

# Final Project for SW Engineering Class CSC 648-848 Spring 2022

GatorTrade

Team05

21 May 2022

URL: [GatorTrade Link](#)

Members:

**Team Lead:** Kishan Patel - kpatel0@sfsu.edu

**Front End Lead:** Michael Cheung

**Front End Member:** Joanne Wong

**Back End Lead:** Faisal Zaheer

**Back End Member:** Juan Hernandez

**Github Master:** Ze Lei

|  |           |
|--|-----------|
| <b>Product Summary:</b>                              | <b>3</b>  |
| <b>Milestone documents - M1-M4</b>                   | <b>4</b>  |
| Milestone 1  | 4         |
| Milestone 2  | 17        |
| Milestone 3  | 40        |
| Milestone 4  | 44        |
| <b>Product Screenshots</b>                           | <b>56</b> |
| <b>Database Organization</b>                         | <b>65</b> |
| <b>Github Organization</b>                           | <b>67</b> |
| <b>Google Analytics Stats Plot for WWW Site</b>      | <b>68</b> |
| <b>Project Management</b>                            | <b>69</b> |
| <b>Team Member Self-Assessment and Contributions</b> | <b>70</b> |

## I. Product Summary:

1. Name of Product: GatorTrade
2. Description of Product:

Our application is tailored to the students, faculty, and staff at SFSU.

Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application can search and filter through categories for the goods they are looking for such as technology, books, or crafts. Also, this application is exclusively for the SF State community only. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

3. Itemized list of committed P1 functions:
  1. The application allows users to:
    - a) register, login, and logout.
    - b) search and browse items.
    - c) sort items by their price and title.
  2. Registered users are allowed to:
    - a) post items for sale.
    - b) message other sellers.
    - c) view the posts they've made and messages sent to them.
    - d) sort messages received by item and date.
    - e) change their password.
  3. Admins are required to:
    - a) approve posts made by users or delete an item listing.
  4. Uniqueness:
    1. GatorTrade allows for its users to experience exclusivity since it is limited to only SFSU personnel and students. Therefore, users can find goods and items within the SFSU community easily.
  5. URL to product: [GatorTrade URL Link](#)

II. Milestone documents - M1-M4

a. Milestone 1

SW Engineering CSC648/848 Spring 2022  
GatorTrade  
Team 05

Milestone 1

5 March 2022

Members:

**Team Lead:** Kishan Patel - kpatel0@sfsu.edu

**Front End Lead:** Michael Cheung

**Front End Member:** Joanne Wong

**Back End Lead:** Faisal Zaheer

**Back End Member:** Juan Hernandez

**Github Master:** Ze Lei

History Table:

| Date Submitted | Date Revised |
|----------------|--------------|
| 5 March 2022   | 9 March 2022 |

|  |           |
|--|-----------|
| <b>Executive Summary:</b>                                      | <b>3</b>  |
| <b>Personae and Main Use Cases:</b>                            | <b>4</b>  |
| Personae:  | 4         |
| Main Use Cases:  | 6         |
| <b>List of Data Items and Entities - Glossary/Definitions:</b> | <b>8</b>  |
| <b>List of Functional Requirements:</b>                        | <b>9</b>  |
| Unregistered Users:  | 9         |
| Registered Users:  | 9         |
| Admin:   | 10        |
| <b>List of Non-Functional Requirements:</b>                    | <b>10</b> |
| <b>Competitive Analysis:</b>                                   | <b>11</b> |
| <b>Technologies and Architectures:</b>                         | <b>12</b> |
| <b>Team Members and Roles:</b>                                 | <b>12</b> |
| <b>Checklist:</b>  | <b>12</b> |

## 1. Executive Summary:

Currently, there is a lack of a dedicated platform for the students, faculty, and staff of San Francisco State University (SFSU) to quickly and securely exchange goods. Students often try to find the cheapest and most efficient way to get the items they need for their classes. If a student wanted to sell a used textbook for a particular class, it is hard to find a suitable platform since they would not be certain what type of platform other students would use to find used textbooks for sale. Although there are some undedicated places for SFSU members to sell their belongings like Amazon, SFSU Reddit, and SFSU Discord, they have less traffic and exposure since not every SFSU member would use or know about that platform. Students and employees that have hobbies in arts and crafts can also list their handicrafts on this platform to be sold and exchanged with students and fellow coworkers. Having a dedicated platform can also provide a sense of safety for the users, especially since having a place to have secure transactions can allow for accountability to occur.

Our application will be tailored to the students, faculty, and staff at SFSU. Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application shall be able to search and filter through tags for the goods they are looking for such as technology, textbooks, or crafts. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

Our team is a student start-up composed of students at SFSU. Each member is well-versed in the problems and concerns we address with our application since we have all experienced it firsthand at least once. Therefore, it is safe to say that we know the target audience well because we are in fact part of that same target audience the application strives to appeal to.

## 2. Personae and Main Use Cases:

Personae:

**Jennifer - Full-Time Student**



<https://www.shutterstock.com/image-photo/smart-female-college-student-on-campus-193300418>

| About/Personality   | Goals/Scenario   |
|---|--|
| <ul style="list-style-type: none"> <li>Full-time SFSU freshman with financial aid and student loan.</li> <li>Is majoring in Psychology.</li> <li>Uses smart phone for social media and computer to watch k-dramas.</li> <li>Have basic WWW skills.</li> <li>Is tight on money, so she likes to save up whenever she can.</li> </ul> | <ul style="list-style-type: none"> <li>She recently moved into a new apartment. As she is on a tight budget, she is looking for cheap and used furniture for her new apartment.</li> </ul> |

**Jerry - part-time student, full-time employee, father**



[https://www.moneytalksnews.com/workers/images/width=1460/wp-content/uploads/2019/09/16141108/shutterstock\\_760359751-scaled.jpg?s=b162175d3d89fef31470f07a51a64cdc13ce87bb102491b1c793b22afe0148b](https://www.moneytalksnews.com/workers/images/width=1460/wp-content/uploads/2019/09/16141108/shutterstock_760359751-scaled.jpg?s=b162175d3d89fef31470f07a51a64cdc13ce87bb102491b1c793b22afe0148b)

| About/Personality   | Goals/Scenario  |
|---|---|
| <ul style="list-style-type: none"> <li>Part time student at SF State</li> <li>Works at an ice cream shop</li> <li>Has to care for his young daughter</li> <li>Limited time: Constantly rushing to get things done</li> <li>Proficient WWW app skills</li> <li>Has big hands so struggles navigating WWW apps on mobile</li> <li>With his limited time and big hands he has low patience for WWW apps</li> </ul> | <ul style="list-style-type: none"> <li>He wants to know the required materials for each class and order them while navigating through as few pages as possible.(On his phone)</li> <li>He wants to get his daughter a present but doesn't have the free time to go shopping. He wants to be able to search for gifts and contact a seller to set up a pickup to his workplace.</li> </ul> |

**Robert - SFSU Faculty**

Member, Registered Admin

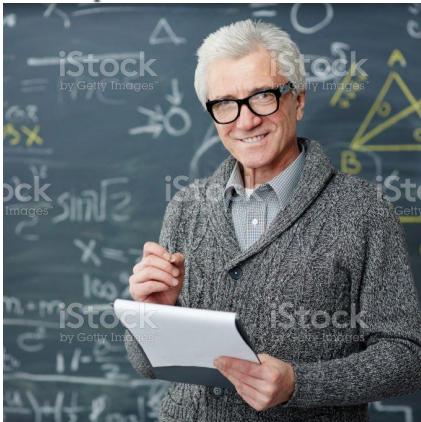


<https://www.scs.gatech.edu/sites/scs.gatech.edu/files/styles/medium/public/images/mercury/scsnew2018.jpg?itok=5jEIXmac>

| About/Personality   | Goals/Scenario  |
|---|---|
| <ul style="list-style-type: none"> <li>Has worked at SFSU as an Office IT worker for ~2 years.</li> <li>Was previously a SFSU student, and is now an alumni.</li> <li>Is knowledgeable and patient when it comes to using and managing WWW apps.</li> <li>Has some free time during his work hours, and plenty after work hours.</li> </ul> | <ul style="list-style-type: none"> <li>He was asked by a student if he could help manage the GatorTrade application part-time as an admin, and with Robert's prior experience in managing WWW applications, he agreed to help.</li> <li>Because of the amount of free time he has, he can manage the GatorTrade app on the side as an admin.</li> <li>If a SFSU student user of GatorTrade has posted an inappropriate item for sale, Robert will remove such items as soon as possible during his free time managing the application.</li> </ul> |

**Franklin - SFSU Staff,**

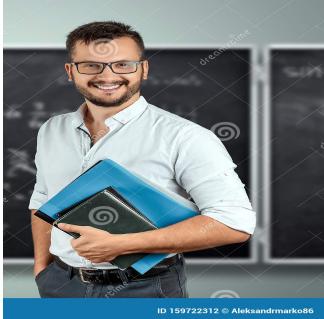
Computer Science Professor



<https://media.istockphoto.com/photos/teacher-with-document-picture-id905256730>

| About/Personality   | Goals/Scenario  |
|---|---|
| <ul style="list-style-type: none"> <li>Familiar with online shopping, has used Amazon before</li> <li>Has a busy schedule, doesn't like to click more than necessary</li> <li>Knows what he wants, isn't looking to browse</li> </ul> | <ul style="list-style-type: none"> <li>Provide resources necessary for his students to succeed</li> <li>Do his tasks quickly and efficiently</li> </ul> |

**Steve** - A graduate student at SFSU



<https://www.dreamstime.com/portrait-young-male-teacher-background-school-blackboard-teacher-s-day-knowledge-da-y-back-to-school-study-image159722312>

| About/Personality  | Goals/Scenario   |
|--|--|
| <ul style="list-style-type: none"> <li>• Very busy with work and study at same time</li> <li>• English major, excels at his analytical papers</li> <li>• Poor time management skills</li> <li>• Need help with his Math class</li> <li>• Average WWW skills, prefers mobile app</li> </ul> | <ul style="list-style-type: none"> <li>• Struggling with Math class in the middle of semester</li> <li>• Wants to buy some resources that might be helpful with the class</li> </ul> |

### Main Use Cases:

#### A. Jennifer - Buying a used textbook from GatorTrade

Jennifer is a **SFSU student**. She finds out about GatorTrade from an email sent by a **SFSU administrator**. She goes to the web site and searches for her courses' textbooks by typing in their name. She sorts the list of used textbooks from the search result by lowest to highest price. She doesn't care about editions because her course allows her to use older editions. After clicking on the **item** that she wants to buy, she gets a description of the **item**, the seller, the price, and an image of the **item**. She wants to buy the **item**, so when she tries to send a one-way message to the seller, she is prompted to login or register an account. Since she is a SFSU student, she can register with her SFSU email and become a **registered user**. After registering, she is now allowed to send a one-way message to the seller, who is also a **registered user**. She sends her phone-number and shortly, she is contacted by the seller. They agreed to meet at the library to conduct the transaction. After the **transaction**, the seller goes to their post and removes it because they have sold the **item**.

#### B. Jerry - Getting Materials for class and personal shopping using GatorTrade

While Jerry is at work, he realizes that the semester is about to begin. He pulls out his phone and navigates to GatorTrade. He quickly locates the large **login button** and enters his **username** and **password**. He is redirected to the **homepage** and searches for a clicker and textbooks. Someone happens to be selling each **item** so he contacts each

**seller** who all agree to meet on campus during his break. He then remembers that his daughter's birthday is soon but is too busy to go shopping. She's been getting into physical conditioning recently so she searches for weights under the Fitness **category**. He eventually finds some dumbbells and contacts the seller who agrees to swing by his workplace tomorrow since Jerry wont be able to make it to the campus tomorrow.

#### C. **Robert** - Managing the GatorTrade application.

Robert is a **Faculty member**. He was approached by a SFSU student to help manage the GatorTrade application as an admin. Using an **admin account** given to Robert, he can manage all **buy, sell, and exchange** postings on the application. One of his major responsibilities include authorizing all posts before they are live on the website. He can browse through the posts on the **home page** or on any specific pages. From there, as an admin, Robert can **delete** the posting should it contain any inappropriate information or content.

#### D. **Franklin** - Listing class materials

Franklin is an **instructor** for computer science classes. He is also responsible for hiring student assistants for various courses. He wants to add a **listing** on the website. After he **verifies** his status as an instructor, he is able to make a listing under the “jobs” **section**. Franklin fills out a **form** that describes the necessary information about the job. After approval by the site admin, student users are able to **view** the job listing.

#### E. **Steve** - Exchanging notes on GatorTrade

Steve is a **graduate student** with an English major at SF State University. He also has a part time job on campus and often struggles managing work and study together. In the middle of the semester, he feels like he needs help with his Math class. While he was working, he only had a phone with him, so he used his phone to navigate through our website. On the website, he found a **post** from another student named Bucky for exchanging notes. He also found notes for his Math class in the list of notes that Bucky wanted to give. So Steve **texted** Bucky using GatorTrade’s **chat feature** and **traded** his English notes for Bucky’s Math notes.

### 3. List of Data Items and Entities - Glossary/Definitions:

1. Key Users:
  - i. **Users:** People who are using the application.
    1. **Unregistered\_users:** They can view and search items, cannot send one-way messages or post items for sale, and do not need to login/register.
    2. **Registered\_users:** They can register if they have a valid SFSU email. They are allowed to view, search, post items, review other users, and send one-way messages. Will have a profile picture, username, password, notifications. They are required to login/register.
    3. **Admin:** They have all the privileges and can access and modify permissions for all users. They also check items before other users are allowed to view them. They are allowed to delete any posts or reviews that violate the policies/terms/conditions.
  2. Data items:
    - i. **Items:** These are products listed by users to be sold or exchanged. Each item has a price and/or other items they are willing to exchange for, the user who posted it, the date it was posted, a brief description of the item, photo(s) of the item, and related tags.
    - ii. **Reviews:** These are posts made by users of a particular seller. It includes the user who posted it, the user that is being reviewed, the item related to the review, and the date it was posted.
    - iii. **Transactions:** It is the exchange of goods between users that includes the date it happened, the two users associated with the exchange, status of the exchange, and the item involved.
    - iv. **Category:** A label to identify related to items, including but not limited to books, tech, supplies, notes.
      1. **Electronics:** a specific category that describes technology related items

2. **Text:** a specific category that describes text related items such as textbooks, notes, books not related to courses [novels, non-fiction, etc].
3. **Handicraft:** a specific category that describes items that are handcrafted.
4. **Edu\_supply:** a specific category that describes supplies such as pens, pencils, notebooks, binders, folders, etc.
5. **Food:** Cooking service for lunch and dinner.

#### **4. List of Functional Requirements:**

1. Unregistered Users:
  - 1.1. Unregistered users shall be able to register a new account with a valid SFSU email address.
  - 1.2. Unregistered users shall be able to access the homepage of the website.
  - 1.3. Unregistered users shall be able to browse items on the website.
  - 1.4. Unregistered users shall be able to view details of an item including price, image, item details, etc.
  - 1.5. Unregistered users shall be able to search for items on the website.
  - 1.6. Unregistered users shall be able to sort and filter search results with specific tags and labels.
  - 1.7. Unregistered users shall be prompted to create an account (lazy registration).
2. Registered Users:
  - 2.1. Registered users inherit all functional requirements from Unregistered users.
  - 2.2. Registered users shall be able to login to their account.
  - 2.3. Registered users shall be able to edit their user profile picture.
  - 2.4. Registered users shall be able to change their password.
  - 2.5. Registered users shall be able to post items to sell on the website.
  - 2.6. Registered users shall be able to exchange items with other registered users.

