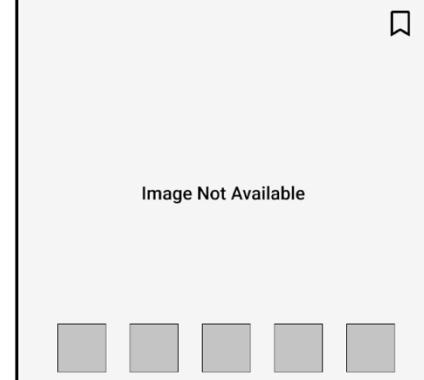
↓

Nikita searches for PCB boards from the home page and is shown a list of PCB boards. Here she can pick and choose from the list of sellers and price. When she has decided, she can click on interested to see a more detailed view.

Gater Barter Home About FAQ Sell

Nikita  Logout 

Cat  



DIY PCB Circuit Board \$9.99

@ajsouza25 Tradeable

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sit amet mattis vulputate enim nulla aliquet porttitor lacus. Est ullamcorper eget nulla facilisi etiam dignissim diam. Sit amet risus nullam eget felis eget nunc lobortis. Et odio pellentesque diam volutpat commodo sed egestas egestas. Erat velit scelerisque in dictum non consectetur a erat. Nulla facilisi cras fermentum odio eu feugiat pretium. Et egestas quis ipsum suspendisse. Aenean sed adipiscing diam donec adipiscing tristique risus nec. Tincidunt praesent semper feugiat nibh. Cum sociis natque penatibus et magnis dis parturient montes. Mauris cursus mattis molestie a. Id donec ultrices tincidunt arcu non sodales neque.

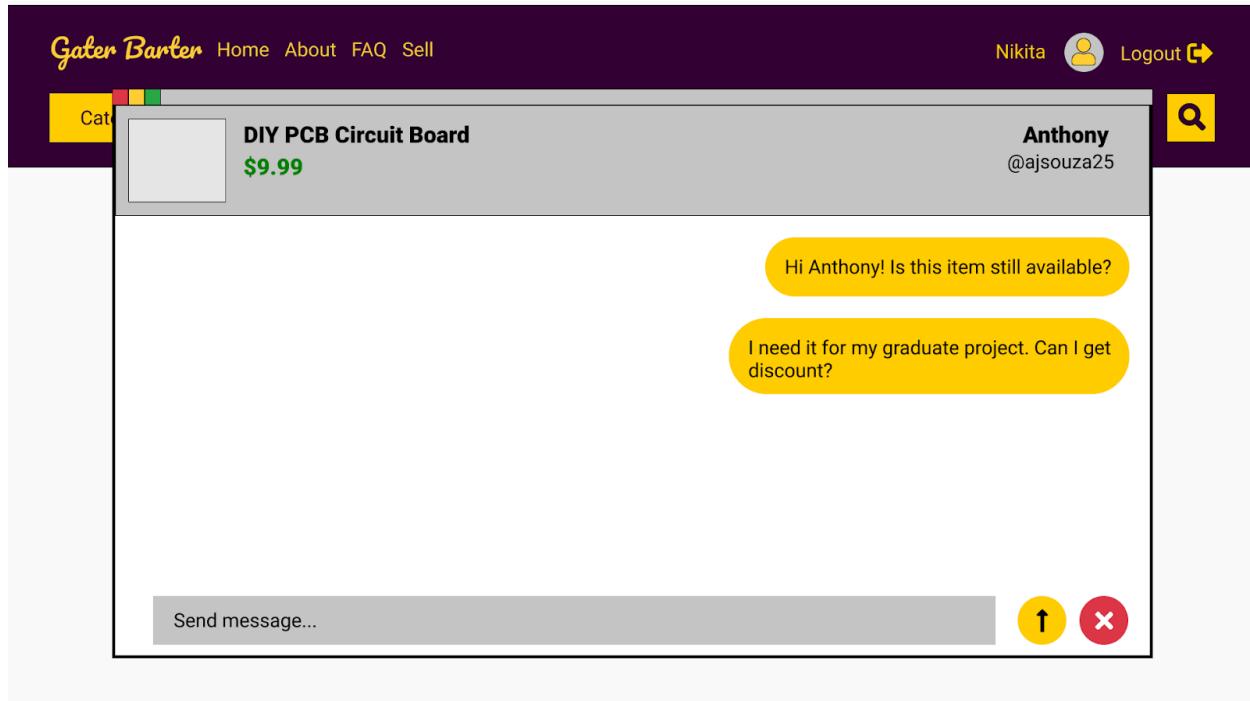
Risus quis varius quam quisque id. Morbi tincidunt ornare massa eget egestas. Tortor vitae purus faucibus ornare suspendisse sed nisi lacus. Pellentesque dignissim enim sit amet. Ac auctor augue mauris augue neque gravida in fermentum.

[Cancel](#) [Interested](#)

@brucewayne's Wishlist

	Item
	Item
	Item
	Item

In this example, Nikita is interested in Anthony's PCB board and clicked interested on his listing. She has a more detailed view on his listed item. She can confirm if she is satisfied with the price and description and then click on interested once again to initiate a chat with Anthony to begin the transaction.



Here she would discuss with Jack if the item is still for sale and finalize the payment method and meeting place.

2. User Case 2:

John is a student from SF State Mathematics department. He recently graduated and got a great job offer in Texas. Now, he is moving out of SF State dorm and wants to sell his room items quickly before moving. Selling online would be a slow process. It would save a lot of time if he can sell those items to someone from the dorms or SF State itself.

The following is the page displayed when the user posts an item to sell.

Gater Barter Home About FAQ Sell John Logout

Category ▾ Search for books, electronics, furnitures, etc...

Small Dorm Table

Dorm ▾

Please buy table! Everything must go!

Drag your item photos here
— OR —
Select a photo from your computer

Would you like to trade items from your wish list for this item?

Yes
No

Cancel **Submit**

Gater Barter Home About FAQ Sell John Logout

Category ▾ Search for books, electronics, furnitures, etc...

Image N

available

Item has been submitted. Pending for approval.

OK

\$9.99 Tradeable

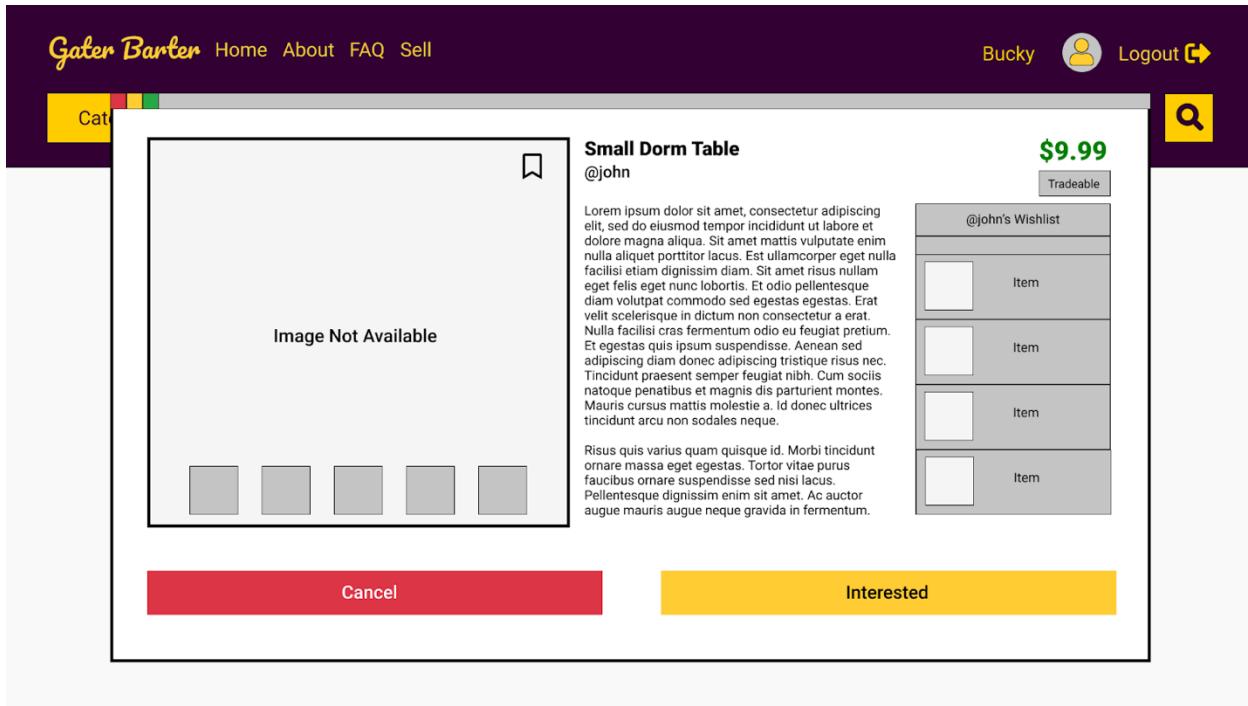
Item @username
Lorem ipsum dolor sit elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua...

More Details Contact Seller

dolore magna aliqua... More Details Contact Seller

dolore magna aliqua... More Details Contact Seller

↓



3. User Case 3:

Bucky is a senior student from the Computer Science Department. He lives in San Jose and is on campus 4 days a week. Bucky has few CS course-related textbooks, which he doesn't need once he graduates. He wants to trade those readers/textbooks with some other stuff required for his new room. As course material is the same for courses, it would be good if someone from SF State CS department can trade those textbooks. As Bucky comes frequently to the school, he can trade things with the students with an ease.

Gater Barter Home About FAQ Sell Bucky  Logout 

Category ▾ Search for books, electronics, furnitures, etc... 

