Final Project for SW Engineering CSC 648-848 Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

http://ec2-34-228-231-71.compute-1.amazonaws.com/

05/22/2024

1. Product Summary

Name of the Product: SwiftSell

List of Features

1. An unregistered User shall be able to search for specific items.
2. An unregistered User shall be able to browse the website's items
3. An unregistered User shall be able to see images for items/services with accurate description.
4. An unregistered user shall be able to register.
5. A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale.
6. A Registered User shall be able to post items and/or services(tutoring,delivery, etc) for sale with approval of admin.
7. A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item.
8. A Registered User shall be able to message sellers about items for sale.
9. Only Registered Users shall be able to message other Registered Users.
10. A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc.
11. A Registered User shall be informed if another user wants one of their items via in-site messaging.
12. A registered User shall be able to login.
13. A registered user shall have the option to delete an item.
14. Verified users shall have profiles via SFSU email authentication.
15. A user shall be able to search items by course number.
16. An Admin shall be required to approve or deny user's posts to go online

Unique Features:

1. Exclusively tailored to the SFSU community with verification through SFSU email addresses.
2. Local listings and **job services** specifically for students, faculty, and staff members.
3. Enhanced search capabilities for finding class-specific textbooks and other items.

http://ec2-34-228-231-71.compute-1.amazonaws.com/

2. Milestone documents – M1-M4

SW Engineering CSC648-848

Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

Milestone 1

Submitted	Revised
March 8th 2024	March 9th 2024

Executive Summary

SwiftSell is a marketplace platform aimed at revolutionizing the way San Francisco State University (SFSU) students, faculty, and staff members buy and sell goods within the university community such as students, faculty and staff at SFSU . Our project is motivated by the need to create a safe, convenient, and student-friendly marketplace that fills a gap in the market that provides unique SFSU functions specially designed for the SFSU community. By providing a platform specifically tailored to the needs of college students, faculty and staff members, SwiftSell seeks to enhance the overall student experience at SFSU.

The motivation behind SwiftSell stems from the recognition of the challenges students , faculty and staff members at SFSU face when buying and selling goods in a university setting. Current platforms lack the safety, convenience, and campus-specific features that are essential for a thriving marketplace within a college community. By developing SwiftSell, we aim to provide SFSU students with a reliable, secure, and user-friendly platform for buying and selling goods, ultimately enriching campus life and fostering a sense of community.

SwiftSell will offer a range of functions and services tailored to the needs of the SFSU community such as students, faculty and staff at SFSU. These include verified user profiles with the use of SFSU emails, local listings convenient for the target clientele to complete their purchases, search items such as textbooks by classes, as well as a dashboard for users to be able to view messages from prospective buyers and sellers. The platform will also feature search categories, including textbooks searchable by class, furniture, and electronics. The platform will also provide the user with the ability to list and accept job listings such as assisting in moving furniture sold on our marketplace and listing tutoring services for university classes. By providing a market focused on student, staff, and faculty needs, SwiftSell aims to simplify the buying and selling process for users, making it easier to find and sell items within the SFSU community.

Our team is composed of driven and innovative students from diverse backgrounds, all united by a shared vision of enhancing the student experience at SFSU. With an emphasis on the user experience, we aim to create an easy to use and attractive marketplace for users to interact with. With expertise in Computer Science and the guidance of our CEO Dragutin Petkovic and CTO Anthony Souza, we are equipped to develop and launch a successful marketplace platform. Together, we are committed to making SwiftSell a valuable resource for members of SFSU and a thriving marketplace within the college community.

Personae

Student: John - Seller and full-time student

"Generated by OpenAI's DALL-E model (OpenAI, 2024)”

About John	Goals and Scenarios
• Meet John, a 21-year-old Computer Science major at San Francisco State University (SFSU). • John is a tech-savvy student enthusiastic about using an online platform for efficient selling.	• John aims to be time efficient and make extra money through the selling process. • He seeks a user-friendly platform that simplifies selling, maximizes item visibility, and ensures safety.

	• The website should cater to John's preferences, providing a seamless and secure experience for SFSU students engaged in buying and selling goods. • Limited time and security concerns are constraints for John in the selling process.

Student: Sarah - Buyer

"Generated by OpenAI's DALL-E model (OpenAI, 2024)”

About Sarah	Goals and Scenarios
● Sarah is a 20-year-old Psychology major at San Francisco State University (SFSU). ● Sarah is in her second year and actively engaged in campus events and social activities.	● Sarah aims to find good deals on essential items like textbooks, furniture, and electronics. ● Sarah thinks it would be awesome to also request services such as tutoring, help with moving out furniture, etc.

• As an eager student buyer, Sarah values convenience, affordability when searching for items and hates wasting time. • Sarah is not as tech-savvy as some of her peers but is open to exploring user-friendly platforms for buying used items. • Limited budget and time constraints due to academics and part-time work are her challenges.	• The website should cater to Sarah's need for an easy-to-navigate platform. • Clear item descriptions and secure transactions are essential for enhancing her overall buying experience as an SFSU student.

Staff/faculty: James

"Generated by OpenAI's DALL-E model (OpenAI, 2024)”

About James	Goals and Scenarios
● James is a 47-year-old staff member at San Francisco State University (SFSU) working in the administrative office. ● James holds a position in student services, assisting with various administrative tasks related to student affairs. ● He is responsible for coordinating events and often helps students navigate administrative processes. ● James, has vision problems and gets confused when there is too much information at once ● Time is a precious resource for him, balancing work responsibilities and personal commitments.	● James seeks a user-friendly platform that facilitates smooth communication between buyers and sellers. ● The website should cater to James's need for simplicity, providing quick access to information. ● James has an sfsu email and would like to engage in either selling or buying products on the website

Admin: Alex

"Generated by OpenAI's DALL-E model (OpenAI, 2024)”

About Alex	Goals and Scenario
● Introducing Alex, the 21-year-old administrator overseeing the online marketplace at San Francisco State University (SFSU). ● With a background in information technology and a wealth of experience in system administration, Alex plays a crucial role in maintaining platform integrity and security.	● Alex seeks a platform that allows seamless oversight, quick problem resolution, and adherence to data security protocols. ● The website should provide a user-friendly administrative interface. ● This interface should enable Alex to efficiently manage the platform while ensuring a secure

● Detail-oriented and security-conscious, Alex ensures that the website complies with university policies and standards. ● As an administrator, Alex values robust features for user management, transaction monitoring, and issue resolution. ● With a busy schedule, time efficiency is a priority for Alex.	and reliable experience for all users in the SFSU community.

