

CSC648-848

Spring 2025

RedJr: SFSU Marketplace

Ria Thakker (Team Lead, rthakker@sfsu.edu), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), Eric/Po-Han Chen(Frontend Dev)

<http://18.191.88.25/>

Milestone 5, 05/20/2025

History

| | |
|----------------|------------|
| Date Submitted | 05/20/2025 |
| Date Revised | |

01. Product Summary

Name of the project: RedJR Marketplace

Major committed P1 functions:

- Registration with SFSU email
- Browse and search listings and descriptions
- Upload item listings to sell
- Messaging to contact sellers about items
- Dashboard to view listings user is selling, saved items & interested listings, messages
- Message inbox to communicate with sellers and coordinate sales
- Admin approval of all posts
- Admin moderation of users (delete/ban problematic users)
- SFSU Specific: Solely SFSU user base; must have a verified SFSU email to register, post listings, message other users. Admins approve all posts to screen for safety and legitimacy.

Product url: <http://18.191.88.25/>

02. Milestone Documents

Milestone 1 Document

SW Engineering

CSC648-848

Spring 2025

RedJr: SFSU Marketplace

Ria Thakker (Team Lead, rthakker@sfsu.edu), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), Eric/Po-Han Chen

Milestone 1

History

| | |
|----------------|------------|
| Date Submitted | 03/12/2025 |
| Date Revised | 03/17/2025 |

01. Executive Summary

In today's fast-paced university environment, students and faculty often struggle to find affordable and convenient ways to buy and sell essential items like textbooks, electronics, furniture, and learning materials. While general online marketplaces exist, they are not tailored to students, and can have unrelated listings, scams, and unreliable sellers. RedJr is a platform designed specifically for the San Francisco State University (SFSU) community, offering an easy-to-use and trusted marketplace where students and faculty can connect to buy and sell items.

RedJr provides essential e-commerce functionalities, including item browsing, keyword-based searching, and direct communication between buyers and sellers. Sellers can effortlessly upload product listings and manage them through a personalized dashboard, while buyers can explore verified items with confidence. A key differentiator of our platform is its high degree of safety due to SFSU-only verification and administrator monitoring of all content. To further distinguish itself from existing platforms, RedJr features student-specific tools such as the ability to rent items from other members of the SFSU community, the ability to set up sales on campus at specific locations, and a rating system for both buyers and sellers to provide feedback on their experiences.

RedJr is a project built by SFSU students for the SFSU community. It's developed by a passionate team: Ria Thakker (Team Lead), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), and Eric Chen (Frontend Contributor). As students ourselves, we are familiar with the challenges faced by college students. Our mission is to reduce waste by promoting selling and giving away old items and to provide an affordable, reliable solution for buying and selling needs to the campus community.

02. Personae

Persona 1

Griffey is a third-year student at San Francisco State University who considers himself financially literate, tech savvy, and organized. He is competent when it comes to websites and can usually figure out how to navigate them if a website is structured properly.

Griffey loves making extra money whenever possible and always looks for ways to get the best value for his items. When selling, he is usually willing to let an item sit until it goes for the price listed.

A pain point of his is that it is the end of the semester and he needs to sell his dorm items quickly before moving out. He doesn't want to deal with the hassle of storing or transporting large furniture and electronics. Also, **Griffey** is currently without a vehicle and does not want to use public transportation to take large items to meet with a prospective buyer.

The goal of using our app to list his items for sale is that because the app is for students attending his college, it is likely that there are students in need of similar items he owns, and being in close proximity, they can quickly purchase and retrieve the item.

Persona 2

Emily is an 18-year-old freshman at San Francisco State University, majoring in Business Administration. She recently moved from Seattle, Washington, and is adjusting to college life and living independently in a campus dorm. While excited about this new chapter, she quickly realizes her dorm lacks essential furniture like a desk, chair, and shelves, making it difficult to study and stay organized. She also wants to add personality to her space without overspending.

Emily is tech-savvy and frequently shops on Amazon, Facebook Marketplace, and Depop, but she prefers simple, easy-to-navigate websites. She needs a reliable and affordable way to buy second-hand furniture and textbooks from other students to stay within her budget.

Without a car, she requires a platform that allows her to coordinate safe, on campus meetups for item pickup.

Her main goals include finding reasonably priced textbooks and furniture through a trusted SFSU exclusive marketplace that ensures legitimate listings and safe transactions. However, she has concerns about scams, misleading posts, and the inability to inspect items before purchase. Transportation limitations also make it difficult for her to pick up larger items or travel far for transactions.

Persona 3

Mike is a second-year computer science student at SFSU who is an administrator for the RedJr site. He is a developer, so is tech-savvy and intimate with the details of the website, its structure, and has access to internal documentation that will assist him with his job.

Mike's job is to go through all new listings that are posted to the site and approve them for public viewing. One of his pain points is that it is tedious to manually view and approve each post, so he wants the process to be as fast and uncomplicated as possible.

Persona 4

Olivia is a 20-year-old junior at San Francisco State University, majoring in Environmental Studies. With a deep passion for sustainability, she actively engages in campus recycling initiatives. She often buys and sells used textbooks, electronics, and everyday items to support eco-friendly practices while saving money. Olivia appreciates discovering quality second-hand items that not only help reduce waste but also provide her with the opportunity to earn some extra cash by selling her pre-owned belongings. To ensure a seamless and

secure experience, she prefers to use a faculty-supported platform that guarantees reliable transactions with fellow students.

Olivia faces challenges using general online marketplaces due to their chaotic nature and lack of moderation, making it hard to assess item conditions and find reliable deals. She's also wary of scams and miscommunications, leading to her reluctance to use larger, non-student-focused platforms.

Persona 5

Jake is a senior Computer Science student at San Francisco State University who loves making extra money through buying and selling items. He frequently purchases underpriced goods, refurbishes them, and resells them at a higher price. Always on the lookout for deals, he refreshes the website constantly to catch new listings before others.

Jake values a smooth selling process and prefers a clean, intuitive interface. While he usually waits for the best offer, he sometimes needs to sell items quickly, especially as graduation approaches. A major pain point for him is dealing with unreliable buyers who back out last minute.

His goal is to efficiently buy and sell within the SFSU community, avoiding the hassle of external marketplaces while ensuring quick and profitable transactions.

03. Use Cases

Use Case 1 - Selling Items

Griffey is a student **seller** on the website because the semester is over and he needs to move out of his dorm. He is looking to list large items that he does not wish to take with him for sale to San Francisco State **buyers**. After arriving at the homepage and logging into his personal account, he navigates to his dashboard and visits the **Listings** page, which displays the items he has listed for sale.

On this page, he clicks **New Listing** and enters the item details, including:

- Item Name
- Category (Furniture, Tech, Clothing, etc.)
- Brief Description
- Condition (Brand New, Used but Like New, Some wearing, etc.)
- Photos of the Item

- Asking Price

After entering all the required information, he clicks **Publish** and the item is now listed for sale on the site. To keep track of buyers interested in his items, Griffey can visit the listings page. If a buyer has messaged him about an item, the listing will display the number of interested buyers. Clicking on this number takes him to the **messages** page, where he can view messages related to that particular item. Griffey can also go directly to the Messages page from the **navigation bar**, where he can see all messages he has received, including ones from potential buyers. If Griffey and a buyer agree on a price and pickup location through messaging, he marks the item as **sold** so that no other buyers can message about it.

Use Case 2 - Buying Items

Emily recently moved into an SFSU dorm and needs a desk and chair for her room. She wants to find a **second-hand set** from another student instead of buying new.

After arriving at the **homepage** and logging into her **personal account**, she navigates to the **Listings page** and selects the "**Furniture**" **category**. She uses **filters** to narrow her search by **price range** and **location**, preferring **sellers located on campus**.

She finds a **listing** for a desk and chair that fits her budget. She clicks on the listing to view details, including:

- Item Name
- Category (Furniture, Tech, Clothing, etc.)
- Brief Description
- Condition (Brand New, Used but Like New, Some wearing, etc.)
- Photos of the Item
- Asking Price

After reviewing the details, Emily uses the **built-in messaging system** to ask the **seller** about the **furniture's condition** and confirm its **availability**. The **seller responds**, and they agree on a **price** and **designated on-campus location** for pickup.

Emily meets the seller, inspects the furniture, and decides to buy it. She **pays via Venmo or cash**, takes the items, and the **seller marks the listing as "Sold"** so no other buyers can message about it.

Use Case 3 - Administrator Functions

Mike is an **administrator** for RedJr and is responsible for **managing item listings** and **user interactions** on the platform.

After arriving at the **homepage** and logging into his **administrator account**, he navigates to the **admin dashboard** and visits the **Listings Approval page**, which displays all **newly submitted item listings** awaiting approval.

On this page, he **reviews** each **pending listing** and can either **approve** or **deny** it. If a listing is **denied**, he provides **feedback** to the **seller**, explaining the **reason for rejection**. Items that are **approved** are **published on the site for buyers** to view.

To maintain **platform security**, Mike also **monitors user interactions**. He visits the **Messages page**, where he can:

- Respond to support queries
- Review chat logs for inappropriate content
- Take action on reports of policy violations
- If he identifies a violation, Mike can issue a warning, temporarily suspend, or permanently ban a user.

Use Case 4 - Creating Account

Olivia is a student who wants to **buy and sell second-hand items** on the **marketplace**. She learns about the platform through a flyer on campus and **positive feedback from her peers**.

After arriving at the **homepage**, she clicks **Sign Up** to register for an account. She completes the **registration process** by entering the following details:

- University Email (for verification)
- Username
- Password
- Profile Information

After **creating her account**, Olivia **logs** in and browses the **Listings page**. She **selects an item** and uses the **messaging system** to ask the seller about its condition.

After agreeing on a price, Olivia meets the seller on campus for **pickup** and leaves a review to help build community trust.

Use Case 5 - Getting updated listings

Jake logs into his account. After he arrives at the homepage and logs into his personal account, he navigates to his **Seller Dashboard** and visits the **Listings page**. He clicks **New Listing** to create a new item post, entering the following details:

- Item Name
- Category (Furniture, Tech, Clothing, etc.)
- Brief Description
- Condition (Brand New, Used but Like New, Some wearing, etc.)
- Photos of the Item
- Asking Price

Once satisfied, he clicks **Publish** to list the item for sale. To monitor interest, Jake frequently checks his Listings page, where he can see the number of buyers who have messaged him

about an item. Clicking on this number takes him to the **Messages** page, where he can negotiate with potential buyers.

At the same time, Jake frequently refreshes the site to check for new listings from other users. He looks for low-priced items, refurbishes them, and relists them at a higher price for profit. Once a deal is finalized, he marks the item as **Sold** to prevent further inquiries.

