

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

## TEAM 8

Application Name: *PurpleMarket*

### Team

Name	Role	Email
Mark Kim	Team Lead	mkim22@mail.sfsu.edu
Khushboo Gandhi	Front End Lead	
Cody Huang	Back End Lead	
Ernesto Diaz	GitHub Master	
Jesus Cervantes Fajardo		
Vivian Kuang		
Jiasheng Li		

<https://csc648.mskim.dev>

May 19, 2022

# Contents

<b>1</b>	<b>Product Summary</b>	<b>4</b>
<b>2</b>	<b>Milestone Documents</b>	<b>5</b>
2.1	Milestone 1 . . . . .	5
2.1.1	Executive Summary . . . . .	6
2.1.2	Personae . . . . .	7
2.1.3	Use Cases . . . . .	8
2.1.4	Main Data Items and Entities . . . . .	11
2.1.5	List of Functional Requirements . . . . .	12
2.1.6	List of Non-Functional Requirements . . . . .	13
2.1.7	Competitive Analysis . . . . .	13
2.1.8	System Architecture . . . . .	14
2.1.9	Team & Roles . . . . .	15
2.1.10	Checklist . . . . .	15
2.2	Milestone 2 . . . . .	16
2.2.1	Executive Summary . . . . .	17
2.2.2	Main Data Items and Entities . . . . .	17
2.2.3	Functional Requirements - Prioritized . . . . .	19
2.2.4	User Interface Storyboards . . . . .	21
2.2.5	Architecture . . . . .	35
2.2.6	Key Risks . . . . .	37
2.2.7	Project Management . . . . .	39
2.3	Milestone 3 . . . . .	40
2.3.1	Summary of feedback on UI . . . . .	41
2.3.2	Summary of feedback on code and architecture . . . . .	41
2.3.3	Summary of feedback on github usage . . . . .	41

2.3.4	Summary of feedback on DB . . . . .	41
2.3.5	Summary of feedback on teamwork . . . . .	42
2.3.6	P1 feature list . . . . .	42
2.3.7	Plan of Action . . . . .	43
2.4	Milestone 4 . . . . .	44
2.4.1	Product Summary . . . . .	45
2.4.2	Usability Test Plan - max 2 pages . . . . .	46
2.4.3	QA Test Plan - max 2 pages . . . . .	48
2.4.4	Code Review . . . . .	51
2.4.5	Self-check on best practices for security – 1/2 page . . . . .	55
2.4.6	Self-check of the adherence to original Non-functional specs performed by Team Leads . . . . .	56
<b>3</b>	<b>Product Screenshots</b>	<b>57</b>
3.1	Home Page . . . . .	57
3.2	Post Item Page . . . . .	59
3.3	Register Page . . . . .	61
3.4	Login Page . . . . .	62
3.5	Password Reset Page . . . . .	62
3.6	About Page . . . . .	63
3.7	User Dashboard . . . . .	64
<b>4</b>	<b>Database Organization</b>	<b>66</b>
4.1	Database Schema . . . . .	66
4.2	Database Table Screenshots . . . . .	67
<b>5</b>	<b>GitHub Organization</b>	<b>69</b>
5.1	Main Branches . . . . .	69
5.2	File Organization . . . . .	69
5.2.1	Top Level Folders . . . . .	69
5.2.2	Application Folder . . . . .	69
5.2.3	GitHub homepage screenshot . . . . .	71
<b>6</b>	<b>Google Analytics</b>	<b>72</b>

<b>7 Project Management</b>	<b>73</b>
7.1 Trello . . . . .	73
7.2 Google Docs . . . . .	74
<b>8 Team member self assessment and contributions</b>	<b>75</b>
8.1 Mark Kim . . . . .	75
8.2 Khushboo Gandhi . . . . .	76
8.3 Cody Huang . . . . .	77
8.4 Jiasheng Li . . . . .	78
8.5 Vivian Kuang . . . . .	79
8.6 Ernesto Diaz . . . . .	80
8.7 Jesus Cervantes Fajardo . . . . .	81

## 1 Product Summary

The name of our product is Purple Market. It is a web application that connects students, staff, and faculty from San Francisco State University. It is a marketplace where students and staff can buy and sell products to others at SFSU. We create a safe way to shop by manually verifying each post before publishing it for sale. We provide a custom tailored experience for people associated with SFSU. Our website is simple to navigate and easy to use. There is no clutter with our clean design. We sort all of our products into general and easy to understand categories. Users are able to search for the items they want in the search bar or filter out the ones they are looking for by categories. Once a user finds their product they can sign up on our site and send the seller a message.

Our application allows users to:

- Search the site for certain posts.
- Filter search results.
- Register for an account

Once registered, users will be able to:

- Create posts.
- Get in contact with poster/seller
- Edit and correct and/or delete their own post.
- Login/out of their account.
- Reset their password.
- Inspect their transaction history.

Finally admins will use MySQL workbench to:

- Do everything a registered user does.
- Approve or decline an item post.

Website url: <https://csc648.mskim.dev>

## 2 Milestone Documents

### 2.1 Milestone 1

# SW Engineering CSC648/848 Spring 2022

*PurpleMarket*

TEAM 8

Name	Role	Email
Mark Kim	Team Lead	mkim22@mail.sfsu.edu
Khushboo Gandhi	Front End Lead	
Cody Huang	Back End Lead	
Ernesto Diaz	GitHub Master	
Jesus Cervantes Fajardo		
Vivian Kuang		
Jiasheng Li		

Milestone 1

March 11, 2022

Milestone/Version	Date
M1V1	03/03/2022
M1V2	03/09/2022

### **2.1.1 Executive Summary**

PurpleMarket is a web application that will connect students, staff, and faculty from San Francisco State together via a virtual marketplace. In this marketplace, they will be able to communicate with each other, as well as buy and sell products on a user to user basis. There will be no intermediate and no surprises. What one user posts will be verified and will be exactly what is sold to another user. With this in mind we hope to create a safer, open community at San Francisco State, while also providing our users with a more convenient way to shop. The value of PurpleMarket is in its ability to provide an experience custom tailored to people associated with San Francisco State. Our vision is to provide a marketplace that facilitate the ability to sell and purchase the types of items and/or services that are most important to our users.

PurpleMarket plans to be easy to navigate, and easy on the eyes. There isn't any clutter which makes the site much more presentable to our users. Our users are presented with a clean design that will make it much easier for users to navigate our site to find exactly what they are looking for. If they have any trouble at all finding their specific product, they are able to narrow their search in three ways. The first method to finding their desired product is by browsing different categories. By providing categories that our users are most interested in, our users will be able to more easily sell or find the products that they want. The second method will be a simple and easy to use search that would allow the user to type specific titles, names, or descriptions where the user will be presented with the results of the search. Specific to the academic environment, the final method for finding a product is by finding products that are associated with a particular course. Once a user finds what they are looking for they can sign up and create a PurpleMarket account that will allow them to communicate with the user that listed the product directly on our site and the rest is history. So what about listing products? Any verified user who opens an account will be able to list a product. Finally, PurpleMarket will be able to facilitate on-campus meetings between buyer and seller, which simplifies the transaction experience.

Our team of passionate San Francisco State students know the frustrations of navigating online classifieds and the unique needs of the people associated with the school. It is because of this that we are exceptionally equipped to understand the San Francisco State community and provide them with the experience that they desire.

## 2.1.2 Personae

### Persona 1

**Actor:** Kobe (Admin)

#### **About:**

- Busy student
- Loves to explore computer technologies
- Likes simple and clean user interfaces
- Expert in maintaining and troubleshooting computer technologies

#### **Goals & Scenario:**

- Wants to ensure a safe and efficient user experience for students, staff, and faculty of San Francisco State University.
- Loves to explore computer technologies
- Likes simple and clean user interfaces
- Expert in maintaining and troubleshooting computer technologies

### Persona 2

**Actor:** Claire (Student)

#### **About:**

- Busy student that's proud of attending SFSU.
- Spends most of time at the university library or attending classes
- Active and loves to exercise
- Very comfortable with web apps

#### **Goals & Scenario:**

- Decided to go to the library. Checks to see if she has all her required notes and materials on her laptop.
- While at the library, she likes the idea of finding SFSU merchandise for sale by other students.
- Carries her gym clothes in her bag so she can go exercise after her study session.

### Persona 3

**Actor:** Anna (Faculty)

#### **About:**

- Likes to have everything in order.
- Not very comfortable with web apps.

- Plans far into the future.
- Interested in e-textbooks and other e-content for her students.
- Loves reading literature and exploring different texts.

**Goals & Scenario:**

- Is reading a research paper on the community history of geographies of displacement and taking mental notes of points that must be shared to her students at class today.
- She uses her notebook and pens them down. She creates a questionnaire for students using google docs which must be uploaded on ilearn later.

**Persona 4**

**Actor:** Curry (Staff)

**About:**

- Has a regular working schedule
- Work in an establishment with administrative functions such as accounting and finance.
- Have a university educational level.
- Have higher-level responsibilities of being a university administrator.
- Prefer to follow their job orders and university principles

**Goals & Scenario:**

- When he has a meeting with someone, he always wants the person arrives on time. As a result, he desires that the person will have a reminder when the meeting comes soon.
- When he sets up a meeting to buy something from other people associated with SFSU, he wants to be able to set up a meeting place and time that is convenient for him, since his schedule is very tight.

### 2.1.3 Use Cases

#### Use Case 1: Administrative Functions (Kobe)

Kobe is an admin user of the app. He opens his laptop and browses to the web page of the app while he is sitting in front of a desk at his home. After logging into his account, he needs to troubleshoot the app to ensure all features and functionalities of the app are working correctly. After that, he wants to review and validate the new posts of products to ensure that they are appropriate and follow company guidelines. Valid posts will be approved by Kobe and can be viewed by all users. Kobe will deny invalid posts and give an explanation for why the post is invalid. Finally, Kobe reviews the feedback from app users and decides any

future actions need to handle.

### **Use Case 2: Purchase SFSU Apparel (Claire)**

Claire is laying on her bed while she thinks about her next swimming event. She thinks that her SFSU suit is worn out and definitely needs an upgrade. She takes her smartphone and opens the web browser. She googles PurpleMarket and opens the website. She goes on categories and clicks “SFSU Apparel”. She browses through the different options on her phone and reads their reviews. While browsing, she decides to also look for more SFSU merchandise and browses through SFSU mugs, keychains and SFSU notebooks for her desk. While she is at it, she also goes through some of the notes which are offered by past students for her classes. Finally, she’s happy and when she attempts to message the sellers in the web application to set up a time and place to meet to purchase the items, the web application prompts her to login or register. Because Claire is on campus most of the time, it is easy for her to meet up on campus to complete the transaction. When she shows up to the agreed location to complete the transaction, one of the sellers does not show up. She is not too upset because she was able to complete a couple of other things, so she leaves a message to the seller on the website to see if they can schedule another meeting. The second time around, the seller shows up and the transaction is completed. She leaves a mediocre review of the seller because the seller did not let her know of the cancellation ahead of time.

### **Use Case 3: Request Items (Anna)**

Anna is sitting on her front desk one day before the beginning of the first day of instruction. She wants to share handwritten notes of her past student and is hoping to contact him for the same. She opens the www website and looks for the notes under the class. Thankfully she finds the student having a post about the same. When she attempts to contact him about it, the website prompts her to login or register. Once logged in and/or registered she is able to message her past student and set up a meeting on campus to complete the transaction.

### **Use Case 4: Purchase through browsing (Curry)**

Curry is an SFSU full-time university staff. He wants to shop for some items on the secondary market held by the SFSU community, so he opens the web application and browses posts of items in his office during his break. First, he just randomly browsed items by categories including books, electronic products, and household supplies. After a couple of minutes, he realizes that there is a book that he was interested to buy recently. He enters the name of that book to search listed posts, and then there are a couple of listed posts that show up. He clicks on one post in order to find more information about the item and sell information.

He messages the seller telling him that he wants to buy the book. Finally, using the website, Curry sets up a meeting time and place on campus to complete the transaction. His experience is great, so he leaves a positive review of the seller.

### **Use Case 5: Selling furniture (Anna)**

Anna feels like her house is too cluttered with some old furniture she no longer uses. One item she wants to sell is a small bookshelf. She has looked at eBay and the process to sell items seems too complicated and she does not want to deal with having to ship something or coordinating a meetup with someone who might be coming from farther away. In addition, Anna has heard that Craigslist has a lot of scammers, so she is uncomfortable with providing any sort of contact information there. Because PurpleMarket only allows people associated with SFSU, she feels safer to meet with a potential buyer. Also, the website gives her a simple way to be contacted without providing her personal contact information and coordinate a meeting to complete the transaction. Once she goes onto the website, she is able to easily find the button to sell her items. Through some simple prompts, she is able to enter in a title, description, price, and photos of her items. Once she composes her sales listing, the website prompts her to login or register. Once she requests the listing, the website informs her of the required administrative approval process. An admin quickly approves the post and she is notified that her listing was approved and published. Three days later, she gets a message directly on the website inquiring about her bookshelf. Through the website, she is able to respond and suggest a meeting spot on-campus. After agreeing on a price and location, she is able to put the meeting on her schedule. When she shows up to the agreed upon location, the buyer never shows up. Because of this, she leaves a bad review for the buyer.

### **Use Case 6: Selling textbook (Claire)**

Before the new semester starts, Claire thinks it is a good time to sell a history textbook she used last semester. Since she knows that her book is required for the class, she is certain that students will need that exact textbook for the history class in the coming semester. Because of this, she understands that she will be able to sell it quickly to students at SFSU without the hassle of shipping or coordinating a meeting someone that is not associated with the school. Since she is at the library, she decides that using her phone would be the easiest way to post her listing. With a prominent post button on the main page, it is easy for her to start her listing. Once on the composition page, she is able to quickly run through the prompts to list her book. One feature she is grateful for is the ability to associate the book with the specific class that requires it. Once she snaps a quick photo on her phone, she completes the listing and the website then prompts her to login or register. Once logged in, the website informs her of the required administrative approval process.

Once approved, Claire is notified that her listing was approved and published. To her pleasant surprise, she is immediately contacted on the website by another student inquiring about her book. Through the website, she is able to respond and suggest a meeting spot on-campus. After agreeing on the price and location, she is able to complete the transaction on-campus. Because she was happy with the transaction, she leaves a glowing review of the purchaser.

#### 2.1.4 Main Data Items and Entities

**user:** The users are the people expected to use the web application. This entity will contain all the pertinent information that describes our user like their name, email address, and other identifying information.

**item:** This entity are the items that are being bought, sold, or provided. This entity contains all the information that helps users find what they want. For example, this entity will have a name, a description, price, a category that it belongs to, and photos so that they can see what the item looks like.

**category:** This is the category a particular item will belong to. For example, items may be furniture, books, electronics, etc. We want this data item to be easily changed or expanded if necessary in case our users find categories that they want.

**account:** This will contain all the account information of our users as well as describe their permissions. This entity will contain the user's account password, the account type (e.g. Admin, Registered User, etc), and the permissions that are associated with the account type.

**location:** This will contain different locations that can be used on-campus where transactions can be completed. This is meant to provide an easy way for buyers and sellers to coordinate a meeting to complete a transaction.

**message:** This entity exists to contain all the messages that are sent from one user to another. This will contain not only the message, but also the sender, recipient, and a time stamp of when the message was sent and received/read.

**review:** This data item contains the reviews for all users. This entity will contain the review itself, the recipient of the review as well as the person who wrote the review, the time stamp of the review, and also the associated transaction/item that goes with the review.

## **2.1.5 List of Functional Requirements**

### **Unregistered User:**

1. Unregistered users shall be able to search the site for certain posts.
2. Unregistered users shall be able to filter search results.
3. Unregistered users shall be able to register for an account.