CSC 600 Course Reader

Books ▾

This item is for the course CSC 600. Please buy!

 Drag your item photos here
— or —
Select a photo from your computer

Would you like to trade items from your wish list for this item?

Yes
No

Cancel **Submit**

Gater Barter Home About FAQ Sell Bucky  Logout 

Category ▾ Search for books, electronics, furnitures, etc... 

Image N

available

Item has been submitted. Pending for approval.

OK

\$9.99 Tradeable

Item
@username
Lorem ipsum dolor sit
elit, sed do eiusmod te
dolore magna aliqua...

dolore magna aliqua...

More Details Contact Seller

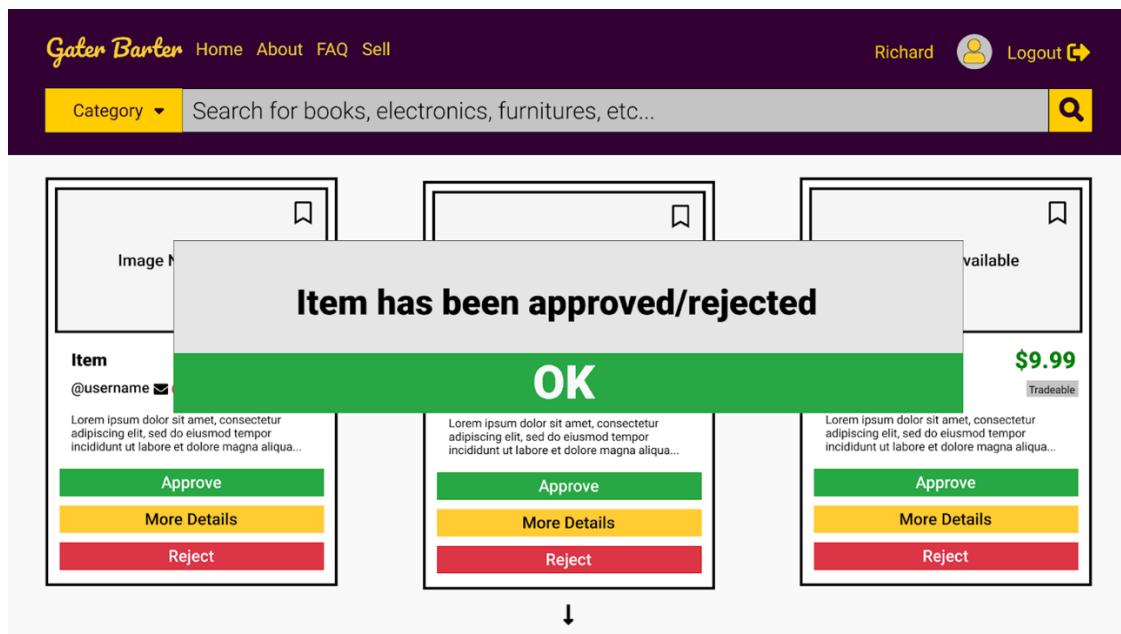
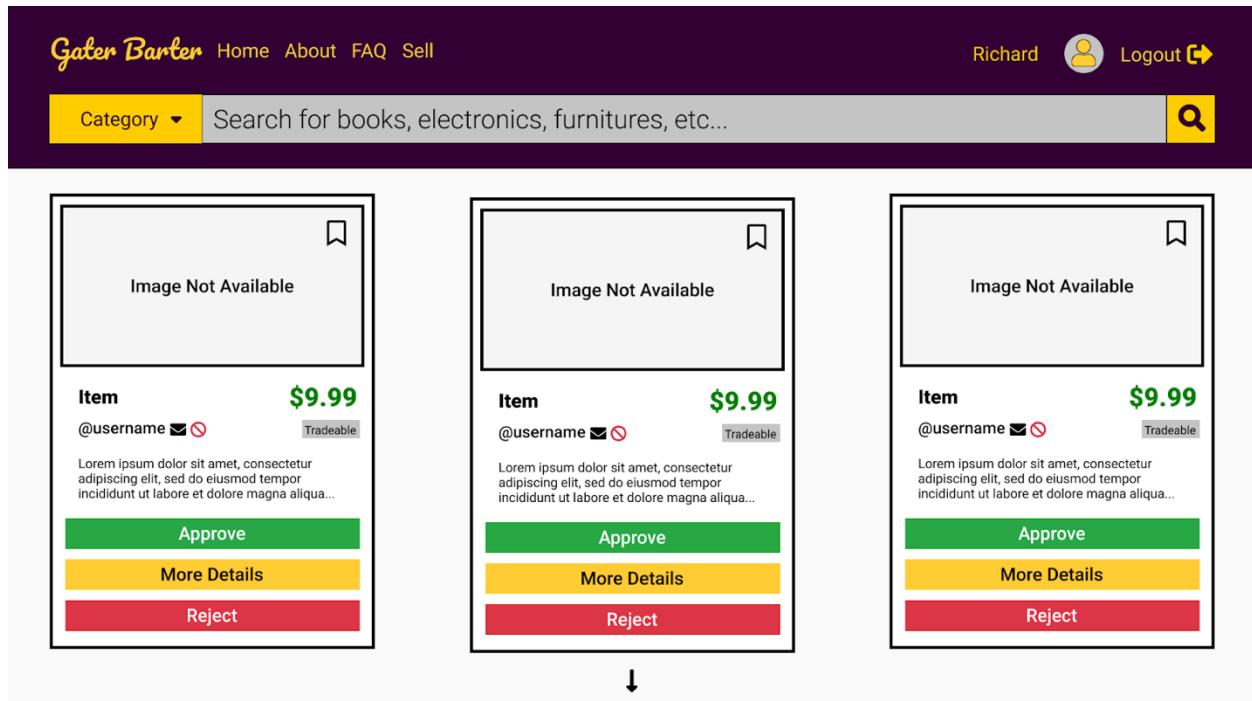
More Details Contact Seller

More Details Contact Seller

↓

4. User Case 4:

Richard is an Administrator on the Gator Barter website. He supervises the website and has the right to approve or reject the items posted by registered users.



3. High level Architecture, Database Organization

3.1 Database Organization

Database will contain below tables –

1. user - This is the user registration table when an unregistered user registers on Gator Barter.

It will have following data fields –

```
`u_id` bigint NOT NULL AUTO_INCREMENT  
`u_email` varchar(128) NOT NULL UNIQUE  
`u_pass` TEXT(256) NOT NULL  
`u_is_admin` int NOT NULL DEFAULT '0'  
`u_created_ts` TIMESTAMP  
`u_updated_ts` TIMESTAMP  
`u_fname` varchar(128) NOT NULL  
`u_lname` varchar(128) NOT NULL  
`u_status` int NOT NULL DEFAULT '1'  
PRIMARY KEY (`u_id`)
```

2. category – This table categorizes item. Whenever a registered user posts an item, he has to select the category for an item. Category selection would be mandatory for sellers and they would not be allowed to enter his choice for category.

It will have the following data fields –

```
`c_id` bigint NOT NULL AUTO_INCREMENT  
`c_name` varchar(128) NOT NULL  
`c_status` int NOT NULL  
PRIMARY KEY (c_id)
```

3. item –

```
`i_id` bigint NOT NULL AUTO_INCREMENT  
`i_title` varchar(256) NOT NULL  
`i_desc` TEXT  
`i_price` double NOT NULL DEFAULT '0'  
`i_is_tradable` int NOT NULL DEFAULT '0'
```

```
`i_u_id` bigint NOT NULL  
`i_created_ts` TIMESTAMP NULL DEFAULT NULL  
`i_updated_ts` TIMESTAMP NULL DEFAULT NULL  
`i_sold_ts` TIMESTAMP NULL DEFAULT NULL  
`i_status` int NOT NULL DEFAULT '1'  
`i_c_id` bigint NOT NULL DEFAULT '0'  
PRIMARY KEY (`i_id`)  
FOREIGN KEY (`i_u_id`) [u_id from user table]  
FOREIGN KEY (`i_c_id`) [c_id from category table]
```

4. item_image

```
`ii_id` bigint NOT NULL AUTO_INCREMENT  
`ii_image` blob  
`ii_url` TEXT  
`ii_i_id` bigint NOT NULL  
`ii_status` int NOT NULL DEFAULT '1'  
PRIMARY KEY (`ii_id`)  
FOREIGN KEY (`ii_i_id`) [i_id from item table]
```

5. message

```
`m_id` bigint NOT NULL AUTO_INCREMENT  
`m_text` TEXT  
`m_sender_id` bigint NOT NULL  
`m_receiver_id` bigint NOT NULL  
`m_item_id` bigint NOT NULL  
`m_item_id` bigint NOT NULL  
`m_sent_ts` TIMESTAMP NULL DEFAULT NULL,  
PRIMARY KEY (`m_id`)
```