High-level Use Cases

1. Search Browse and Message

   Users, like John, an SFSU student, have the ability to effortlessly browse or search for items on the SFSU marketplace (SwiftSell). For instance, John is looking for a specific textbook for his computer science class. He navigates to the website, enters the book title, and reviews the search results. Once he finds the desired item, he wants to be able to initiate a conversation with the seller directly on the platform to inquire about the book's condition, price, and potential meeting arrangements. The goal is to facilitate seamless communication between buyers and sellers within the SFSU community, enhancing the overall user experience. Upon sending the message he composed, he is prompted to login or register.

2. Posting Content

   Users, such as Sarah (student at SFSU), have the ability to easily post content for sale on the SFSU marketplace. Sarah has a set of furniture items she wants to sell before moving off-campus. Sarah will be prompted to either log in or register once she wants to post an item. She logs into the platform to sell her item and proceeds by creating a new listing.

   Sarah uploads clear images, provides a concise description, and a price for each item. The system allows her to efficiently post multiple items for sale within a short timeframe. The objective is to empower sellers like Sarah to showcase their items effectively, promoting a diverse range of listings on the platform.

   Sarah might also want to request a service, she can use our platform to request a service to be executed, she can write a description of what the service entails and the price she is willing to pay for such service.

3. Dashboard View What is For Sale:

   Any user of the website such as students, faculty and staff members at SFSU have a dashboard view. James for example is a staff member, in his late forties, who benefits from a user-friendly SFSU marketplace website designed for easy interaction. Featuring good color contrast and appropriately-sized text. James, like all users of SwiftSell have access to the dashboard view.

   James accesses a comprehensive dashboard upon login. The dashboard categorizes listings, including textbooks, electronics, furniture, and more, facilitating quick navigation, viewing of trending items, and application of filters for specific criteria.

   James logs in and is greeted with a dashboard displaying categorized listings, including textbooks, electronics, furniture, and more. The dashboard allows him to quickly navigate through sections of interest, view trending items, and access filters for specific criteria.

4. Admin Approval:

Administrators, like Alex, have the authority and obligation to review and approve listings before they go live on the SFSU marketplace. For example, a user submits a listing for a high-value electronic item. Alex receives a notification and accesses the administrative dashboard, where pending listings are displayed.

Alex reviews the listing for compliance with university policies, ensuring it aligns with the platform's guidelines. If everything meets the criteria, Alex approves the listing, making it visible to users.

The focus here is on empowering administrators like Alex to control and curate the content that goes live on the platform.

Main Data Items and Entities

Items for Sale: Objects available for sale by registered users.
Associated data fields include:

- Category (e.g., Furniture, Electronics, Textbooks)
- Description (e.g., description of item)
- Price (price of item)
- Picture (photo of item)
- Thumbnail (generated from the user-supplied image)
- Item id

Services for Sale: Services available for sale by registered users.
Associated data fields include:

- Category (e.g., Tutoring, Moving)
- Description (e.g., tutoring subject and hours)
- Price (hourly rate for services)
- Picture (image of flyer for services)
- Thumbnail (generated from the user-supplied image)
- Item id

Dashboard: A screen view within the website, available to registered users, which displays messages sent to the user but also displays the user’s posted items.

Messages: Communication between two registered users on the website.
Associated data fields include:

- Sender
- Recipient
- Content of message
- Timestamp of when it was sent
- Unread/read by the recipient
- Message id

Register/Registration: The process whereby an unregistered user becomes a registered user by creating a username and password.

User: Visitors to the web application are part of one of three groups: unregistered users, registered users, and admin.
Associated data fields include:

- Email address (sfsu.edu)
- Password
- Date joined
- User id

Unregistered User: An unregistered user is not allowed to sign into the website. The unregistered user is not allowed to login, post, or message, and cannot view the dashboard; the unregistered user is allowed to view items for sale, to fill out the form to post an item or send an item (but not submit the form), and to register.

Registered User: In addition to the actions allowed to an unregistered user, a registered user is allowed to sign into the website. Once signed in, a registered user is allowed to post items for sale (with admin moderation), message other registered users, and view the dashboard.

Admin: In addition to the actions allowed to a registered user, the admin shall have the responsibility to approve posts for sale that conform to terms of service.

Functional Requirements

5.**List high level functional requirements**

1. **Unregistered User**
    1.1. An unregistered User shall be able to search for specific items. - Priority 1

1.2. An unregistered User shall be able to browse the website's items - Priority 1
1.3. An unregistered User shall be able to categorize items for sale by area of use (electronics, textbooks, furniture, etc) - Priority 2
1.4. An unregistered User shall be able to sort items for sale by majors. - priority 2
1.5. An unregistered User shall be able to see images for items/services with accurate description - priority 1
1.6. An unregistered User shall be able to filter through items based on price, date posted, etc. - priority 2
1.7 An unregistered user shall be able to register - priority 1

2. Registered Users shall have ability to do all things above plus…

2.1. A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale. - priority 1
2.2. A Registered User shall be able to post items and/or services(tutoring,delivery, etc) for sale with approval of admin - priority 1
2.3. A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item. - priority 1
2.4. A Registered User shall be able to delete a post without the approval of an admin. - priority 3
2.5. A Registered User shall be able to message sellers about items for sale. - priority 1
2.6. Only Registered Users shall be able to message other Registered Users. - priority 1
2.7. A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc. - priority 1
2.8. A Registered User's post may take up to 24 hours to be approved by the admin. - priority 2
2.9. A Registered User shall be informed if another user wants one of their items via in-site messaging. Priority 1
2.10. A registered user shall be able to remove items for sale once a transaction is complete via in-site messaging. - priority 3
2.11. A registered User shall be able to login. - priority 1

3. Admin

3.1. An admin shall be required to approve or deny user's posts to go online. - priority 1
3.2. An admin shall be able to remove users. - priority 3

Non-functional Requirements

6.**List of non-functional requirements**

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0.

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.

3. All or selected application functions shall render well on mobile devices.

4. Data shall be stored in the database on the team's deployment server.

5. No more than 50 concurrent users shall be accessing the application at any time.

6. Privacy of users shall be protected.

7. The language used shall be English (no localization needed).

8. Application shall be very easy to use and intuitive.

9. Application shall follow established architecture patterns.

10. Application code and its repository shall be easy to inspect and maintain.

11. Google analytics shall be used.

12. No e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application.

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

14. Site security: basic best practices shall be applied (as covered in the class) for main data items.

15. Media formats shall be standard as used in the market today.

16. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools.