- 2.7. Registered users shall be able to delete items posted by them only.
  - 2.8. Registered users shall be able to send messages to other registered users.
  - 2.9. Registered users shall be able write reviews for items from other registered users only.
  - 2.10. Registered users shall be able to logout from their account.
3. Admin:
    - 3.1. Admin inherits all functional requirements from Registered users.
    - 3.2. Admin shall be required to authorize items posted by registered users.
    - 3.3. Admin shall be able to delete any item from the website.
    - 3.4. Admin shall be able to delete user accounts permanently from the website.
    - 3.5. Admin shall be able to add or delete categories of items from the website.

## 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.
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.
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 email 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, Spring 2022. For Demonstration Only*" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

## 6. Competitive Analysis:

| Key Features               | GatorTrade   | Competitor A:<br>(SFSU Bookstore) | Competitor B:<br>(Etsy) | Competitor C:<br>(Craigslist) | Competitor D:<br>(Ebay) |
|----------------------------|--------------|-----------------------------------|-------------------------|-------------------------------|-------------------------|
| Sell a variety of products | +            | +                                 | +                       | ++                            | ++                      |
| Job Listings               | +            | -                                 | -                       | +                             | +                       |
| Reviews for items/sellers  | +            | -                                 | +                       | -                             | +                       |
| Communication              | +            | -                                 | ++                      | +                             | +                       |
| Browse for items           | +            | +                                 | +                       | +                             | +                       |
| Exclusivity                | ++           | -                                 | -                       | -                             | -                       |
| Feature exists: +          | Superior: ++ |                                   |                         | Does not exist: -             |                         |

Unlike the current market, our product will offer a user experience that will be easier for students to navigate. Students will easily be able to locate all the required or recommended materials for their classes. In addition, students are able to read reviews left by other users. All of this information will help students make a decision on whether they should make a purchase. In addition, our product will offer services exclusively for students such as discounted prices. Students may find offerings intended for students, such as discounted items or job listings.

## **7. Technologies and Architectures:**

1. Server Host: AWS 1 vCPU 1 GB RAM
2. Operating System: Ubuntu 20.04 Server
3. Server Database: MySQL v8.0
4. Web Server: Apache v2.4
5. Server-Side Language: Javascript (Node v17.0)
6. Additional Technologies:
  - a. Web Framework: Express, Bootstrap
  - b. IDE: Visual Studio Code
  - c. Web Analytics: Google Analytics
  - d. SSL Cert: Lets Encrypt (Cert Bot)

## **8. Team Members and Roles:**

1. **Team Lead:** Kishan Patel
2. **Front End Lead:** Michael Cheung
3. **Front End Member/Document Master:** Joanne Wong
4. **Back End Lead:** Faisal Zaheer
5. **Back End Member:** Juan Hernandez
6. **Back End Member/Github Master:** Ze Lei

## **9. Checklist:**

1. So far all team members are engaged and attending ZOOM sessions when required. **DONE**

1. Team found a time slot to meet outside of the class. **DONE**
2. Back end, Front end leads and Github master chosen. **DONE**
3. Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing. **DONE**
4. Team lead ensured that all team members read the final M1 and agree/understand it before submission. **DONE**
5. Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.). **DONE**

b. Milestone 2

SW Engineering CSC648/848 Spring 2022

GatorTrade

Team05

Milestone 2

20 March 2022

Members:

**Team Lead:** Kishan Patel - kpatel0@sfsu.edu

**Front End Lead:** Michael Cheung

**Front End Member:** Joanne Wong

**Back End Lead:** Faisal Zaheer

**Back End Member:** Juan Hernandez

**Github Master:** Ze Lei

History Table:

| Date Submitted | Date Revised  |
|----------------|---------------|
| 20 March 2022  | 23 March 2022 |

|  |           |
|--|-----------|
| <b>Executive Summary:</b>  | <b>3</b>  |
| <b>List of Data Items and Entities - Expanded As Necessary</b>   | <b>4</b>  |
| <b>Functional Requirements - Prioritized</b>                     | <b>6</b>  |
| Priority 1:  | 6         |
| Priority 2:  | 6         |
| Priority 3:  | 7         |
| <b>UI Storyboards - Low-Fidelity and B/W Diagrams</b>            | <b>8</b>  |
| Use Case: Buying a textbook from GatorTrade                      | 8         |
| Use Case: Reviewing the seller after a transaction               | 11        |
| Use Case: Steve receives a message and completes a transaction   | 15        |
| Use Case: Managing the GatorTrade application                    | 18        |
| Use Case: Listing a job opening                                  | 20        |
| <b>High-Level Architecture and Database Organization Summary</b> | <b>23</b> |
| <b>Key Risks</b>   | <b>24</b> |
| <b>Project Management</b>  | <b>25</b> |

## **1. Executive Summary:**

Currently, there is a lack of a dedicated platform for the students, faculty, and staff of San Francisco State University (SFSU) to quickly and securely exchange goods. Students often try to find the cheapest and most efficient way to get the items they need for their classes or for their stay at SFSU. If a student wanted to sell a used textbook for a particular class, it is hard to find a suitable platform since they would not be certain what type of platform other students would use to find used textbooks for sale. Although there are some undedicated places for SFSU members to sell their belongings like Amazon, SFSU Reddit, and SFSU Discord, they have less traffic and exposure since not every SFSU member would use or know about that platform. Students and employees that have hobbies in arts and crafts can also list their handicrafts on this platform to be sold and exchanged with students and fellow coworkers. Having a dedicated platform can also provide a sense of safety for the users, especially since having a place to have secure transactions can allow for accountability to occur.

Our application will be tailored to the students, faculty, and staff at SFSU. Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application shall be able to search and filter through tags for the goods they are looking for such as technology, textbooks, or crafts. Also, since this application is exclusively for the SF State community, users will get some unique features like being able to search for class materials by class name. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

Our team is a student start-up composed of students at SFSU. Each member is well-versed in the problems and concerns we address with our application since we have all experienced it firsthand at least once. Therefore, it is safe to say that we know the target audience well because we are in fact part of that same target audience the application strives to appeal to.

## 2. List of Data Items and Entities - Expanded As Necessary

- a. Users: stores user information such as user password, email, and name.
  - 1. idUsers: primary id used in database identification of the user
  - 2. password: user-created key used for login
  - 3. email: user's SFSU email account; used for login
  - 4. firstname: user's first name
  - 5. lastname: user's last name
- b. Items: stores item information such as title, price, and description
  - 1. idItems: primary id used in database identification of the item
  - 2. category: used to reference the Categories table
  - 3. seller: id that is linked to the user's id
  - 4. photopath: path of the photo of the item
  - 5. description: description of the item
  - 6. title: title of the item
  - 7. price: price set for the item by seller
  - 8. approved: boolean to check if the post has been approved by admin or not
- c. Reviews: stores review information such as reviewee, rating, and reviewer
  - 1. idReviews: primary id used in database identification for review
  - 2. reviewer: used the reference the Users table for the person making the review
  - 3. reviewee: used to reference the Users table for the person being reviewed
  - 4. rating: value given to the reviewer gave the reviewee
  - 5. header: title of the review
  - 6. body: context of the review
- d. Messages: stores information of the messages being sent [one way only]

1. `idMessages`: primary id used in database identification of messages
  2. `item`: used to reference the `Items` table for the product the message is about
  3. `sender`: used to reference the `Users` table for the person sending the message
  4. `receiver`: used to reference the `Users` table for the person receiving the message
  5. `body`: content of the message
- e. Categories: It will contain the following fields for title: Electronic, Text, Handicraft, Educational\_supply, and Food.
1. `idCategories`: primary id used in database identification of categories
  2. `title`: name of the category

### **3. Functional Requirements - Prioritized**

#### **Priority 1:**

##### **1. Unregistered Users:**

- 1.1** Unregistered users shall be able to register a new account with a valid SFSU email address.
- 1.2** Unregistered users shall be able to access the homepage of the website.
- 1.3** Unregistered users shall be able to browse items on the website.
- 1.4** Unregistered users shall be able to view details of an item, including price, image, item details, etc.
- 1.5** Unregistered users shall be able to search for items on the website.
- 1.6** Unregistered users shall be able to create an account (lazy registration).
- 1.7** Unregistered users shall be able to sort search results by price and title.

##### **2. Registered Users:**

- 2.1** Registered users inherit all functional requirements from Unregistered users.
- 2.2** Registered users shall be able to login
- 2.3** Registered users shall be able to change their password.
- 2.4** Registered users shall be able to post items to sell on the website.
- 2.5** Registered users shall be able to send messages to other registered users.  
[one way only]
- 2.6** Registered users shall be able to logout from their account.

##### **3. Admin:**

- 3.1** Admin inherits all functional requirements from Registered users.
- 3.2** Admin shall be required to authorize items posted by registered users.
- 3.3** Admins shall be able to delete any item from the website.

#### **Priority 2:**

##### **2. Registered Users:**

**2.7** Registered users shall be able to exchange items with other registered users.

**2.8** Registered users shall be able to delete items posted by them only.

**2.9:** Registered users shall be able to write reviews for other registered users only.

3. Admin:

**3.4** Admin shall be able to delete user accounts permanently from the website.

**Priority 3:**

1. Unregistered Users:

**1.7** Unregistered users shall be able to filter search results with specific tags and labels.

2. Registered Users:

**2.10** Registered users shall be able to edit their user profile picture.

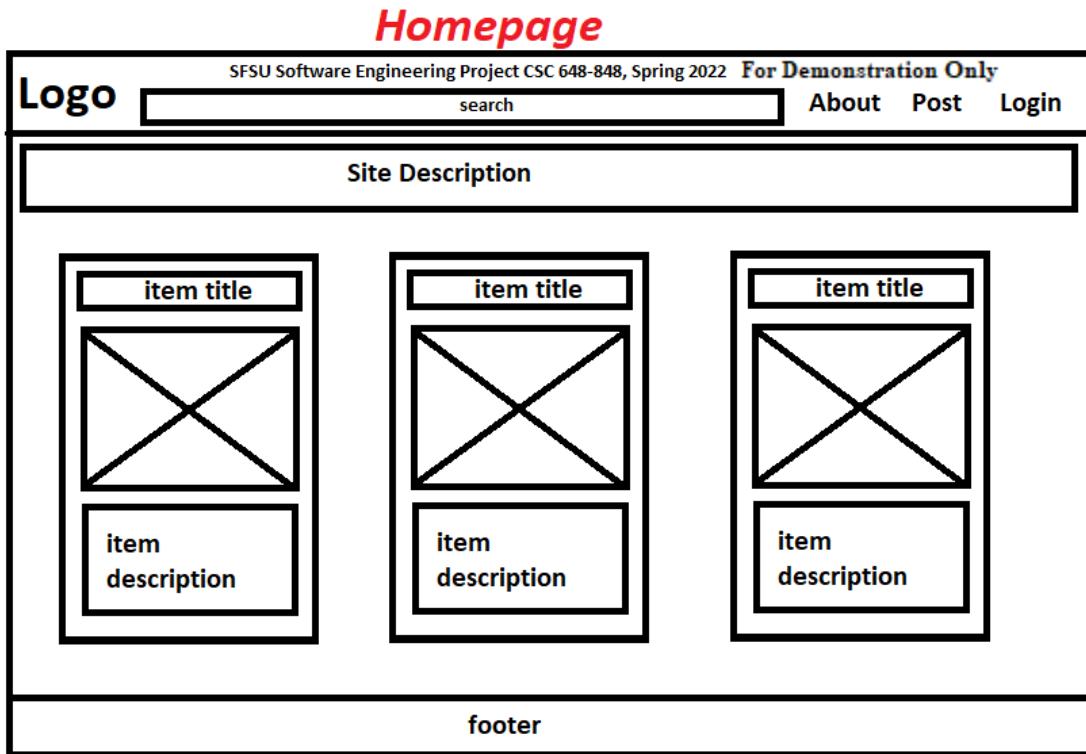
3. Admin:

**3.5** Admin shall be able to add or delete categories of items from the website.

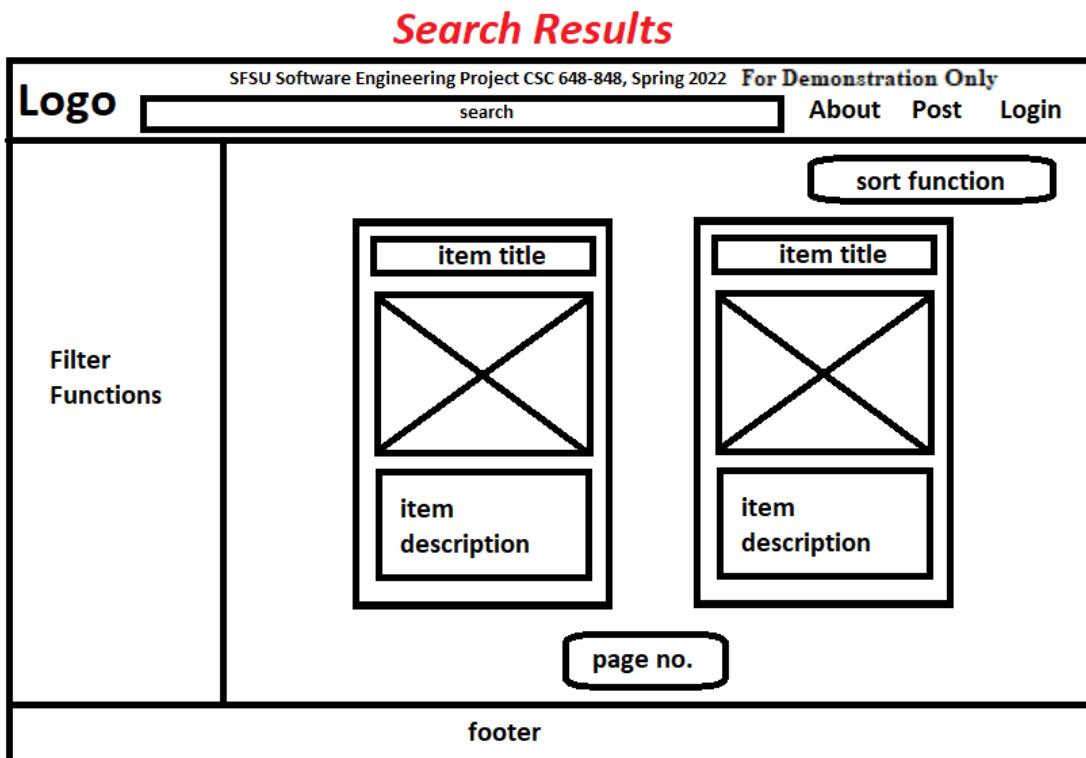
## 4. UI Storyboards - Low-Fidelity and B/W Diagrams

