

Final Project for SW Engineering CSC 648/848 Fall 2020

Team 07

SFSU Trade Mart

Team Members:

Team Lead: Alicia Ramirez
(Email: aramirez23@mail.sfsu.edu)

Front-End Lead: Jonathan Pak

Back-End Lead: Valeria Vallejo

Github Master: Ricardo Carretero

Member: Chandler Cruz

Member: Chris Manaoat

URL: <http://54.176.241.134/>

Date: 12/18/2020

2. Product Summary:

Name of the product: SFSU TradeMart

All Major Committed Functions:

All Users (Unregistered/Registered):

- 1: All users shall be able to register accounts with the website.
- 2: All users who have not logged in shall be able to access the login/register page, and log in through any page of the website.
- 3: All users who are not logged in shall be suggested the option to log in or register upon making an offer or submitting a post for approval.
- 4: All users shall be able to view listings on the website.
- 5: All users shall be able to search for specific listings based on words in its title, description, or category.
- 6: All users shall be able to see descriptions of the items for sale on the search results and browsing pages of the website.
- 7: Users who are logged in shall be able to log out.
- 8: Registered users shall be able to create listings for their items and put them on sale through the website, thus known as posts. Postings contain the name of the seller, condition, and price.
- 9: Registered Users interested in specific listings shall be able to contact the seller to decide themselves the details of the transaction.
- 10: Registered Users shall be able to choose the time and meeting location from a set of predefined locations when contacting a seller.
- 11: Registered Users shall be able to access a dashboard either upon signing in or clicking the “Welcome [username]” on the navbar, where they can access three tabs, My Postings, Messages Sent, and Messages Received.
- 12: Admins shall be required to approve or disapprove listings according to the TOS using Workbench.
- 13: Admins shall be required to remove users for violating the TOS using Workbench.

URL(This URL is different than the one in other milestones): <http://54.176.241.134/>

3. Milestone Documents:

Milestone 1:

SW Engineering CSC 648/848 Fall 2020
SFSU Trade Mart
Team 7

Team Members:

Team Lead: Alicia Ramirez
(Email: aramirez23@mail.sfsu.edu)

Front-End Lead: Jonathan Pak

Back-End Lead: Valeria Vallejo

Github Master: Ricardo Carretero

Member: Chandler Cruz

Member: Chris Manaoat

Milestone 1
September 28, 2020

History Table

Date Submitted: September 22, 2020
Date Revised: September 28, 2020

1. Executive Summary:

Students sometimes have a need to partake in exchanges to get what they need and get rid of what they don't need. Whether that be getting rid of old electronics, furniture, books, etc. or maybe they are in need of services such as tutoring or essay editing. They may know they want these services, but to go through the work to carry out that exchange can quickly become difficult. Our website will relieve students of the anxiety that comes with trying to find or sell the items and services they desire and ensures they will get the best deal whenever possible.

On our site, Dragutin, faculty and alike can sell and buy items and services to other people at SFSU safely and efficiently. Sellers can make their item/service easy to find by tagging the category the item/service falls under. They can choose to list things as electronics, furniture, books, or services like tutoring. This allows buyers to narrow down what they're looking for by selecting the category they want. For books specifically, the user will be able to search for books by class, allowing students to easily find the materials they'll need before classes even start. We know that item quality is a major factor for a lot of students and our site requires that all sellers provide detailed descriptions of the product condition.

The site has lazy registration as buyers are only forced to register/log in once they want to make an offer to a seller, and sellers when they want to send the item/service post off to be approved. Sellers may choose from any bids and offers that come their way. They also have the option to set a suggested price to help give buyers an idea of what they expect to get from the transaction or even a fixed price so the buyer knows they won't accept anything else. While the site helps buyers and sellers agree on a price, the site does not actually handle any payment processing, and is instead handled by the buyers and sellers themselves. Our sellers can be sure that they'll receive what they believe their product is worth.

Our unique system also gives buyers and sellers reviews in order to give people peace of mind that they are buying from or selling to a trustworthy person. When deciding on where to meet, the site offers predefined places in the school so users are comfortable knowing they will carry out their exchange in a public place.

Our student team is ready to offer a fast, easy to use website that SFSU staff, students, and faculty can accept as their go to trade website.

2. Personae and Main Use Cases:

(All images are under the free, attribution-free, commercial use approved Pixabay and Pexels licenses. There is no endorsement of our product by the actual people in the images used.

Pixabay License: <https://pixabay.com/service/license/>

Pexel License: <https://www.pexels.com/license/>)

Kahleed



Demographics

Age: 23
Occupation: College Student

Goals

- Fast experience
- Minimal Interaction

Pain Points

- Forced customization
- Requires too much person-to-person interaction

Traits

- Time Efficient 
- Precise 
- Introvert 
- Safety Focused 

Donald



Demographics

Age: 30
Occupation: College Student

Goals

- Form connection with other user
- Know quality of product and reliability of other user

Pain Points

- Slow Connection
- Site Aesthetic Is Ugly

Traits

- Tech Savvy 
- Business Oriented 
- Extrovert 
- Artistic 

Prof. Pugman



Demographics

Age: 51

Occupation: College Professor

Goals

- Simple navigation
- Fast order update notifications

Pain Points

- Too complicated
- Not able to provide details she wants to

Traits

- Busy

- Basic Tech Skills

- Detailed

- Prompt

Lisa



Demographics

Age: 42

Occupation: Administrator

Goals

- Options to enforce rules effectively
- Various options for conflict resolution

Pain Points

- Disapproving without giving reason
- Weak admin options

Traits

- Thorough

- Diplomatic

- Empathetic

- Rule Oriented

Use Case #1: A **student** needs to get his book before the midterm. He thought he wouldn't need it since he has google, but it was not enough. The bookstore would take a few days to stock, so he is looking for a used copy. After talking with classmates he learns about **SFSU Trade Mart**. He goes to the site and browses through to see if anyone is selling what he needs. He **browses by class** and finds what he is looking for. Knowing about the multiple versions available made him hesitant, but he found his book after looking at **images** on the **items for sale** section. Once he finds what he needs, he is asked to make an account through **lazy registration** to **make an offer to the seller**, which he does without much hassle on his part. He also wants as little communication as possible so he was relieved to see that the only communication was to **choose from one the predefined locations at SFSU** and time to meet.

Use Case #2: A **student** has a pretty large group of friends in their major. While the student was able to order all of their books and class materials ahead of time, some of their friends taking similar classes weren't able to grab the books or Clickers before the items sold out at the bookstore. Since they're always eager to help their friends out, the student looks at the **items on sale** and **new arrivals** in hopes to help find the materials for everyone. The student then utilizes the search function, and due to the clean and easy to read **object listings**, they were able to find what they were looking for. The student takes note of the **post id number** listed and notifies their friends about this site. After searching for the material themselves and browsing through the **diverse selection of items**, the friends decided to **register with their SFSU emails** and were then able to contact the sellers and make deals.

Use Case #3: A **professor** just finished teaching her class. She has some additional resources for her class, so she **registers on the homepage and logs** into her SFSU TradeMart account and lists them as optional materials in the **class resource form**, so her future students can see them when they **search by class**. On hand, she also has extra copies left over from previous years so she **posts some listings** on the site. The site is easy to navigate and she completes this process in a couple of minutes with the details she wants to include. Afterwards, the site provides **notifications** that inform her of **pending offers**. The simple **accept/accept but with location or time change/counter with different price/denial options** allows her to easily manage and finalize all of her transactions. For the next semester, she is able to **update her class resource form** to account for the new version of the textbook she is using.

Use Case #4: An office worker has just finished sending out a batch of emails to her colleagues and has a free moment to **approve listings**. Since she has such scattered moments of reprieve throughout her workday, she can't devote long stretches of time to navigating through the

website as an **admin** to approve all incoming student and teacher listings. However, since the UI was easy and quick to navigate, she was able to **approve** a healthy amount of posts and **disapprove posts** with reasons while reading over the necessary details of each one during her small break.

Use Case #5: A student is moving out of her dorm and has some furniture that she doesn't want to take. She remembers that SFSU Trademart has a section for **furniture listings**. She makes sure to take plenty of **pictures to show the conditions** and is easily able to make detailed posts for everything she needs to get rid of. Before sending her first post off to be approved, she is **asked to log in or register an account**, which she fills out quickly and gets on with submitting the rest of her posts. Since she is on a time crunch, she **allows for bids and offers from buyers** rather than forcing buyers to offer only a fixed price.

3. List of main data items and entities – data glossary/description:

- Image database: Houses the collections of images used to visually describe items.
- Object listing: The data structure of the item for sale.
 - Name of person selling
 - Rating
 - Items listed
 - Items sold
 - Price
 - Description
 - Image
 - Category
- Database table:
 - ID
 - Date
 - Text
 - Sender
 - Location
- Users: Anybody browsing or using the website.
 - Unregistered: Users using the site who haven't registered with an account yet.
 - Registered: Users using the site who have previously created an account.
 - Admin: School faculty who have registered with the site and browse it to moderate posts according to TOS.
- Contacts: Data structure containing the users information, items for sale, and registered users transactions.
- List of past orders: Sub class that is stored in one of the contact data structures containing all orders by the subject
 - Array of items purchased or sold
- Comment history: Data structure containing all the comments an offer chain(the optional messages that can be attached on the offers between one buyer and the seller) has received.
- Notification history: Data structure containing all the notifications the user has received.
- Registration form: Using this form, unregistered users can create accounts with the website.
 - Name
 - Email with sfsu.edu suffix
 - Password
- Login form: Using this form, registered users can login to the website to interact with it further such as with purchasing items.
 - Email
 - Password
- Class resource form: Using this form, staff can easily input all required and optional resources that students will need for their class.

4. Initial List of Functional Requirements:

1. Unregistered users shall be able to register accounts with the website.
2. All users who have not logged in shall be able to access the login/register page, and log in through any page of the website. Unregistered users will be offered the chance to register.
3. Registered users have all the access that unregistered users do and more.
4. Users who are not logged in shall be suggested the option to log in or register upon making an offer or submitting a post for approval.
5. Users who are logged in shall be able to log out.
6. Registered users shall be able to create listings for their items and put them on sale through the website, thus known as posts. Postings contain the name of the seller, condition, price and whether it is negotiable.
7. Unregistered Users shall be able to view listings on the website.
8. Unregistered Users shall have the ability to search for specific listings based on words in its title or description.
9. Unregistered Users shall be able to filter their search results based on the prices of the listings.
10. Unregistered Users shall be able to filter their search results based on the available book versions of any textbook listings.
11. Unregistered Users shall be able to filter their search results based on which materials are required for specific courses.
12. Unregistered Users shall be able to filter their search results based on which class materials are required for classes taught by specific professors.
13. Unregistered Users shall be able to filter their search results based on which class materials are required for classes taken for specific majors.
14. Registered Users looking to purchase items on the website shall be able to comment on their questions or concerns on the listing.
15. Descriptions of the items for sale shall be visible to a(n) unregistered/registered user on the search results and browsing pages of the website.
16. Admins shall be required to approve or disapprove listings according to the TOS.
17. Admins shall be required to remove users for violating the TOS.
18. Registered Users shall receive notifications when their listing is accepted/rejected, someone purchases their listing, they receive a message, or they receive a rating.
19. Registered Users purchasing products on the website shall be able to choose the meeting location and time on the website.
20. Registered Users interested in specific listings shall be able to make bids and offers to the sellers of those listings.
21. Registered Users who receive bids and offers from users about their listings shall be able connect with the sellers and negotiate.
22. Registered Users shall be able to rate buyers and sellers they've had transactions with on a 5 point rating scale.

5. List of Non-Functional Requirements:

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

6. Competitive analysis:

	Craigslist	Amazon	Ebay	Trade mart
Search by Item/Category	y	y	y	y
Search by class	n	n	n	y
Barter	y	n	y	y
Version compare	n	n	n	y
Rate/review seller	y	y	y	y

One feature our product has that is advantageous compared to other sites is that they do not have the ability to produce listings of goods for sale that are relevant to that semester only. This can be managed through the class resource form that sellers will complete to organize the products in categories. The ability to search by class has to be one of the most useful features compared to our competitors, this allows any student to purchase their books for the upcoming semester easily.