17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application).

Competitive Analysis

7.**Competitive analysis (functions/features only, not business or marketing)**:

Feature	**SwiftSell**	**Facebook Marketplace**	**Amazon**	**OfferUp**	**Craigslist**
Product Listings	++	++	++	++	++
Local Services (e.g., Tutoring/Delivery)	++	+	-	+	+
User Authorization (SFSU Email)	++	-	-	-	-
Admin Approval	++	-	-	-	-
Text Search	++	+	+	+	+

"++" Superior; "+" Feature Exists; "-" Does Not Exist

Swiftsell is designed to exclusively serve San Francisco State University, facilitating a direct connection between students, faculty, and staff for buying/selling goods. By bringing the SFSU community (e.g. students, faculty, and staff) together, our site serves as the perfect intermediary to connect people together, allowing the buying/selling of items relevant to student, faculty and staff. Differentiating ourselves from competitors, Swiftsell goes beyond the typical buy/sell of goods, where users will be able to offer services such as tutoring or delivery services to benefit the SFSU community (e.g. students, faculty, and staff). Furthermore, by implementing SFSU email verification and admin-approved listings, it ensures a trusted environment tailored towards this academic lifestyle. Swiftsell facilitates connections within the SFSU community (e.g. students, faculty, and staff) for academic materials and local services, providing a seamless experience for those looking to engage in exchanges within the SFSU setting.

Our search feature is superior to that of the competition because it is tailored to SFSU students, allowing them to search for items by class/course number. This customization makes it easier for students to find relevant academic materials and other items specific to their courses, providing a more efficient and effective search experience within the SFSU community.

High-level System Architecture and Technologies

8.**High-level system architecture and technologies used:**

Server Host: AWS: EC2
Operating System: Ubuntu 22.04.3

Database: MySQL v. 8.3.0

Web Server: Apache 2.4.58
Server-Side Language: Python 3.12.2

Additional Technologies:
Web Framework: Flask
Frontend Framework: Bootstrap
IDE: VsCode
Web Analytics: Google Analytics
SSL Cert: DV SSL
SASS: 3.5.5

Use of GenAI Tools

9.Use of GenAI tools like ChatGPT and copilot

OpenAI's ChatGPT (Versions 4.0 and 3.5)

1. ***Task: Competitive Analysis***
    a. ***GenAI Tool/Version:*** *ChatGPT (GPT-4.0)*
    b. ***Helpfulness Rating:*** *High*
    c. ***Comments:***
        i. *Helpful in searching for a unique idea which differs us from other marketplaces, after giving insight on potential ideas which we had along with how our platform works, chatgpt gave recommendations on which implementation would be best suited for a user.*
        ii. *Furthermore it was helpful in creating the paragraph for our executive summary, allowing for comments from the professor to be taken into consideration when reviewing our summary. Looking at the use of certain verbages and how to effectively communicate our ideas.*
        iii. *Example prompt: "In creating our marketplace, we are trying to establish a unique feature which differentiates us from the rest, some ideas we have are the ability to offer local services such as tutoring/delivery, would this be beneficial and unique to our platform? Are there potentially other implementations that may be better?"*
        iv. *Example prompt2: "Within the competitive analysis summary, suggestions were made to avoid the use of the word 'transaction' along with the word 'community' as they are too broad, what are some suggestions to alter our summary to fit as best as possible?"*

Team and Roles

Team and roles:

Aymane Arfaoui	aarfaoui@sfsu.edu	Team Lead
Markus Reyer	mreyer1@sfsu.edu	GitHub Master, Frontend and Backend help.
Amandeep Singh	asingh51@sfsu.edu	Frontend Lead
Alexis Alvarez	aalvarez26@mail.sfsu.edu	Backend
David Daly	ddaly@sfsu.edu	Backend Lead

Team Lead Checklist

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
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing | Done
- 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.) | Done

SW Engineering CSC648-848

Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

Milestone 2

Submitted	Revised
March 23rd 2024	April 2nd 2024

Executive Summary

There are many challenges when it comes to buying, selling and trading items related to university functions. Our team has realized that it is very difficult for students, faculty and staff at San Francisco State University to have access to a safe online marketplace that caters to their needs as members of the university. Other online marketplaces often do not offer products such as used textbooks that are usable for SFSU classes and users are forced to interact with strangers with no ties to their school's community which increases their risk to their safety. This is where our team has come up with the online marketplace SwiftSell. Our project is motivated by the need to create a safe, convenient, and user-friendly marketplace that provides unique functions specially designed for the SFSU community. SwiftSell will differ from other online retailers by providing services specifically tailored to the needs of college students, faculty and staff members. Doing so, SwiftSell seeks to enhance the overall experience at SFSU.

SwiftSell will offer a range of functions and services tailored to the needs of the members of SFSU. These include verified user profiles with the use of SFSU emails, local listings convenient for users to complete their transactions, as well as a dashboard for sellers to be able to view messages from prospective buyers. The platform will also feature search categories, including textbooks searchable by class, furniture, and electronics. The platform will also provide the user with the ability to list and accept job listings such as assisting in moving furniture sold on our marketplace and listing tutoring services for university classes. By providing a market focused on student, staff, and faculty needs, SwiftSell aims to be the all in one platform for local school transactions and provide additional services related to the campus.

Our team is composed of driven and innovative students from diverse backgrounds, all united by a shared vision of enhancing the student experience at SFSU. With an emphasis on the user experience, we aim to create an easy to use and attractive marketplace for users to interact with. With expertise in Computer Science and the guidance of our CEO Dragutin Petkovic and CTO Anthony Souza, we are equipped to develop and launch a successful marketplace platform.

Together, we are committed to making SwiftSell a valuable resource for members of the SFSU community.

List of main data items and entities

Items for Sale: Objects available for sale by registered users.
Associated data fields include:

- Owner (registered user who created the post)
- Category (e.g., Furniture, Electronics, Textbooks, Services)
- Description (e.g., description of item, tutoring subject and hours)
- Price (price of item or hourly rate for services)
- Picture (photo of item or image of flyer for services)
- Thumbnail (generated from the user-supplied image)
- Live (is the item visible on the site? - admin function)
- Date posted
- Item id

Dashboard: A screen view within the website, available to registered users, which displays messages sent to the user and items posted by the user
.