### Use Case: Buying a textbook from GatorTrade

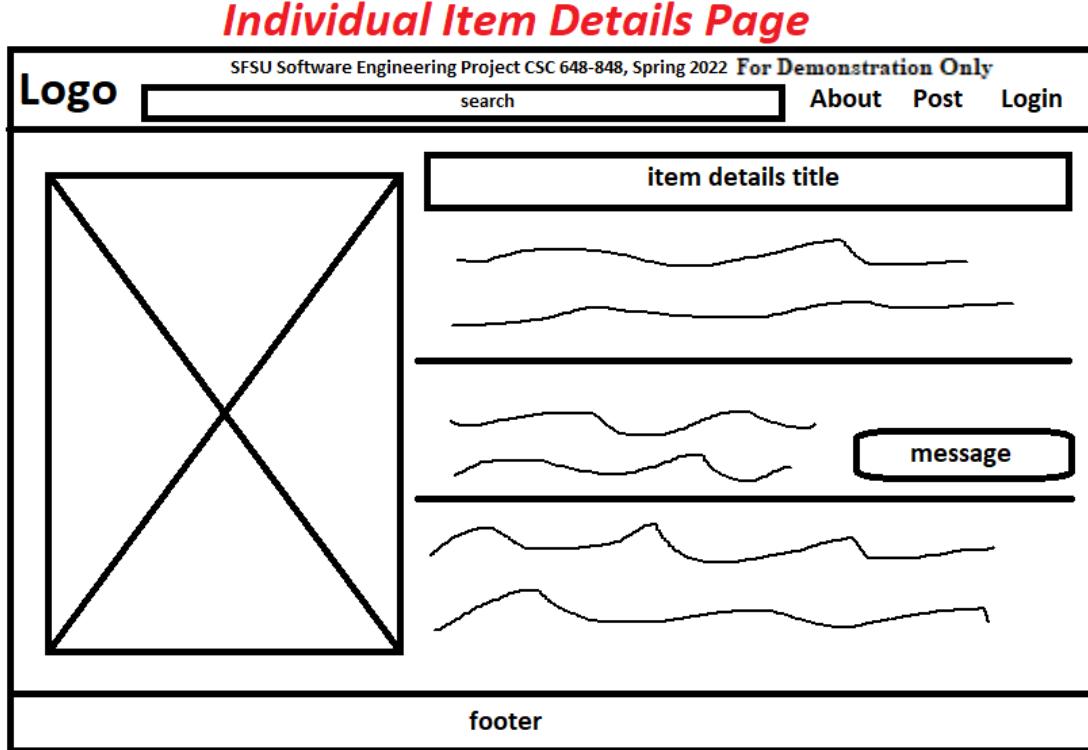
A. Jennifer lands on the homepage and searches for a textbook in the search bar.



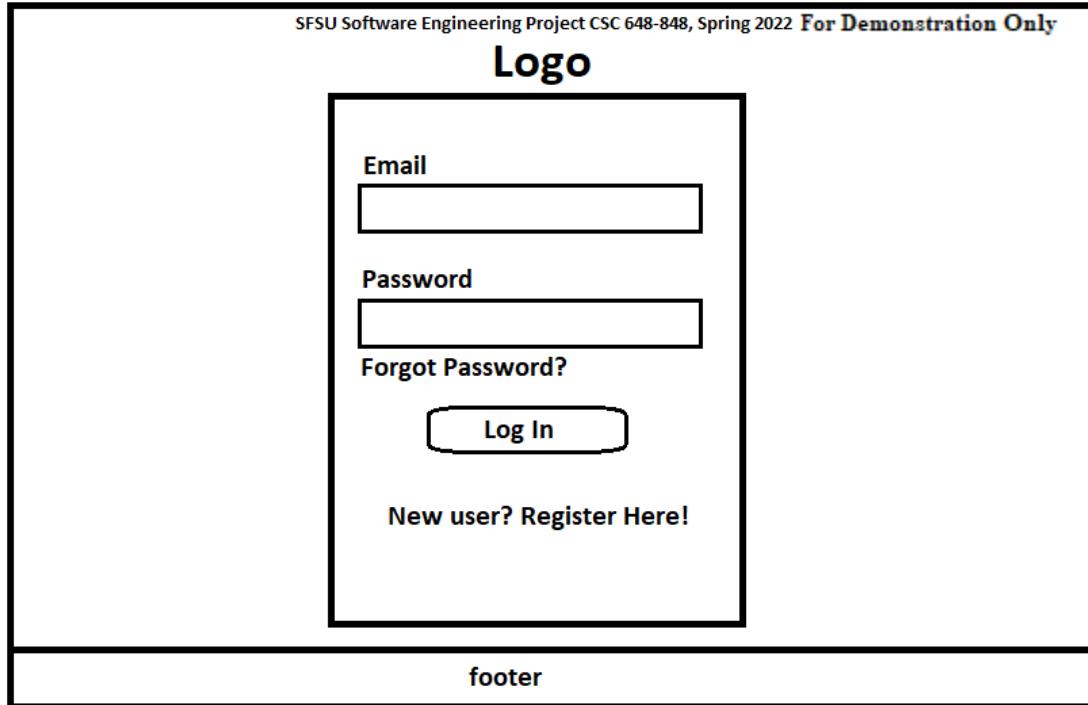
B. Jennifer gets taken to the search results page and clicks on a post



C. Jennifer wants to buy and tries to send a message.



D. Jennifer cannot send a message unless she is logged in.



E. But Jennifer doesn't have an account, so she registers one

### Register Page

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**Logo**

\* First Name

\* Last Name

\* Email

\* Password

\* Re-enter Password

\*  terms and conditions

**Register**

footer

F. After successfully registering for an account, Jennifer gets taken to the message page.

### Send Message Page

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**Logo** search About Post Login

Item Title

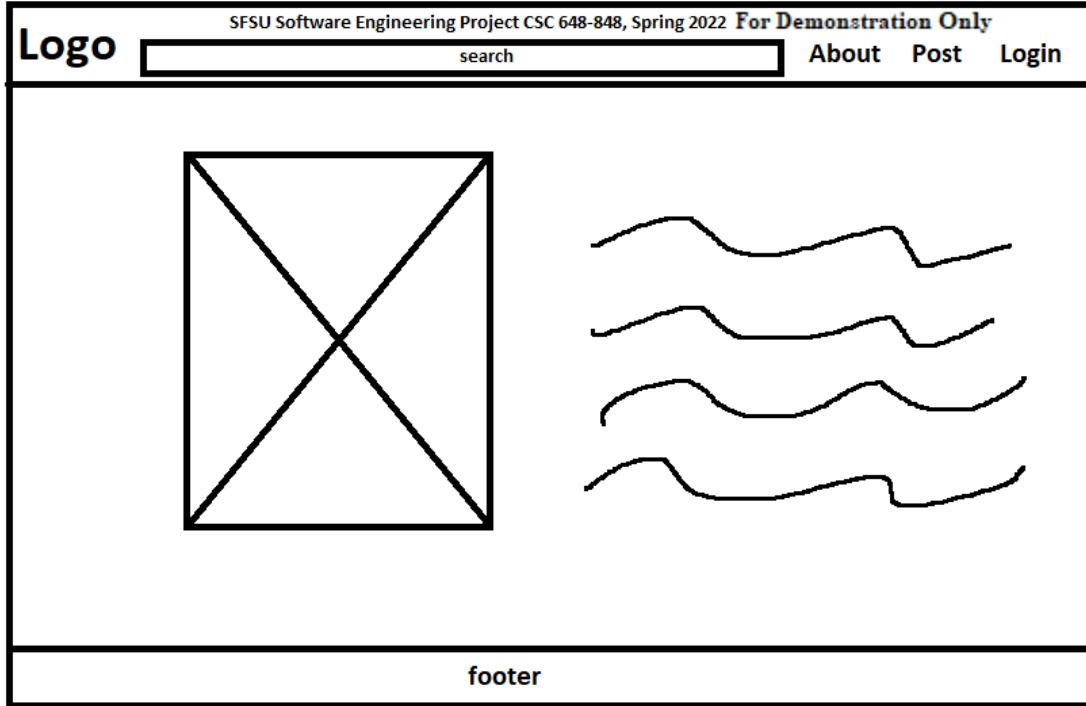
X

**Send**

footer

G. Jennifer gets a confirmation that her message was sent successfully

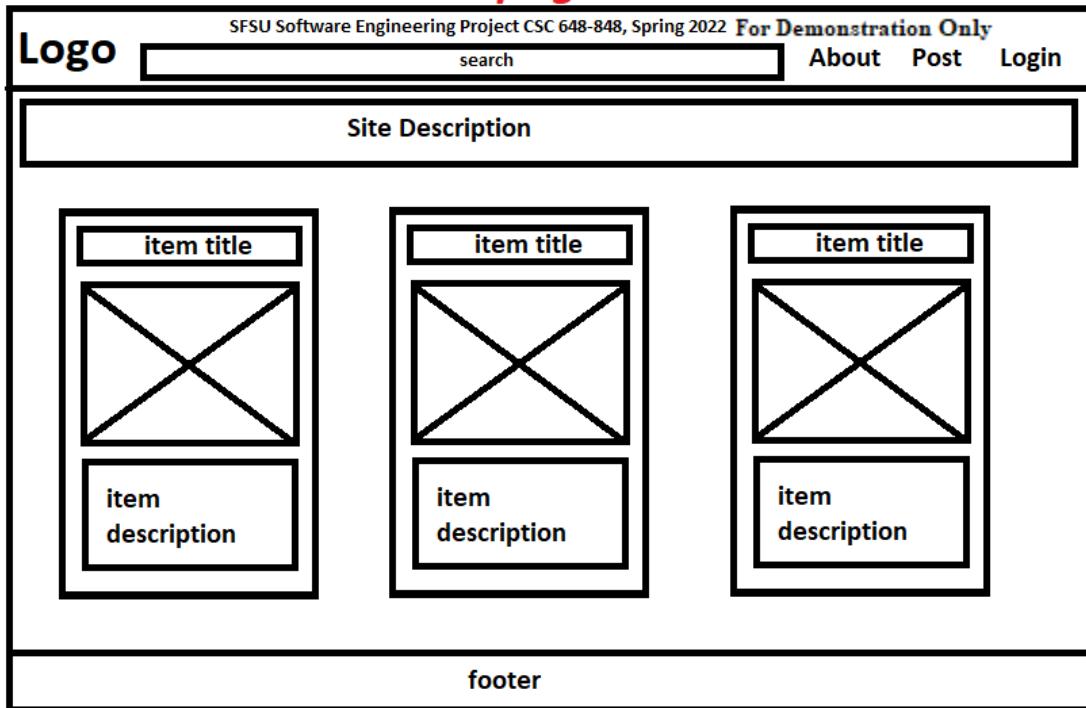
### Confirmation Page



### Use Case: Reviewing the seller after a transaction

A. Jerry just finished a transaction and the seller asked him to do a review.

### Homepage



B. Jerry goes to the login page from the homepage.

**Login Page**

SFSU Software Engineering Project CSC 648-848, Spring 2022 **For Demonstration Only**

**Logo**

Email

Password

[Forgot Password?](#)

**Log In**

[New user? Register Here!](#)

footer

C. After successfully logging in, Jerry gets taken to the homepage.

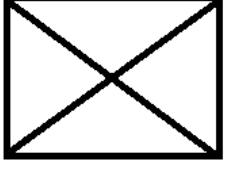
**Homepage**

SFSU Software Engineering Project CSC 648-848, Spring 2022 **For Demonstration Only**