7. High-level system architecture and technologies used:

- Server Host: AWS, 1 3.3 Ghz Intel Scalable Processor, 1 GB Ram
- Operating System: Linux 20.04
- Database: mySQL 8.0.21
- Web Server: Nginx 1.14.0
- Server-Side Language: Python 3.8.2
- Additional Technologies:
 - Web Framework: Bootstrap, Flask
 - IDE: VS Code, Vim

8. Team and roles:

<i>Team Lead/Document Master:</i>	Alicia Ramirez
<i>Front-End Lead:</i>	Jonathan Pak
<i>Back-End Lead:</i>	Valeria Vallejo
<i>Github Master:</i>	Ricardo Carretero
<i>Member:</i>	Chandler Cruz
<i>Member:</i>	Chris Manaoat

9. Checklist:

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

Milestone 2:

SW Engineering CSC 648/848 Fall 2020
SFSU Trade Mart
Team 7

Team Members:

Team Lead: Alicia Ramirez
(Email: aramirez23@mail.sfsu.edu)
Front-End Lead: Jonathan Pak
Back-End Lead: Valeria Vallejo
Github Master: Ricardo Carretero
Member: Chandler Cruz
Member: Chris Manaoat

Milestone 2
October 15, 2020

History Table

Date Submitted: October 15, 2020
Date Revised: October 22, 2020

1. Executive Summary:

Students, staff, and faculty sometimes have a need to participate in exchanges to get what they need and get rid of what they don't. Whether that includes getting rid of old electronics, furniture, books, etc; maybe they are in need of services such as tutoring or essay editing. The student may find itself in need of such services or products and no good way to access those things. Our website, the SFSU Trade Mart, will relieve them of the anxiety that comes with trying to find or sell the items and services they desire and maximizes the possibility that they will get the best deal.

On our site, faculty and alike can sell and buy items and services to other people at SFSU safely, and efficiently. Sellers can make their item/service easy to find by tagging the category the item/service falls under. They can choose to list things as electronics, furniture, books, or services like tutoring. This allows buyers to narrow down what they're looking for by selecting the category they want. For books specifically, the user will be able to search for books by class, allowing students to easily find the materials they'll need before classes even start. We know that item quality is a major factor for a lot of students and our site requires that all sellers provide detailed descriptions of the product condition.

The site has lazy registration as buyers are only forced to register/log in once they want to make an offer to a seller, and sellers when they want to send the item/service post off to be approved. Sellers may choose from any bids and offers that come their way. Our sellers have the option to set a suggested price to help give buyers an idea of what range they can expect to pay for the transaction or even a fixed price for those unwilling to haggle. While the site helps buyers and sellers agree on a price, the site does not actually handle any payment processing, and is instead handled by the buyers and sellers themselves. Our sellers can be sure that they'll receive what they believe their product is worth.

When deciding on where to meet, the site offers predefined places in the school so users are comfortable knowing they will carry out their exchange in a public place.

Our student team is ready to offer a fast, easy to use website that SFSU staff, students, and faculty can accept as their go to trade website.

2. List of main data items and entities – data glossary/description:

- Image database: Houses the collections of images ,stored as BLOBs, used to visually describe items.
- Object listing: The data structure of the item for sale.
 - Username of person selling
 - Rating
 - Items listed
 - Items sold
 - Suggested price
 - Description
 - Image
 - Category
 - Approval status
 - Offer type (fixed or bid)
 - Listing date/time
 - Image of object
 - Condition of object
- Categories: A way to filter data when searching for items by the category items correspond to.
 - Books
 - Electronics
 - Furniture
 - Tutoring
 - Editing
- Users: Anybody browsing or using the website.
 - Unregistered: Users using the site who haven't registered with an account yet.
 - Registered: Users using the site who have previously created an account.
 - Admin: People hired to moderate the listings. They are in charge of approving and disapproving posts.
 - Listing submissions waiting with no approval status
 - Class resource updates
 - List of reports/complaints
- User profile: Data structure containing the user's information, items for sale, and registered users' transactions.
 - Full name
 - Email
 - Address
 - Current listings
 - Past offers
- List of past offers: Sub class that is stored in one of the user profile data structures containing all offers by and to the user.
 - Array of items purchased or sold

- Comment history: Data structure containing all the comments an offer chain(the optional messages that can be attached on the offers between one buyer and the seller) has received.
- Notification history: Data structure containing all the notifications the user has received.
- Registration form: Using this form, unregistered users can create accounts with the website.
 - Full name
 - Email with sfsu.edu suffix
 - Password
- Login form: Using this form, registered users can login to the website to interact with it further such as with purchasing items.
 - Email
 - Password
- Class resource form: Using this form, staff can easily input all required and optional resources that students will need for their class.
 - Title
 - ISBN
 - Edition/version
 - Term
 - Department
 - Course
 - Section

3. Functional Requirements - Prioritized:

Priority 1:

All Users (Unregistered/Registered):

- 1: All users shall be able to register accounts with the website.
- 2: All users who have not logged in shall be able to access the login/register page, and log in through any page of the website
- 3: All users who are not logged in shall be suggested the option to log in or register upon making an offer or submitting a post for approval.
- 4: All users shall be able to view listings on the website.
- 5: All users shall be able to search for specific listings based on words in its title, description, or category.
- 6: All users shall be able to see descriptions of the items for sale on the search results and browsing pages of the website.
- 7: All users shall be able to filter their search results based on which class materials are required for classes taught by specific professors.
- 8: All users shall be able to filter their search results based on which class materials are required for classes taken for specific majors.

Registered Users Only:

- 9: Users who are logged in shall be able to log out.
- 10: Registered users shall be able to create listings for their items and put them on sale through the website, thus known as posts. Postings contain the name of the seller, condition, price and whether it is negotiable.
- 11: Registered Users interested in specific listings shall be able to contact the seller to decide themselves the details of the transaction.
- 12: Registered Users shall be able to choose the time and meeting location from a set of predefined locations when making an offer.
- 13: Registered Users shall be able to access a dashboard either upon signing in or clicking the “Welcome [username]” on the navbar, where they can access four tabs, My Postings, Messages Sent, Messages Received, and My Class Resources.
- 14: Registered Users looking to purchase items on the website shall be able to add a comment when making an offer.
- 15: Registered Users shall be able to fill out a Class Resource Form to associate a book with a class for the designated term and section.

Admins Only:

- 16: Admins shall be required to approve or disapprove listings according to the TOS.
- 17: Admins shall be required to remove users for violating the TOS.

Priority 2:

All Users (Unregistered/Registered):

- 18: All users shall be able to filter their search results based on the prices of the listings.
- 19: All users shall be able to filter their search results based on the available book versions of any textbook listings.
- 20: All users shall be able to filter their search results based on which materials are required for specific courses.

Priority 3:

Registered Users Only:

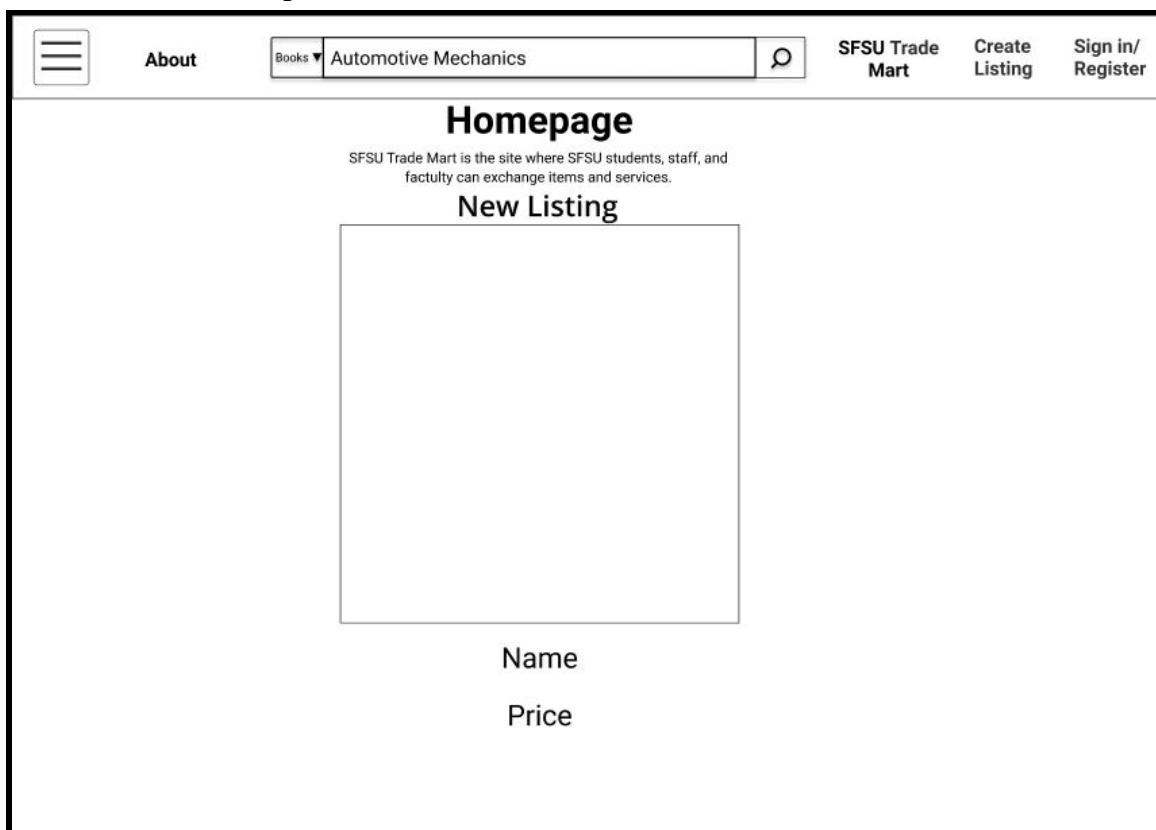
- 21: Registered Users shall receive notifications when their listing is accepted/rejected, someone purchases their listing, they receive a message, or they receive a rating.
- 22: Registered Users shall be able to rate buyers and sellers they've had transactions with on a 5 point rating scale.

Admins Only:

- 23: Admins shall be able to access a dashboard that allows them to approve/reject listings and class resource forms.

4. UI Mockups and Storyboards:

Use Case #1: A student needs to get his book before the midterm. He thought he wouldn't need it since he has google, but it was not enough. The bookstore would take a few days to stock, so he is looking for a used copy. After talking with classmates he learns about **SFSU Trade Mart**. He goes to the site and browses through to see if anyone is selling what he needs. He **browses by class** and finds what he is looking for. Knowing about the multiple versions available made him hesitant, but he found his book after looking at **images** on the **items for sale** section. Once he finds what he needs, he is asked to sign in or make an account through **lazy registration** to **make an offer to the seller**, which he does without much hassle on his part. He also wants as little communication as possible so he was relieved to see that the only communication was to **choose from one the predefined locations at SFSU** and time to meet.



