SW Engineering CSC648/848 Fall 2022

MarketGator

Team 05

Patrick Celedio: pceledio@mail.sfsu.edu	Team Lead/Frontend Lead
Michael Feger: mfeger@mail.sfsu.edu	Backend Lead
Ahmad Adnan: aadnan@mail.sfsu.edu	Github Master
Rohit Devdhar: rdevdhar@mail.sfsu.edu	Frontend Team Member
Nyan Ye Lin: nlin2@mail.sfsu.edu	Frontend Team Member
Ethan Lunderville: elunderville@sfsu.edu	Backend Team Member

Demo Link: http://34.211.120.29:3000/

Milestone 5

December 17, 2022

History Table		
Milestone	Date Submitted	Date Revised
Milestone 01	October 08, 2022	October 12, 2022
Milestone 02	October 29, 2022	October 31, 2022
Milestone 03	November 16, 2022	November 21, 2022
Milestone 04	December 09, 2022	
Milestone 05	December 17, 2022	

Table of Contents

2. Product Summary

Name of product:

MarketGator

Description of Product:

The purpose of MarketGator is to connect the San Francisco State University community and create a platform where SFSU students and faculty can buy, sell, or share for free digital media. This platform shall facilitate the process of gathering items which shall range from obtaining textbooks, finding media resources that can be used for university projects, or to share created content among students and faculty. Our technology shall empower the SFSU students and faculty to pursue their academic and personal goals by providing an e-commerce platform that will undermine the old ways of purchasing works at high prices from online vendors who charge at a premium price.

Itemized list of all majors committed functions:

1. Registered users shall inherit all the access that unregistered users have.
2. Unregistered users shall be able to create a new account with valid SFSU email.
3. Unregistered users shall be able to view listings
4. Unregistered users shall be able to search listings
5. Registered users shall be able to post items for buying or selling.
6. Registered users shall be able to log in with valid SFSU emails.
7. Registered users shall be able to view their posts through the dashboard.
8. Registered users shall be able to view their messages through the dashboard.

9. Registered users shall be able to inquire about the items by sending a message to the sellers.
10. Registered users shall be able to receive responses from the sellers.
11. Administrators shall be able to log in with valid SFSU emails.
12. Administrators shall inherit all the access of registered users.
13. Administrators shall be able to review the listings from the registered users who are currently advertising a digital item for sale.
14. Administrators shall be able to deny listings that violate the rules and policies of the MarketGator before they are published.
15. Administrators shall be able to ban the registered users who violate the rules and policies of the website.
16. Unregistered users shall be able to view the details of the listings such as price, pictures of the digital media, description of the digital media, classes and professors related to the digital media, ratings, and the reviews of sellers.
17. Unregistered users shall be able to sort and filter the items and the listings based on categories and classes etc.

Unique feature of product:

A feature that makes MarketGator unique from competitors is its straightforward user interface.

URL to MarketGator:

http://34.211.120.29:3000/

3. Milestone Documents - M1- M4:

Milestone 1-3 Revised and Milestone 4

Milestone 1

Table of Contents

1. Executive Summary

The purpose of MarketGator is to connect the San Francisco State University community and create a platform where students and faculty can buy, sell, or share for free digital media. This platform shall facilitate the process of gathering items which shall range from obtaining textbooks, finding media resources that can be used for university projects, or to share created content among students and faculty. Our technology shall empower the SFSU students and faculty to pursue their academic or business goals and create a culture of commerce that will undermine the old ways of purchasing works at high prices from online vendors who charge at a premium price.

MarketGator will allow users to browse its selection and register MarketGator accounts for free in order to advertise, sell, and or buy digital media. MarketGator users will also facilitate the communication between buyers and sellers in the case users need to ask about certain items listed on the market. MarketGator will also have Administrators who shall be required to approve each media item before it becomes advertised in order to verify that it conforms to SFSU rules on digital media and other common ethical rules. Items deemed unsuitable or inappropriate shall be deleted by MarketGator Administrators.

A component of MarketGator that makes it uniquely useful for SFSU students and faculty is the ability for users to not only buy their digital and or physical media with United States dollars, but also facilitate the trade of digital media for digital media. MarketGator users will be able to advertise or offer their media items for the exchange of other media items listed. The reason for this is that we are aware that as university students, money is a valuable tool that is also used to obtain nourishment and supplies, and we want not only to provide an online marketplace, but also make it easier for students to obtain the resources they need.

Another component of MarketGator will be a search by major and course. We believe that this feature will help students and faculty find the digital media that they need in a very

clean and organized way. This feature will also help users who are advertising their digital media for sell or for sharing become discovered in an easily accessible way by other users who are looking to find and or purchase the digital media that they may need.

As Team 05, we are dedicated to get MarketGator in production as we believe this ecommerce application will be beneficial for the students and faculty of SFSU. We believe we are the right fit for this project because we are all senior students of SFSU, and we know first-hand the challenges of searching for media for our courses. With our expertise, we want to apply our technological skills and build MarketGator to expedite and ease the access for the required media for our classes.

2. Personae and Main Use Cases

Persona One: Jack (Student)

About Jack:

- Jack is a very busy but determined student.
- Jack is tech savvy and understands the internet well.
- Jack's main priority is having something that is minimalistic and simple to understand. His busy schedule does not afford him time to play around with badly made and overly complicated technology.

Goals and Scenario

- Jack needs to buy a textbook for his class and wants to have a way to buy it online and at a good price.
- Sometimes Jack is frustrated by the anonymity of sites like craigslist and wants a safer way to buy the things that he needs preferably from people in his immediate area.

Persona Two: John (Professor)

About John:

- John is a lecturer at SFSU who has been teaching for many years and is very passionate about his job.
- As John has aged, he has become increasingly frustrated with how complicated websites have become since the 90s when he started his profession.
- John wants an easily usable interface and is annoyed by extra details like ads.

Goals and Scenario

- With rising inflation John has had many issues where students of his were unable to purchase the required reading material for his class. He thinks that it would be much

easier if he could share his own ebook or other class related medias with students for a semester.

Persona Three: Jane (Student)

About Jane:

- Jane is a faculty member at SFSU who deeply values things like honesty and integrity.
- Jane is quite tech savvy and spends the majority of her free time on websites like reddit which has given her a strong grasp of the web.
- User interface is not huge issue for Jane however like most people she would prefer that it is easy to use and understand

Goals and Scenario

- Jane feels a strong conviction to make the web a safer place for the people who use it. After being scammed on craigslist multiple times she has become resolute in her new goal of policing the internet.

Persona Four: Joe (Admin)

About Joe:

- Joe is one of the admins of MarketGator who deeply cares about the user experience of the website. He has a strong user focus and likes the website to have good functionality and smooth user interface.
- Joe understands the web technologies, knows the ins and outs of the architecture of the website and has advanced programming skills.
- Joe has a lot of ideas for new features and changes but does not know if the users will like them or not.

Goal and scenario:

- Joe wants to release new features to stay relevant and ahead of competitors and get feedback on the new features from the existing users.

- Joe also wants the suggestions from the existing users to improve the website.

Persona Five: Jacob (Seller)

About Jacob:

- Jacob is a busy senior student at SFSU.
- Jacob has basic www skills and knows how to navigate and use all kinds of websites.
- Jacob has a lot of old textbooks which are in good condition and they have been in his apartments since he finished his junior semesters.

Goal and scenario:

- Jacob wants to get rid of his old textbooks and wants to get some extra cash for himself.
- Since being a full time senior student with a part time job, Jacob does not have a lot of free time. Jacob wants to sell his old textbooks at his convenience.

Use Cases

Use Case 1: Student Online Bookstore

Actor: Students, Faculty

Description:

Jane finds it difficult to get products and services at reasonable prices including books for their classes. As a result, she has to buy books often at expensive prices which is not ideal since she does not work enough to afford all the books she needs at full price. MarketGator will help with this problem by creating an online bookstore for the students and faculty of SFSU. Jane will browse MarketGator for books that were uploaded by either other students or faculty directly from the application at discounted prices instead of using services like Amazon and eBay where sellers would usually apply a premium price. Jane also discovers through MarketGator that faculty also hosts their course required ebooks for all the students in the class to download for free. Ultimately, MarketGator helps Jane get all her required textbooks in ebook format and helps her save money at the same time.

Use Case 2: User Moderation and Administration

Actor: Moderator, Students and Faculty

Description:

Joe is working as an administrator for MarketGator. MarketGator appears to be working properly however he seems to observe that users are uploading inappropriate media that is waiting to be approved to be published. Joe documents this inappropriate media and pinpoints the user who posted it for approval. Ultimately Joe denies the media from being published, and he gives a strike to the user who attempted to publish the inappropriate media. If the user does this again, they are banned from MarketGator as there will be a two-strike policy for account banning.

Use Case 3: Selling Items

Actor: Seller

Description:

Jacob wants to sell his notes containing important information and chapter summaries from the textbook he read after he passed his CSC256 course in order to recuperate some of the money he spent on the textbook. However, Jacob does not have the time to go around in person and ask colleagues if they are interested in buying his notes. Jacob then discovers MarketGator and becomes impressed by the easy navigation of media that is for sale organized by major and course number. Jacob then registers a MarketGator account and uploads his textbook notes under the tags CSC and 256 and advertises them for a dollar each download. By the end of the week, Jacob has already gotten more than ten sales on MarketGator.

Use Case 4: Free Download of Digital Media

Actor: Faculty Professor

Description:

Professor John is about to start a new semester of CSC 415 and he wants to have his students use the most recently released textbook on Computer Operating Systems. However, since the textbook is $150 for a new edition, he wants to share his own digital copy that he bought with his new students. Professor John then uses MarketGator to register an account and upload the textbook under the tags CSC and 415. He chooses the option to have the digital textbook be downloaded for free so his students can easily access it. His students go on MarketGator, navigate to the CSC 415 results, register MarketGator accounts, and eventually download the newest operating systems textbook for free.

Use Case 5: Digital Media for Digital Media Trade

Actor: Buyer and Seller

Description:

Joe wants to purchase a digital movie off MarketGator, however they want to save their money. Joe messages the seller and asks if they are open to trading their personal library of tardigrade pictures for the digital movie and the seller accepts. MarketGator then facilitates the trade and sends the digital movie to the buyer and the tardigrade pictures to the seller. The buyer and sellers are satisfied, and the buyer ultimately saves money.

3. List of main data items and entities - data Glossary and Description

1. **Users**
    - Unregistered_user: A general user without an account that can only search and navigate MarketGator and view seller posts.
    - Registered_user: Inherits all functionality of unregistered users plus view all public pictures. Can post items for sale and buy items, but must be logged in.
    - Admin: Inherits all functionality of registered users, but can modify user privileges, access all content, remove posts, modify database.
2. **Picture:** Public image accessible to all users
3. **Review:** Review that users are able to write for other users after a transaction. These include a rating and a description of why that rating was given.
4. **Transaction:** Transaction information with two user IDs of buyer and seller, transaction status, and a timestamp of transaction completion date.
5. **Item:** A media item with a listing price, sub-list items such as Course ID and or Major, description, and a single or multiple low-res photos of the digital media item.
6. **Private_message:** Message between buyer and seller to exchange contact or item information to complete a transaction.
7. **Registration_record:** The record of registration of a MarketGator user that holds information such as registration date and SFSU email.

4. Initial list of Functional Requirements

1. **Unregistered User**

   1.1 Unregistered users shall be able to create a new account with valid SFSU email.

   1.2 Unregistered users shall be able to access the homepage.

   1.3 Unregistered users shall be able to browse listings.

   1.4 Unregistered users shall be able to view the details of the listings such as price, pictures of the digital media, description of the digital media, classes and professors related to the digital media, ratings, and the reviews of sellers.

   1.5 Unregistered users shall be able to sort and filter the items and the listings based on categories, price range, classes, reviews and etc.

2. **Registered User**

   2.1 Registered users shall be able to sell items.

   2.2 Registered users shall be able to buy items.

   2.3 Registered users shall be able to log in with valid SFSU emails.

   2.4 Registered users shall inherit all the access that unregistered users have.

   2.5 Registered users shall be able to rate the sellers.

   2.6 Registered users shall be able to write reviews on the items and the sellers.

   2.7 Registered users shall be able to inquire about the items by sending a message to the sellers.

   2.8 Registered users shall be able to receive responses from the sellers.

   2.9 Registered users shall be able to respond to the messages from sellers.

   2.10 Registered users shall be able to view the photos, conditions, and descriptions of the items they are selling.

   2.11 Registered users shall be able to view the rating and reviews from the sellers.

2.12 Registered users shall receive the reviews and ratings on the items that have been sold to by another registered user.

2.13 Registered users shall be able to view the rating and reviews from the buyers.

3. Administrator

3.1 Administrators shall be able to log in with valid SFSU emails.

3.2 Administrators shall inherit all the access of registered users.

3.3 Administrators shall be able to review the listings from the registered users who are currently advertising a digital item for sale.

3.4 Administrators shall be able to deny listings that violate the rules and policies of the MarketGator before they are published.

3.5 Administrators shall be able to ban the registered users who violate the rules and policies of the website.

3.6 Admins shall be able to ask feedback from registered users on new features, user experience, and suggestions to further improve the website.

3.7 Admins shall be able to create, update, remove and improve the features and overall usability and functionality of the website.

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

Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

6. Competitive Analysis

The planned advantages between MarketGator and other online marketplaces already available is enticing for San Francisco State University students and faculty. While other online e-commerce marketplace competitors charge a fee for commission on buying and selling of items, MarketGator imposes 0% fees for all trades and even supports non-currency trade by providing the feature of trading digital media for digital media. This business model supports students especially as this will be the cheapest and most accessible way for them to buy and sell digital media without having to spend their own money. Another advantage is the specific search for digital media by SFSU course number, professor, and major. This search feature will help students and faculty facilitate the findings of the digital media that they need in order to pursue their academic goals.

Marketplace	**MarketGator**	**Competitor 1: eBay Inc.**	**Competitor 2: Amazon, LLC**	**Competitor 3: AliExpress**
Fee	MarketGator takes away 0% portion of fees from sales	eBay takes away up to 12.9% of the sale price as a fee	Amazon takes away from 8% to 15% of the sale price as a fee	AliExpress takes away from 5% to 8% of the sale price as a fee
Fulfillment Method	MarketGator provides direct download of digital all items or local pickup	By courier or local pickup	Fulfillment by Amazon program or by courier	By courier

Purchase	Auction House and Buy Now	Auction House and Buy Now	Buy Now	Buy Now
Advertisement	Items are listed for free with no hidden fees	Sellers are charged when buyer purchases item	Sellers are charged when buyer clicks on item	Seller is charged the difference of their markup item price and wholesale item price
Registration	MarketGator provides absolutely free registration and service	Registering an eBay business account will impose selling fees on listing and selling of products	Basic Amazon account is free, however Amazon Prime is behind paywall to unlock all perks for buying items	Registering an account on AliExpress is free, however a commission fee is imposed when selling products
Search by SFSU course, major, professor	MarketGator is facilitates the search for digital media specialized for students and faculty	eBay does not have this feature	Amazon does not have this feature	AliExpress does not have this feature

Competitive Analysis Chart:

MarketGator vs Competitors (eBay, Amazon, AliExpress)

7. High-Level System Architecture and Technologies Used

- Server Host: Amazon Web Servers 1vCPU 2 GB RAM
- Operating System: Ubuntu Server 16.04 LTS
- Database: MySQL Community v8.0.30
- Web Server: Express 4.16.1
- Server-Side Language: JavaScript
- Additional Technologies:
    - Web Framework: Handlebars 4.2.0
    - IDE: Visual Studio Code 1.70.2
    - Web Analytics: Google Analytics 4
    - Image upload: Multer 1.4.5

8. Team Members and Roles

Patrick Celedio	Team Lead/Frontend Lead Engineer/Document Master
Michael Feger	Backend Lead Engineer
Ahmad Adnan	Github Master/Frontend Engineer
Rohit Devdhar	Backend Engineer
Nyan Ye Lin	Frontend Engineer
Ethan Lunderville	Backend Engineer

9. Checklist

1. So far all team members are engaged and attending ZOOM sessions when required
    a. **DONE/OK**
2. Team found a time slot to meet outside of the class
    a. **DONE/OK**
3. Back end, Front end leads and Github master chosen
    a. **DONE/OK**
4. 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
    a. **DONE/OK**
5. Team lead ensured that all team members read the final M1 and agree/understand it before submission
    a. **DONE/OK**
6. Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
    a. **DONE/OK**

Milestone 2

Table of Contents

1. Executive Summary

The purpose of MarketGator is to connect the San Francisco State University community and create a platform where SFSU students and faculty can buy, sell, or share for free digital media. This platform shall facilitate the process of gathering items which shall range from obtaining textbooks, finding media resources that can be used for university projects, or to share created content among students and faculty. Our technology shall empower the SFSU students and faculty to pursue their academic and personal goals by providing an e-commerce platform that will undermine the old ways of purchasing works at high prices from online vendors who charge at a premium price.

MarketGator will allow users to browse its selection and register MarketGator accounts for free in order to advertise, sell, and or buy digital media. MarketGator users will also facilitate the communication between buyers and sellers in the case users need to ask about certain items listed on the market. MarketGator will also have Administrators who shall be required to approve each media item before it becomes advertised in order to verify that it conforms to SFSU rules on digital media and other common ethical rules. Items deemed unsuitable or inappropriate shall be deleted by MarketGator Administrators before being posted.

Another component of MarketGator will be a search by major and course. We believe that this feature will help students and faculty find the digital media that they need in a very clean and organized way. This feature will also help users who are advertising their digital media for sale or for sharing become discovered in an easily accessible way by other users who are looking to find and or purchase the digital media that they may need.

As Team 05, we are dedicated to get MarketGator in production as we believe this ecommerce application will be beneficial for the students and faculty of SFSU. We believe we are the right fit for this project because we are all senior students of SFSU, and we know first-hand the challenges of searching for media for our courses. With our expertise, we want to

apply our technological skills and build MarketGator to expedite and ease the access for the required media for our classes.

2. List of main data items and entities - data Glossary and Description

1. **Users**
    - Unregistered_user: A general user without an account that can only search and navigate MarketGator and view seller posts.
    - Registered_user: Inherits all functionality of unregistered. Can post items for sale and buy items, but must be logged in.
    - Admin: Inherits all functionality of registered users, but can modify user privileges, access all content, remove posts, modify database.
        - Email: Email address of each user
2. **Reviews:**
    - **User_Review:** Review that users are able to write for other users after a transaction. These include a rating and a description of why that rating was given.
        - User_review_id: ID of review
        - User_review_text: Message body of the review
        - Reviewing_user: User who created review
        - Reviewed_user: User who is being reviewed
        - Created: Time the review was created
        - Stars: Rating system of 1 to 5 stars
    - **Product_Review:** Review that users are able to write for products after a transaction. These include a rating and a description of why that rating was given.
        - Product_review_id: ID of review
        - Product_review_text: Message body of the review

    - Item: Item ID of being reviewed
    - Sender: User who created review
    - Created: Time the review was created
    - Stars: Rating system of 1 to 5 stars

3. **Transaction:** Transaction information with two user IDs of the two MarketGator accounts transaction status, and a timestamp of transaction completion date.
4. **Listing:** A media item with a listing price, sub-list items such as Course ID and or Major, description, and a single or multiple low-res photos of the digital media item.
    - Listing_Id: Identification number to keep track of listings in database
    - Title: Title to be displayed on listing page
    - Price: Price of the item
    - Poster: SFSU Email of seller
    - Photopath: file path to photo of the item
    - Sold: Boolean value, 0 for available, 1 for sold
    - Created: Time the listing was created
    - Category: What type of item it falls under
5. **Messages**
    - **Private_message:** Message between two users to exchange contact or item information to complete a transaction.
        - message_id: id tag of message
        - message_user_id: user id of message
        - timestamp: timestamp of message

6. **Registration_record:** The record of registration of a MarketGator user that holds information such as registration date and SFSU email.
    a. Memo_id: Message Id

   b. Memotext: Message text body
   c. Sender: Person who sent the message
   d. Receiver: Person who receives the message

7. **Report:** Report for listings that violate terms of the website
   - Report_id: Id number of the report
   - Report_text: Message body of the report
   - Listing: Listing ID that is being reported
   - Sender: User who initiated the report
   - Created: Time the report was created

3. Functional Requirements

Priority 1

1. Registered users shall inherit all the access that unregistered users have.
2. Unregistered users shall be able to create a new account with valid SFSU email.
3. Unregistered users shall be able to view listings
4. Unregistered users shall be able to search listings
5. Registered users shall be able to post items for buying or selling.
6. Registered users shall be able to log in with valid SFSU emails.
7. Registered users shall be able to view their posts through dashboard.
8. Registered users shall be able to view their messages through dashboard.
9. Registered users shall be able to inquire about the items by sending a message to the sellers.
10. Registered users shall be able to receive responses from the sellers.
11. Administrators shall be able to log in with valid SFSU emails.
12. Administrators shall inherit all the access of registered users.
13. Administrators shall be able to review the listings from the registered users who are currently advertising a digital item for sale.
14. Administrators shall be able to deny listings that violate the rules and policies of the MarketGator before they are published.
15. Administrators shall be able to ban the registered users who violate the rules and policies of the website.

16. Unregistered users shall be able to view the details of the listings such as price, pictures of the digital media, description of the digital media, classes and professors related to the digital media, ratings, and the reviews of sellers.
17. Unregistered users shall be able to sort and filter the items and the listings based on categories, price range, classes, reviews etc.

- **Priority 2**

1. Registered users shall be able to view the photos, conditions, and descriptions of the items they are selling.
2. Admins shall be able to ask feedback from registered users on new features, user experience, and suggestions to further improve the website.
3. Registered users shall be able to respond to the messages from sellers.