3.2 Media Storage

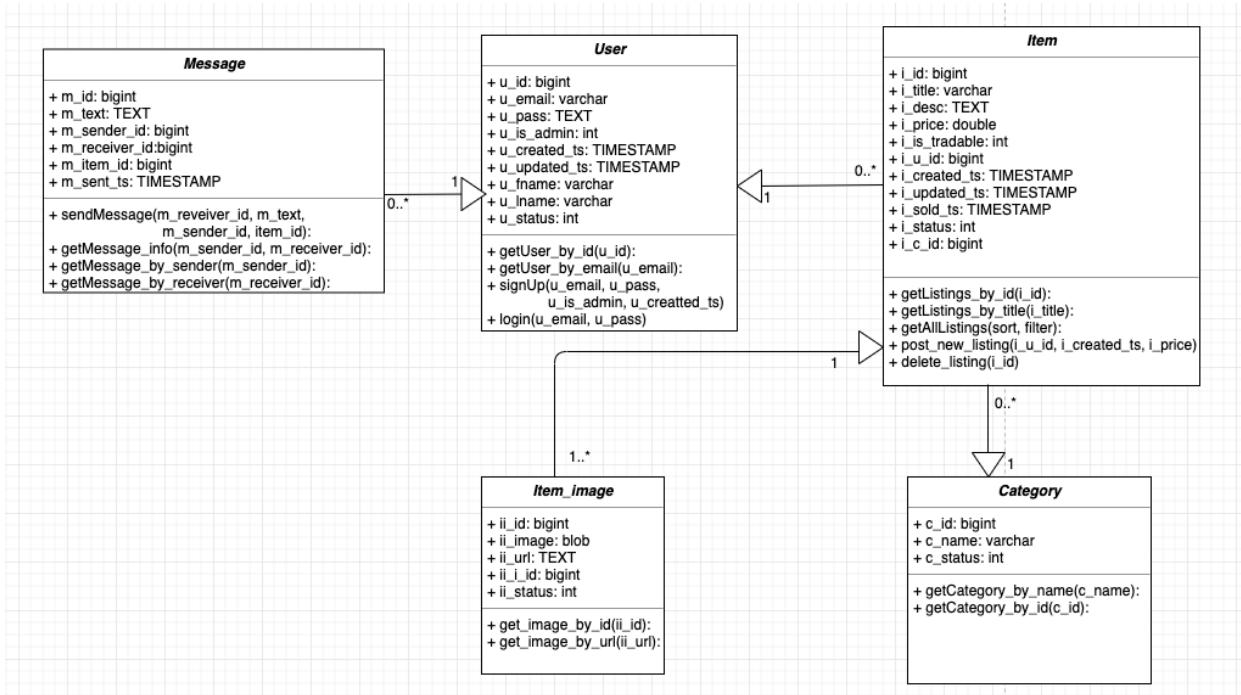
We shall use DB URLs for media storage. Standard images shall be stored locally. The server shall hold chat sessions locally up to some amount (1 GB tentatively). We shall create a thumbnail version of each item's image in order to speed up the load times of the front page.

3.3 Search/Filter Architecture and Implementation

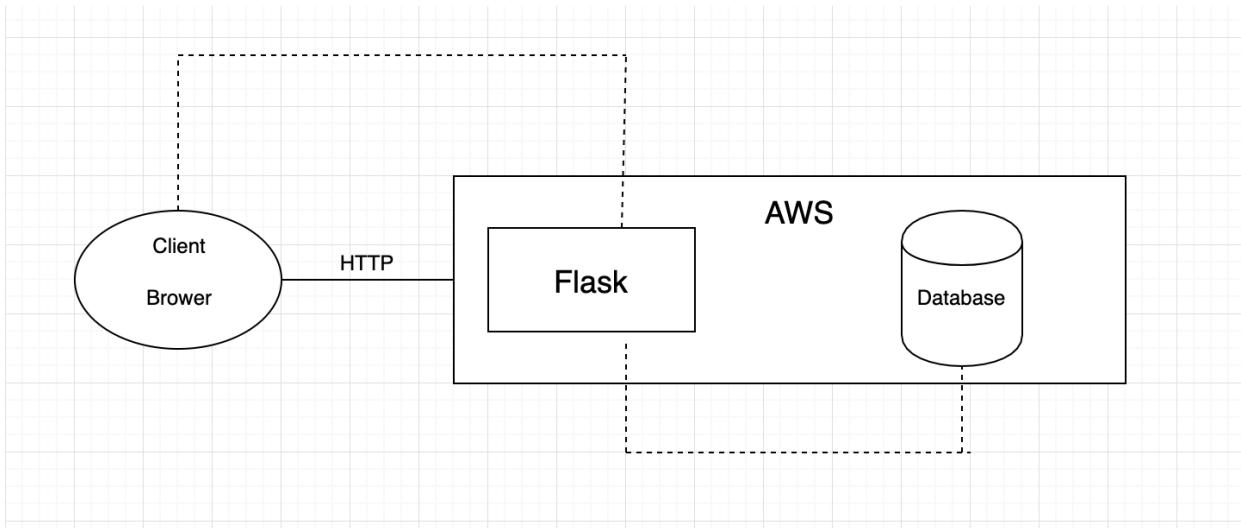
Our website will include a search box where a user can search by text typed in search field or by category. The user can search by some keywords such as item name, etc. which would then display the specific listings according to the keyword using %LIKE MYSQL query.

4. High Level UML Diagrams

4.1 UML Diagram



4.2 Component and Deployment Diagram



5 . Identify Actual Key Risks for Your Project at this Time

5.1 Skills Risks

Our team has development experience but have rarely designed such a buying/selling website completely. In this case, team members would study on their own and Vertical Prototype would serve as learning tool to all the team members. We would restrict the scope to priority 1 tasks. Developing this website would be an interesting and challenging environment for us to work end to end.

5.2 Schedule Risks

We have completed all the milestones till now with proper team meetings and communication between our team members. We would focus on all priority 1 tasks. Also, our team has set a Trello account for project management with all the team members. Hence, this part should not be challenging for us.

5.3 Technical Risks

The technical risk was associated with the git push /pull/ merge branches and the server deployment. However, we have identified a GitHub master who will handle the GitHub conflicts and also, a server master for deploying the working code remotely. Also, the vertical prototype will help as a learning tool for an end to end development and would resolve technical risks if any at an early stage of the development process.

5.4 Teamwork Risks

Our team works well together, and we have good communication between the backend and frontend team. There can be some risks if one of the team members is unavailable during crucial time and the dependency is high on that member. Our team lead takes care that everyone is on the same page.

5.5 Legal/Content Risks

If the site gets deployed publicly, the website shall prominently display the following text on all pages “SFSU- Software Engineering Project CSC 648-848, Fall 2019 - For Demonstration Only” in order to avoid any conflicts later.

6. Project management

We use Trello to help organize development tasks. All tasks are explained by the Team Lead and then delegated to the front-end engineers and back-end engineers. Front-end team lead takes the responsibility of completion of all front-end tasks and back-end team lead takes the responsibility of completion of all back-end tasks. For communication among team members, we use discord (a team communication platform). We have organized channels in discord as per the sub-teams like the front end, back end, documentation, database. For the documentation part, we use google docs so that everyone can work on their assigned tasks at one place.

The Trello board 'Team04' contains the following cards:

- To-Do-List**:
 - Team meeting time: Wednesday 3:00pm to 6:00pm
 - Task 1 - Github branches
 - Task 3 - workflow, db schema, mockups
 - Task 4 - M2 document review
 - FE task - ALignment issues
 - M3-Add dashboard page
 - M3-Message
 - M3-Admin page
- In Progress**:
 - M2 tasks:
 - M1 documentation and discussion
 - M2 documentation
 - Questions for professor:
- Review**:
 - [Back-End] Configure Flask with routes for static files for root user.
- Links**:
 - Document: <https://docs.google.com/document/d/1rp6lnyW4BgLCnGDXCfrqBdZd5JxV8X5GQsgGA9Oxro/edit>
 - Github: <https://github.com/CSC-648-SFSU/csc648-fall2019-Team04>
 - Milestone 1 Documentation https://docs.google.com/document/d/1uiRapkM7n4l8j6pxo67cwV9pTjFTLV4wWPnAr3_hzI/edit
 - M2 <https://docs.google.com/document/d/1gEGbfPT2Rlzv1zrAOQ70TAXdbOE33cf02e3aE6BMTYIA/edit>
- Completed**:
 - + Add a card

The conversation in the '# design-discussion' channel:

- Tejasvi** Last Sunday at 1:40 AM
ok
I am mentioning user email id as UNIQUE in the document, @Alex Kohanim or @Akshay Kasar make changes accordingly in the SQL
- Alex Kohanim** Last Sunday at 1:41 AM
It will be unique in the final draft.
*it shall
- Tejasvi** Last Sunday at 1:43 AM
Also message table is not there
- Aditya Bodhi** Last Sunday at 1:44 AM
If we plan to replace a category, old category status will be made 0 instead of deleting it from the db, and the categories shown to the users will be 1.
- Tejasvi** Last Sunday at 1:44 AM
okay

Milestone3

1. Summary of Feedback and Tasks to do

During the milestone3 meeting, we have got great feedback from the professor. Before the meeting, we took the hard copies of the screenshots of our web pages to get feedback on our User Interface (UI). We made sure that our website is hosted and running on a remote server.

During the meeting, the professor checked our website, surfed through different web pages and gave us feedback and improvements. Some of these improvements were like to modify the search bar, navigation bar, adding some information. We made a list of all the necessary improvements and the team lead assigned tasks accordingly. Secondly, the professor and our team discussed P1 tasks. The ‘Wishlist’ option has been removed from our P1 list as this is a bit complicated to implement and is not so easy for a user to understand. Next, our database was reviewed, and we got neutral feedback with no improvements required. Later, the professor checked at our code and appreciated the comments and messages. Lastly, our GitHub was reviewed, and we got good feedback from the professor for the comments and commits.

The main change the professor mentioned was the search functionality. As we didn’t have proper persistent search functionality, our professor has given a time period of 5 days to make necessary improvements. The team would be priory working on this.

It was a great experience for us to learn from this meeting. We found many improvements we didn’t pay attention to. At last, we had a team meeting with the whole right after the meeting for M3 with the professor. We have discussed the necessary improvements and made a list of tasks to do based on the feedback. In addition, we made a plan for what to focus on in the next step.