 About All  SFSU Trade Mart Create Listing Sign In / Register

1-6 of 10 results for Books : "Automotive Mechanics"

Sort by  Best Match

Category

- Items
- Furniture
- Electronics
- Books**
- Services
- Tutoring
- Editing

Class

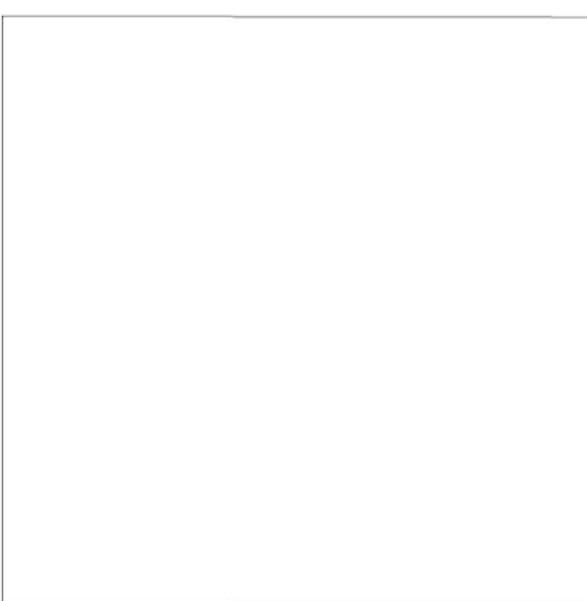
- Biology
- Chemistry
- Engineering
- Automotive**
- Music
- Physics
- Science

Filters

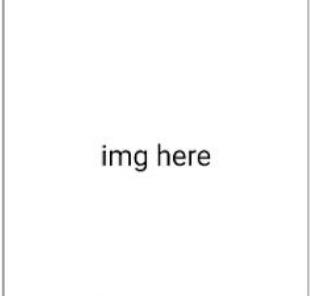
Automotive Mechanics Quality: New Suggested Price \$100.00 Contact	Automotive Mechanics Quality: Used Suggested Price \$50.00 Contact	Automotive Design Quality: Fair Suggested Price \$85.00 Contact
Automotive Design Quality: New Suggested Price \$125.00 Contact	Mechanic's Special Quality: Used Suggested Price \$45.00 Contact	Aerodynamics Quality: Fair Suggested Price \$45.00 Contact

( 1 of 2 pages )

 About All  SFSU Trade Mart Create Listing Sign In / Register

		Automotive Mechanics From: User2342 (4.5/5 14 reviews) Quality: Used Suggested Price: \$50.00 Description: Automotive Mechanics textbook. Some signs of wear. Contact Return
		
		
		
		

Listing ID: 123456789

 img here

Contact

Listing ID:

Description
Automotive mechanics textbook.
Some signs of wear

Listing Title

Condition

Seller's preferred location

Your Preferred location

Pick-up Time

Suggested Price

Make an Offer

Message
(Be sure to include contact information)

Cancel **Contact**

 You must have an account to make an offer

Contact

Listing Title

Condition

Seller's preferred location

Your Preferred location

Pick-up Time

Suggested Price

Make an Offer

Message
(Be sure to include contact information)

LOG IN

Username

Password

Create an account

[Or Forgot Password](#)

Cancel **Contact**

 About All ▾  SFSU Trade Mart Create Listing Welcome, Donald

My Postings	Messages Sent			
Messages Sent	1 of 1 item(s)			
Messages Received	Date Sent	Receiver id	Item Title	Details
My Class Resources	10/10/20	12345	Automotive Mechanics Textbook	View Details

Use Case #2: A **student** has a pretty large group of friends in their major. While the student was able to order all of their books and class materials ahead of time, some of their friends taking similar classes weren't able to grab the books or Clickers before the items sold out at the bookstore. Since they're always eager to help their friends out, the student looks at the **items on sale** and **new arrivals** in hopes to help find the materials for everyone. The student then utilizes the search function, and due to the clean and easy to read **object listings**, they were able to find what they were looking for. The student takes note of the **post id number** listed and notifies their friends about this site. After searching for the material themselves and browsing through the **diverse selection of items**, the friends decided to **register with their SFSU emails** and were then able to contact the sellers and make deals.

The screenshot shows a web-based marketplace interface. At the top, there's a navigation bar with links for 'About', 'All' (with a dropdown arrow), a search icon, 'SFSU Trade Mart', 'Create Listing', and 'Sign In / Register'. Below the navigation, a message says '1-6 of 10 results for Electronics : "Clicker"' and there are filter options for 'Items on sale' (selected) and 'New Arrivals'. A 'Sort by' dropdown is set to 'Best Match'. On the left, a sidebar has sections for 'Category' (with 'Electronics' selected), 'Class' (closed), and 'Filters' (with 'Items on sale' checked). The main content area displays six item cards in a grid. Each card shows a thumbnail, the item name ('Clicker'), its quality ('Quality: New', 'Used', or 'Fair'), the suggested price, and a 'Contact' button. The items are:

Thumbnail	Name	Quality	Suggested Price	Action
[Blank]	Clicker	Quality: New	\$45.00	Contact
[Blank]	Clicker	Quality: Used	\$15.00	Contact
[Blank]	Clicker	Quality: Fair	\$25.00	Contact
[Blank]	Clicker	Quality: New	\$50.00	Contact
[Blank]	Clicker	Quality: Used	\$20.00	Contact
[Blank]	Click	Quality: Fair	\$20.00	Contact

At the bottom center, there are navigation arrows for '1 of 2 pages'.



About All

SFSU Trade Mart Create Listing Sign In / Register

Clicker
From: User2343 (4.5/5 14 reviews)
Quality: Used

Suggested Price:
\$15.00

Description: Only used about once or twice. Works good.

Listing ID: 987654321



About All

SFSU Trade Mart Create Listing Sign In / Register

Contact

 img here

Listing Title: Clicker
Condition: Used
Seller's preferred location: Outside bookstore
OR
Your Preferred location: Select a location
Pick-up Time: 12:00 pm
Suggested Price: \$15
OR
Description: Only used about once or twice. Works good
Make an Offer: \$10
Message (Be sure to include contact information): Hi. Would you take \$10? You can contact me at my school e-mail *****@mail.sfsu.edu

 [About](#) [SFSU Trade Mart](#) [Create Listing](#) [Sign In / Register](#)

Contact

You must have an account to make an offer

LOG IN

Username

Password

OR

[Create an account](#)

[Or Forgot Password](#)

 [About](#) [SFSU Trade Mart](#) [Create Listing](#) [Sign In / Register](#)

Register Account

* First Name * = Required

* Last Name

* Username

* Email address

Only accepts SFSU email

* Enter Password

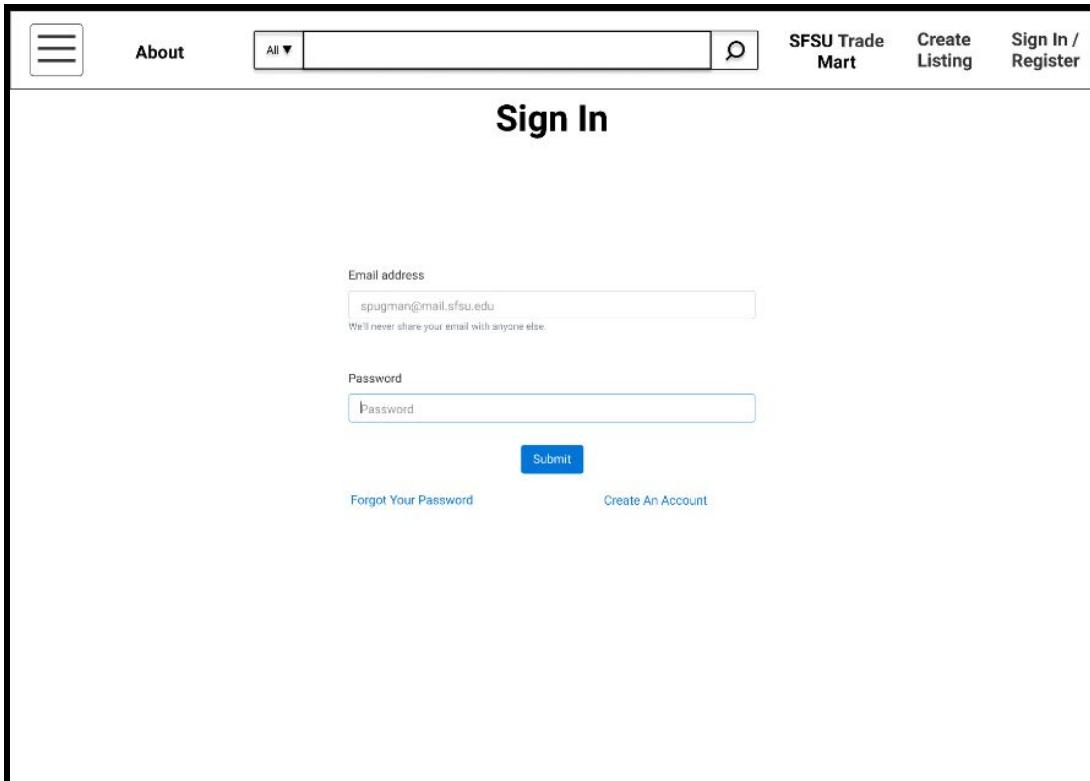
* Confirm Password

I accept the Terms of Service.

 I am human.
 Captcha

Use Case #3: A professor just finished teaching her class. She has some additional resources for her class, so she registers on the homepage and logs into her SFSU TradeMart account and lists them as optional materials in the class resource form, so her future students can see them when they search by class. On hand, she also has extra copies left over from previous years so she posts some listings on the site. The site is easy to navigate and she completes this process in a couple of minutes with the details she wants to include. Afterwards, the site provides a my listings tab that informs her of pending offers. For the next semester, she is able to update her class resource form to account for the new version of the textbook she is using.

The screenshot shows the 'Register Account' page of the SFSU TradeMart website. At the top, there is a navigation bar with links for 'About', 'All ▾', a search bar, 'SFSU Trade Mart', 'Create Listing', and 'Sign In / Register'. The main title 'Register Account' is centered above the form fields. The form consists of several input fields: 'First Name' (Sarah), 'Last Name' (Pugman), 'Username' (Pug1234), 'Email address' (spugman@mail.sfsu.edu), 'Enter Password' (*****), 'Confirm Password' (*****), and a checkbox for 'I accept the Terms of Service'. Below the password fields is a CAPTCHA box containing a grid with an 'X' and the text 'I am human. Captcha'. At the bottom are 'Cancel' and 'Submit' buttons.



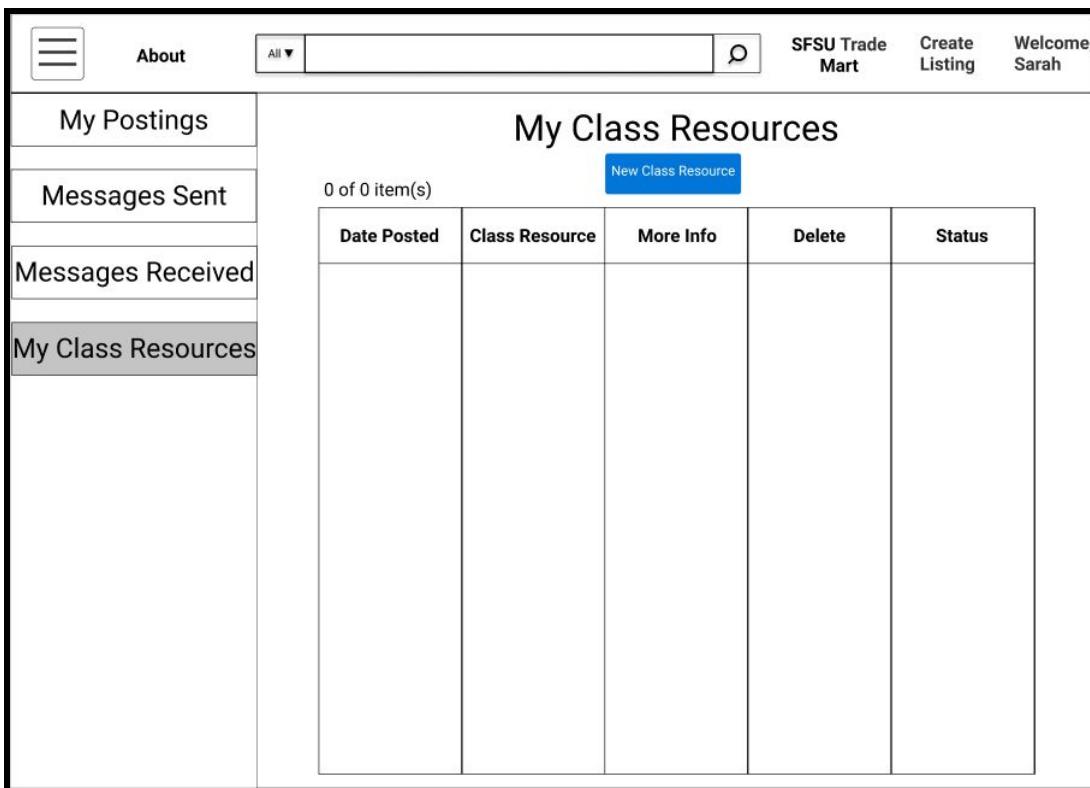
The Sign In page features a header with a menu icon, 'About', a search bar, and links for 'SFSU Trade Mart', 'Create Listing', and 'Sign In / Register'. The main title 'Sign In' is centered above a form. The form includes fields for 'Email address' (spugman@mail.sfsu.edu) and 'Password', a 'Submit' button, and links for 'Forgot Your Password' and 'Create An Account'.

Email address
spugman@mail.sfsu.edu
We'll never share your email with anyone else.

Password

Submit

Forgot Your Password Create An Account



The My Class Resources page shows a sidebar with 'My Postings', 'Messages Sent', 'Messages Received', and 'My Class Resources' (which is highlighted). The main area displays 'My Class Resources' with a count of '0 of 0 item(s)' and a 'New Class Resource' button. A table below has columns for 'Date Posted', 'Class Resource', 'More Info', 'Delete', and 'Status'.