04. Data Items & Entities

| User | Description |
|---------------|---|
| Administrator | User responsible for system maintenance, views a different UI compared to other users |
| Buyer | A user that has made an offer on an item and can rent listings |
| Seller | A user that has items up for sale or has put a listing up for rent |

| Entities | Description |
|----------|---|
| item | Can be listed and bought from users and includes: <ul style="list-style-type: none">● Title: Name of the item● Category: Furniture, Tech, Clothing, etc.● Description: Brief details about the item● Condition: Brand New, Used but Like |

| | |
|---------------|--|
| | <p>New, Some Wearing, etc.</p> <ul style="list-style-type: none"> ● Photos: Images of the item ● Price: The amount the seller is asking for |
| Offer | Bid on an item that's up to be sold |
| Rental item | <p>An item that is only being offered for a temporary time then returned to seller</p> <p>Includes</p> <ul style="list-style-type: none"> ● Title: Name of the item ● Category: Furniture, Tech, Clothing, etc. ● Description: Brief details about the item ● Condition: Brand New, Used but Like New, Some Wearing, etc. ● Photos: Images of the item ● Rental period: How long the item will be with the buyer ● Rental rate: Price of renting the item through one rental period |
| Sale | <p>A pre-arranged meetup for buyers and sellers to exchange items and payment. Based on the outcome/fulfillment of both parties, the users' ratings are impacted.</p> <p>Includes</p> <ul style="list-style-type: none"> ● Meeting location ● Meeting time ● Post-sale, buyer rating ● Post-sale, seller rating |
| Seller Rating | Score out of 5 buyers give based on quality of transaction process |
| Buyer Rating | Score out of 5 sellers give based on quality of transaction process |

| Data Structure | Description |
|-------------------|--|
| Messaging | Log of chats sent between buyer and seller users |
| Locations | List of on-campus areas that a sale/pickup is able to be fulfilled |
| Items bought/sold | Structure contains list of items a user has purchased and sold on the site |

05. Functional Requirements

1. Unregistered users shall only be able to access the register/login/home page/about pages.
2. To register, a user shall verify their sfsu.edu email.
3. A registered user shall be able to browse and search for items listed on the website and descriptions of the items.
4. A registered user shall be able to upload item listings and descriptions for sale.
5. A registered user shall be able to make an offer to purchase an item of their choice.
6. A registered user shall be able to edit their item listings after uploading them.
7. A registered user shall be able to edit an offer they have made on an item.
8. A registered user shall be able to cancel an item listing and remove it from public view even if it is not sold.
9. A registered user shall be able to cancel an offer on an item at any time before purchase is finalized.
10. A registered user shall be able to message another registered user.
11. A registered user shall have a dashboard for their posted items and view messages from prospective buyers.
12. Registered users who have agreed to a purchase shall be able to create a sale that contains the meeting location on campus and time.
13. A registered user who has sold an item shall be able to provide feedback and contribute to the rating of the user who has bought their item after a sale.
14. A registered user who has bought an item shall be able to provide feedback and contribute to the rating of the user who has sold the item after a sale.
15. A registered user shall be able to mark items as sold and remove them from public view after a sale is finalized.

16. A registered user shall be able to report an item to an administrator if they feel the item violates a term or condition of the website.
17. A site administrator shall be able to approve all items uploaded to the site before they are available for public view.
18. A site administrator shall be able to delete items deemed inappropriate from public view.
19. A site administrator shall be able to delete/ban users that have been reported, violating platform policies.
20. A site administrator shall be able to view the chat logs of registered users.

06. Non-functional requirements

1. Application shall be developed, tested and deployed using tools and cloud 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 shall render well on mobile devices (no native app to be developed)
4. Posting of sales information and messaging to sellers shall be limited only to SFSU students
5. Critical data shall be stored in the database on the team's deployment server.
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application shall follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used
13. No e-mail clients or chat services 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
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
16. Media formats shall be standard as used in the market today
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools
18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only" at the top of the WWW page Nav bar.

07. Competitive Analysis

| Features | facebook Marketplace | craigslist | Offerup | LetGo | RedJr |
|--|----------------------|------------|---------|-------|-------|
| User Verification | + | - | + | + | ++ |
| User Interface | + | + | + | + | ++ |
| Internal Messaging | + | - | - | + | ++ |
| Admin Communication and customer support | + | + | - | + | ++ |
| Community Integration | - | - | - | - | ++ |
| Safety & trust | + | - | + | + | ++ |

+ Has feature, - feature not included, ++ superior

Analysis Summary

Craigslist lacks an internal messaging system, requiring users to communicate via email, whereas platforms like Facebook Marketplace, OfferUp, and Letgo provide built-in messaging for convenience. Our website will include an internal messaging system to streamline communication between users. Also, craigslist design is very minimal compared to Facebook Marketplace and OfferUp's more modern interfaces, which require photos for listings and offer a smoother user experience. A common complaint among Craigslist users is the inability to edit listings, an issue we aim to resolve. User verification is a major weakness on Craigslist, Facebook Marketplace, OfferUp, and Letgo, as they only require an email for registration, allowing scammers to create temporary or fake accounts. For our website, we will require an official SFSU email, ensuring that each user is a verified student or faculty member, reducing fake accounts and scams. Because our website is only SFSU students or faculty, we can have more responsive admin support, unlike larger platforms where direct communication with administrators is often slow or ineffective. Unlike these broader marketplaces, our site is specifically designed for the SFSU community, allowing us to adjust features to user needs and provide a safer, more efficient experience.

08. High-level System Architecture & Technologies Used

- Server Host: Amazon Web Services
- Operating System: Ubuntu 16.04 Server
- Database: MySQL v 8.0.41

- Web Server: Apache2
- Server-Side Language: Python
- Additional Technologies: Web Framework: Flask
 - IDE: JetBrains PyCharm, Visual Studio Code
 - Web Analytics: Google Analytics
 - SSL Cert: Let's Encrypt (Cert Bot)
 - HTML, CSS, Javascript (frontend)
- Browsers Supported: Google Chrome 134.0, Firefox 136.0

09. Use of GenAI Tools

- Tool used: ChatGPT-4-Turbo
- Tasks used:
 - Executive Summary: HIGH usefulness. Streamlined the writing process, and only minor modifications were needed to increase brevity and specificity with regards to the SFSU-related functions and some requirements. I didn't have to spend more than 15 minutes on the executive summary because its content was useful on the first go-around.
 - Personae, Use Cases: MEDIUM usefulness. ChatGPT provides a solid starting point for defining user personas and creating structured use cases. However, some personas fell short in addressing the specific challenges faced by SFSU students, so I had to refine the descriptions.
- Example prompts:
 - Can you generate an executive summary for a project with the following requirements [Copied & pasted functional requirements from project overview, excluded for space]

The executive summary must contain the following: [Description of executive summary from M1 overview, excluded for space]

10. Team & Roles

| Team Member | Email | Role |
|--------------------|--------------------|----------------------|
| Ria Thakker | rthakker@sfsu.edu | Team Lead |
| Rohith Gannoju | rgannoju@sfsu.edu | Backend Lead |
| Danny Duenas | dduenas@sfsu.edu | Frontend Lead |
| James Richards | jrichards@sfsu.edu | GitHub Master |
| Eric/Po-Han Chen | pchen@sfsu.edu | Frontend Contributor |

11. Team Lead Checklist

- So far, all team members are fully engaged and attending team sessions when required
 - DONE
- Team found a time slot to meet outside of the class
 - DONE - Fridays 3pm - 4/5pm
- Team ready and able to use the chosen back and front-end frameworks and those who need to learn are working on learning and practicing
 - ON TRACK
- Team reviewed class slides on requirements and use cases before drafting Milestone 1
 - DONE
- Team reviewed non-functional requirements from “How to start...” document and developed Milestone 1 consistently
 - DONE
- Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission
 - DONE
- Team lead ensured that all team members read the final M1 and agree/understand it before submission
 - DONE
- Team shared and discussed experience with GenAI tools among themselves
 - DONE
- GitHub organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
 - ON TRACK

SW Engineering

CSC648-848

Spring 2025

RedJr: SFSU Marketplace

Ria Thakker (Team Lead, rthakker@sfsu.edu), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), Eric/Po-Han Chen(Frontend Dev)

Milestone 2 Part I, 03/25/2025

History

| | |
|----------------|------------|
| Date Submitted | 03/25/2025 |
| Date Revised | 04/01/2025 |

01. Executive Summary

In today's fast-paced university environment, students and faculty often struggle to find affordable and convenient ways to buy and sell essential items like textbooks, electronics, furniture, and learning materials. While general online marketplaces exist, they are not tailored to students, and can have unrelated listings, scams, and unreliable sellers. RedJr is a platform designed specifically for the San Francisco State University (SFSU) community, offering an easy-to-use and trusted marketplace where students and faculty can connect to buy and sell items.

RedJr provides essential e-commerce functionalities, including item browsing, keyword-based searching, and direct communication between buyers and sellers. Sellers can effortlessly upload product listings and manage them through a personalized dashboard, while buyers can explore verified items with confidence. A key differentiator of our platform is its high degree of safety due to SFSU-only verification and administrator monitoring of all content. To further distinguish itself from existing platforms, RedJr features student-specific tools such as the ability to rent items from other members of the SFSU community, the ability to set up sales on campus at specific locations, and a rating system for both buyers and sellers to provide feedback on their experiences.

RedJr is a project built by SFSU students for the SFSU community. It's developed by a passionate team: Ria Thakker (Team Lead), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), and Eric Chen (Frontend Contributor). As students ourselves, we are familiar with the challenges faced by college students. Our mission is to reduce waste by promoting selling and giving away old items and to provide an affordable, reliable solution for buying and selling needs to the campus community.

02. Data Items & Entities

| User | Description |
|---------------|---|
| Administrator | User responsible for system maintenance, views a different UI compared to other users |
| User | <p>General term for any non-administrator user who is registered and can buy or sell items</p> <ul style="list-style-type: none"> ● Buyer: A user that has made an offer on an item and can rent listings <ul style="list-style-type: none"> ○ Boolean indicating if the user is able to buy/rent items ● Seller: A user that has items up for sale or has put a listing up for rent <ul style="list-style-type: none"> ○ Boolean indicating if the user is able to sell/rent out items <p>A user is able to be both buyer and seller or can be one or the other.</p> |

| Entities | Description |
|----------|---|
| item | <p>Can be listed and bought from users and includes:</p> <ul style="list-style-type: none"> ● Title: Name of the item - mandatory ● Category: Furniture, Tech, Clothing, etc. - mandatory ● Description: Brief details about the item - mandatory <ul style="list-style-type: none"> ○ Minimum 50 characters ○ Maximum 1500 characters ● Condition: Brand New, Used but Like New, Some Wearing, etc. - mandatory ● Photos: Images of the item - mandatory <ul style="list-style-type: none"> ○ Must have at least one photo or video (limit 10MB) ● Price: The amount the seller is asking for - mandatory |
| Offer | Bid on an item that's up to be sold - |

| | |
|---------------|---|
| | mandatory to have a price |
| Rental item | <p>An item that is only being offered for a temporary time then returned to seller</p> <p>Includes</p> <ul style="list-style-type: none"> ● Title: Name of the item - mandatory ● Category: Furniture, Tech, Clothing, etc. - mandatory ● Description: Brief details about the item - mandatory <ul style="list-style-type: none"> ○ Minimum 50 characters ○ Maximum 1500 characters ● Condition: Brand New, Used but Like New, Some Wearing, etc. - mandatory ● Photos: Images of the item - mandatory <ul style="list-style-type: none"> ○ Must have at least one photo or video (limit 10MB) ● Rental period: How long the item will be with the buyer - mandatory <ul style="list-style-type: none"> ○ Minimum 1 hour ○ Maximum 1 week ● Rental rate: Price of renting the item through one rental period - mandatory |
| Sale | <p>A pre-arranged meetup for buyers and sellers to exchange items and payment. Based on the outcome/fulfillment of both parties, the users' ratings are impacted.</p> <p>Includes</p> <ul style="list-style-type: none"> ● Meeting location - mandatory ● Meeting time - mandatory ● Post-sale, buyer rating - optional ● Post-sale, seller rating - optional |
| Seller Rating | Score out of 5 buyers give based on quality of transaction process |
| Buyer Rating | Score out of 5 sellers give based on quality of transaction process |

| Data Structure | Description |
|-----------------------|--|
| Messaging | Log of chats sent between buyer and seller users |
| Locations | List of on-campus areas that a sale/pickup is able to be fulfilled |
| Items bought/sold | Structure contains list of items a user has purchased and sold on the site |

03. Functional Requirements, Prioritized

Priority 1

- Unreg user
 - 1. Unregistered users shall only be able to access the register/login/home page/listing page/about pages.
 - 2. Unregistered users can create an account using a verified SFSU email address.
 - 3. A registered/unregistered user shall be able to browse and search for items listed on the website and descriptions of the items.
- Reg user
 - 3. A registered/unregistered user shall be able to browse and search for items listed on the website and descriptions of the items.
 - 4. A registered user shall be able to upload item listings and descriptions for sale.
 - 5. A registered user shall be able to make an offer to purchase an item of their choice.
 - 10. A registered user shall be able to message another registered user.
 - 11. A registered user shall have a dashboard for their posted items and view messages from prospective buyers.
 - 12. Registered users who have agreed to a purchase shall be able to create a sale that contains the meeting location on campus and time.
- Admin
 - 17. A site administrator shall be able to approve all items uploaded to the site before they are available for public view.
 - 19. A site administrator shall be able to delete/ban users that have been reported, violating platform policies.