- **Priority 3**

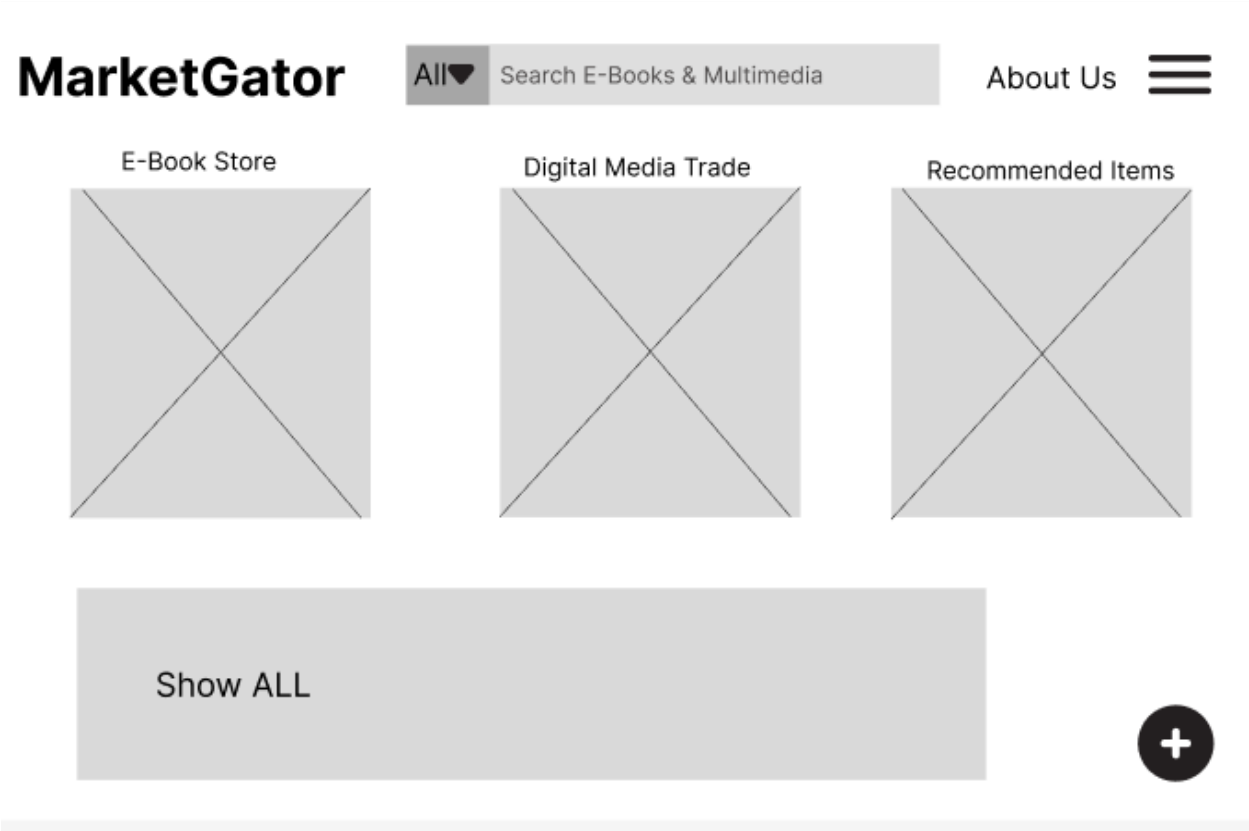
1. Registered users shall be able to rate the sellers.
2. Registered users shall be able to write reviews on the items and the sellers.
3. Registered users shall be able to view the rating and reviews from the sellers.
4. Registered users shall receive the reviews and ratings on the items that have been sold to by another registered user.
5. Registered users shall be able to view the rating and reviews from the buyers.
6. Admins shall be able to create, update, remove and improve the features and overall usability and functionality of the website

4. UI Storyboards for each main use case

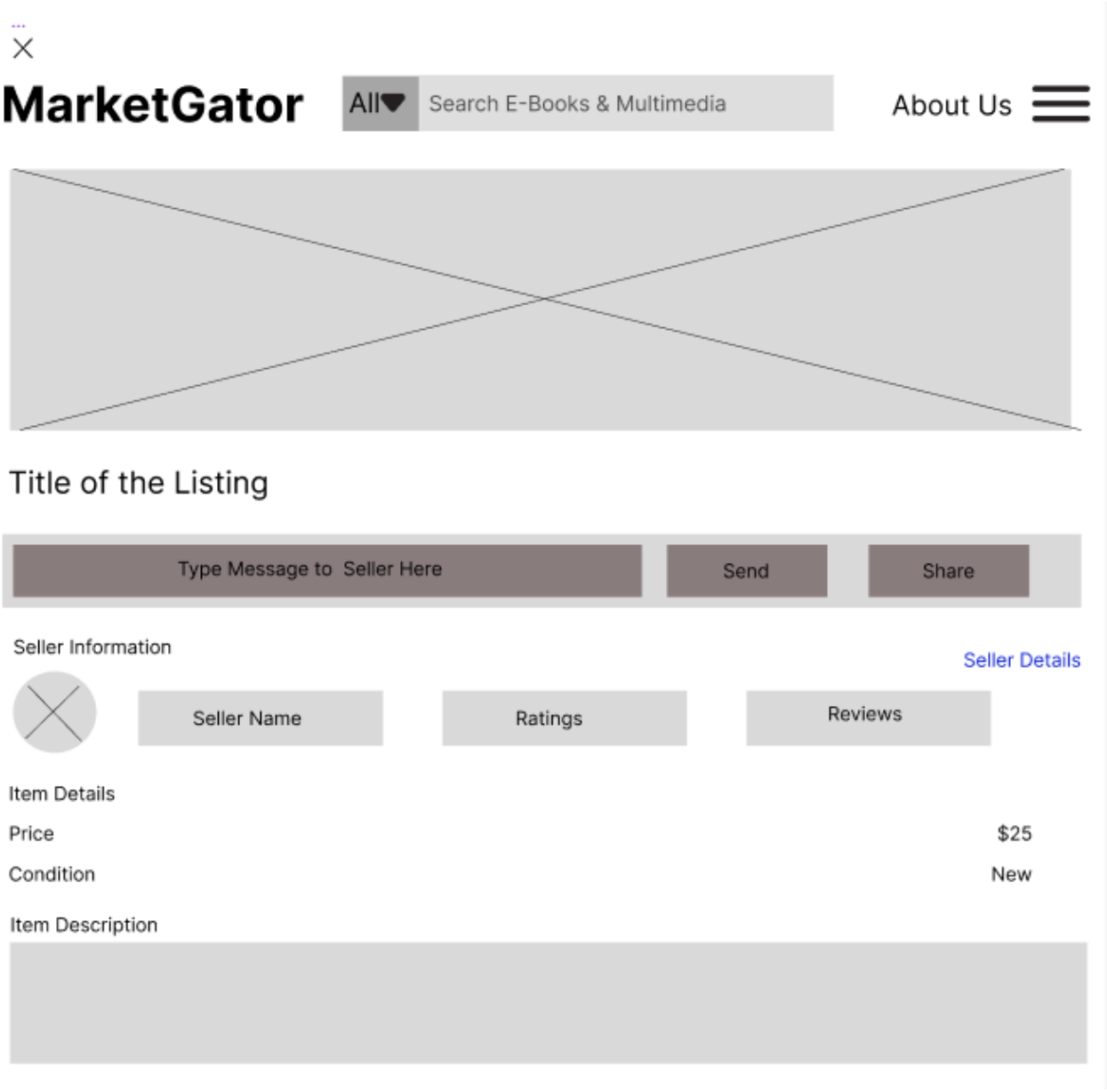
Use Case: MarketGator Registration

In this case, Jane discovers MarketGator. She browses for the first time, views a listing that she is interested in, and ultimately registers to the site in order to message the seller.

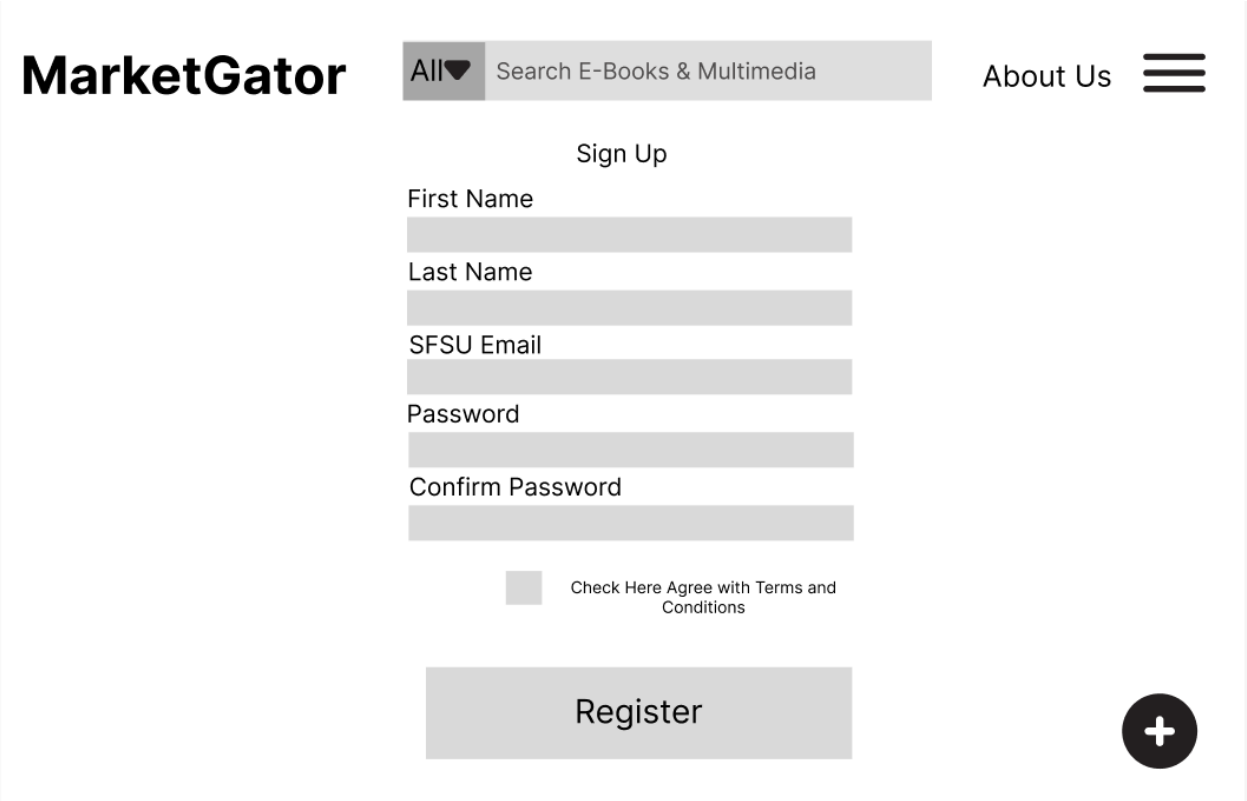
Jane is trying out MarketGator for the first time. She is browsing the homepage and looking at all the items.



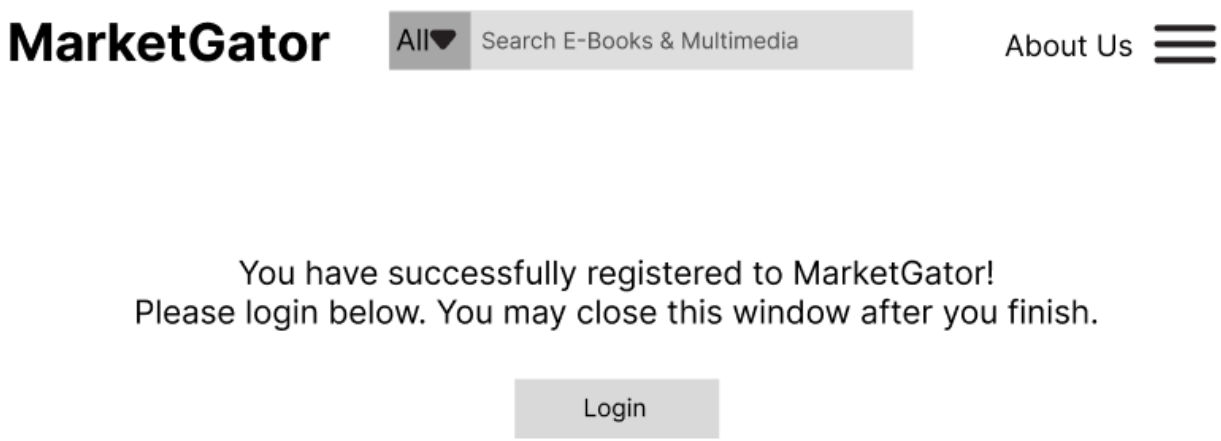
Jane clicks on a listing to view an item she is interested in. MarketGator displays the webpage of the listing. At this point, she is very interested and has questions for the seller.



When Jane wants to message the seller, she clicks the “Type Message to Seller Here” button. MarketGator responds by opening a new tab and showing them the webpage to register to MakerGator.



After Jane registers, she is shown a screen that confirms that she registered successfully and is prompted to login to her new account to resume her business.

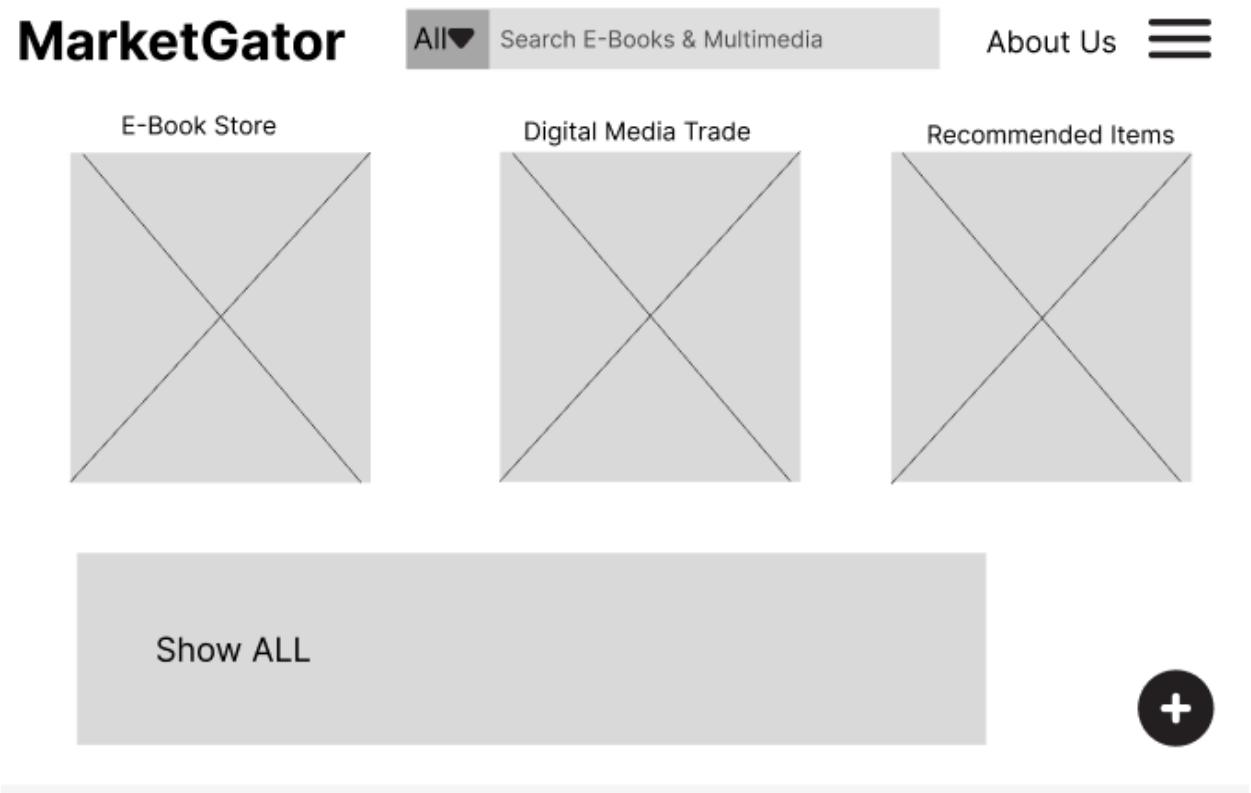


At this point, Jane messages the seller regarding information about their listing.

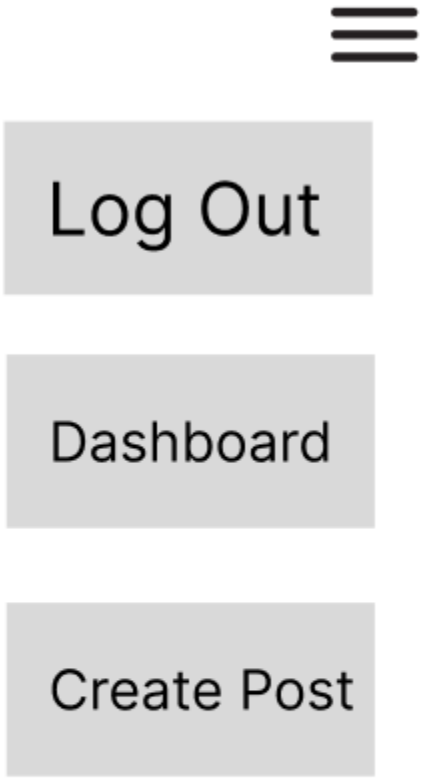
Use Case: MarketGator Selling Items

Jacob is a SFSU student who wants to sell his notes containing important information and chapter summaries from the textbook he read after he passed his CSC256 course in order to recuperate some of the money he spent on the textbook. However, Jacob does not have the time to go around in person and ask colleagues if they are interested in buying his notes. Jacob then discovers MarketGator, becomes impressed by the platform, and then registers a MarketGator account.

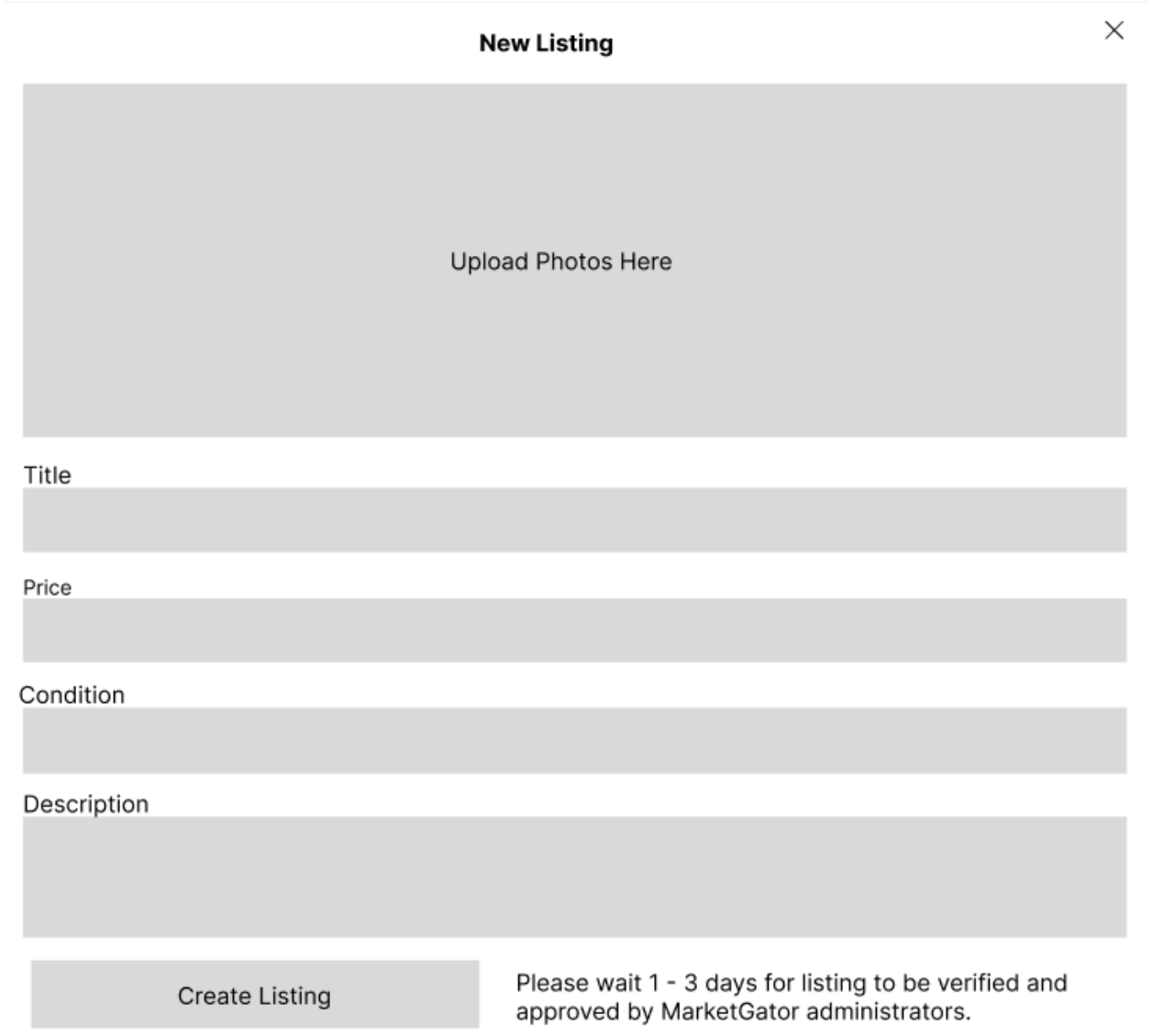
Now that Jacob has a registered account on MarketGator, he is ready to post a listing for his digital notes. On the MarketGator homepage, he clicks on the hamburger button on the top right corner of the webpage.



When Jacob clicks the hamburger button, a drop down menu appears showing the functions his account is able to access. Jacob clicks on ‘Create Post’ in order to begin creating the listing for his digital item.

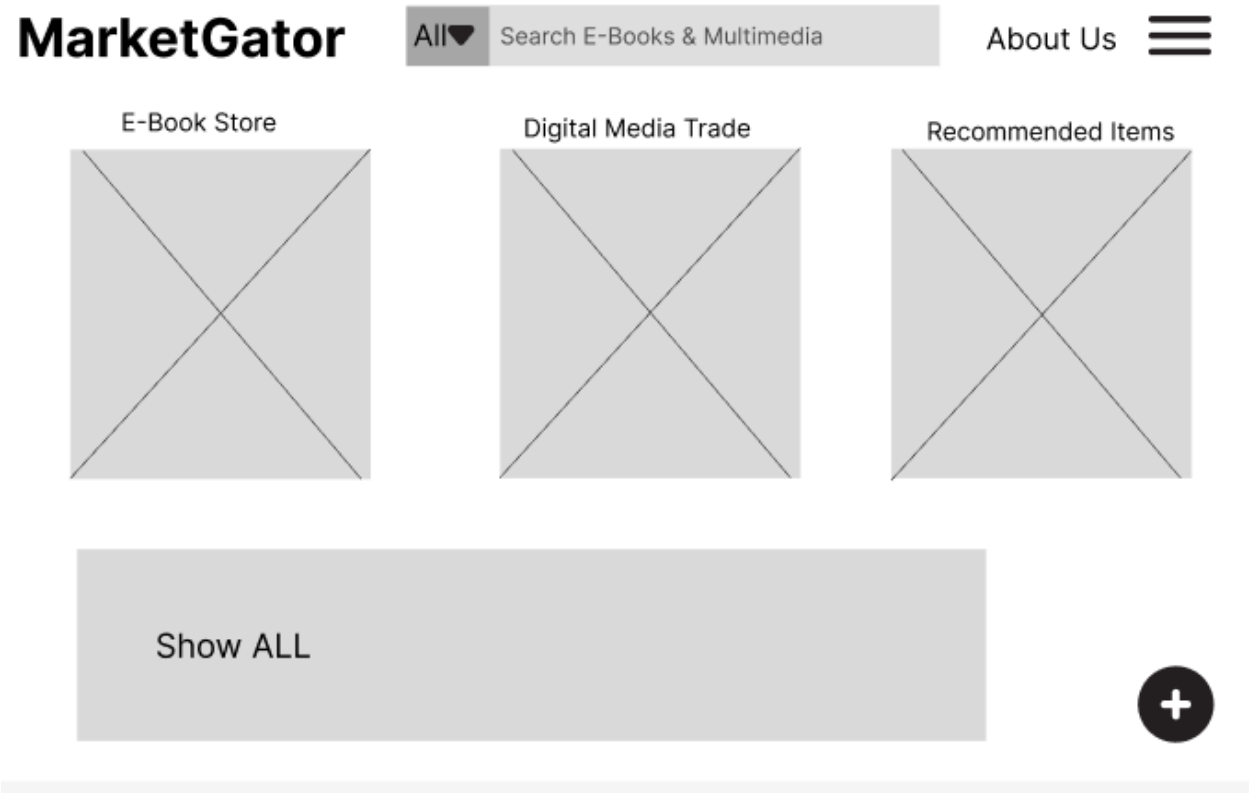


After Jacob clicks the ‘Create Post’ button, he is brought to a new page called ‘New Listing’ which will be a form that he has to fill out such as title of content, price, condition, description, and photos. When Jacob is finished filling out the form, he will have to click the ‘Create Listing’ button which will have this listing be sent to the queue of listings waiting to be approved. Next to this button explicitly states that he must wait one to three days for his listing to be verified and approved by MaketGator administrators. At this point, Jacob has finished work on creating a listing and will wait on MarketGator to review his listing.