About All ▾ SFSU Trade Mart Create Listing Welcome, Sarah

My Postings

Messages Sent

Messages Received

My Class Resources

My Class Resources

0 of 0 item(s)

New Class Resource

Date Posted	Class Resource	More Info	Delete	Status

 [About](#) [SFSU Trade Mart](#) [Create Listing](#) [Welcome, Sarah](#)

Class Resource Form

Book Title

ISBN

Edition/Version

Number only. Ex: 1, not 1st.

Term

Department

Course

Section

 [About](#) [SFSU Trade Mart](#) [Create Listing](#) [Welcome, Sarah](#)

Your class resource has been submitted. Once it has been approved by our Admin, listings of the resource provided will be associated with your class.

[About](#)

All ▾

[SFSU Trade Mart](#)[Create Listing](#)[Welcome, Sarah](#)

Create A Listing

Listing Title

Category

Rating (condition)

Price

Make this price fixed

Location

Description

Image [Browse](#)

[Cancel](#)[Submit](#)

Use Case #4: An office worker has just finished sending out a batch of emails to her colleagues and has a free moment to **approve listings**. Since she has such scattered moments of reprieve throughout her workday, she can't devote long stretches of time to navigating through the website as an **admin** to approve all incoming student and teacher listings. However, since the UI was easy and quick to navigate, she was able to **approve** a healthy amount of posts and **disapprove posts** with reasons while reading over the necessary details of each one during her small break. (*Priority 3 so not implemented in Priority 1. We will be using workbench instead.*)

The screenshot shows the login interface for the SFSU Trade Mart. At the top, there is a navigation bar with icons for a menu, About, Books (with a dropdown arrow), a magnifying glass search icon, SFSU Trade Mart, Create Listing, and Sign in/Register. Below the navigation bar is a large, empty rectangular area, likely a placeholder for a banner or advertisement. The main form area contains fields for Email address and Password, both with placeholder text ('Admin' and '*****' respectively). Below these fields are links for 'Forgot Your Password' and 'Create An Account', and a blue 'Submit' button.

Email address
Admin
We'll never share your email with anyone else.

Password

Forgot Your Password Create An Account

Submit

This screenshot shows the Admin Home page. On the left is a vertical sidebar with links: Admin Home, Listing Submissions, Class Resource Updates, and Complaints/Reports. The main content area displays a welcome message and three summary statistics:

- Welcome Admin
- There are X amount of Listing Submissions that need to be reviewed
- There are X amount of Class Resource Updates that need to be reviewed
- There are X amount of Complaints/Reports that need to be reviewed

This screenshot shows the Admin Home page after receiving new listing submissions. The sidebar remains the same. The main content area now lists three new submissions with "Accept" and "Reject" buttons:

Submission Details	Action
Peter has submitted a Listing	See Form Accept Reject
Mary has submitted a Listing	See Form Accept Reject
Donald has submitted a Listing	See Form Accept Reject

Use Case #5: A student is moving out of her dorm and has some furniture that she doesn't want to take. She remembers that SFSU Trademart has a section for **furniture listings**. She makes sure to take plenty of **pictures to show the conditions** and is easily able to make detailed posts for everything she needs to get rid of. Before sending her first post off to be approved, she is **asked to log in or register an account**, which she fills out quickly and gets on with submitting the rest of her posts. Since she is on a time crunch, she **sets a non-fixed price** rather than forcing buyers to offer only a fixed price.

The screenshot shows a web-based application for creating a listing. At the top, there's a navigation bar with links for 'About', 'Books' (with a dropdown arrow), a search icon, 'SFSU Trade Mart', 'Create Listing', and 'Sign in/Register'. Below the navigation is a title 'Create A Listing'. The form consists of several input fields: 'Listing Title' (text input: 'Red Couch'), 'Category' (dropdown menu: 'Furniture'), 'Rating (condition)' (dropdown menu: 'Fair'), 'Price' (text input: '50.00') with a checkbox below it labeled 'Make this price fixed' (unchecked), 'Location' (dropdown menu: 'SFSU Library Cafe'), 'Description' (text area: 'It's a red couch that can seat about 4 people.'), and 'Image' (file input field with a 'Browse' button). Below the image input, there's a list of five files: 'image00.png', 'image01.png', 'image02.png', 'image03.png', and 'image04.png', each with an 'X' next to it. At the bottom left is a 'Cancel' button, and at the bottom right is a large 'Submit' button.

Field	Value
Listing Title	Red Couch
Category	Furniture
Rating (condition)	Fair
Price	50.00
Description	It's a red couch that can seat about 4 people.
Image	image00.png, image01.png, image02.png, image03.png, image04.png

SFSU Trade Mart

About Books Create Listing Sign in/ Register

Create Listing

X

Listing Title You must have an account to post a listing

Category

Rating (cond)

Location

Description

Image

LOG IN

Username

Password

OR

Create an account

[Or Forgot Password](#)

image00.png X
image01.png X
image02.png X
image03.png X
image04.png X

Cancel **Submit**

SFSU Trade Mart

About All Create Listing Sign In / Register

Register Account

* First Name * = Required

* Last Name

* Username

* Email address

Only accepts SFSU email

* Enter Password

* Confirm Password

I accept the Terms of Service.

I am human.
Captcha

Cancel **Submit**

Your listing has been submitted. Once it has been approved by our Admin, the listing will be posted on the site. Be sure to check your notifications later to see if the listing has been approved.

[Return](#)

My Postings

3 of 3 item(s)

Date Posted	Post Title	More Info	Delete	Status
10/10/20	Wooden Desk	More Info	Delete	Pending
9/9/20	Black Chair	More Info	Delete	Pending
8/8/20	Red Couch	More Info	Delete	Pending

5. High Level Architecture, Database Organization Summary:

- Database Organization:

Our database consists of 4 tables: User, Listing, Object, Offer, and Categories. The User table will hold any information that has to do with users of the application, both registered and unregistered. It will also identify who is an admin. The Object table holds information for individual objects, like books or iClickers. It will hold information of any materials that users input using the Class Resource Form, after admin verification, and will help to organize listings by common titles and version numbers. The Listing table will be used to store data specific to a user's listing. Attributes such as the title, price, image, and preferred location will be shown on the listing's Item Page, the Search Results page, and on the buyer and sellers' Make an Offer pages. The Offer table will hold information that has to do with the form on the buyer's Make and Offer page. It also contains data from the Listing table such as both parties' user identification numbers and the listing's identification number. Lastly, the Categories table holds the data from each entry in Listing organized into their appropriate category.

- User:

Attributes:

- user_id
- user_email
- user_name
- user_addr
- user_pass
- reg_status
- user_rating
- user_listings
- is_admin

- Listing:

Attributes:

- list_id
- user_id
- object_id
- list_title
- list_category
- pref_location
- list_desc
- approval_status

- offer_type
- list_date
- list_time
- image
- suggest_price
- condition

- Object:

- Attributes:

- object_id
 - obj_name
 - department
 - course
 - term
 - section
 - version
 - isbn

- Offer:

- Attributes:

- offer_id
 - seller_id
 - buyer_id
 - history
 - listing_id
 - offer_amount
 - location
 - Msg_time

- Categories:

- Attributes:

- books
 - electronics
 - furniture
 - tutoring
 - editing

- Media Storage:

The images and other files related to the post and/or user will be stored as BLOBS, more research is currently needed to define the specifics but we will be going forward with this decision.

- Search/Filter Architecture and Implementation:

SQL and %like will be implemented to use identifiers from the items or users and find the matching characteristics of the desired search.

- Custom APIs:

None known yet.

- Algorithms:

We will be allowing registered users to rate other registered users based on how their interactions went. If a registered user has been rated at least once, the mean of all of the ratings will appear next to their name on the Item Page of amount listings they make.

6. Key Risks For Project:

- Skills Risk: While the team has been learning the tools that will be used to complete the project, learning is an ongoing process. There are bound to be errors or things that will need to be learned that were not accounted for earlier. To solve this should an issue occur, we can search the internet for tutorials or the information we need in order to complete our project, as we have done already up to this point.
- Schedule Risk: In a project where time is a limiting factor, there is always the worry that the team will not be able to implement what we have committed. We have already made efforts to solve this during milestone 1 by reducing scope. We are also prioritizing features and using task management software like Trello to be better able to gauge what needs to be done, when it needs to be done, and who is going to do it.

7. Project Management:

In order to divide the work to complete milestone 2 on time, we used Trello as our task management tool. Everyone made an account and the team lead put everyone into a group. The group contained a to-do board with 3 sections: to-do, doing, and done. This basic format was to make it as easy as possible to use as well as give each team member a quick overview of who had which tasks and where they were at in the process of completing them. The front end team worked mostly on the mockups and other sections of the document, and the back end team was in contact with them as they worked on the vertical prototype. We also have a Discord chat so that any team member can talk to each other in a quick and efficient way. This cut down on the amount of Zoom meetings as we could make quick plan adjustments through text.

The current arrangement described above is how we will plan for future tasks as well.

Milestone 3:

Milestone 3 review summary and plans - CSC 648-848 Fall 2020 Team 07

Date of Milestone 3 review with Prof. D. Petkovic: 11/16/20

Date of this summary and plan document: 11/20/20

Summary of feedback and tasks to do:

1. Going through UI Pages, Professor giving feedback
 - a. Home / Class Resource Form
 - i. Put Class ID above the search bar. Also recommended to change the textbox to a welcome message instead.
 - ii. Ensure carousel items are clickable.
 - iii. Don't mix "SFSU Trade Mart" with other navbar options. Logos go on the left of the navbar/left of the search bar.
 - iv. Hamburger menu on the right; make names of menu options in hamburger options clearer to the user.
 - v. Class Resource Form should only be visible to logged in teachers, on dashboard.
 - vi. Make Class Resource Form (and associated functionality) P2 or P3.
 - b. Search
 - i. Show total item count (Showing items 1-20 out of 55) in the top left.
 - ii. Sort items by price.
 - iii. Display items with more data relevant to the item.
 - iv. Include contact seller button from search results:
 1. Display items one per row, to make it more visible
 2. Also removes the need for a separate item page.
 - v. Don't need two category lists, remove category list from side of search page (drop down list).
 - vi. Add filter by rating (remove rating entirely if you cannot implement this).
 - c. Item Page
 - i. Remove offers, allow only messaging.
 - ii. Make the item page load in a new tab from the search page.
 - d. Create A Listing Page
 - i. Remove (condition) from Rating.
 - ii. Add "It will take up to 24 hours to approve listing" next to the submit button.

- iii. Align fields in popup for lazy registration.
 - iv. Add forgot password in popup.
 - e. Contact Page
 - i. Reorder right side options for a more intuitive approach.
 - ii. Remove offer ability.
 - iii. Message text list should be first.
 - iv. Followed by desired pickup location.
 - v. Then other options.
 - vi. This item should also open in a new tab, or open in a popup if possible.
 - f. Register Page
 - i. Make “must be sfsu email” alert text red.
 - ii. Underline Terms of Service and make it a dummy link.
 - g. Sign In Page
 - i. Submit should be below forgot password.
2. Other Notes
- a. Focus on P1 stuff for final submission.
 - b. Don’t add functions to P1.
 - c. Make sure everything is responsive.
3. Database Tables
- a. Take out offers.
4. GitHub
- a. Make commit comments specific.
 - i. Don’t use “some”, “this”
 - b. Include purpose of commit in header comment.

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

We have decided to cut the class resource form/class resource search function as well as offers and will focus on completely connecting the current pages with the database while improving the user experience. This will be done by following the feedback above to improve the UI so the user has an easier time navigating the site.

List of final product P1 functions agreed at the meeting:

All Users (Unregistered/Registered):

- 1: All users shall be able to register accounts with the website.
- 2: All users who have not logged in shall be able to access the login/register page, and log in through any page of the website.
- 3: All users who are not logged in shall be suggested the option to log in or register upon making an offer or submitting a post for approval.

- 4: All users shall be able to view listings on the website.
- 5: All users shall be able to search for specific listings based on words in its title, description, or category.
- 6: All users shall be able to see descriptions of the items for sale on the search results and browsing pages of the website.

Registered Users Only:

- 7: Users who are logged in shall be able to log out.
- 8: Registered users shall be able to create listings for their items and put them on sale through the website, thus known as posts. Postings contain the name of the seller, condition, price and whether it is negotiable.
- 9: Registered Users interested in specific listings shall be able to contact the seller to decide themselves the details of the transaction.
- 10: Registered Users shall be able to choose the time and meeting location from a set of predefined locations when contacting a seller.
- 11: Registered Users shall be able to access a dashboard either upon signing in or clicking the “Welcome [username]” on the navbar, where they can access three tabs, My Postings, Messages Sent, and Messages Received.