2. List of tasks the team chose to focus on and implement

2.1 Front-end to-do list:

2.1.1 Home-page:

1. Navigation bar connected with search bar like Amazon
2. Make "SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only" small and shift to the top of the page
3. Add a welcome message just below the navigation bar

4. Interchange sort and “Showing results” message position
5. Narrow white space between the item images and the item description
6. Reduce image size
7. Remove ‘FAQ’ in the navigation bar
8. Shift ‘Sell’ to the left in the navigation bar
9. Make images clickable
10. Remove Wishlist

2.1.2 Item-details page

1. If the images are not available, do not display them as not ‘N/A’
2. Make ‘Contact Seller’ button small

2.1.3 Messaging page

1. Change the text
2. Add information about the item like item-title and item-price
3. Make the buttons small and shift the ‘cancel’ button to the left

2.1.4 Item-posting page

1. Include page title
2. Make the columns short
3. Shift the ‘Submit’ button to the right and make it yellow color
4. Include red asterisks as required to all the fields

2.1.5 Login/register page

1. Make the window size small
2. Make it only one column
3. Include page title

2.1.6 User Dashboard

1. Add 'Back' button

2.2 Back-end to-do list:

1. Improve the search feature;
2. Add count how many items for the searching result, email professor by Friday
3. List of P1 functions agreed at the meeting
 1. Home page;
 2. Search + result + details;
 3. Trading;
 4. Filter and sorting;
 5. Message;
 6. Dashboard for seller, and admin;

Milestone 4

Product summary

Name of our product – Gator Barter

Below is the list of all major committed functions for Gator Barter:

1. Home Page
2. Search results (with search text validation)
3. Filter search by category
4. Sort the search results
5. Item details on separate page including information like images, description, price and seller
6. User Login
7. User Registration (including lazy registration)
8. Item posting (with all fields validation)
9. Messaging from a user to other user
10. Item trading (unique feature)
11. Seller dashboard (including seller items and messages)
12. Admin dashboard (to approve or reject the items posted for selling)

Gator Barter is a buying and selling website specially designed for San Francisco State University students. Gator Barter will be marketed as a website created by SFSU students for SFSU students. This website would serve as a platform for students to sell their items, class materials, etc. as well as buy essential items at a cheaper rate compared to the market price that too within campus. Students can also give away items free of cost on the site or trade their items with other items. It will provide an affordable and safe marketplace for students to exchange their belongings.

Moreover, this website builds direct communication between buyer and seller. This unique approach will eliminate long wait times and no shipping costs would be involved. The basic attraction of our website is its simplicity and user-friendliness. Also, the good design of Gator Barter can be eventually used as a template for other universities looking to design their own buy and sell website.

URL to our website: <https://gatorbarter.store/>

Usability test plan

Test Objective

The goal of usability testing is to identify any usability problems, collect quantitative data on participants' performance as well as determine user satisfaction for the website. If a website is difficult to navigate or doesn't clearly articulate a purpose, users will leave it. Ideally, usability testing is done before a site launches, and then many times after. Currently, our website is in its development and testing phase. Hence, usability testing will help us to find out the flaws or weaknesses in the usability flow at an early stage. User comments or reviews will help us to judge and improve our website in terms of usability.

We will be conducting a usability test for searching item(s) on our website. This will allow the user to navigate through a series of usage flows touching on the main components of the website. Also, test users will be asked to fill a survey consisting of a Likert scale. Success factors would then be determined based on feedback from the test users.

Test Background and Setup

Purpose:

The purpose of this test is to identify any usability problems within the search functionality of our website.

System setup:

A laptop or a mobile with an internet connection is required and access to web browsers like chrome (v77), firefox (v69). We will be testing our website on the mentioned two versions and the latest versions of these browsers on laptops and mobile in order to check the consistency while using our website on different browsers.

Starting point:

The starting point will be the homepage of our website located on URL:
<https://gatorbarter.store/>

Intended Users:

The intended users would be SFSU students.

URL for the system/function for Testing:

The URL for the search function is same as the home page/starting point of our website:
<https://gatorbarter.store/>

Task Description:

There are different tasks that the user needs to go through in order to successfully searching an item and then view more details of that item. This consists of searching and selecting the item they are interested in buying.

Task	Machine State	Successful completion Criteria	Benchmark
Open Gator Barter website on browser	Loaded URL	Homepage of the website displayed	
Choose an appropriate category if required	Appropriate category is highlighted in the 'categories drop-down'	Category Selected	
Searching an item	Loaded the results page	Filled out the search field and searched for an item for purchase	

Measure effectiveness:

1. Check that the task is able to be performed and completed
2. Measure the number of people who completed the tasks
3. One suggested way to implement:

Test/use case	% completed	errors	comments
Search	100	0	

Measure efficiency:

1. Test Efficiency in time: average time to complete the task, the average time of those who completed the task/av. time of all users
2. Test Efficiency in the effort: number of clicks
3. Test Efficiency in content/design: number of screens, Number of clicks, and number of pages of instructions

Completion Criteria:

User has successfully completed the following tasks: -

1. Launched the website URL on the browser.
2. Found the Login button and Logged in successfully.
3. Found the search button and searched for an item successfully.

User Satisfaction Questionnaire:

Website user survey. Select one option for below questions –

Task	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The GUI is simple and user friendly.	✓				
It was easy to search an item for sale on Gator Barter.			✓		
It was easy to locate the desired item from searching		✓			
Item details were easy to understand when displayed in more details.			✓		

General Comments/Feedback:

The website is easy to understand and easy to use

QA test plan

Test Objective

Quality assurance (QA) is a systematic process of verifying whether a software meets required specifications and customer expectations. The objective of below QA tests would be to test the functionality of searching for an item and viewing it on Gator Barter website, find bugs and resolve them prior to product release.

HW and SW setup

1. A laptop with an internet connection is required and access to any of the web browsers among Google Chrome, Mozilla Firefox.
2. Browse to the website using the following URL. This will direct to the homepage of our website.
3. URL: <https://gatorbarter.store/>
4. Test the following cases for searching items.

Feature to be tested: Search an item for different categories.

Test Plans:

Test 1:

Input: Choose “Furniture” in the Category drop down menu then, enter “couch” in the search text field. Output: Check the results consist of exactly 2 couch results

Test 2:

Input: Keep “All” in the Category drop down menu then, enter “mouse” in the search text field. Output: Check the results consist of exactly one mouse result

Test 3:

Input: Choose any category from Category drop down menu and search for an item that does not exist in the database

Output: Feedback saying no results, but suggest other products from the website

Firefox

Number	Description	Test input	Expected output	Pass/Fail
1	test % like in search for name field in specific category	Choose “Furniture” in the Category drop down menu then, enter “couch” in the search text field.	get 2 results, all have “couch” in name field	Pass
2	test % like in search for name field	Keep “All” in the Category drop down menu then, enter “mouse” in the search text field.	get 1 result with “mouse” in name field	Pass
3	test search for item not in database	Choose any category from Category drop down menu and enter “dog” in	0 results from query, but get suggestions.	Pass

		the search text field.		
--	--	------------------------	--	--

Google Chrome

Number	Description	Test input	Expected output	Pass/Fail
1	test % like in search for name field in specific category	Chose “Furniture” in the Category drop down menu then, enter “couch” in the search text field.	get 2 results, all have “couch” in name field	Pass
2	test % like in search for name field	Keep “All” in the Category drop down menu then, enter “mouse” in the search text field.	get 1 result with “mouse” in name field	Pass
3	test search for item not in database	Chose any category from Category drop down menu and enter “dog” in the search text field.	0 results from query, but get suggestions.	Pass

Code Review

Coding Style:

- Use of PEP8 python coding style has been targeted.
- MVC architecture has been followed in the project

- Implementation of flask-blueprints framework had also been used to make the code modular and reusable.

Code Review Example:

Code Review screenshot of an email:

@AlexKohanim requested changes on this pull request.

Please fix these issues before we can merge with full-stack

In [application/helperFunctions.py](#):

```
> @@ -0,0 +1,133 @@
+<!-- This module contains or the utility/helper functions which can be reused later.

+from dbCursor import getCursor
+from queries import query
+import gatorProduct as product
+
+from flask import session
+import time
+import os
+from werkzeug.utils import secure_filename # for input picture loading
+
+
+db = getCursor()[0]
+
+ALLOWED_EXTENSIONS = set(['pdf', 'png', 'jpg', 'jpeg'])</pre>
```

Please remove pdf as we only accept image files.

In [application/dbCursor.py](#):

```
> @@ -0,0 +1,5 @@
+<!-- This module contains functions for initializing connection to the MySQL database
+import pymysql
+db = pymysql.connect('0.0.0.0', 'root', None, 'gatorbarter')</pre>
```

Instead of literal values for DB config, read from another class or from a read-only file

—
You are receiving this because you authored the thread.

Reply to this email directly, [view it on GitHub](#), or [unsubscribe](#).

Following are the same steps on the Github review request page:

AlexKohanim requested changes 1 hour ago

AlexKohanim left a comment

Please fix these issues before we can merge with full-stack

application/helperFunctions.py Outdated

```
10 +  
11 + db = getCursor()[0]  
12 +  
13 + ALLOWED_EXTENSIONS = set(['pdf', 'png', 'jpg', 'jpeg'])
```

AlexKohanim 1 hour ago
Please remove pdf as we only accept image files.

akshaypkasar 2 minutes ago Author
Made the change and pushed the fix!

Reply...