### **Registered User:**

5. Registered users shall be able to do everything an unregistered user can do.
6. Registered users shall be able to create posts.
7. Registered users shall be able to get in contact with the poster/seller.
8. Registered users shall be able to edit and correct and/or delete their own post after it was submitted.
9. Registered users shall be able to report violations of terms of service of another user.
10. Registered users shall be able to login to their account.
11. Registered users shall be able to logout of their account.
12. Registered users shall be able to request/reset their password if they forgot.
13. Registered users shall be able to leave reviews of other users regarding completed transactions.
14. Registered users shall be able to propose meeting locations or modify location proposals to complete transactions.
15. Registered users shall be able to inspect their transaction history.

### **Administrator:**

16. Admins shall be able to do everything a registered user can do.
17. Admins shall be able to delete and edit posts done by any user.
18. Admins shall be able to contact any user.
19. Admins shall be able to ban users.
20. Admins shall be required to approve posts before they can go live.
21. Admins shall be able to inspect any user's transaction history.

### **2.1.6 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 e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging.  
One round of messaging (from user to seller) is enough for this application
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "*SFSU Software Engineering Project CSC 648-848, 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).

### **2.1.7 Competitive Analysis**

Our web app has numerous planned advantages compared to our competitors. We make sure that all users of our app are SFSU students, staff, or faculty. We make sure that buyers can easily contact the seller. Another advantage is that our seller and buyer will be able to meet in person, in a safe and public environment, whether it be on campus or otherwise. We also have a site administrator that guarantees the validity of each item posted and swiftly approves each post or rejects inappropriate posts.

*Table 2.1: Competative Analysis Table*

Specification	PurpleMarket	Amazon	Ebay	Craigslist
Short time to make post	-	+	++	+
Easy to contact seller	++	+	+	++
Seller associated with SFSU	++	-	-	-
Ability to meet seller in-person	++	-	+	+
Propose on-campus meeting location	++	-	-	-

++ = Specification is Outstanding

+ = Specification is Good

- = Specification is Bad

### 2.1.8 System Architecture

Server Host:	Amazon EC2 2vCPU 4GB Ram
Operating System:	Amazon Linux 2
Database:	MySQL v8.0.19
Web Server:	Nginx v1.19
Server-Side Language:	JavaScript/Node.js 10.19.0
Web Framework:	React.js 16.14.0
IDE:	VSCode
Web Analytics:	Google Analytics

### 2.1.9 Team & Roles

Name	Role
Ernesto Diaz	GitHub Master/Front End Dev
Jesus Cervantes Fajardo	Back End Dev
Khushboo Gandhi	Front End Lead
Cody Huang	Back End Lead
Mark Kim	Team Lead/Document Master
Vivian Kuang	Back End Dev
Jiasheng Li	Front End Dev

### 2.1.10 Checklist

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

## 2.2 Milestone 2

# SW Engineering CSC648/848 Spring 2022

*PurpleMarket*

TEAM 8

Name	Role	Email
Mark Kim	Team Lead	mkim22@mail.sfsu.edu
Khushboo Gandhi	Front End Lead	
Cody Huang	Back End Lead	
Ernesto Diaz	GitHub Master	
Jesus Cervantes Fajardo		
Vivian Kuang		
Jiasheng Li		

Milestone 2

April 6, 2022

Milestone/Version	Date
M2V1	03/22/2022
M2V2 (revised per CEO's instructions)	03/30/2022

### **2.2.1 Executive Summary**

PurpleMarket is a web application that will connect students, staff, and faculty from San Francisco State together via a virtual marketplace. In this marketplace, they will be able to communicate with each other, as well as buy and sell products on a user to user basis. There will be no intermediate and no surprises. What one user posts will be verified and will be exactly what is sold to another user. With this in mind we hope to create a safer, open community at San Francisco State, while also providing our users with a more convenient way to shop. The value of PurpleMarket is in its ability to provide an experience custom tailored to people associated with San Francisco State. Our vision is to provide a marketplace that facilitate the ability to sell and purchase the types of items and/or services that are most important to our users.

PurpleMarket plans to be easy to navigate, and easy on the eyes. There isn't any clutter which makes the site much more presentable to our users. Our users are presented with a clean design that will make it much easier for users to navigate our site to find exactly what they are looking for. If they have any trouble at all finding their specific product, they are able to narrow their search in three ways. The first method to finding their desired product is by browsing different categories. By providing categories that our users are most interested in, our users will be able to more easily sell or find the products that they want. The second method will be a simple and easy to use search that would allow the user to type specific titles, names, or descriptions where the user will be presented with the results of the search. Specific to the academic environment, the final method for finding a product is by finding products that are associated with a particular course. Once a user finds what they are looking for they can sign up and create a PurpleMarket account that will allow them to communicate with the user that listed the product directly on our site and the rest is history. So what about listing products? Any verified user who opens an account will be able to list a product. Finally, PurpleMarket will be able to facilitate on-campus meetings between buyer and seller, which simplifies the transaction experience.

Our team of passionate San Francisco State students know the frustrations of navigating online classifieds and the unique needs of the people associated with the school. It is because of this that we are exceptionally equipped to understand the San Francisco State community and provide them with the experience that they desire.

### **2.2.2 Main Data Items and Entities**

***user:*** The users are the people expected to use the web application. This entity will contain all the pertinent information that describes our user like their name, email address, and other identifying information.

1. user\_id: unique user identification
2. user\_fname: user's first name
3. user\_lname: user's last name
4. user\_email: user's email
5. user\_registrationrecord: user registration record/type (e.g. admin, staff, faculty, student, unregistered)
6. user\_password: user's password

**item:** This entity are the items that are being bought, sold, or provided. This entity contains all the information that helps users find what they want. For example, this entity will have a name, a description, price, a category that it belongs to, and photos so that they can see what the item looks like.

1. item\_id: unique item identification
2. item\_created: time that posting was created
3. item\_category: item's category
4. item\_seller\_id: user id of seller
5. item\_price: price of item
6. item\_name: name of item
7. item\_desc: description of item
8. item\_pic: full sized picture of item (path)
9. item\_thumbnail: thumbnail sized picture of item (path)
10. item\_course: course associated with item
11. item\_golive\_time: time of approval

**category:** This is the category a particular item will belong to. For example, items may be furniture, books, electronics, etc. We want this data item to be easily changed or expanded if necessary in case our users find categories that they want.

1. category\_id: unique category identification
2. category\_name: name of category (e.g. furniture, books, etc).

**session:** This will contain session information for users that are on our web application.

1. session\_id: unique session identification
2. session\_user: user associated with session
3. session\_expires: session expiration time
4. session\_data: session data

**message:** This entity exists to contain all the messages that are sent from one user to another. This will

contain not only the message, but also the sender, recipient, and a time stamp of when the message was sent and received/read.

1. msg\_id: unique message identification
2. msg\_sender: sender of message
3. msg\_recipient: recipient of message
4. msg\_item: item associated with message
5. msg\_meet\_time: proposed meeting time
6. msg\_location: proposed meeting location
7. msg\_contactinfo: contact information of sender
8. msg\_body: body of message
9. msg\_timestamp: timestamp of message

*review:* This data item contains the reviews for all users. This entity will contain the review itself, the recipient of the review as well as the person who wrote the review, the time stamp of the review, and also the associated transaction/item that goes with the review.

1. review\_id: unique review identification
2. review\_item: item associated with review
3. review\_recipient: user id of recipient of review
4. review\_contributor: user id of the contributor the review
5. review\_rating: rating of review recipient
6. review\_body: body of review
7. review\_timestamp: time of review

### 2.2.3 Functional Requirements - Prioritized

#### Priority 1

##### Unregistered Users

No.	Requirement
1	Unregistered users shall be able to search the site for certain posts.
2	Unregistered users shall be able to filter search results.
3	Unregistered users shall be able to register for an account.

##### Registered Users

No.	Requirement
6	Registered users shall be able to do everything an unregistered user can do.
7	Registered users shall be able to create posts.
8	Registered users shall be able to contact the poster/seller.
11	Registered users shall be able to login to their account.
12	Registered users shall be able to logout of their account.
13	Registered users shall be able to request/reset their password if they forgot.

### Administrators

No.	Requirement
17	Admins shall be able to do everything a registered user can do.
18	Admins shall be able to delete and edit posts done by any user.
21	Admins shall be required to approve posts before they can go live.
23	Admins shall be able to remove users (added 3/22/22)

### Priority 2

#### Unregistered Users

No.	Requirement
N/A	None

#### Registered Users

No.	Requirement
9	Registered users shall be able to edit and correct and/or delete their own post after it was submitted.

### Administrators

No.	Requirement
19	Admins shall be able to contact any user.
20	Admins shall be able to ban users.

## Priority 3

### Registered Users

No.	Requirement
10	Registered users shall be able to report violations of terms of service of another user.
14	Registered users shall be able to leave reviews of other users regarding completed transactions.
15	Registered users shall be able to propose meeting locations or modify location proposals to complete transactions.
16	Registered users shall be able to inspect their transaction history.

### Administrators

No.	Requirement
22	Admins shall be able to inspect any user's transaction history.

## 2.2.4 User Interface Storyboards

### Administrative Functions: Kobe

As an administrator, Kobe will need to approve or deny posts as needed and provide explanations for approval or denial (Figure 2.1). Before approval, Kobe will inspect each post to ensure that the posting follows the website's terms and conditions (Figure 2.2). In addition, he will be able to contact and ban users, delete and edit user posts, and inspect user transaction history. Priority 1 implementation shall be through MySQL workbench and/or MySQL command line interface. The following user interface storyboards illustrate a priority 3 implementation for administration through the website.

Figure 2.1: Post Management View

PurpleMart

All  Search Logout Admin

CSC648 Spring 2022 Team 8

Admin Dashboard

Post Management

User Feedbacks

Name	Time	Status	Explanation	Details
Jiasheng Li	03/13/2022 08:03	Approved	Your Post Publicized	Details
Khushboo Gandhi	03/12/2022 21:07	Invalid	Invalid Item to Sell	Details
Ernesto Diaz	03/11/2022 10:11	Approved	Your Post Publicized	Details
Mark Kim	03/10/2022 07:21	Pending		Details

Figure 2.2: Post Detail View

PurpleMart

All  Search Logout Admin

CSC648 Spring 2022 Team 8

Admin Dashboard

Post Management

User Feedbacks

**Post Details**

Name :	Jiasheng Li	Explanations
Email :	jli29@mail.sfsu.edu	Descript what you expect.....
Status :	Approved	
Post Time :	03/13/2022 08:03	

Submit

## Purchase SFSU Apparel: Claire

Claire is purchasing SFSU apparel from other users. Using her laptop, she goes onto PurpleMarket and selects “SFSU Apparel” from the categories and browses the available listings (Figure 2.3). Once she finds the swimsuit, she views the item details (Figure 2.4), then attempts to message the seller for a swimsuit. The website then informs her that to message a seller, she must first register for a free account or login (Figure 2.5). Since she is not signed up, she must register (Figure 2.6). Once signed up, she is presented with a message page that allows her to message the seller (Figure 2.7) and the website tells her that her message was successfully sent (Figure 2.8).

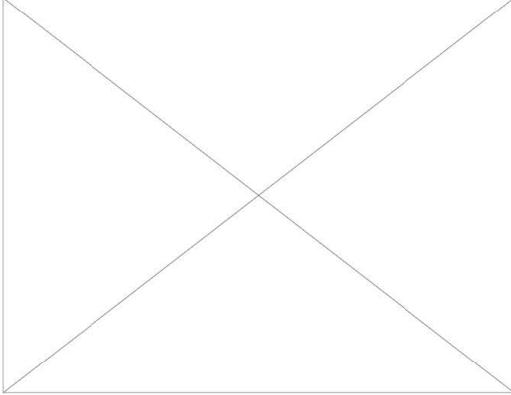
Figure 2.3: SFSU Apparel Category Search

The screenshot shows a web browser window for the PurpleMarket website. At the top, there is a header with the text "CSC648 Spring 2022 Team 08". Below the header, there is a navigation bar with buttons for "PurpleMarket", "SFSU Apparel", "Search", "Post Items", "Login", and "Register". The main content area is titled "Welcome to PurpleMarket" and contains the text "A Market place connects people only associated with SFSU to sell or purchase items.". A search bar at the top of the content area has the text "1-6 of over 20 results for \"operating\"". To the right of the search bar is a dropdown menu labeled "Sort by: post date". The results are displayed in a grid of six items. Each item card has a large red X overlaid on it. The first item card shows the following details: "Title: Operating System Concepts", "Price: \$33.60", "description", and buttons for "Details" and "Message Seller". The second item card shows: "Title", "Price", "description", and buttons for "Details" and "Message Seller". The third item card shows: "Title", "Price", "description", and buttons for "Details" and "Message Seller". The fourth item card shows: "Title: Operating System Design", "Price: \$23.60", "description", and buttons for "Details" and "Message Seller". The fifth item card shows: "Title", "Price", "description", and buttons for "Details" and "Message Seller". The sixth item card shows: "Title", "Price", "description", and buttons for "Details" and "Message Seller". At the bottom of the content area, there is a contact email address: "Contact us: csc648spring2022team08@mail.sfsu.edu".

Figure 2.4: Item Detail View

PurpleMarket

All ·  Search Post Items Login Register



**Title**

**Price**

**Category**

description  
xxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxx

[Message to the Seller](#)

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.5: Login View

PurpleMarket

Books · Operating Search Post Items Login Register

**Start Posting and Buying Your Items**

**Log in to your account**

SFSU Email\* :

Password\* :

[Forgot password?](#)

[Need an Account? Register](#)

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.6: Registration View

CSC648 Spring 2022 Team 08

PurpleMarket

Welcome to PurpleMarket

**Sign up for a free account**

\* Mandatory Fields

First Name\* :

Last Name\* :

SFSU Email\* :

Password\* :

Confirm Password\* :

I acknowledge that I agree to the Term of Use and have read the Privacy Policy.

Already Have an Account? [Login](#)

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.7: Message View

PurpleMarket

**Message Page**

**Item Name :** XXXXXXXXXXXXXXXXXXXX

**Seller Name :** XXXXXXXX

**Meetup Date and Time\***

**Meetup Location\***

**Contact\***

**Addition Information**

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.8: Message Confirmation View

The screenshot shows the PurpleMarket website's message page. At the top, there is a navigation bar with links for 'PurpleMarket', 'All' (in a dropdown menu), 'Search', 'Post Items', 'Logout', and 'Dashboard'. Below the navigation bar, the title 'Message Page' is centered. On the left side, there are input fields for 'Item Name' (containing 'xxxxxxxxxxxxxxxxxxxx'), 'Seller Name' (containing 'xxxxxx'), 'Meetup Date and Time\*' (with 'Date' and 'Time' buttons), and 'Meetup Location\*'. A large gray rectangular box contains the message 'Your Message Was Successfully Sent to the Seller!'. To the right of this message box is a large, empty rectangular area with a thin gray border. At the bottom right of the page is a 'Send' button. At the very bottom center, there is a small line of text: 'Contact us: csc648spring2022team08@mail.sfsu.edu'

### Request Items: Anna

Anna wants to get a copy of notes from her class from a previous student of hers. She goes to PurpleMarket to search for notes from her course (Figure 2.9). After finding a student posting some notes that he took from Anna's class last semester (Figure 2.10), Anna messages him to get a copy. When she attempts to contact him, the website prompts her to login or register (Figure 2.11). Since she is already registered, she logs in and messages the student to set up a meeting on campus (Figures 2.12, 2.13).