Priority 2

- Unreg user
- Reg user
 - 6. A registered user shall be able to edit their item listings after uploading them.
 - 7. A registered user shall be able to edit an offer they have made on an item.
 - 8. A registered user shall be able to cancel an item listing and remove it from public view even if it is not sold.
 - 9. A registered user shall be able to cancel an offer on an item at any time before purchase is finalized.
 - 15. A registered user shall be able to mark items as sold and remove them from public view after a sale is finalized.
 - 16. A registered user shall be able to report an item to an administrator if they feel the item violates a term or condition of the website.

- Admin
 - 18. A site administrator shall be able to delete items deemed inappropriate from public view.

Priority 3

- Unreg user
- Reg user
 - 13. A registered user who has sold an item shall be able to provide feedback and contribute to the rating of the user who has bought their item after a sale.
 - 14. A registered user who has bought an item shall be able to provide feedback and contribute to the rating of the user who has sold the item after a sale.
- Admin
 - 20. A site administrator shall be able to view the chat logs of registered users.

04. Storyboards

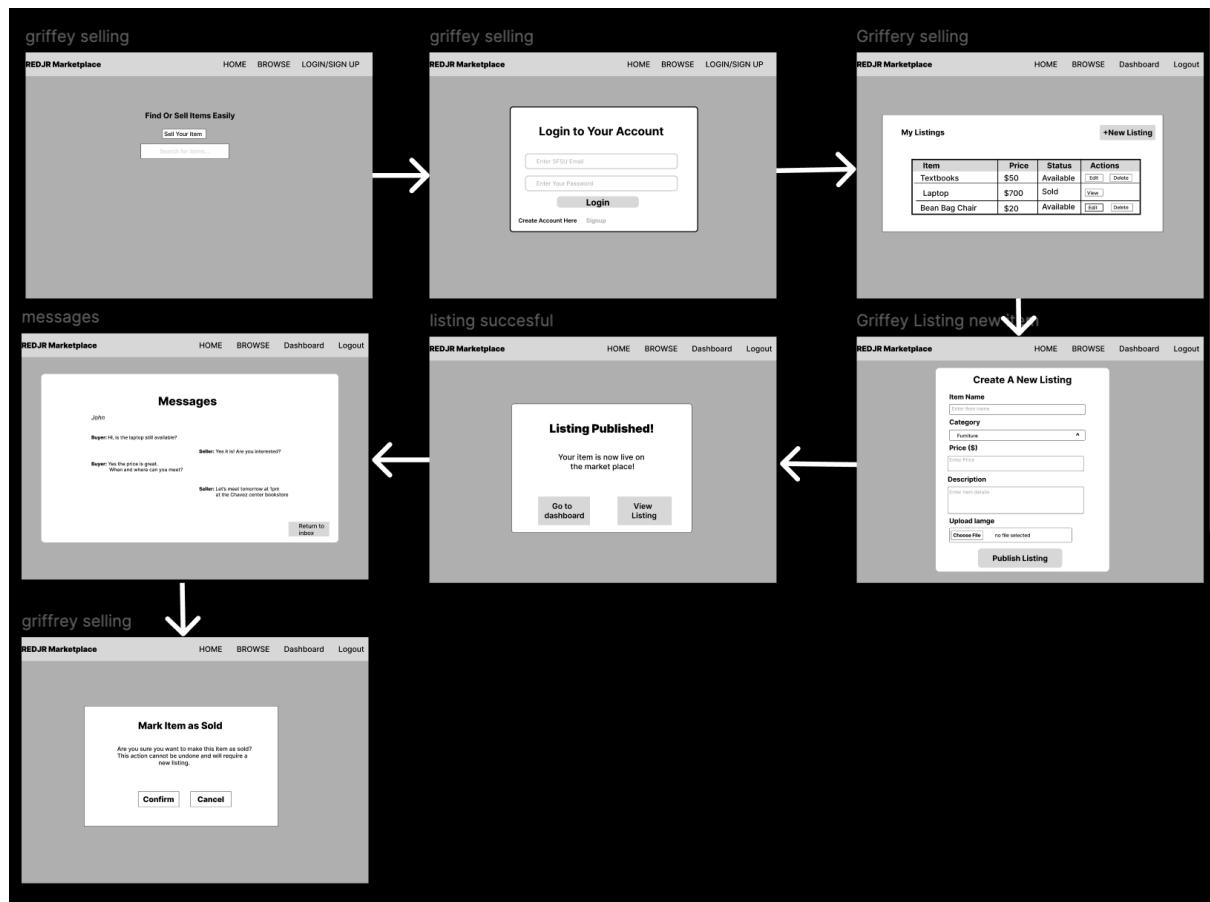
This presents the low-fidelity wireframes and UI storyboards for our REDJR Marketplace project. These designs focus on desktop browser interaction only and represent the key user flows identified in Milestone 1. The purpose of these mockups is to visualize the basic layout, navigation, and interaction flow without distractions like color or styling.

We created wireframes for the three major use cases:

- Griffey's selling flow
- Olivia's sign-up and buying flow
- Emily's buying flow

Griffey's Selling Flow:

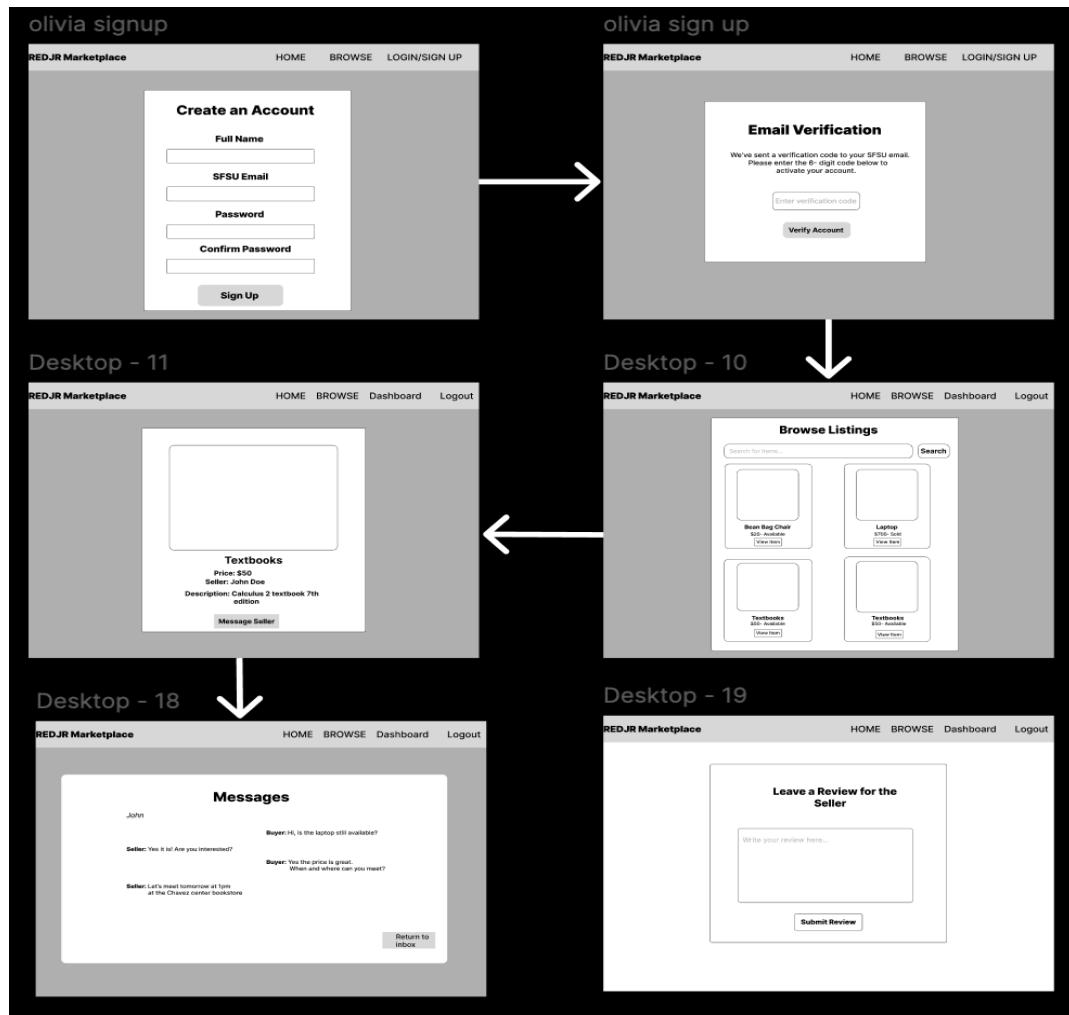
Use Case 1: Seller logs in, creates listings, and manages them.



In this mockup, Griffey accesses the site and is brought to the homepage which then he proceeds to login since he already has an account and existing listings. After he logs in he views his current listings. He also creates a new listing and finalizes it. Somebody messages him about an item and they agree to meet. After they meet and he sells the item, then marks the item as sold.

Olivia's Signup Flow

Use Case 2: Buyer registers, browses, and communicates with sellers.