Messages: Communication between two registered users on the website.
Associated data fields include:

- Sender
- Recipient
- Associated item
- Content of message
- Timestamp of when it was sent
- Unread/read by the recipient

- Message id

Register/Registration: The process whereby an unregistered user becomes a registered user by creating a username and password.

User: Visitors to the web application are part of one of three groups: unregistered users, registered users, and admin.
Associated data fields include:

- Email address (sfsu.edu)
- Username
- First name
- Last name
- Password
- Date joined
- User id

Unregistered User: An unregistered user is not allowed to sign into the website. The unregistered user is not allowed to login, post, or message, and cannot view the dashboard; the unregistered user is allowed to view items for sale, to fill out the form to post an item or send an item (but not submit the form), and to register.

Registered User: In addition to the actions allowed to an unregistered user, a registered user is allowed to sign into the website. Once signed in, a registered user is allowed to post items for sale (with admin moderation), message other registered users, and view the dashboard.

Admin: In addition to the actions allowed to a registered user, the admin shall have the responsibility to approve posts for sale that conform to terms of service.

Functional Requirements - prioritized

Priority 1

- Unreg user
  - An unregistered User shall be able to search for specific items.
  - An unregistered User shall be able to browse the website's items
  - An unregistered User shall be able to see images for items/services with accurate description
  - An unregistered User shall be able to categorize items for sale by area of use (electronics, textbooks, furniture, etc)
  - An unregistered User shall be able to sort items for sale by majors.
  - An unregistered User shall be able to filter through items based on price, date posted.
  - An unregistered user shall be able to register
- Reg user inherits all functions of unregistered user plus shall have these below.
  - A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale.
  - A Registered User shall be able to post items and/or services(tutoring,delivery, etc) for sale with approval of admin
  - A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item.
  - A Registered User shall be able to message to sellers about items for sale.
  - A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc.
  - A Registered User shall be informed if another user wants one of their items via in-site messaging.
  - A Registered User's post shall wait to be approved by the admin.
  - A Registered User shall be able to delete a post without the approval of an admin.
  - A registered User shall be able to login.

- Admin inherits all functions of registered user plus shall have these below
    - An admin shall be required to approve or deny User posts to go online.
    - An admin shall be able to delete users.