Admins Only:

- 12: Admins shall be required to approve or disapprove listings according to the TOS using Workbench.
- 13: Admins shall be required to remove users for violating the TOS using Workbench.

Milestone 4:

SW Engineering CSC 648/848 Fall 2020
SFSU Trade Mart
Team 07

Team Members:

Team Lead: Alicia Ramirez
(Email: aramirez23@mail.sfsu.edu)

Front-End Lead: Jonathan Pak

Back-End Lead: Valeria Vallejo

Github Master: Ricardo Carretero

Member: Chandler Cruz

Member: Chris Manaoat

Milestone 4
December 8, 2020

History Table

Date Submitted: December 8, 2020
Date Revised: December 15, 2020

1. Product Summary:

Name of the product: SFSU TradeMart

All Major Committed Functions:

All Users (Unregistered/Registered):

- 1: All users shall be able to register accounts with the website.
- 2: All users who have not logged in shall be able to access the login/register page, and log in through any page of the website.
- 3: All users who are not logged in shall be suggested the option to log in or register upon making an offer or submitting a post for approval.
- 4: All users shall be able to view listings on the website.
- 5: All users shall be able to search for specific listings based on words in its title, description, or category.
- 6: All users shall be able to see descriptions of the items for sale on the search results and browsing pages of the website.
- 7: Users who are logged in shall be able to log out.
- 8: Registered users shall be able to create listings for their items and put them on sale through the website, thus known as posts. Postings contain the name of the seller, condition, and price.
- 9: Registered Users interested in specific listings shall be able to contact the seller to decide themselves the details of the transaction.
- 10: Registered Users shall be able to choose the time and meeting location from a set of predefined locations when contacting a seller.
- 11: Registered Users shall be able to access a dashboard either upon signing in or clicking the “Welcome [username]” on the navbar, where they can access three tabs, My Postings, Messages Sent, and Messages Received.
- 12: Admins shall be required to approve or disapprove listings according to the TOS using Workbench.
- 13: Admins shall be required to remove users for violating the TOS using Workbench.

URL: <http://52.53.215.116/>

2. Usability Test Plan:

Test Objective

The selected function we are testing for this usability plan is the search function. This test is to ensure the search function is easy and fast to use for users. The test objectives for this usability test plan is based on how the user can execute these actions:

- Search product listings by category
- Search product listings by title and description
- Sort product listings by price

Through this testing we will be able to take feedback from users and use it to improve the ease of use for our search function.

Test Background and Setup

The test is to receive usability feedback on the search function for the SFSU TradeMart website. This includes searching by category, title, and description as well as sorting based on price. The web browsers it will be tested on are Google Chrome, Firefox, and Safari. The intended users of the site are SFSU students and faculty.

The test will begin on the website's homepage on Google Chrome ([52.53.215.116](#)). The user will be tasked with searching for certain items, as well as utilizing the sorting options. They will receive no assistance other than starting on the homepage and being given these tasks. Once the user has completed these tasks we will assess the results of the usability test through the criteria discussed in the Usability Task Description table and the responses of the Likert test.

The URL of the website is [52.53.215.116/](#) which is also the URL of the search page we are testing. The user can input the search term which can include the option of specifying a category from a dropdown menu. Once searched, they can then sort their results from a dropdown menu above the listings. The usability test is successful and completed if the user finds the appropriate result by searching the product listings available.

Usability Task Description

Task	Description
Task	Find all furniture listings sorted by lowest to highest price
Machine State	Search query empty
Successful completion criteria	Search results properly loaded related to search term with chosen sorting option correctly applied

Benchmark	Completed in 1 minute
-----------	-----------------------

Effectiveness: Effectiveness would be measured by how many testers were able to properly complete the task in time and how many errors they made (clicks that were not necessary to complete the task).

Efficiency: Efficiency would be measured by the average time it took testers to complete the task, the average number of clicks it took to accomplish the task, and the average number of screens needed to accomplish the task.

Subjective Testing

(“X” column that is your answer)

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was able to find the search bar easily.					
I was able to find the desired listings easily.					
I was able to sort the search results to search listings easily.					

Additional Comments (if any):

3. QA Test Plan:

Test Objectives

We are testing the search function and its sorting function to see if relevant search results are listed and sorted in the correct order.

HW and SW setup

Hardware: Windows 10 PC

Software setup: The setup will have the website's homepage (which is also the search page) loaded on two different browsers: Google Chrome and Firefox. We check to make sure the navbar at the top of the screen contains the search bar as well as the category dropdown list.

URL: <http://52.53.215.116/>

Feature to be tested

Search/Sorting Function

QA Test plan - table format

Number	Title	Description	Test Input	Expected Output	PASS/FAIL Chrome	PASS/FAIL Firefox
1	Test % like in search for name field	The search button should return the search results page with the correct results.	Type "iclicker" into the search bar with "All" as the category option and click the search button.	The search results page contains two listings: "take my iclicker" and "iclicker for any class".	PASS	PASS
2	Test category in search	The category search should return only listings that are of the selected category.	Select the "Tutoring" category with no search term and click the search button.	The search results page contains three listings: "Math Tutoring (1 hour)", "Art Mentor", and "Physics Tutoring (1 hour)".	PASS	PASS

3	Test search result sorting	The results should order themselves according to the chosen sorting option.	Select the “Books” category with no search term and click the search button. Click the “sorted by” button and select “Prices low to high”.	The search results contains four listings in this order: “cool math games textbook, math 100”, “engineering textbook, hardcover”, “calc 3 text book”, and “Math 100 Book for Sale”	PASS	PASS
---	----------------------------	---	--	--	-------------	-------------

4. Code Review:

Done over email as specified. Screenshots below:

CSC 648 Fall 2020 Section 3 Team 07 Milestone 4 Code Review

 **Ricardo Carretero**
Tue 12/8/2020 5:38 PM
To: Chris Jerome Belino Manaoat
Cc: Alicia Jane Ramirez; Dragutin Petkovic; Valeria Vallejo; Chandler Thomas Cruz; Jonathan Pak



3 attachments (200 KB) [Download all](#)

As requested, the code review has been completed for the file and comments have been added to the master branch. See attached images.

Regards,
Ricardo.

...

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

 **Chris Jerome Belino Manaoat**
Tue 12/8/2020 5:31 PM
To: Ricardo Carretero
Cc: Alicia Jane Ramirez; Dragutin Petkovic; Valeria Vallejo; Chandler Thomas Cruz; Jonathan Pak

Hello,
I am attaching a link to one of our files on GitHub (website.py) to get a code review. Our QA focused on the search function which can be found on lines 137-259 in this file.

<https://github.com/CSC-648-SFSU/csc648-03-fa20-team07/blob/master/application/website.py>

Thank you,
Chris Manaoat
Sent from [Mail](#) for Windows 10

```
dasfiter@ip-172-31-7-74: ~/CSC648/application
#
# website.py
# Description:
# This file handles the routing of each page endpoint and takes care of the queries
#
# Contents:
# -endpoints
# -function to convert blobs to images
#
# TODO implement rest of the endpoints with sessions
"""

Review: Need to standardize the usage of ' vs ", some verbose usage of if and
elif is fine and helps with readability. Some of the lines could be shorted to
one liners, but it would be more difficult to read/debug for those with an
untrained eye.
"""

# imports
import pathlib
import random
import re
import subprocess
import sys

from db_tools.hashing_tools import *
from db_tools.encrypt_tools import *
from db_tools.key import *
from flaskext.mysql import MySQL
from flask import Flask, redirect, url_for, render_template, request, session
from PIL import Image
# end imports

app = Flask(__name__)
app.secret_key = 'csc648sfutrademart' # key for session purposes

"website.py" 529L, 24773C
11,52 Top ▾
```

```
dasfiter@ip-172-31-7-74: ~/CSC648/application
# TODO: make sure to default rest of the acc info(if needed) and revert default register to 0
idExists = True # check if user id exists
userId = random.randint(100000000, 99999999) # generate id
while idExists: # loop for id generation will quit if id does not exist
    returnId = cursor.execute('SELECT user_id'
                             'FROM Trademart.User'
                             'WHERE user_id= % s ', userId)
    conn.commit()
    # print(returnId)
    if returnId: # id exists so randomize another
        userId = random.randint(100000000, 99999999)
        # print(userId)
    else: # id doesn't exist so it will quit loop with appropriate id
        idExists = False
        # print('id can be created')
    # print(userId)
"""
Review, this exact code snippet is used in encrypt tools, is there a reason it is
repeated here and not called from that file as a function?"""
# at this point all user data is verified (no repeating username or email address)
key = load_key() # gets key
f = Fernet(key) # makes appropriate fernet key
encryptedEmail = encrypt_email(userId, email, f) # encrypts the email to be stored
# print(encryptedEmail)

passToHash = str(passwordConfirm)+'CSC675' # adds appropriate suffix when hashing
hashedPass = hash_password(userId, passToHash) # creates hashed pass to be stored
# print(hashedPass)
accCreated = cursor.execute('INSERT INTO Trademart.User'
                            '(user_id, user_email, fname, lname, user_name, user_pass, reg_status)'
                            'VALUES(%s, %s, %s, %s, %s, %s, %s)', (userId, encryptedEmail, firstname, lastname, username, hashedPass,
                            '0'))
conn.commit()
if accCreated: # if query was successful
    return redirect(url_for('home', message='Account successfully created!', popUp='True', username=username))
}

354,12 67% ▾
```

```
dasfiter@ip-172-31-7-74: ~/CSC648/application
    return render_template('itempage.html', data=data, username=username) # load listing page
    return render_template('itempage.html') # load listing page

# this function converts a blob to an image of type jpg
def blob2Img(listing):
    fileName = str(listing[3]) + '.jpg' # the file name using listing id
    path = '/home/dasfiter/CSC648/application/static/listing_images/' +fileName # path to image
    # path = 'static/listing_images/' +fileName # path to image
    #print(path)
    # size = sys.getsizeof(listing[11])
    # print(size)
    #print(listing[2])
    sizes = [(4, 'quarter'), (2, 'half')] # resize values
    if listing[2]: #checks if pulled image from DB isn't empty
        test_path = pathlib.Path(path) # gets path
        if not test_path.exists(): # if path doesnt exist
            #print('exists')
            with open(path, 'wb') as file: # open the file
                file.write(listing[2]) # convert blob to image
                """Review: file close not needed, with open automatically closes the file pipe
                   once it goes out of scope"""
                file.close()
        # loop to create thumbnails
        for size, name in sizes:
            im = Image.open('/home/dasfiter/CSC648/application/static/listing_images/%s' % fileName) # opens image
            # im = Image.open('static/listing_images/%s' % fileName) # opens image
            im.thumbnail((im.width//size, im.height//size)) # creates thumbnail of ratio size
            im.save('/home/dasfiter/CSC648/application/static/listing_images/thumbnail_%s_%s_size.jpg' % (fileName[:-4], name)) #saves image
            # im.save('static/listing_images/thumbnail_%s_%s_size.jpg' % (fileName[:-4], name)) #saves image

if __name__ == '__main__':
    app.run(debug = True)
```

516,20 Bot ▾

5. Self-Check on Best Practices for Security:

Major assets being protected:

- User's password
- User's email
- User's address
- Listing's image

Major threats to assets:

- Admin getting access to private information from Users through MySQL Workbench.
- Access to our server via SSH from anywhere.
- Insertion of SQL queries in the search bar that could access our database.
- Listing images getting corrupted upon upload/insertion to the database.

How we're protecting each asset:

- We're protecting our users' passwords by taking each user's password, inserting the string 'CSC675' as a suffix to their password, and hashing that altered password before inserting it into the MySQL database. For the user's emails, we hash them before inserting them into the database. This will prevent any site admins from having access to user sensitive information, even with direct access to the database. Admins are also given limited privileges with the database account they're given, so they aren't able to make drastic changes to our database.
- To prevent unwanted access via SSH to our server, we added inbound rules that only allow verified connections. This is in addition to the default security group through AWS.
- To prevent users from making SQL queries directly from the search bar, we implemented data validation directly into the search bar. We limited the user's search bar input to 40 alphanumeric characters maximum. In addition to the search bar, we also implemented data validation into the register and sign-in pages to ensure users were inputting SFSU email addresses.
- To protect the listing images from being corrupted, we've stored them as BLOBs rather than paths in the database. This way, we don't have to give remote clients file system access.