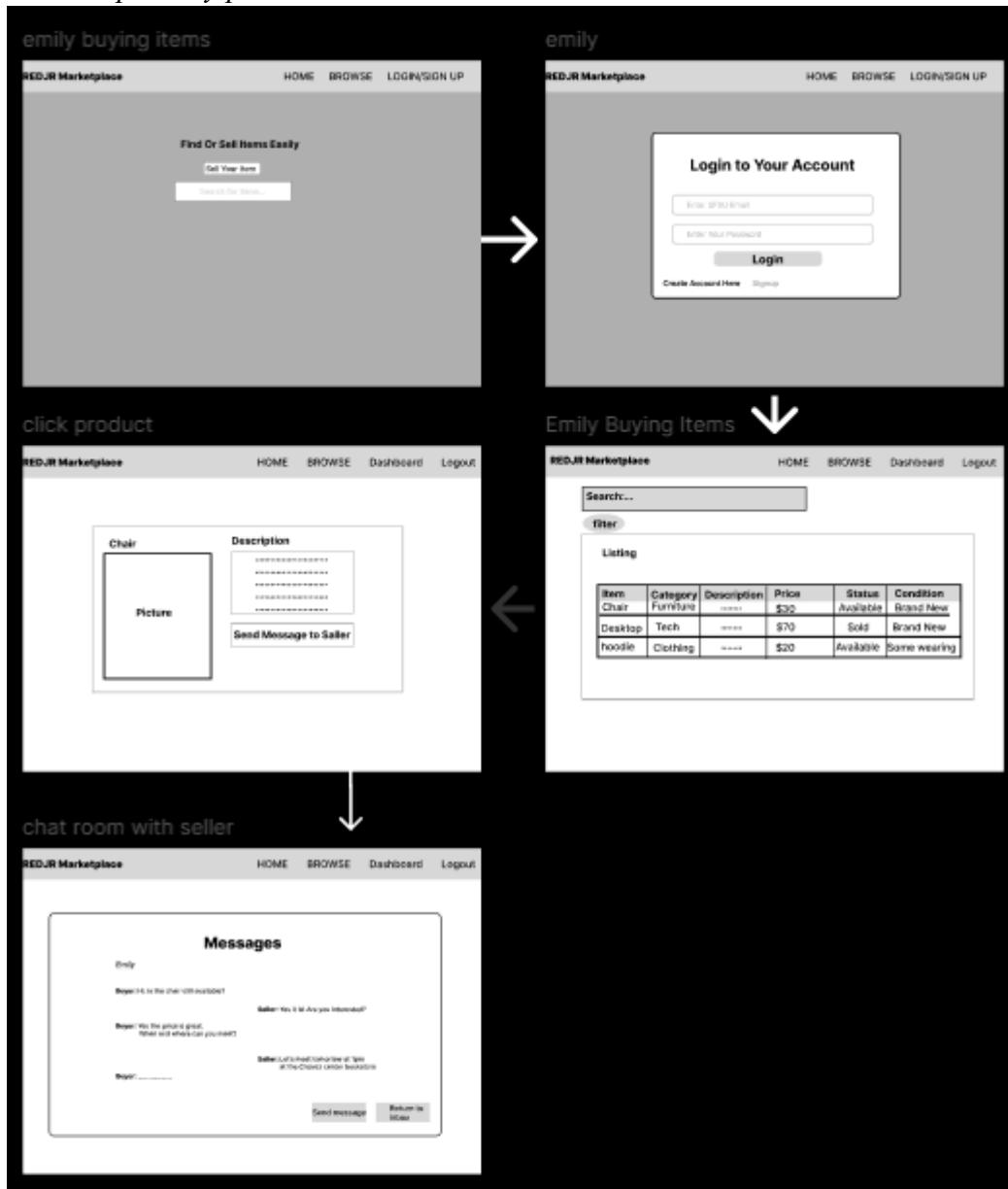


In this mockup, Olivia signs up to REDJR marketplace with her SFSU email (the homepage is the same as griffey mockup) and then she is brought to a page to verify her email. Once she does she officially has an account and access. She then browses for items and clicks on the textbook listing. She messages the seller for availability and then proceeds to make the exchange. After the exchange she leaves a review for the seller.

Emily's Buying Flow

Use Case 3: Buyer logs in, searches, and buys an item.

This mockup is slightly similar to Olivia's but in this one Emily already has an account. Emily accesses the website and then proceeds to login. She then browses for items and sees one to her liking. She then messages the seller and agrees to meet at a given location to view the item and possibly purchase.



05. High Level Architecture Design

DB Organization:

Tables:

1. Categories table

Stores various categories of items available for rent or sale.

Table Name : categories

Columns:

| Column name | Data Type | Description |
|---------------|--------------|--------------------------------|
| category_id | INT | Primary key, auto-increment |
| category_name | VARCHAR(100) | Unique name of the category |

2. Items table

Stores item details uploaded by users.

Table name : items

Columns:

| Column name | Data Type | Description |
|------------------|--|--|
| item_id | INT | Primary key, auto-incremented. |
| title | VARCHAR(255) | Title or name of the item. |
| category_id | INT | Foreign key referencing categories(category_id) |
| item_description | TEXT | Detailed description of the item. |
| item_condition | ENUM('Brand New', 'Used - Like New', 'Used - Good', 'Used - Acceptable') | Condition of the item. |
| price | DECIMAL(10,2) | Price of the item. |
| created_at | TIMESTAMP | Automatically stores the timestamp of when the item was created. |

3. ItemImages Table (Using BLOB Storage)

Stores images of items as binary data (BLOBs).

Table Name : ItemImages

Columns:

| Column name | Data Type | Description |
|-------------|-------------|--|
| image_id | INT | Primary key, auto-increment |
| item_id | INT | Foreign key referencing items(item_id) |
| image_data | LONGBLOB | Stores the thumbnails. |
| mime_type | VARCHAR(50) | MIME type of the image (e.g., 'image/jpeg', 'image/png'). |
| uploaded_at | TIMESTAMP | Automatically stores the timestamp of when the image was uploaded. |

Media Storage :

Choice : Using LONGBLOB for Storing Images

- Images are stored directly in the database as thumbnails using the LONGBLOB data type in the ItemImages table. The original uploaded image is resized on the server before storage to reduce size and improve performance.
- The mime_type column helps ensure proper rendering on the client side. Images are served via a separate /image/<image_id> route with caching to improve load times and reduce database queries.

Why BLOB Storage?

1. **Easier Data Integrity:** Images and item details are stored together, ensuring consistency.
2. **Simpler Backup and Restore:** Everything is stored within the database.
3. **Avoids File Handling Issues:** No need to manage file systems or AWS S3 for image storage.

Search/Filter:

In the navigation bar, there will be a text box for searching items listed on the website. To the left of the search bar will be a drop down menu showing a list of the main item categories on the website.

The main categories are:

- Textbooks/School Supplies

- Electronics
- Furniture
- Clothing
- Miscellaneous

When searching through items by category, the default sorting is by newest listings first. Users can also sort by oldest listing if they want. Sorting items by condition (brand new, like new, some wear, and so on) is also an option, as well as listing items by seller rating from highest rating to lowest. Lastly, a buyer can sort by the asking price of the item, either lowest to highest or highest to lowest.

For searching on keywords, we will compile a list of items first by searching the item table for its title attribute, first selecting all the items that match the search exactly, then we'll query the database from items that match the first three words of search (if applicable), then the first two words, and then the first word. The items will be listed to the user in respective order. For sorting the list by default, users have the same filter features as with doing categories: newest to oldest, oldest to newest, seller rating, price by highest to lowest and vice versa, and condition of the item. When users use filters, the list will no longer be sectioned by closest matching to least, but will be mixed. If the sql query does not return any results, the site will show that no items match the search.

If the user does not type anything in the search box but clicks the search button, the site will list all items sorted according to how recently the item was listed. When filtering for items, if none of the listed items fit the conditions, the website will give a no results message. Users can also filter by if the item has an image or not. This filter will reduce the search based on the other filters that are applied.

Data items used for search:

Item Table

- Title: for keywords search
- Category: For listing items that only belong to a specific category
- Condition: For filtering the search list on condition of item
- Price: For sorting the search item list based on price
- Created_at: From sorting the items from when it was posted

The Seller Table

- Seller Rating: For sorting based on rating score(if seller rating implemented)

The Image Table

- Item_id(FK): used for relating an image with an item so to only show listings that have images

06. Key Risks

Skills risks: Some members of the group have lower experience in software engineering / with the Flask framework for backend. We will mitigate this using the following techniques: Task breakdown to keep things manageable, checkpoints with team lead, reading up on documentation & learning how to use Flask / asking for help from knowledgeable teammates.

07. Project Management

Project management is done with the help of **GitHub projects** linked to the group's main repository. The project is organized so each member has a column of tasks as well as a column for team-wide tasks and tasks that have been completed. Tasks are broken down by Team Lead (Ria) and assigned to the relevant team member's column; once tasks are finished, they are moved to the "Done" column. Code-related tasks are put in as GitHub issues and pushed to the repository. Team Lead checks in at the halfway point of the business week to assess progress and see if there are any blockers/items that team members need support with.

This method was used for Milestones 1 and 2 and will be utilized going forward.

08. Use of GenAI tools

Used ChatGPT in doing research in how %like functions in SQL

- Got an understanding for how it works and asked if I could use %like to find items if the search was slightly off (fuzzy search), and explained that %like is used for pattern matching a sequence of characters so it will not bring up items if search spelled incorrectly.

Question - for % like can I do it where it lists items that match the spelling correctly then list items where the search description is slightly off from item name

ChatGPT response - Great question — you're essentially asking if you can prioritize exact or close matches first, then show partial or fuzzy matches after. With standard SQL **LIKE**, you can only do basic pattern matching, but if you want ranked results based on how closely they match the search term, you've got a few solid options depending on your database system.

- Also ask ChatGPT if possible to stack %like where it gives a list of items that match the search exactly and then items who have the first 3 words, then first 2 words and so on.

Question - Can I do it where it searches the whole phrase then it searches first three words, then first two words, then the first word?

ChatGPT response - Yes, you absolutely can do that! That kind of **progressive fallback search** is a smart way to return the most relevant results first and still show something useful if no exact match is found.

The Strategy:

Assume the user searches for: "**MacBook Pro 13-inch 2020**"

You want to:

1. Try full phrase match first
2. Then match first 3 words: "MacBook Pro 13-inch"
3. Then match first 2 words: "MacBook Pro"
4. Then match just "MacBook"

```

SELECT * FROM items
WHERE item_name LIKE '%MacBook Pro 13-inch 2020%'
    OR item_name LIKE '%MacBook Pro 13-inch%'
    OR item_name LIKE '%MacBook Pro%'
    OR item_name LIKE '%MacBook%'

ORDER BY
CASE
    WHEN item_name LIKE '%MacBook Pro 13-inch 2020%' THEN 1
    WHEN item_name LIKE '%MacBook Pro 13-inch%' THEN 2
    WHEN item_name LIKE '%MacBook Pro%' THEN 3
    WHEN item_name LIKE '%MacBook%' THEN 4
    ELSE 5
END;

```

Analysis - I found it pretty useful and gave me a good understanding of how the %like operator works, and I believe in a quicker time than if I would have

tried to Google or YouTube search about it. A thing I like about ChatGPT is that I could ask follow up questions so I could make sure my understanding of a topic is correct

09. Team Lead Checklist

- So far all team members are fully engaged and attending team sessions when required
 - **ON TRACK**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
 - **ON TRACK**
- Team reviewed suggested resources before drafting Milestone 2
 - **DONE**
- Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission
 - **DONE**
- Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission
 - **DONE**
- Team shared and discussed experience with GenAI tools among themselves
 - **DONE**

Milestone 3 Review

Team number: 07

Meeting date: 04/21/2025

- Summary of feedback on UI (record all pages that need revision)
 - Home page
 - Registration
 - Suggestion for listings page
 - Messages
 - Post listing
- Summary of feedback on code and architecture
 - No major feedback
- Summary of feedback on GitHub usage
 - No major feedback
- Summary of feedback on DB
- Summary of feedback on teamwork and risk management
 - No major feedback
- Confirm that you have done architecture review to check that developers adhere to MVC pattern, coding style, minimal agreed documentation etc. Record if OK or list the issues found. Request developers follow up on corrections and follow up later by doing code reviews
 - OK
- 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 and will be used for grading
 - Unreg user
 - 1. Unregistered users shall only be able to access the register/login/home page/listing page/about pages.
 - 2. Unregistered users can create an account using a verified SFSU email address.
 - 3. A registered/unregistered user shall be able to browse and search for items listed on the website and descriptions of the items.
 - Reg user
 - 3. A registered/unregistered user shall be able to browse and search for items listed on the website and descriptions of the items.
 - 4. A registered user shall be able to upload item listings and descriptions for sale.