Priority 2

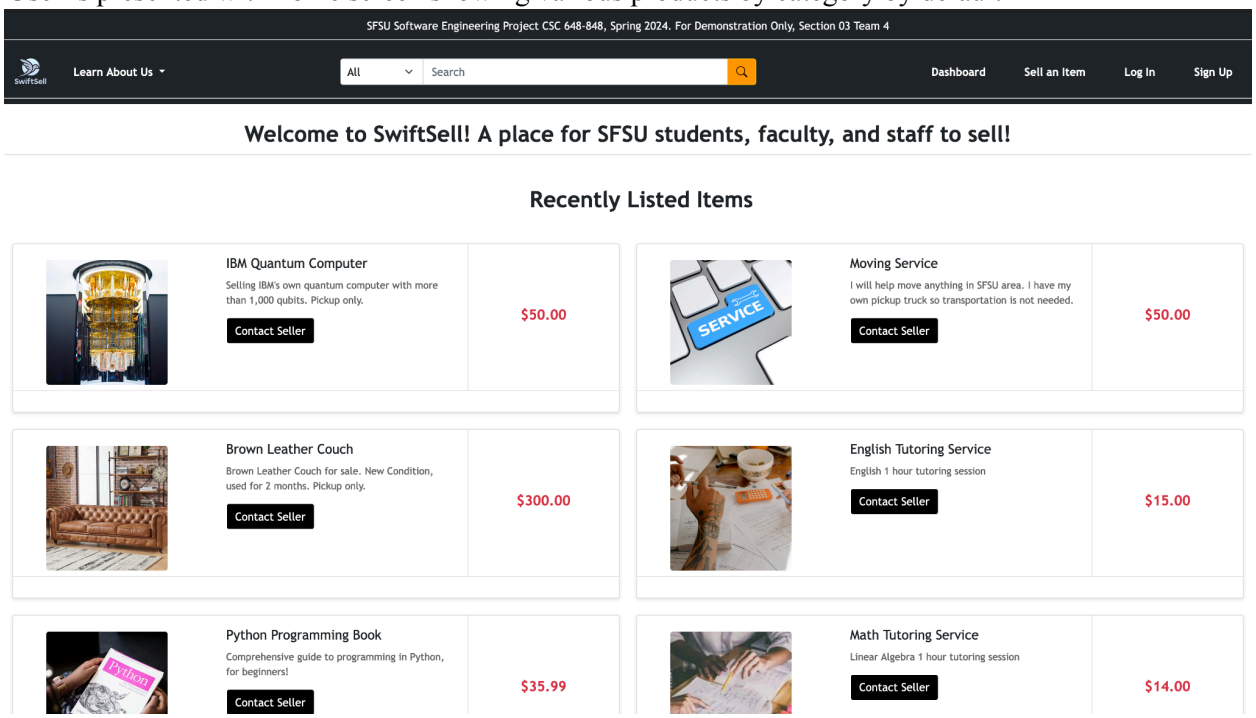
- Unregistered user
- Reg user

Priority 3

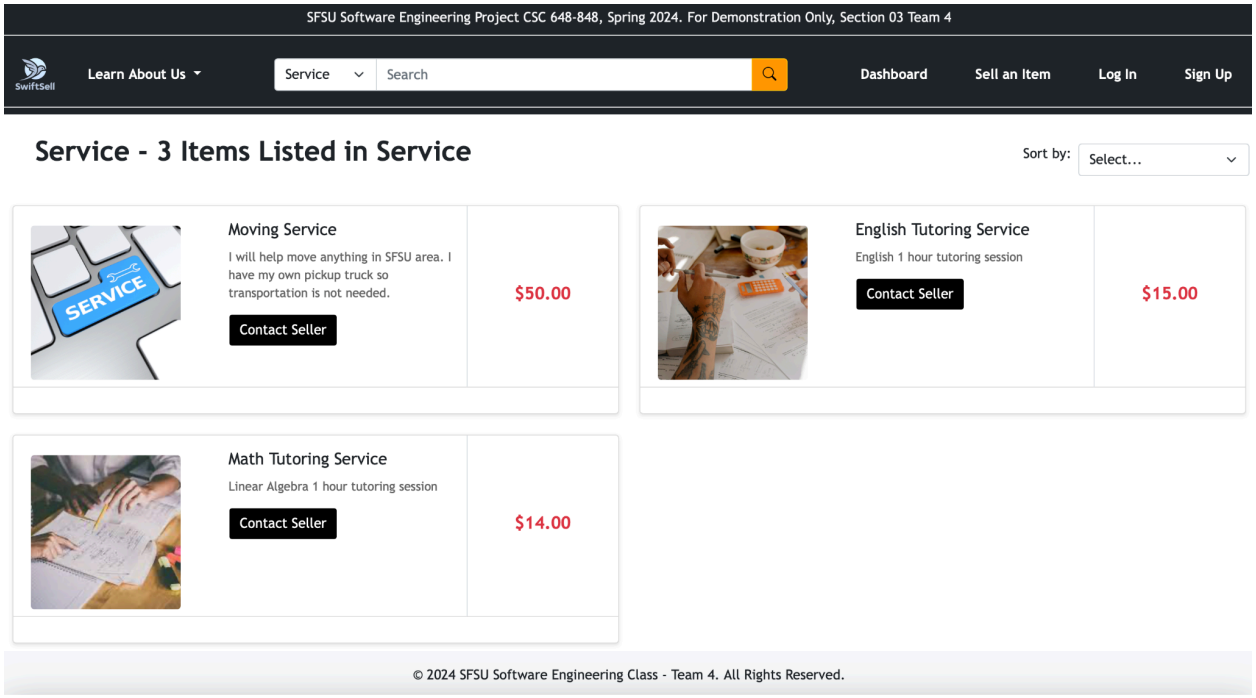
- Reg user
- Admin

UI Storyboards

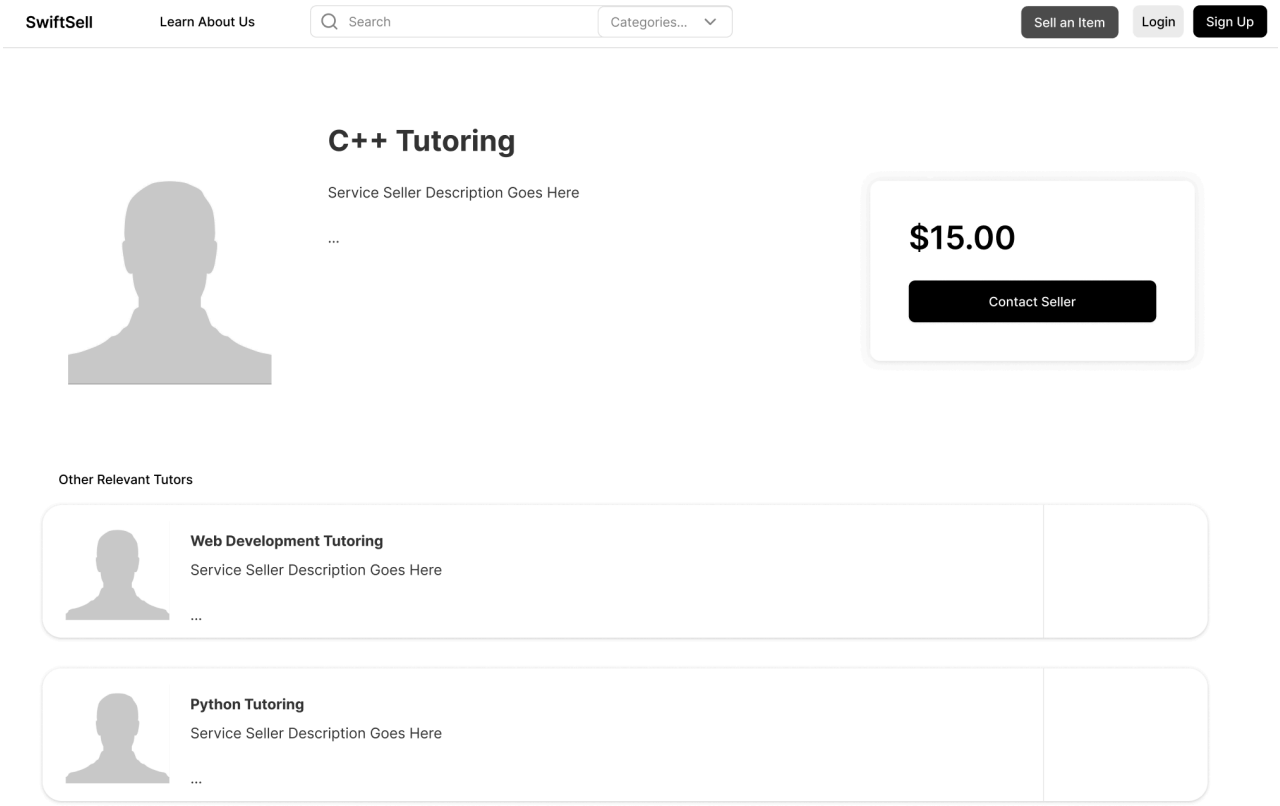
User is presented with home screen showing various products by category by default



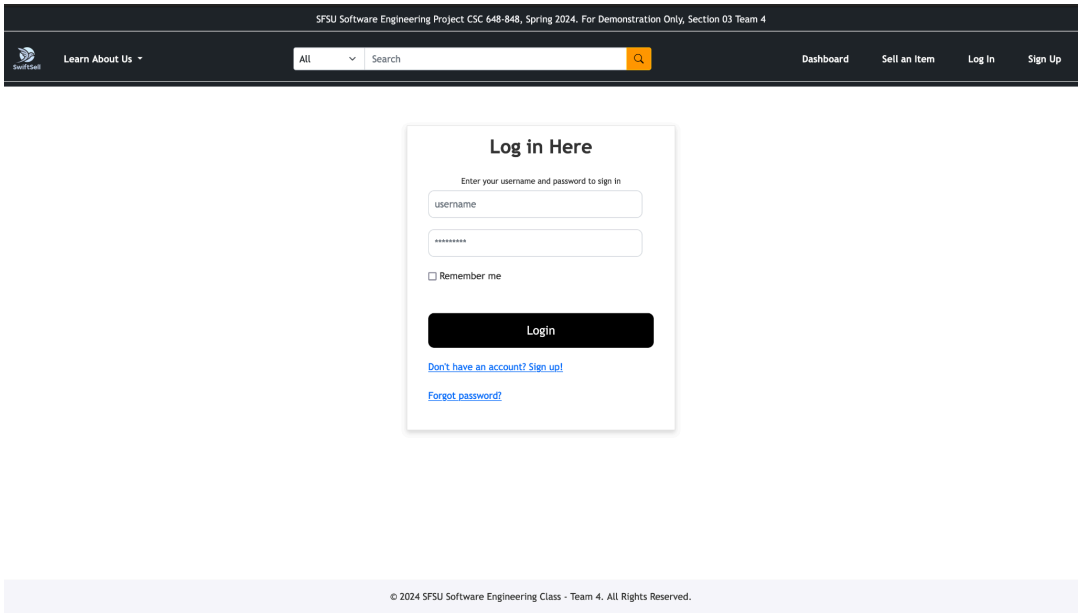
- User will be able to search for products/services showing SFSU specific function