6. 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 - **ON TRACK**
3. All or selected application functions must render well on mobile devices - **ON TRACK**
4. Data shall be stored in the database on the team's deployment server. - **DONE**
5. No more than 50 concurrent users shall be accessing the application at any time - **ON TRACK**
6. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users. - **ON TRACK**
7. The language used shall be English (no localization needed) - **DONE**
8. Application shall be very easy to use and intuitive - **ON TRACK**
9. Application should follow established architecture patterns - **ON TRACK**
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 - **ON TRACK**
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development - **ON TRACK**
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "*SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only*" at the top of the WWW page. (Important so as to not confuse this with a real application). - **ON TRACK**

(Back to Milestone 5) 4. Product Screen Shots:

The screenshot shows a web browser window for the "SFSU Trade Mart" website. The title bar indicates it's a "Not secure" connection. The page header includes the site name, a search bar, and navigation links for "About", "Create", "Sign In/Listing", and "Register". A banner at the top says "Welcome to SFSU Trademart" and "New Listings". Below this, there is a large image of a computer case with multiple RGB-lit fans. Below the image, the text "PC" and "\$100" is displayed.

SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7

All Search About Create Sign In/
Listing Register

Welcome to SFSU Trademart

New Listings

PC
\$100

SFSU Trade Mart Sign In

Not secure | 52.53.215.116/signIn

SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7

About Create Sign In/
Listing Register

Search

Sign In

* Email address

We'll never share your email with anyone else

Password

[Forgot your password](#) [Create An Account](#)

Submit

SFSU Trade Mart Sign In Register

Not secure | 52.53.215.116/register

SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7 About Create Sign In/ Listing Register

All Search

Register Account

Note: It may take up to one day for your account to be in the system.

* = Required

* First Name

* Last Name

* Username

* Email address

Only accepts SFSU email

* Enter Password

* Confirm Password

I accept the [Terms of Service](#)

I'm not a robot 
reCAPTCHA
[Privacy](#) • [Terms](#)

SFSU Trade Mart | Sign In | Register | Create Listing | Not secure | 52.53.215.116/createListing | SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only | Incognito | ...

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7 About Create Sign In/ Listing Register

Create A Listing

* = Required

* Listing Title:

* Category:

* Rating:

* Price:

Make this price fixed

* Location:

Description:

Image: No file chosen

**It will take up to 24 hours to approve listing

The screenshot shows a web browser window with five tabs open:

- SFSU Trade Mart
- Sign In
- Register
- Create Listing
- About Us

The current page is the "About Us" page, indicated by the active tab. The page title is "Software Engineering class SFSU Fall 2020 Section 3 Team 7". The header includes a search bar with dropdown options "All" and "Search", and a magnifying glass icon. To the right of the search bar are links for "About", "Create", "Sign In/", "Listing", and "Register".

About Us

[About Chris](#) [About Ricardo](#) [About Valeria](#) [About Alicia](#) [About Chandler](#) [About Jonathan](#)

SFSU TradeMart Not secure | 52.53.215.116/search SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7 About Create Sign In/ Listing Register

All test

Sorted By



Math 100 Book for Sale

Suggested Price: \$50

Description: required textbook for math 100 with Professor

Quality: good

[View More Details](#)



Math Tutoring (1 hour)

Suggested Price: \$20

Description: Tutoring for any Algebra or Calculus class

Quality: n/a

SFSU TradeMart Not secure | 52.53.215.116/listing

Gaming PC Case

From: cpugman

Quality: poor

Suggested Price:

\$10

Description: Lighting doesn't work anymore, cracked on one side

Contact Seller

SFSU TradeMart | SFSU Trade Mart | SFSU TradeMart | Contact Seller | Listing Register

Not secure | 52.53.215.116/contact

Mart Search Listing Register

Contact



Listing Title: calc 3 text book

Condition: great

Seller's preferred location: Plaza

Message:

Listing ID: 45735

Description	Price:
current version of the book that prof. a uses, bought it new	47

Your preferred location: Bookstore

SFSU TradeMart | SFSU Trade Mart | SFSU TradeMart | Contact Seller | Not secure | 52.53.215.116/contact

Mart All Search Listing Register

Contact

You must have an account to contact a seller

LOG IN

Email:

Password:

[Forgot Password](#)

Submit

Listing ID: 45735

OR

[Create an Account](#)

current version of the book that prof. a uses, bought it new

Your preferred location:

Cancel Contact

SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

Software Engineering class SFSU Fall 2020 Section 3 Team 7

About Create Welcome, Listing cpugman ▾

Dashboard

My Postings

8 of 8 item(s)

Date Posted	Post Title	More Info	Delete	Status
2020-10-05	Gaming PC Case	More Info	Delete	Approved
2020-05-20	Intro to Drawing Textbook	More Info	Delete	Pending
2020-05-15	calc 3 text book	More Info	Delete	Approved
2020-05-10	cool math games textbook, math 100	More Info	Delete	Approved

SFSU TradeMart x | SFSU Trade Mart x | SFSU TradeMart x | Contact Seller x | SFSU TradeMart x |

Not secure | 52.53.215.116/dashboard/ SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only

SFSU Trade Mart Software Engineering class SFSU Fall 2020 Section 3 Team 7 About Create Welcome, Listing cpugman ▾

All Search 

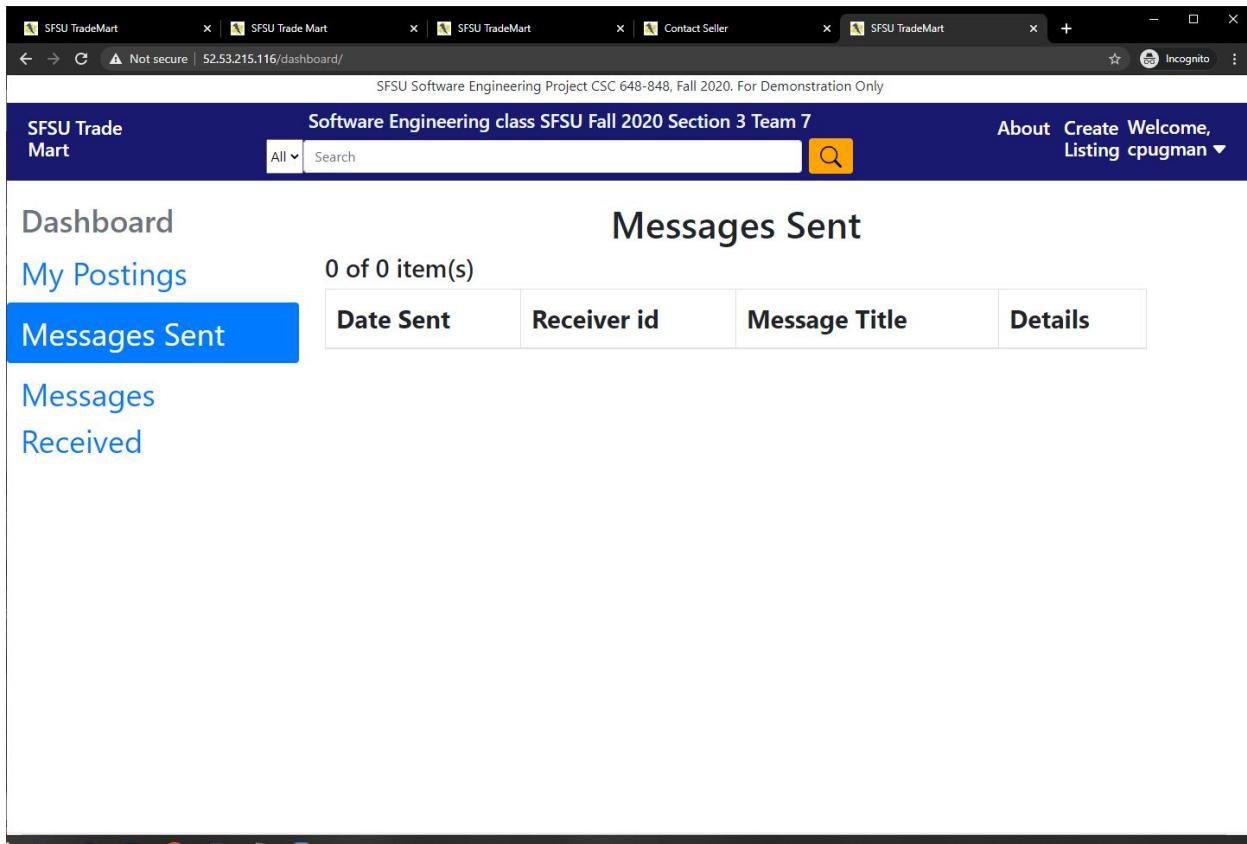
Dashboard **Messages Sent**

My Postings

Messages Sent **0 of 0 item(s)**

Date Sent	Receiver id	Message Title	Details
-----------	-------------	---------------	---------

Messages Received



5. Database Organization:

User Table:

	user_id	user_email	fname	lname	user_addr	user_name	user_pass	reg_status	is_admin
▶	194724822	BLOB	Kahleed	A	BLOB	kahleed	\$5\$ounds=535000\$ux6a04v6AyHH.YSP\$/5pR...	1	0
	265926826	BLOB	Donald	B	BLOB	donald	\$5\$ounds=535000\$4sScXlRzLJCu0mQM\$6yer2...	1	0
	266926598	BLOB	C	Pugman	BLOB	cpugman	\$5\$ounds=535000\$YajppXYT83ECrbF\$sETTa...	1	0
	280020835	BLOB	first	last	NULL	user2	\$5\$ounds=535000\$vnN7rszC9Nc00pUKq\$mfR4...	1	0
	355893574	BLOB	test	testo	NULL	best	\$5\$ounds=535000\$1RFYu92rHxD/SE\$K2l0AL...	0	0
	373967284	BLOB	Ken	D	BLOB	kend	\$5\$ounds=535000\$GKwdLPGdgFC7JCs\$J9nB...	0	0
	384927509	BLOB	Charlie	E	BLOB	chalie	\$5\$ounds=535000\$fmRQ9f9Tlb6yMdNn\$2Wu...	0	0
	491490194	BLOB	first	last	NULL	user4	\$5\$ounds=535000\$bu129Lu14oFxNgk6ff.Tm...	0	0
	572957392	BLOB	Lisa	F	BLOB	lisa	\$5\$ounds=535000\$GGLpnTj9fhYmwMEA\$Ohxt...	1	1
	580556409	BLOB	first	last	NULL	user	\$5\$ounds=535000\$tkpgxXEUeI1uhOIJ\$xansw...	1	0
	854309436	BLOB	first	last	NULL	user3	\$5\$ounds=535000\$qcxjnV1GUEiFl8o\$klIs6Q7/...	1	0
	995711381	BLOB	Bob	Smith	NULL	bobsmith	\$5\$ounds=535000\$HXaNblxlnr/IEuWa\$uyeq3...	0	0