Resolve conversation

application/dbCursor.py Outdated

```
... ... @@ -0,0 +1,5 @@  
1 + import pymysql  
2 + db = pymysql.connect('0.0.0.0', 'root', None, 'gatorbarter')
```

AlexKohanim 1 hour ago
Instead of literal values for DB config, read from another class or from a read-only file

akshaypkasar 1 minute ago Author
Made the change and pushed the fix!

Reply...

Self-check on best practices for security

List of Major Assets:

1. User Database
2. System Data
3. Images
4. AWS Server
5. Sales data

Protection Description:

1. User Database - We are sanitizing the user's input for the items to be fetched from the database. This prevents the SQL injection.
2. System Data- We are using 'https' protocol to have secure requests and protect our system data.
3. Images - We are accepting only valid image types and we have a system in place to never execute the images
4. AWS System - We are using cronjobs to monitor various system errors and executing appropriate services.
5. The password of a user account is hashed/encrypted before updating the user information in the database.
6. Any input data is validated to prevent any SQL injection attacks. We have used flask-bleach in the backend to validate any input data.
7. Any input data validation to prevent cross browser scripting for the search field.

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 MO (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). - ON TRACK
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. Selected application functions must render well on mobile devices - ON TRACK
4. Data shall be stored in the team's chosen database technology 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. - DONE
7. The language used shall be English. - DONE
8. Application shall be very easy to use and intuitive. - DONE
9. Google analytics shall be added - DONE
10. No e-mail clients shall be allowed - DONE
11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. - DONE
12. Site security: basic best practices shall be applied (as covered in the class) - DONE
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development - DONE
14. The website shall prominently display the following exact text on all pages "*SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only*" at the top of the WWW page. (Important so as to not confuse this with a real application). - DONE

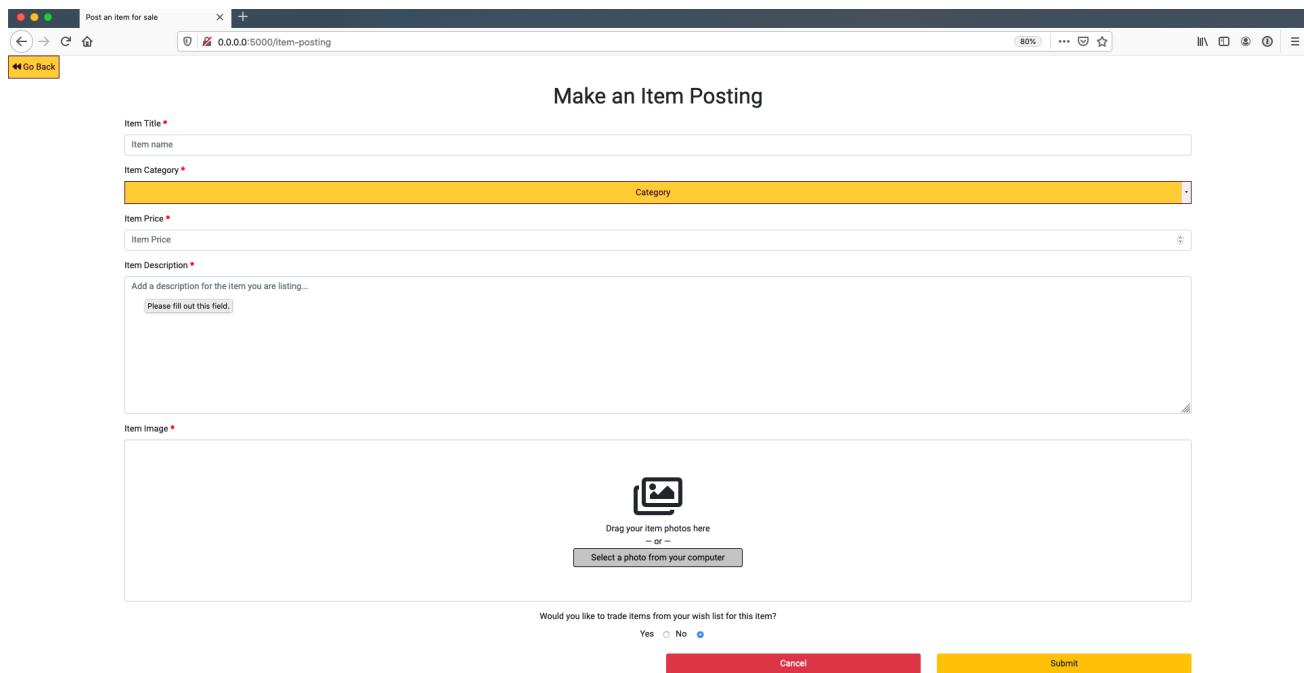
ScreenShots of the actual final product as shown in the demo

Home Page:

The screenshot shows a Firefox browser window displaying the Gator Barter website. The title bar reads "Firefox File Edit View History Bookmarks Tools Window Help" and "Gator Barter". The address bar shows "0.0.0.0:5000". The page header includes "Gator Barter" with "Sell" and "About" links, a "Login" button, and a search bar with placeholder text "Search for books, electronics, furnitures...". Below the header, a message says "Here are the latest Items Showing 1-5 of 5 results". The main content area has a heading "Home Welcome to Gator Barter!". It displays five items in a grid:

- Globe Title** \$9.99
@Joe_Smith
Category: Other
Globe Description Other
[More Details](#)
- Black Couch Title** \$9.99
@Joe_Smith
Category: Furniture
Black Couch Description Furniture
[More Details](#)
- Laptop Title** \$9.99
@Joe_Smith
Category: Electronic
Laptop Description Electronic
[More Details](#)
- 
- 

Post Page:

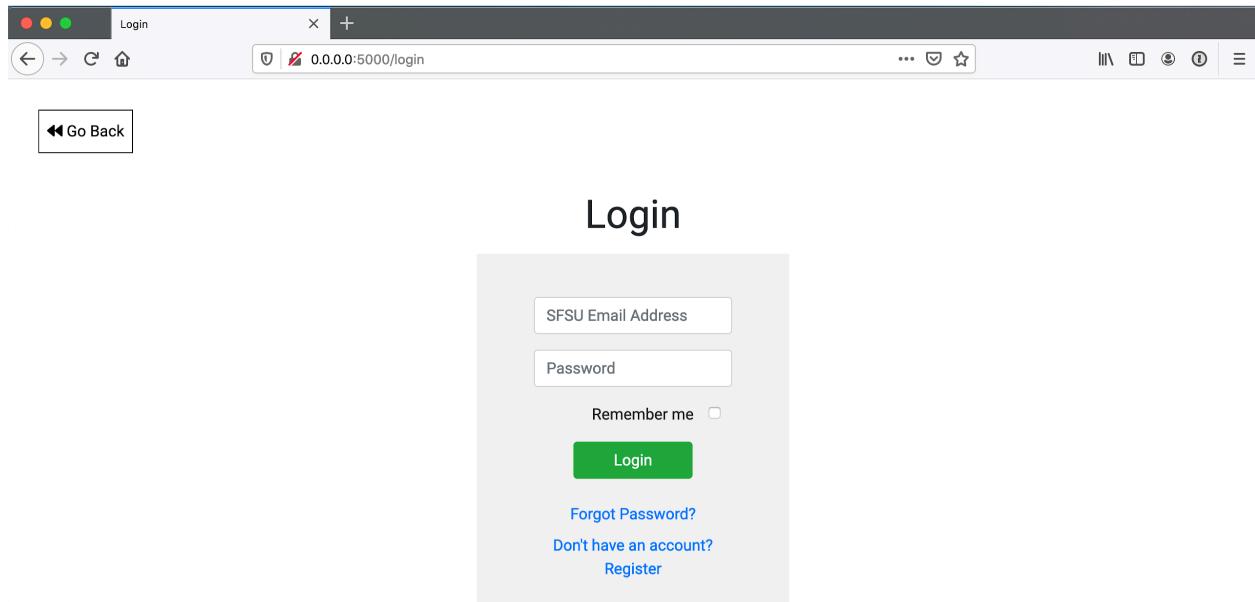


The screenshot shows a web browser window titled "Post an item for sale" with the URL "0.0.0.0:5000/item-posting". The main title is "Make an Item Posting". The form fields include:

- Item Title * (text input)
- Item name (text input)
- Item Category * (dropdown menu)
- Item Price * (text input)
- Item Description * (text area)
Add a description for the item you are listing...
Please fill out this field.
- Item Image * (file upload area)
Drag your item photos here
— or —
Select a photo from your computer

At the bottom, there is a question: "Would you like to trade items from your wish list for this item?". The "No" radio button is selected. Below the form are two buttons: "Cancel" (red) and "Submit" (yellow).

Login Page:



The screenshot shows a web browser window titled "Login" with the URL "0.0.0.0:5000/login". The main title is "Login". The form fields include:

- SFSU Email Address (text input)
- Password (text input)
- Remember me
- Login (green button)

Below the form are links: "Forgot Password?", "Don't have an account?", and "Register".

Register Page:

The screenshot shows a registration form titled "Registration". It includes fields for First Name, Last Name, Email Address, SFSU Email Address, Password, Confirm Password, and a checkbox for agreeing to Terms and Conditions. Below these is a CAPTCHA challenge asking "Are you human?" with the question "What is 2 * 9?". A "Register" button is at the bottom, and a link to "Login" is below it. A note above the CAPTCHA states: "Password must contain the following:
* A lowercase letter
* A capital (uppercase) letter
* A number
* Minimum 8 characters".

First Name * Last Name *

Email Address * SFSU Email Address

Password * Confirm Password *

I agree to the [Terms and Conditions](#)

Are you human?