- 5. A registered user shall be able to make an offer to purchase an item of their choice.
- 10. A registered user shall be able to message another registered user.
- 11. A registered user shall have a dashboard for their posted items and view messages from prospective buyers.
- 12. Registered users who have agreed to a purchase shall be able to create a sale that contains the meeting location on campus and time.
- Admin
 - 17. A site administrator shall be able to approve all items uploaded to the site before they are available for public view.
 - 19. A site administrator shall be able to delete/ban users that have been reported, violating platform policies.
- Any other comments and issues
 - N/A

CSC648-848

Spring 2025

RedJr: SFSU Marketplace

Ria Thakker (Team Lead, rthakker@sfsu.edu), Rohith Gannoju (Backend Lead), Danny Duenas (Frontend Lead), James Richards (GitHub Master), Eric/Po-Han Chen(Frontend Dev)

Milestone 4, 05/08/2025

History

| | |
|----------------|------------|
| Date Submitted | 05/08/2025 |
| Date Revised | |

1. Product Summary

Product Name: REDJR Marketplace

Description:

In today's fast-paced university environment, students and faculty often struggle to find affordable and convenient ways to buy and sell essential items like textbooks, electronics, and furniture. While general online marketplaces exist, they are not tailored to students and often come with unrelated listings and scams. REDJR is a marketplace platform designed specifically for the SFSU community, offering a trusted space to buy and sell items safely. The platform features verified SFSU-only access, item listings, keyword search, and a messaging system for buyers and sellers. Unique tools like on-campus meetups, a dashboard to manage listings, and optional item rentals help REDJR stand out as a student-first solution.

Final P1 Functional Commitment:

- Unregistered users can browse and search for items on the home, listing, and about pages.
- Unregistered users can register using a verified SFSU email address.
- Registered users can log in and out.
- Registered users can post item listings with images and descriptions.
- Registered users can search for listings using keywords and category.
- Registered users can send offers to sellers via the messaging system.
- Registered users can message and respond to other registered users.
- Registered users can view a dashboard showing posted items, sold items, and saved-for-later items.
- Registered users who agree to a purchase can set up a sale with meeting location and time on campus.
- Registration form includes email validation (must contain "@sfsu.edu"), required field indicators, and terms agreement checkbox.
- A site administrator shall be able to approve all items uploaded to the site before they are available for public view.

Deployment URL: <http://18.191.88.25/>

2. Usability Test Plan

Selected Function: Contact Seller via Message System ("Make Offer")

1. Test Objectives: To evaluate if users can successfully contact sellers and understand the messaging flow. This tests clarity, usability, and task completion.

2. Test Background and Setup:

- **System setup:** REDJR Marketplace deployed on AWS (<http://18.191.88.25/>).
- **Starting point:** Logged-in homepage.
- **Hardware/software:** Laptop/Desktop with Chrome or Firefox.
- **Target users:** SFSU students, 18–30 years old.
- **Environment:** Remote, user tested from home, no camera required, screen sharing optional. No training prior to test.

3. Usability Task Description:

You are browsing REDJR and found a listing titled "Laptop - \$1000". Please contact the seller and send a message offering to meet on campus. Confirm your message was sent.

4. Evaluation of Effectiveness:

Success is determined by whether the user locates the item, opens it, sends a message, and receives confirmation. We'll keep track of task completion.

5. Evaluation of Efficiency:

The time taken from finding the item to confirming the message will be tracked. A duration of under 90 seconds is deemed efficient.

6. Likert Questionnaire (User Satisfaction):

- The message feature was easy to find.(1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly Agree)
- I felt confident sending a message to the seller.(1 = Strongly Disagree → 5 = Strongly Agree)
- I would use this feature again in the future.(1 = Strongly Disagree → 5 = Strongly Agree)

7. GenAI Use: ChatGPT (GPT-4, May 2025)

Use: Drafted usability instructions, suggested Likert questions.

Prompt: "Generate a usability task flow for a marketplace message system."

Utility Rank: HIGH

3. QA Test Plan

1. Test Objectives: Validate that the core messaging feature (Make Offer) works across browsers.

2. HW and SW Setup:

Devices: Windows 11 laptop

Browsers: Chrome 124, Firefox 125

App URL:

3. Feature to be Tested: Message system (sending a message to the seller)

4. QA Test Plan Table:

| Test # | Title | Description | Input | Expected Output | Chrome | Firefox |
|--------|---------------------|---|--------------------------------|---|--------|---------|
| T01 | Valid Message | Send a message | "Hi, is this still available?" | The message appears in the seller's inbox | PASS | PASS |
| T02 | Empty Message Block | Try sending a blank message | "" | Error: message cannot be empty | PASS | PASS |
| T03 | Navigation Test | Navigate back to the item after sending | Click back after sending | Returns to the item listing page | PASS | PASS |

5. Tested on Two Browsers: Results above (Chrome and Firefox)

6. GenAI Use: Tool: ChatGPT (GPT-4)

Use: Helped formulate test cases and expectations

Prompt: "Suggest 3 QA test cases for a marketplace message feature."

Utility Rank: MEDIUM

7. Developer Note:

During QA testing, we discovered a mismatch between the selected category in the search bar and the actual filtered results. For example, selecting "Electronics" showed books, and "Books" returned nothing. After investigation, it was found that the issue stemmed from mislabeled category values in the DEV database. Our team lead (Ria) created a new listing to confirm and submitted a pull request to fix the category ID mapping during item posting. This ensures consistent behavior across both the production and development databases.

4. Peer Code Review

Code Feature: Dashboard interface layout and frontend integration

Review Process:

Po-Han Chen (Eric) implemented the dashboard.html page, which displays selling listings, sold items, and saved-for-later items. This involved integrating dynamic sections with backend API responses and ensuring visual consistency with the rest of the site. James Richards reviewed the code for HTML structure, layout consistency, and route correctness.

He verified that:

- The navbar styling matched other pages (aligned with navbar-search.css)
- The dashboard sections (selling/sold/saved) rendered properly
- The saved items section displayed correctly, even if empty

GenAI (ChatGPT) was also used to assist in code review.

Prompt: "Review a Flask HTML template for a user dashboard with item cards and conditional sections. Suggest improvements for clarity and reusability."

ChatGPT Suggestions:

Use Jinja conditionals to handle empty item sections

Confirm use of consistent class names for CSS reuse

Suggested semantic HTML tags for readability

Submitted:

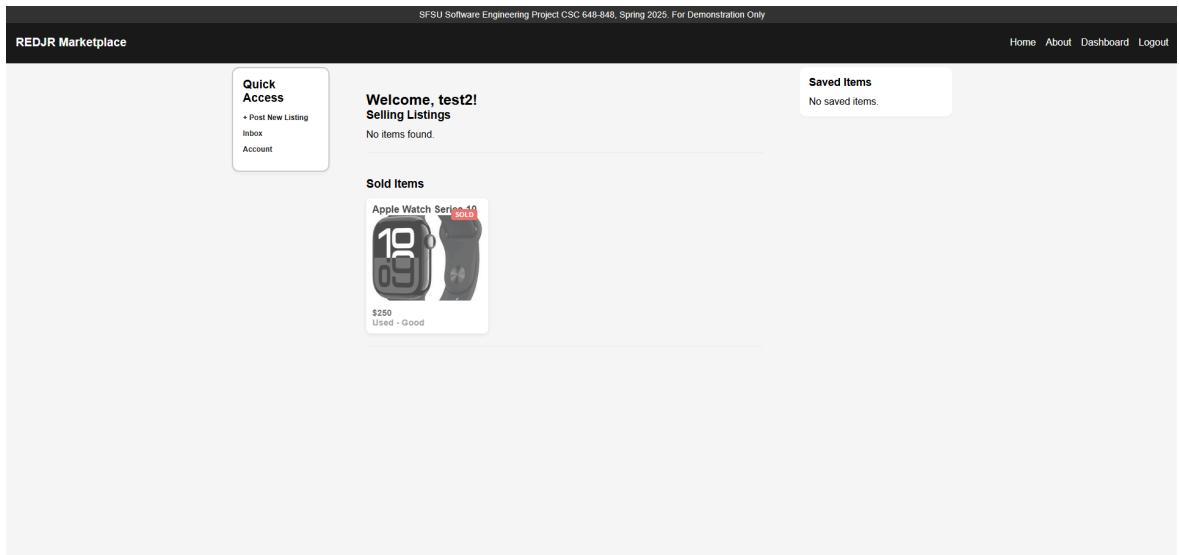
- Screenshot of dashboard HTML layout

```

64  <div class="dashboard-container">
65    <h2>Welcome, {{ username }}!</h2>
66
67    <div class="section">
68      <h3>Selling Listings</h3>
69      <div id="selling-listings" class="card-container">
70        <p>Loading...</p>
71      </div>
72    </div>
73
74    <div class="section">
75      <h3>Sold Items</h3>
76      <div id="sold-items" class="card-container">
77        <p>Loading...</p>
78      </div>
79    </div>
80
81    <div class="section" style="display: none;">
82      <h3>Saved for Later</h3>
83      <div id="saved-items" class="card-container"></div>
84    </div>
85
86    <div class="saved-sidebar">
87      <h3>Saved Items</h3>
88      <div id="saved-sidebar-items" class="scrollable-saved-items"></div>
89    </div>
90
91  </div>

```

- Screenshot of rendered dashboard



- Screenshot of ChatGPT review output

Response:

Section Clarity:

Consider using Jinja `{% if items %}` to conditionally show each section header only when items exist. This will reduce visual clutter for empty sections.

Semantic HTML:

Use `<section>` tags instead of generic `<div>`s for each dashboard block (e.g., Selling Listings, Sold Items) to improve readability and accessibility.

CSS Consistency:

All card containers use the class `card-container`, which is good for reuse. Make sure nested elements like `.item-card` and `.sold-badge` follow consistent class naming.

Responsiveness Suggestion:

Add `flex-wrap: wrap;` to `.card-container` for better responsiveness on narrow screens.

Code Suggestion:

In your JS rendering logic, check for `items.length === 0` explicitly before inserting a `<p>No items found.</p>` message, to avoid false positives.

5. Security Self-Check

| Asset | Threats | Consequences | Mitigation Strategy |
|----------------|------------------------|------------------|--|
| User Passwords | DB leaks, weak hash | Account takeover | Stored as bcrypt hash |
| Listing Data | Unauthorized edit | Wrong info, spam | Access control per user |
| Message | Snooping, spam | Privacy breach | Only visible to involved users, message sanitization |
| Email | Spoofing, fake signups | Fake users | SFSU email domain required |

Validations:

- Passwords encrypted in DB (bcrypt)
- Input fields validated
- Email: must contain “@sfsu.edu”

6. Non-Functional Requirement Self-Check

1. Application shall be developed, tested and deployed using tools and cloud servers approved by Class CTO and as agreed in M0
 - a. DONE
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
 - a. DONE
3. All or selected application functions shall render well on mobile devices (no native app to be developed)
 - a. ON TRACK
4. Posting of sales information and messaging to sellers shall be limited only to SFSU students
 - a. DONE
5. Critical data shall be stored in the database on the team's deployment server.
 - a. DONE
6. No more than 50 concurrent users shall be accessing the application at any time
 - a. ON TRACK
7. Privacy of users shall be protected
 - a. ON TRACK
8. The language used shall be English (no localization needed)
 - a. DONE
9. Application shall be very easy to use and intuitive
 - a. ON TRACK
10. Application shall follow established architecture patterns
 - a. DONE
11. Application code and its repository shall be easy to inspect and maintain
 - a. DONE
12. Google analytics shall be used
 - a. ON TRACK
13. No e-mail clients or chat services 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
 - a. DONE
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
 - a. DONE
15. Site security: basic best practices shall be applied (as covered in the class) for main data items
 - a. DONE
16. Media formats shall be standard as used in the market today
 - a. DONE
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools

a. DONE

18. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only" at the top of the WWW page Nav bar.

a. DONE

7. GenAI Use Summary

Used GPT-4 for:

- Drafting the usability test plan, including task flow and Likert scale wording
- Suggesting QA test case titles, descriptions, and validation logic
- Reviewing dashboard layout code for structure clarity, not design
- Recommending improvements on conditional rendering and mobile responsiveness

Clarification:

The UI layout and structure (HTML/CSS) were designed and implemented by team members.