**Logo**  search [About](#) [Post](#) [Login](#)

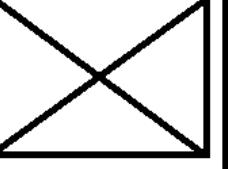
Site Description

item title



item description

item title



item description

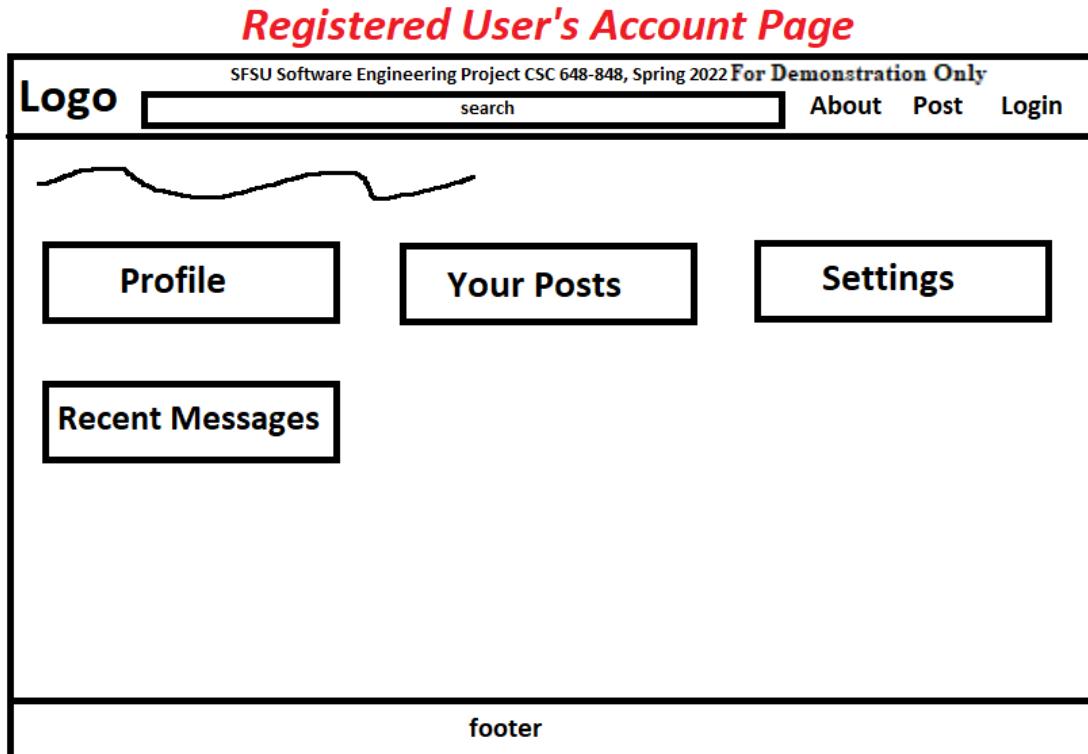
item title



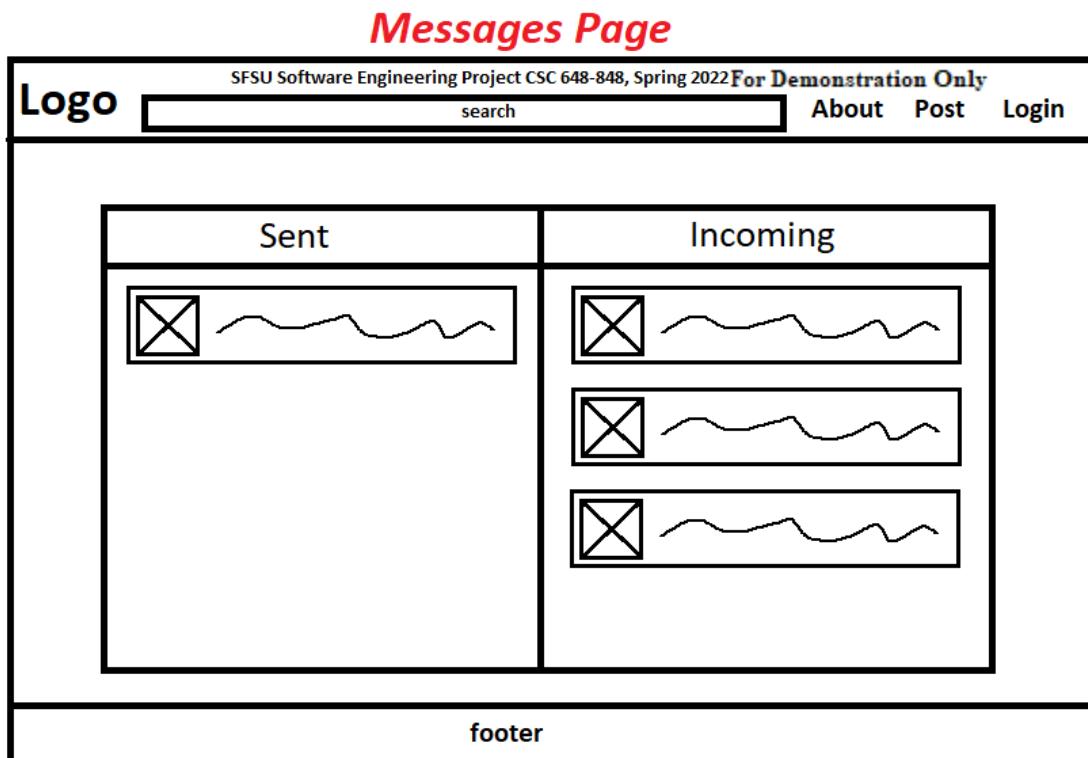
item description

footer

D. Jerry goes to his account to find his recent messages.



E. Jerry sees the message he sent to the seller he just completed a transaction with.



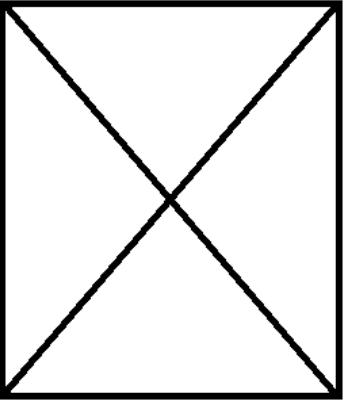
F. Jerry writes his review and clicks submit

### Review Page

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

Logo  About Post Login

Site Description

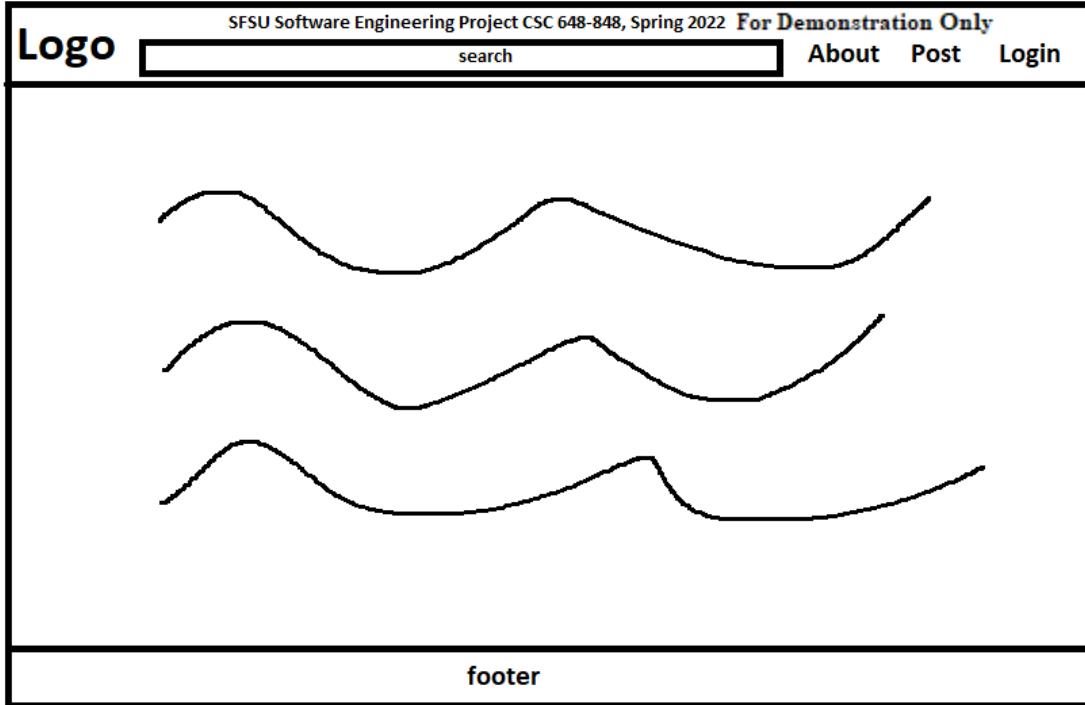




footer

G. Jerry gets a thank you page for submitting a review

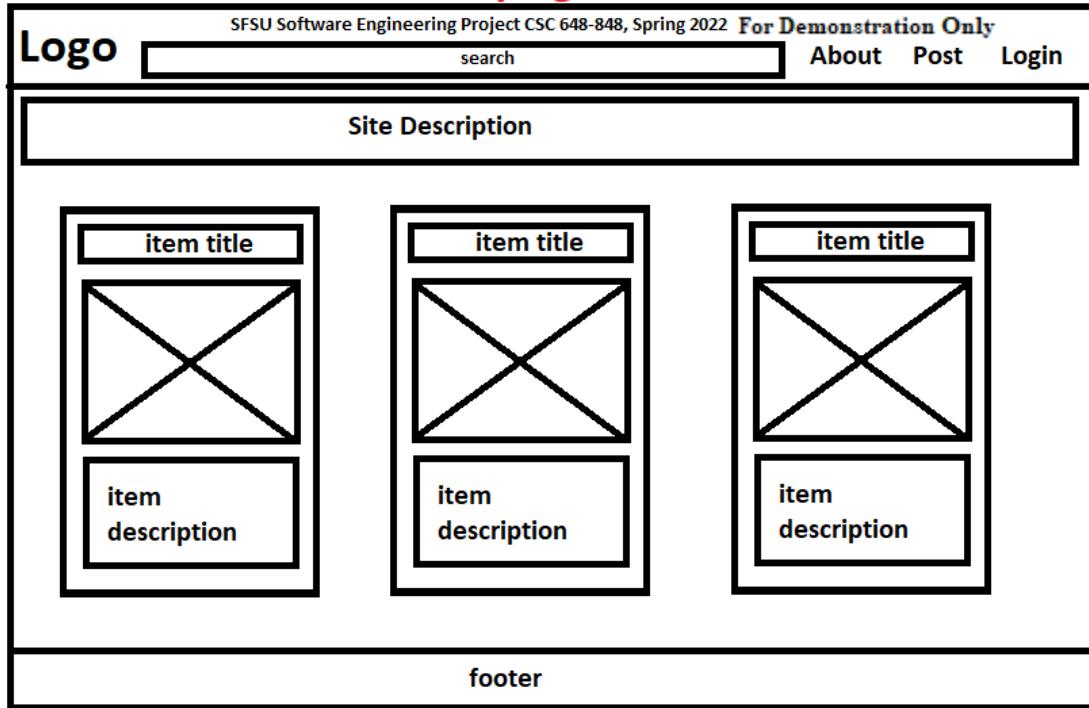
### Thank You Page



### **Use Case: Steve receives a message and completes a transaction**

A. *Steve goes to the homepage to login..*

#### ***Homepage***

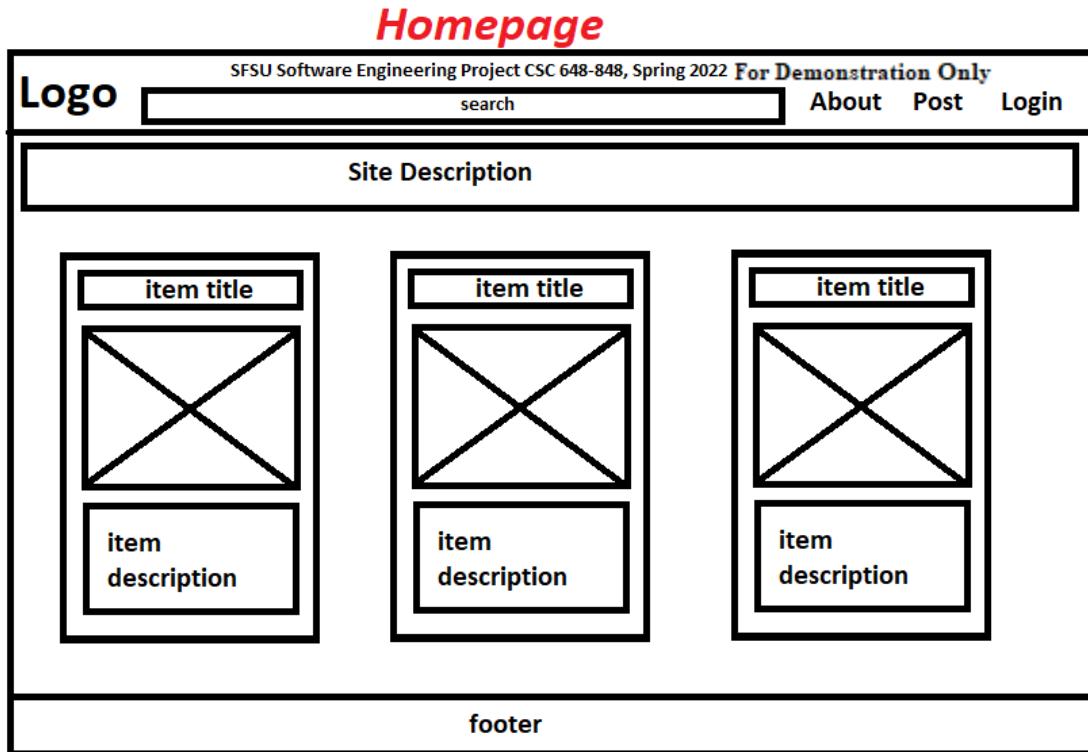


B. *Steve is taken to the homepage after logging in.*

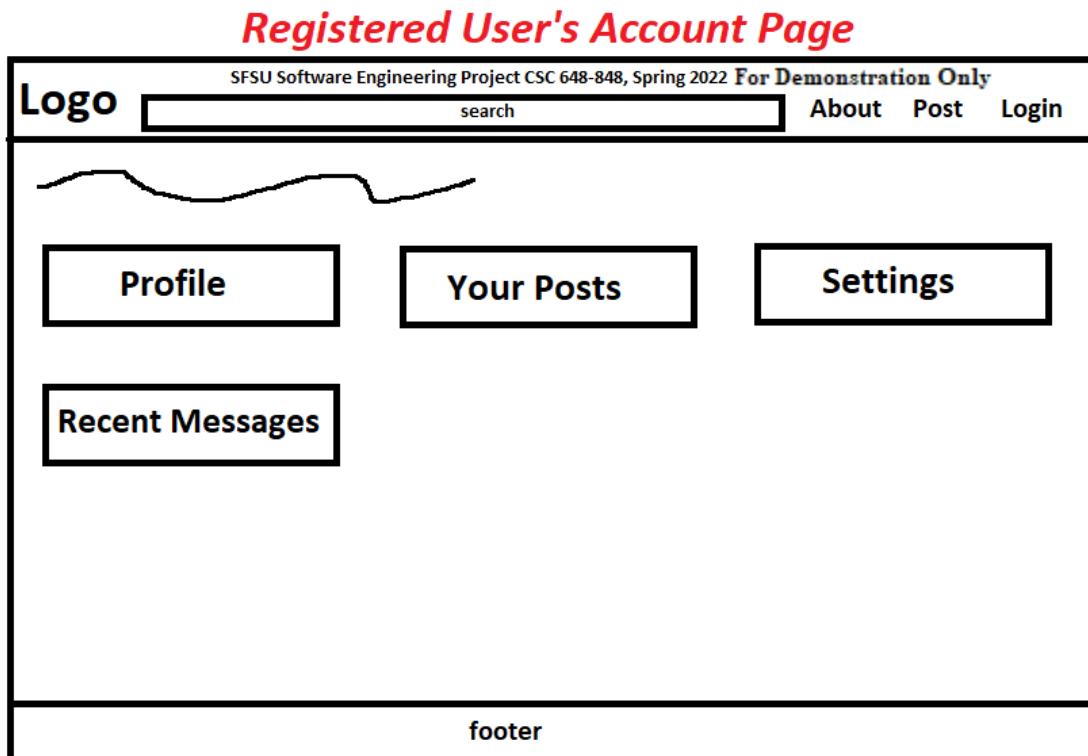
#### ***Login Page***

The diagram shows a wireframe of a login page. At the top is a header bar with a logo, a search bar, and links for About, Post, and Login. The main content area contains fields for "Email" and "Password", a "Forgot Password?" link, and a "Log In" button. Below the main content is a link for "New user? Register Here!". At the bottom is a footer section.

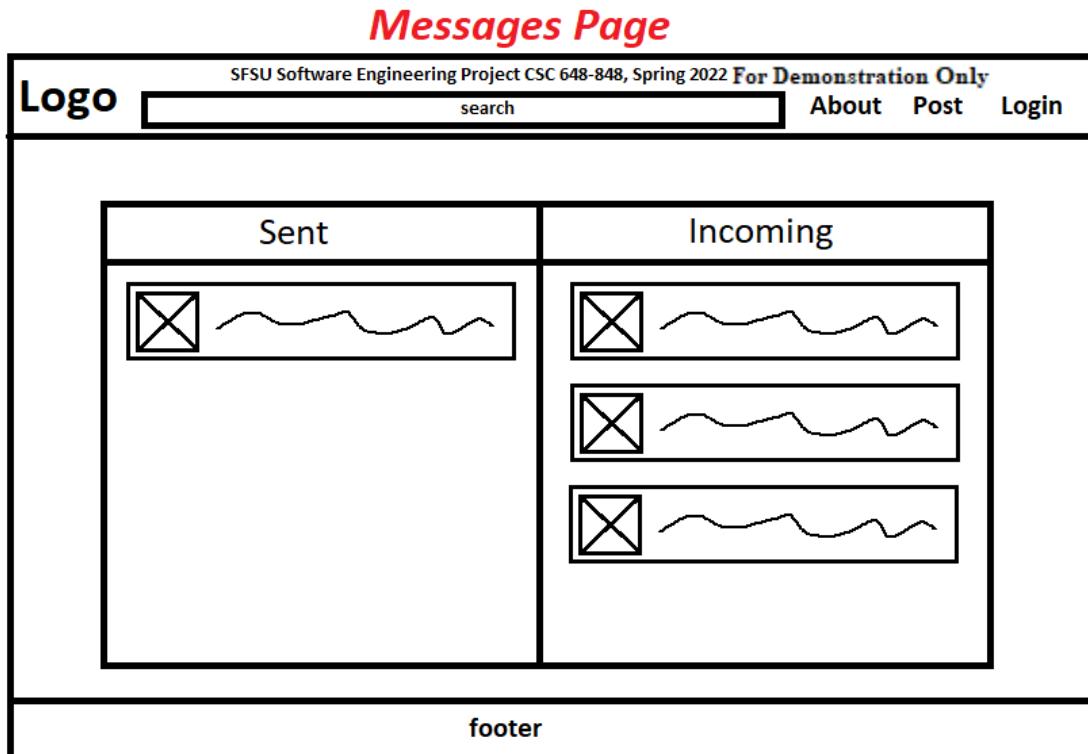
C. Steve goes to his account page from the homepage.