Figure 2.9: Search for Notes



## Welcome to PurpleMarket

A Market place connects people only associated with SFSU to sell or purchase items.

This screenshot shows the search results for 'operating' notes. At the top, it says '1-6 of over 20 results for "operating"'. There are six items listed in a grid. The first item is 'HIST 471 notes' for \$33.60. The second item is 'Operating System Design' for \$23.60. Each item has a 'Title', 'Price', 'description', 'Details' button, and 'Message Seller' button. A vertical scroll bar is on the right. At the bottom, there's a contact email: 'Contact us: csc648spring2022team08@mail.sfsu.edu'

Figure 2.10: Item Detail View



This screenshot shows the detail view for an item. The title is 'Title' (crossed out). The price is 'Price' (crossed out). The category is 'Category' (crossed out). The description is 'description' followed by several 'XXXXXXXXXXXXXX' placeholder characters. A 'Message to the Seller' button is at the bottom. At the bottom of the page, there's a contact email: 'Contact us: csc648spring2022team08@mail.sfsu.edu'

Figure 2.11: Login View

PurpleMarket

Books Operating Search Post Items Login Register

**Start Posting and Buying Your Items**

**Log in to your account**

SFSU Email\* :

Password\* :

[Forgot password?](#)

**Log In**

Need an Account? [Register](#)

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.12: Message View

PurpleMarket

All Search Post Items Logout Dashboard

**Message Page**

**Item Name :** XXXXXXXXXXXXXXXXXXXXXXX

**Seller Name :** XXXXXXXX

**Meetup Date and Time\***

**Meetup Location\***

**Contact\***

**Addition Information**

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.13: Message Confirmation View

The screenshot shows the 'Message Page' of the PurpleMarket website. At the top, there is a navigation bar with links for 'PurpleMarket', 'All', 'Search', 'Post Items', 'Logout', and 'Dashboard'. Below the navigation bar, the page title 'Message Page' is centered. The main form area contains fields for 'Item Name' (xxxxxx), 'Seller Name' (xxxxxx), 'Meetup Date and Time\*' (with 'Date' and 'Time' buttons), and 'Meetup Location\*'. A large gray callout box displays the message 'Your Message Was Successfully Sent to the Seller!'. Below the form, there is a section labeled 'Addition Information' with a text input field and a 'Send' button. At the bottom of the page, a contact email is listed: *Contact us: csc648spring2022team08@mail.sfsu.edu*.

### Browsing for Items: Curry

Curry is just curious about what used items he can score on PurpleMarket, so he visits the website to browse around. While browsing, he remembers that there is a book he's been wanting to read. He does a search on the page for the book title that he's looking for (Figure 2.14). He checks the details of the item post to make sure it is exactly what he wants (Figure 2.15). As a long-time website user, he is already logged in, so he immediately messages the user to set up a time and location for the purchase (Figures 2.16, 2.17).

Figure 2.14: Item Detail View

CSC648 Spring 2022 Team 08

PurpleMarket Books Operating Search Post Items Login Register

## Welcome to PurpleMarket

A Market place connects people only associated with SFSU to sell or purchase items.

1-6 of over 20 results for "operating"

Sort by: post date

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.15: Login View

PurpleMarket All Search Post Items Login Register

Title

Price

Category

description  
xxxxxxxxxxxxxxxxxxxxxx  
xxxxxxxxxxxxxxxxxxxxxx

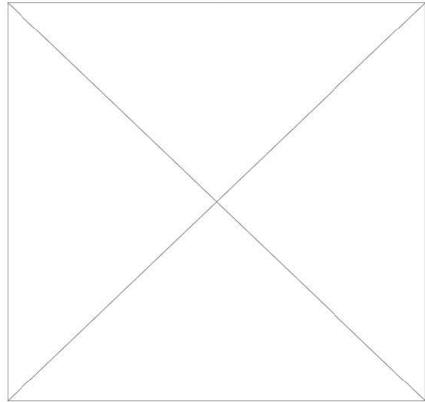
Message to the Seller

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.16: Message View

PurpleMarket

### Message Page

Item Name : xxxxxxxxxxxxxxxxxxxxxxxx  
Seller Name : xxxxxxxxx  
Meetup Date and Time\*    
Meetup Location\*   
Contact\*   
Addition Information   
  


Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.17: Message Confirmation View

PurpleMarket

### Message Page

Item Name : xxxxxxxxxxxxxxxxxxxxxxxx  
Seller Name : xxxxxxxxx  
Meetup Date and Time\*    
Meetup Location\*   
  
  

Your Message Was Successfully Sent to the Seller!

  
Contact\*   
Addition Information

Contact us: csc648spring2022team08@mail.sfsu.edu

## Selling Furniture: Anna

Anna has a bookshelf to sell, so she visits PurpleMarket to list the item. From the main homepage, she easily finds the button to post her item for sale (Figure 2.18). Once on the post page (Figure 2.19), she enters the details of her bookshelf and uploads a photo. When she attempts to post, the site requests that she log in (Figure 2.20). Once logged in, she can post her item. The website tells her that she will need to wait about 24 hrs for her post to be approved and go live (Figure 2.21). She receives a message when it is approved and published. She goes to her dashboard to look at the approval message (Figure 2.22).

Figure 2.18: Search for Notes

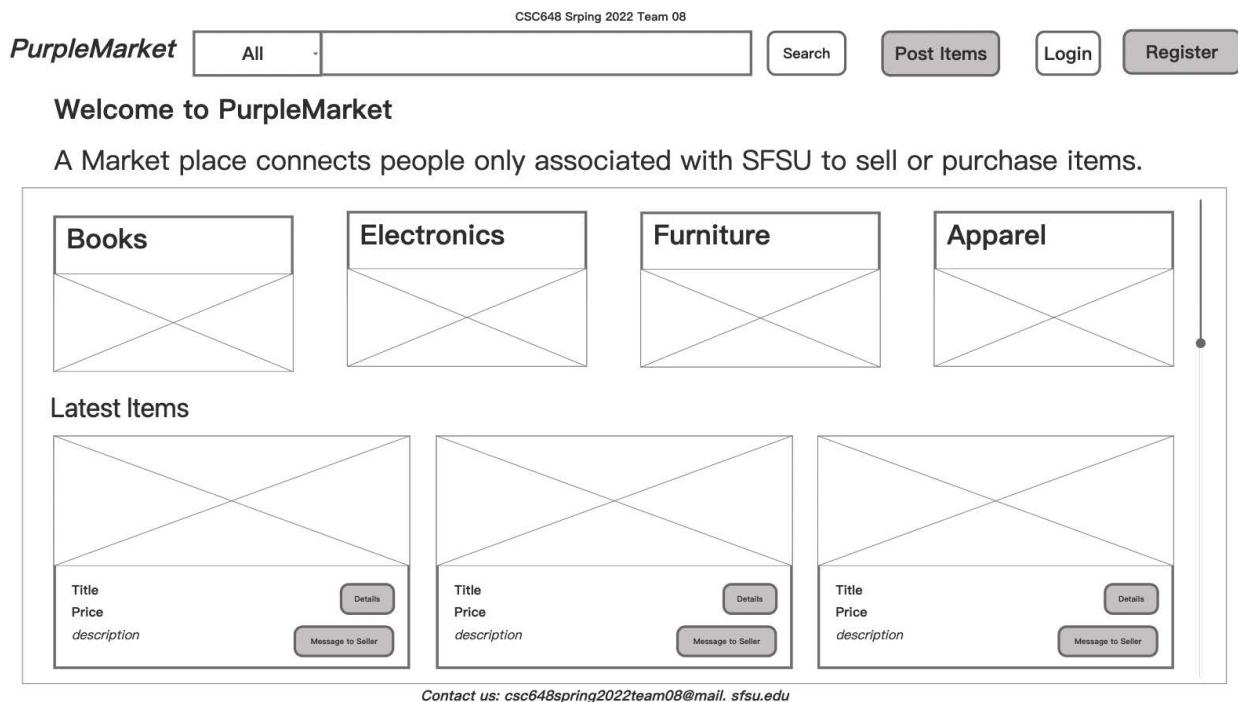


Figure 2.19: Item Detail View

CSC648 Spring 2022 Team 08

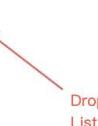
PurpleMarket

### Item Post Page

**Posting Your Items**

**Item name\***

**Price\***

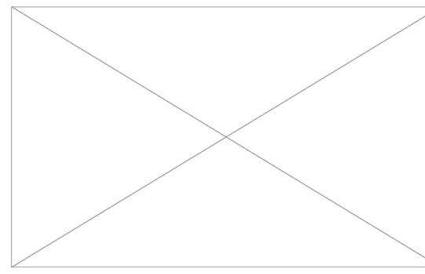
**Category\***  

**Course Number\***  

\* MANDATORY FIELDS

**Description :**

I accept the terms of use and privacy policy



Contact us: csc648spring2022team08@mail. sfsu.edu

Figure 2.20: Login View

PurpleMarket

### Start Posting and Buying Your Items

#### Log in to your account

**SFSU Email\* :**

**Password\* :**

[Forgot password?](#)

[Need an Account? Register](#)

Contact us: csc648spring2022team08@mail. sfsu.edu

Figure 2.21: Message View

PurpleMarket All Search Post Items logout Dashboard

## Item Post Page

**Posting Your Items**

**Item name\***

**Price\***

**Category\***

**Upload image :**  **No file chosen**

**Descriptions :**

I accept the terms of use and privacy policy

**Post**

Your Post Was Uploaded Successfully.  
 Please Wait for 24 hours to check Your Post in Your Dashboard!

Contact us: csc648spring2022team08@mail.sfsu.edu

Figure 2.22: Message Confirmation View

CSC648 Spring 2022 Team 08

PurpleMarket All Search Post Items Logout Dashboard

**MyPage** Username: Jiasheng Li Email: jli29@mail.sfsu.edu

Post History Messages Account Details

### Message Page

User Name	Meet Time	Location	Additional Info	Contact
Jiasheng Li	03/13/2022 08:03	HSS	xxxxxxxxxx	415-706-3250
Khushboo Gandhi	03/12/2022 21:07	J. Paul Library	xxxxxxxxxx	jli29@mail.sfsu.edu
Ernesto Diaz	03/11/2022 10:11	Fine Art	xxxxxxxxxx	415-706-3250
Mark Kim	03/10/2022 07:21	Business	xxxxxxxxxx	415-706-3250

Contact us: csc648spring2022team08@mail.sfsu.edu

## 2.2.5 Architecture

### DB Organization

#### user table

1. user\_id: unique user identification
2. user\_username: user's username
3. user\_fname: user's first name
4. user\_lname: user's last name
5. user\_email: user's email
6. user\_registrationrecord: user type (e.g. admin, staff, faculty, student, unregistered)
7. user\_password: user's password

#### item

1. item\_id: unique item identification
2. item\_created: time that posting was created
3. item\_category: item's category
4. item\_seller\_id: user id of seller
5. item\_price: price of item
6. item\_name: name of item
7. item\_desc: description of item
8. item\_pic: picture of item (path)
9. item\_thumbnail: thumbnail of item pic (path)
10. item\_course: course associated with item
11. item\_golive\_time: time of approval

#### category

1. category\_id: unique category identification
2. category\_name: name of category

#### session

1. session\_id: unique session identification
2. session\_user: user associated with session
3. session\_expires: session expiration time
4. session\_data: session data

## **message**

1. msg\_id: unique message identification
2. msg\_sender: sender of message
3. msg\_recipient: recipient of message
4. msg\_item: item associated with message
5. msg\_meet\_time: proposed meeting time
6. msg\_location: proposed meeting location
7. msg\_contactinfo: contact information of sender
8. msg\_body: body of message
9. msg\_timestamp: timestamp of message

## **review**

1. review\_id: unique review identification
2. review\_item: item associated with review
3. review\_recipient: user id of recipient of review
4. review\_contributor: user id of the contributor the review
5. review\_rating: rating of review recipient
6. review\_body: body of review
7. review\_timestamp: time of review

## **Media Storage**

User uploaded images will be stored in an AWS S3 bucket. The S3 bucket will give read-only access to the public. We want users who haven't created accounts yet to be able to view some posts with their respective images while securing write permissions to registered users only. We will use the S3 AWS SDK API for certified users who created accounts to upload images to the S3 bucket when creating a post.

## **Search/filter architecture and implementation**

1. Database queries shall be utilized for all search items for the user.
2. Item names and descriptions shall be utilized for keyword searches.
3. Item categories shall be browsable and searchable.
4. Course materials shall be searchable by course name.
5. Search results shall be sortable by price, post age, category, item name, or seller username.

## **Team 8 APIs**

None

## **Algorithms or Processing**

None

## **SW Tool and Framework Changes**

S3 storage for image storage. (seeking approval)

### **2.2.6 Key Risks**

#### **Skills**

#### **Risks**

- Most of the team has very little experience in deploying a project of this scale on the cloud. Local development environments may be quite different from the cloud production environment, which requires a set of skills that have not been practiced thoroughly.
- The skill level for various frameworks is varied from team member to team member.

#### **Proposed Solution**

- Reduce product scope to viable minimum.
- Schedule substantial time for research and trial-and-error.
- Promote collaboration between team members so that the entire team can benefit from each member's strengths.

#### **Schedule**

#### **Risks**

- The scope of the project is substantial compared to the limited time we have. This is a risk in the sense it is not certain whether we are able to successfully complete the entire website as required.
- With varied course loads between students, each student is available at different times throughout each week.

## **Proposed Solution**

- We may reduce the scope of the project and focus on completing highest priority items first.
- Divide the work amongst team members as much as possible and delegate according to each students' needs.
- Create deadlines with contingency plans in place to ensure that work gets completed on-time.

## **Technical**

### **Risks**

- Due to the scale of the project and the amount of new technologies involved there is a chance that as a team we may not be able to accomplish something successfully due to lack of expertise.
- Breaking of systems due to lack of expertise and/or mistakes.

## **Proposed Solution**

- Use simple designs.
- We plan to follow the method of practice and learn until we succeed!
- Use tutorials online and follow articles and forums which will help us accomplish our desired output.
- Try to simplify our project as much as possible to reduce the need for too many new technologies.
- Maintain strict controls over each branch of code and the server to reduce the chance of mistakes.

## **Teamwork**

### **Risks**

- There is a chance of conflict amongst team members where some members might think they do more work than others.

## **Proposed Solution**

- Use project management tools like Trello to track tasks amongst team members for efficient planning.
- Ensure that all team leads divide work efficiently.

## **Legal/Content**

None at the moment.