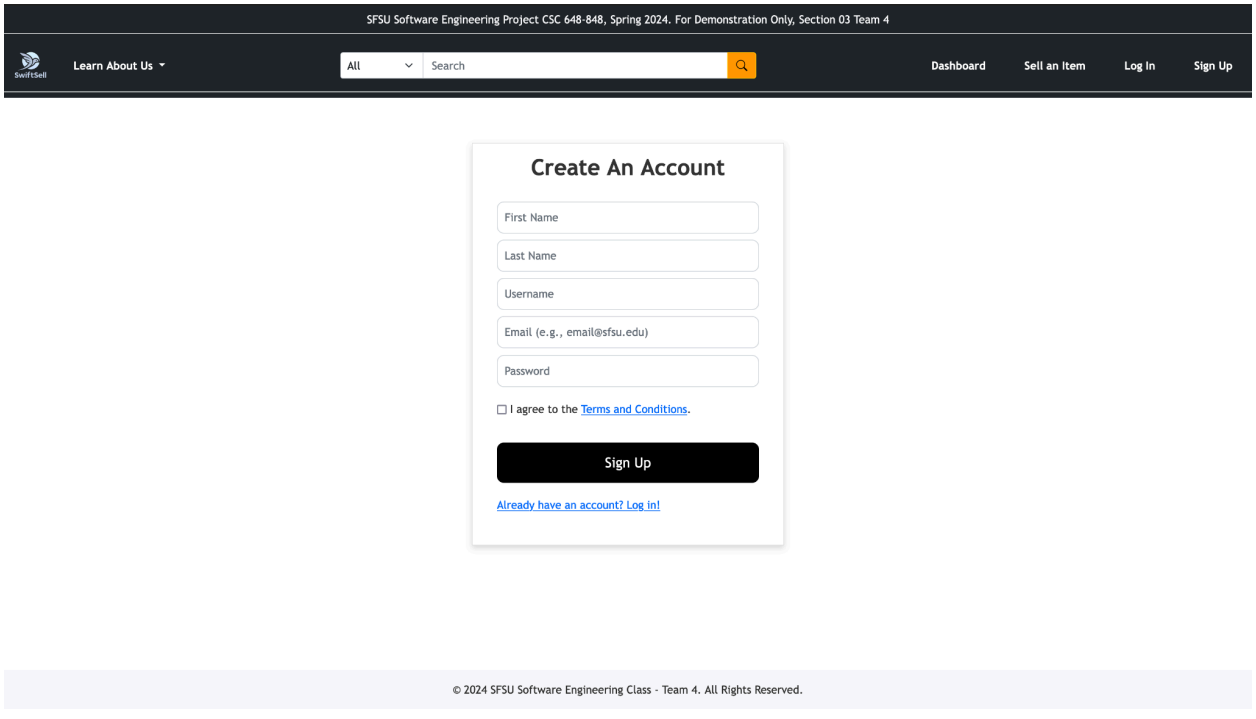


- Users shall be able to click on the offer to see the expanded view allowing the ability to contact the seller directly, other relevant listings will also be shown.

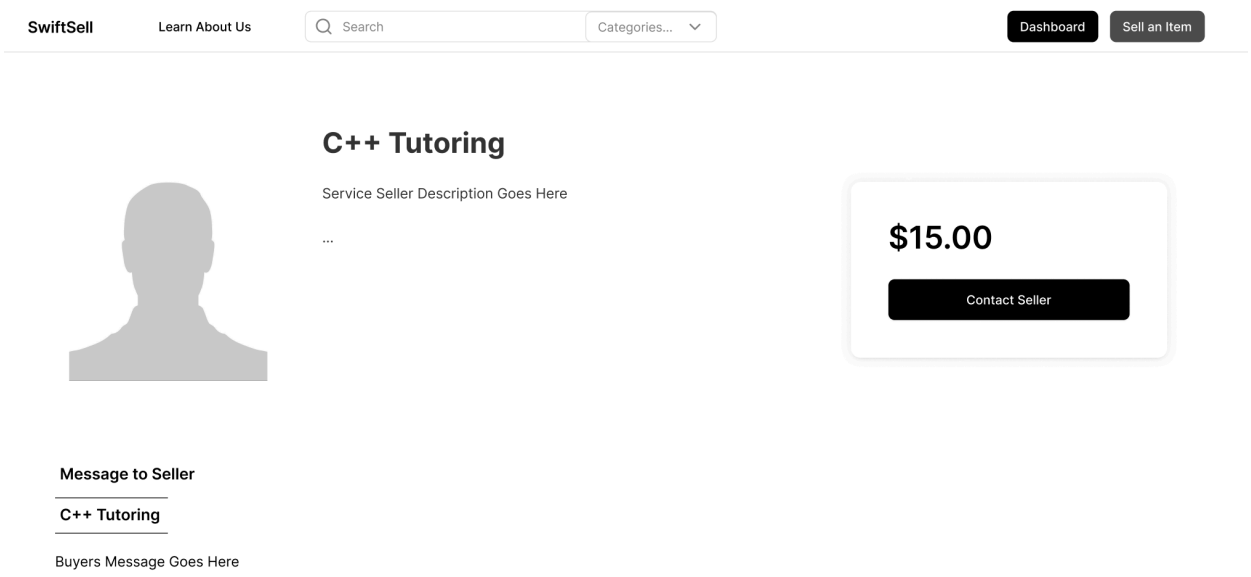


- User shall be required to log in or create an account when trying to contact the seller as they must be a part of SFSU.