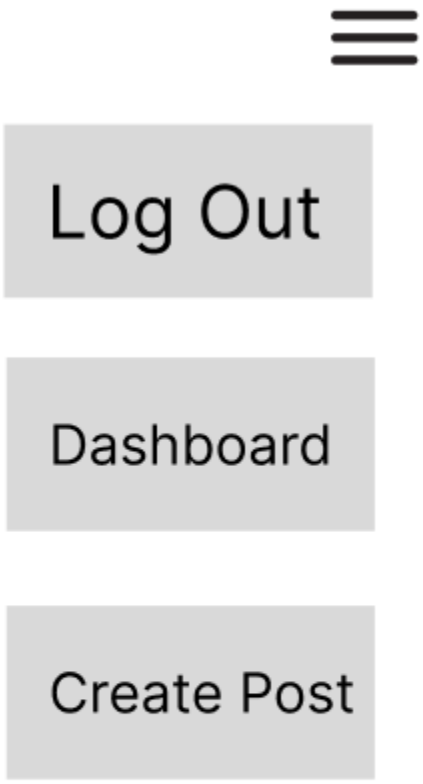


Use Case: MarketGator User Uses Dashboard to Access Personal Listings and Messages

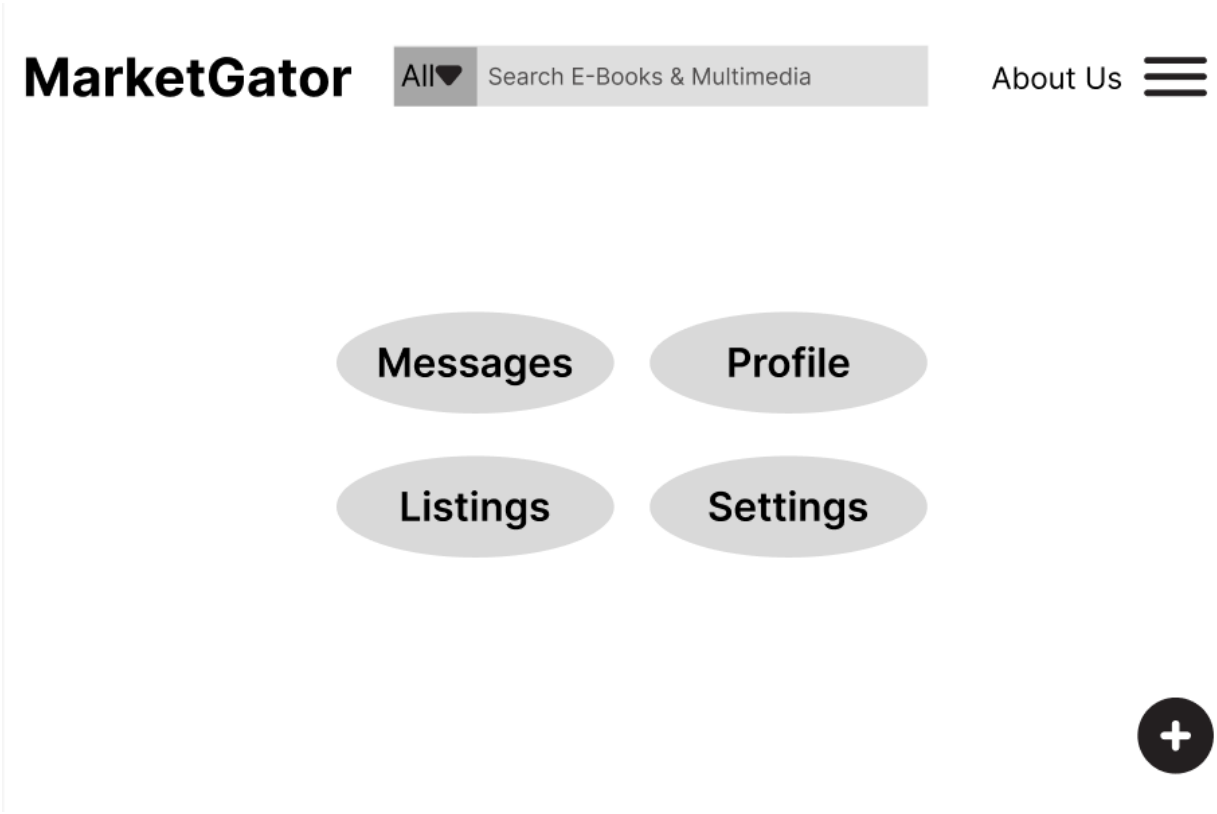
The next day after creating his listing that he sent for approval to MarketGator, Jacob wants to check the status of it. On the homepage, he clicks on the hamburger button on the top right corner of the page.



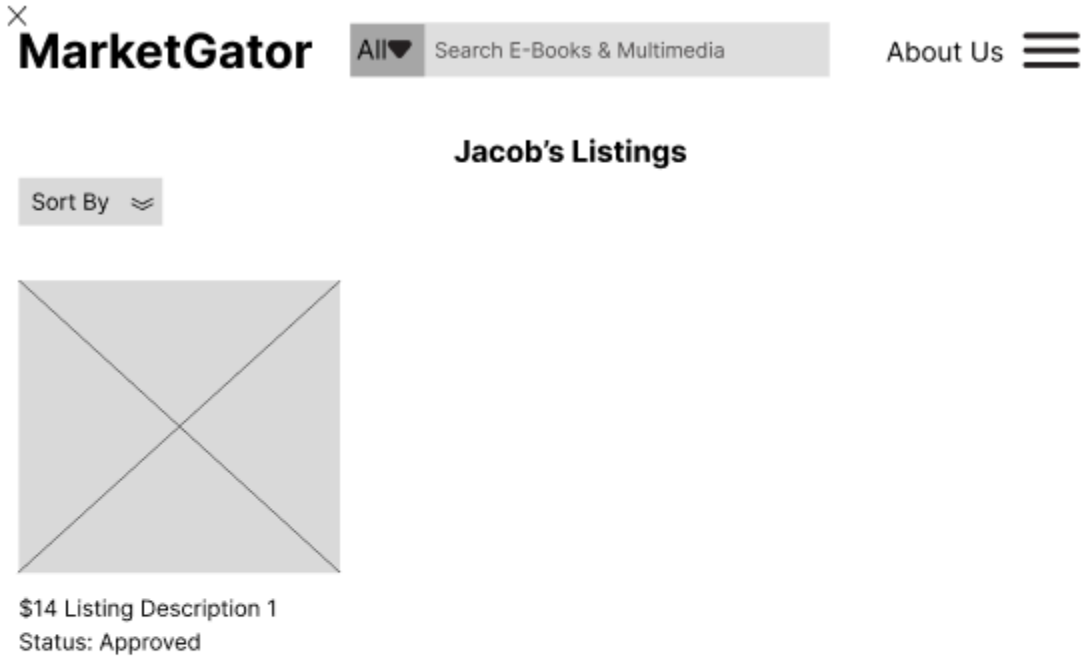
At this point, Jacob clicks on the ‘Dashboard’ button to view information related to his personal account on MarketGator.



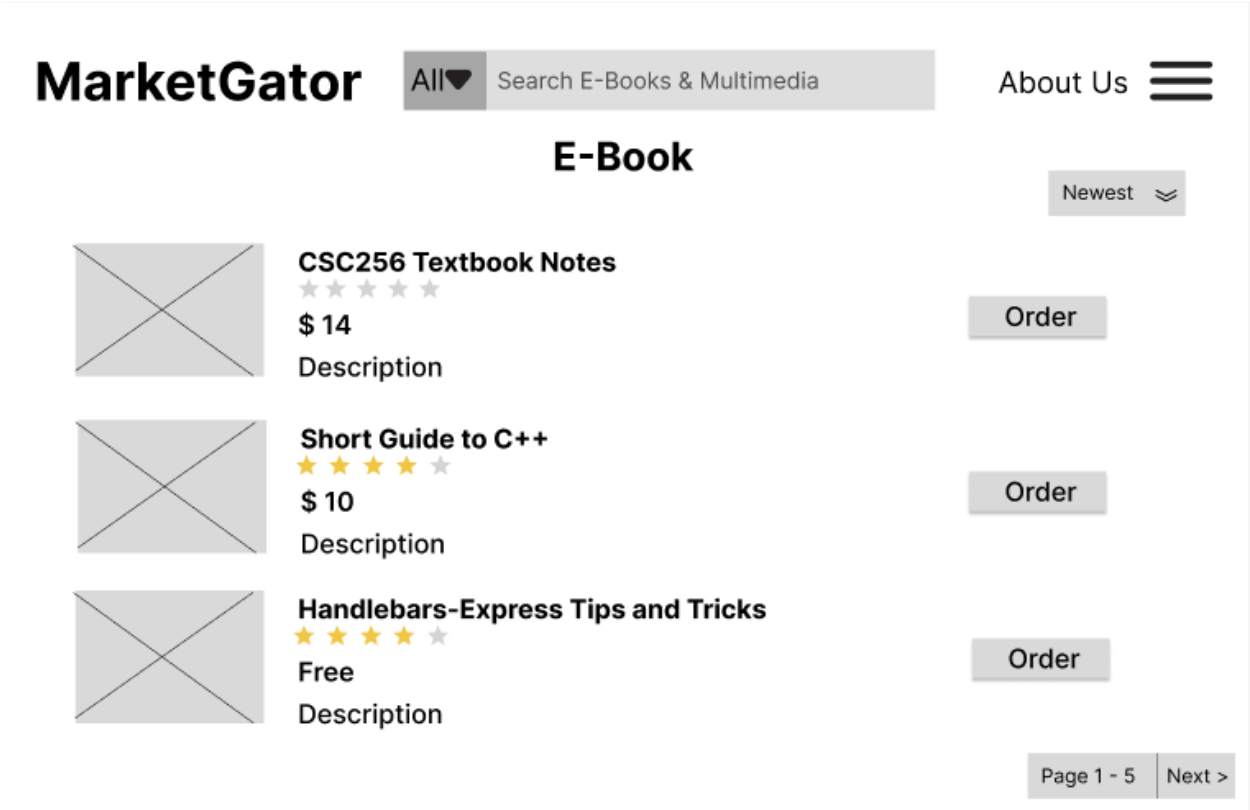
From here, Jacob is able to access functions such as viewing his personal messages, listings, profile, and settings. Since Jacob is curious about the status of his listing that he created, he clicks on the ‘Listings’ button.



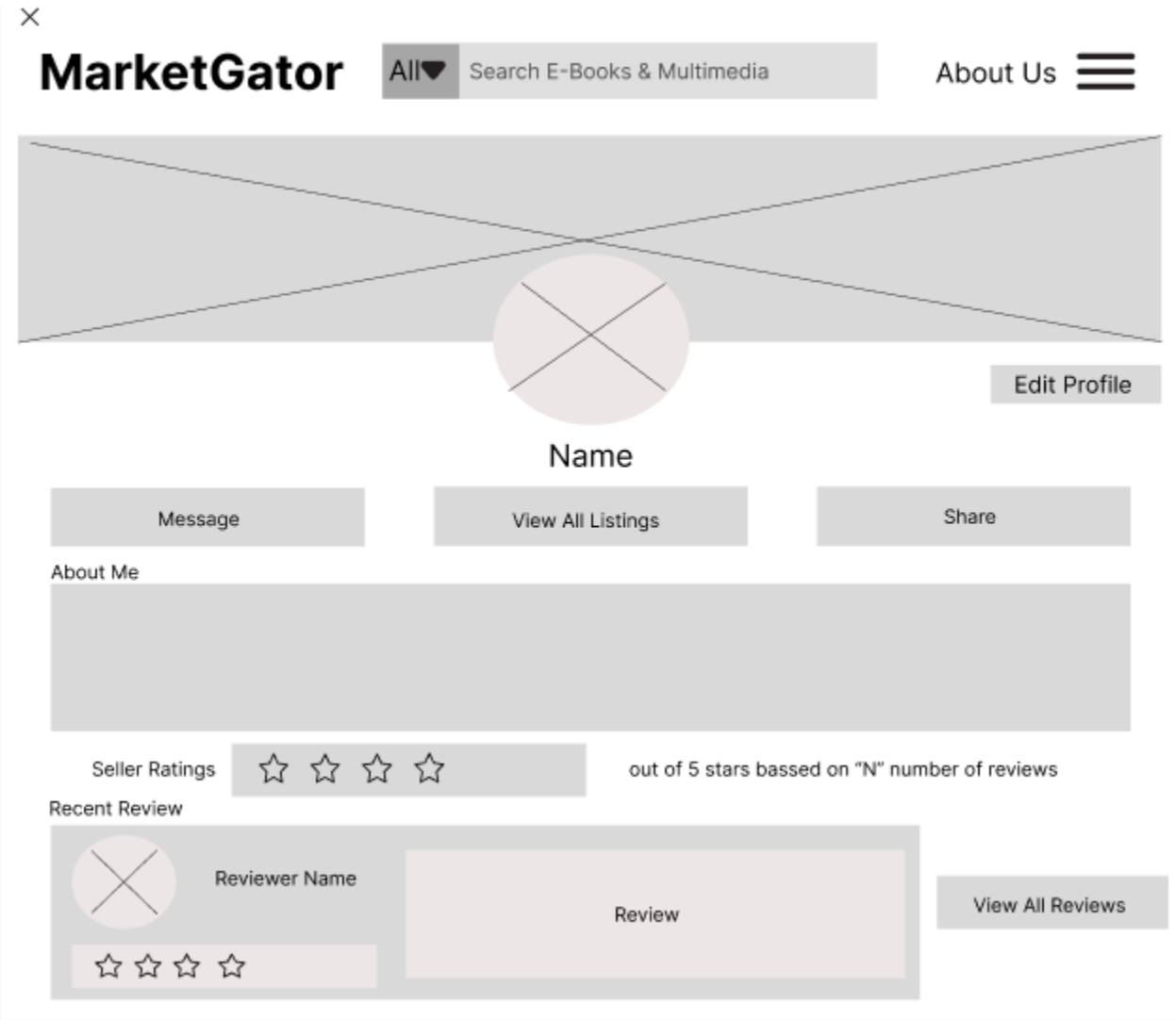
After clicking the ‘Listings’ button, Joe sees that his listing that posted has been approved. He discovers this as underneath his personal listing, there is a status field with the word ‘Approved’ next to it meaning that it has been reviewed and is now available for the public to access on MarketGator.



Jacob could verify this by looking up his listing through the MarketGator search function and he sees his listing is first as the listings are displayed by ‘Newest’.

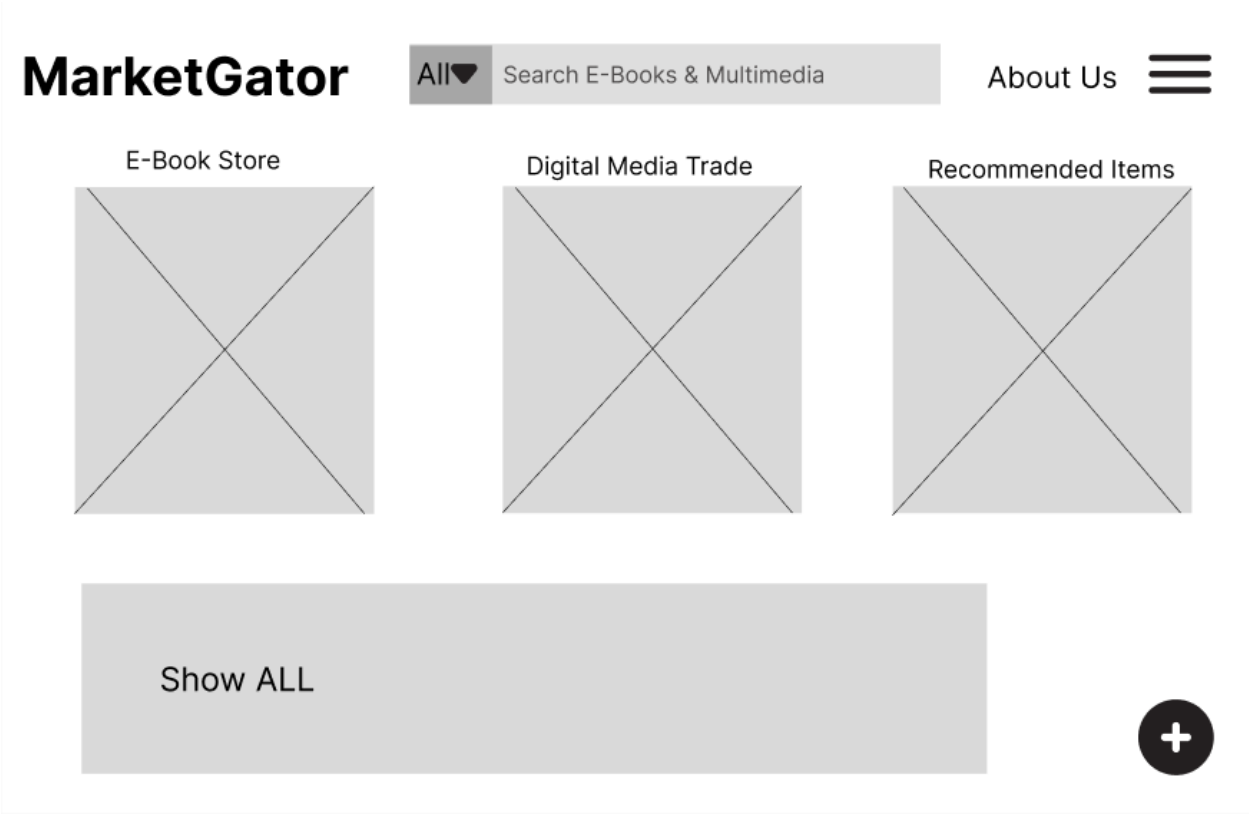


Jacob is excited that his listing has become public on MarketGator. He then thinks about his user profile so he goes back to his dashboard to view and edit his MarketGator account. He clicks the ‘Profile’ button inside his dashboard. It brings him to this page, and from here Jacob is able to view and edit his profile.

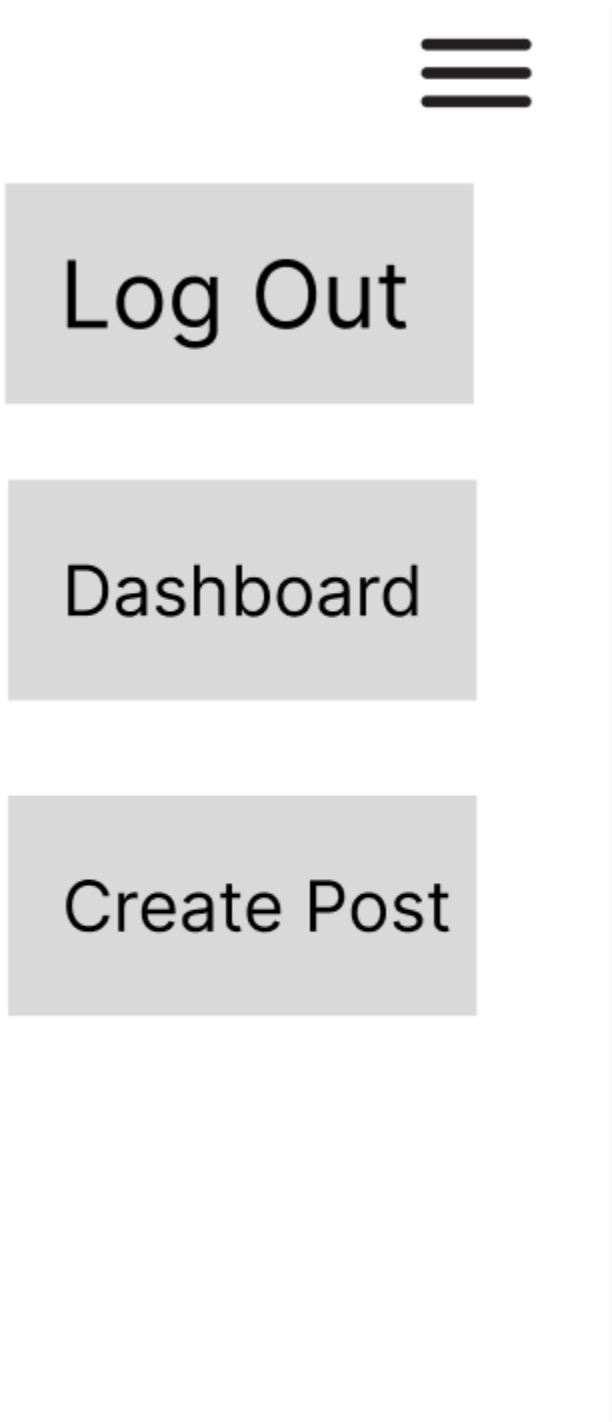


Use Case: MarketGator User Moderation and Administration

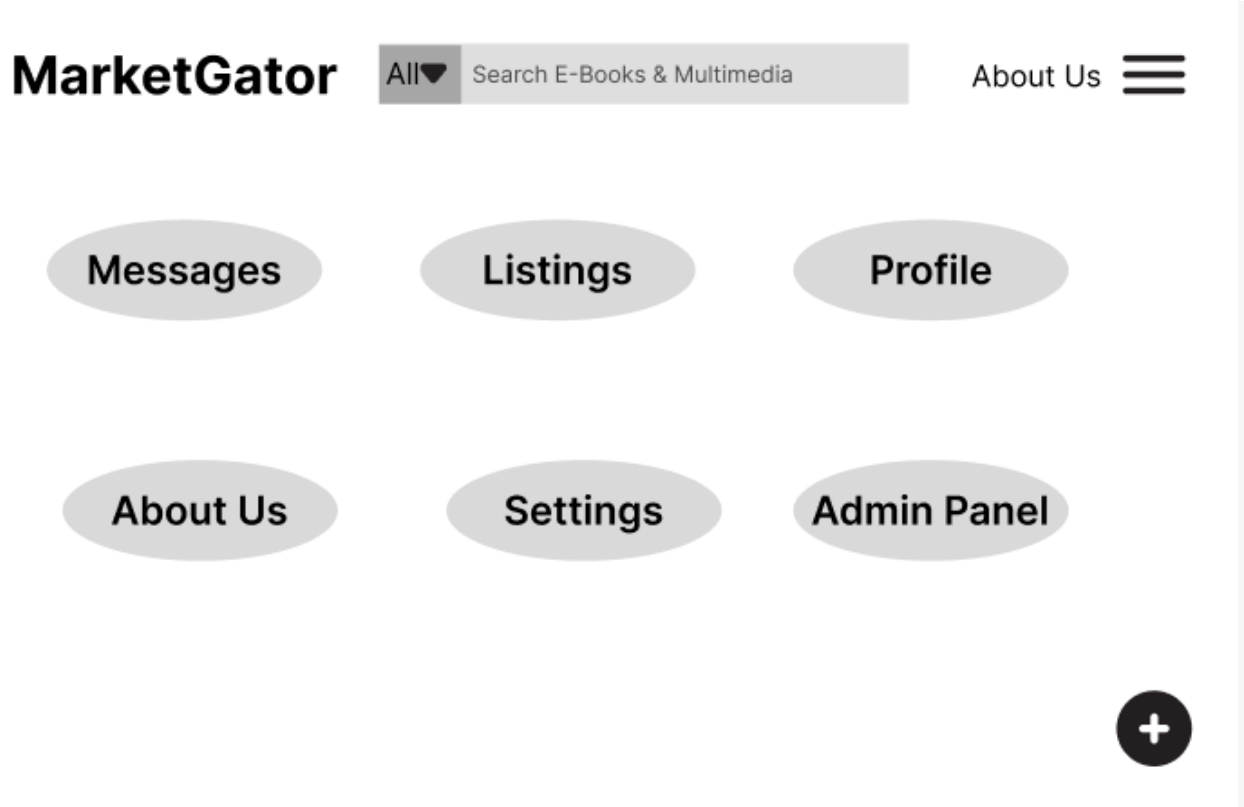
Joe is working as an administrator for MarketGator. He is logged in to MarketGator with his registered account that also has administrator privileges. Joe is required to check the queue of listings that are waiting to be published.



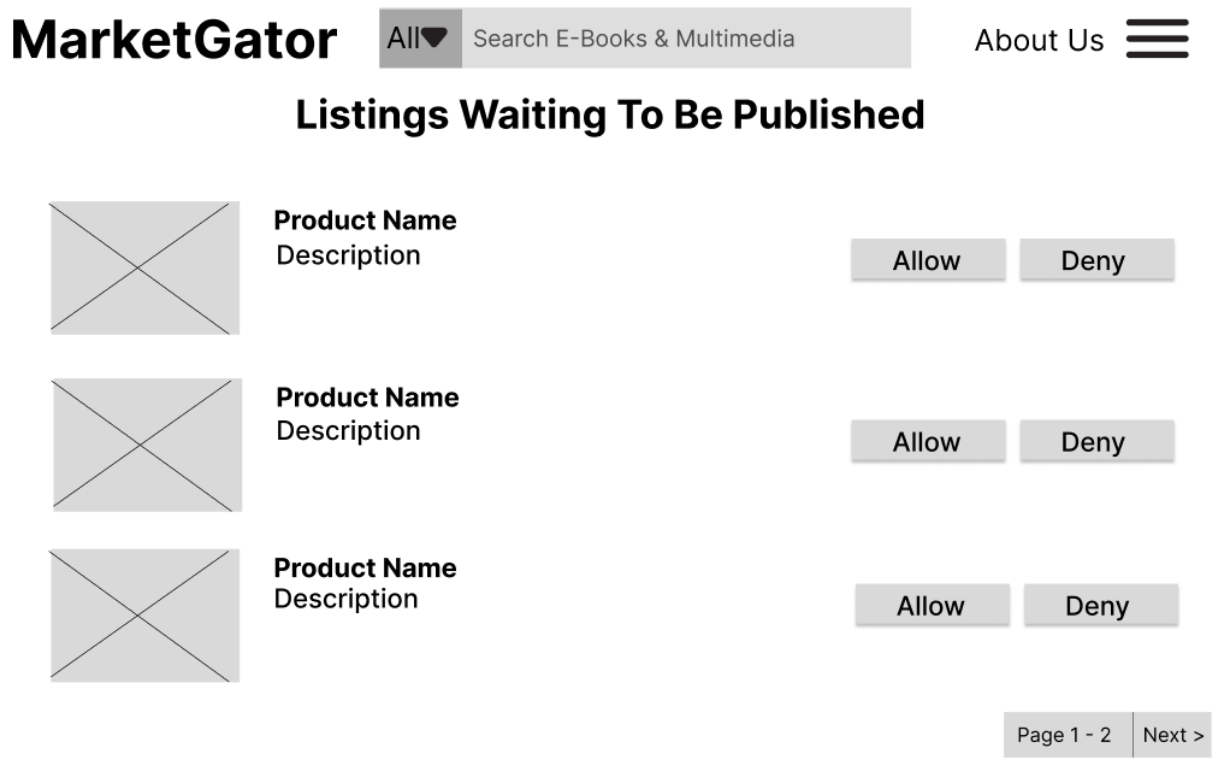
Joe clicks on the hamburger button on the top right corner of the screen. A drop down navigation menu appears. Joe clicks on ‘Dashboard’ in order to access his account’s personal MarketGator tools.