## **2.2.7 Project Management**

The team is divided into two main categories: front-end and back-end. Each category has been assigned a lead, with two developers assigned to each lead. The team lead is responsible for tracking and maintaining project progress and establishing high-level goals while contributing to either the front-end or back-end as needed. With the input of the team, the team lead will set deadlines for deliverables and work with the front-end and back-end leads to create a plan of action to complete objectives. The front-end and back-end leads are responsible for assigning tasks for their particular portion of the project while managing their particular deliverables. All tasks are tracked via shared Trello boards and progress (or problems) will be communicated via Discord. In the case that either the front-end or back-end teams encounter setbacks, the team lead will coordinate with all members to redistribute the workload. Our number one priority is maintaining an active line of communication between all stakeholders to ensure that the back-end provides the tools necessary for the front-end to deliver a great user experience. 25

## 2.3 Milestone 3

# SW Engineering CSC648/848 Spring 2022

*PurpleMarket*

TEAM 8

Name	Role	Email
Mark Kim	Team Lead	mkim22@mail.sfsu.edu
Khushboo Gandhi	Front End Lead	
Cody Huang	Back End Lead	
Ernesto Diaz	GitHub Master	
Jesus Cervantes Fajardo		
Vivian Kuang		
Jiasheng Li		

Milestone 3

April 23, 2022

Milestone/Version	Date
M3V1	04/23/2022

Summary of Milestone 3 ZOOM meeting review with Prof. Petkovic and plans for further development  
Team Number: 8  
Meeting Date: 4-13-22

### **2.3.1 Summary of feedback on UI**

(list of items, note that details are available in ZOOM recording)

- Search button should be right of text input field
- Descriptive text summarizing website purpose below nav bar
- Explanation of what products are being shown
- Logo clickable to lead to home page
- Text input should be persistent
- Text input should not allow more than 40 alphanumeric characters w/ error message
- Message seller button should have same look as rest of page
- Post page: all entries should be in one column; cancel button should be left of post; there should be a note that approval may take up to 24hrs
- MyPage: Dates column for posts and messages
- Login: needs “forgot password?” link
- Register: need to have a note indicating “\* - mandatory fields”

### **2.3.2 Summary of feedback on code and architecture**

- Code documentation is lacking (headers missing, inline code commenting insufficient)
- Variable naming looks okay
- Architecture looks okay, but remote server is not displaying css styling correctly

### **2.3.3 Summary of feedback on github usage**

- Commit messages are horrible. They are not descriptive enough. Future commit messages should be descriptive and provide enough detail to understand the steps done in commit. Possibly replace old commit messages to fix issues.

### **2.3.4 Summary of feedback on DB**

- DB usage looks good for the most part.

- User messages should have an “approved” flag to determine its status
- Using a timestamp for message approval creates unnecessary complexity: BAD IDEA
- User role should be shown as a registration record NOT as a role.

### **2.3.5 Summary of feedback on teamwork**

- N/A: Responded to questions about the teamwork of the group: teamwork is pretty good by our estimation
- Anthony gave feedback about being concerned about number of commits being unbalanced (after the checkpoint); responded with possible reasons for this (e.g. problems with deployment causing a large number of commits by one person) Any other comments and issues:
  - Major disappointment for not completing tasks as required.
  - Team was completely unprepared which indicated a lack of professionalism

**Checkpoint, DUE: 4-18-22**

### **2.3.6 P1 feature list**

#### **Unregistered Users**

1. Unregistered users shall be able to search the site for certain posts.
2. Unregistered users shall be able to filter search results.
3. Unregistered users shall be able to register for an account.

#### **Registered Users**

6. Registered users shall be able to do everything an unregistered user can do.
7. Registered users shall be able to create posts.
8. Registered users shall be able to contact the poster/seller.
11. Registered users shall be able to login to their account.
12. Registered users shall be able to logout of their account.
13. Registered users shall be able to request/reset their password if they forgot.

#### **Administrators (ALL in MySQL Workshop)**

17. Admins shall be able to do everything a registered user can do.

18. Admins shall be able to delete and edit posts done by any user.
21. Admins shall be required to approve posts before they can go live.
23. Admins shall be able to remove users (added 3/22/22)

### **2.3.7 Plan of Action**

First of all, all hands will be on deck to complete all deficiencies for the checkpoint as given in feedback. Front-end lead shall determine front-end workload and notify team lead of any extra resources necessary to complete UI tasks. Once the workload is determined, the Front-end lead will delegate tasks as necessary to their team. Team lead will work with Back-end lead to divert back-end resources to the front-end as necessary. Back-end lead will allocate back-end team members to complete code commenting and provide routes as needed to the front-end team.

Once the checkpoint is completed, we will prioritize getting all P1 requirements working within the constraints of the non-functional requirements. Over the course of the project, the team lead will do spot checks on code with full code-review to be completed by front-end and back-end lead.

Scheduling and assignment of specific tasks will be tracked via Trello and a shared Google Doc of the feedback given from both the class CEO and CTO. ONLY P1 features will be implemented. Number 1 priority (after checkpoint) is getting all front-end pages connected with the back end. This will require the back-end to communicate with the front-end to provide routes for interfacing with the database. Once front-end requirements are provided, the back-end team will implement models and routes and test both before providing the routes and specifications to the front-end team. Then the front-end team will implement connection to the back-end API and provide feedback for any necessary changes. Once all connections are made and work as per front-end specifications, the front-end will complete all the UI elements as needed to allow for registration, login, and mypage data. The back-end team will then comment code (if missing) and provide support for front-end development. The back-end team will also focus on documentation and code-review of the front-end.

## 2.4 Milestone 4

# SW Engineering CSC648/848 Spring 2022

*PurpleMarket*

TEAM 8

Name	Role	Email
Mark Kim	Team Lead	mkim22@mail.sfsu.edu
Khushboo Gandhi	Front End Lead	
Cody Huang	Back End Lead	whuang17@mail.sfsu.edu
Ernesto Diaz	GitHub Master	ediaz8@mail.sfsu.edu
Jesus Cervantes Fajardo		
Vivian Kuang		
Jiasheng Li		

Milestone 4

May 15, 2022

Milestone/Version	Date
M4V1	03/15/2022

#### **2.4.1 Product Summary**

The name of our product is Purple Market. It is a web application that connects students, staff, and faculty from San Francisco State University. It is a marketplace where students and staff can buy and sell products to others at SFSU. We create a safe way to shop by manually verifying each post before publishing it for sale. We provide a custom tailored experience for people associated with SFSU. Our website is simple to navigate and easy to use. There is no clutter with our clean design. We sort all of our products into general and easy to understand categories. Users are able to search for the items they want in the search bar or filter out the ones they are looking for by categories. Once a user finds their product they can sign up on our site and send the seller a message.

**Our application allows users to:**

- Search the site for certain posts.
- Filter search results.
- Register for an account

**Once registered, users will be able to:**

- Create posts.
- Get in contact with poster/seller
- Edit and correct and/or delete their own post.
- Login/out of their account.
- Reset their password.
- Inspect their transaction history.

**Finally admins will use MySQL workbench to:**

- Do everything a registered user does.
- Approve or decline an item post.

**Website url:** <http://csc648.mskim.dev>

## **2.4.2 Usability Test Plan - max 2 pages**

*Function: Post an Item*

### **Test Objective**

Posting is a priority one functional requirement of the app and related to one key user case of the app, so it is necessary to test if users can easily post an item and try to identify problems with usability with the app. Testing allows developers to revise, iterate and retest the app based on the analytical results and the report from the test plan.

### **Test background and setup:**

The system is running on an Amazon EC2 instance. Usability testers can use Chrome and Safari for browsers to participate in this usability test. The starting point of this usability test is the home page with the user already logged into the system. Then the testers are to navigate to the post item page and proceed to post an item on the system.

The intended users of the system are students, staff, and faculty of San Francisco State University. The URL of the system to be testing is as follows: <https://csc648.mskim.dev/createpost>

The first thing to be measured is the task completion success rate which is the percentage of users who can complete the usability task in the defined time. The second thing to be measured is the number of errors for completing the usability task which are to be reported by the testers. The third thing to be measured is the average time it takes users to complete posting an item. The fourth thing to be measured is the average number of clicks and average number of screens to complete posting an item. Finally, the usability test will implement a Likert questionnaire to measure their comfort and confidence at posting an item on the system.

### **Usability Task Description:**

As a usability tester, you will need to follow the requirements of system setup before entering the start point. Once logged in, and on the home page, please attempt to post an item to the system. You will also be the test monitor, so please record the number of screens you visit, the number of clicks you use, any errors you encounter, and the time it takes to complete the entire process of posting an item (if budget and/or time allows, this data will be gathered from action recording software on the system that the tester uses). After that, the testers will finish an evaluation of efficiency form and an evaluation of user satisfaction form.

Task	Description
Task	Post an item on PurpleMarket
Machine State	The prioritized functionalities of PurpleMarket have been implemented
Successful Completion criteria	The post is loaded on the user's post history of PurpleMarket
Benchmark	Completed in 1 minute

### Evaluation of Effectiveness

We measure effectiveness of posting by measuring the percentage of usability testers who completed the posting task one minute. Also, we measure the number of errors incurred during their task completions.

Test/Use case	% Completed	Errors	Comments
Post	0%	One error receiving an error after clicking a post item button	The function for sending the posting data from front-end to back-end fails, so all testers could not post an item successfully.

### Evaluation of Efficiency

We measure efficiency in time which includes average time of those who complete the task, efficiency in effort which is the number of clicks, and efficiency in content which include the number of screens and number of pages of instructions to complete posting.

Situation two: usability user does not have an account

1. Average time of those tester who complete the task: 100 seconds
2. Number of screens: 7 (Homepage → Post Items → Login → Register → Login → Homepage  
→ Post Items)
3. Number of clicks : 30 clicks  
(Post Items → Item Name → Price → Category → Choose File → Select File → Select OK → Description → Post Item → Select OK for error → Login → Register here → First Name → Email → Password → Confirm Password → Show Password → Register → Email → Password → Login → Post Items → Item Name → Price → Category → Choose File → Select File → Select OK → Description → Post Item → Select OK on the error (Got a 500 error and not the 24 hr approval message))
4. Number of pages of instructions: 2 (register page & posting page)

Situation one: usability user has an account

1. Average time of those tester who complete the task: 42 seconds
2. Number of screens: 2 (Homepage → Post Items)
3. Number of clicks : 14 clicks

(Login → Email → Password → Login → Post Items → Item Name → Price → Category → Choose File → Select File → Select OK → Description → Post Item → Select OK on the error (Got a 500 error and not the 24 hr approval message))

4. Number of pages of instructions: 1 (posting page)

#### Evaluation of User Satisfaction

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I found the website easy to use.	2				
It is easy to post an item.	2				
It is easy to know when posting an item is successful.			3		

#### 2.4.3 QA Test Plan - max 2 pages

*Function: Post an Item*

#### Test Objective

Test whether our post function is usable. This means that the user is able to create a post through our post page by inserting certain information within the data fields. Testing shall also show whether the process contains any bugs. The function shall also be tested to see if it works as per specs, and that the output of the tests match the expected output defined in the following defined test cases.

#### Hardware Used

#### System Architecture

- Server Host: Amazon EC2 2vCPU 4GB RAM

- Amazon Linux 2 Operating System
- MySQL v8.0.19 Database
- Nginx v1.19 Web Server
- Web Analytics: Google Analytics

## User System Specs

1. PC specs
  - Intel i7 11700KF
  - 16GB RAM
  - Nvidia 3080 10gb graphics card
  - ASRock motherboard
  - 500 gb SSD
2. MacBook specs
  - Intel Core I5
  - 16 GB RAM
  - 512 GB SSD

## Software Used

1. PC software (Test Table 2.2)
  - Windows 11 Operating System
  - Chrome Web browsers
  - Link to post page: <https://csc648.mskim.dev/createpost>
2. Mac software
  - macOS Monterey v12.3 (Test Table 2.3)
  - Safari Web Browser
  - Link to post page: <https://csc648.mskim.dev/createpost>

## Feature to be tested

Our feature to be tested is our post function. As stated above, this function shall allow a user to create a new post within our website by inserting a description, price, image, and other information into the data fields listed within our post page.

*Table 2.2: Test cases with Windows 11 and chrome web browser*

Test #	Test Title	Description	Input	Expected Output	Result
1	Posting an Item	Create a new post.	Click on Post Items. Input for form fields: Macbook Air, 950, Electronics, Picture of macbook (item to be posted), Selling a used macbook air.	Message “Your post has been uploaded. It will take upto 24 hours for the post to be reviewed and approved.” received when the button “Post Item” is clicked	PASSED
2	Check if data is recorded in the database	Check whether the data used is correctly inserted into our database.	After clicking “Post Item” and being approved, open MySQL Workbench and run the following query: “SELECT * from csc648.item;”	Check that the following is recorded in the database: item_id = 14, item_seller_id = 10, item_category = 4, item_name = Macbook Air	PASSED
3	Check item is not live	Check if the item is posted on the home page	Open Chrome web browser and input “https://csc648.mskim.dev/” into the search bar. Find the posted item	Item posted is the top left item, since it is the latest item	PASSED

*Table 2.3: Test cases with macOS and Safari web browser*

Test #	Test Title	Description	Input	Expected Output	Result
1	Posting an item	Create a new post.	Click on Post Items. Input for form fields: Futon Sofa, 200, Furniture, Picture of futon sofa (item to be posted), Moving out soon and need to get rid of this sofa.	Message received when the button “Post Item” is clicked	FAILED
2	Check if data is recorded in the database	Check whether the data used is correctly inserted into our database.	After clicking “Post Item” and being approved, open MySQL Workbench and run the following query: “SELECT * from csc648.item;”	Check that the following is recorded in the database: item_id = 12, item_seller_id = 9, item_category = 4, item_name = Macbook Air	FAILED
3	Check item is not live	Check that the item is not posted on the home page	Open Safari web browser and input “https://csc648.mskim.dev/” into the search bar. Find the posted item	Item posted is the top left item, since it is the latest item	PASSED

## 2.4.4 Code Review

### Email Screenshots

Re: CSC648 Team 8 Backend Code Review

**From:** Vivian Kuang <vkuang1@mail.sfsu.edu>  
**Sent:** Friday, May 13, 2022 9:47 PM  
**To:** Khushboo Gandhi <kgandhi1@mail.sfsu.edu>  
**Cc:** Jesus Cervantes Fajardo <jcervantesfajardo@mail.sfsu.edu>  
**Subject:** CSC648 Team 8 Backend Code Review

Hello Khushboo,

Not sure if I'm doing this email correctly.  
For code review of backend for files that relate to posting an item.

In order to post an item

1. A user has to login (not sure we have to review the login)  
backend code is in:  
application/server/index.js  
application/server/routes/login.js  
application/server/models/login.js
2. A user posts an item  
backend code is in:  
application/server/index.js  
application/server/routes/posting.js  
application/server/models/post.js

Thanks,  
Vivian

Re: CSC648 Team 8 Backend Code Review

Thanks,  
Vivian

---

**From:** Khushboo Gandhi <kgandhi1@mail.sfsu.edu>  
**Sent:** Friday, May 13, 2022 11:41 PM  
**To:** Vivian Kuang <vkuang1@mail.sfsu.edu>; Jesus Cervantes Fajardo <jcervantesfajardo@mail.sfsu.edu>  
**Subject:** Re: CSC648 Team 8 Backend Code Review

Hello!  
I just realised I have to inform you about the files too.

Posting an item!  
application/web/src/components/itemPost

The QA mentions about deleting and updating post but our website does not allow us to do that lol.  
so idk what we are supposed to do there. We will ask mark tomorrow.

I am not fully sure about this mail either. Just let me know if you want the locations of some specific pages.

Thanks,  
Khushboo

Re: CSC648 Team 8 Backend Code Review

**From:** Vivian Kuang <vkuang1@mail.sfsu.edu>  
**Sent:** Saturday, May 14, 2022 2:43 PM  
**To:** Khushboo Gandhi <kgandhi1@mail.sfsu.edu>; Jesus Cervantes Fajardo <jcervantesfajardo@mail.sfsu.edu>  
**Subject:** Re: CSC648 Team 8 Backend Code Review

Hello,

Here's my summary:

Yes, there is a header for the code maybe it could be more descriptive.  
Yes, there are inline comments for the code maybe there should be an inline comment for the main functions.  
The functions are named appropriately so that someone would know what it does without the need of a comment.  
The code is also organized well in the order of imports, functions/variables, then html.  
Yes, there are proper and consistent class/methods/variable names.  
No, there is no consistency with naming established in the DATA section of Milestone 2 but I don't think that's for frontend. Those naming is more for backend MySQL.  
The commit comments are okay.

And a link to doc I commented on. I copy and paste the code onto a Google Doc.