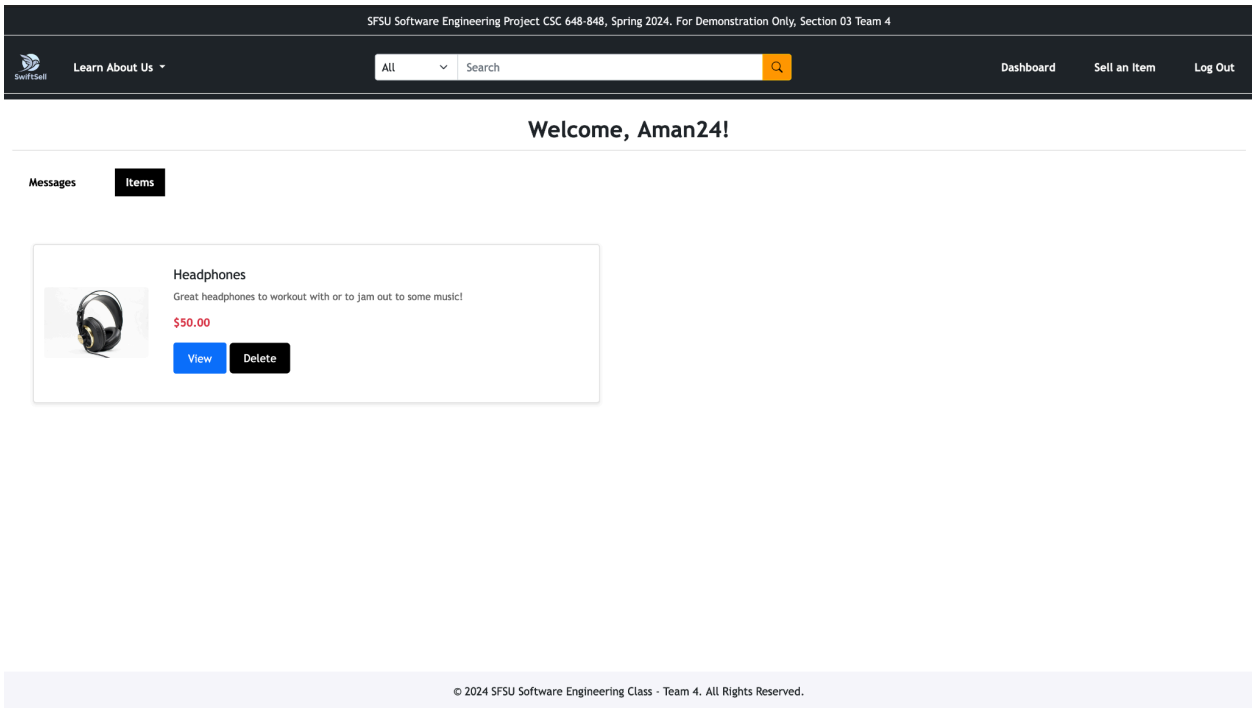




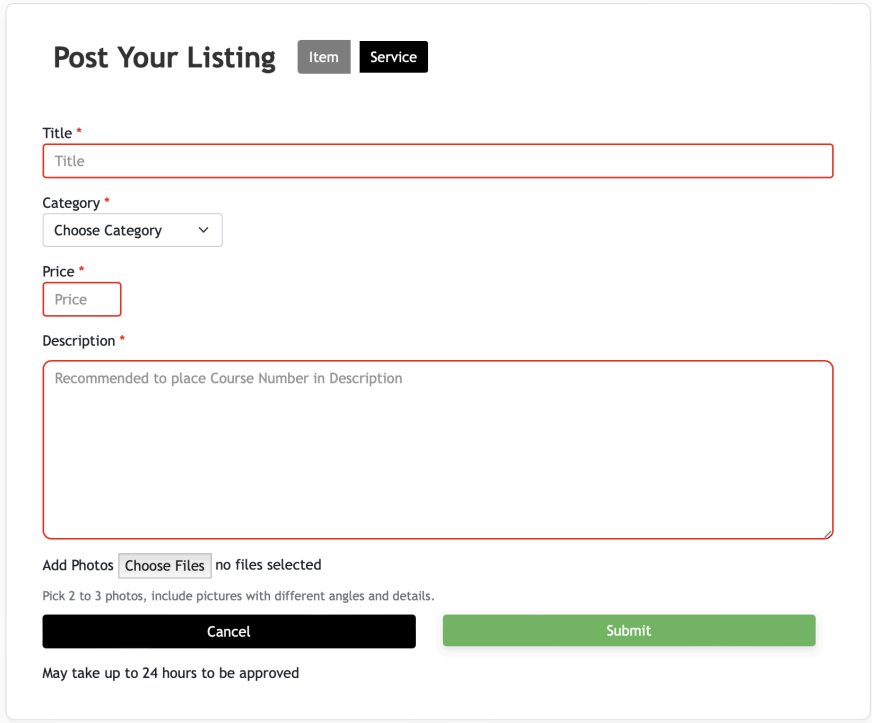
- Registered user shall be allowed to send a message directly to the seller, where they must input a way of communication.

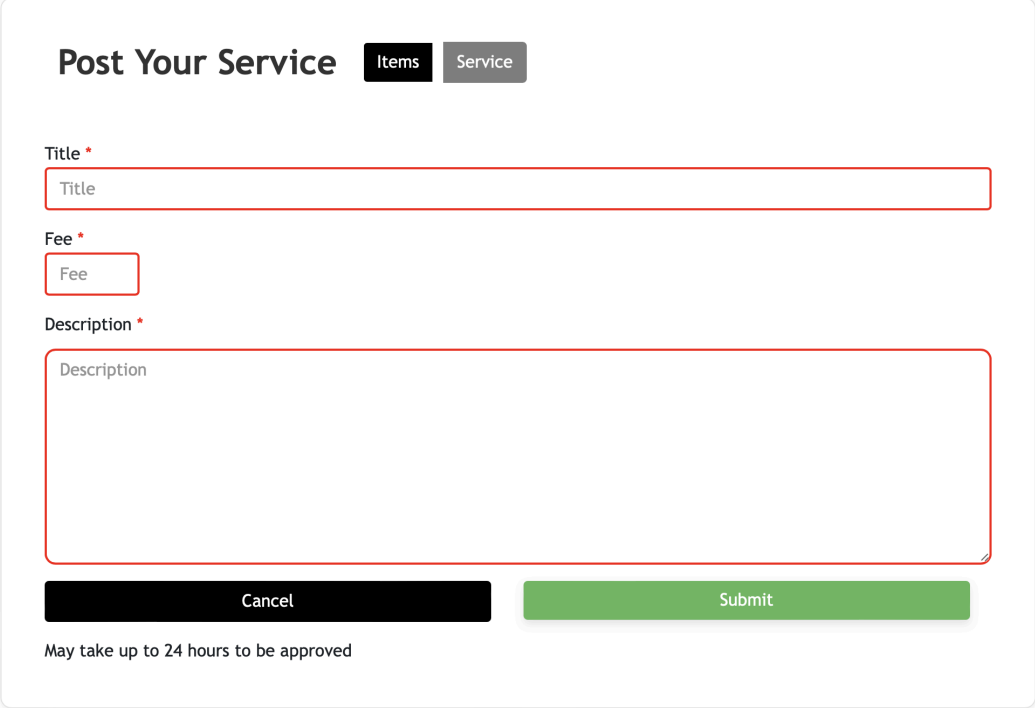


- A user’s dashboard shall contain a tab for messages and items showing the most recent inquiries on their items.