Listing Table:

	list_id	user_id	list_title	list_category	pref_location	list_desc	approval_status	list_date	list_time	image	suggest_price	listing_condition
▶	12381	265926826	Math 100 Book for Sale	Books	Gym	required textbook for math 100 with Professor	1	2020-01-01	12:39:19	BLOB	50	good
	23421	194724822	pointer textbook for classes in c/c++	Books	Park	Best textbook for automatic A+	0	2020-01-01	11:23:12	BLOB	10000	great
	23466	373967284	Math Tutoring (1 hour)	Tutoring	Library	Tutoring for any Algebra or Calculus class	1	2020-01-28	05:25:32	BLOB	20	n/a
	27472	373967284	engineering textbook, hardcover	Books	Entrance	no notes written inside	1	2020-01-01	01:43:12	BLOB	40	great
	29472	265926826	Essay Editing	Editing	Remote	I will edit any essay for you, max 15 pages.	1	2020-03-12	23:00:00	BLOB	10	n/a
	34535	266926598	Art Mentor	Tutoring	Art Room	I can provide help with painting techniques as w...	1	2020-01-23	23:42:26	BLOB	25	n/a
	34633	384927509	Metal Bed Frame (Full)	Furniture	Dorms	Solid frame, box spring not included. One leg is...	0	2020-05-15	15:16:16	BLOB	35	good
	35653	266926598	Gaming PC Case	Electronics	Dorms	Lighting doesn't work anymore, cracked on one ...	1	2020-10-05	11:26:22	BLOB	10	poor
	45432	265926826	selling my house plants (priced per plant)	Furniture	Entrance	18 total, looking for someone with house plant ...	1	2020-05-10	04:03:00	BLOB	5	great
	45452	266926598	Intro to Drawing Textbook	Books	Garden	needed for ART101	0	2020-05-20	14:15:16	BLOB	15	fair
	45635	266926598	creative writing editor	Editing	Remote	I'll look at your story, fix mistakes, and give advi...	1	2020-01-20	12:34:13	BLOB	50	n/a
	45735	266926598	calc 3 text book	Books	Plaza	current version of the book that prof. a uses, b...	1	2020-05-15	11:42:42	BLOB	47	great
	53453	384927509	50" TV	Electronics	Dorms	TV for sale, only used for one semester	0	2020-05-15	15:15:15	BLOB	300	great
	54643	266926598	cool math games textbook, math 100	Books	Quad	very useful for learning basic math in a fun way....	1	2020-05-10	15:25:30	BLOB	23	good
	58374	265926826	idclerk for any class	Electronics	Park	works well, slightly scratched on the corner	1	2020-04-26	04:42:23	BLOB	20	fair
	64563	266926598	usb mic	Electronics	Cafe	good mic, no stand	0	2020-01-21	20:33:07	BLOB	25	good
	78234	373967284	Physics Tutoring (1 hour)	Tutoring	Library	Tutoring for any Physics class by a Physics Grad...	1	2020-01-28	05:23:40	BLOB	20	n/a
	98245	266926598	Editing for CS paper	Editing	Remote	Editing for CSC300GW and more	1	2020-05-01	23:42:23	BLOB	10	n/a
	4830...	265926826	PC	Electronics	Bookstore	It's a PC.	1	2020-12-17	20:49:10	BLOB	100	good

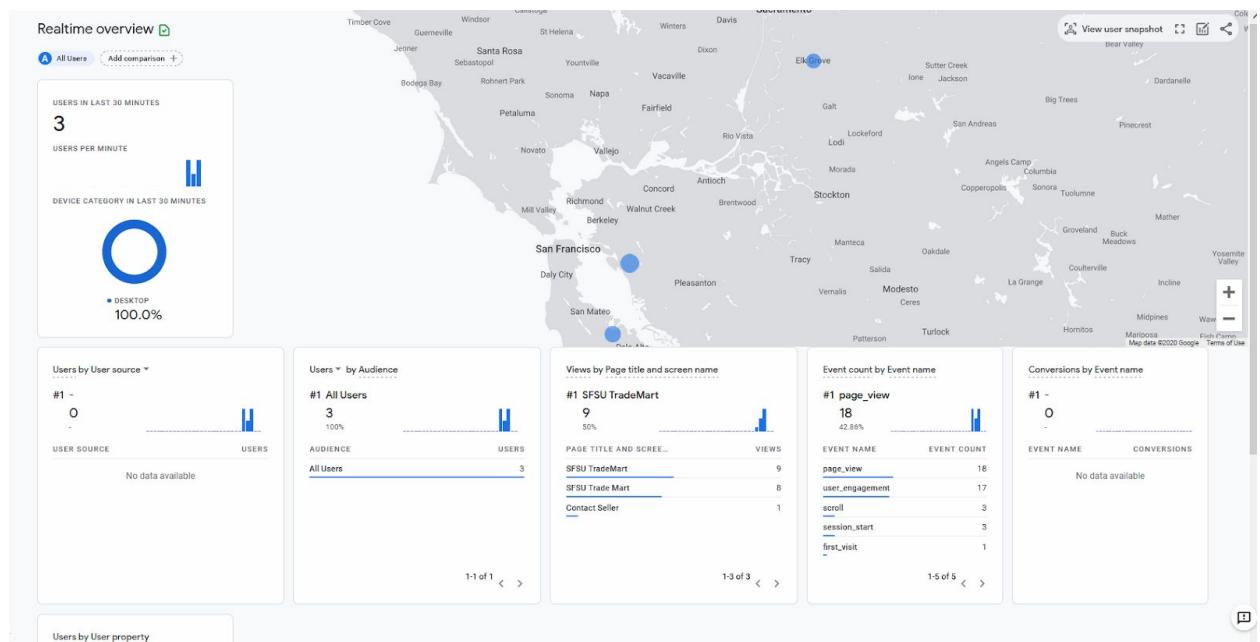
Message Table:

	sender_id	receiver_id	offer_id	title	text	msg_datetime
▶	265926826	265926826	269383741	Interested in Math 100 Book for Sale	Give me book!	2020-12-17 23:11:11

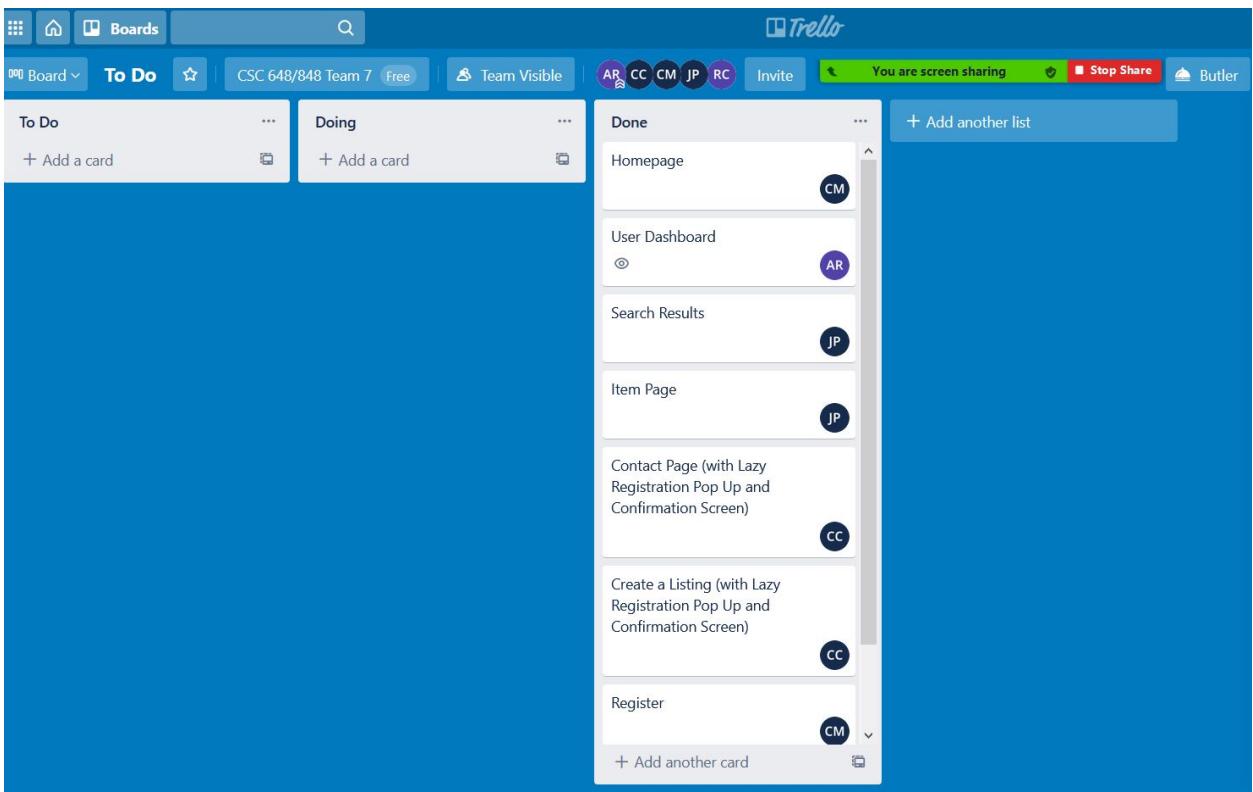
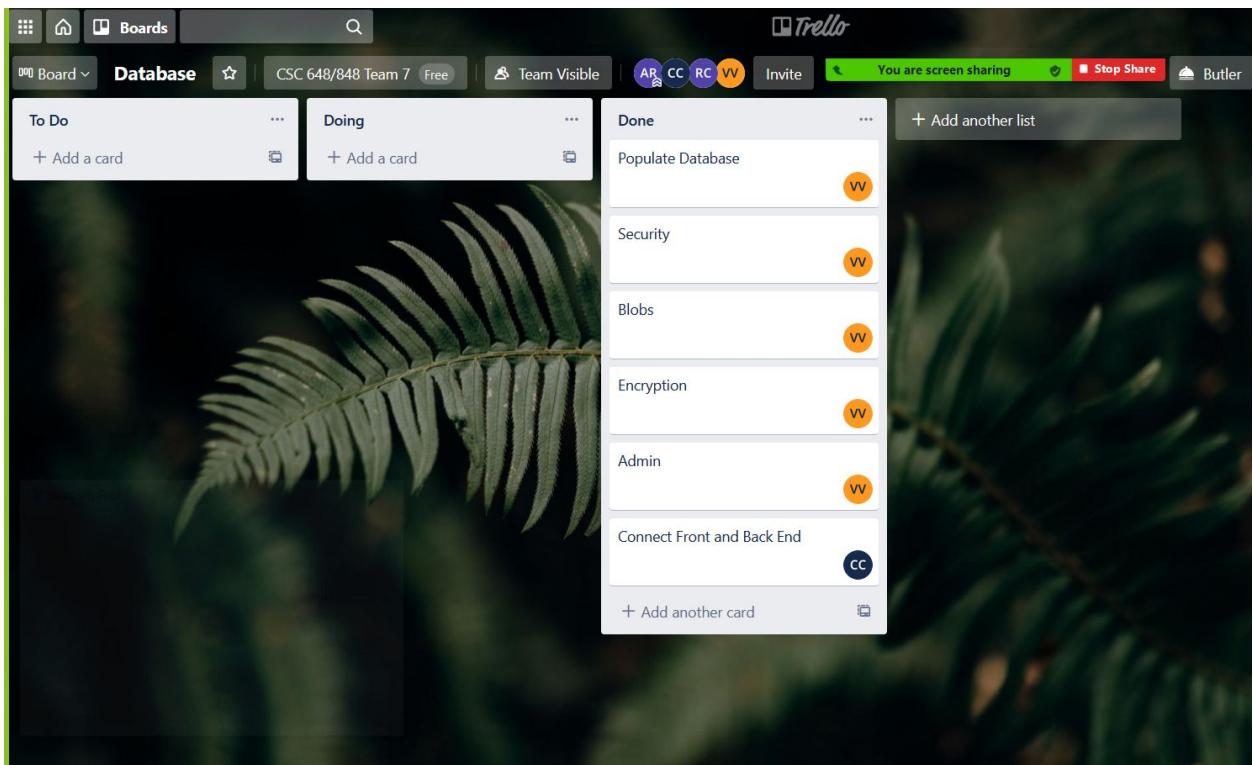
Offer Table:

	offer_id	seller_id	buyer_id	listing_id	offer_amount	location
▶	269383741	265926826	265926826	12381	0	library

6. Google Analytics:



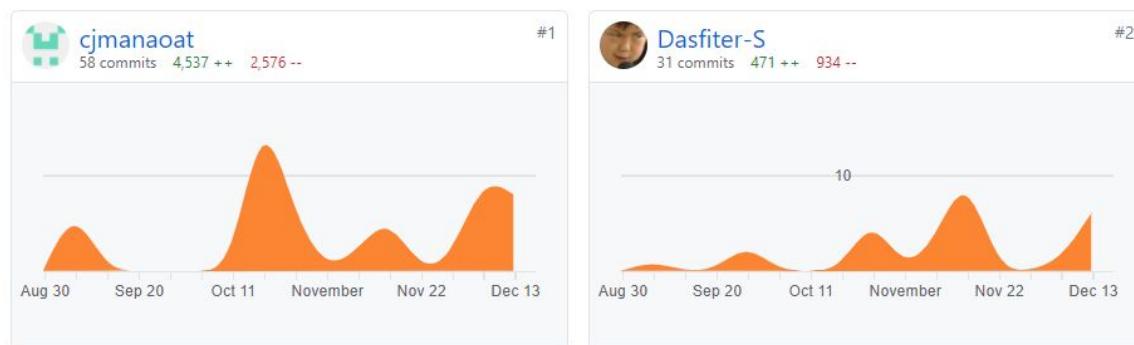
7. Project Management:

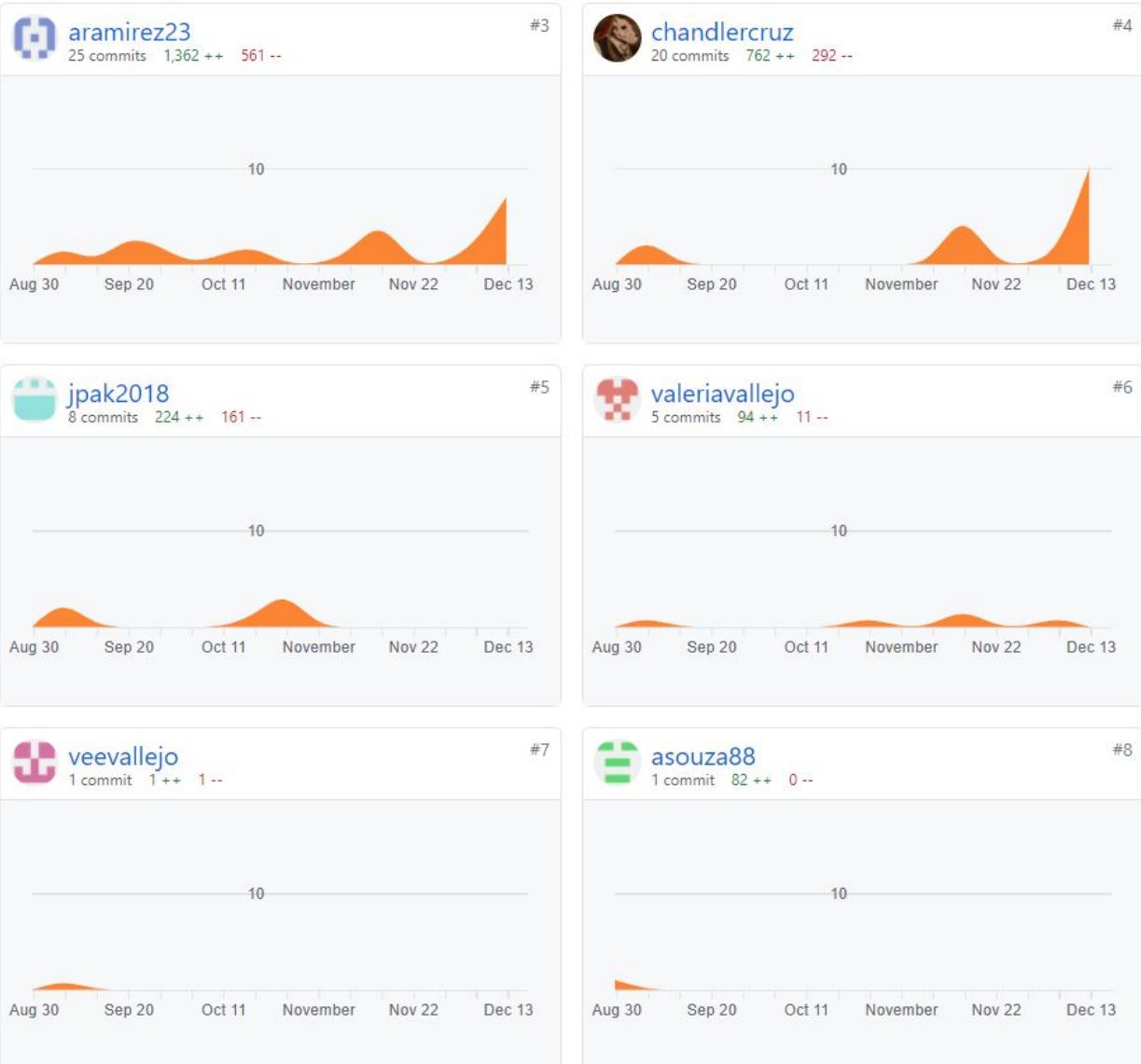


Aug 30, 2020 – Dec 18, 2020

Contributions: Commits ▾

Contributions to master, excluding merge commits





8. Team Member Assessment:

Self Assertion



Ricardo Carretero
Fri 12/18/2020 1:59 AM

To: Alicia Jane Ramirez
Cc: Valeria Vallejo; Chandler Thomas Cruz; Jonathan Pak; Chris Jerome Belino Manaoat



This is a collection of e-mails that each team member sends to ALL of their respective team members (so that all team members can see all the mail on this) outlining: a) His/her contributions to team project and teamwork (technical and any other) in no more than half a page – list item format is OK. b) Number of submissions he/she made to github team Dev. branch c) One brief paragraph on main challenges he/she encountered in team project d) One brief paragraph on what would he/she do better next time

Ricardo aka "Narancia" Carretero

Implemented and maintained the server infrastructure (excluding databases)
Integrated new tech and changes from different installations on the server
Created a service to enable the website to run unsupervised (sudo systemctl start trademark.service)
Suggested compatible technologies for our stack
Maintained the git code base
Created and implemented a git workflow with documentation
Optimized image caching for faster website loads times
Fixed any noticeable unpythonic implementation

git stats: 31 commits 471++ 934--

The stats do not reflect all the work done on the AWS instance preparing and maintaining the server.

Best team I've had to be honest.

My main issue coming into this project was making sure I gave the team enough space to grow as developers. No particular challenge I couldn't solve except for anything too in depth in the frontend. Maybe I can focus more on the frontend next time, but I don't really care for that side of development.

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

Self Assessment



Chris Manaoat Designed and implemented the home page Designed and implemented the navbar Implemented user accounts (account creation...

Fri 12/18/2020 4:09 AM



Valeria Vallejo

Fri 12/18/2020 3:51 AM

To: Ricardo Carretero; Alicia Jane Ramirez
Cc: Chandler Thomas Cruz; Jonathan Pak; Chris Jerome Belino Manaoat

Valeria Vallejo

a. For this project, I used AWS to create an EC2 instance for our server and an RDS instance for our database.

Some of the things I did on the database were:

- Planning out and creating the database tables
- Adding constraints to the tables (for foreign key, deletion...)
- Creating the Admin account with restricted permissions'
- Creating functions to grab data from the database (encryption tools, blob conversion tools)

For the AWS instance, I just created IAM user accounts for the team and added alarms that reboot the instance if anything is going wrong.

b. 5 commits 94++ 11--

c. The things that were challenging for me mostly came about because I had absolutely no idea about servers or database security or where to begin. For example, things like the VPC, security groups, and inbound/outbound rules were hard to figure out when I was first trying to set everything up. Luckily Amazon has a lot of documentation for how to set these things up, so it was very doable once I gathered all of the necessary resources. Another thing that was challenging was trying to plan out the database since it felt like I had to guess what we would need before I fully understood how everything was going to function. It was fine once I stopped trying to overthink it and I eventually altered the tables as I found out what we would need.

d. There are several things I feel like I could have done better, mostly having to do with my understanding of inbound rules and how they should work with people who SSH early on in the project. Other things that I could have been better were mostly rooted in early on planning, such as which tables we needed for the database and encrypting the user's data. Overall, these things weren't too hard to do later, but they would have been better to have there early on so I didn't have to keep manually changing the data in the database to fit the new way of storing data.

Self Assessment



Chris Jerome Belino Manaoat
Fri 12/18/2020 4:09 AM

To: Valeria Vallejo; Ricardo Carretero; Alicia Jane Ramirez
Cc: Chandler Thomas Cruz; Jonathan Pak



Chris Manaoat

Designed and implemented the home page
Designed and implemented the navbar
Implemented user accounts (account creation, logging in, logging out)
Provided basic skeleton for incorporating flask (templates included)
Put together search function(not the page results itself, but the functionality and queries)
Assisted with connecting front and back end
Sent good vibes – “Form”

git stats: 58 commits 4,537++ 2,576--

One challenge I encountered was trying to balance time to work on this project in parallel to other work from outside classes. Another problem encountered was getting the navbar to look a little more presentable when scaling down to different window sizes. Although I did take a class on Databases, it took me a while to remember how to implement this with our website since I wasn't too familiar with Python. I also had trouble with into the flow of using Github in a team environment.

If there was something I could improve on, it would be doing more tests on the stuff I implemented. Sometimes the fixes ended up being something small. From this class I learned about how much communication actually plays a role during the SE product process. I don't talk an awful lot, I'd rather type out what I did, but I feel like talking more would be nicer. I could also improve on sticking with a consistent code format. Overall, the team was very helpful, and I am glad I got to work on this project with them.

Sent from [Mail](#) for Windows 10

...

Self Assessment



Alicia Jane Ramirez
Fri 12/18/2020 5:08 AM

To: Chris Jerome Belino Manaoat; Valeria Vallejo; Ricardo Carretero
Cc: Chandler Thomas Cruz; Jonathan Pak



Hello Team 07,

A - My contributions

- Organized all team zoom meetings
- Kept the rest of the team on task with checkpoints and frequent communication
- Was lead document editor, doing majority of document revising
- Completely designed and implemented the user dashboard
- Fixed formatting issues for Listing Details, Create Listing, Search Results, and Register to better fit screens on desktop and mobile
- Helped discover and fix several bugs arising from interaction of front and back end for Search Results, Register, Create Listing, and Contact

B - GitHub Commits:

25 commits 1,362++ 561--

C - Main Challenges I Encountered:

One of the main early challenges I had was figuring out how to both lead a team and how to work on software in a group. I may have technically led some school group projects, but never with the formal role of "team leader". It was a learning experience to figure out how to manage a group to accomplish one big goal. Also, while I had some familiarity with the tools used, I did need help to catch up to speed. It was also a little difficult to foresee future problems due to lack of experience on this kind of software development, but I did see myself eventually able to prepare for some scenarios.

D - What I Would Do Better Next Time:

One thing I would do is have more code review sessions with a more strictly enforced coding style. The lack of this led to some readability issues in code that took more work to fix later than if we enforced them more strictly earlier. Another thing I would do better on is having a better idea of the bigger picture earlier on. When more in the planning stage I still hadn't grasped how every aspect of the site would come together, and as I went along it became clearer and I'd want to take that experience with me into future projects.

Best,

Team 7 Team Lead, Alicia Ramirez

Self Assessment

 Jonathan Pak
Fri 12/18/2020 8:57 AM
To: Alicia Jane Ramirez; Chris Jerome Belino Manaoat; Valeria Vallejo; Ricardo Carretero
Cc: Chandler Thomas Cruz



Jonathan Pak

Designed and implemented Search Result Page
Implemented Listing/Item Page
Implemented and assisted in filtering functionality
Assisted in connecting/pathways for HTML pages
Assisted in base index and templates

8 commits 224 ++ 161 --

One of the main challenges I encountered was trying to format pages based on screen size. Whenever I tried to fix a formatting issue for a page, sometimes some functions would break, and I would have to fix that as well. Image scaling was also difficult since there were many methods so I had to constantly test to see what would work the best. It was also confusing to learn and remember the git workflow. This was important to maintain the master branch as well as the developer branches and not recklessly push everything into master.

One thing I could have done better is do more tests with the whole group. It was much easier spotting mistakes with your whole group rather than looking for them yourself. I would also like to have communicated with the team more. Since we were assigned to work on different parts, sometimes I got too focused on my own part and forgot to see the bigger picture. So, when we met, there could be pathway issues because I didn't properly communicate the changes I made to the others. Overall, I learned a lot with working with my teammates and it was a great experience working with them.

...

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

Self Assertion for submission

 Chandler Thomas Cruz
Fri 12/18/2020 8:57 AM
To: Ricardo Carretero; Alicia Jane Ramirez
Cc: Valeria Vallejo; Jonathan Pak; Chris Jerome Belino Manaoat



Helped brainstorm ideas for website features
Helped write out priority features
Took notes during Milestone 3 meeting
Wrote HTML for Create Listing and Contact pages on the website
Implemented functionality for Create Listing and Contact pages

GitHub stats: 20 commits 762++ 292--

I had a blast working with my team, everyone was super friendly and knowledgeable on all the relevant materials for the project, so we never got too stuck on anything due to helping each other. And due to the friendly atmosphere of the group, I never felt uncomfortable with asking people for help.

My biggest challenge was correctly implementing the lazy registration to the pages. It took me a bit to figure out how to handle my Contact and Create Listing pages sending different data depending on if the user was logged in already or not, but I eventually managed to figure it out thanks to help from my teammates.

Next time, I'd put more emphasis on testing for bugs and errors. While I did test my code for basic functionality, I could've done a lot more specific bug testing earlier on, which would've saved both myself and the rest of the team a lot of work. Instead, a lot of bugs were caught late into the work process and it resulted in a prolonged group testing session, one which could've easily been avoided or shortened if I had put more emphasis on testing earlier on.

[Get Outlook for iOS](#)
...

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