<https://docs.google.com/document/d/11YBckz1rCBk1pQMagqcsFix0j1GesNv-BaH7LHs-E/edit?usp=sharing>

### Frontend Code Review

```
// HEADER:Create A Post Code import React, { useState, useEffect } from "react"; import { Row, Col, Container, Button, Dropdown, ButtonGroup, Form } from "react-bootstrap"; import styles from './index.module.css'; import image from '../images/image.png'; import axios from "axios" ...
```

Re: CSC648 Team 8 Backend Code Review

 Kg  
To: Vivian Kuang; Jesus Cervantes Fajardo

🔗 ↲ ↳ ➔ ⋮  
Sat 5/14/2022 7:21 PM

Hello,

Here's my summary for the backend code review.

The pages which I am reviewing are linked here:

<https://github.com/CSC-648-SFSU/csc648-03-sp22-team8/blob/devm3-route-testing/application/server/routes/posting.js>

<https://github.com/CSC-648-SFSU/csc648-03-sp22-team8/blob/devm3-route-testing/application/server/models/post.js>

Summary:

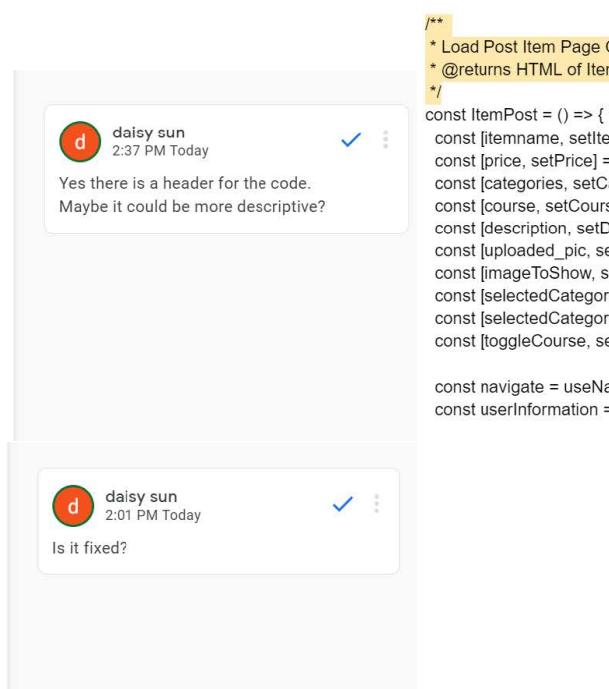
There is a header present for the code that provides a clear description of what it is.  
There are proper and consistent class names, variable names are well defined, and it does what is needed without comments.  
A bit more description on what the storage function is doing and how multer works. That part seems confusing to understand.  
The code is also organized in terms of functions, variables and the SQL functions in the model is also easy to understand.  
There is consistency with the class names from the DATA section of the milestone 2.  
Git commits are good with explanation.

Thanks,  
Khushboo

## Backend reviewing Frontend Code screenshots

```
// HEADER: Create A Post Code
import React, { useState, useEffect } from "react";
import {
  Row,
  Col,
  Container,
  Button,
  Dropdown,
  ButtonGroup,
  Form
} from "react-bootstrap";
import styles from "./index.module.css";
import image from "../../images/image.png";
import axios from "axios";
import { useNavigate } from "react-router-dom";

//TODO: Fix this
const checkValidity = () => {
  console.log("In validity");
  if (
    uploaded_pic === null ||
    price === "" ||
    itemname === "" ||
    description === "" ||
    selectedCategoryId === null
  ) {
```



## Backend review of Frontend summary

- Yes, there is a header for the code, maybe it could be more descriptive.
- Yes, there are inline comments for the code, maybe there should be an inline comment for the main functions.
- The functions are named appropriately so that someone would know what it does without the need of a comment.
- The code is also organized well in the order of imports, functions/variables, then html.
- Yes, there are proper and consistent class/methods/variable names.
- No, there is no consistency with naming established in the DATA section of Milestone 2 but I don't think that's for frontend. Those naming is more for backend mySQL.
- The commit comments are okay.

## Frontend reviewing Backend Code screenshots

```
/*
 * This is the post route.
 * Data incoming from the form will be validated and sent to the database item table
 */

const express = require("express");
const router = express.Router();
const Validator = require("../validator/loginValidation");
const PostModel = require("../models/post");
const multer = require("multer");
const PostError = require("../error/userError");
const sharp = require("sharp");
const crypto = require("crypto");

const storage = multer.diskStorage({
  destination: function (req, file, cb) {
    cb(null, "./user_images");
  },
  filename: function (req, file, cb) {
    let fileExt = file.mimetype.split("/")[1];
    let randomName = crypto.randomBytes(22).toString("hex");
    cb(null, `${randomName}.${fileExt}`);
  }
});

var upload = multer({ storage: storage });

/**
 * Post Router Test
 */
router.get("/", (req, res) => {
  res.send("Successful Posting route response");
  /*
   * Redirect to Home Page upon successful post
   */
  .then((postLogged) => {
    if (postLogged) {
      //redirect somewhere
      console.log("is post in database?: " + postLogged);
      //res.send('/'); // FOR DEPLOYMENT
      res.send("Item posted successfully");
    } else {
      throw new PostError("Unable to put post data into db", "/post", 200);
    }
  })
  .catch((err) => {
    //do something here
    console.log(err);
  });
});

module.exports = router;
```

Khushboo Gandhi 12:06 AM Today Yes there's a header for the code

Khushboo Gandhi 12:06 AM Today the storage code seem's abrupt and out of place

Khushboo Gandhi 12:07 AM Today Where?

Khushboo Gandhi 12:08 AM Today Some comments can be a bit more descriptive

## Frontend review of Backend summary

- There is a header present for the code that provides a clear description of what it is.
- There are proper and consistent class names, variable names are well defined, and it does what is needed without comments.
- A bit more description on what the storage function is doing and how multer works. That part seems confusing to understand.
- The code is also organized in terms of functions, variables and the SQL functions in the model is also easy to understand.

- There is consistency with the class names from the DATA section of the milestone 2.
- Git commits are good with explanation.

#### 2.4.5 Self-check on best practices for security – 1/2 page

Asset to be protected	Types of possible/expected attacks	Your strategy to mitigate/protect the asset
Passwords of Users	Hackers can brute force attack and hash collision to try to solve the password. Also attacks can be done through phishing	We're using Bcrypt to hash the passwords. Bcrypt can slow down the attackers and is adaptive
Messages of Users	Phishing, SQL code injection.	Message route only made available to registered users; unregistered users cannot message. When you login you only have access to your messages.
Posts of Users	Phishing, SQL code injection	Post route only made available to registered users; unregistered users cannot post or see their post history until they login or register. Keep your login information secured in your notes.
Input Data Validation: Search Input limit (40 char)	Code injection	Show the error in the frontend when you try to search over 40 characters. The 40 character limit reduces the number of characters available for code injection.
Input Data Validation: Registration Page	They can do SQL injection to get information. If user input is incorrectly filtered, SQL statements can be executed by the application.	Show the errors in the frontend when you incorrectly put the password, email, name. Validation prevents code from being injected into the form
Input Data Validation: Login Page	They can do SQL injection to get information. If user input is incorrectly filtered, SQL statements can be executed by the application.	Show the errors in the frontend when you incorrectly put the password, email, name. Validation prevents code from being injected into the form

## 2.4.6 Self-check of the adherence to original Non-functional specs

performed by Team Leads

1.	Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0	DONE
2.	Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers (Safari & Chrome)	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	DONE
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	ON TRACK
9.	Application should follow established architecture patterns	ON TRACK
10.	Application code and its repository shall be easy to inspect and maintain	ON TRACK
11.	Google analytics shall be used	DONE
12.	No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application	DONE
13.	Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.	DONE
14.	Site security: basic best practices shall be applied (as covered in the class) for main data items	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	ON TRACK
17.	The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, 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

## 3 Product Screenshots

### 3.1 Home Page

The screenshot shows the main landing page of the Purple Market website. At the top, there is a purple header bar with the text "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only." and "A Market place that connects to people only associated with SFSU to sell or purchase items". Below the header, the title "Purple Market" is displayed. On the right side of the header, there are buttons for "Post Items", "Login", and "Register". The main content area has a white background and features a grid of six items for sale:

- Stapler**: Price: 5.00. Description: It's my stapler.
- Snowboard**: Price: 500.00. Description: Strap this popsicle stick on your ...
- The Constitution and the Nation:...**: Price: 10.00. Description: History Textbook
- The Constitution and the Nation:...**: Price: 10.00. Description: History Textbook
- Brown v. Board of Education**: A BRIEF HISTORY WITH DOCUMENTS, SECOND EDITION
- Headphones**: A pair of black over-ear headphones.

Below the grid, there are two more book covers from the "The Bedford Series in History and Culture" series: "The Constitution and the Nation: The Regulatory State, 1850-1945" and "Brown v. Board of Education: A BRIEF HISTORY WITH DOCUMENTS, SECOND EDITION".

Figure 3.1: Main Landing Page

React App

csc648.mskim.dev

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

A Market place that connects to people only associated with SFSU to sell or purchase items

Purple Market

All Items ▾ the constitution Search Post Items Login Register

2 items found

Time: Newest to Oldest ▾

Latest Items  
Price: High to Low  
Price: Low to High

The Constitution and the Nation:...  
Price: 10.00  
Description: History Textbook  
Message Seller

The Constitution and the Nation:...  
Price: 10.00  
Description: History Textbook  
Message Seller

Figure 3.2: All Item Keyword Search

## 3.2 Post Item Page

The screenshot shows a web browser window for a React App at the URL [csc648.mskim.dev/createpost](https://csc648.mskim.dev/createpost). The page title is "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only." Below it is a subtitle: "A Market place that connects to people only associated with SFSU to sell or purchase items". On the left, there's a logo for "Purple Market". At the top right are buttons for "Post Items", "Login", and "Register". The main content area is titled "Item Post Page". It contains fields for "Item Name:" (e.g. Macbook), "Price:" (e.g.\$25), "Category:" (Select Category dropdown), "Upload Image:" (Choose File button showing "the-mona-lisa-lf1032.jpg"), and "Description:" (e.g. This product has so many features). A note says "\* - mandatory fields" and "May take up to 24 hours for item post to be approved." To the right of the form is a large rectangular area with a large 'X' drawn through it. At the bottom are "Cancel" and "Post Item" buttons.

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.  
A Market place that connects to people only associated with SFSU to sell or purchase items

Purple Market

All Items ▾ the constitution Search Post Items Login Register

Item Post Page

Item Name: \* e.g. Macbook

Price: \* e.g.\$25 May take up to 24 hours for item post to be approved.

Category: \* Select Category ▾

Upload Image: Choose File the-mona-lisa-lf1032.jpg

Description: \* e.g. This product has so many features

\* - mandatory fields

Cancel Post Item

Figure 3.3: Post Item Page - Blank

React App    New Tab    csc648.mskim.dev/createpost

SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.  
**A Market place that connects to people only associated with SFSU to sell or purchase items**

**Purple Market**

All Items ▾ Search Here Search Post Items Login Register

Item Post Page

Item Name\*: The Mona Lisa \* - mandatory fields

Price\*: 1000000 May take up to 24 hours for item post to be approved.

Category\*: Select Category

Upload Image: Choose File the-mona-lisa-if1032.jpg

Description\*: This is the famous painting by Leonardo Da Vinci



Cancel Post Item

Figure 3.4: Post Item Page - With Data Entered

### 3.3 Register Page

The screenshot shows a web browser window for a project named "React App". The address bar displays the URL `csc648.mskim.dev/register`. The page has a purple header with the text "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only." and "A Market place that connects to people only associated with SFSU to sell or purchase items". On the left, there is a logo for "Purple Market". The main content area is titled "Register". It contains fields for First Name\*, Last Name\*, Email\*, Password\*, and Confirm Password\*. There is also a "Show Password" button and a note stating "\* - mandatory fields". A large blue "Register" button is at the bottom, and a link to "Login" is provided below it.

*SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only.*  
*A Market place that connects to people only associated with SFSU to sell or purchase items*

Purple  
Market

All Items ▾ Search Here Search Post Items Login Register

## Register

First Name\*: Enter your first name

Last Name\*: Enter your last name

Email\*: Enter your SFSU email

Password\*: Enter your password

Confirm Password\*: Confirm your password

Show Password

\* - mandatory fields

Register

Already have an account? [Login](#)