D. Steve is taken to the recent message page after clicking on recent messages



E. Steve checks the most recent incoming message and reaches out to sender

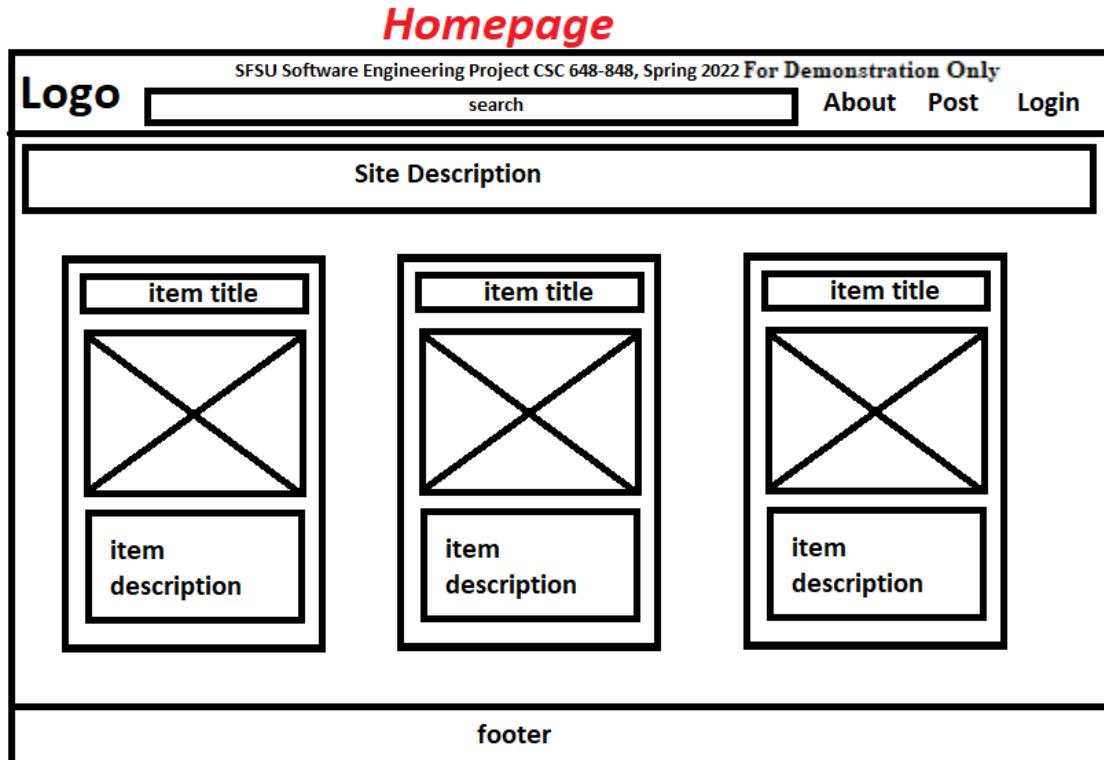


### **Use Case: Managing the GatorTrade application**

Robert is a Faculty member. He was approached by a SFSU student to help manage the GatorTrade application as an admin. Using an admin account given to Robert, he can manage all buy, sell, and exchange postings on the application. One of his major responsibilities include authorizing all posts before they are live on the website. He can browse through the posts on the home page or on any specific pages. From there, as an admin, Robert can delete the posting should it contain any inappropriate information or content. He can perform these actions using Workbench.

### **Use Case: Listing a job opening**

- A. *Franklin, an instructor, goes to the Homepage and tries to make a new post.*



B. Since he is not logged in, he is taken to the login page.

**Login Page**

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**Logo**

Email

Password

[Forgot Password?](#)

**Log In**

New user? [Register Here!](#)

footer

C. Since Franklin is not a registered user, he registers on the register page

**Register Page**

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**Logo**

\* First Name

\* Last Name

\* Email

\* Password

\* Re-enter Password

\*  terms and conditions

**Register**

footer

- D. After registration, Franklin goes to add a new post and fills out all the information required.

**Create Post Page**

SFSU Software Engineering Project CSC 648-848, Spring 2022 **For Demonstration Only**

**Logo** search About Post Login

**Creating new post**

\* Title:  \* Photo(s):

\* Price: \$

\* Tag:

\* Description:

Photo function

footer

- E. Franklin gets an alert after submitting the form that tells him an admin will review his posting first.

**Alert Page**

SFSU Software Engineering Project CSC 648-848, Spring 2022 **For Demonstration Only**

**Logo** search About Post Login

alert message

footer

## 5. High-Level Architecture and Database Organization Summary

### A. Database Organization:

- a. Users Table:
  - i. idUsers: INT (Primary Key)
  - ii. email: VARCHAR(45)
  - iii. password: VARCHAR(45)
  - iv. firstname: VARCHAR(45)
  - v. lastname: VARCHAR(45)
  
- b. Items Table:
  - i. idItems: INT (Primary Key)
  - ii. category: INT (Foreign Key)
  - iii. seller: INT (Foreign Key)
  - iv. photopath: VARCHAR(45)
  - v. description: VARCHAR(128)
  - vi. title: VARCHAR(45)
  - vii. price: DOUBLE
  - viii. approved: TINYINT
  
- c. Categories Table:
  - i. idCategories: INT (Primary Key)
  - ii. title: VARCHAR(45)
  
- d. Reviews Table:
  - i. idReviews: INT (Primary Key)
  - ii. reviewer: INT (Foreign Key)
  - iii. reviewee: INT (Foreign Key)
  - iv. rating: TINYINT(5)
  - v. header: VARCHAR(45)
  - vi. body: VARCHAR(128)
  
- e. Messages Table:
  - i. idMessages: INT (Primary Key)
  - ii. item: INT (Foreign Key)
  - iii. sender: INT (Foreign Key)
  - iv. receiver: INT (Foreign Key)
  - v. body: VARCHAR(128)

### B. Media Storage:

Images are stored on the file system. The database will store pointers to the images, relative to the application directory. The webpages have a link to their corresponding images. The images will be in a designated folder. We will

downscale all images to save space. We do not have plans to support other forms of media (e.g. video/audio/GPS)

C. Search/filter architecture and implementation:

Search is implemented using the MySQL % Like operator. String comparison will be used to find corresponding items. Search items will be organized in FIFO order, so that old listings can be resolved. Item titles (and potentially descriptions) will be searched. Items can also be filtered using their assigned categories. Categories are selected from a drop down menu.

D. Own API(s): [if any]

None

E. Describe any significant non-trivial algorithm or process if any (like rating, ranking, automatic prioritizing of items etc.):

None

F. Changed SW Tools or Frameworks:

None

## 6. Key Risks

A. Skill Risks:

- Team needs to learn MVC Architecture.
- Unfamiliarity with AWS

**How we will address:**

- Team is learning and following the resources and tutorials of MVC and AWS.

B. Schedule Risks: N/A

C. Technical Risks:

- Remote hosting might be an issue temporarily as we have very few free hours left on our AWS account due to an extra instance kept running on the server.

**How we will address:**

- The issue shall be resolved from the month of April as we have shut down the extra instance.

D. Teamwork Risks:

- Sometimes, there is a communication gap between the frontend and backend teams.

**How we will address:**

- Team lead has proposed that the backend lead and the frontend lead schedule a weekly meeting where they each discuss the plans and progress of their respective teams.

E. Legal/Content Risks: N/A

## 7. Project Management

Team members coordinate with each other over Discord and Zoom meetings. The agenda, the work assigned to individual team members and the deadlines are all discussed in the meeting and posted on Discord to keep everyone updated about the same. There are several channels created on the Discord server group which makes it easier to access the particular information for everyone in the team. Once a team member is done with the individual assigned work, he/she updates it on discord to let everyone else know and then also help others if something is left to do before the deadline.

Except for the mandatory zoom meeting after the class, we have our own individual team meeting to discuss the plans for the next few days, assign deliverables and deadlines to all individuals and communicate about any doubts or difficulties. Apart from this general meeting, the backend and frontend teams schedule a short meeting of their own every week. The team lead distributes tasks amongst the frontend and backend team in the general meeting. The frontend and backend leads then distribute tasks to each individual of their respective team in their respective meeting. Also, if the work is interlinked between team members, they schedule internal team meetings on their own to discuss the same.

Team members make sure to complete the tasks assigned and upload the final edited version of it at least a day or two before the deadline so that the document master can just merge all of it, properly format it and submit it on/before time. Once finalized, it is then committed on Github.

As of now, we haven't set up Trello as communication has been pretty smooth over zoom and discord and we didn't want to make things complex. However, we plan to set up Trello after finishing up Milestone 2 as there would be a lot of coding moving forward and hence Trello will simplify communication and project management.

### c. Milestone 3

## **Summary of Milestone 3 Zoom meeting review with Prof. Petkovic and plans for further development**

**Team Number:** 05

**Meeting Date:** 13 April 2022. 7:00 PM

### **Summary of feedback on UI:**

1. Navbar: make search bar shorter
2. Navbar: move “About” and “Post” a bit to the left
3. Homepage: change “Text” to “Books”
4. Sort By Function: change “default” to “choose”
5. Item Detail Cards: add “message seller” button
6. Results Page: change style of item cards to match homepage style
7. Individual Items Page: make font larger
8. Message Modal: move “submit” button to the right
9. Register: move “all fields mandatory” to the top
10. Enable lazy registration
11. Post Page: put “title” field first
12. Post Page: input fields too long; make them “left-justified” and input field length approximate to what would be entered
13. Post Page: mark mandatory fields
14. Post Page: add “cancel” button to the left of “post” button; not too close
15. Post Page: add disclaimer note that says “posts may take up to 24 hours for approval”
16. Post Page: remove black background
17. User Page: remove “reviews”
18. User Page: make one [posts or messages] default and allow toggle between the two
19. User Messages: add “date” to be displayed
20. User Messages: add a ‘sort’ function to allow sorting by item/product
21. User Posts: add ‘delete’ button

**Summary of feedback on code architecture:**

Files include a header section to identify what the file does but for later files, it would be good to include the owner or creator of the file in case something needs clarification. Naming conventions are good.

**Summary of feedback on github usage:**

Keep commit messages descriptive and up to one line. Github management and usage are good.

**Summary of feedback on DB:**

An item's photopath is a relative path. User's passwords are encrypted when saved. The Messages database should include a date column.

**Summary of feedback on teamwork:**

Teamwork has been excellent thus far.

**Any other comments and issues:**

N/A

**Check Point (if given), DUE:**

N/A

**Agreed P1 List of Features for Final Delivery:**

1. Unregistered Users:
  - 1.1** Unregistered users shall be able to register a new account with a valid SFSU email address.
  - 1.2** Unregistered users shall be able to access the homepage of the website.
  - 1.3** Unregistered users shall be able to browse items on the website.
  - 1.4** Unregistered users shall be able to view details of an item, including price, image, item details, etc.

- 1.5** Unregistered users shall be able to search for items on the website.
- 1.6** Unregistered users shall be able to create an account (lazy registration).
- 1.7** Unregistered users shall be able to sort search results by price and title.

2. Registered Users:

- 2.1** Registered users inherit all functional requirements from Unregistered users.
- 2.2** Registered users shall be able to login
- 2.3** Registered users shall be able to change their password.
- 2.4** Registered users shall be able to post items to sell on the website.
- 2.5** Registered users shall be able to send messages to other registered users.  
[one way only]
- 2.6** Registered users shall be able to logout from their account.
- 2.7** Registered users shall be able to sort received messages by ‘items’ and ‘date’.

3. Admin:

- 3.1** Admin inherits all functional requirements from Registered users.
- 3.2** Admin shall be required to authorize items posted by registered users.
- 3.3** Admins shall be able to delete any item from the website.

**Plan of Action:**

For the first two weeks after Milestone 3 Review, the frontend team will work on making the UI changes suggested by Professor Petkovic. In the meantime, the backend team will work on making the dimensions same for image thumbnails on the item cards as suggested by Class CTO. The backend team will also start connecting the user pages like My Posts and My Messages to the database.

For the next two weeks, the frontend team will work on making the UI responsive for smaller screens like mobile devices. The backend team plans to finish all Priority 1 features listed above by this time. After finishing all priority 1 features, the team lead will test all these features extensively for any bugs and then the team will work on resolving them.

In the final week, the team plans to meet two times to do a mock demo of the website. We feel this is important as this exercise will help find and resolve any hidden bugs that might show up on the day of final presentation. So it will make sure the final presentation will happen smoothly without any last minute errors.

d. Milestone 4

# SW Engineering CSC648/848 Spring 2022

GatorTrade

Team05

Milestone 4

14 May 2022

Members:

**Team Lead:** Kishan Patel - kpatel0@sfsu.edu

**Front End Lead:** Michael Cheung

**Front End Member:** Joanne Wong

**Back End Lead:** Faisal Zaheer

**Back End Member:** Juan Hernandez

**Github Master:** Ze Lei

History Table:

| Date Submitted | Date Revised |
|----------------|--------------|
| 14 May 2022    |              |

## Table of Content

|   |           |
|---|-----------|
| <b>Product Summary</b>  | <b>3</b>  |
| <b>Usability Test Plan</b>  | <b>4</b>  |
| <b>QA Test Plans</b>  | <b>6</b>  |
| <b>Code Review</b>  | <b>8</b>  |
| <b>Self-Check on Best Practices for Security</b>                    | <b>11</b> |
| <b>Self-Check on the Adherence to Original Non-functional specs</b> | <b>12</b> |

## 1. Product Summary:

- a. Name of Product: GatorTrade
- b. Description of Product:

Our application is tailored to the students, faculty, and staff at SFSU.

Belongings they may not want anymore or think others can benefit from can be listed here. The platform will allow for SFSU students, faculty, and staff to buy, sell, or exchange items with one another. This means that the used textbooks, electronics, and other used or handcrafted goods can find new owners amongst the SFSU community. Users of this application can search and filter through categories for the goods they are looking for such as technology, books, or crafts. Also, this application is exclusively for the SF State community only. It can also be an outlet for artistic students to showcase their skills and sell handmade goods to fellow students and SFSU employees.

- c. Itemized list of committed P1 functions:

- i. The application allows users to:
    1. register, login, and logout.
    2. search and browse items.
    3. sort items by their price and title.
  - ii. Registered users are allowed to:
    1. post items for sale.
    2. message other sellers.
    3. view the posts they've made and messages sent to them.
    4. sort messages received by item and date.
    5. change their password.
  - iii. Admins are required to:
    1. approve posts made by users or delete an item listing.