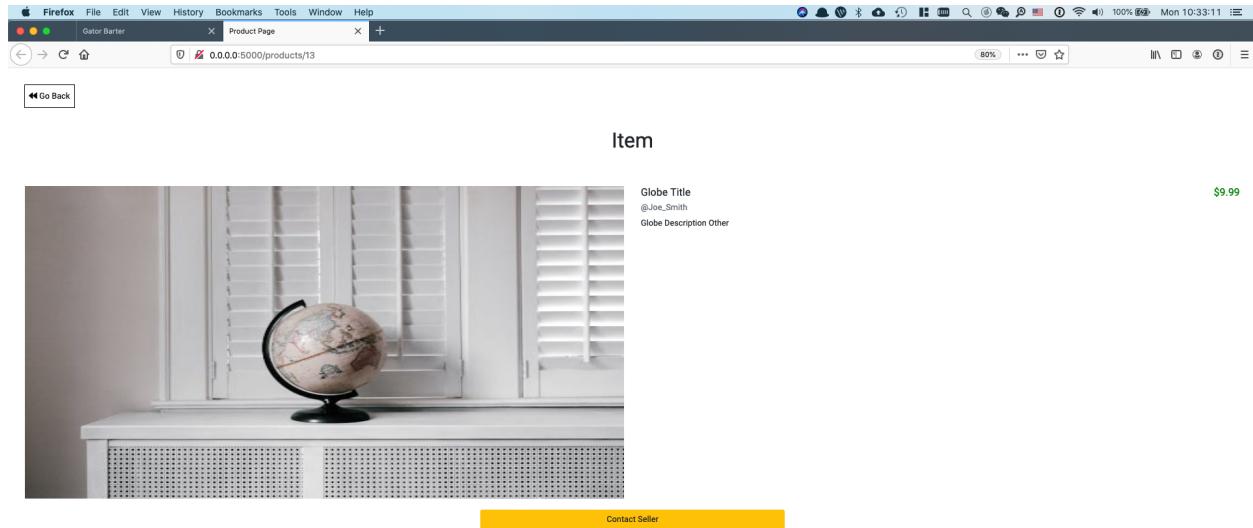
What is $2 * 9$?

Already have an account? [Login](#)

Password must contain the following:

- * A lowercase letter
- * A capital (uppercase) letter
- * A number
- * Minimum 8 characters

Product page



Contact Seller Page:

The screenshot shows a web browser window with the title "Gator Barter" at the top. The main content area is titled "Contact Seller". It contains two input fields: "Contact Method" (with placeholder "Enter your phone number or email address...") and "Message" (with placeholder "Add a message to the seller..."). Below the message field is a note: "Please fill out this field." At the bottom right are two buttons: "Cancel" (red) and "Send Message" (yellow).

About Page:

The screenshot shows a web browser window with the title "Gator Barter" at the top. The main content area is titled "About". It displays the following text:
Software Engineering Class SFSU
Fall 2019
Section
Team 4
Below this, under "ABOUT US", is the heading "Team Four Star". A list of team members' names is displayed in yellow rectangular boxes:
Aditya Bodhi
Deming Yan
Akshay Kasar
Phillip Yu
Alan Ng
Alex Kohanim
Tejasvi Belsare
At the bottom of the page, a dark footer bar contains the text "Copyright © 2019 Gator Barter".

Personal Page:

The screenshot shows a Firefox browser window with the following details:

- Title Bar:** Firefox - Gator Barter
- Address Bar:** 0.0.0.0:5000/about/abodi
- Page Title:** "SFSU Software Engineering Project CSC 648-848, Fall 2019. For Demonstration Only"
- Header:** Gator Barter, Sell, About, Login
- Search Bar:** All, Search for books, electronics, furniture...
- Profile Section:** A circular profile picture of a man (Aditya Bodhi) in a suit.
- Name:** Aditya Bodhi
- Title:** Team Lead
- Description:** I am a Computer Science Graduate student at San Francisco State University. I have worked in Database Administration domain for 1.5 years at Cognizant Technology Solutions. I will be leading the Software Engineering team to build a website for an online market place for SF state students.
- Icons:** LinkedIn, GitHub, YouTube
- Footer:** Copyright © 2019 Gator Barter

Screenshots of key DB tables

user table

	u_id	u_email	u_pass	u_is_admin	u_created_ts	u_updated_ts	u_fname	u_lname	u_status
▶	1	jsmith@mail.sfsu.edu	\$5\$rounds=535000\$nuLafy8TPKQf4vzbicAMJ5...	0	HULL	HULL	Joe	Smith	1
	2	jstevens@mail.sfsu.edu	\$5\$rounds=535000\$BLT7xqCfQf2pmB9Xiv...	0	HULL	HULL	Jane	Stevens	1
	3	alederson@mail.sfsu.edu	\$5\$rounds=535000\$6vzQgndrx5TnBw\$2mH6...	1	HULL	HULL	Elliot	Alderson	1
	4	jhenry@mail.sfsu.edu	\$5\$rounds=535000\$TyerS9mlQ1L26GUuQ0T...	0	HULL	HULL	Jack	Henry	1
	5	sadams@mail.sfsu.edu	\$5\$rounds=535000\$kd0xDJswernAoOs9Wp3...	0	HULL	HULL	Sam	Adams	1
	6	rhendricks@mail.sfsu.edu	\$5\$rounds=535000\$DmxKoHrrnX9Jptfzw...	1	HULL	HULL	Richard	Hendricks	1
	7	sample@mail.sfsu.edu	\$5\$rounds=535000\$H9h+12ekfEHj6Y3BJMZ...	0	2019-12-02 19:08:37	2019-12-02 19:08:37	sample	testing	1

item table

	i_id	i_title	i_desc	i_price	i_is_tradable	i_u_id	i_created_ts	i_updated_ts	i_sold_ts	i_status	i_c_id
▶	1	Computer Keyboard Title	Computer Keyboard Description Electronic	9.99	0	1	HULL	HULL	HULL	0	1
	2	Computer Monitor Title	Computer Monitor Item Description Electronic	9.99	0	1	2019-11-13 19:13:30	HULL	HULL	1	1
	3	Computer Mouse Title	Computer Mouse Description Electronic	9.99	0	1	HULL	HULL	HULL	1	1
	4	Laptop Title	Laptop Description Electronic	9.99	0	1	2019-11-13 19:15:30	HULL	HULL	1	1
	5	Network Switch Title	Network Switch Description Electronic	9.99	0	1	HULL	HULL	HULL	1	1
	6	Black Chair Title	Black Chair Description Furniture	9.99	0	1	HULL	HULL	HULL	0	2
	7	Black Couch Title	Black Couch Description Furniture	9.99	0	1	2019-11-13 19:16:30	HULL	HULL	1	2
	8	Night Stand Title	Night Stand Description Furniture	9.99	0	1	HULL	HULL	HULL	1	2
	9	Orange Couch Title	Orange Couch Description Furniture	9.99	0	1	2019-11-13 19:11:30	HULL	HULL	1	2

Categories table

Result Grid				Filter Rows:
	c_id	c_name	c_status	
▶	1	Electronic	1	
	2	Furniture	1	
	3	Other	1	
	HULL	HULL	HULL	

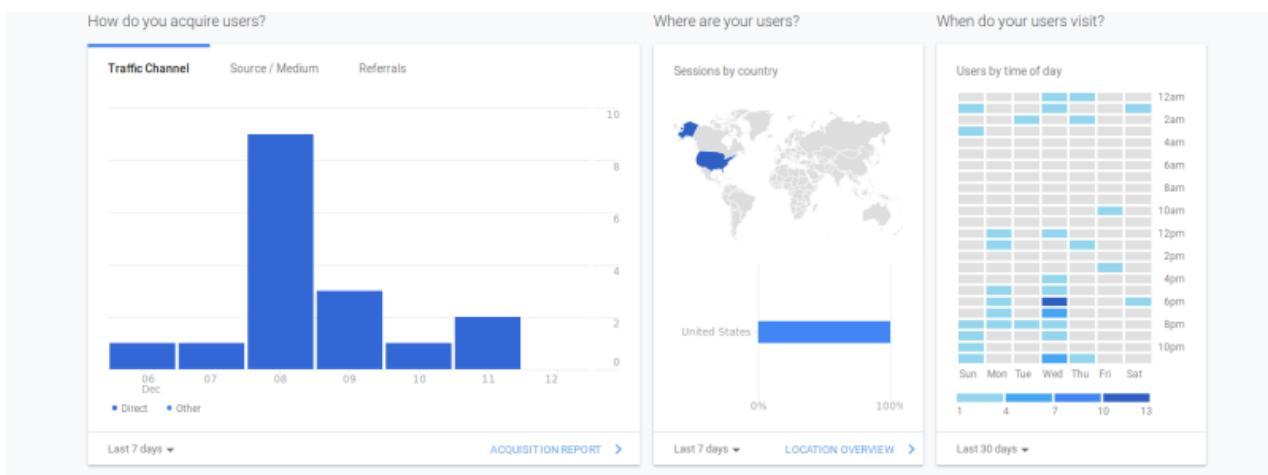
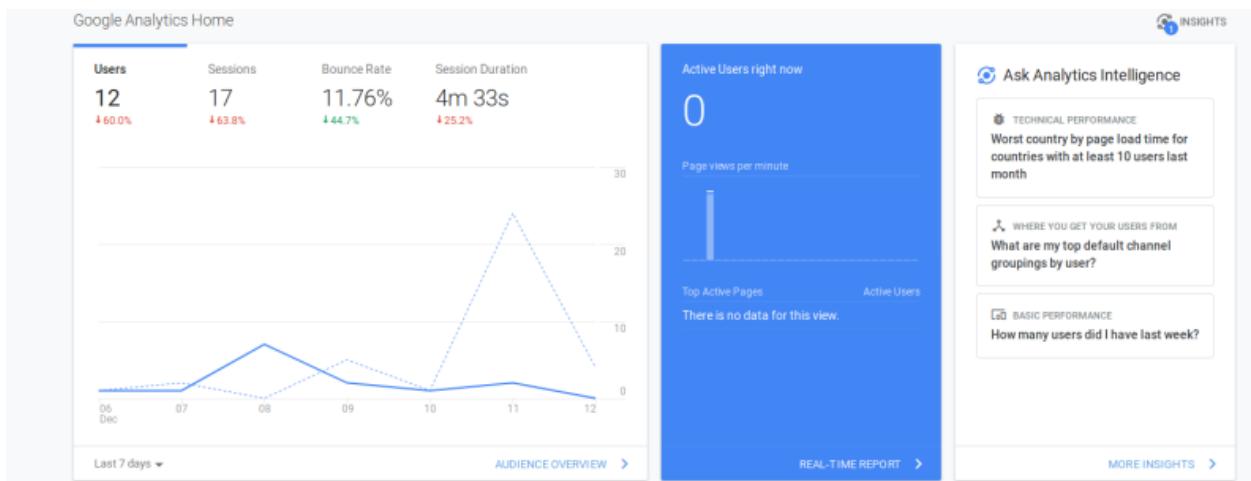
Message table