GenAI was only used to provide feedback or suggestions, which were then reviewed and selectively applied by the team.

Utility Evaluation: MEDIUM–HIGH

Final Note: All GenAI suggestions were human-reviewed and modified as needed.

03. Product Screenshots

The screenshot shows the REDJR Marketplace homepage. At the top, there is a banner with the text "Where SFSU Students Buy, Sell, and Trade" and a "Post Listing" button. Below the banner, there are three promotional boxes:

- Easy to Use**: Post, buy, browse, and message with just a few clicks.
- Made for SFSU**: Student-to-student deals. No outside listings.
- Safe & Secure**: SFSU email required to keep things local and trusted.

The screenshot shows the REDJR Marketplace homepage with the "Recently Posted" section. It displays six items:

| Item | Description | Price |
|------------------------------|------------------------------|-------|
| Wireless Over-Ear Headphones | Wireless Over-Ear Headphones | \$150 |
| Wooden Chair from Dorms | Wooden Chair from Dorms | \$10 |
| TV 55 inch | TV 55 inch | \$699 |
| HP Laptop | HP Laptop | \$500 |
| Bible | Bible | \$2 |
| T-Shirts | T-Shirts | \$20 |

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace

Home About Login/Sign Up

Meet Our Team

| | | |
|---|--|---|
|  Danny Role: Front-End Lead View Profile |  Ria Role: Team Lead View Profile | |
|  James Role: GitHub Master View Profile |  Rohith Role: Backend Lead View Profile |  Eric Role: Front-End Developer View Profile |

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace

Home About Login/Sign Up

Login to Your Account

Enter SFSU Email

Enter Your Password

[Login](#)

Create Account Here [Signup](#)

CSC 648-848 Spring 2025 Sec: X Register | REDJR Marketplace

Not Secure 18.191.88.25/register

Calendar Outlook Gateway - SF State ... Discord 848 Repo CS RL Slack Other Bookmarks

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace

Home About Login/Sign Up

Create an Account

* Full Name

* SFSU Email

* Password
Minimum 12 characters
At least one uppercase letter
At least one number
At least one special character

* Confirm Password

* is required field

Sign Up

CSC 648-848 Spring 2025 Sec: X User Dashboard | REDJR Marketplace

Not Secure 18.191.88.25/dashboard

Calendar Outlook Gateway - SF State ... Discord 848 Repo CS RL Slack Other Bookmarks

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace

Home About Dashboard Logout

+ Post New Listing
Inbox
Account

Welcome, Emily!
Selling Listings

Wireless Over-Ear Headphones

\$150
Brand New
[Edit](#) [Mark as Sold](#)

Sold Items
No items found.

Saved Items
No saved items.

CSC 648-848 Spring 2025 Sec X Post New Listing +

Not Secure 18.191.88.25/new_listing

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace Home About Dashboard Logout

Post a New Listing

Title: *

Category: *

Price: *

Description: *

Condition: *

Upload Image: *

Browse... No file selected.

CSC 648-848 Spring 2025 Sec X Post New Listing +

Not Secure 18.191.88.25/new_listing

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

REDJR Marketplace Home About Dashboard Logout

Furniture

Price: *

Description: *

Condition: *

Upload Image: *

Browse... No file selected.

* Required fields

Post Listing

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

Inbox

Ria Thakker ●
this now works to filter out the message... 5/20/2025, 6:04:58 AM

Griffey ●
Yeah that's cool. TTYL... 5/20/2025, 6:58:37 AM

SFSU Software Engineering Project CSC 648-848, Spring 2025. For Demonstration Only

Messaging Page

← Return to Inbox

Griffey

Hi Griffey, I'm interested in your item: chair listed for \$100.00. Is it still available? 23:08

Hi Griffey, I'm interested in your item: chair listed for \$100.00. Is it still available? 23:10

hey 23:10

lets meet 23:10

haha 23:10

Wooden Chair from Dorms with Griffey

Type your message...

Screenshot of the REDJR Marketplace Search Results page.

Header: REDJR Marketplace, All, Search..., Search

Search Results:

- Category: All
- Total results: 10
- Sort by: Date: Oldest

| Item | Description | Price | Action |
|-----------------------|--|----------|---------------------------|
| Couch | Modern sofa with a dark blue fabric cover. | \$500.00 | View Item |
| Chair | Wooden chair with a simple design. | \$50.00 | View Item |
| Macbook Pro | Laptop with a silver finish. | \$750.00 | View Item |
| Data Science Textbook | Book titled "Introduction to Data Science". | \$20.00 | View Item |
| T-Shirts | A stack of several black t-shirts. | \$20.00 | View Item |
| Bible | Black leather-bound book titled "HOLY BIBLE". | \$2.00 | View Item |
| HP Laptop | Pink laptop with a visible screen. | \$500.00 | View Item |
| TV 55 inch | Flat-screen television displaying a landscape image. | \$699.00 | View Item |

Screenshot of the REDJR Marketplace Edit Listing page.

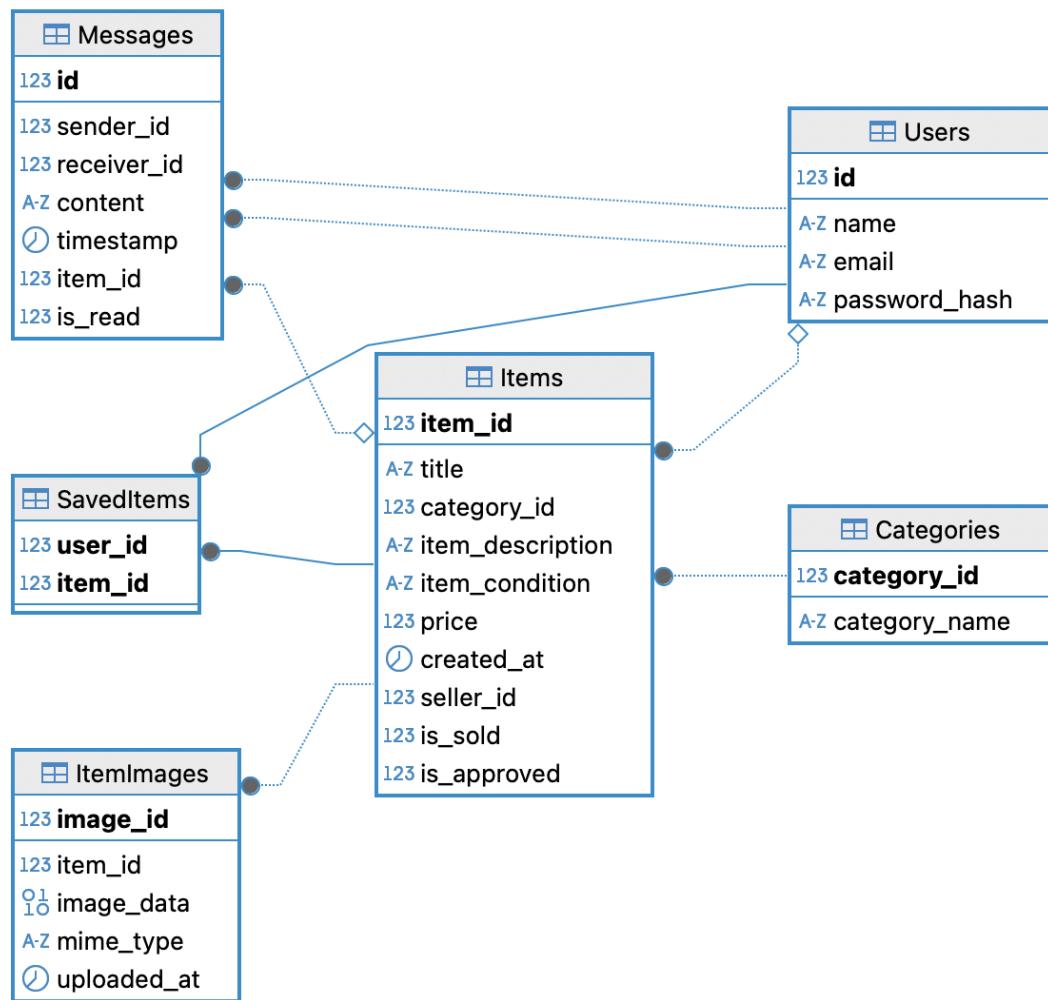
Header: REDJR Marketplace, Home, About, Dashboard, Logout

Section: Edit Your Listing

Form fields:

- Title: Wireless Over-Ear Headphones
- Description: (Empty text area)
- Price (\$): 150.00
- Category: Books
- Update Image (optional): Browse... No file selected.
- Buttons: Update Listing

04. Database Organization



Categories X

Properties Data Diagram

Categories | Enter a SQL expression to filter results (use Ctrl+Space)

| Grid | 123 ↗ category_id | A-Z category_name |
|------|-------------------|-------------------|
| 1 | 4 | Books |
| 2 | 3 | Clothing |
| 3 | 1 | Furniture |
| 4 | 5 | Misc |
| 5 | 2 | Tech |
| | | |
| | | |
| | | |
| | | |

| ItemImages | | | | | | | | | |
|------------|---|---------------|-----------------|---------------|---------------|---------------------|--|--|--|
| | Properties | Data | Diagram | | | | | | |
| | Enter a SQL expression to filter results (use Ctrl+Space) | | | | | | | | |
| Grid | | | | | | | | | |
| | 123 ↗ image_id | 123 ↗ item_id | 01 ↗ image_data | A-Z mime_type | ⌚ uploaded_at | ⋮ | | | |
| 1 | 10 | 7 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-17 23:50:22 | | | |
| 2 | 11 | 8 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-18 00:31:44 | | | |
| 3 | 12 | 9 | ÿØÿà JFIF | ÿÛ C | image/webp | 2025-05-18 02:08:40 | | | |
| 4 | 13 | 10 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-18 02:09:18 | | | |
| 5 | 14 | 11 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-18 02:09:44 | | | |
| 6 | 15 | 12 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-18 03:26:49 | | | |
| 7 | 16 | 13 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-18 03:27:55 | | | |
| 8 | 17 | 14 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-19 19:46:37 | | | |
| 9 | 18 | 15 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-19 21:31:26 | | | |
| 10 | 19 | 16 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-19 21:31:59 | | | |
| 11 | 20 | 17 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-19 23:06:46 | | | |
| 12 | 21 | 18 | ÿØÿà JFIF | ÿÛ C | image/jpeg | 2025-05-19 23:32:12 | | | |
| 13 | 22 | 19 | ÿØÿà JFIF | ÿÛ C | image/webp | 2025-05-20 00:37:02 | | | |
| 14 | 23 | 20 | ÿØÿà JFIF | ÿÛ C | image/webp | 2025-05-20 01:12:01 | | | |
| 15 | 24 | 21 | ÿØÿà JFIF | ÿÛ C | image/webp | 2025-05-20 01:23:12 | | | |