Figure 3.5: Register Page

### 3.4 Login Page

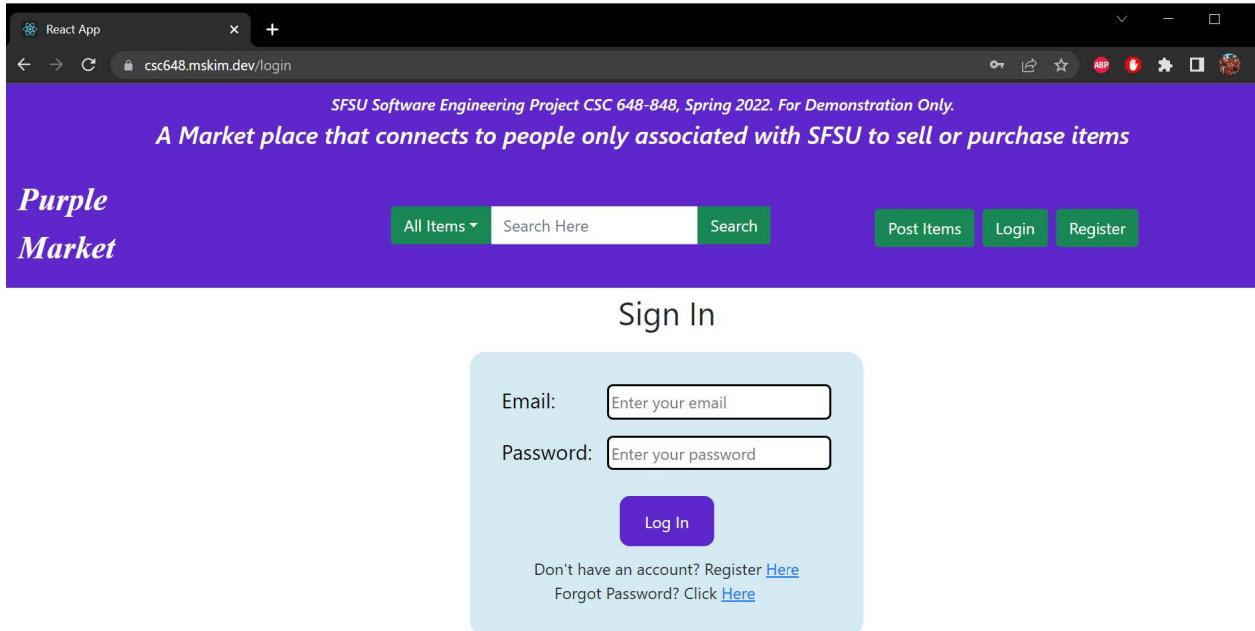


Figure 3.6: Login Page

### 3.5 Password Reset Page

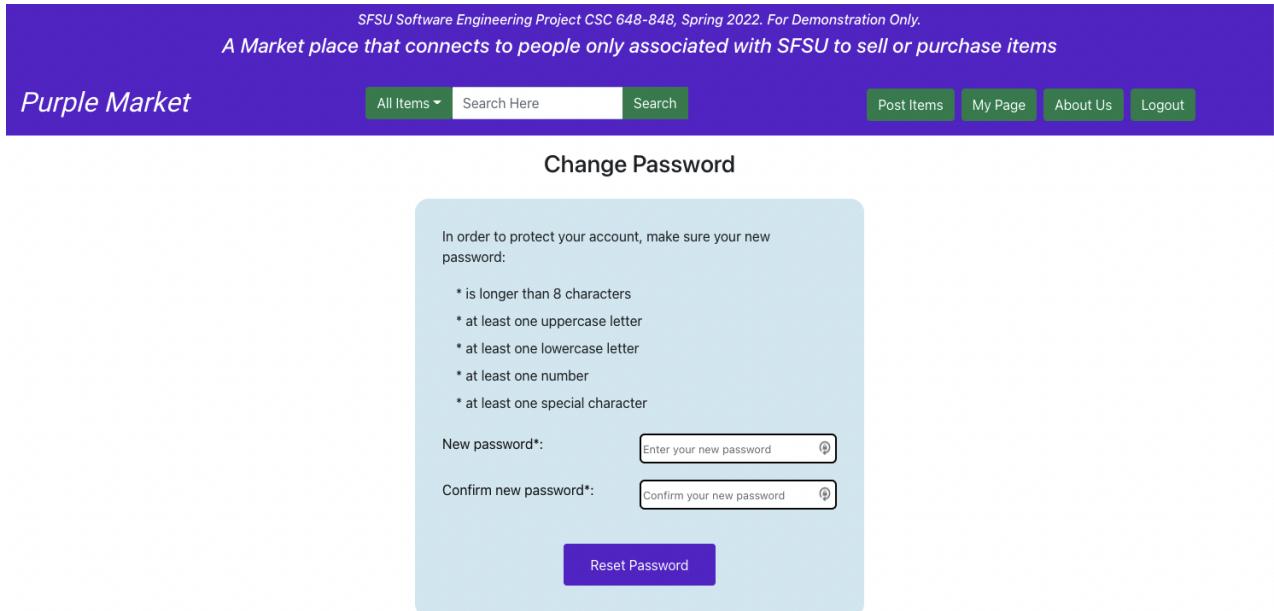
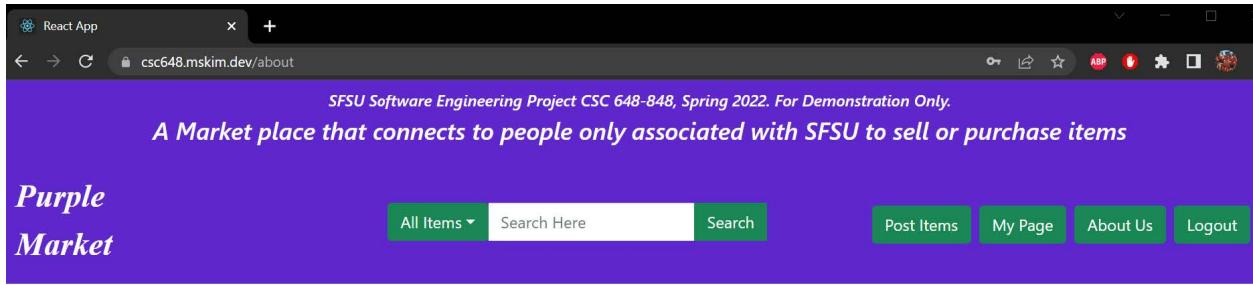


Figure 3.7: Password Reset Page

### 3.6 About Page



Software Engineering  
Class  
SFSU Spring 2022  
Section 01  
Team 08

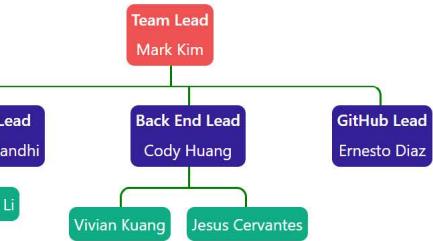


Figure 3.8: About Page

### 3.7 User Dashboard

The screenshot shows a web browser window for a project titled "Purple Market". The URL is `csc648.mskim.dev/myPage`. The page header includes the text "SFSU Software Engineering Project CSC 648-848, Spring 2022. For Demonstration Only." and "A Market place that connects to people only associated with SFSU to sell or purchase items". On the left, there is a sidebar with the "Purple Market" logo. The main content area is titled "MyPage" and contains tabs for "Post History", "Message" (which is selected), and "Account Details". Under the "Message" tab, there are two sub-tabs: "Message Inbox" (selected) and "Sent Message". Below these tabs is a table listing messages. The table has columns: Item Name, Receiver Name, Message, Receiver Email, and Contact Info. One message is listed: "Snowboard" sent to "Sporty Spice" with the message "Is this still available?" and emails "t4@mail.sfsu.edu" and "whuang16@mail.sfsu.edu".

Item Name	Receiver Name	Message	Receiver Email	Contact Info
Snowboard	Sporty Spice	Is this still available?	t4@mail.sfsu.edu	whuang16@mail.sfsu.edu

Figure 3.9: Messages Tab

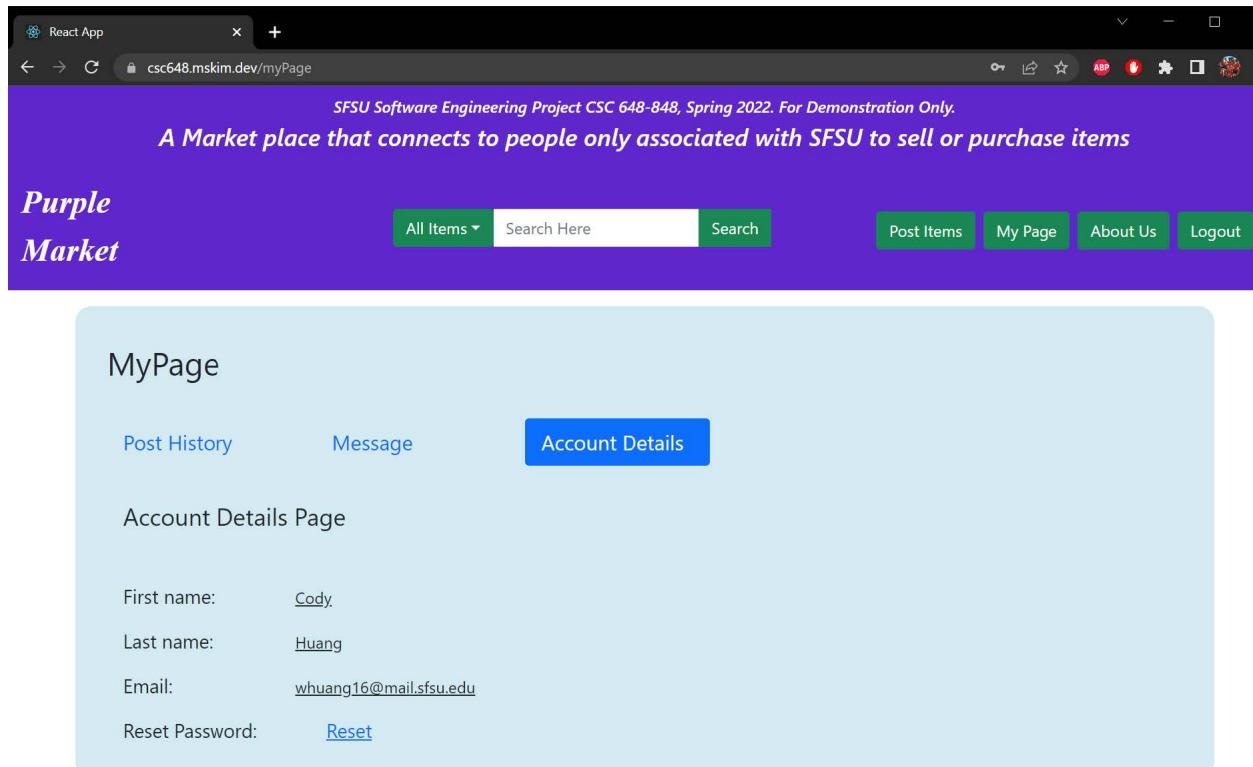


Figure 3.10: Account Details Tab

## 4 Database Organization

### 4.1 Database Schema

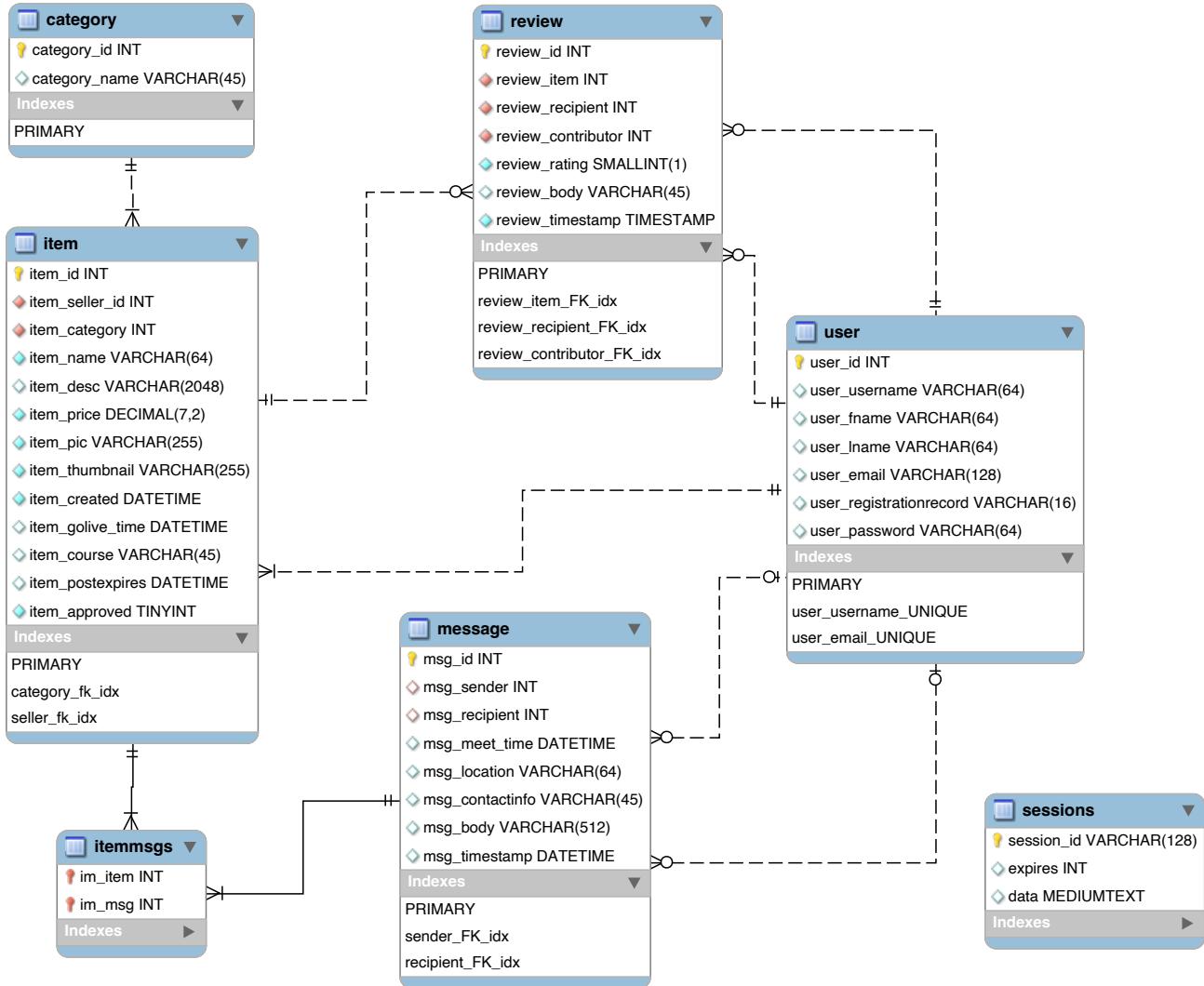


Figure 4.1: Enhanced Entity Relationship Diagram

## 4.2 Database Table Screenshots

The screenshot shows the MySQL Workbench interface with the 'csc648db\_model.mwb' database selected. The left sidebar displays the 'SCHEMAS' tree, which includes the 'item' schema. The main pane shows the results of a query against the 'item' table:

```
1 • SELECT * FROM csc648.item;
```

The results grid contains 19 rows of data, each representing an item with its details such as name, description, price, and creation date.

item_id	item_seller_id	item_category	item_name	item_desc	item_price	item_pic	item_th...	item_created	item_golive_t...	item_course	item_postexpir...	item_approv...
2	1	6	testname1031	test cup of tea at 1031 now 1031	1031.00	9c18fb9e0294417...	thumb-9...	2022-05-16 05:32:30	HULL	NULL	1	
3	2	6	Kettle	It's white and it lights up.	50.00	079549ed95d9...	thumb-2...	2022-05-16 05:32:30	HULL	1	1	
4	3	6	red solo cups	Red solo cups for midnight partying and getting...	1940.00	b930-a41a1a58282...	thumb-3...	2022-05-16 05:41:11	HULL	1	1	
5	4	11	Painting of a MUNI train	It's art and pretentious	2000.00	0677d7a650c0eb183...	thumb-0...	2022-05-16 05:41:41	HULL	1	1	
6	4	2	Dictionary	Who would've thought you can look up words in...	2.00	277728559b2e51...	thumb-2...	2022-05-16 05:42:37	HULL	1	1	
7	5	8	Helmet	Protect your noggin	25.00	cdd6779e929359b...	thumb-5...	2022-05-16 05:43:43	HULL	1	1	
8	5	1	Ice Sculpture	Appreciate it while you can, because it's just go...	1000.00	2abcc0b3c9b7df4...	thumb-2...	2022-05-16 05:44:40	HULL	1	1	
9	5	5	White Lamp	It's white and it lights up.	30.00	9b04ae1e8a870cd3...	thumb-9...	2022-05-16 05:45:29	HULL	1	1	
10	6	4	Used Laptop	A used still working laptop for your homeworks ...	1044.00	8b5a537bcdc7fd...	thumb-8...	2022-05-16 05:45:33	HULL	1	1	
11	6	4	Used Headphones	Used headphones. Used headphones. Right side is a b...	10.00	9593a033a0f3...	thumb-1...	2022-05-16 05:45:36	HULL	1	1	
12	7	3	Brown v Board	A required book	5.00	ca7790609305d07...	thumb-0...	2022-05-16 05:50:18	HIST471	1	1	
13	7	3	The Constitution and t...	History Textbook	10.00	3f472569a0204b4...	thumb-3...	2022-05-16 05:51:40	HIST471	1	1	
14	7	3	The Constitution and t...	History Textbook	10.00	8b05b53645587b3...	thumb-8...	2022-05-16 05:52:24	HIST471	1	1	
15	9	8	Snowboard	Strap this popsicle stick on your feet and slide ...	500.00	3fc6e5b361c23c...	thumb-3...	2022-05-16 05:54:22	HULL	1	1	
16	9	7	Stapler	It's my stapler.	5.00	7c05391140c7bd...	thumb-7...	2022-05-16 05:54:43	HULL	1	1	
17	8	7	My melody notebook	A cute my melody notebook to write down all yo...	1104.00	3a949b17443ee58...	thumb-3...	2022-05-16 06:06:06	HULL	0	0	
18	8	6	Hello Kitty Toaster	Breakfast is the most important meal of the day...	111.00	c8c8486bad1cd74...	thumb-0...	2022-05-16 06:12:15	HULL	0	0	
19	11	4	Gudetama Tamagotchi	If you're a lover and have a difficult time making fr...	117.00	0fe472e291916954...	thumb-0...	2022-05-16 06:18:31	HULL	0	0	

Figure 4.2: Item Table

The screenshot shows the MySQL Workbench interface with the 'csc648db\_model.mwb' database selected. The left sidebar displays the 'SCHEMAS' tree, which includes the 'message' schema. The main pane shows the results of a query against the 'message' table:

```
1 • SELECT * FROM csc648.message;
```