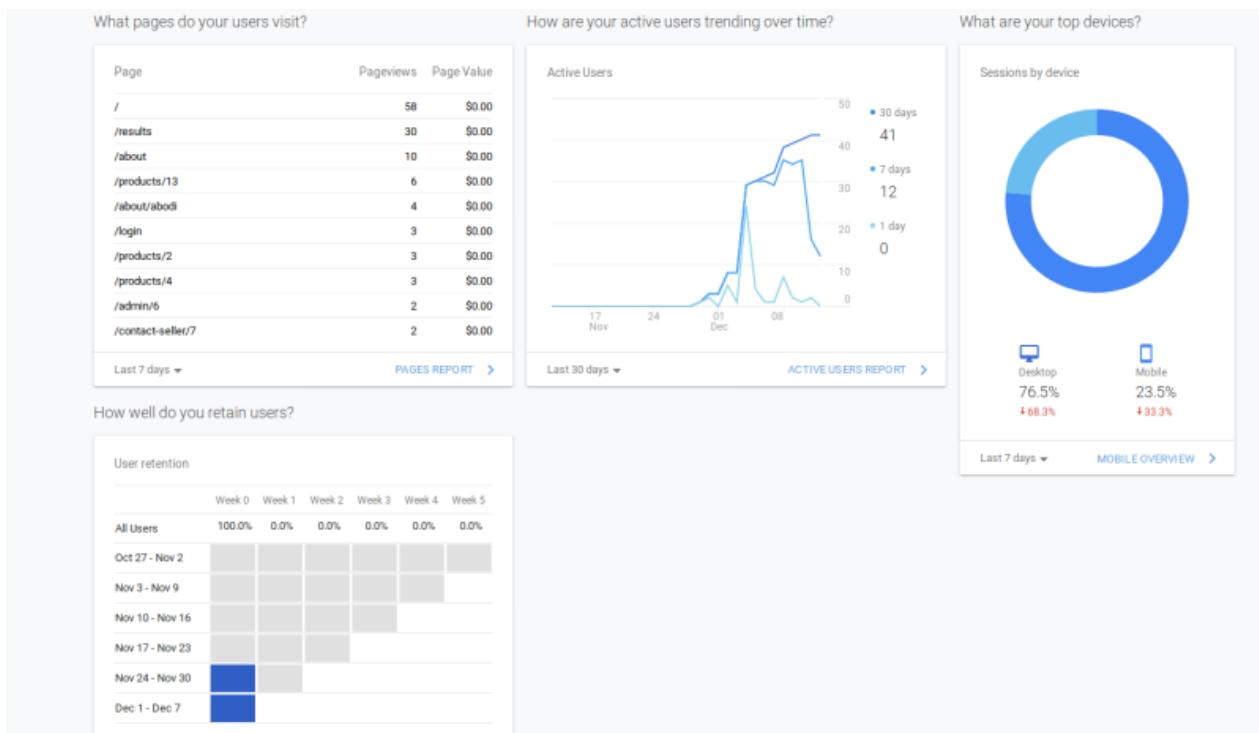
	m_id	m_text	m_sender_id	m_receiver_id	m_item_id	m_sent_ts
▶	1	Hello, I am interested in the Black Couch. Please...	2	1	7	2019-11-14 12:16:30
	2	This is a message in regards to Computer Moni...	5	1	2	2019-11-17 21:11:09
	3	This is a message in regards to Computer Mou...	5	1	3	2019-11-17 22:08:34
	4	This is a message in regards to Computer Mou...	5	1	3	2019-11-18 00:08:31

item_image table

Result Grid			Edit:		Export/Import:
ii_id	ii_image	ii_url	ii_j_id	ii_status	
1	NULL	/static/images/Electronic/ComputerKeyboard_m...	1	1	
2	NULL	/static/images/Electronic/ComputerMonitor_med...	2	1	
3	NULL	/static/images/Electronic/ComputerMouse_medi...	3	1	
4	NULL	/static/images/Electronic/Laptop_medium.jpg	4	1	
5	NULL	/static/images/Electronic/NetworkSwitch_medi...	5	1	
6	NULL	/static/images/Furniture/BlackChair_medium.jpg	6	1	
7	NULL	/static/images/Furniture/BlackCouch_medium.jpg	7	1	
8	NULL	/static/images/Furniture/NightStand_medium.jpg	8	1	

Google analytics plot for your WWW site





Screenshot(s) of the task management system

Team04

CSC-648-SFSU/csc648-fall2019-Team04 Free

Team Visible

DY AB A PY T +2 Invite

Butler Show Menu

To-Do-List

- Team meeting time: Wednesday 3:00pm to 6:00pm
- Task 1 - Github branches
- Task 3 - workflow, db schema, mockups
- Task 4 - M2 document review
- FE task - Alignment issues
- M3-Add dashboard page
- M3-Message
- M3-Admin page

+ Add another card

In Progress

- M2 tasks:
- M1 documentation and discussion
- M2 documentation
- Questions for professor:

+ Add another card

Review

- [Back-End] Configure Flask with routes for static files for root user.
- Front end bugs & changes

+ Add another card

Links

- Document: <https://docs.google.com/document/d/1rp6lnyW4BgLCiNGDXCfrqBdZd5JxV8X5GQsgGA9OXro/edit>
- Github: <https://github.com/CSC-648-SFSU/csc648-fall2019-Team04>
- Milestone 1 Documentation https://docs.google.com/document/d/1uRapkM7n4l8j6pxo67cwV9pTjFTLV4wWPnAr3_hzI/edit
- M2 <https://docs.google.com/document/d/1gEGbfPT2Rlzv1zrAOQ70TAXdbOE33cf02e3aE6BMTYA/edit>

+ Add another card

Feedback

- M0
- M1
- M2
- M3

+ Add another card

Team member contributions

CSC648/848 Section01 Team04 Team Member Contributions

Inbox Personal

Aditya Bodhi
To: Akshay Pradeep Kasar, Alan Ng and 4 more

Thursday, Dec 12, 11:51 PM

Hello All,

Name: Aditya Bodhi

Role: Team Lead

Contributions:
I will be describing the contributions I made to our team. Firstly, I have managed the team by dividing the tasks appropriately and making sure the milestone submissions are done on time. I have divided tasks based on the front end, back end and documentation. I held meetings every week for team interaction and discussion. I held a few important meetings with the backend lead, frontend lead, GitHub master for any milestone submissions and/or to merge the front end and back end branches. I frequently had some checkpoints to check the progress of our project.
With respect to the technical work, I made more contributions to the backend part compared to the front end. I have worked on the item posting page, search functionality, SQL queries for these functionalities. Before any milestone document submission or document freezing, I made sure that we have covered all the required/mandatory sections of the document. My individual contribution to the team is my 'about page'. Lastly, I had a great experience working with all of my team members to develop a final software product.

Submissions to GitHub: 28 commits

Regards,
Aditya Bodhi

Reply Forward Quick Reply

Akshay Pradeep Kasar

Friday, Dec 13, 8:46 PM  ...

> To: Aditya Bodi, Alan Ng and 4 more

Hello All,

Name: Akshay Kasar

Role: Git-Master

Contributions:

- Besides being git-master I also contributed on the backend development.
- Managed Git hub PRs, and merging of branches.
- Designing of database.
- Developed initial master script for database schema.
- Implementation of the backend part of login, register and hashing passwords.
- Designing of the messaging feature.

Total Commits: 52

Thank you.

Warm Regards,

Akshay Kasar

 [Show History](#)

 Reply |  Forward |  Quick Reply

CSC648/848 Section01 Team04 Team Member Contributions

Inbox  Personal

SFSU

> To: Philip Yu, Aditya Bodi and 1 more Cc: Akshay, Alexander and 1 more

Saturday, Dec 14, 9:06 PM  ...

Hello All,

Name: Deming Yan

Role: Full-Stack Engineer

Contributions:

- Write and manage documentation
- Manage Trello tasks
- Fix front end about page's footer
- Change buttons of personal page
- Participated mockup design
- Participated in database scheme design
- Use cases
- Persinas
- Advised team members on page design

Total Commits: 6

Thank you!

Sincerely,

Deming Yan

Tejasvi Arun Belsare

Saturday, Dec 14, 11:20 PM ⏪ ...