- d. Uniqueness:

- i. GatorTrade allows for its users to experience exclusivity since it is limited to only SFSU personnel and students. Therefore, users can find goods and items within the SFSU community easily.

- e. URL to product: [GatorTrade URL Link](#)

## 2. Usability Test Plan:

### a. Test Objective:

The purpose of this test is to assess the usability of the post item for sale function.

### b. Test Background and Setup:

- i. System Set-Up: First, have access to a device that can connect to the internet. Next, enter or click on the link that leads to the GatorTrade website application.
- ii. Starting Point: The starting point to test this function is the homepage of the GatorTrade website application.
- iii. Intended Users: The intended users of this function are the students, staff, and faculty of San Francisco State University.
- iv. URL of function to be tested: <http://3.84.15.167:3000/post>

### c. Usability Task Description:

First, make sure you are on the GatorTrade homepage. Next, post an item. Then, enter relevant information regarding the item you want to list. If prompted to register or log in, please do so. Afterwards, submit your post and check the status of the post in the My Pages section.

### d. Evaluation of Effectiveness:

Effectiveness shall be measured by whether a user is able to upload a post to the website. The user should be able to access the homepage, go to the post form page, and enter relevant information to be uploaded for sale or exchange. This shall be measured by the number of successful completion of the function out of the total number of tests that are to be performed. Errors and bugs detected shall also be taken into account for the effectiveness of the function.

### e. Evaluation of Efficiency:

Efficiency shall be measured by the amount of effort and time it takes the user to perform the function. The effort would be calculated from the number of webpages required to be accessed before being able to complete the task. The amount of time would be determined from the average time it takes each

successful completion of the task out of the average time of all tasks, completed or uncompleted.

f. Evaluation of User Satisfaction:

- i. 1. It was easy to reach the form to upload an item to the website.

| Strongly Disagree | Disagree | Somewhat Disagree | Neutral | Somewhat Agree | Agree | Strongly Agree |
|-------------------|----------|-------------------|---------|----------------|-------|----------------|
|                   |          |                   |         |                | ✓     |                |

- ii. 2. I was able to upload information about the item I wanted to list.

| Strongly Disagree | Disagree | Somewhat Disagree | Neutral | Somewhat Agree | Agree | Strongly Agree |
|-------------------|----------|-------------------|---------|----------------|-------|----------------|
|                   |          |                   |         |                |       | ✓              |

- iii. 3. My experience with the user interface/design was pleasant.

| Strongly Disagree | Disagree | Somewhat Disagree | Neutral | Somewhat Agree | Agree | Strongly Agree |
|-------------------|----------|-------------------|---------|----------------|-------|----------------|
|                   |          | ✓                 |         |                |       |                |

### 3. QA Test Plans:

- a. Test Objectives:
  - i. The post item function is to be tested for any bugs.
- b. Hardware and Software Setup:
  - i. **HW:** A computer or smartphone that has a connection to the internet.
  - ii. **SW:** The test will start at this url: <http://3.84.15.167:3000/post> while the user is logged in or not logged in.
- c. Feature to be Tested:
  - i. Post functionality
- d. QA Test Plan - Table Format:

| Test No. | Test Title                  | Description  | Test Input   | Expected Output   | Results: (Chrome)<br>PASS/FAIL | Results: (MS Edge)<br>PASS/FAIL |
|----------|-----------------------------|--|--|---|--------------------------------|---------------------------------|
| 1        | Post with user logged in    | Test post function with title, category, price, description, and image field filled out and user logged in.      | <b>Title:</b> iPhone 10<br><b>Category:</b> Electronics<br><b>Price:</b> 300<br><b>Description:</b> Used iPhone 10 for sale.<br><b>Image:</b> 1952 x 1384 png file | The Items table in the database should have a new row with all columns populated respectively and approved field 0 for false. | PASS                           | PASS                            |
| 2        | Post without user logged in | Test post function with title, category, price, description, and image fields filled out and user not logged in. | <b>Title:</b> iPhone 10<br><b>Category:</b> Electronics<br><b>Price:</b> 300<br><b>Description:</b> Used iPhone 10 for sale.<br><b>Image:</b> 1952 x 1384 png file | Users get redirected to login/register, but all fields data are saved for when they go back to the post page.                 | FAIL                           | FAIL                            |

|   |                 |  |  |  |      |      |
|---|-----------------|--|--|--|------|------|
| 3 | Size validation | Test title, price, and description fields for size validation. | <b>Title:</b> type a title with characters > 45.<br><b>Category:</b> Electronics<br><b>Price:</b> 2147483648<br><b>Description:</b> type a description with characters > 1024. | Title only allows max 45 characters.<br>Category will have Electronics.<br>Price will not allow users to input numbers greater than 2147483647 or less than 0.<br>Description will not allow characters > 1024 | FAIL | FAIL |
|---|-----------------|--|--|--|------|------|

## 4. Code Review:

### Sent Emails

FZ Faisal Zaheer  
To: Kishan Ripalkumar Patel

Hi Kishan,

Here is the post feature code of the GatorTrade application code that needs to be code reviewed. I've attached a screenshot of the entire section of code that needs to be reviewed. This portion of the code is from the milestone4 branch on Github. The two files needing reviewed are the posts.js file, located in ./application/routes/posts.js for reference, and Posts.js, located in ./application/models/Posts.js, and the code is from lines 19 - 97 in the /routes/posts.js file mentioned, and lines 10 - 21 in the /models/Posts.js file mentioned. My suggestion for code review would be to include comments with the preface of KP for your name (for example: // KP: this is a comment from Kishan). I am ok with another form of written review, so long as you let me know how and where to find the suggestions/comments. Thank you very much Kishan.

Regards,

Faisal

**Faisal Zaheer** (Back End Lead)  
fzaheer@mail.sfsu.edu

Wed 5/11/2022 11:52 AM

KP Kishan Ripalkumar Patel  
To: Faisal Zaheer

Hey Faisal,

I've reviewed the code and pushed the commented files on the milestone4 branch.

I've added all my comments at the bottom of both files and added line numbers where changes are required. I think this would make it easier for you as you won't need to track each comment separately.

Here's my summary:

1. Header comments are present for each file.
2. In line comments are ample and explain code sufficiently.
3. Method and variable names are consistent. They properly define the role of the method or variable.
4. Not a lot of coding errors overall. I've addressed the few that are in my comments in the code.

Thanks a lot Faisal.

Best,  
Kishan (Team Lead)

Fri 5/13/2022 7:23 AM

### Reviewed Code

```

routes > JS posts.js > ⚡ router.post('/createPost') callback > ⚡ then() callback
  19 // Creates storage in public/images folder to store user uploaded images.
  20 var storage = multer.diskStorage({
  21   destination: function(req, file, cb) {
  22     cb(null, "public/images");
  23   },
  24   filename: function(req, file, cb) {
  25     let fileExt = file.mimetype.split('/')[1];
  26     let randomName = crypto.randomBytes(22).toString("hex");
  27     cb(null, `${randomName}.${fileExt}`);
  28   }
  29 });
  30
  31 // Connects established multer storage to upload files.
  32 var uploader = multer({storage: storage});
  33
  34 // CREATE POST:
  35 router.post('/createPost', uploader.single("itemImage"), (req, res, next) => {
  36   // Uploaded File + Created Thumbnail paths
  37   let fileUploaded = req.file.path;
  38   let filePath = "images/" + req.file.filename;
  39   let fileAsThumbnail = `thumbnails-${req.file.filename}`;
  40   let destinationOfThumbnail = req.file.destination + "/thumbnails/" + fileAsThumbnail;
  41   let thumbnailPath = "images/thumbnails/" + fileAsThumbnail;
  42
  43   // Targeted Post Item form input data:
  44   let title = req.body.itemTitle;
  45   let category = req.body.category;
  46   let categoryId = parseInt(category);
  47   let price = req.body.itemPrice;
  48   let description = req.body.itemDesc;
  49
  50   // Gets seller's user id from their logged in session:
  51   let seller = req.session.userId;
  52
  53   if(seller == undefined){
  54     res.redirect("/login");
  55   }
  56
  57   // Sharp resizes uploaded image to a thumbnail version of the image
  58   // with a 200x200 size, then exports to destinationOfThumbnail,
  59   // in thumbnails folder.
  60   else{
  61     sharp(fileUploaded)
  62     .png()
  63     .resize(200, 200, {
  64       // Sets image to fit the 200x200 dimensions as much as possible while prioritizing its aspect ratio.
  65       fit: 'contain',
  66       background: { r: 255, g: 255, b: 255, alpha: 0 } // Empty space is transparent.
  67     })
  68     .toFile(destinationOfThumbnail)
  69     .then(() => [
  70       // Sends user inputted post info to the post model to enter into the DB.
  71       return PostModel.create(title, categoryId, description, filePath, thumbnailPath, price, seller)
  72     ])
  73     .then((postWasCreated) => {
  74       if(postWasCreated) {
  75         console.log("Your post was created successfully!");
  76         req.flash('success', "Your post was created successfully!");
  77         // Redirects to home page after successful post creation.
  78         req.session.save( err => {
  79           res.redirect('/');
  80         });
  81       } else {
  82         console.log("Your post was NOT created.");
  83         throw new PostError('Post could not be created!!', '/post', 200);
  84       }
  85     })
  86     .catch((err) => {
  87       if(err instanceof PostError) {
  88         errorPrint(err.getMessage());
  89         req.flash('error', err.getMessage());
  90         res.status(err.getStatus());
  91         res.redirect(err.getRedirectURL());
  92       } else {
  93         next(err);
  94       }
  95     });
  96   }
  97 });

```

```
models > js Posts.js > o create > o then() callback
  9
10 | // Handles sending the Post info to the database to create a new Post record.
11 | PostModel.create = (title, category, description, photopath, thumbnail, price, seller) => {
12 |   let baseSQL = 'INSERT INTO GatorTrade.Items (title, category, description, photopath, thumbnail, price, seller) VALUES (?, ?, ?, ?, ?, ?, ?)';
13 |   console.log("Parameters:", title, category, description, photopath, thumbnail, price, seller);
14 |   return db
15 |     .execute(baseSQL, [title, category, description, photopath, thumbnail, price, seller])
16 |     .then(((results, fields) => [
17 |       // Returns whether the Promise statement of creating a new post has succeeded or not.
18 |       return Promise.resolve(results && results.affectedRows);
19 |     ])
20 |     .catch(((err) => Promise.reject(err)));
21 | };

```

## Code Review Feedback

Feedback for ./application/routes/posts.js

```
// KP: Code Review by Kishan Patel for Milestone 4
//      1. Header and in-line comments are properly used. Provide all the information required and makes understanding the code really easy.
//      2. Method and variable names are consistent. The names clearly defines the role of the method or variable.
//      3. Line 60. Try to put the else part directly below if. Nothing in between.
//      4. Line 53. Not a big issue. I would just recommend using === instead of ==

```

Feedback for ./application/models/Posts.js

```
// KP: Code Review by Kishan Patel for Milestone 4
//      1. Header and in-line comments are properly used. Provide all the information required and makes understanding the code really easy.
//      2. Method and variable names are consistent. The names clearly defines the role of the method or variable.
//      3. In the header comment for this file, name of the author is missng.
//      4. Line 12. There is a console log present. Remove it if not required.
//      5. Line 13. Not a big issue. I think it would be better if the .execute will either be on the same line, directly following
//          db or indented directly below db instead of return.( Just a suggestion )

```

## 5. Self-Check on Best Practices for Security:

| Asset to be Protected | Types of Possible/Expected Attacks | Strategy to Mitigate/Protect Asset                      |
|-----------------------|------------------------------------|---|
| User Passwords        | Database Breach                    | Password Encryption (bcrypt)                            |
| Image Links           | Database Exploit                   | File Encryption(sharp/multer)                           |
| Input Validation      | Invalid input -> app crash         | Try catch blocks, regex validation on client and server |
| Product Information   | Invalid input                      | Try catch blocks, client and server javascript          |
| User email            | Invalid email                      | Regex on client and server checking for sfsu.edu        |
| Search Bar Input      | Invalid input                      | Try catch blocks, client and server regex               |
| Search Bar Input      | SQL injection                      | Use of ? in SQL query                                   |

## 6. Self-Check on the 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.**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.**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.**DONE**
7. The language used shall be English (no localization needed).**DONE**
8. Application shall be very easy to use and intuitive.**DONE**
9. Application should follow established architecture patterns.**DONE**
10. Application code and its repository shall be easy to inspect and maintain.**DONE**
11. Google analytics shall be used.**ON TRACK**
12. No email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application.**DONE**
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.**DONE**
14. Site security: basic best practices shall be applied (as covered in the class) for main data items.**DONE**
15. Media formats shall be standard as used in the market today.**DONE**
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development.**DONE**
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "*SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only*" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).**DONE**

### III. Product Screenshots

#### Home

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**GATOR TRADE**

All Search About Post Login

**Buy, Sell, and Exchange On GatorTrade!**

**Electronics**

iPad

Like New nulla malesuada pellentesque elit eget gravida cum

iPhone

Like New nulla malesuada pellentesque elit eget gravida cum

Samsung S22

Like New nulla malesuada pellentesque elit eget gravida cum

Razer Orochi Mouse

Like New nulla malesuada pellentesque elit eget gravida cum

**Electronics**

iPad

Like New nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

\$200.00 Message Seller

iPhone

Like New nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

\$400.00 Message Seller

Samsung S22

Like New nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

\$400.00 Message Seller

Razer Orochi Mouse

Like New nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

\$40.00 Message Seller

**Books**

CRIMINALLY GOOD ARTEMIS

## About

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post Login

The screenshot shows a grid of six team member profiles. Each profile card has a dark blue header with the member's name and title. Below the header is a circular placeholder image. The profiles are arranged in two rows of three.

| Profile | Name           | Title                          |
|---------|----------------|--------------------------------|
|         | Kishan Patel   | Team Lead                      |
|         | Ze Lei         | Back-End Member, Github Master |
|         | Faisal Zaheer  | Back-End Lead                  |
|         | Juan Hernandez | Back-End Member                |
|         | Joanne Wong    | Front-End Member               |
|         | Michael Cheung | Front-End Lead                 |

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post Login

The screenshot shows a detailed view of Ze Lei's profile. It features a large circular placeholder image at the top. Below the image, the member's name is displayed in bold. A brief bio follows, along with contact information and a logo for Frogger-Software.

**Ze Lei**

Ze is in his last semester at SFSU. He is studying for a B.S. in Computer Science. He plays video games like Teamfight Tactics.

✉ zlei1@sfsu.edu  
📞 (123) 456 - 7890  
🌐 Frogger-Software

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

## Post

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All  About Post Login

**Fill out the form below to post an item for sale!**

Title Of Item  Category  \$

Description of Item

No files currently selected for upload

\*It will take up to 24 hours for your post to be approved.

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

## Search

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All  About Post Login

Showing 30 out of 30 items.

Sort By:

|   |  |  |
|---|--|--|
|  <p><b>Lord of The Rings</b><br/>Category: Books<br/>A fantasy book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis</p> <p>\$11.00 <input type="button" value="Message Seller"/></p> |  <p><b>Harry Potter</b><br/>Category: Books<br/>A magical book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis</p> <p>\$12.00 <input type="button" value="Message Seller"/></p> |  <p><b>Percy Jackson</b><br/>Category: Books<br/>A mythology book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis</p> <p>\$13.00 <input type="button" value="Message Seller"/></p> |
|    |   |    |

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All   About Post Login

Showing 2 out of 30 items.