Joe is now viewing his personal dashboard for his MarketGator account. Since his MarketGator account has administrator privileges, he is able to see the Admin Panel button in his dashboard. Joe clicks on the Admin Panel button.

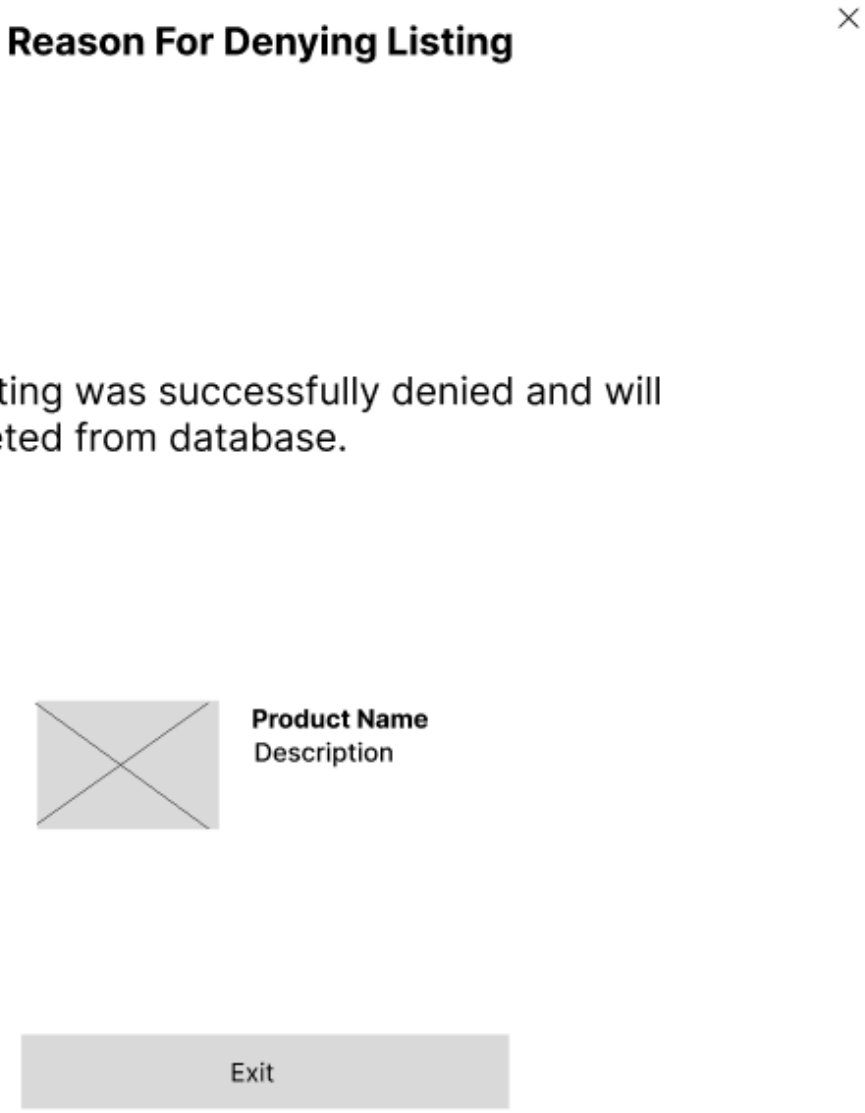


After clicking the Admin Panel button, Joe is brought to this page where he is able to view the queue of listings waiting to be published. Joe is able to click and view the listings as if they are regular MarketGator listings however they are not public yet. He clicks each listing and verifies their content, and if they are within the standard of SFSU and MarketGator content rules, then he ultimately clicks the ‘Allow’ button attached to that listing in order to make it public for MarketGator.



When Joe views a listing that breaks the standard of SFSU and MarketGator content rules, he ultimately clicks the ‘Deny’ button. This brings up a new tab prompting the administrator to send a response back to the account of that listing on why their listing has been denied.

Once Joe completes the deny listing form, he is prompted by a new screen which confirms his action and that the listing was deleted from the database. Joe can click the ‘Exit’ button which will close the tab on which this window was open and resume his work.



5. High Level Architecture, DatabaseOrganization Summary

High-Level System Architecture and Technologies Used

- Server Host: Amazon Web Servers 1vCPU 2 GB RAM
- Operating System: Ubuntu Server 16.04 LTS
- Database: MySQL Community v8.0.30
- Web Server: Express 4.16.1
- Server-Side Language: JavaScript
- Additional Technologies:
    - Web Framework: Handlebars 4.2.0
    - IDE: Visual Studio Code 1.70.2
    - Web Analytics: Google Analytics 4
    - Image upload: Multer 1.4.5

Database Organization

- User
    - User_id
    - Email
    - Password

- Listing
    - Category
    - Listing_ID
    - Title
    - Price
    - Poster
    - Photopath
    - Sold
    - Created
    - Itemname

- Message
    - Memo_id
    - Memotext
    - Sender
    - Receiver

- Feedback
    - Feedback_id
    - Feedback_text
    - Sender
    - Created

- Report
    - Report_id
    - Report_text
    - Listing
    - Sender
    - Created

- Product_Review
    - Product_review_id
    - Product_review_text
    - Item
    - Sender
    - Created
    - Stars

- User_review
    - User_review_id
    - User_review_text
    - Reviewing_user
    - Reviewed_user
    - Created
    - Stars

Media Storage

Media items will be kept in file systems rather than DB BLOBs.

Search/Filter Architecture and Implementation

Listings will have tags along with categories. Users will search in a category, and the tags will be used with AND with the category. Terms that will be searched are titles, categories, and tags such as the author, book name, and class that it is used in. SQL and %like will be used to perform this function.

6. Key Risks

Team 05 has identified actual and specific risks in our current work such as:

- MySQL setup and development knowledge
    - The back-end members of Team 05 have greater knowledge with MySQL development so they have been sharing their knowledge and resources they use as references to the rest of the team. They shall also be required to answer questions and offer help based on MySQL and database design.
    - Document Milestone 2 Vertical Prototype so all team members can learn how to create their own pages
- Team members have different schedules from each other
    - To work around our schedules, we have managed to find a short pocket of time within our weeks where we meet on Zoom for half an hour to discuss and update ourselves on the status of our tasks.
    - However, there may be times where we may not make a meeting. Therefore, we have established a Discord server to communicate through messages and also a Trello team board for the team to access and update for when we want to know the status of a certain task.
    - Reduce Priority 1 list to bare minimum if necessary

7. Project Management

For Milestone 02, Team 05 has been meeting and discussing every Monday to sync up and plan which project requirements are needed to be fulfilled before the deadline. We have divided each required task based on the skill needed. For example, the front-end team is handling the user interface mock-ups while the back-end team is handling the high level architecture and database organization.

To keep track of status for each task, I have created a Trello project which contains the Trello board for Milestone 02 and its required tasks to be fulfilled. Each member shall be required to update the board when they are finished, and Team Lead will check periodically to ensure that we are on track on fulfilling Milestone 02 on time.

For future tasks, we plan to continue using Trello, and also other tools such as Zoom and Discord to stay in communication with each other. We will also continue using team productivity tools such as Figma and Google Docs to work tasks together in real-time or individually as it has proven to be worthy to Team 05’s productivity so far.

Milestone 3:

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

Team number: 05

Meeting date: November 14, 2022

Summary of feedback on UI (record all pages that need revision):

Feedback of the UI was negative as most of our basic components for homepage, search results, search details, and message to the application and database were not fully implemented or at least hard coded into the application.

Pages that need revision are:

- main.hbs
- login.hbs
- registration.hbs
- searchresults.hbs
- dashboard.hbs
- inbox.hbs

Summary of feedback on code and architecture:

For code, there are files that are missing headers and it needs more comments in order to ensure that the code is easy to maintain and easy to contact the person who wrote the code.

For architecture, we need to divide our code into partitions between front-end and back-end. Therefore, we have to create front-end and back-end folders in our application directory which will store the appropriate code based on its function.

Summary of feedback on github usage

For GitHub usage, the team has to be more descriptive in their message commits. Even though we may have said, “Fixed bug in home.hbs”, it would be more helpful to be more distinct on what bug we exactly fixed.

Summary of feedback on DB

For the database, feedback was good. We mostly have all the tables and data that we need for the application. One thing missing was a column for listings-table that determines if the listing is live or not.

Summary of feedback on teamwork and risk management

For feedback on teamwork and risk management, it was mostly negative as our site did not reflect what we were reporting to the professor before M3. We will mitigate this by implementing SCRUM meetings and attempting to use Trello to track our milestones.

Confirm that you have done architecture review to check that developers adhere to MVC pattern, coding style, minimal agreed documentation etc.

We have done an architecture review and the MVC pattern is not there yet.

Record if OK or list the issues found. Request developers follow up on corrections and follow up later by doing code reviews

- Project directory does not follow MVC format
- Back-end and front-end files are all over the place

List below agreed upon P1 list of features for final delivery which constitute product plan. NOTE: after this meeting the team focuses solely on this P1 list of features, e.g. the development is in "feature freeze mode". All listed P1 features (no more no less) MUST be delivered in usable way, free of bugs

1. Homepage
2. Search results
3. Search details
4. Message to the application

Any other comments and issues

We must not continue our current work organization and start becoming more organized and incorporate more communication.

Check Point (CP) if given, DUE: November 22, 2022

IMPORTANT for selection of P1 features: analyze what needs to be done, prioritize based on two factors: a) importance for the product/user and b) cost/ability to deliver it in given schedule. Based on this come up with the plan (list of P1 features) then execute it. After this the team is in "feature freeze" mode, focus is on P1 features only

Milestone 4

Table of Contents

1. Product Summary

Name of product:

MarketGator

Description of Product:

The purpose of MarketGator is to connect the San Francisco State University community and create a platform where SFSU students and faculty can buy, sell, or share for free digital media. This platform shall facilitate the process of gathering items which shall range from obtaining textbooks, finding media resources that can be used for university projects, or to share created content among students and faculty. Our technology shall empower the SFSU students and faculty to pursue their academic and personal goals by providing an e-commerce platform that will undermine the old ways of purchasing works at high prices from online vendors who charge at a premium price.

Itemized list of all majors committed functions:

1. Registered users shall inherit all the access that unregistered users have.
2. Unregistered users shall be able to create a new account with valid SFSU email.
3. Unregistered users shall be able to view listings
4. Unregistered users shall be able to search listings
5. Registered users shall be able to post items for buying or selling.
6. Registered users shall be able to log in with valid SFSU emails.
7. Registered users shall be able to view their posts through the dashboard.
8. Registered users shall be able to view their messages through the dashboard.
9. Registered users shall be able to inquire about the items by sending a message to the sellers.
10. Registered users shall be able to receive responses from the sellers.
11. Administrators shall be able to log in with valid SFSU emails.
12. Administrators shall inherit all the access of registered users.

13. Administrators shall be able to review the listings from the registered users who are currently advertising a digital item for sale.
14. Administrators shall be able to deny listings that violate the rules and policies of the MarketGator before they are published.
15. Administrators shall be able to ban the registered users who violate the rules and policies of the website.
16. Unregistered users shall be able to view the details of the listings such as price, pictures of the digital media, description of the digital media, classes and professors related to the digital media, ratings, and the reviews of sellers.
17. Unregistered users shall be able to sort and filter the items and the listings based on categories and classes etc.

Unique feature of product:

A feature that makes MarketGator unique from competitors is its straightforward user interface.

URL to MarketGator:

Copy and paste in URL field of web browser:

http://54.218.4.54:3000/ **DECOMMISSIONED LINK**

2. Usability Test Plan

Test objectives:

For the usability test plan, we shall focus on item listing posting by the user. The goal of this usability test plan is to evaluate how feasible posting items for viewing on MarketGator is to use by the user through metrics such as effectiveness, efficiency, and satisfaction. Our target audience ranges from young adults to senior adults who work in academia. We want to ensure users of MarketGator that they can achieve the goals that they seek from the product in a manner that requires the least resources spent by them. And ultimately provide a comfortable user experience which maximizes the satisfaction of the user.

Test background and setup:

For the usability test plan, the user will enter the MarketGator site using the latest version of Google Chrome, which is 108.0.5359.98 as of December 8, 2022. This will ensure that MarketGator is being tested by the user on the latest and most popular cross-platform web browser and that the results coming from this usability test is the most accurate and most up-to-date.

Users will start already logged in to a registered MarketGator account. This eliminates any chances of the users running into any issues related to user registration and login, and saves time for the user testing the posting of items feature. However, each test user will have their own separate registered account in order to keep track of which user posted which listing.

The intended users of this usability test will be adults who study or work at San Francisco State University. The reason why these are our intended users is because MarketGator is specifically crafted for the SFSU community.

Usability task description:

Please create a listing indicating an item to advertise on MarketGator.

Evaluation of effectiveness:

To measure effectiveness, will make sure that the available functions allow the completion of creating a listing. Then we will measure what percentage of people completed the task in under 1 minute.

Evaluation of efficiency:

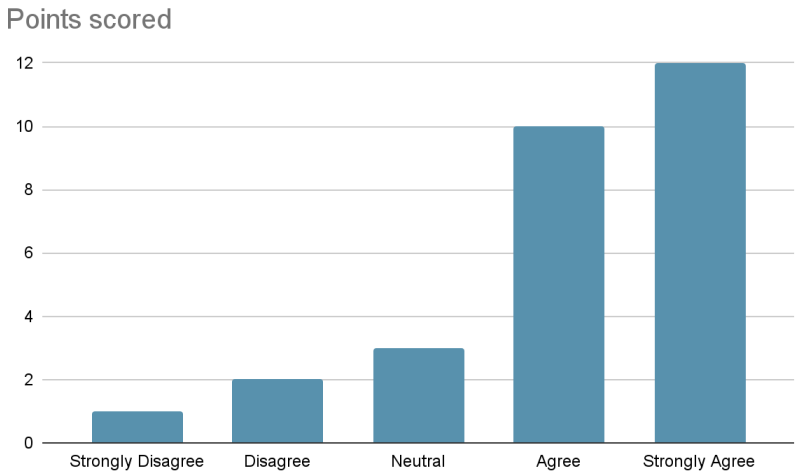
We will measure efficiency based on how many clicks it takes to reach “create listing” and the total time it takes from the dashboard to a successful listing post.

Evaluation of user satisfaction:

	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
The process of finding “Create Listing” is straightforward					

Your comments are appreciated to further improve our product
Enter Comment Here..

Percentage of users who agree or strongly agree	Average among all users	Standard deviation from average



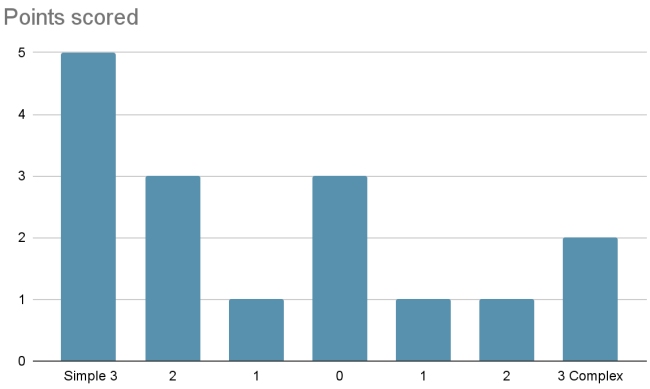
Not based on actual data. For demonstration only.

I found data entry into the input variables (title, description, etc) to be:

Simple	3	2	1	0	1	2	3	Confusing

Your comments are appreciated to further improve our product
Enter Comment Here..

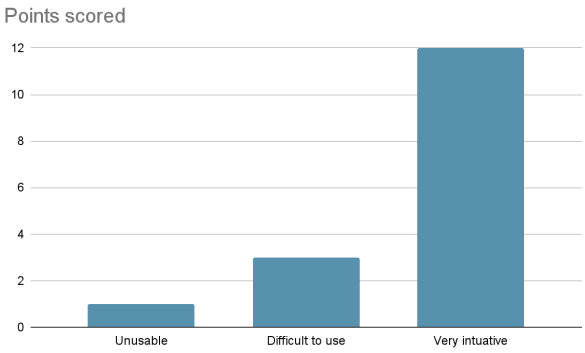
Percentage of users who selected towards simple	Average among all users	Standard deviation from average



Not based on actual data. For demonstration only.

	Unusable	Difficult to use	Very Intuative
Rate the look and feel of the listing creation page			

Percentage of users who selected Very Intuative	Average among all users	Standard deviation from average



Not based on actual data. For demonstration only.

3. QA Test Plan

Test objectives: To identify areas of improvement in the user experience of MarketGator
HW and SW setup:

Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz
Operating system: Windows 10 version 21H2
Browser: Google Chrome Version 108.0.5359.98 (Official Build) (64-bit)
Product URL: http://54.218.4.54:3000/create-listing

Feature to be tested: Listing post
QA Test plan:

Use case: User is logged in and creates a listing

Number	Description	Test Input	Expected output	Pass/Fail
1	Create listing post	Title: test1 Category: Book Item Description: Test1 of listing post Price: 20 Upload photo: png/jpg file	User is redirected to their new listing page	Pass

Number	Description	Test Input	Expected output	Pass/Fail
2	Check database for posted listing that has just been posted	select * from listing WHERE listing.title="test1"	Listing appears in database with all fields correctly set from test case 1. A photopath is created to the /images/products destination and a thumbnail is created in /images/products/thumbnails	Pass

Number	Description	Test Input	Expected output	Pass/Fail

3	Check if item is NOT live	select listing.live from listing WHERE listing.title="test1"	Live: 0	Pass

Number	Description	Test Input	Expected output	Pass/Fail
4	Create listing post	Title: test2 Category: Book Item Description: Test1 of listing post Price: 20 Upload photo: Pdf file	Only images are allowed	Pass

4. Code Review

Create Listing Code Review

MF Michael Feger
To Patrick Celedio

Reply | Reply All | Forward | ...
Fri 12/9/2022 5:42 PM

Hello Patrick,

I am requesting a code review for the given documents involving the code used to create a listing on MarketGator.

/application/back-end/models/Listings. Function name: ListingModel.create()
Link: https://github.com/CSC-648-SFSU/csc648-03-fa22-team05/blob/michaelfeger/application/back-end/models/Listings

/application/back-end/routes/Listing.js
Link: https://github.com/CSC-648-SFSU/csc648-03-fa22-team05/blob/michaelfeger/application/back-end/routes/Listing.js

I appreciate your input, please let me know if there should be any changes.

Michael Feger
Team 5 Backend Lead

```
/*
  Author: Michael Feger
  Purpose: Contains routes relating to listings
*/

var sharp = require('sharp');
var multer = require('multer');
var express = require('express');
const path = require('path');
const fs = require('fs')
var router = express.Router();
const {getListingById} = require('../middleware/listingmiddleware');

/*
  @ Patrick Celedio
  - getListingbyId is not used, we should remove or implement this line before sending to production
 */


var ListingModel = require('../models/Listings');
var ListingError = require('../errors/ListingError');
const { successPrint, errorPrint } = require('../middleware/errormiddleware');

/*
  @ Patrick Celedio
  - successPrint is not used, we should remove or implement this line before sending to production
 */

var crypto = require('crypto');

router.get('/', (request, response) => {
  response.render('unauthenticated/dashboard');
});

```

```
36
37  //Create multer storage variable, set image destination with randomized file name
38  var storage = multer.diskStorage({
39    destination: function(req, file, cb){
40        cb(null, "../application/front-end/public/images/products");
41    },
42    filename: function(req,file,cb){
43        let fileExt = file.mimetype.split('/')[1];
44        let randomName = crypto.randomBytes(22).toString("hex");
45        cb(null, `${randomName}.${fileExt}`);
46    }
47  });
48
49  var uploader = multer({storage: storage,
50    fileFilter: function (req, file, callback) {
51      var ext = path.extname(file.originalname);
52      if(ext !== '.png' && ext !== '.jpg' && ext !== '.jpeg') {
53          console.log("Not an image file");
54          return callback(new Error('Only images are allowed'))
```



```
55      }
56      callback(null, true)
57  },
58  limits:{
59      fileSize: 1024 * 1024 //Limit image size to not overload database
60  }
61  });
62
63  /*
64    @ Patrick Celedio
65    - Good use of multer and limiting picture resolution
66   */
67
```

Listing.js code review

```
73          }
74
75          return db.execute(baseSQL)
76          .then(([results, fields])=>{
77
78              req.searchResult = results;
79              req.searchTerm = "";
80              req.category = "";
81
82              return results;
83          })
84          .catch((err) => Promise.reject(err));
85  }
86
87  ListingModel.getListing = (idlisting) => {
88      let baseSQL =
89      `SELECT * FROM listing
90      WHERE idlisting=`+idlisting+``;
91
92
93    return db.execute(baseSQL, [idlisting])
94    .then(([results, fields]) => {
95        return Promise.resolve(results);
96
97    })
98    .catch(err => Promise.reject(err))
99  }
```

localhost:59395/b35833a7-ac77-4fe5-afc4-1f122fe591ca/

12/9/22, 6:18 PM Untitled-1

```
100
101 ListingModel.test = function() {
102
103         let baseSQL = "SELECT * FROM listing WHERE listing.live=1;";
104         return db.execute(baseSQL)
105
106 .then(([results, fields])=>{
107
108     console.log(results);
109     return results;
110
111 })
112 .catch((err) => Promise.reject(err));
113 }
114
115 /*
116     @Patrick Celedio
117     Code is clean and looks sound. Perhaps before the final push to production we could
    remove the ListingModel.test function.
118 */
119
120 module.exports = ListingModel;
```

Listings; SQL code review

RE: Create Listing Code Review

PC

Patrick Celedio
To Michael Feger

Reply | Reply All | Forward | ...

Fri 12/9/2022 6:22 PM

Listings.pdf
205 KB

Listing.js.pdf
282 KB

Hello Michael,

Thank you for submitting your code for review. I have reviewed your code and left appropriate comments.
I have attached these comments as PDF files. Please let me know if you have any questions or feedback.