> To: SFSU, Philip Yu and 2 more Cc: Akshay, Alexander

Hello All,

Name: Tejasvi Arun Belsare

Role: Document master, Backend Engineer

Contributions:

Being a document master, I managed all the documentation part for this project. I made initial draft with all the content for all milestone documents (except M3). I modified all documents as per Professor's feedback, reviewed and froze (except M3).

With respect to project design, I designed personae and use cases. Also designed main data items and entities for the project. I designed initial draft of all the functional requirements for all users of the project. I designed competitive analysis features. I made initial database schema including all designed entities. I identified actual key risks for the project and designed usability test plan for the project.

From back-end's point of view, I implemented item posting page functionality which included accepting image files from user and storing them on the server. I worked on validation of input fields. I contributed in implementation of lazy registration for item posting page. Also, worked on my about page.

Total commits: 13

Best regards,

Tejasvi

CSC648/848 Section01 Team04 Team Member Contributions

Delegate | [Inbox](#) | [Personal](#) | ...

Sunday, Dec 15, 12:42 AM [Reply](#) [Forward](#) [Quick Reply](#)

Alexander R Kohanim
To: Tejasvi Arun Belsare, SFSU and 3 more Cc: Akshay

Hello Everyone,
Name: Alex Kohanim

Role: Backend Lead/ Fullstack/ Temp Team Lead/Admin/Server Master/Devops Enginer

Contributions:

- added a template file for test routes
- fixed dummy links and updated routes for static files also fixed minor css t
- updated paths for static files and fixed links
- Removed some css files that are not in use
- Front page is populated with many items, and handled some 404 logic
- set up categories for data base
- got search bar working with sql queries
- got categories to work statically, next step is to mix categories and search
- fixed some query edge cases
- made product page more dynamic
- made more restrictions based on itemStatus for Admin uses
- moved the queries to a class, cleaned up a bit and made the admin dashboard use the frontend html
- got Akshay's registration and login to work, also added logout.
- added some flash messages for login and home
- got lazy registration working for contact seller
- added search result to search bar
- added documentation for making a new branch
- wrote some sql scripts for setup
- updated sql script to use smaller images
- fixed some bugs with banning users
- added more 404 logic based on unregistered user
- added sanitization for login and registration
- added richard to the db
- updated home page with most recent items.
- user dash board and seller messages are both populated from the DB.
- Got contact seller page working, might need more info, also updated sqllite file to drop the db on init
- fixed login bug
- fixed password
- updated most recent query and added ts for new items
- added header comments
- removed idea folder and added it to gitignore
- got search updated combining filters, categories and sorting
- fixed some bugs with better search including edge cases and adding time stamps to dummy data. Also made the feedback a bit more dynamic
- converted app.py single file into more portable blueprints
- changed 'categories' in dropdown to 'all' to let user know more about their search
- made all images use static linking for better cache utilization
- added google analytics
- got marked as sold working on user-dashboard
- fixed links in login and registration, they don't jump to home anymore
- added a flash message for successfully posted items
- added error messages and implemented password checking with confirm-password
- Added Cron jobs to catch 500 and misc errors on the server to reduce down time.
- Setup and managed AWS server including the domain name and SSL certificate

Submissions to GitHub: 100+ Commits

Regards,
Alex Kohanim

[Show History](#)

CSC648/848 Section01 Team04 Team Member Contributions

(Inbox) Personal

Alan Ng

> To: Philip Yu, Aditya Bodl Cc: Akshay, Alexander and 2 more

Saturday, Dec 14, 2:25 AM ⏪ ...

Name: Alan Ng

Role: Frontend lead

Contributions:

- rough template for about page
- corrected path to stylesheet
- changed dummy links to proper
- incorporated profile css for a universal css file
- fixed links to single css/nav bar
- corrected css path
- nav bar padding fix
- made text more clear in body
- text no longer stretch too far
- incorporated old css to new css
- added a description to my profile page
- removed some unnecessary css
- updated regis and navbars
- prototype modal log in
- prototype registration page
- corroborated with frontend team on posting pages
- updated styles for various pages
- added password validation and json conversion
- changes to modal login
- added a test page with search bar and filter
- scaled images
- helped with m1 and m3 documents
- commented out local files and un-comment app.run
- Merge pull request #27 from CSC-648-SFSU/frontend-navbar
- Merge pull request #30 from CSC-648-SFSU/fe-admin-db
- updated to use new header/footer
- base jinja template for header/footer
- added styles for about pages
- added even smaller images for faster loading
- implemented captcha and visual for pw verification
- modified input placements in registration and log in
- fixed footer bug on about pages
- fixed search bar sizing
- implemented javascript function for back, home and close button functionalities
- utilized script for back buttons for log in, register, and item posting pages
- incorporated close button for contact seller page and back button to product page
- implemented back button for seller inbox page
- updated about page navbar

Total Commits: 34

Thank you!

Best Regards,

Alan Ng

Show History

Reply Forward Quick Reply

CSC648/848 Section01 Team04 Team Member Contributions

Inbox Personal

Philip Yu

> To: Aditya Bodl Cc: Akshay, Alan and 3 more

Friday, Dec 13, 11:58 PM



...

Hello All,

Name: Philip Yu

Role: Frontend Engineer

Contributions:

- Add template member profile pages
- Update UI/UX of home page to match mockups
- Update product details page to match mockups
- Fix CSS collision between home and product details page
- Change placeholders with backend data
- Fix search-container misalignment
- Merge branch 'frontend' into philip-test
- Add login page based on mockups
- Add register page based on mockups
- Merge branch 'frontend' into login-reg-page
- Add item posting page based on mockups
- Merge branch 'frontend' into item-posting-page
- Merge branch 'frontend' into item-posting-page
- Update .gitignore
- Remove .DS_Store
- Reorganize file structure with updated html
- Update nav-bar based on m2 feedback
- Add Admin Dashboard based on mockups
- Update Admin Dashboard with tabs and functionality
- Merge pull request #32 from CSC-648-SFSU/fe-admin-update
- Merge pull request #34 from CSC-648-SFSU/frontend-m3
- Update nav bar for logged in users
- Add User Dashboard
- Add Seller Inbox Page
- Add contact seller page
- Merge pull request #36 from CSC-648-SFSU/fe-user-db
- Merge pull request #38 from CSC-648-SFSU/frontend-m3
- Add sorting dropdown in home page
- Merge branch 'full-stack-m3' into frontend-m3
- Merge pull request #44 from CSC-648-SFSU/frontend-m3
- Update .gitignore
- Re-add sorting dropdown button in home page
- Add wishlist field prompts in item posting page
- Redesign item posting buttons and select dropdown
- Connect search bar with search button and category select
- Change sort-by dropdown to select tag and move edu purpose label
- Add CSS for category select
- Remove wishlist dropdown from product details page
- Update image to become clickable
- Fix cancel button in item posting page
- Merge pull request #61 from CSC-648-SFSU/fe-m3-feedback
- Fix responsiveness
- Add number of items in home page and fix footer in about page
- Add titles to all pages and remove show more button
- Add hint to tradable label and Alex add actual username
- Add red asterisk and required fields for registration, item posting, and contact seller page
- Add design for back buttons
- Add item details in contact seller page and Alex add logic for sending products to frontend

Total Commits: 48 (frontend-m4 as of 12/13/2019 @ 11:45pm)

Thank you!

Best Regards,
Philip Yu

Reply Forward Quick Reply

Post analysis – lessons learned

CSC 648/848 01 Team 04 Post Analysis Inbox 8 Personal

Aditya Bodhi Sunday, Dec 15, 5:58 PM

> To: Akshay Pradeep Kasar, Alexander R Kohanim and 4 more ...

Hello Team,

As we are coming to the end of the project, I would like to summarize the journey of our project.

There were a few mistakes, issues and challenges we faced while developing the product. The initial mistake we did was we didn't maintain a proper hierarchy for the GitHub branches. We have later done a good thing by documenting our GitHub flow and usage. We had some issues with CSS conflicts. Sometimes we have underestimated some things and features. For example, to make the search functionality persistent, we assumed that it wouldn't take more than a day. But, to make it work efficiently, we had to make a lot of other changes like fetching the categories from the database which increased our work a lot. I would make an overestimation next time by assuming we at least need twice the time of our assumption. We had issues with different OSes. Our team was working on Ubuntu, Windows, and Mac. To save the files in the local directory, we found that our code was working in one OS but not others. We had to spend more time than expected to fix this issue. I would categorize the functionalities and possible considerations. Based on functionality, I'll list down the additional considerations our team should make. This will reduce any possible bugs. We then had issues with the DB credentials. We have hardcoded the credentials and it is not a convenient and correct way to implement. We should fetch it from a file. Coming to our team challenges part, the first challenge was the team meeting. It was a bit hard for everyone to be on time due to their different schedules. But, we could manage by communicating through calls, discord, etc. The other challenging part was the documentation. Documentation is an important thing for our product. It describes about our implementation and sometimes we have to think way ahead to have some requirements. One person will not be sufficient for the documentation. We need at least two or three people to work on it. The team felt the messaging feature to be challenging as it was not clear about what was required. In such cases, I would suggest the team to discuss with the professor before starting on the implementation part. Lastly, the blueprints merge was very challenging as it has changed the entire code structure. Making sure that there are no merge conflicts was very essential here. In such cases, I would split one merge request into multiple so that it becomes easy to review and merge.

I really had a great experience and fun with my team. I would like to thank all my team members for successfully completing all the P1 functionalities to deliver a final software product. I hope we get another opportunity to work as a team.

Regards,
Aditya Bodhi.

[Reply](#) | [Forward](#) | [Quick Reply](#)