The results grid contains 10 rows of data, each representing a message exchange between users.

msg_id	msg_sender	msg_recipient	msg_meet_time	msg_location	msg_contactinfo	msg_body	msg_timestamp
1	9	6	NULL	5551212	Hi there! I want some headphones!	2022-05-16 05:59:28	
2	9	6	NULL	5551212	Why won't you message me back?	2022-05-16 05:59:44	
3	9	3	NULL	5551212	I need them, now!	2022-05-16 05:59:03	
4	9	1	NULL	5551212	MM...tea	2022-05-16 05:57:12	
5	9	1	NULL	5551212	I'd like some tea please	2022-05-16 05:57:29	
6	8	9	NULL	105-912-1234	Hello nice snowboard. Can I get 50% off? Test 1...	2022-05-16 05:59:46	
7	10	9	NULL	null	hey how much is this?	2022-05-16 06:00:30	
8	8	9	NULL	111-111-1111	Hello there. Nice Stapler. I'll pay you double. tes...	2022-05-16 06:01:13	
9	8	5	NULL	111-111-1111	Hello nice white lamp. I am totally not a moth. N...	2022-05-16 06:02:57	
10	12	9	NULL	null	whuang16@mail.sfsu.edu Is this still available?	2022-05-20 00:43:00	

Figure 4.3: Message Table

MySQL Model csc648db EER csc648aws

Administration Schemas

Filter objects

Columns

- user\_id
- user\_username
- user\_fname
- user\_lname
- user\_email
- user\_registrationrecord
- user\_password

Indexes

Object Info Session

No object selected

Query 1 inserts\* messagequeries inserts SQL File 8\* SQL File 9\* SQL File 11\* SQL File 12\* SQL File 10\* SQL File 10\* >>

Limit to 1000 rows

1 • SELECT \* FROM csc648.user;

Result Grid Filter Rows: Search Edit: Export/Import: Result Grid Form Editor Field Types Query Stats Execution Plan

user_id	user_username	user_fname	user_lname	user_email	user_registrationrecord	user_password
1	test1028@sfsu.edu	test	1028	test1028@sfsu.edu	registered	\$2b\$10\$12RF04a8f06w9HmRg4uSODXUvFb...
2	test@sfsu.edu	Mark	Kim	test@sfsu.edu	admin	\$2b\$10\$nTlJSQwzgHJ5Dml4xoLg0jh3FJqeN...
3	test1034@sfsu.edu	test	1034	test1034@sfsu.edu	registered	\$2b\$10\$KExoxR1f1MU7HHQqikgcoOLdMqnsj...
4	t1@sfsu.edu	Joe	Schmoes	t1@sfsu.edu	registered	\$2b\$10\$xsSU20dazxwMpG3uz9fSBurGM0kq...
5	t2@sfsu.edu	Jane	Austin	t2@sfsu.edu	registered	\$2b\$10\$GejuCEfsu5A0t9nJ7w.4tpc3nXg/ZP...
6	test1043@sfsu.edu	test	1043	test1043@sfsu.edu	registered	\$2b\$10\$OpupuLAZp206uIZVnejBDxp.706xLjkCS...
7	t3@sfsu.edu	Amy	Student	t3@sfsu.edu	registered	\$2b\$10\$hSecBUO6P/awtEuUSEWrmutW2QRU...
8	test1049@sfsu.edu	test	1049	test1049@sfsu.edu	registered	\$2b\$10\$Vn6rJcqwJHrpnaxajn4uSFgy4rvMC...
9	t4@mail.sfsu.edu	Sporty	Spice	t4@mail.sfsu.edu	registered	\$2b\$10\$SWoZxL8zZ.qTYUY3vFOED6QdR...
10	tester1@mail.sfsu.edu	tester1	tester1	tester1@mail.sfsu.edu	registered	\$2b\$10\$KeUS1Zkanx.QE1cnRX0XabPug.wRL...
11	test1115@sfsu.edu	test	1115	test1115@sfsu.edu	registered	\$2b\$10\$1x4rbnHxn25q4u51AdsWO4s3eT3jh...
12	whuang16@mail.sfs...	Cody	Huang	whuang16@mail.sfs...	admin	\$2b\$10\$k4VXIXhlwHc02NGmRoDaeV4sCuQn...
NULL	NULL	NULL	NULL	NULL	NULL	NULL
user 5						

Query Completed Apply Revert

Figure 4.4: User Table

## 5 GitHub Organization

### 5.1 Main Branches

All team members had full access to all branches *except* for the main branch.

1. main: Only Team Lead and GitHub Master has access
2. devM0: 1st Milestone development branch
3. devM2: 2nd Milestone development branch
4. devM3: 3rd Milestone development branch
5. devM4: 4th Milestone development branch
6. devM5: 5th Milestone development branch
7. deploytest: staging branch for deployment to remote server

### 5.2 File Organization

#### 5.2.1 Top Level Folders

**Milestones** Contains all milestone documentation.

**application** Contains all source files and build files for the application.

**credentials** Contains credentials to access the server instance as well as the database.

#### 5.2.2 Application Folder

##### **deploy**

Contains nginx web server configuration files and various deploy scripts.

##### **server**

Contains all back-end code, database configuration and setup files, models, and routes. Contents include the following folders:

- config: Database connection pool configuration
- database: Database model, table creation queries, dummy data insert queries, and some query templates for early development
- models: Contains all functions to query the database (e.g. items, users, messages, etc.)
- routes: Restful API routes that allow the front-end to communicate with the back-end, query the database, and retrieve images.

- error & validator: Helper functions to display errors and perform back-end validation.

## **web**

Contains all front-end react and bootstrap code; also contains build files for deploying the front end to the nginx web server.

- public: Contains base index.html and associated files for front-end build
- src: Contains all front end react source code; the subfolders contained herein are as follows:
  1. Pages: Provides routing to React component pages
  2. components: Contains modular React components that provide all front-end logic for web page rendering and interactivity
  3. images: Static images for the about page

### 5.2.3 GitHub homepage screenshot

The screenshot shows a GitHub repository page for 'csc648-SFSU/csc648-03-sp22-team8'. The repository is private, with 1 watch, 1 fork, and 0 stars. It has 24 branches and 0 tags. The commit history shows several commits from 'Mark Kim' and 'Mark Kim' renaming files like 'Milestone4' and updating 'README.md'. The repository description includes instructions for teams to add their application URL or IP to the description. Below the repository details, there are sections for releases, packages, contributors, and languages.

**Commits**

- Mark Kim and Mark Kim renamed Milestone4 document (6a67fab, 4 days ago) 462 commits
- Milestones renamed Milestone4 document (4 days ago)
- application changed all deploy urls back to develop urls (4 days ago)
- credentials Update README.md (last month)
- .gitignore Fixed a small detail on labeling for sort (4 days ago)
- LICENSE Initial commit (3 months ago)
- README.md Update README.md (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.
  - o Please follow the naming convention assigned by your instructor.
2. PLEASE REMOVE THE USERNAME FROM THE REPOSITORY NAME!!!
3. Add ALL members of your team to this repository. For it to count, they must ACCEPT the invite.
4. Fill out the table below

Student Name	Student Email	GitHub Username
--------------	---------------	-----------------

Figure 5.1: Team GitHub Homepage

## 6 Google Analytics

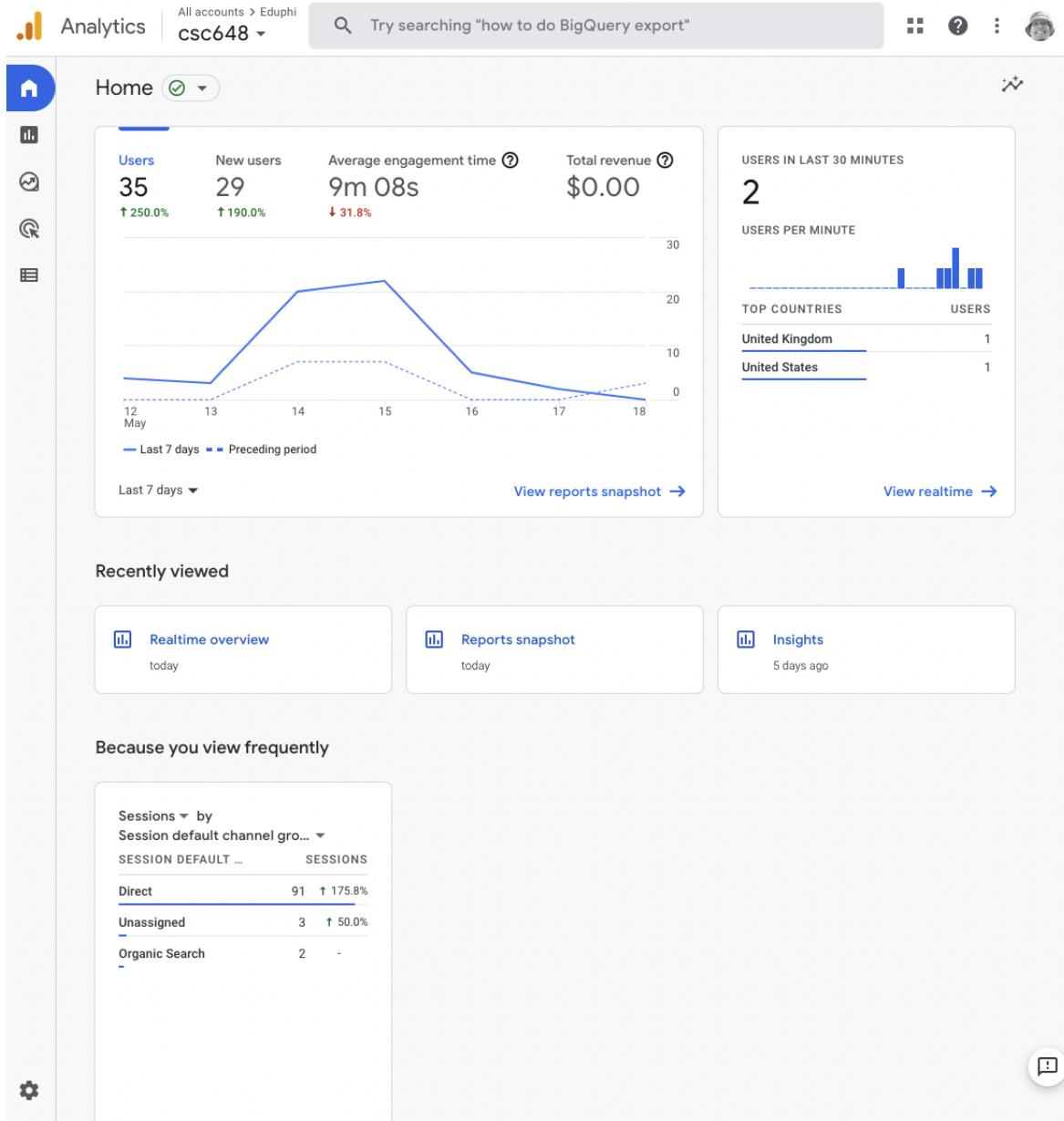


Figure 6.1: Google Analytics Screenshot

## 7 Project Management

We used a combination of Discord, Trello, and Google Docs to manage our project. Discord was used to do regular check-ins on progress as well as communicate any difficulties that team members may be having. Trello was used for high level management of the project and Google Docs was used for more granular management.

### 7.1 Trello

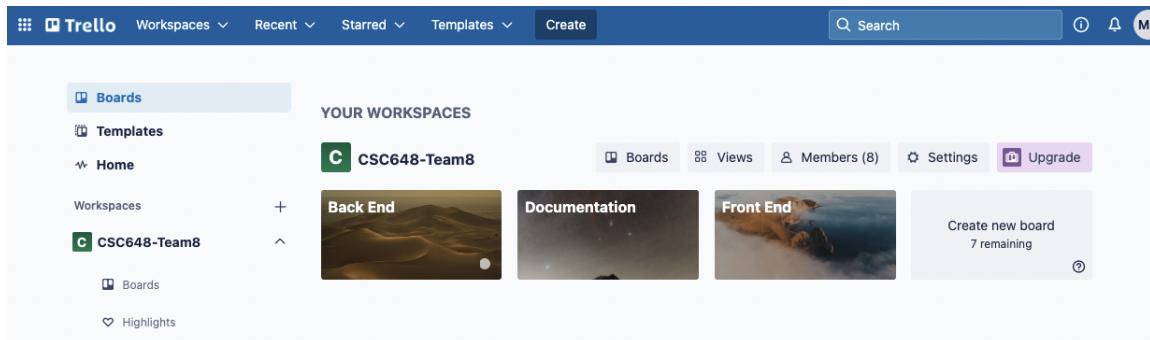


Figure 7.1: Trello Homepage

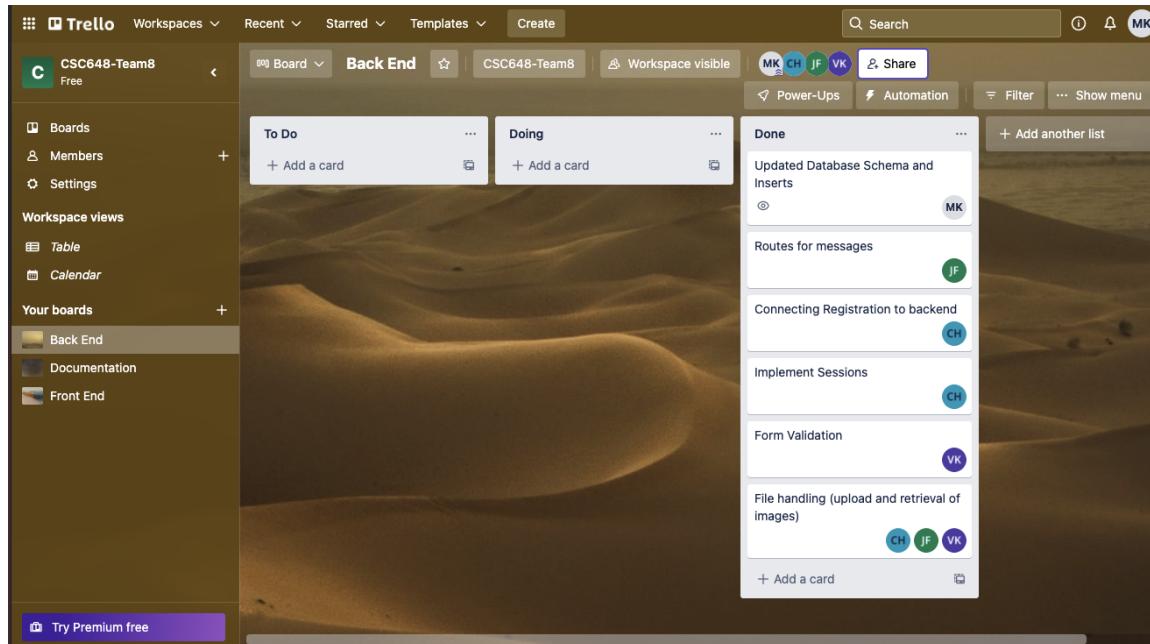


Figure 7.2: Trello Backend

## 7.2 Google Docs

The screenshot displays a Google Docs page with several sections of text and tables, along with a series of Google Chat messages from team members (Mark Kim, KHUSHBOO, JESUS, VIVIAN, ERNESTO, and MARK) regarding the document's content.