Best,
Patrick
Team 05 Lead

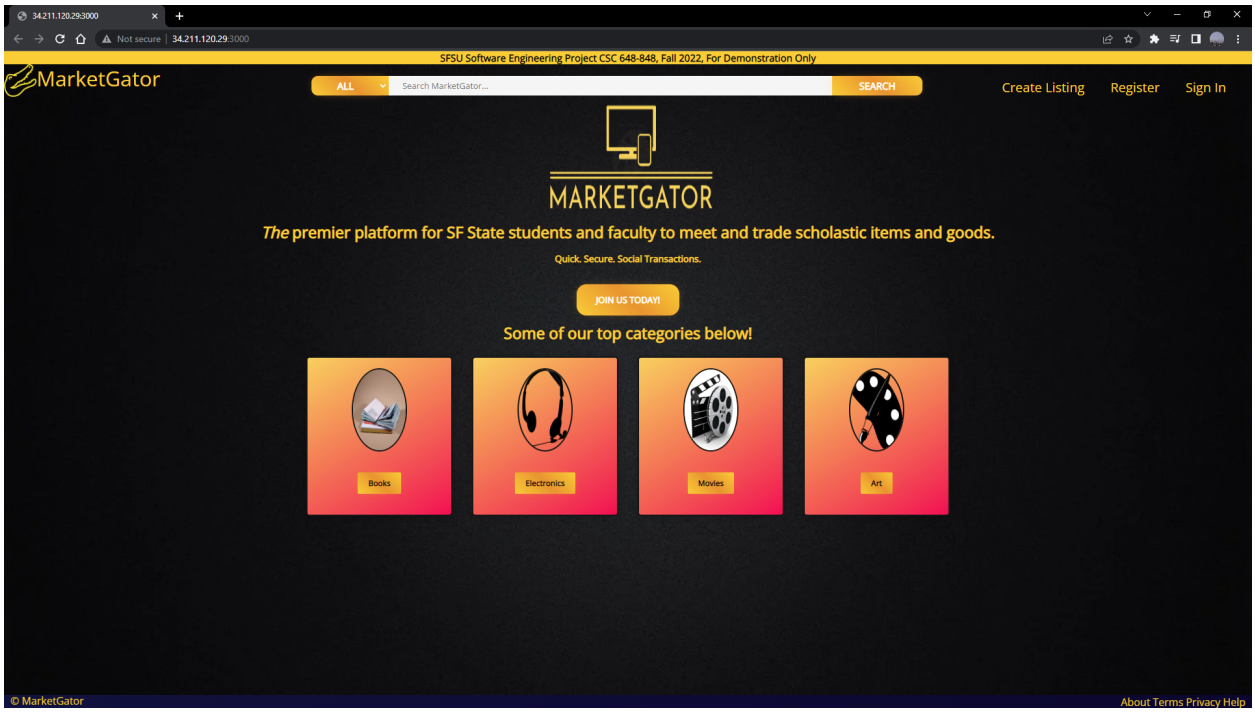
5. Self-Check on Best Practices for Security

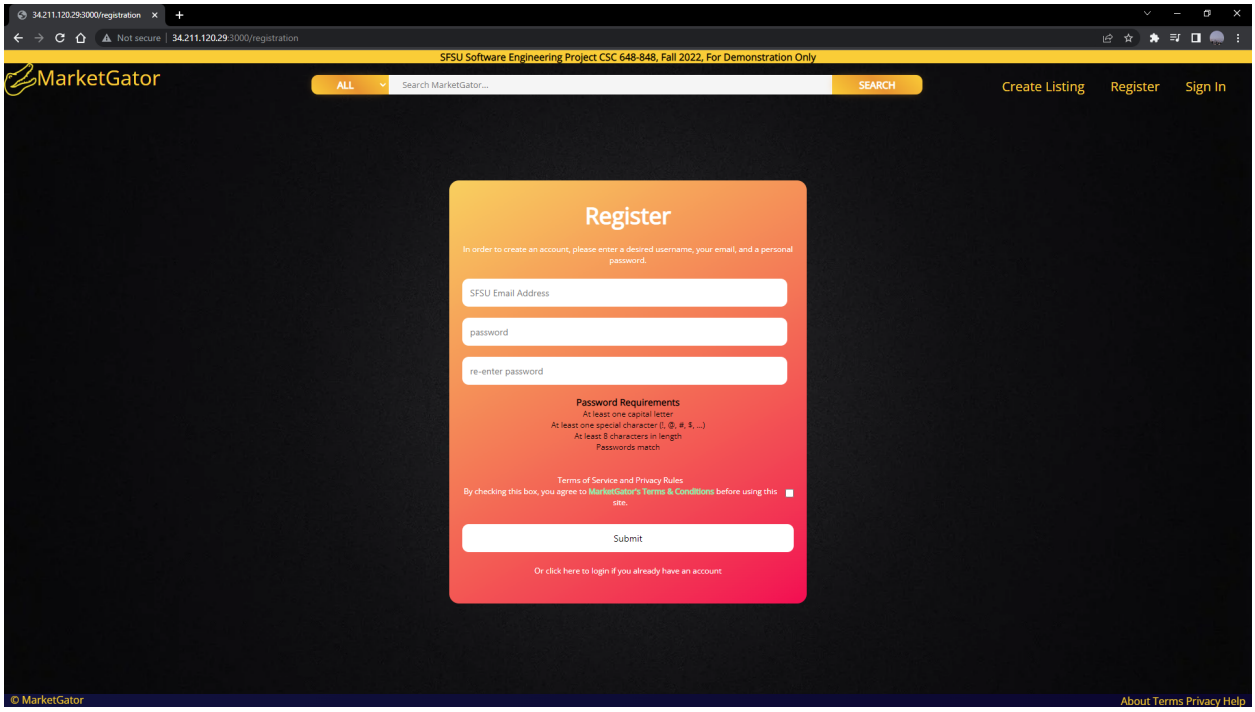
Asset to be protected	**Types of possible/expected attacks**	**Strategy to mitigate/protect the asset**
User images	Illegal content, nudity, or images irrelevant to the marketplace	Admins must approve listing posts using the listing.live variable. Listings will not be displayed to users until manually approved by admin.
User passwords	Database compromised by hacker	Passwords are encrypted inside the program and then inserted into database.
Database	SQL injection	Search function will not allow more than 40 alphanum chars. Search results and user login/registration are validated and not inserted into the database through template strings.

6. Self-check of the adherence to original Non-functional specs – performed by team leads

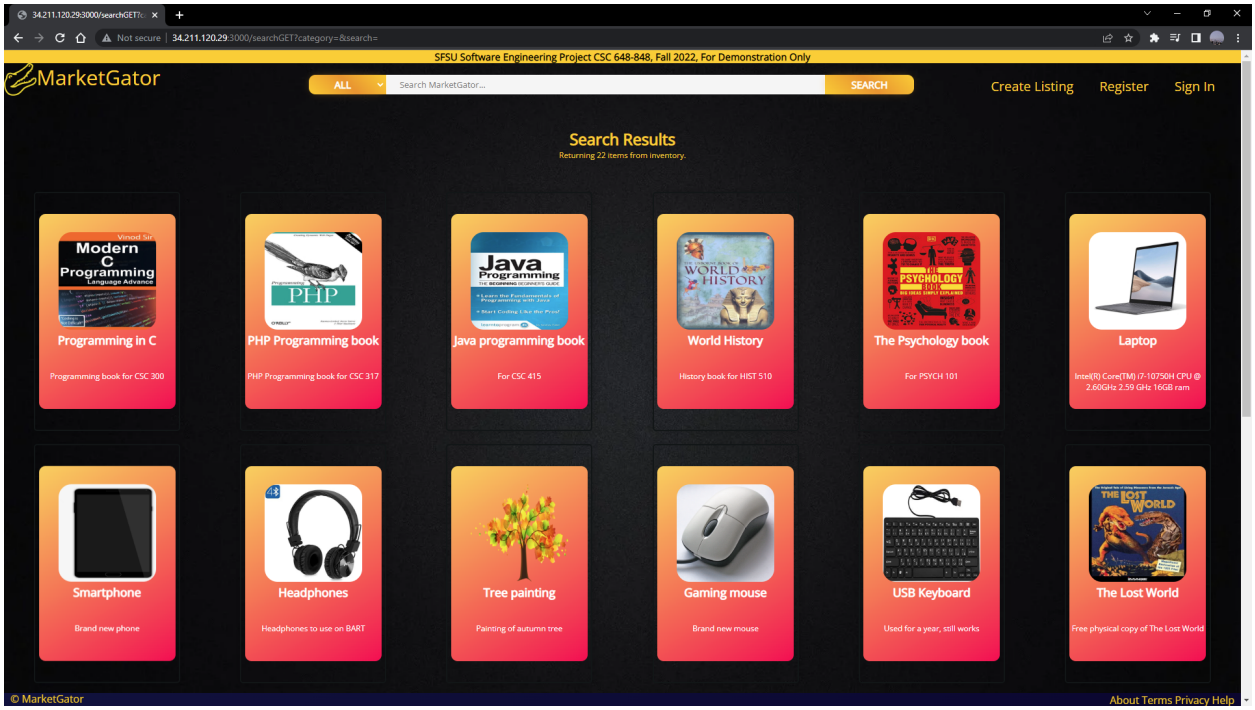
• MySQL setup and development knowledge	DONE
• Accommodating to team members have different schedules from each other	DONE
• Efficient storage of MarketGator listing images	DONE
• Efficient speed of retrieving images and search results	DONE
• Site and UI compatibility of MarketGator when the web browser is resized	DONE

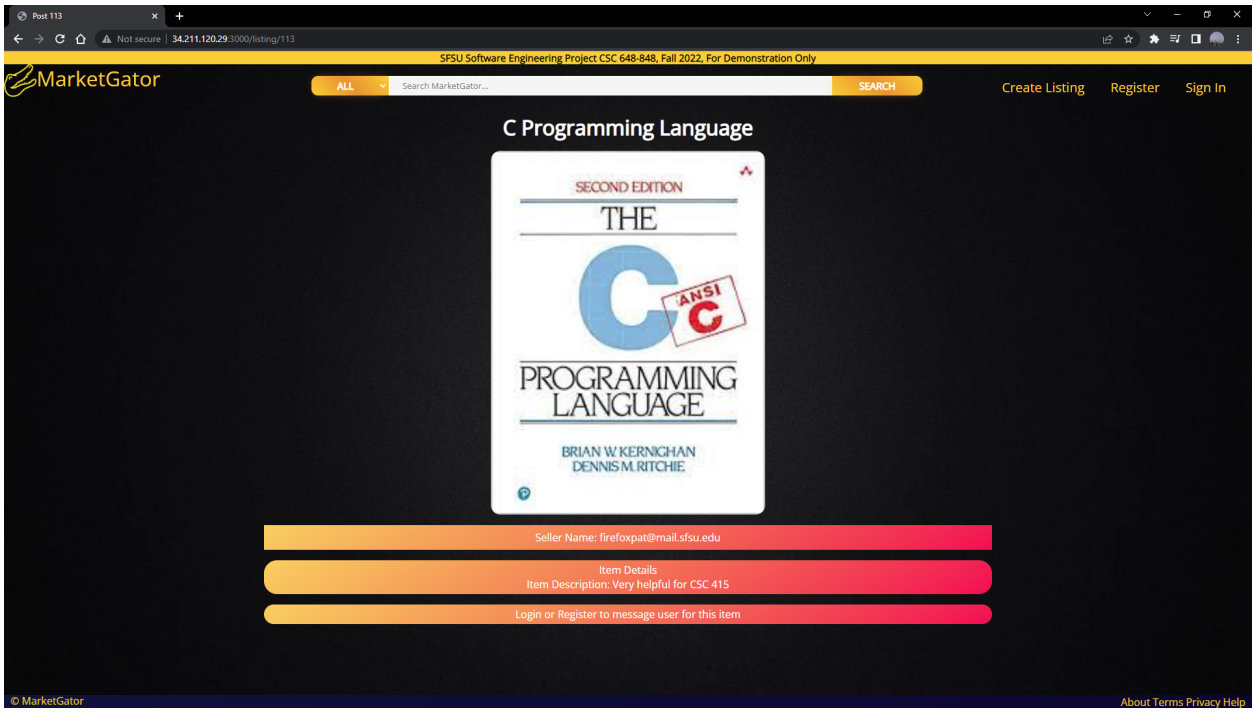
4. Product Screenshots:

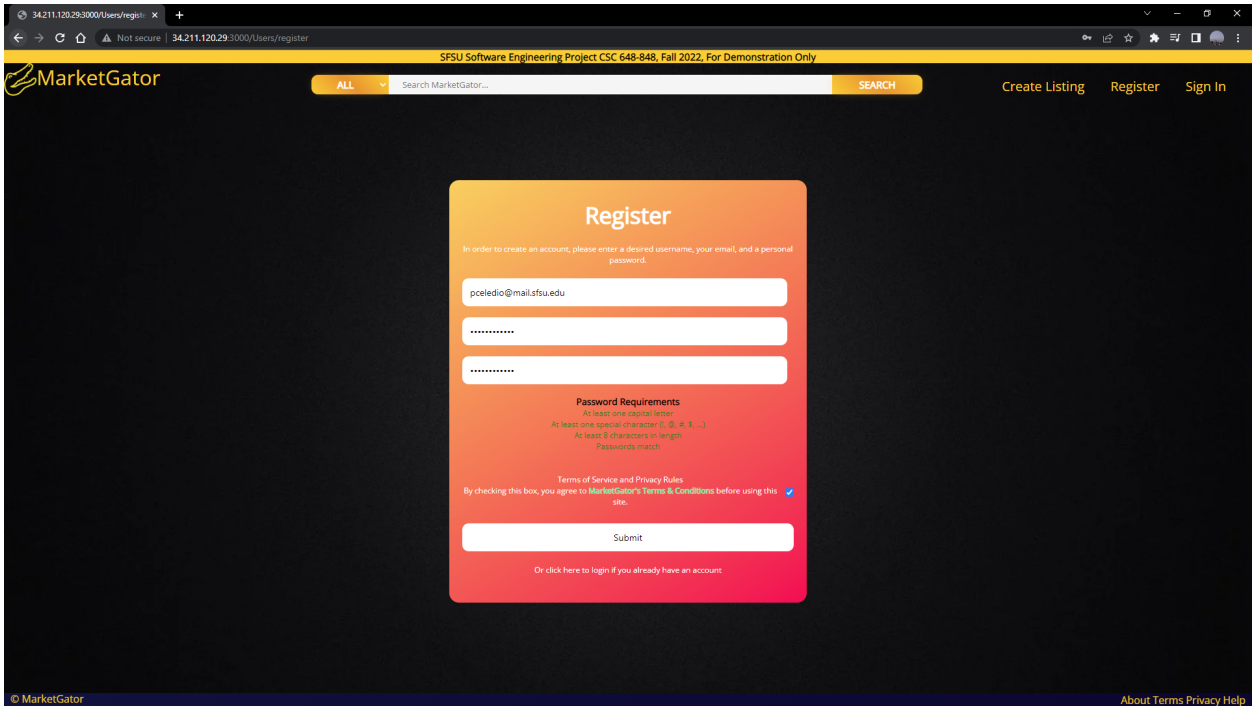


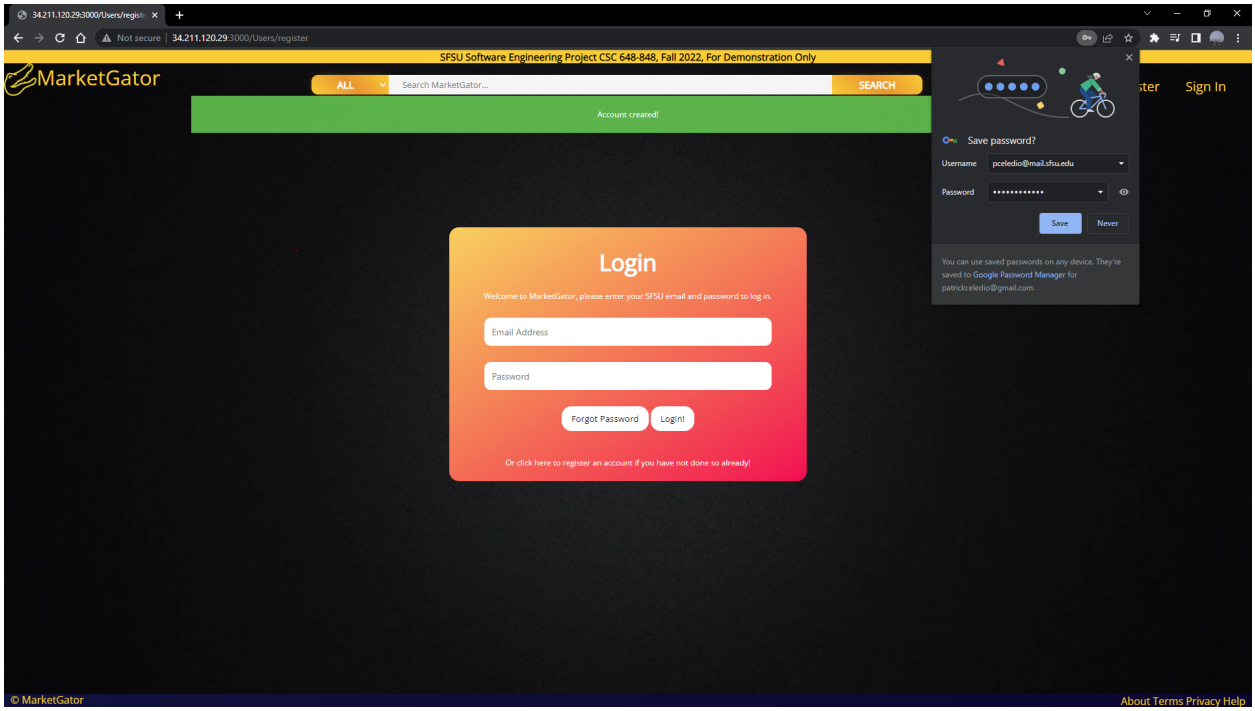


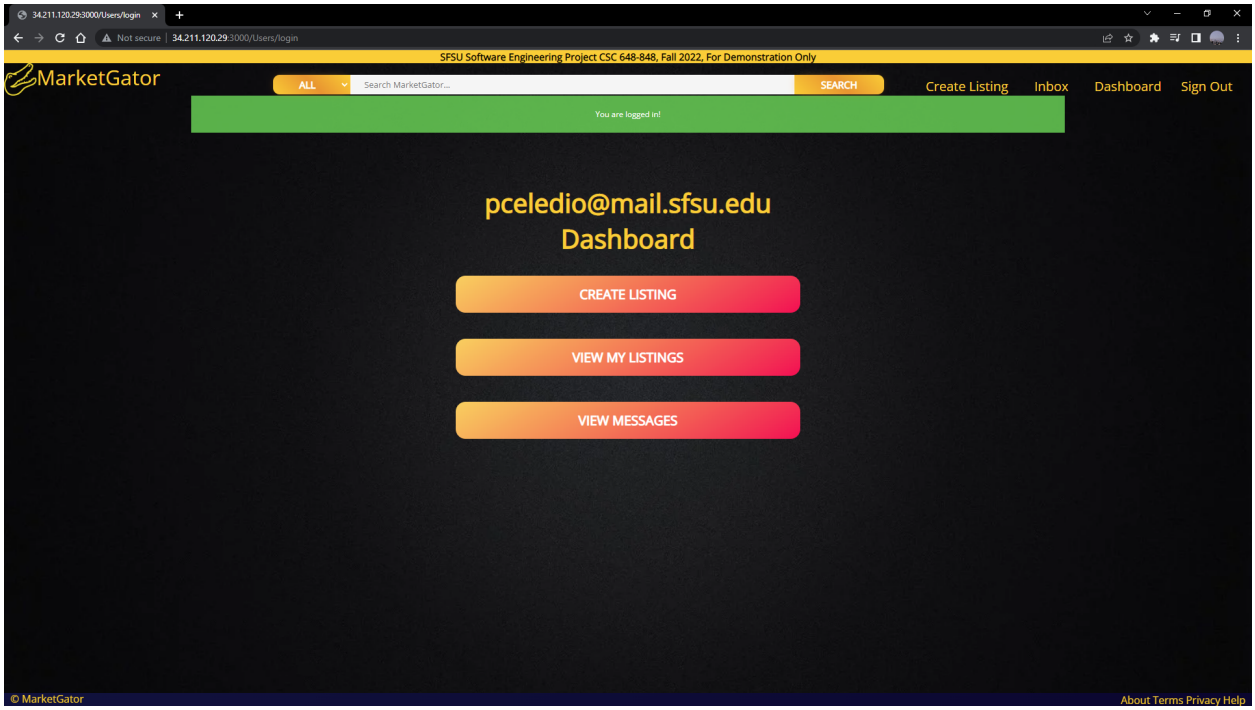
Team 05 Milestone 05

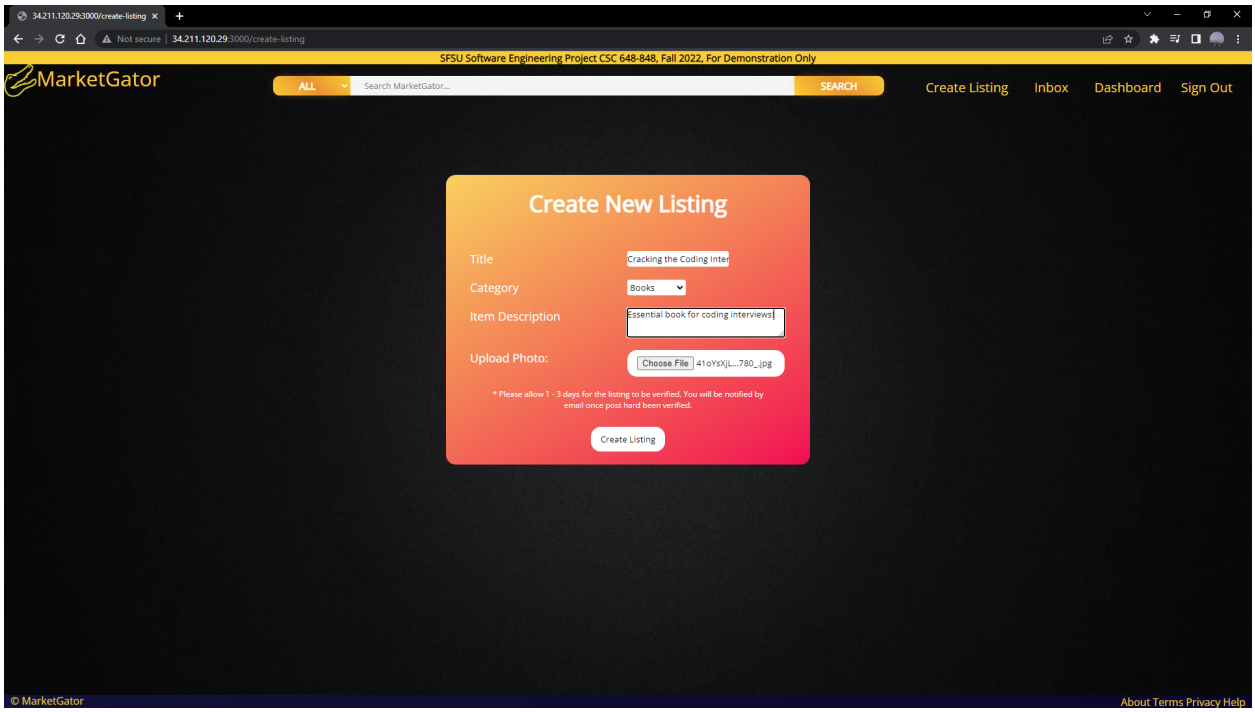


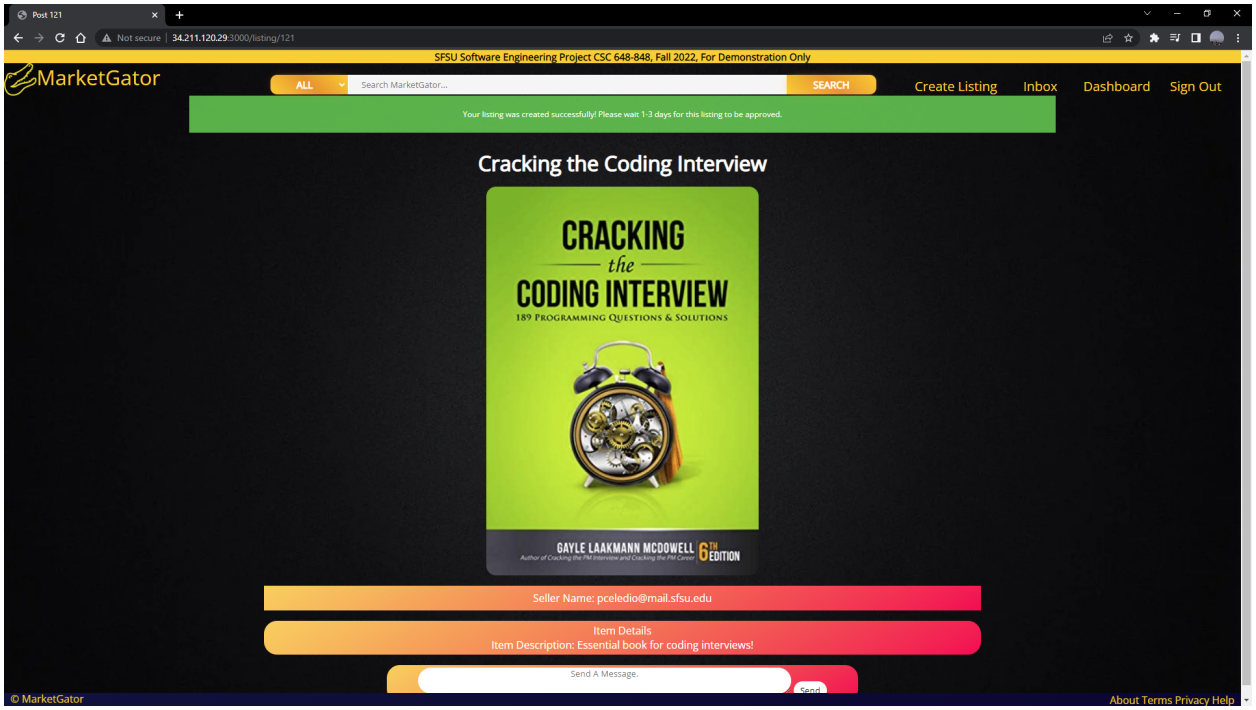


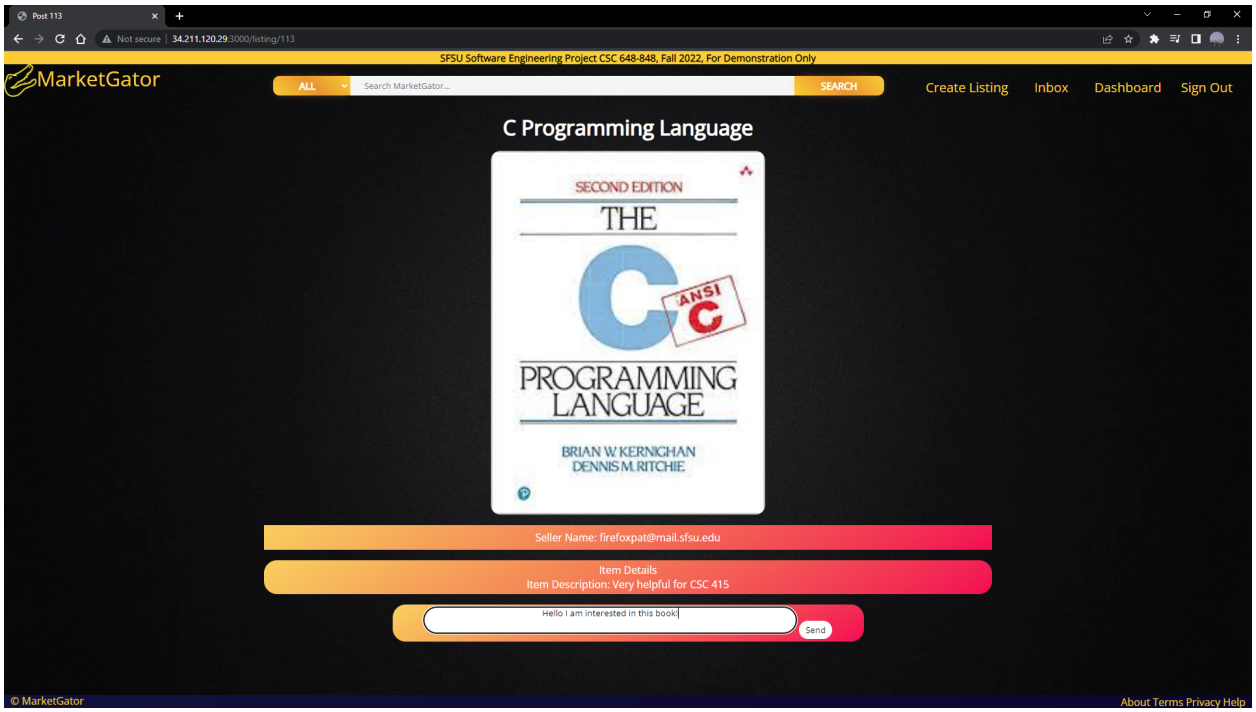

Not secure | 34.211.120.29:3000/Users/register
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Register
Sign In
Register
In order to create an account, please enter a desired username, your email, and a personal password.
pceledio@mail.sfsu.edu
Password Requirements
At least one capital letter
At least one special character (!, @, #, $, ...)
At least 8 characters in length
Passwords match
Terms of Service and Privacy Rules
By checking this box, you agree to MarketGator's Terms & Conditions before using this site.
Submit
Or click here to login if you already have an account
© MarketGator
About Terms Privacy Help

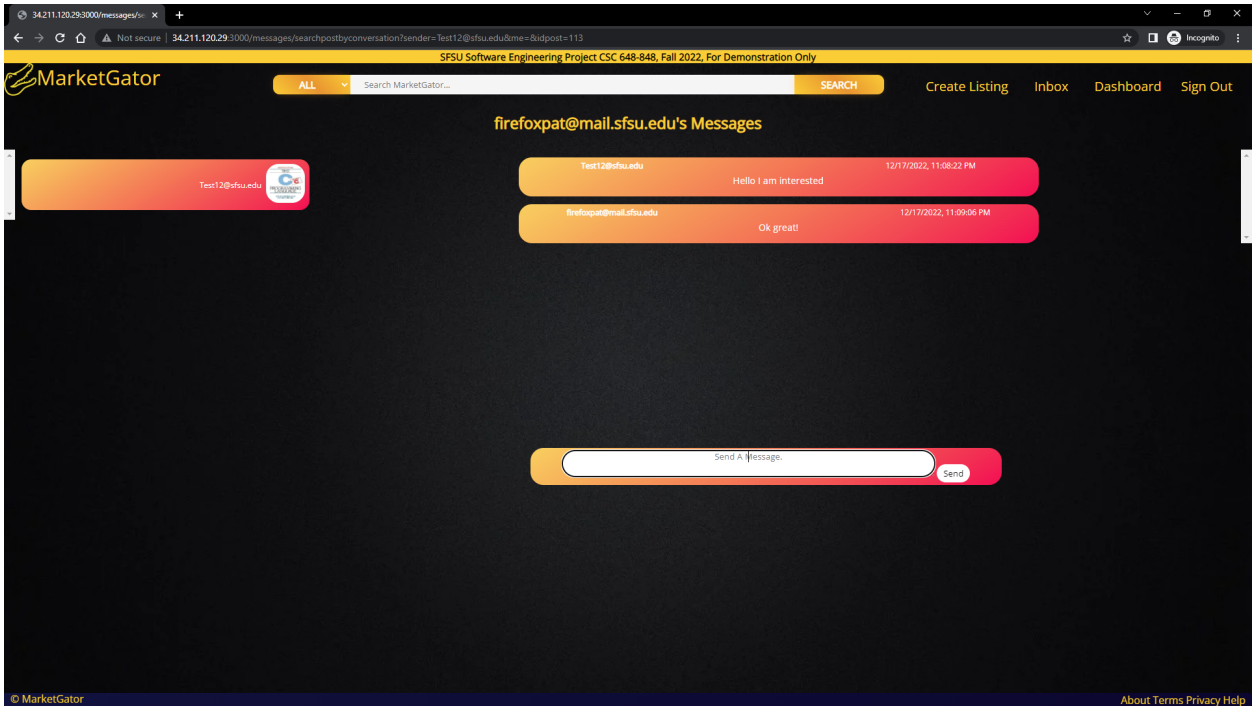

Not secure | 34.211.120.29:3000/Users/register
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Sign In
Account created!
Save password?
Username
pceledio@mail.sfsu.edu
Password
Save
Never
You can use saved passwords on any device. They're saved to Google Password Manager for patrickceledio@gmail.com.
Login
Welcome to MarketGator, please enter your SFSU email and password to log in.
Email Address
Password
Forgot Password
Login!
Or click here to register an account if you have not done so already!
© MarketGator
About Terms Privacy Help

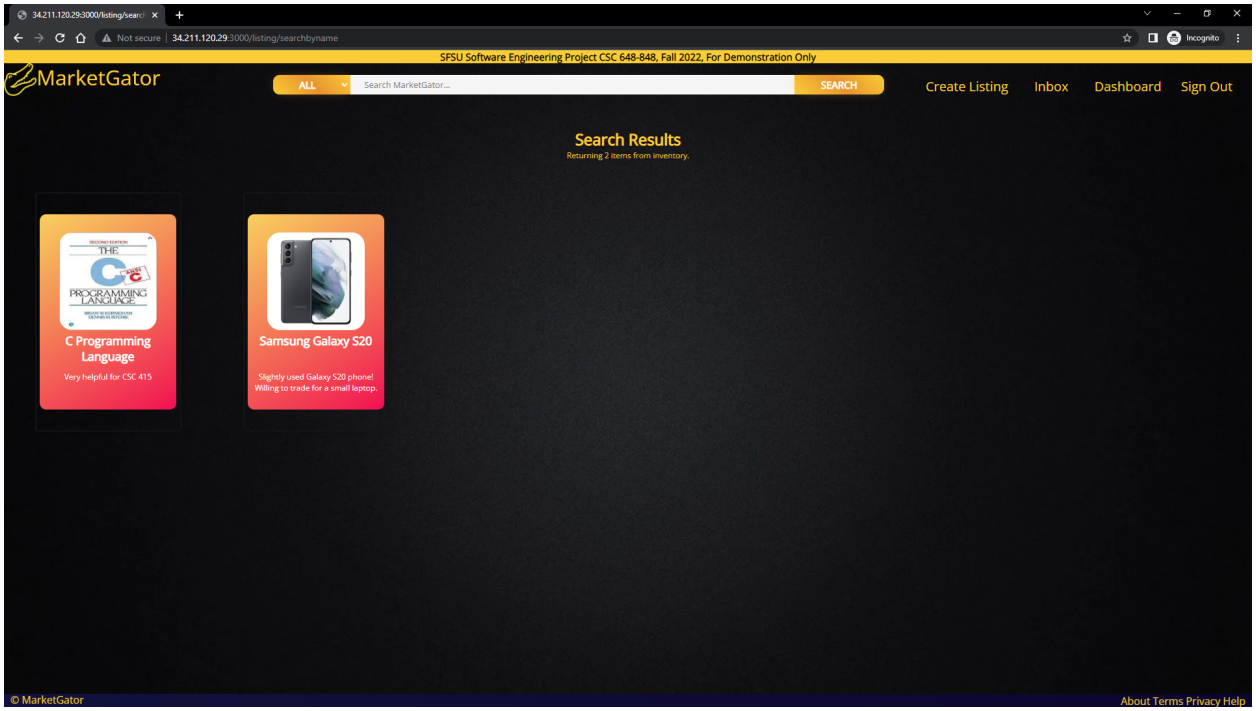

34.211.120.29:3000/Users/login
Not secure | 34.211.120.29:3000/Users/login
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
You are logged in!
pceledio@mail.sfsu.edu
Dashboard
CREATE LISTING
VIEW MY LISTINGS
VIEW MESSAGES
© MarketGator
About Terms Privacy Help


34.211.120.29:3000/create-listing
Not secure | 34.211.120.29:3000/create-listing
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
Create New Listing
Title
Cracking the Coding Inter
Category
Books
Item Description
Essential book for coding interviews
Upload Photo:
Choose File
41oYsXjL...780_.jpg
* Please allow 1 - 3 days for the listing to be verified. You will be notified by email once post hard been verified.
Create Listing
© MarketGator
About Terms Privacy Help

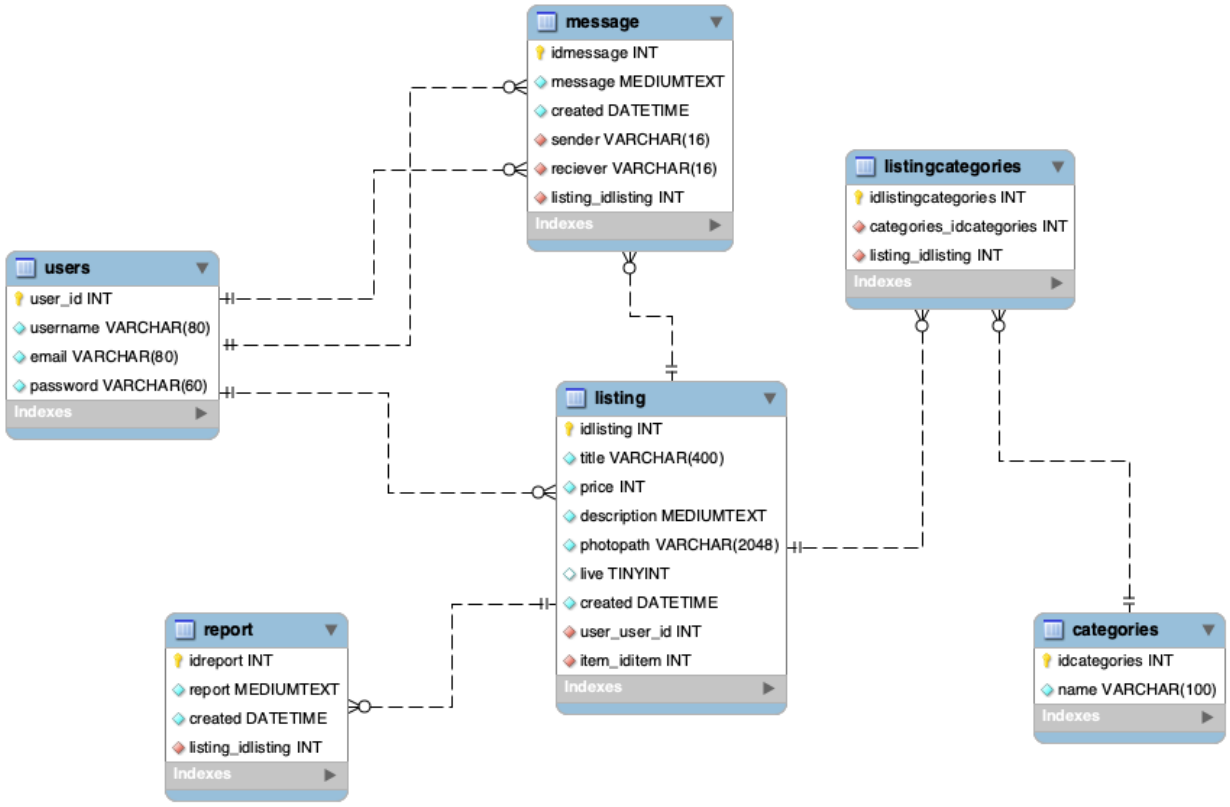

Post 121
Not secure | 34.211.120.29:3000/listing/121
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
Your listing was created successfully! Please wait 1-3 days for this listing to be approved.
Cracking the Coding Interview
CRACKING
the
CODING INTERVIEW
189 PROGRAMMING QUESTIONS & SOLUTIONS
GAYLE LAAKMANN MCDOWELL
6TH EDITION
Seller Name: pceledio@mail.sfsu.edu
Item Details
Item Description: Essential book for coding interviews!
Send A Message.
Send
© MarketGator
About Terms Privacy Help

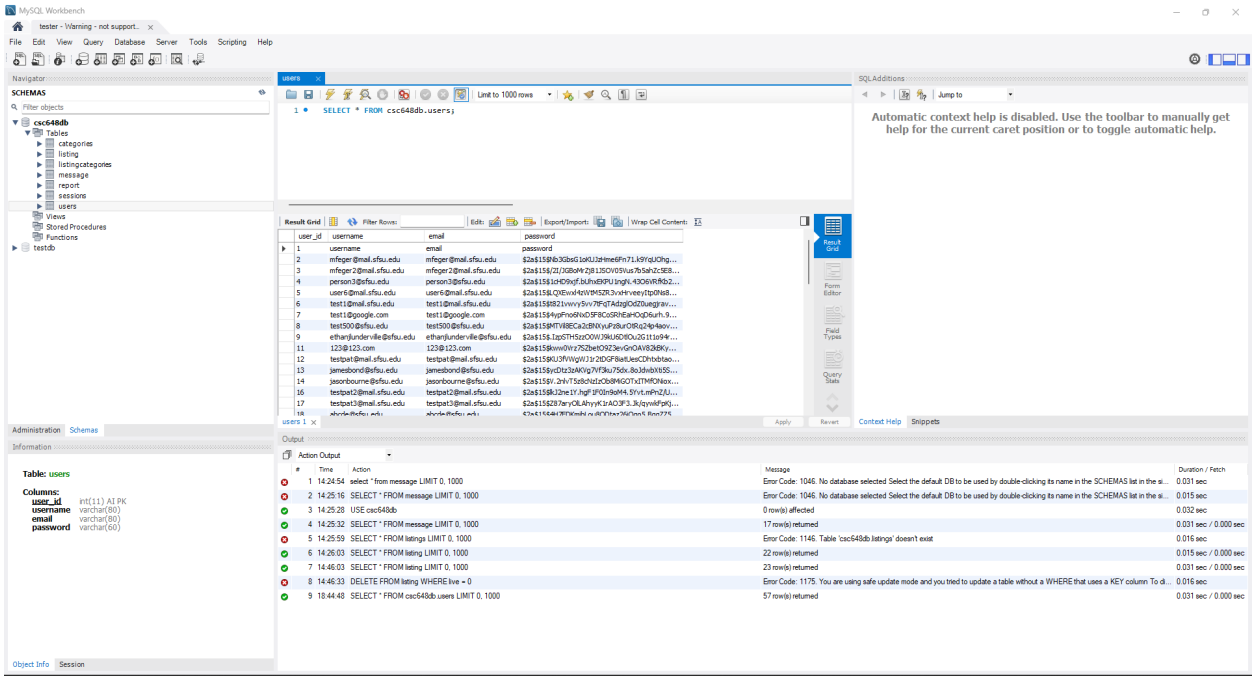

Post 113
Not secure | 34.211.120.29:3000/listing/113
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
C Programming Language
SECOND EDITION
THE
C
ANSI C
PROGRAMMING
LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE
Seller Name: firefoxpat@mail.sfsu.edu
Item Details
Item Description: Very helpful for CSC 415
Hello I am interested in this book!
Send
© MarketGator
About Terms Privacy Help

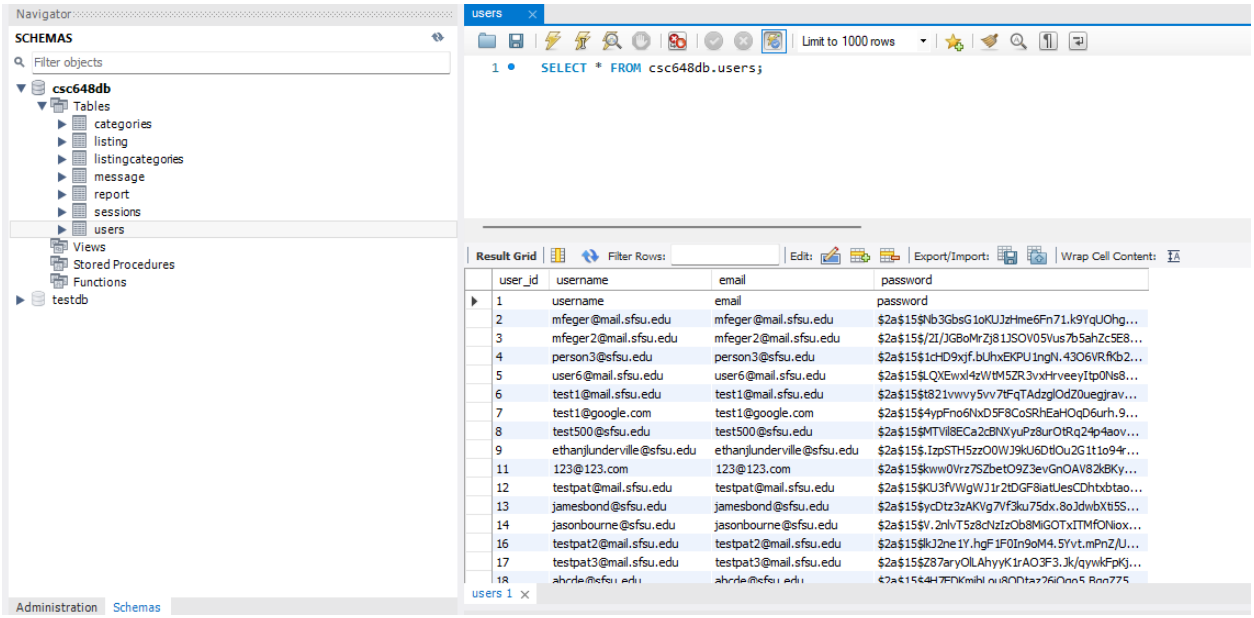

Not secure | 34.211.120.29:3000/messages/searchpostbyconversation?sender=Test12@sfsu.edu&me=&idpost=113
Incognito
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
firefoxpat@mail.sfsu.edu's Messages
Test12@sfsu.edu
Test12@sfsu.edu
12/17/2022, 11:08:22 PM
Hello I am interested
firefoxpat@mail.sfsu.edu
12/17/2022, 11:09:06 PM
Ok great!
Send A Message.
Send
© MarketGator
About Terms Privacy Help

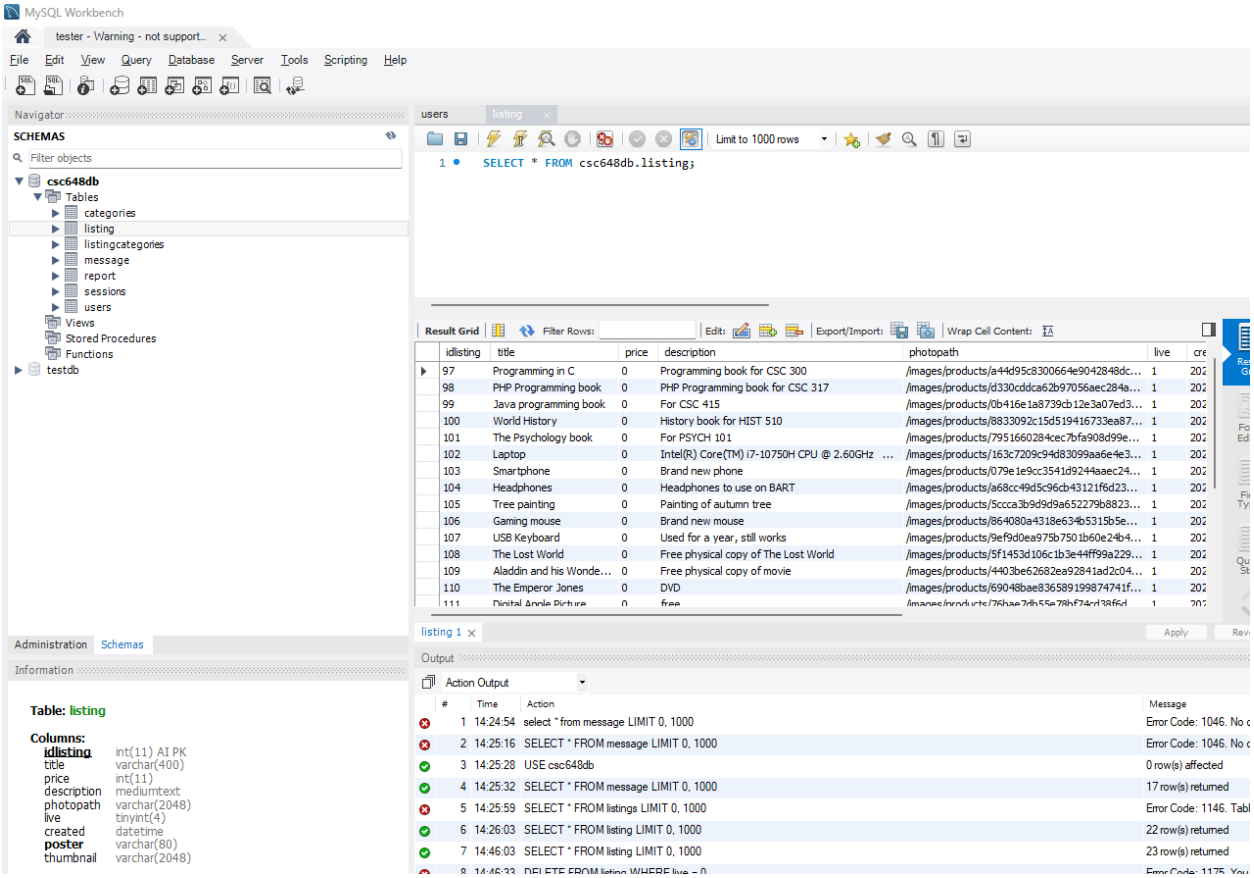

Not secure | 34.211.120.29:3000/listing/searchbyname
Incognito
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
MarketGator
ALL
Search MarketGator...
SEARCH
Create Listing
Inbox
Dashboard
Sign Out
Search Results
Returning 2 items from inventory.
C Programming Language
Very helpful for CSC 415
Samsung Galaxy S20
Slightly used Galaxy S20 phone! Willing to trade for a small laptop.
© MarketGator
About Terms Privacy Help