Items

Properties Data Diagram

Enter a SQL expression to filter results (use Ctrl+Space)

| | 123 item_id | A-Z title | 123 category_id | A-Z item_description | A-Z item_condition | 123 price | created_at | 123 | 123 is_sold | 123 is_approved |
|------|-------------|------------------------------|-----------------|--|--------------------|-----------|---------------------|-----|-------------|-----------------|
| Grid | 1 | Couch | 1 | Here's a sample description. Brand New | Brand New | 500 | 2025-05-17 23:50:22 | 1 | 0 | 1 |
| Text | 2 | Wooden Bedframe | 1 | Graduating soon and will not | Used - Like New | 150 | 2025-05-18 00:31:44 | 2 | 1 | 1 |
| Text | 3 | Chair | 1 | New description 2. Edited 05 | Used - Like New | 50 | 2025-05-18 02:08:40 | 1 | 0 | 1 |
| Text | 4 | Macbook Pro | 2 | New description to test editin | Used - Good | 750 | 2025-05-18 02:09:18 | 1 | 0 | 1 |
| Text | 5 | Data Science Textbook | 4 | Updating the description on C | Used - Acceptable | 20 | 2025-05-18 02:09:44 | 1 | 0 | 1 |
| Text | 6 | T-Shirts | 3 | Soft, breathable cotton | Brand New | 20 | 2025-05-18 03:26:49 | 3 | 0 | 1 |
| Text | 7 | T-Shirts | 3 | Soft, breathable cotton | Brand New | 20 | 2025-05-18 03:27:55 | 3 | 0 | 0 |
| Text | 8 | Orange Bike | 5 | Orange Bike | Used - Like New | 150 | 2025-05-19 19:46:37 | 4 | 1 | 1 |
| Text | 9 | Bible | 4 | Brand new bible for cheap. | Brand New | 2 | 2025-05-19 21:31:26 | 7 | 0 | 1 |
| Text | 10 | HP Laptop | 2 | Lightly used HP laptop, suita | Used - Like New | 500 | 2025-05-19 21:31:59 | 7 | 0 | 1 |
| Text | 11 | chair | 1 | wooden chair | Used - Good | 100 | 2025-05-19 23:06:46 | 5 | 1 | 1 |
| Text | 12 | TV 55 inch | 2 | Nice TV for netflix and chill | Used - Like New | 699 | 2025-05-19 23:32:12 | 5 | 0 | 1 |
| Text | 13 | Wooden Chair from Dorms | 1 | These are the wooden rockin | Brand New | 10 | 2025-05-20 00:37:02 | 5 | 0 | 1 |
| Text | 14 | Wireless Over-Ear Headphones | 2 | Brand new headphones. | Brand New | 150 | 2025-05-20 01:12:01 | 6 | 0 | 1 |
| Text | 15 | sfsu shirt | 3 | sfu shirt gray | Used - Good | 15 | 2025-05-20 01:23:12 | 5 | 1 | 1 |

Messages

Properties Data Diagram

Enter a SQL expression to filter results (use Ctrl+Space)

| | 123 id | 123 sender_id | 123 receiver_id | A-Z content | timestamp | 123 item_id | 123 is_read |
|------|--------|---------------|-----------------|--|---------------------|-------------|-------------|
| Grid | 1 | 2 | 1 | Hi Ria Thakker, I'm interested in your item: Couch listed for \$500.00. Is it still available? | 2025-05-18 00:40:28 | 7 | 0 |
| Text | 2 | 2 | 1 | hello? | 2025-05-18 00:51:51 | 7 | 0 |
| Text | 3 | 3 | 4 | Hi James, I'm interested in your item: Orange Bike listed for \$150.00. Is it still availab | 2025-05-19 19:47:15 | 14 | 0 |
| Text | 4 | 4 | 6 | Hi Ria Thakker, I'm interested in your item: Data Science Textbook listed for \$20.00. I | 2025-05-19 21:47:41 | 11 | 0 |
| Text | 5 | 5 | 1 | Hi! Yes, it's still available. | 2025-05-19 21:48:01 | 11 | 0 |
| Text | 6 | 6 | 1 | I'm willing to pay \$15 for it, are you okay with that? | 2025-05-19 21:48:44 | 11 | 0 |
| Text | 7 | 7 | 6 | Double texting for funsies. | 2025-05-19 21:48:49 | 11 | 0 |
| Text | 8 | 8 | 6 | Hey Ria! U there? | 2025-05-19 23:04:28 | 11 | 0 |
| Text | 9 | 9 | 6 | Hi Griffey, I'm interested in your item: chair listed for \$100.00. Is it still available? | 2025-05-19 23:08:30 | 17 | 0 |
| Text | 10 | 10 | 6 | Hi Griffey, I'm interested in your item: chair listed for \$100.00. Is it still available? | 2025-05-19 23:10:13 | 17 | 0 |
| Text | 11 | 11 | 5 | hey | 2025-05-19 23:10:34 | 17 | 0 |
| Text | 12 | 12 | 5 | lets meet | 2025-05-19 23:10:46 | 17 | 0 |
| Text | 13 | 13 | 5 | hey yo | 2025-05-19 23:33:31 | 17 | 0 |
| Text | 14 | 14 | 5 | Hi Emily, I'm interested in your item: Wireless Over-Ear Headphones listed for \$150.0 | 2025-05-20 01:24:30 | 20 | 0 |
| Text | 15 | 15 | 6 | hey yo | 2025-05-20 01:24:42 | 20 | 0 |
| Text | 16 | 16 | 1 | hi! yes I am available | 2025-05-20 02:13:18 | 11 | 0 |
| Text | 17 | 17 | 6 | sweet, let me know what time you can meet | 2025-05-20 02:14:50 | 11 | 0 |
| Text | 18 | 18 | 6 | hi | 2025-05-20 02:30:11 | 11 | 0 |
| Text | 19 | 19 | 1 | hi | 2025-05-20 02:30:20 | 11 | 0 |
| Text | 20 | 20 | 6 | trying again | 2025-05-20 02:31:35 | 11 | 0 |
| Text | 21 | 21 | 6 | hi | 2025-05-20 05:25:51 | 11 | 0 |
| Text | 22 | 22 | 6 | hi | 2025-05-20 05:26:22 | 11 | 0 |
| Text | 23 | 23 | 6 | test 2 | 2025-05-20 05:30:32 | 11 | 0 |

SavedItems

Properties Data Diagram

Enter a SQL expression to filter

Grid

| | 123 user_id | 123 item_id |
|---|-------------|-------------|
| 1 | 2 | 7 |
| 2 | 2 | 8 |
| 3 | 2 | 9 |
| 4 | 2 | 10 |
| 5 | 5 | 20 |

Text

Users

Properties Data Diagram

Enter a SQL expression to filter results (use Ctrl+Space)

Grid

| | 123 id | A-Z name | A-Z email | A-Z password_hash |
|---|--------|--------------|--------------------|---|
| 1 | 1 | Ria Thakker | rthakker@sfsu.edu | pbkdf2:sha256:100000\$HtdqwShwc39KqkJ3\$b3d0153ff686ff58957c6f9ccb795d71a29b70fff7eea4695b02ddcb507b69d1 |
| 2 | 2 | Danny Duenas | dduenas@sfsu.edu | pbkdf2:sha256:100000\$KP4YMsWnWD53Qz9r\$477c58df2d3c42f07e3f114a1dacb5bed7782ec0d2dcf2c2d7ec26b96b107d3f |
| 3 | 3 | Eric | pchen@sfsu.edu | pbkdf2:sha256:100000\$CGSGC9IEjRY0J2K\$a5a0598bc296ac73d87c49273bcdcbef2a205b08bf67bb448c1c1b49e80e4d82 |
| 4 | 4 | James | jrichards@sfsu.edu | pbkdf2:sha256:100000\$cvWJBZQda2X8Fson1\$42f58ae6d0c5b17c827bd383023ab9d44ca799b17308651b772f2d3b7e9e9af8 |
| 5 | 5 | Griffey | griffey@sfsu.edu | pbkdf2:sha256:100000\$Mm9SJXmg2dTJKyp\$83407876849d33f269a1511e89667b0c2d49e722f316a495500f284412168e83 |
| 6 | 6 | Emily | emily@sfsu.edu | pbkdf2:sha256:100000\$shICpgGQ5krbvyw\$e21d967de45c710893f3db49c10f1d48f00a8c5d50219b59ea6fc044b73792c |
| 7 | 7 | Jake | jake@sfsu.edu | pbkdf2:sha256:100000\$PoSadFgmWjzpirMT\$53ac0142bb5338565690a79d156f8a9d71de43b38e809249bcd2c963de8f26d1 |

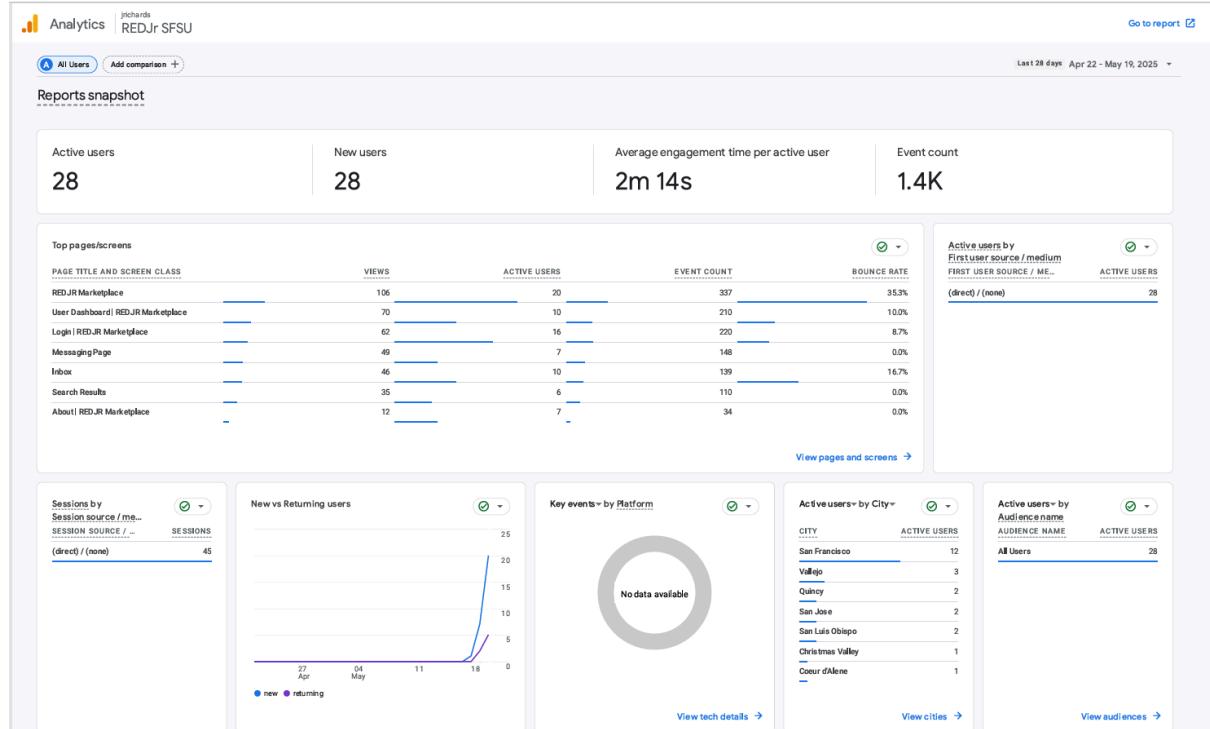
Text

05. GitHub Organization

Main branches in team github:

1. Main branch: No one has direct access. All pushes to main must be approved by another teammate. Our general practice is to only merge to main from the staging branch.
2. Staging branch: Testing branch. No one has direct access; all pushes must be approved. We push new features to staging, code review, test, then merge to main.
3. Feature branches: Each individual teammate makes their own feature branches for specific items they are working on or specific bug fixes that are then deleted upon successful merge.