**Document Sections:**

- Milestone4.pdf**: A PDF titled "Milestone4.pdf" containing instructions for executing a QA test plan. It includes a "Suggested format for QA Test Plan Table" with columns for test #, test title, test description, test input, expected correct output, and test results (PASS or FAIL). It also specifies performing testing on 2 major WWW browsers.
- 4) Code Review:** Instructions for code review, including:
  - Chose the code (substantial portion of it) related to the feature you used for QA and usability test. One team member should submit code to other team member(s) for peer review.
  - Peer review should be performed by one other team member of your team.
  - Peer review is to be done in writing (e-mail or related methods like in github) and some review comments are to be included in the code
- Important Note:** Peer review must include checking for basic header and in-line comments, as well as proper and consistent class/methods/variable names and consistency with naming established in DATA section of Milestone 2. Commit comments in github should also be reviewed.
- What to submit in M4 report to document code review?** Note that sending code via e-mail attachment is not possible due to security reasons. Here is what we recommend:
  - Person who code is being reviewed sends e-mail to reviewer with pointer to the code and asks for review
  - Reviewer reviews the code in whatever way is most practical for you (e.g. commenting on code in repository, using github review options)
  - Reviewer sends summary of review in email back to coder
  - To document the above in M4 submit the following: copy of all emails from above (screen shot OK) AND also include green shot capture of commented reviewed code or the commented-reviewed code in PDF format or similar not executable format (limit to 3 pages).
- Important:** It is critical that code reviews are friendly and helpful, intended to help and educate, and not to criticize. It is strongly suggested that you use peer review in the development of the whole system.
- 5) Self-check on best practices for security – ½ page** (with a table):
 

Asset to be protected	Types of possible/expected attacks	Your strategy to mitigate/protect the asset
- How to present:** Best is to present this in the table below, include ALL assets and methods you plan to implement
- 6) Self-check of the adherence to original Non-functional specs – performed by team leads**

**Google Chat Messages:**

- Mark Kim: "7:27 PM May 11 JESUS" (with a checkmark)
- Mark Kim: "7:32 PM May 11 KHUSHBOO (back end); JESUS & VIVIAN (front end)" (with a checkmark)
- Mark Kim: "7:33 PM May 11 KHUSHBOO (back end); JESUS & VIVIAN (front end)" (with a checkmark)
- Mark Kim: "7:34 PM May 11 ERNESTO" (with a checkmark)
- Mark Kim: "7:35 PM May 11 MARK" (with a checkmark)

**Page Footer:** Page 4 / 6

Figure 7.3: Google Docs

## 8 Team member self assessment and contributions

### 8.1 Mark Kim

**CSC6484 Self Assessment Mark Kim**

 Mark S Kim ☀️ ⏪ ⏴ ⏵ ⏶ ⏷ ⏸ ⏹ ⏺ Sat 5/14/2022 3:57 PM

To: Jesus Cervantes Fajardo; Ernesto Diaz; Khushboo Gandhi; Cody Huang; Vivian Kuang; Jiasheng Li

Hey all,

Here's my self-assessment (sorry...it's a long one):

a. Contributions:

1. Set up, deployed and maintained server instance
2. Set up MySQL schema with dummy data for early development
3. Testing of queries, models, routes, and UI
4. Code review
5. General high-level project management
6. Documentation maintenance (e.g. collate team contributions to submit as one document)
7. Added some models and routes
8. Assisted in debugging

b. Number of submissions to GitHub: 233 commits (but as stated earlier, this number is inflated due to my incorrect management of the server deployment—many of those commits were extremely small and insignificant).

c. Understanding how development workflow interacts with the production environment was definitely a learning experience for me. Managing the various moving pieces of the project was certainly challenging. Since all parts of the project interacted with each other, the contributions of one team member can have a significant impact on other portions of the project, so communication was imperative. Unfortunately, I found that I was not very effective at facilitating that communication and collaboration. With the variability between all my teammates' schedules, it was hard to maintain that consistent line of communication and/or collaboration between all the team members (which is understandable since we all have different class, work, personal life schedules). In addition, maintaining a clean delineation between each team member's project scope could also be problematic. Team members making small (but possibly significant) changes to parts of the project assigned to another team member could cause havoc if those changes are not clearly communicated. On multiple occasions, I also found work was duplicated because team members either did not push or pull their changes to or from the repository, or because the changes were not communicated effectively. The common theme here is that communication and collaboration are probably the greatest keys to success in a team environment. There were some team members that communicated well with each other, but that was the exception rather than the rule. Finally, this particular environment has its own set of minefields from a leadership standpoint because I am working with fellow students (some of whom are far more talented than I in programming), which makes this leadership position a tentative one.

On a technical level (as an individual contributor), one major challenge was server deployment. Building the front-end to be deployed on the nginx web server differs significantly from testing on a local development testbench. Multiple strategies were attempted and many attempts were total failures. I spent an inordinate amount of time just trying to get the front-end to communicate with the back-end; I have a special kind of hatred for Cross Origin Resource Sharing.

d. I think one of the ways that I would do better next time is to have a much more structured project management process. Although Trello and Google Docs were used to manage the team, these technologies were not utilized to their fullest potential. That falls on me, since I approached a lot of the project management with a laissez faire approach. Anthony mentioned to me that sometimes the team lead just needs to make an executive decision and bring the entire team in line with it; that is advice that I will take to heart. It became readily apparent that an iterative approach to SE is necessary. Often, we found that code needed to be changed because a necessary feature became apparent much later in the development cycle. In future projects, I will definitely use a more robust method of tracking the status of requirements (e.g. new, completed, etc). I am still struggling to figure out how to better facilitate active communication between all team members—I have definitely not found a solution to this yet.

◀ Reply ◀ Reply all ↗ Forward

Figure 8.1: Mark

## 8.2 Khushboo Gandhi

CSC 648 Self Assessment Khushboo Gandhi

KG Khushboo Gandhi To: Mark S Kim; Cody Huang; Jiasheng Li; Ernesto Diaz; Vivian Kuang; Jesus Cervantes Fajardo Sat 5/14/2022 8:09 PM

Hey Everyone,  
The following is my self-assessment.

**Contributions:**

1. Designed the project structure of the front-end project and divided the different parts into respective pages and modules.
2. Set up the react project (front end).
3. Created the about page, (individual user pages), Navbar, Home page.
4. Connected the login, registration, message post page, message detail page, myPage to the backend with API calls and array mapping.
5. Setup all the routes using react router dom and implemented route protection.
6. Created the search component and generated search logic with respect to front end.
7. Assisted in debugging the front-end application with respect to react router dom, CSS styling, responsiveness, Bootstrap, material UI Cards.
8. Implemented the search with filters, categories and sorting through filters (date, price) with and without search term.
9. Managed the overall front end and ensured the overall website runs smoothly as expected.
10. Assisted in setting up temporary hosting for backend on heroku (which was later removed when the actual AWS was working fine.)
11. Assisted in general high level project management.
12. Assisted in back end in acting as a bridge with the front end in terms of mentioning required routes (eg: /getItems) and testing them using front end.

Total Commits made on GitHub: 53 commits

**Challenges Faced:**

In terms of challenges, I would say the search function was the most difficult to implement. This was mainly because there were multiple different permutations and combinations possible (with category and search term selected, with category and no search term selected, only search selected and so on). Further, for each of them, sorting was to be enabled with all the combinations and the search bar had to be such that it should be implemented from all pages. It was later I found out about something called React.Context which helps to transfer data from one component to the next on the same page.

Another challenge I faced was getting the route protection done (ie: only signed in users can have access to all the pages and certain button clicks and otherwise you are prompted to login. It took a while to figure out how to achieve the same. Later we found out about local history storage in the browser and how to navigate through only required pages if a user is logged out.

As the front-end team lead, there were a few other challenges like managing timings of everyone in the team to gather everything and handling multiple branches by the front end alone. However, my teammates and team lead helped a lot in helping us achieve the required proficiency.

**What have I learned:**

I have certainly learned a lot about handling multiple branches on GitHub, pulling, committing, creating new branches, switching branches, and merging branches on GitHub. I have learned about team development, handling documentation and how to work together. I am grateful for my team lead (Mark Kim) for helping us learn so much about GitHub, management skills and the fact that he was always available whenever there was a bug. I am also thankful to Jiasheng Li who closely worked with the front end and created most of the static pages, helped in validation and took responsibility on his own. If I am working on a similar project next time, I would like the front-end code to be much more organized. For eg: The navbar code has more than 400 lines of code alone because it also does a lot of work like calling API from backend and setting the items. Ideally this could be done in another file which can be called here. A lot of strings are hard coded and can be a problem to the deployment team to change again. I would also like the code to be smoother and easier to read with all functions neatly organized into its own files.

Figure 8.2: Khushboo

### 8.3 Cody Huang

**CSC648 Self Assessment Cody Huang**

 Cody Huang

To: Mark S Kim

Sat 5/14/2022 12:22 PM

Cc: Vivian Kuang; Jesus Cervantes Fajardo +3 others

**Contributions:**

- added order by and direction for filtering of items in model and routes
- testing of routes and models
- connecting registration to backend
- create sessions on login
- destroy sessions on logout
- flash messages
- added docker setup for environment (not used)
- implement s3 api to store images (not used)

30 commits made

One of the main challenges I had was setting up the environment to run. I first used docker containers and planned on using S3 buckets to store user uploaded images. However, there were very various problems I encountered using these methods and ended up being scrapped. Another challenge was using git and dealing with merge conflicts, but that was quickly solved when work was separated on different branches.

One thing I learned from being a part of this team is that communication should be the focus. This was my first time being a team lead (backend), I was used to working on things by myself and doing things my own way. Some problems developed from me isolating myself and trying to work on things on my own. I am grateful for the team lead, Mark Kim, explaining how I can do a better job as the backend lead and creating a friendly and productive environment. It has helped me fix my communication issues and helped me be more hands on.

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

Figure 8.3: Cody

## 8.4 Jiasheng Li

**CSC 648 Self Assessment Jiasheng Li**

JL Jiasheng Li To: Vivian Kuang; Mark S Kim; Khushboo Gandhi; Cody Huang; Ernesto Diaz +1 other Sun 5/15/2022 11:39 AM

Hi Everyone,

This is my self-assessment.

**Contributions:**

1. Drafted and revised two key user cases
2. Designed and created register, login, homepage, message page, post item page, MyPage, forgot password, and reset password mockups
3. After revised the mockups of all pages, created UI for register, message, MyPage, forgot password. And reset password mockups
4. Create front-end password validation and designed how the error messages show for users if their password validation fails.
5. Worked along with front-end leader Khushboo to connect some pages with API, the filter function and the sorting function for user search results.
6. Checked and modified the responsiveness for register, homepage, message, MyPage, forgot password, and reset password pages.

**Total Commit made on Github:** 31

**Challenges Faced:**

The first challenge for me is to learn how to collaborate, support, and learn with my teammates. Since we have implemented a semester-long project, I needed to learn and contribute lots of things for the project and my team. Although it is a challenge, it is also a significant opportunity for me to work with excellent and knowledgeable people who are my teammates. My second challenge is I need to learn a completely new frontend framework, React as a beginner. Thus, I reviewed so many tutorial videos and documentations for React along this semester. However, I learned lots of things for React finally. My third challenge is I need to keep much patience to test and solve the bugs for the projects. Since the project is a big project, each problem can be caused from multiple components.

**What have I learned:**

I definitely learned lots of stuff including corporations with my team and knowledge of developing a full-stack software. Effective communication with other members in a team is not an easy thing, but I have learned how to work with people who are from different cultures in a remote working environment. Besides, I have learned a popular frontend framework, React, so I have basic knowledge about it. I have learned how to connect frontend stuff with the API and how to use react bootstrap. Finally, I have also learned how to create routes and validation in backend side.

Figure 8.4: Jiasheng

## 8.5 Vivian Kuang

**CSC 648 Self-Assessment Vivian Kuang**

Vivian Kuang To: Jesus Cervantes Fajardo; Ernesto Diaz +4 others Sat 5/14/2022 11:28 PM

Hello,

This is my self-assessment

Contributions

- Worked on models and routes for post, login, registration, getting items.
- Tested routes using Postman
- Tested queries using Workbench
- Was tester for usability test plan in M4
- Part of Frontend code review in M4
- Tried to connect the backend with the frontend but it didn't work

Number of commits: 13 commits

In the beginning I misunderstood the routes and made the get item functions instead of router.get. When we were just getting started with the AWS, learning about the S3 bucket and how to get the images from the S3 bucket was a bit confusing. And the reverse proxy thing was also confusing. I overslept 10 minutes for one of the 10 AM Sunday meetings. The most difficult thing was debugging the code and console.logging many variables.

Next time I will communicate better and ask for help and clarification more often. I will comment on my code in the very beginning. I will make more descriptive commit messages. I will go through my code and ask for code review from my teammates. I learned about git stash and git stash pop and git merge.

Thanks,  
Vivian

Figure 8.5: Vivian

## 8.6 Ernesto Diaz

**CSC 648 Self Assessment-Ernesto Diaz**

 Ernesto Diaz ☀️ ⌂ ↲ ↳ → ⋮

To: Khushboo Gandhi; Jiasheng Li; Vivian Kuang +3 others Mon 5/16/2022 12:41 PM

Hello Everyone,

**Contributions:**

1. Set up the GitHub Repository and send out invites.
2. Assisted with GitHub commands, such as merging, creating branches, add, commit, and push.
3. Started the Mock Pages for the Homepage, Login page and Item Details using Figma
4. Started the Search Bar component using React.
5. Colloborated with Khushboo to make the Post Item page using React and React-Bootstrap to style it.
6. Revised the Login Page using React and React-Bootstrap to style it.
7. Did responsiveness testing for login, homepage and post an item page.

Commits: 15

**Challenges Face:**

One of challenges I faced was learning how to use React and React-Bootstrap, I really struggle with it at first and it was very frustrating. But I did get help from Khushboo, she was very patient and helpful with me. Another challenge was the communication between all the members, and where to push your code or pull it from. For example, I believe the Login page was duplicated twice on accident due to just the lack of communication on what branch to pull.

**Things I Learned:**

I learned as GitHub master, I should've put a limit on how many branches we should have created and just organized it better. I should've voiced my concern because it got confusing towards the end of the project on which branch to pull or push.

Thanks,  
Ernesto

↶ Reply ↶ Reply all ↷ Forward

Figure 8.6: Google Docs

## 8.7 Jesus Cervantes Fajardo

**CSC 648 Self-Assessment**

 Jesus Cervantes Fajardo ☀️ ⌂ ↲ ↳ → ⋮

To: Mark S Kim; Cody Huang; Khushboo Gandhi +3 others Wed 5/18/2022 8:37 PM

Hey Team-8,

My self-assessment for milestone 5 can be found below:

Contributions:

1. Connected S3 bucket to server (not used)
2. Implemented static image retrieval (not used)
3. Added revisions to message route and model
4. Added revisions to item route and model
5. Completed the QA test plan portion of Milestone 4
6. Minor corrections to grammar found within our website

Number of commits to our GitHub: 14

Main Challenges:

I would say this was my first real experience creating a full-fledged web project. With the lack of experience, it was difficult to work on the tasks assigned to me. Not only that, but at the beginning of the project it was hard for me to understand how our deployment server worked.

Another thing to note is that I am used to working on my own, and I am a very quiet person. This meant that it was hard for me to communicate with my team, especially in the first couple team meetings. However, the more I worked with the group I became more comfortable, allowing me to ask questions and get the help I needed to get my tasks completed.

Things I Would Improve:

In the future, I will try to break out of my comfort zone sooner and build a better work environment with my group. This way I can communicate with the group when I am struggling or need help sooner. I will also put more effort into researching the topics needed to get my work completed. Another thing I would want to improve on is to meet with my groups more often. I felt that when the back-end team met separately I would learn more and feel more engaged.

Thanks,  
Jesus Cervantes

⤵ Reply ⤲ Reply all ⤶ Forward

Figure 8.7: Google Docs