Sort By: Choose



**Lord of The Rings**  
Category: Books  
A fantasy book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$11.00** [Message Seller](#)



**Lord of The Rings**  
Category: Books  
A fantasy book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$11.00** [Message Seller](#)

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

Books   About Post Login

Showing 10 out of 10 items.

Sort By: Choose



**Lord of The Rings**  
Category: Books  
A fantasy book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$11.00** [Message Seller](#)



**Harry Potter**  
Category: Books  
A magical book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$12.00** [Message Seller](#)



**Percy Jackson**  
Category: Books  
A mythology book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$13.00** [Message Seller](#)







SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**GATOR TRADE**

Food  Search

About Post Login

Showing 5 out of 5 items.

**Sort By:**

|   |  |  |
|---|--|--|
|    |   |    |
| <b>Apple Pie</b><br>Category: Food<br>Tastes Good nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes | <b>Cupcakes</b><br>Category: Food<br>Tastes Good nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes | <b>Italian Coffee</b><br>Category: Food<br>Tastes Good nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes |
| \$15.00 <input type="button" value="Message Seller"/>   | \$13.00 <input type="button" value="Message Seller"/>  | \$14.00 <input type="button" value="Message Seller"/>  |
|    |   |  |

## Login

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**GATOR TRADE**

All  Search

About Post Login

**Sign In**

Show Password

Forgot your password?

SIGN IN

**Hello, Friend!**

Enter your personal details and start journey with us

SIGN UP

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

## Registration

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post Login

Welcome Back Gator!

Login using your SFSU email and continue Trading.

**SIGN IN**

**Create Account**

\* All fields are mandatory.

First Name Last Name

SFSU Email

Password

Re-enter Password

**SIGN UP**

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

## My Page

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post

**My Posts**

**My Messages**

Sort By: Choose

**Lord of The Rings**  
Category: Books  
A fantasy book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$11.00** **Delete**

**Harry Potter**  
Category: Books  
A magical book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$12.00** **Delete**

**Percy Jackson**  
Category: Books  
A mythology book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis

**\$13.00** **Delete**

**Pending Approval**

My Page Settings Log Out

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post

### My Posts

|  |  |   |
|--|--|---|
|  |  |   |
| <b>Artemis Fowl</b><br>Category: Books<br>A book with a horrible movie nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis | <b>Burger</b><br>Category: Food<br>tastes good | <b>Divergent</b><br>Category: Books<br>A dystopian book nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis |
| \$14.00 <a href="#">Delete</a>   | \$2.00 <a href="#">Delete</a>                  | \$15.00 <a href="#">Delete</a>  |

Sort By: Title (A-Z) Choose Price (Low to High) Price (High to Low) Title (Z-A) Title (Z-A)

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post

### My Posts

|  |                        |
|--|------------------------|
|  | Date sent: 2022 May 10 |
| Item: <b>Percy Jackson</b><br>From: Kim Possible<br>Message: Lorem ipsum dolor sit amet, consectetur adipiscing.   |                        |
|  | Date sent: 2022 May 10 |
| Item: <b>Logitech Lightspeed Mouse</b><br>From: Joe Schmuck<br>Message: Insectetur adipiscing elit, sed do ei. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do ei. |                        |
|  | Date sent: 2022 May 10 |
| Item: <b>Artemis Fowl</b><br>From: Jake Long<br>Message: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do ei.   |                        |
|  | Date sent: 2022 May 10 |
| Item: <b>Crochet Plush</b><br>From: Kim Possible<br>Message: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do ei. Lorem ipsum dolor sit amet, co.                   |                        |

Sort By: Choose

## Settings

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post

← Return to My Page

**Settings**

SFSU E-mail: student6@mail.sfsu.edu

First Name: Joe

Last Name: Schmuck

Current Password: Current Password

New Password: New Password  
Must be at least 10 characters long.

Confirm new password: confirm new password

**Confirm** **Cancel**

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

## Product Page

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

All Search About Post

iPhone

\$400.00

About This Item:  
Like New nulla malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

**Message Seller**

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05

SFSU Software Engineering Project CSC 648-848, Spring 2022 For Demonstration Only

**GATOR TRADE**

All Search

**Contact Seller**

X  About Post 

Product: iPhone  
Seller: Jake Long

Custom Message:  
Don't forget to send your contact info where the seller can reach you.

**Submit**

\$400.00

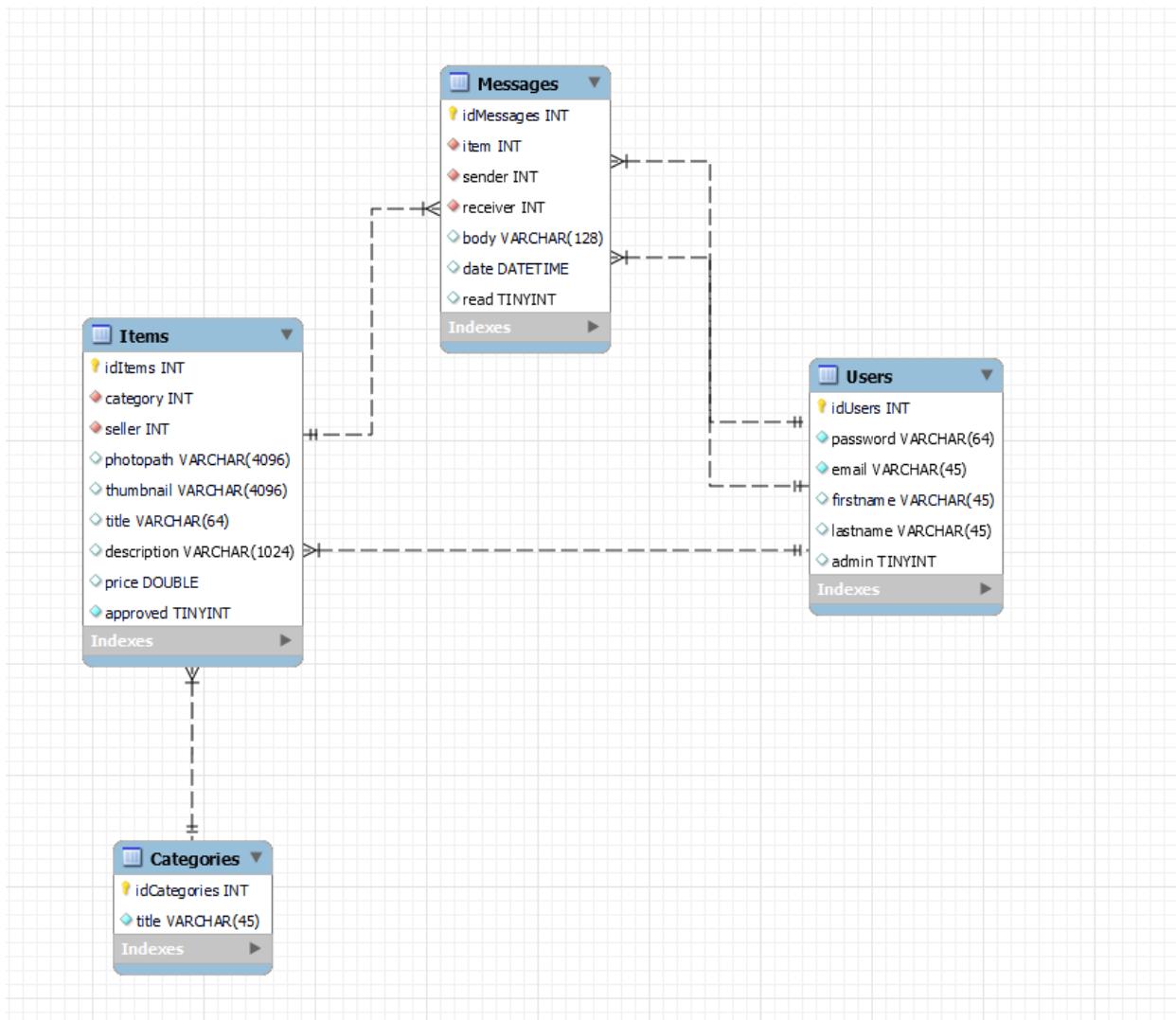
**About This Item:**  
Like New nullà malesuada pellentesque elit eget gravida cum sociis natoque penatibus et magnis dis parturient montes

**Message Seller**

SFSU Software Engineering Project CSC 648-848, Spring 2022, © TEAM 05



#### IV. Database Organization



Messages - Table ×

| Table Name: Messages |              |                                     |                                     |                                     |                          |                                     |                          |                                     |                          |                    |   |
|----------------------|--------------|-------------------------------------|-------------------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|-------------------------------------|--------------------------|--------------------|---|
| Column Name          | Datatype     | PK                                  | NN                                  | UQ                                  | B                        | UN                                  | ZF                       | AI                                  | G                        | Default/Expression |   |
| idMessages           | INT          | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |                    |   |
| item                 | INT          | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |                    |   |
| sender               | INT          | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |                    |   |
| receiver             | INT          | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |                    |   |
| body                 | VARCHAR(128) | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |                    |   |
| date                 | DATETIME     | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |                    |   |
| read                 | TINYINT      | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> | now()              | 0 |

## V. Github Organization

Main Branches:

Main

Dev

Milestone#

Access to Main Branch was restricted to Ze Lei (Github Master) and Kishan Patel (Team Lead)

Top Level Folders:

Milestones - folder that houses documentation for agile software engineering process.

Application - folder that contains the source code for the project

Credentials - folder that contains the documentation relating to the AWS server and database.

The credentials folder also contains the pem file that allows access to the AWS server

**csc648-SFSU / csc648-03-sp22-team05** Private

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main · 3 branches · 0 tags

Frogger-Software Merge branch 'milestone4' into dev · 4 minutes ago · 299 commits

Milestones · application · credentials · .gitignore · LICENSE · README.md

Added Milestone 4 document · Minor bug fix on UI · enabled apache proxy to port 3000 · removed files that do not belong · Initial commit · Update README.md

6 days ago · 27 minutes ago · last month · last month · 3 months ago · 3 months ago

**csc648 Repository**

Please when ready add your teams application URL or IP to the repository description. This will help with grading. Teams are expected to keep this value up to date.

Please do the following steps before completing Milestone 0.

1. Change the name of the repository. All that needs to change is the NN to your respective team number. Team numbers whose value is less than 10, please pad with a 0. Ex team 1 is team01 team 11 is team11. Please make sure to also remove the username from the repository as well. Teams with incorrectly name repository will have points deducted from their milestone 0 grades.

About · Releases · Packages · Contributors · Languages

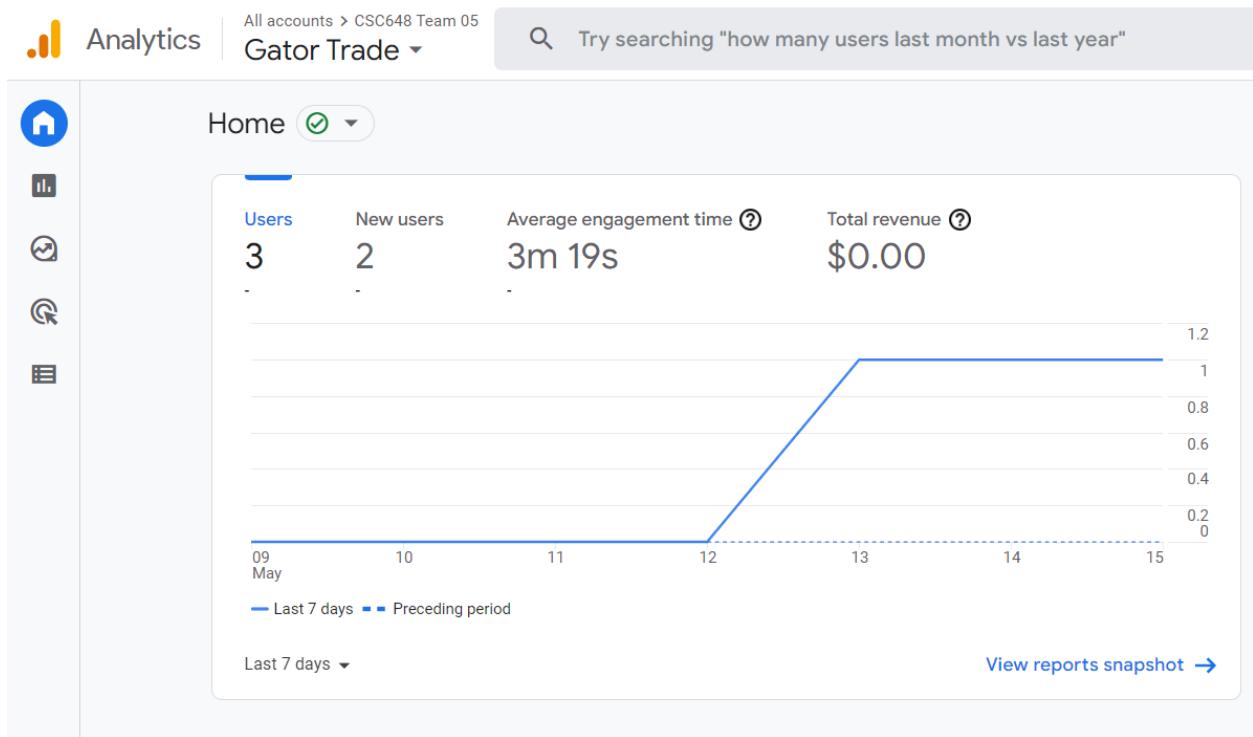
Readme · MIT license · 0 stars · 1 watching · 0 forks

No releases published · Create a new release

No packages published · Publish your first package

JavaScript 83.3% · EJS 14.0% · CSS 1.8%

## VI. Google Analytics Stats Plot for WWW Site



## VII. Project Management

1. We used Trello for project management and Discord as our main communication channel.

Trello:

The screenshot shows a Trello board titled "Milestone 4". The board has four columns: "ToDo", "Doing", "Needs Testing", and "Finished".

- ToDo:**
  - M4 Doc Section 4: Code Review (Due May 18)
  - Sort By Feature on My Posts (Due May 18)
  - Flash Messages (Due May 15)
- Doing:**
  - M4 Doc Section 4: Code Review (Due May 18)
  - Sort By Feature on My Posts (Due May 18)
  - Flash Messages (Due May 15)
- Needs Testing:**
  - Settings Page - Connect to DB (Due May 11)
  - M4 Doc Section 3: QA Test (Due May 13)
- Finished:**
  - Post Page - UI Update (Due May 6)
  - M4 Doc Section 5: Self Check for Security (Due May 11)
  - M4 Doc Section 1: Product Summary (Due May 11)
  - Fix Count of Displayed Results (Due May 11)
  - My Posts - Connect to DB (Due May 11)
  - About Pages - UI Update (Due May 11)

Discord:

The screenshot shows a conversation in the "# general" channel of a server named "CSC648-848 Team 5".