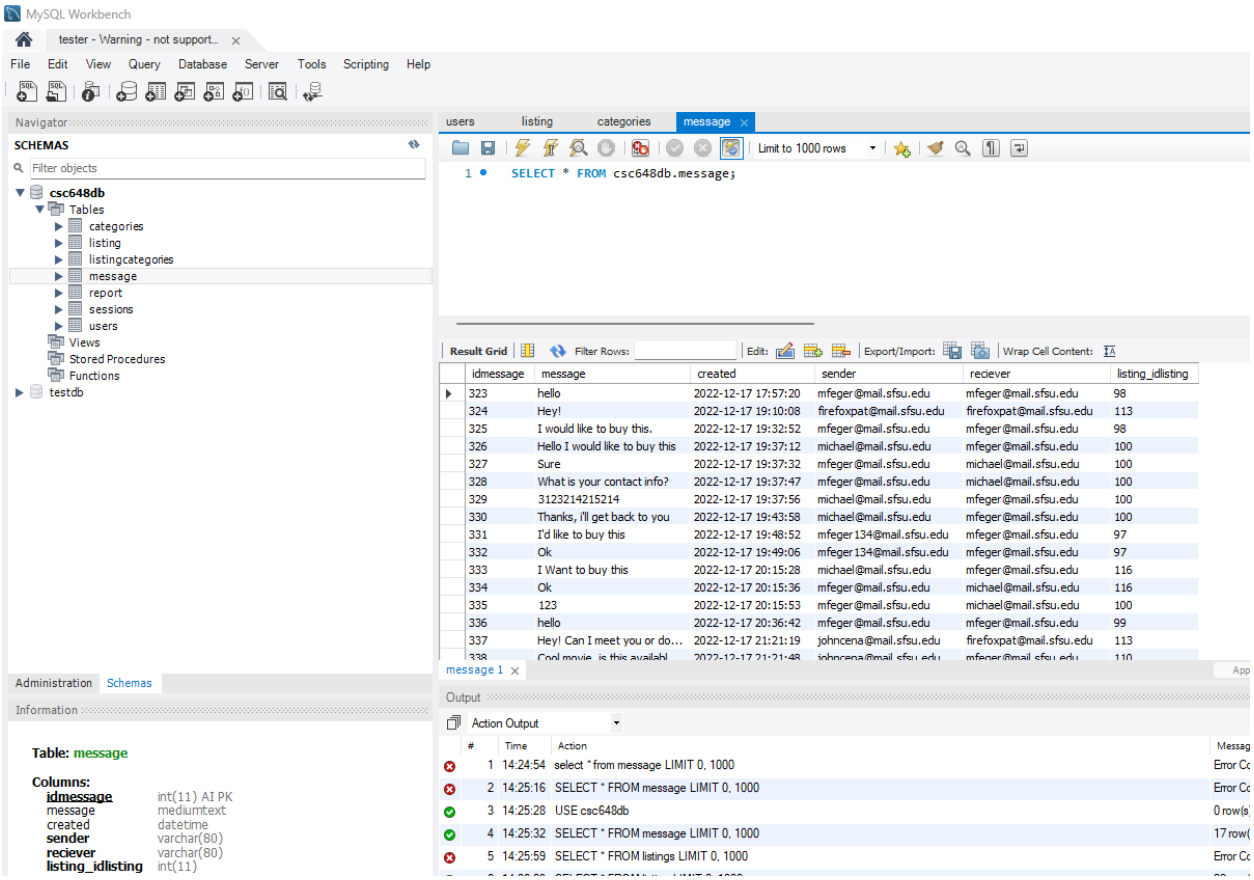
5. Database Organization:

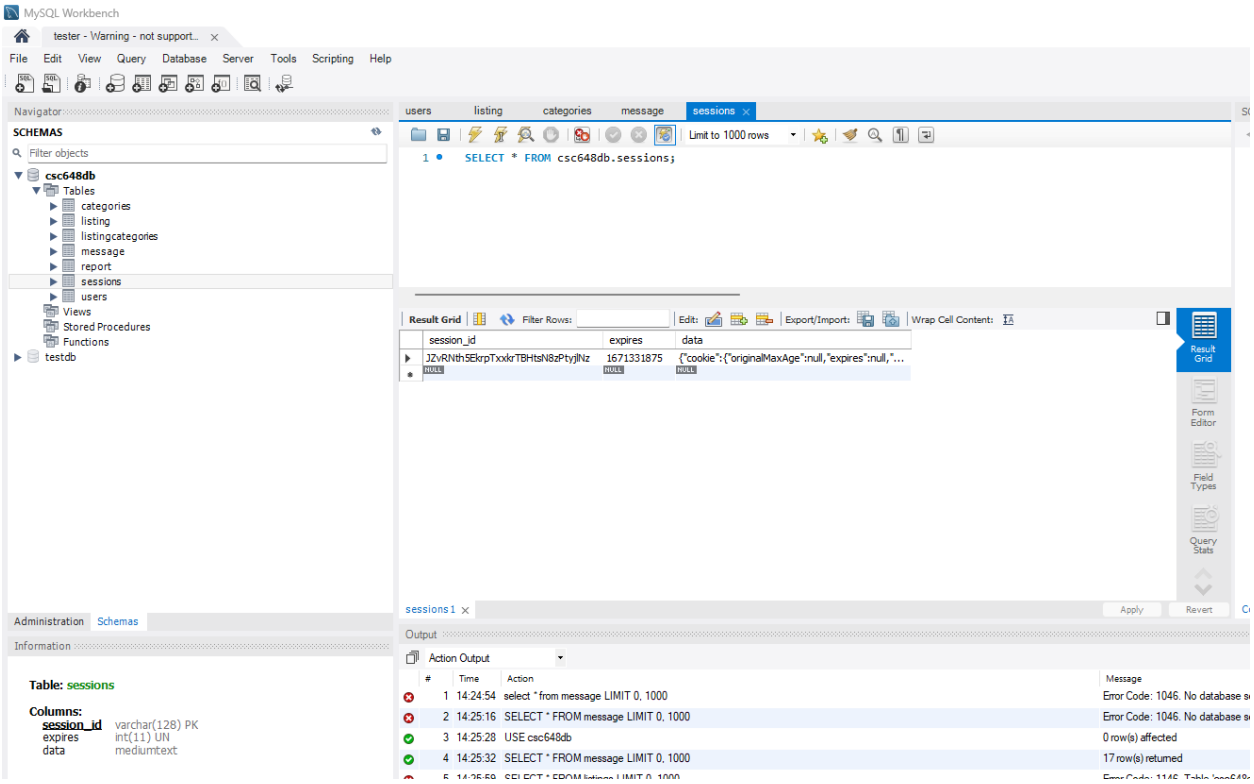





Navigator
SCHEMAS
Filter objects
csc648db
Tables
categories
listing
listingcategories
message
report
sessions
users
Views
Stored Procedures
Functions
testdb
Administration
Schemas
users
Limit to 1000 rows
1 • SELECT * FROM csc648db.users;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
user_id
username
email
password
1 username email password
2 mfeger@mail.sfsu.edu mfeger@mail.sfsu.edu $2a$15$Nb3GbsG1oKUJzHme6Fn71.k9YqUOhg...
3 mfeger2@mail.sfsu.edu mfeger2@mail.sfsu.edu $2a$15$/2I/JGBoMrZj81JSOV05Vus7b5ahZc5E8...
4 person3@sfsu.edu person3@sfsu.edu $2a$15$1cHD9xjf.bUhxEKPU1ngN.43O6VRfKb2...
5 user6@mail.sfsu.edu user6@mail.sfsu.edu $2a$15$LQXEwxl4zWtM5ZR3vxHrveeyItp0Ns8...
6 test1@mail.sfsu.edu test1@mail.sfsu.edu $2a$15$t821vwvy5vv7tFqTAdzglOdZ0uegjrav...
7 test1@google.com test1@google.com $2a$15$4ypFno6NxD5F8CoSRhEaHOqD6urh.9...
8 test500@sfsu.edu test500@sfsu.edu $2a$15$MTVil8ECa2cBNXyuPz8urOtRq24p4aov...
9 ethanjlunderville@sfsu.edu ethanjlunderville@sfsu.edu $2a$15$.IzpSTH5zzO0WJ9kU6DtlOu2G1t1o94r...
11 123@123.com 123@123.com $2a$15$kww0Vrz7SZbetO9Z3evGnOAV82kBKy...
12 testpat@mail.sfsu.edu testpat@mail.sfsu.edu $2a$15$KU3fVWgWJ1r2tDGF8iatUesCDhtxbtao...
13 jamesbond@sfsu.edu jamesbond@sfsu.edu $2a$15$ycDtz3zAKVg7Vf3ku75dx.8oJdwbXti5S...
14 jasonbourne@sfsu.edu jasonbourne@sfsu.edu $2a$15$V.2nlvT5z8cNzIzOb8MiGOTxITMfONiox...
16 testpat2@mail.sfsu.edu testpat2@mail.sfsu.edu $2a$15$lkJ2ne1Y.hgF1F0In9oM4.5Yvt.mPnZ/U...
17 testpat3@mail.sfsu.edu testpat3@mail.sfsu.edu $2a$15$Z87aryOlLAhyyK1rAO3F3.Jk/qywkFpKj...
users 1


MySQL Workbench
tester - Warning - not support...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
csc648db
Tables
categories
listing
listingcategories
message
report
sessions
users
Views
Stored Procedures
Functions
testdb
Administration
Schemas
Information
Table: listing
Columns:
idlisting int(11) AI PK
title varchar(400)
price int(11)
description mediumtext
photopath varchar(2048)
live tinyint(4)
created datetime
poster varchar(80)
thumbnail varchar(2048)
users
listing
Limit to 1000 rows
1 • SELECT * FROM csc648db.listing;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
idlisting title price description photopath live
97 Programming in C 0 Programming book for CSC 300 /images/products/a44d95c8300664e9042848dc... 1
98 PHP Programming book 0 PHP Programming book for CSC 317 /images/products/d330cddca62b97056aec284a... 1
99 Java programming book 0 For CSC 415 /images/products/0b416e1a8739cb12e3a07ed3... 1
100 World History 0 History book for HIST 510 /images/products/8833092c15d519416733ea87... 1
101 The Psychology book 0 For PSYCH 101 /images/products/7951660284cec7bfa908d99e... 1
102 Laptop 0 Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz ... /images/products/163c7209c94d83099aa6e4e3... 1
103 Smartphone 0 Brand new phone /images/products/079e1e9cc3541d9244aaec24... 1
104 Headphones 0 Headphones to use on BART /images/products/a68cc49d5c96cb43121f6d23... 1
105 Tree painting 0 Painting of autumn tree /images/products/5ccca3b9d9d9a652279b8823... 1
106 Gaming mouse 0 Brand new mouse /images/products/864080a4318e634b5315b5e... 1
107 USB Keyboard 0 Used for a year, still works /images/products/9ef9d0ea975b7501b60e24b4... 1
108 The Lost World 0 Free physical copy of The Lost World /images/products/5f1453d106c1b3e44ff99a229... 1
109 Aladdin and his Wonde... 0 Free physical copy of movie /images/products/4403be62682ea92841ad2c04... 1
110 The Emperor Jones 0 DVD /images/products/69048bae836589199874741f... 1
listing 1
Apply
Output
Action Output
Time Action Message
1 14:24:54 select * from message LIMIT 0, 1000 Error Code: 1046. No
2 14:25:16 SELECT * FROM message LIMIT 0, 1000 Error Code: 1046. No
3 14:25:28 USE csc648db 0 row(s) affected
4 14:25:32 SELECT * FROM message LIMIT 0, 1000 17 row(s) returned
5 14:25:59 SELECT * FROM listings LIMIT 0, 1000 Error Code: 1146. Tabl
6 14:26:03 SELECT * FROM listing LIMIT 0, 1000 22 row(s) returned
7 14:46:03 SELECT * FROM listing LIMIT 0, 1000 23 row(s) returned


MySQL Workbench
tester - Warning - not support...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
csc648db
Tables
categories
listing
listingcategories
message
report
sessions
users
Views
Stored Procedures
Functions
testdb
Administration Schemas
Information
Table: message
Columns:
idmessage int(11) AI PK
message mediumtext
created datetime
sender varchar(80)
reciever varchar(80)
listing_idlisting int(11)
users listing categories message
Limit to 1000 rows
1 • SELECT * FROM csc648db.message;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
idmessage message created sender reciever listing_idlisting
323 hello 2022-12-17 17:57:20 mfeger@mail.sfsu.edu mfeger@mail.sfsu.edu 98
324 Hey! 2022-12-17 19:10:08 firefoxpat@mail.sfsu.edu firefoxpat@mail.sfsu.edu 113
325 I would like to buy this. 2022-12-17 19:32:52 mfeger@mail.sfsu.edu mfeger@mail.sfsu.edu 98
326 Hello I would like to buy this 2022-12-17 19:37:12 michael@mail.sfsu.edu mfeger@mail.sfsu.edu 100
327 Sure 2022-12-17 19:37:32 mfeger@mail.sfsu.edu michael@mail.sfsu.edu 100
328 What is your contact info? 2022-12-17 19:37:47 mfeger@mail.sfsu.edu michael@mail.sfsu.edu 100
329 3123214215214 2022-12-17 19:37:56 michael@mail.sfsu.edu mfeger@mail.sfsu.edu 100
330 Thanks, i'll get back to you 2022-12-17 19:43:58 michael@mail.sfsu.edu mfeger@mail.sfsu.edu 100
331 I'd like to buy this 2022-12-17 19:48:52 mfeger134@mail.sfsu.edu mfeger@mail.sfsu.edu 97
332 Ok 2022-12-17 19:49:06 mfeger134@mail.sfsu.edu mfeger@mail.sfsu.edu 97
333 I Want to buy this 2022-12-17 20:15:28 michael@mail.sfsu.edu mfeger@mail.sfsu.edu 116
334 Ok 2022-12-17 20:15:36 mfeger@mail.sfsu.edu michael@mail.sfsu.edu 116
335 123 2022-12-17 20:15:53 mfeger@mail.sfsu.edu michael@mail.sfsu.edu 100
336 hello 2022-12-17 20:36:42 mfeger@mail.sfsu.edu mfeger@mail.sfsu.edu 99
337 Hey! Can I meet you or do... 2022-12-17 21:21:19 johncena@mail.sfsu.edu firefoxpat@mail.sfsu.edu 113
message 1
Output
Action Output
Time Action Message
1 14:24:54 select * from message LIMIT 0, 1000 Error Co
2 14:25:16 SELECT * FROM message LIMIT 0, 1000 Error Co
3 14:25:28 USE csc648db 0 row(s)
4 14:25:32 SELECT * FROM message LIMIT 0, 1000 17 row(
5 14:25:59 SELECT * FROM listings LIMIT 0, 1000 Error Co


MySQL Workbench
tester - Warning - not support...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
csc648db
Tables
categories
listing
listingcategories
message
report
sessions
users
Views
Stored Procedures
Functions
testdb
Administration Schemas
Information
Table: sessions
Columns:
session_id varchar(128) PK
expires int(11) UN
data mediumtext
users listing categories message sessions
Limit to 1000 rows
1 • SELECT * FROM csc648db.sessions;
Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:
session_id expires data
JZvRNth5EkrpTxxkrTBHtsN8zPtyJlNz 1671331875 {"cookie":{"originalMaxAge":null,"expires":null,"...
sessions 1
Apply Revert
Output
Action Output
Time Action Message
1 14:24:54 select * from message LIMIT 0, 1000 Error Code: 1046. No database s
2 14:25:16 SELECT * FROM message LIMIT 0, 1000 Error Code: 1046. No database s
3 14:25:28 USE csc648db 0 row(s) affected
4 14:25:32 SELECT * FROM message LIMIT 0, 1000 17 row(s) returned

6. Github organization:

Link to Team 05 GitHub repo: https://github.com/CSC-648-SFSU/csc648-03-fa22-team05

Main branches in GitHub:

- Ahmad
- Ethan
- michaelfeger
- nyanyelin
- patrick
- rohit

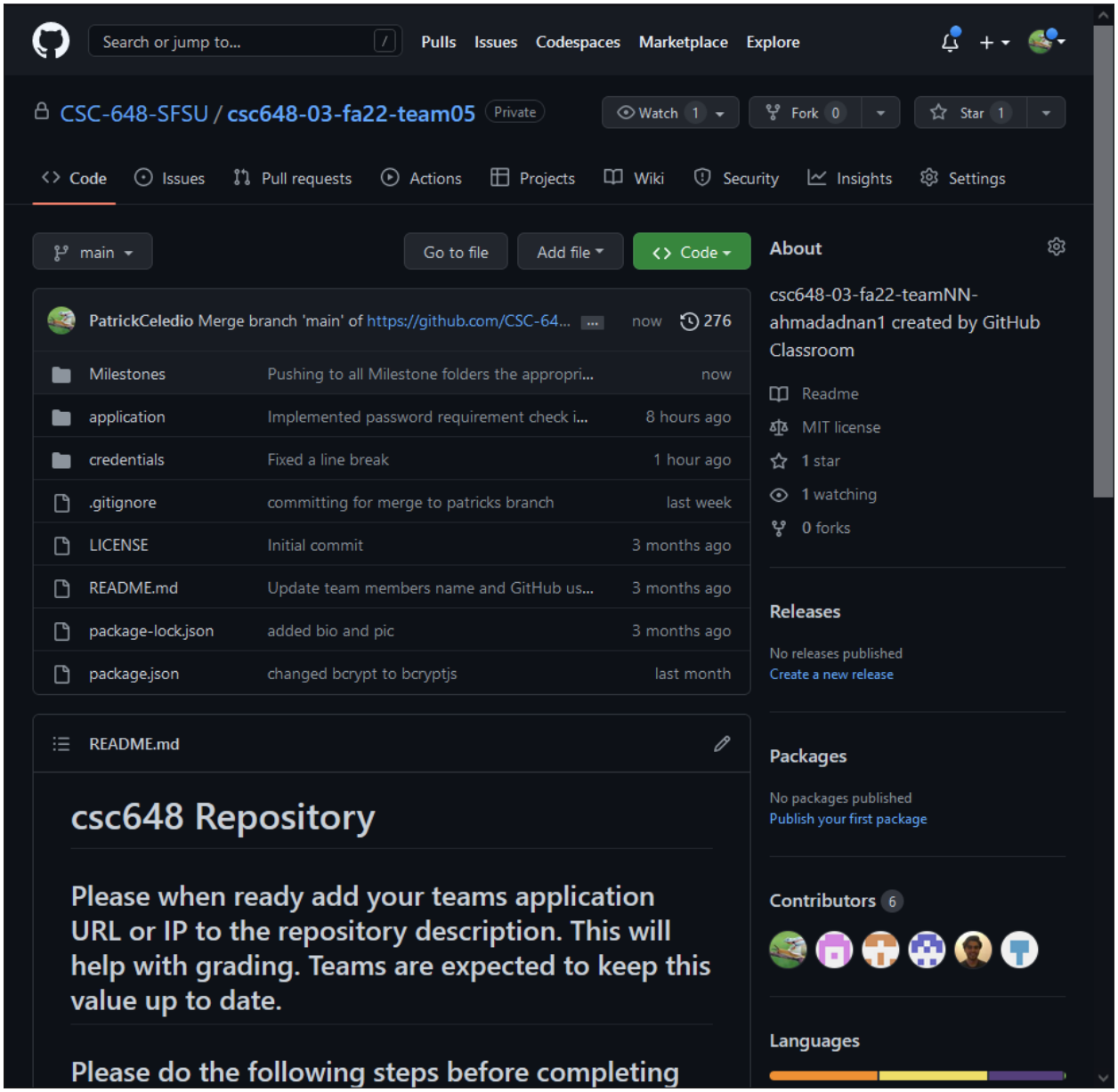
Access to master branch:

- Ahmad
    - GitHub Master
- patrick
    - Team 05 Lead

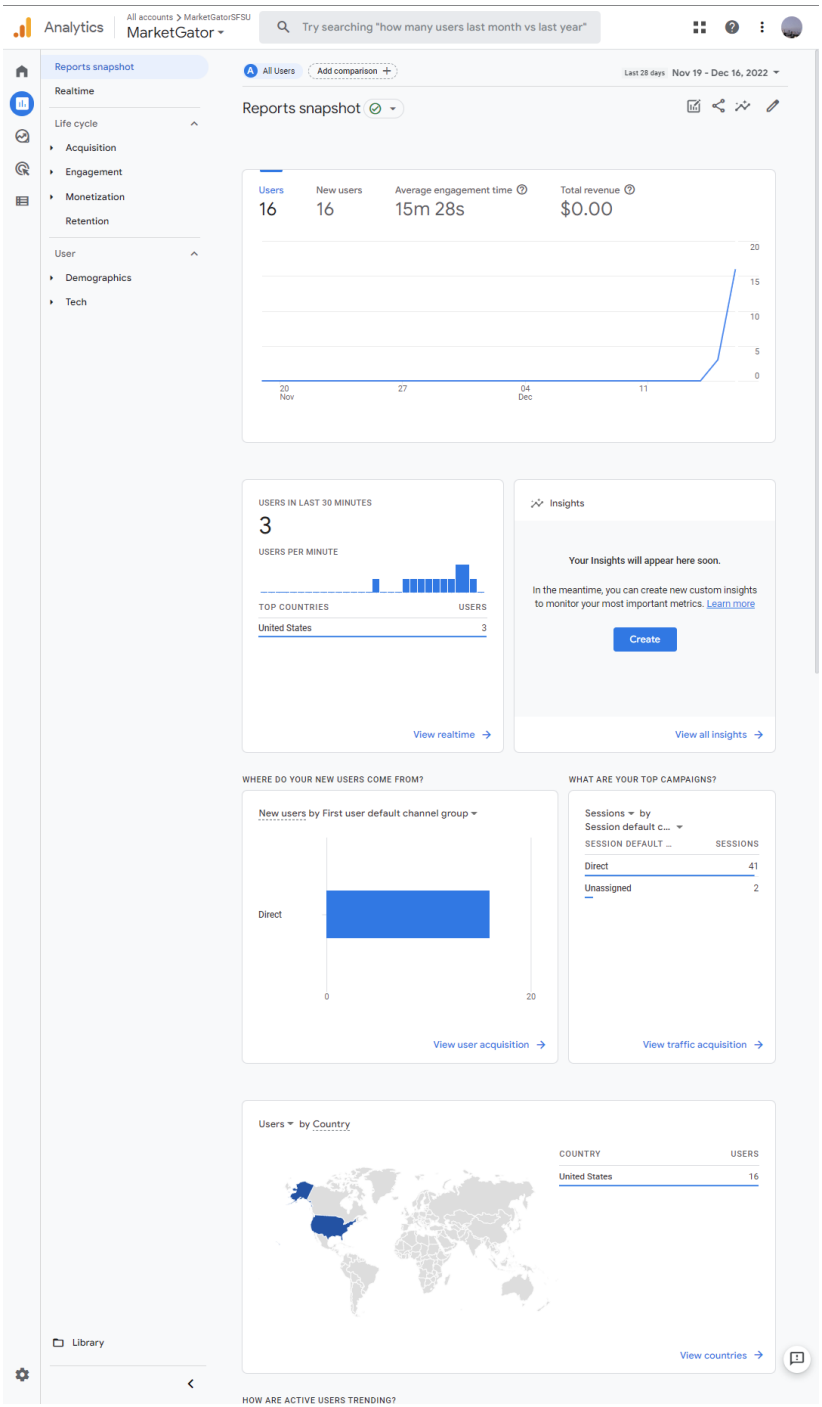
Top level folders in GitHub file organization

- Milestones
    - Contains all the documents for each Milestone: 0, 1, 2, 3, 4, and 5
- application
    - Contains the source code of MarketGator
        - Front-end code
        - Back-end code
- credentials
    - Contains all the credentials needed to log into the MarketGator server and databases

Screenshot of Team 05 GitHub home page

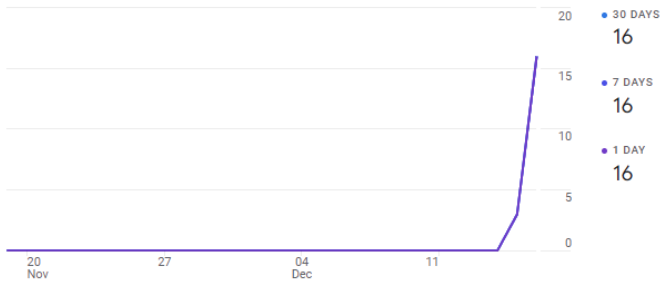


7. Google analytics stats:



HOW ARE ACTIVE USERS TRENDING?