06. Google Analytics



07. Project Management

Used GitHub projects for task management. Separate sections for all team members and columns for tasks completed together as a team and for finished tasks. Team lead archives tasks that are finished at the end of each milestone.

Screenshot of a GitHub Project Backlog board for CSC648-0104-team07.

The board has four columns:

- Done**: 8 items, Estimate: 0. Sub-tasks include: Login, register, user verification; Post item listings; Messaging; User dashboard; Implement M3 UI feedback.
- Team tasks**: 0 items, Estimate: 0.
- Ria**: 0 items, Estimate: 0.
- Rohith**: 0 items, Estimate: 0.

Buttons at the bottom of each column: + Add item.

Screenshot of a GitHub Project Backlog board for CSC648-0104-team07.

The board has four columns:

- Rohith**: 0 items, Estimate: 0.
- James**: 0 items, Estimate: 0.
- Danny**: 3 items, Estimate: 0. Sub-tasks include: Draft M4 Document + Tasks; Draft Integrate with Backend as it's developed; Draft Mobile responsiveness.
- Eric**: 3 items, Estimate: 0. Sub-tasks include: Draft M4 Document + Tasks; Draft Integrate with backend as it's developed; Draft Mobile responsiveness.

Buttons at the bottom of each column: + Add item.

08. Team Member Self-Assessment & Contributions

From: rthakker@sfsu.edu

To: pchen@sfsu.edu, jrichards@sfsu.edu, dduenas@sfsu.edu, rgannoju@sfsu.edu

CSC 648 Milestone 5 – Team Member Self-Assessment

Hey all,

Adding my self-assessment.

1. Role & Contributions:

- a. Team Lead: Task division & management, team direction, function prioritization.
 - i. Contributed to the backend
 1. Created dashboard API
 2. Modularized code on the backend
 3. Real-time messaging
 4. Bug fixes post-testing
 5. Milestone Documents: 1, 2, 5
 - ii. Led team meetings, facilitated weekly updates and check-ins with team members at the halfway point to all tasks
- b. Submissions to GitHub: 25
- c. Main challenges during team project: Balancing with my other commitments, like clubs and work. I also had to work in Flask for the first time, which was a new technology for me. Had to put in a bit of work to understand the backend for this reason, but I got the hang of it eventually 😊
- d. GenAI experience: Very useful for writing applications, slightly less so for coding. Have to double-check all outputs, but it requires less editing when writing the executive summary for example. It takes a bit of fiddling with when coding as GenAI often loses context of the specifics of your implementation and defaults to the algorithms or implementations it knows best.
- e. What I'd do better next time: Be more specific with my task breakdowns and keep abreast of requirements more thoroughly.
- f. Overall Feedback from Team Lead:
 - i. Very smooth experience overall working with this group. No issues with communication or getting team members to stay on task. All members kept well on track and stayed within their deadlines. The main challenge was making sure we were able to manage design/architectural problems effectively, which we managed via team discussions of the pros/cons/tradeoffs of particular solutions.

- ii. GenAI seemed most effective on the milestone documents and for small coding tasks for the group overall. As team lead, I most often used it for assistance with the milestone documents.
- iii. Overall, SE management and processes were pretty effective in our group. What I would do better next time is probably introduce more thorough code review as a few bugs did make it through to staging. Would also probably implement the QA process earlier to get more feedback on the frontend.

Thanks for an amazing group experience. Couldn't have asked for a better team. Enjoy your summer!

Best,

Ria Thakker

Team Lead

From: pchen@sfsu.edu

To: rthakker@sfsu.edu, jrichards@sfsu.edu, dduenas@sfsu.edu, rgannoju@sfsu.edu

CSC 648 Milestone 5 – Team Member Self-Assessment

Hi team,

Here's my self-assessment for Milestone 5 and the project overall.

- Role & Contributions:
- Frontend developer
- Created the initial UI design prototype in Figma during early milestones
- Contributed to the Milestone 2 revision and Milestone 4 documentation content
- Built the dashboard.html with selling, sold, and saved item sections
- Created edit_item.html and edit_item.css pages and connected them from the dashboard edit button
- Implemented consistent navbar-search layout across pages using navbar-search.css
- Designed and styled the login, register, and email verification pages.
- Fixed layout alignment across multiple pages
- Assisted with UI responsiveness and page structure debugging
- Collaborated via Discord and GitHub, with active communication and timely updates
- GitHub activity:

I made 30 commits primarily to the staging-branch, focusing on frontend components. Some of the UI elements I developed, such as page layout

structure, form components, and styling, were integrated or reused by teammates, so not all my contributions may be fully visible under my name in the commit history.

- Biggest challenge:

Maintaining layout consistency across pages while syncing dynamic frontend behavior with backend data and session logic.

- GenAI usage:

Used ChatGPT for layout guidance, form structure, CSS adjustments, and dashboard component logic. It helped streamline the workflow and gave clarity when troubleshooting UI bugs.

- If I were to do this again:

I would establish a standardized set of UI components early on, similar to a component library, and work in tandem with the backend development from the outset to minimize integration issues.

- Other notes:

My main focus was on UI/UX and frontend functionality, so I didn't contribute to deployment or backend service logic. That was handled by others.

Thanks again for all the collaboration!

Po-Han Chen

From: rgannoju@sfsu.edu

To: rthakker@sfsu.edu, jrichards@sfsu.edu, dduenas@sfsu.edu, pchen@sfsu.edu

CSC 648 Milestone 5 – Team Member Self-Assessment

Hello Team,

I'm writing to share my self-assessment for our team project this semester.

- **Role & Contributions**

As the Backend Lead, I was primarily responsible for:

- Designing the backend architecture
- Writing and maintaining REST APIs
- Deploying the project to the server

- Managing the database: creating and altering tables to support evolving features
- Maintaining API documentation for seamless collaboration with the frontend team

Feature-wise, I implemented chat functionality, register user, item posting, home page backend logic, and handled bug fixes throughout development.

b) GitHub Contributions

Total commits: 16

c) Challenges Faced

Fortunately, I did not face major obstacles. The project proceeded smoothly as planned. Although Flask was new to me at the start, I quickly got comfortable with it and enjoyed working with the framework.

d) GenAI Experience

- I used ChatGPT-4o for documenting APIs efficiently, which streamlined communication with the frontend team.
- The ChatGPT O4 mini-high model was useful in generating boilerplate for test classes, saving time and ensuring consistency.

e) What I'd Do Better Next Time

From a technical perspective, I'd aim to:

- Implement a CI/CD pipeline (e.g., with Jenkins) for automatic deployment
- Enhance API security with proper authorization layers and input validation

From a team management and workflow standpoint, everything worked very well this time.

f) Overall Feedback (As Backend Lead)

1. I had the pleasure of working with an incredibly reliable and respectful team — **@James Edward Richards**, **@Danny Duenas**, and **@Po-Han Chen** were all highly engaged throughout the semester. Everyone met their deadlines, followed clean code practices, and actively contributed to team discussions.
2. Our team lead, **@Ria Bharat Thakker**, was outstanding. Despite juggling on-campus work, she managed milestone coordination, frontend and backend

contributions, and even led the final presentation solo. Her leadership was the backbone of our project's success.

3. This project was a truly enjoyable experience. Everything was well-coordinated, and I gained a lot not just technically, but also from working alongside such talented and collaborative teammates.

Thanks for a great semester. Wishing everyone a fantastic summer!

Best regards,

Rohith Gannoju

Backend Lead

From: jrichards@sfsu.edu

To: rthakker@sfsu.edu, dduenas@sfsu.edu, pchen@sfsu.edu, rgannoju@sfsu.edu

CSC 648 Milestone 5 – Team Member Self-Assessment

Hello Team,

Here is my self assessment

- **Role & Contributions**

For this team I had the role of GitHub Master and backend team member

- Creating the Project Repo and inviting all team members to it.
- Work on the strategy for of team repo workflow when it comes to branching and merging
- Handled any merge conflicts that did arrive in order to properly integrate all team members code
- Would maintain clean up the repository by deleting any branches that were outdated or unnecessary
- developed backend login functionality in order for front end member to work with
- Created and implemented the search functionality for our website which included being able to search by name and category.
- Added sorting functionality for listed items with options for date (newest/oldest) and price (high/low).

b) GitHub Contributions

Total commits: 11

c) Challenges Faced

Biggest Challenge would be learning Flask, and MySQL as before the semester started I was unfamiliar with both.

d) GenAI Experience

- I would use ChatGPT as debugging tool for when any errors happened. Also I would use it somewhat like a tutor I would explain my plan in implementing some functionality and asking if this would achieve my intended goal

e) What I'd Do Better Next Time

Help more when it comes to managing the server and deploying the website.

f) Overall feedback from GitHub Master.

This project went by pretty smoothly. Never felt we were behind on anything, and I felt everyone had a good understanding of their roles and responsibilities and I would credit that to good communication. Because that there were very few merge conflicts and the ones that happened were easily fixable

Thanks everyone for all your hard work this semester!

James

From: dduenas@sfsu.edu

To: rthakker@sfsu.edu, jrichards@sfsu.edu, pchen@sfsu.edu, rgannoju@sfsu.edu

CSC 648 Milestone 5 – Team Member Self-Assessment

Hello Team,

Here is my self assessment

a) Role and Contributions

As frontend lead I...

- Designed and implemented the frontend structure using HTML, CSS, and JavaScript
- Developed and styled all major pages including
- Ensured mobile responsiveness across all pages using media queries
- Created dynamic components like the navbar, side drawer menu, and card-based layouts
- Coordinated with backend developers to integrate frontend with Flask API endpoints
- Contributed to overall UI/UX consistency and tested styling across devices

b) GitHub Contributions

- Total submissions: 17 commits

c) Main Challenges Encountered

One of the biggest challenges was making sure that the frontend layout remained consistent across different screen sizes and that it adapted cleanly on mobile devices. Balancing design preferences across team members and merging frontend updates with backend logic without conflicts was also challenging. Another challenge was balancing out other projects and classes.

d) Use of GenAI

I used GenAI (ChatGPT 4.0) with CSS styling for the overall appearance and layout, but it couldn't really design my code how I had pictured in my mind so I took over from there. Frontend wise it pretty much did the lazy work for me and provided a great structure.

e) What I Would Do Better Next Time

Next time, I would propose using a shared Figma design prototype from the start to avoid inconsistent layout assumptions. That would have been crucial because I was just spitballing and deleting multiple styles and layouts. A UX designer would have been very handy. I would also communicate with my team more and not try to take on more than I can chew, because there were a ton of late nights.

f) Overall Feedback (As Frontend Lead)

Honestly this project went pretty smoothly. I was blessed to be placed in such an amazing group. We legit had no problem communicating and everything flowed very easily. We had each other's backs when extra help was needed. Our professor even called us boring due to the fact that there was not much help to give us. Everyone took their role seriously and we brought the heat.