**narcolepticctaco** 05/08/2022: is it fine if i change the input type for price on the post page to "number". I put text into the price and it displays an error page. If not i can add some checks if this is not possible,

**Michael Cheung** 05/08/2022: ohhh  
are you saying if you type something other than an integer in the price field and click on post, it will display an error page?

**narcolepticctaco** 05/08/2022: Yup,

**Michael Cheung** 05/08/2022: yes, you can change the type to number  
good catch  


**narcolepticctaco** 05/08/2022: ok cool. Thanks.

**KP4137** 05/08/2022: If you are using input type number, then make sure to add step="0.01" attribute.

**Michael Cheung** 05/08/2022: also min="0" for positive numbers only

## VIII. Team Member Self-Assessment and Contributions

### 1. Michael Cheung Self-Assessment

#### CSC648-848 Spring 2022 Milestone 5 Team 5 Michael Cheung's Self Assessment



Michael Cheung

To: Kishan Ripalkumar Patel; Ze Lei; Joanne A Wong; Juan Roberto Hernandez; Faisal Saqib Zaheer

Reply Reply All Forward ...

Thu 5/19/2022 9:42 PM

#### Contributions:

1. Created a full-time student use case.
2. Created low-fidelity storyboards for two different use cases.
3. Designed and coded UI for homepage, individual product page, and post page.
4. Implemented QA test plan.
5. Updated responsiveness in smaller devices for Navbar.

Number of commits/submissions made in Github: 24

#### Challenges:

One main challenge that I encountered in this team project was learning new things like how to code in EJS. However, simply by searching on Google, I was able to solve all my problems. For example, I figured out how to transfer data from my router javascript files and use them in my EJS files. Another challenge I faced in this project was designing lazy registration for posting. There were multiple ways to do this, but I chose the fastest way because of time constraints.

#### Improvements:

In future team projects, if I oversaw QA, I could use Selenium to automate the task of testing UI. The usefulness of Selenium would increase as the project size increases. I think it is also a good idea to make sure the page I'm coding currently is responsive on all devices so that I won't have to go back and update responsiveness.

### 2. Faisal Zaheer Self-Assessment

#### CSC648-848 Spring 2022 Milestone 5 Team 5 Faisal Zaheer's Self Assessment



✉ Faisal Saqib Zaheer <fzaheer@mail.sfsu.edu>

To: ✉ Kishan Ripalkumar Patel; ✉ Ze Lei; ✉ Joanne A Wong; ✉ Michael Cheung; ✉ Juan Roberto Hernandez

Yesterday at 10:26 PM

#### Contributions:

1. Created a SFSU Faculty use case.
2. Worked on M1 Functional Requirements list.
3. Developed the Vertical SW Prototype (setting up Node/Express app, setting up routes, refactoring webpages).
4. Connected application to the MySQL database.
5. Created/hosted the AWS instance and helped deploy the application and database on the AWS server.
6. Programmed the search functionality with category selection functionality.
7. Implemented user authentication (Login/Register functionality and sessions).
8. Programmed basic version of Post feature to connect with database.
9. Automated the creation of product image thumbnails whenever a post is created, coded thumbnails to appear on home/search results versus original image on product page (updated DB in accordance with thumbnail/photo paths).
10. Worked with Kishan on Code Review for the Post feature.

Number of commits/submissions made in GitHub: 38

#### Challenges:

Some of the earlier challenges in the development process was setting up and deploying our app on AWS. After searching up several tutorials online on how to deploy a Node app and a MySQL database on AWS, Ze (Peter) and I finally were able to figure it out to get the application online and running. Later challenges with that included issues with the free tier limit, keeping the AWS application on and running, and other minor issues. Another issue I faced was learning how to assign or delegate tasks as a lead for the back-end. I'm used to tackling a coding challenge myself, or being given a task to do, so I had to learn how to split the work given to me and my teammates to get the task done efficiently.

#### Improvements:

In the future, I would want to approach the being a smaller team lead with a mindset to be more knowledgeable in the high level of what is being assigned to my team. I'd communicate more with my team lead and my team members to be more active instead of just completing my tasks and waiting to do the next task. As a team lead, I would take more ownership not just of my tasks but of the whole sub team's tasks and work. To offer my help and to ask for more help whenever applicable.

### 3. Joanne Wong Self-Assessment

**CSC648-848 Spring 2022 Milestone 5 Team 5 Joanne Wong's Self Assessment**

JA  To: Joanne A Wong <jwong45@mail.sfsu.edu> Kishan Ripalkumar Patel; Faisal Saqib Zaheer; Ze Lei; Juan Roberto Hernandez; Michael Cheung

Today at 1:11 AM

Contributions:

1. created and managed milestone documents to ensure completion.
2. worked on milestone 1's list of data items and entities section.
3. designed and created low-fidelity storyboards for use cases in milestone 2.
4. designed and coded the early versions of the navbar, user's post page, user's message inbox page, settings page, about all members page, and about individual member page.
5. connected user's post page and user's message inbox page to the database to grab user data.
6. acted as scribe for milestone 3.
7. worked on product summary and usability test sections in milestone 4.

Number of commits/contributions made in GitHub: 47

Main challenges encountered:  
 One of the main challenges I encountered in the team project was adapting to the different coding styles and techniques used by other members. Another was learning how to use and implement EJS effectively and expanding on a language I've had basic knowledge on like JavaScript. In addition, a smaller challenge that occurred during this team project was reading code written by others and understanding what it is to do. All challenges mentioned were able to be fixed with a bit of research and time.

Improvements to be made:  
 For later team projects, I would like to be more communicative about my issues since many struggles I had could probably be clarified by my group. Another thing I need to improve on is to have intermittent unit testing on the UI before proceeding onto the next UI feature I want to implement instead of implementing multiple features at the same time.

### 4. Ze Lei Self-Assessment

**CSC648-848 Spring 2022 Milestone 5 Team 5 Ze Lei's Self Assessment**

ZL  Ze Lei  
To: Kishan Ripalkumar Patel; Michael Cheung; Joanne A Wong; Faisal Saqib Zaheer +1 other

Fri 5/20/2022 10:18 AM

My Contributions:

- management of GitHub branches
- design, deployment, and maintenance of SQL database
- creation of database sample data
- implementation of Delete Post
- implementation of Flash Messages
- implementation of User Messages (sending to seller)
- implementation of Apache web server
- management of AWS server instance
- auditing of code organization
- auditing of database queries
- fair share of documentation

Submissions to development branch: 92

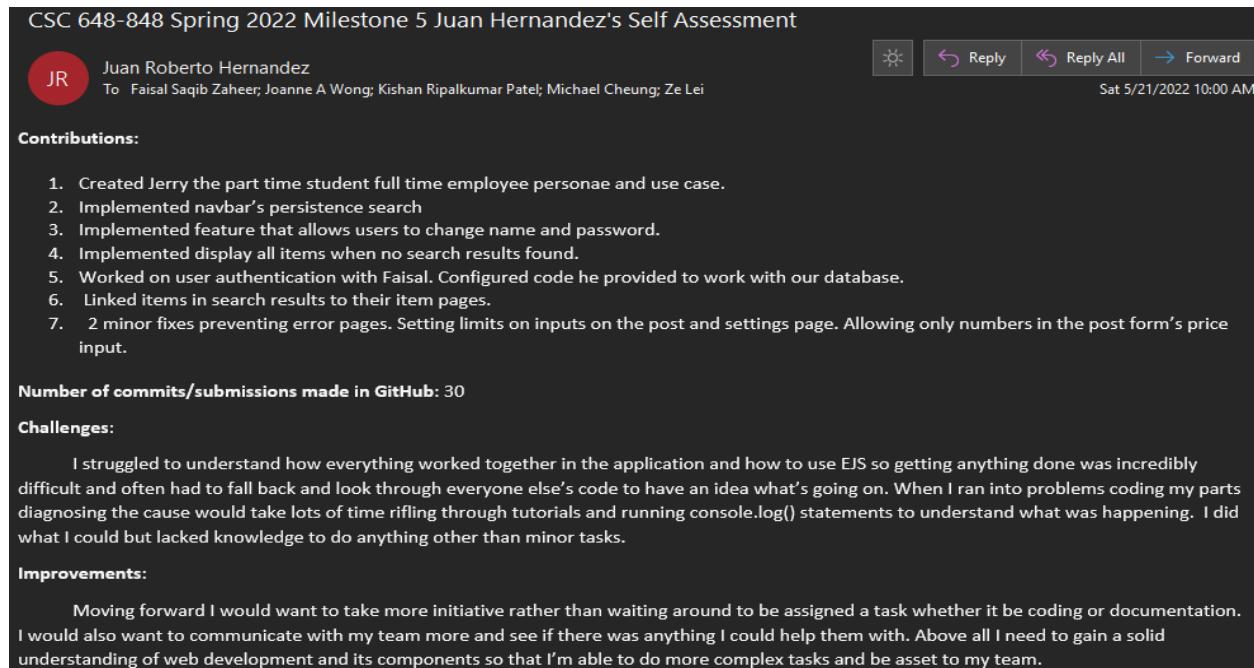
My biggest challenge that I encountered was the burden of knowledge needed to work as a backend member. I have never done anything on the backend before. During this project, I learned how to manage an AWS server. The MySQL on Ubuntu is much more finicky than the MySQL managed by Workbench. However, my experience with databases helped me with the later parts of the project.

The things that I would have done better include coordinating with team members and doing research more often. I have some commits that did not need to be pushed. I have improved my knowledge of git to avoid unnecessary commits. Also, the team's usage of GitHub was suboptimal. Although we avoided having major merge conflicts, we did not make branches for each feature. It was my responsibility as GitHub Master to oversee the usage over git

## 5. Juan Hernandez Self-Assessment

CSC 648-848 Spring 2022 Milestone 5 Juan Hernandez's Self Assessment

JR Juan Roberto Hernandez  
To: Faisal Saqib Zaheer; Joanne A Wong; Kishan Ripalkumar Patel; Michael Cheung; Ze Lei



**Contributions:**

- Created Jerry the part time student full time employee personae and use case.
- Implemented navbar's persistence search
- Implemented feature that allows users to change name and password.
- Implemented display all items when no search results found.
- Worked on user authentication with Faisal. Configured code he provided to work with our database.
- Linked items in search results to their item pages.
- 2 minor fixes preventing error pages. Setting limits on inputs on the post and settings page. Allowing only numbers in the post form's price input.

**Number of commits/submissions made in GitHub:** 30

**Challenges:**

I struggled to understand how everything worked together in the application and how to use EJS so getting anything done was incredibly difficult and often had to fall back and look through everyone else's code to have an idea what's going on. When I ran into problems coding my parts diagnosing the cause would take lots of time rifling through tutorials and running console.log() statements to understand what was happening. I did what I could but lacked knowledge to do anything other than minor tasks.

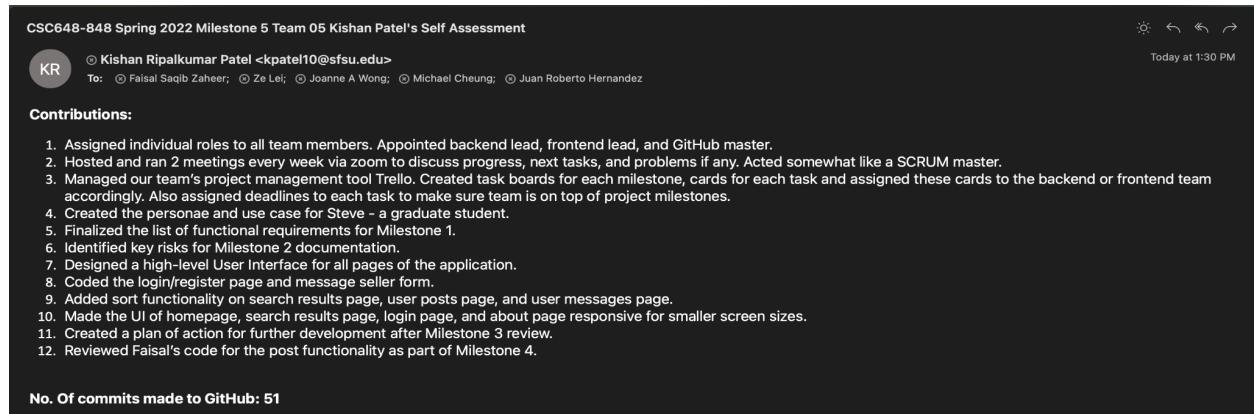
**Improvements:**

Moving forward I would want to take more initiative rather than waiting around to be assigned a task whether it be coding or documentation. I would also want to communicate with my team more and see if there was anything I could help them with. Above all I need to gain a solid understanding of web development and its components so that I'm able to do more complex tasks and be asset to my team.

## 6. Kishan Patel Self-Assessment

CSC648-848 Spring 2022 Milestone 5 Team 05 Kishan Patel's Self Assessment

KR Kishan Ripalkumar Patel <kpatel10@fsu.edu>  
To: Faisal Saqib Zaheer; Ze Lei; Joanne A Wong; Michael Cheung; Juan Roberto Hernandez



**Contributions:**

- Assigned individual roles to all team members. Appointed backend lead, frontend lead, and GitHub master.
- Hosted and ran 2 meetings every week via zoom to discuss progress, next tasks, and problems if any. Acted somewhat like a SCRUM master.
- Managed our team's project management tool Trello. Created task boards for each milestone, cards for each task and assigned these cards to the backend or frontend team accordingly. Also assigned deadlines to each task to make sure team is on top of project milestones.
- Created the personae and use case for Steve - a graduate student.
- Finalized the list of functional requirements for Milestone 1.
- Identified key risks for Milestone 2 documentation.
- Designed a high-level User Interface for all pages of the application.
- Coded the login/register page and message seller form.
- Added sort functionality on search results page, user posts page, and user messages page.
- Made the UI of homepage, search results page, login page, and about page responsive for smaller screen sizes.
- Created a plan of action for further development after Milestone 3 review.
- Reviewed Faisal's code for the post functionality as part of Milestone 4.

**No. of commits made to GitHub:** 51

**Challenges:**

I've developed a few web applications before, but they were either individual projects or I was working in a team of 3 people. This was the first time I was working in a team of 6 people for a software development project, and I was also the team lead. Initially, one of the biggest challenges was working with git and GitHub as I had no experience doing that in a team of more than 3. A couple of online tutorials got me started and then Peter (our GitHub master) did a great job and took a lot of burden off me. As a team lead, I found it difficult at first to communicate with the team. It was more like I was assigning tasks, and everyone was finishing them by their own. I feel the team wasn't comfortable enough to discuss their problems with each other and so, preferred to solve the problems by themselves. However, after we had a few meetings, I feel the team became more comfortable with each other and started working on tasks as a team and not just as a group of individuals.

**Improvements:**

If I get a chance to lead a team for a SE project in the future, I will do some things differently. One of them will be GitHub. While we have managed GitHub quite well and didn't have a single conflict, I acknowledge that we could have done a better job especially with branching. So, I would take the experience gained by that mistake and try to improve the way my team uses GitHub in the future. Also, I feel now I'm much more suited to communicate with a team and I think I'll be able to break the ice much sooner with my team in the future (even if we are meeting online).

Best,  
Kishan