User activity over time



HOW WELL DO YOU RETAIN YOUR USERS?

User activity by cohort

Based on device data only

	Week 0	Week 1	Week 2	Week 3	Week 4	Week 5
All Users	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Oct 30 - Nov 5						
Nov 6 - Nov 12						
Nov 13 - Nov 19						
Nov 20 - Nov 26						
Nov 27 - Dec 3						
Dec 4 - Dec 10						

6 weeks ending Dec 10

View retention →

WHICH PAGES AND SCREENS GET THE MOST VIEWS?

Views by Page title and screen class

PAGE TITLE AND SCREEN CLASS	VIEWS
Post 4	15
Post 3	10
Post 6	10
Post 5	7
Post 83	6
Post 10	5
Post 2	5

View pages and screens →

WHAT ARE YOUR TOP EVENTS?

Event count by Event name

EVENT NAME	EVENT COUNT
page_view	1.8K
scroll	1.7K
user_engagement	1.4K
form_submit	400
form_start	286
click	231
session_start	41

View events →

WHAT ARE YOUR TOP CONVERSIONS?

Conversions by Event name

Include Is conve...

EVENT NAME	CONVERSIONS

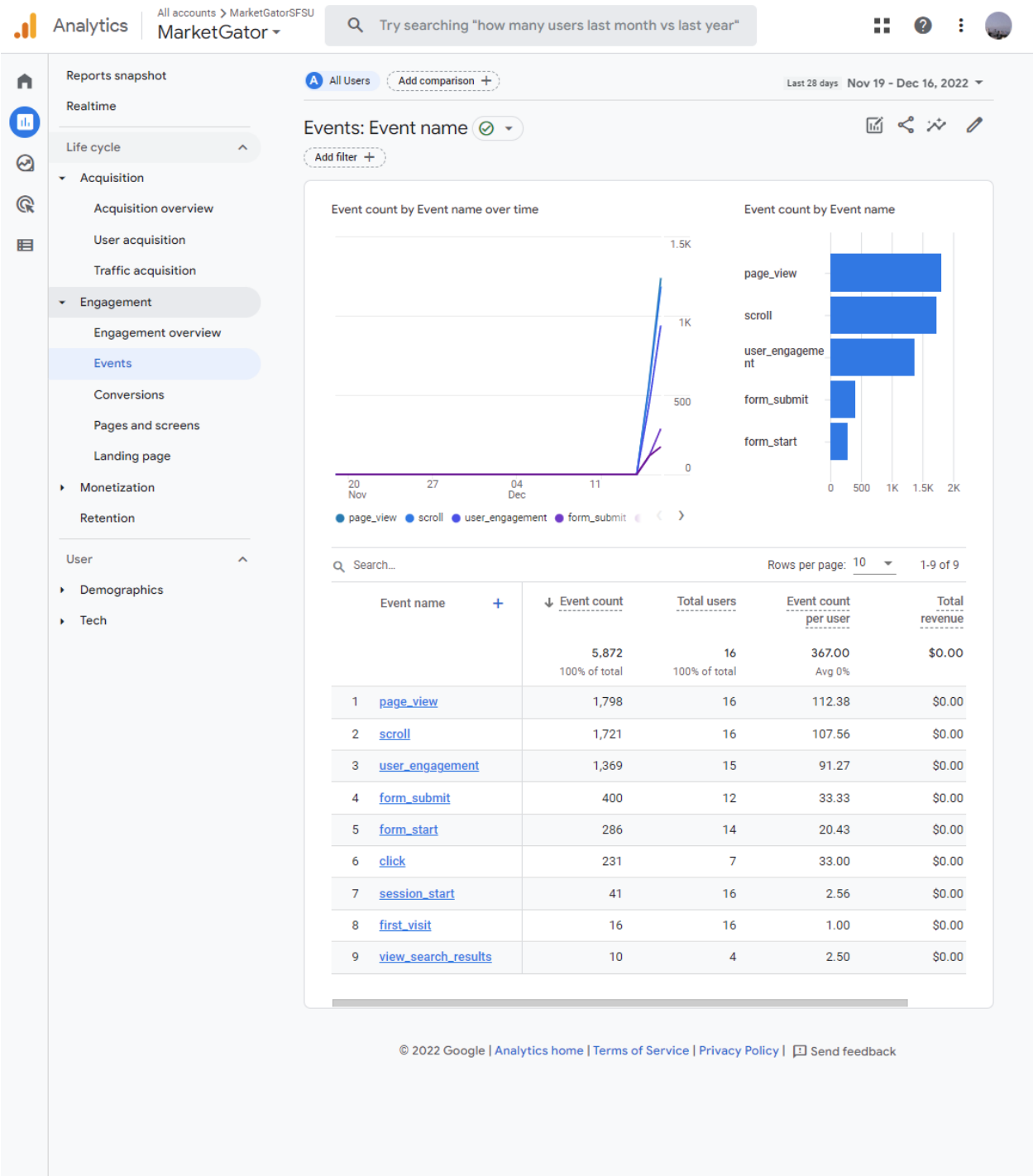
No data available

WHAT ARE YOUR TOP SELLING PRODUCTS?

Ecommerce purchases by Item name

ITEM NAME	ECOMMERCE PURCHASES

No data available


Analytics
All accounts > MarketGatorSFSU
MarketGator
Try searching "how many users last month vs last year"
Reports snapshot
Realtime
Life cycle
Acquisition
Acquisition overview
User acquisition
Traffic acquisition
Engagement
Engagement overview
Events
Conversions
Pages and screens
Landing page
Monetization
Retention
User
Demographics
Tech
All Users
Add comparison
Last 28 days Nov 19 - Dec 16, 2022
Events: Event name
Add filter
Event count by Event name over time
1.5K
1K
500
0
20 Nov
27
04 Dec
11
page_view
scroll
user_engagement
form_submit
Event count by Event name
page_view
scroll
user_engagement
form_submit
form_start
0
500
1K
1.5K
2K
Search...
Rows per page: 10
1-9 of 9
Event name
Event count
Total users
Event count per user
Total revenue
5,872
100% of total
16
100% of total
367.00
Avg 0%
$0.00
1 page_view 1,798 16 112.38 $0.00
2 scroll 1,721 16 107.56 $0.00
3 user_engagement 1,369 15 91.27 $0.00
4 form_submit 400 12 33.33 $0.00
5 form_start 286 14 20.43 $0.00
6 click 231 7 33.00 $0.00
7 session_start 41 16 2.56 $0.00
8 first_visit 16 16 1.00 $0.00
9 view_search_results 10 4 2.50 $0.00
© 2022 Google | Analytics home | Terms of Service | Privacy Policy | Send feedback

8. Project management:

Tools used in the implementation of this projects are following:

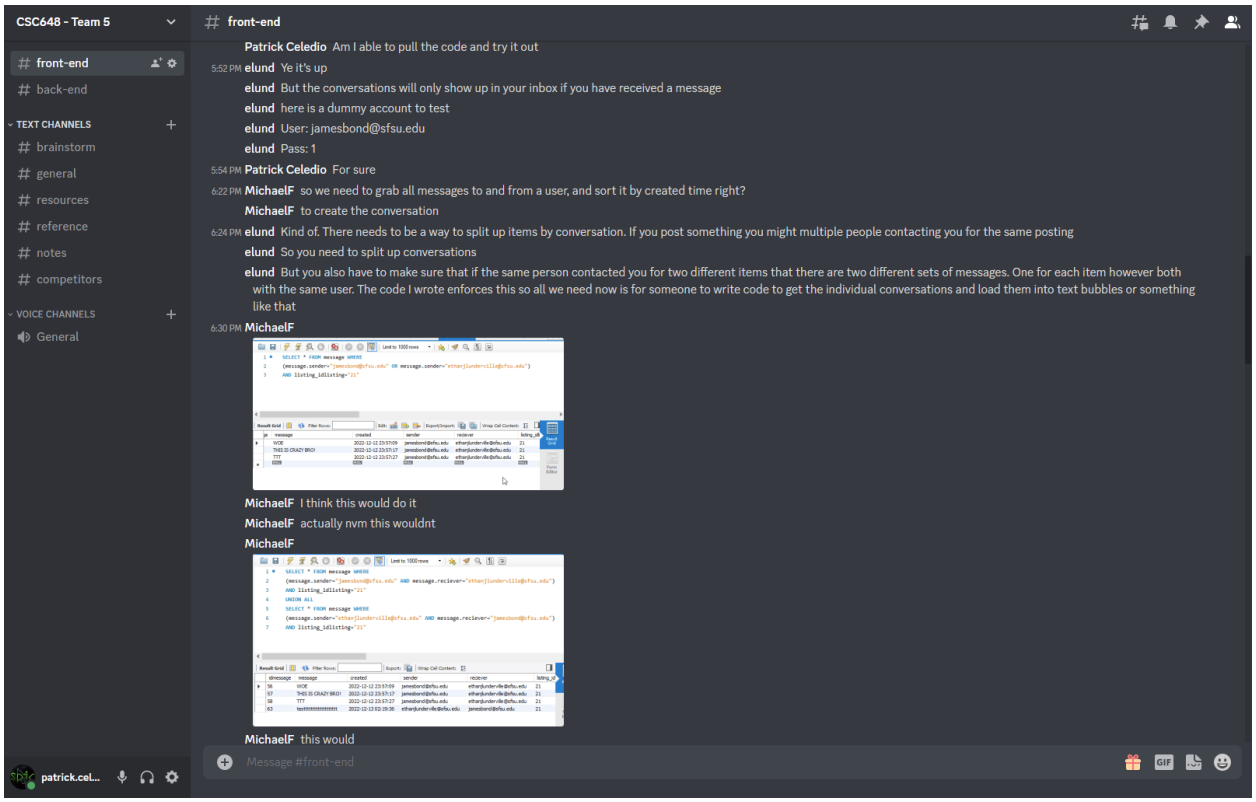
a. Visual Studio Code (programming languages: Javascript, HTML, CSS and SQL)

b. MySQL Workbench (Database)

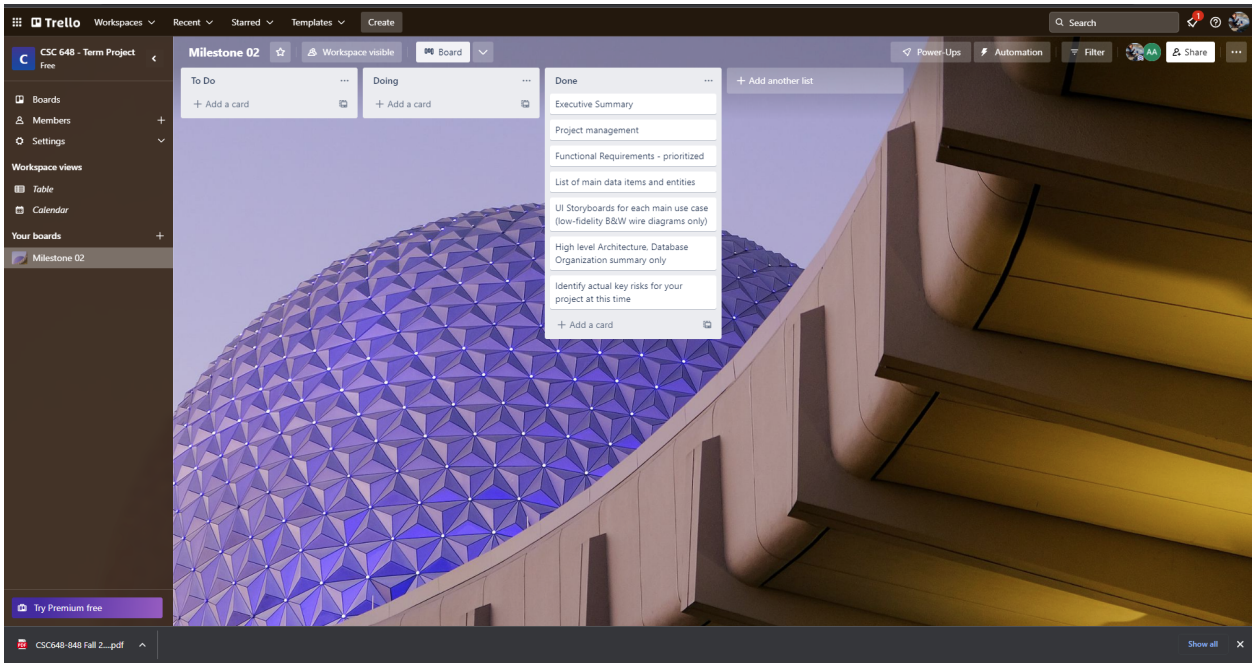
c. AWS

d. Google Analytics

e. Discord (Ad-hoc group conversation)



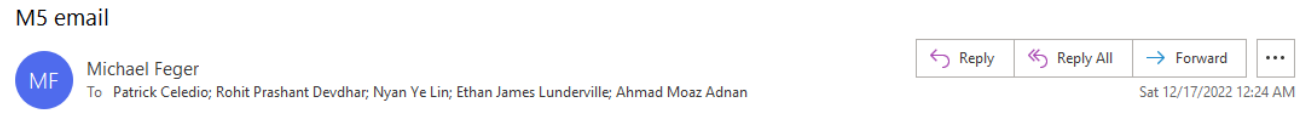
f. Trello (Project task management)



9. Team member self assessment and contributions:

Team member Emails with a list of what our contribution was and what was challenging during the project.

Michael Feger



a. Contributions for Michael Feger:

- Created remote Amazon server and resolved several issues with code integration
- Helped manage/edit SQL database
- Linked create listing from backend to frontend
- Created and implemented search algorithm
- Listing model, listing middleware, and listing routes
- Message model, message middleware, and message routes
- Frontend flash messages for success/errors
- QA Testing

b. Total GitHub commits: 43

c. The main challenge was figuring out how to split up the work efficiently and merge our code. It was my first time working with a big group on GitHub, so there was quite a bit of learning involved. I did not commit or pull from the main branch frequently enough and ended up with large commits that involved unnecessary merge conflicts, which wasted my time.

a. Next time, I will commit more frequently. I would do more expensive unit testing before having it merged with the main branch, as a few bugs slipped by in the search algorithm. I also should have done more and asked for more code reviews. Learning more about how GitHub works will also be valuable.

Ahmad Adnan

M5 email

AA

Ahmad Moaz Adnan

To: Patrick Celedio; Ethan James Lunderville; Nyan Ye Lin; Rohit Prashant Devdhar; Michael Feger

Sat 12/17/2022 8:33 AM

a. Contribution of Ahmad Adnan

- Created and designed dashboard front end
- Designed create listing page
- Fixed login and registration page front end text box
- Fixed minor front end design issues in project

b. Total commits in GitHub: 11

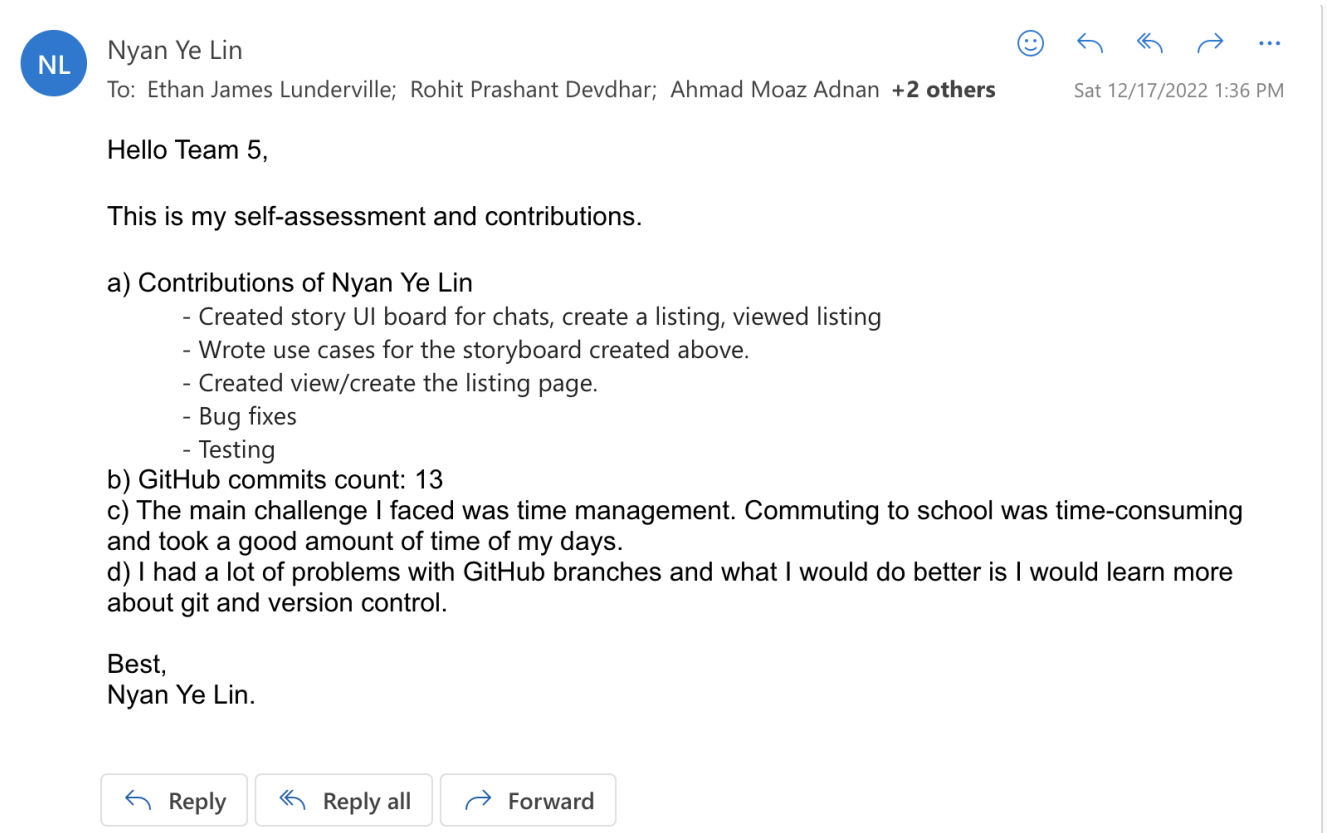
c. The main challenge that I believe encountered during this project was working as a group more efficiently in a group setting and dividing the workload evenly with teammates and get the tasks completed on time. I think we should have put more checkpoint to see where we are with our task early in the milestone, so we didn't have to some stuff at the end.

d. I should have committed my tasks more frequently. I mainly pushed my code to GitHub when I was done with my whole task. In future I will do more commits frequently.

Best,
Ahmad

Reply | Reply all | Forward

Nyan Ye Lin



NL

Nyan Ye Lin

To: Ethan James Lunderville; Rohit Prashant Devdhar; Ahmad Moaz Adnan **+2 others**

Sat 12/17/2022 1:36 PM

Hello Team 5,

This is my self-assessment and contributions.

a) Contributions of Nyan Ye Lin
- Created story UI board for chats, create a listing, viewed listing
- Wrote use cases for the storyboard created above.
- Created view/create the listing page.
- Bug fixes
- Testing

b) GitHub commits count: 13
c) The main challenge I faced was time management. Commuting to school was time-consuming and took a good amount of time of my days.
d) I had a lot of problems with GitHub branches and what I would do better is I would learn more about git and version control.

Best,
Nyan Ye Lin.

Reply | Reply all | Forward

Rohit Prashant Devdhar

M5 Email

RD

Rohit Prashant Devdhar

To: Patrick Celedio; Michael Feger; Ahmad Moaz Adnan; Nyan Ye Lin; Ethan James Lunderville

Fri 12/16/2022 10:54 PM

a. Rohit Devdhar's contributions to the project:

- CSS and HTML of home page (background image, cards, etc..)
- Changing minor front end details of other pages (login page, view listing page)

b. Total number of commits to main branch:
Should be around 14-15 commits when all said and done.

c. Main challenges encountered
I think the main challenges I encountered were just not doing enough work when it was earlier in the project. If I had done more work
when we were just beginning the project, we would have been less stressed to finish the project on time and could have used the last days to work out
the fine print of the project. Other than that, I was pretty familiar with git from previous projects, but this was the largest group I've used it with and
it was definitely challenging finding out what was the most recent branch to pull from. On that end, we could have been more organized and specific on which
files we were working on so that we did not have a lot of conflicts during merges and such.

d. What I would do better
Like I said above, I would definitely start earlier on the work next time so that there would be less of a time crunch down the road. I would try to
find more ways to contribute to the team, so we were on track for all milestones (we needed a checkpoint for Milestone 3). Other than that, I think we had a
solid group, with better cohesiveness we probably could have done a little better.

Ethan James Lunderville

M5 Email

EL

Ethan James Lunderville

To: Patrick Celedio; Ahmad Moaz Adnan; Rohit Prashant Devdhar; Michael Feger; Nyan Ye Lin

Fri 12/16/2022 11:51 PM

a) Ethan Lunderville project contribution:

- Created database
- Set up MVC application
- Login, Sessions and registration
- Message inbox logic for both front and backend
- Restyled all pages to be more modern
- Created cards for listings
- Refractored app to maintain architecture

b) GitHub commit count: 12

c) The main challenge we encountered was working together in an organized way. This is understandable since we all have different styles however at times this lack of consistent patterns in the code made it harder to understand what was going on and make changes to the code.

d) I think I should have split my commits into smaller chunks as opposed to just submitting huge parts of the project all at one time. That way, if a piece of my code had a bug in it, it would be easier to track down what went wrong. I also think I should have used my own branch more instead of just pushing to main all of the time.

Schedule a meeting

Reply Reply all Forward

Patrick Celedio

M5 - Team member self assessment and contributions

PC

Patrick Celedio

To: Ethan James Lunderville; Michael Feger; Ahmad Moaz Adnan; Rohit Prashant Devdhar; Nyan Ye Lin

Cc: Patrick Celedio

Sat 12/17/2022 1:08 PM

Hello Team 05,

This is my email of the team member self-assessment and contributions:

a)

- Front-end CSS styling and rendering data from Handlebars routes such as checking if the user is currently in a logged in session
- Implemented nav bar CSS and responsive search bar CSS
- Implemented front-end password validation CSS
- Implemented skeleton HTML and front-end foundation for main.hbs, partials, about pages, and authenticated and unauthenticated pages
- Implemented front-end search functionality foundation
- Refactored project directory
- Worked on and revised milestone documents

b) Number of submissions to Team 05 GitHub branch: 116 commits

c) The main challenges I encountered from this team project was managing a team of six including myself as a team lead and balancing my part-time job and four other computer science classes. It was my first time being a leader of a project and I learned that it requires a lot of advanced people skills. In the beginning, my way of leading was based more on the assumption that that my group members will be on top of their tasks and that they already knew what to do and what must be fulfilled. However, there were occurrences where the work effort felt uneven and some were even missing towards the end of the project. I am partly to blame as I should have been more firm with assigning tasks to the group members, checking in with them by having SCRUM meetings, and assigning smaller milestones. How the team actually operated was that the individuals who were active and cared about the project volunteered and pulled their weight while others remained silent at times. It felt like a brute force approach by a few individuals more than an organized work effort, and that is an experience I learned from and would not want to repeat again if I am ever in a team lead position again.

Balancing my part-time job and four other computer science classes was very challenging. This has been probably the most intense four months of my life so far. Every week I had to handle deliverables from both my work and school, and towards the end of this semester I felt exhausted. However, I am proud of the work that was accomplished, especially with the help of my team members.

d) What I would do better next time based on what I learned in this class about SE management and processes would definitely be being more firm with checking in on team members and managing my time more efficiently on working on the application. I feel like after this experience of being a team lead for a software engineering project, there is a lot of executive management that has to be done every day such as checking in on team members and making sure we are on track with milestones. There has also been times where I would underestimate the time given before a deadline of a milestone and with the circumstances of my schedule and everyone else's, I would have done better at coordinating with everyone about a deadline and also figuring out ways to carve out time from my own schedule if possible.

I would also enforce more code review and daily check-ins on the work that was being submitted into the GitHub repo. There were times during the process of creating MarketGator where there would be merge conflicts or front-end code that was not presentable. What the team started doing later in the process was submitting their code on their repo and posting screenshots on our Discord channel so its easy for all the team members to review the code and application. I found that very effective and would incorporate that into SCRUM meetings in the future if I were to facilitate one.

Best regards,
Patrick Celedio
Team 05 Lead

Reply | Reply all | Forward