- Registered users will have the option to either post an item or service for sale where a tab at the top will indicate the type of entity they are selling





High Level Architecture

registered_user:

- user_id : INT AUTO_INCREMENT PRIMARY KEY, Unique identifier for each user.
- Username: VARCHAR(255), Username for other users to see
- first_name : VARCHAR(255), First name of the user
- last_name: VARCHAR(255), Last name of the user
- password: Hashed password for user authentication.
- email: VARCHAR(255), Email address of the user.
- registration_date: DATETIME(), Date and time when the user registered.
- last_login: Date and time of the user's last login.
- account_status: Status of the user account (active, suspended, etc.).

message:

- Message_id: INT PRIMARY KEY Unique identifier for each message.
- sender_id : INT FOREIGN KEY, Links to the UserID in the Registered Users table.
- recipient_id: INT FOREIGN KEY, Links to the UserID in the Registered Users table
- item_id: INT FOREIGN KEY, Links to the ItemID in the Items For Sale table
- content: VARCHAR(255), Text content of the message.
- message_date: DATETIME(), Date and time when the message was sent.

items_for_sale:

- item_id : Unique identifier for each item.
- seller_id : Links to the UserID in the Registered Users table.
- title: Title of the item.
- description: Description of the item.
- price: Price of the item.
- category_id: Category of the item
- availability: Availability status of the item (available, sold, pending, etc.).

- listed_date: Date and time when the item was listed.
- live: Is the posting visible on the website (admin function)
- high_res_image_url: URL of the high-resolution image of the item.
- thumbnail_image_url: URL of the thumbnail image of the item.

categories :

- Books
- Electronics
- Furniture
- Moving
- Tutoring

1. **Media Storage**

   We will be using file system approach to store our images.

   Storage Location: images will be stored directly on the server's file system. This entails creating a designated folder or directory within the server's file structure where these images will be saved.

   Accessing Images: To facilitate access to these stored images on the frontend of the application, Flask provides a mechanism to serve static files. Flask's send from directory function serves static files, including images.

   We will use relative pointers to root, and ensure file system protection.

2. **Search architecture and Implementation**

We will be using the preferred option of SQL %like method as described in architecture class slides.

In our search implementation, we will utilize the SQL function Like to enable users to search for terms and find results that are similar to their query.

When a user performs a search, we will use SQL queries with the LIKE operator to match search terms against the attributes of goods.

Implementation:

The backend will receive user queries, construct SQL queries with the LIKE operator, execute them against the database, and retrieve matching good entries.

Key Risks

Schedule Risk:

One of the potential issues is related to the schedule risk. Considering the fact that we would have to meet multiple times, especially regarding the coding, it is not always easy to find time considering how we all have different schedules. This is why we decided to separate tasks and reduce our meeting time where we would tackle problems instantly and quickly. If more time is needed, only the people who are assigned tasks can talk for longer periods of time. We constantly use discord to communicate (through chat). We also are experimenting using Trello to manage our tasks and see if it is a good fit for the needs of our team.

Skill Risks:

One of the potential risks we will probably face is related to skills. Considering the fact that it is the first time we work in a team project of this size. We have encountered some technical issues previously regarding the connection of our website to the amazon server. It is also the first time we work with a python framework flask. To address this issue, we will dedicate a discord channel specifically to issues our team might encounter in order to use our combined knowledge to try to fix it. Strong communication is key in solving such issues. We also plan on leveraging AI tools to guide us in solving our issues. Finally, if a certain problem persists, we will seek guidance from an experienced developer such as CTO (professor Souza).

Project Management

Our team has primarily made use of Discord to post updates on our project's progress. Our team lead has consistently set up meetings through Discord 1-2 times per week to allow the team to check in and ensure each team member is aware of the requirements that need to be completed for our project milestone. The meetings also serve as a place to assign tasks to team members so that each part of the project is being worked on. Discord is also used to send quick messages to the team of project updates, such as if a new branch has been created in the github, or if a problem has arisen. Using Discord has proven effective to keep the team updated through its direct messaging. Our team has also set up accounts with Trello and have begun posting checklist objectives and updating our "Doing" card to keep the team up to date on if a problem is being worked on. Our team plans to continue to make use of Trello, as it provides a team member the ability to check if a certain part of the project is being worked on and whether or not tasks have been completed. Our team is also able to add tasks to the Trello Dashboard which proves helpful for when new problems arise as updates can be lost in the Discord channel.

8. Use of genAI tools like ChgatGPT and copilot

- GenAI was used as a guide for the Executive Summary. Once feedback was given for the executive summary, the team modified the summary to fit the requirements for the milestone. Usefulness for this part was Medium as GenAI did not fully understand the the task at hand

- GenAI was used for suggestions on the story board in terms of what user shall see when logging into the page. Along with how to format the products, giving ideas on the design of our item “cards”. Usefulness in this scenario as well was low, as the real design was fully constructed by us and many changes were brought while making the mock-up

- GenAI was used to establish some concrete solutions in terms of the schedule risk and skill risks. GenAI was also used to provide examples of what the database tables should look like for a registered user, a sales_item, messages and categories. In this scenario, the usefulness of GenAI was High as it provided clear options to choose from to make our database.

- GenAI was used to provide critical and in-depth information as to how images should be stored as well as best practices for our search algorithm. GenAI also provided visual information as to how images could best be stored and managed in our project. GenAI also provided examples for implementing our search algorithm. Usefulness here was Medium as it guided the search on how to incorporate these implementations.

Team Lead Checklist:

So far all team members are fully engaged and attending team sessions when required | **DONE**

Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing | **DONE**

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**

SW Engineering CSC648-848
Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

Milestone 3

04/24/2024

Team Number: Team 04
Meeting Date: April 24th 2024

Summary of Feedback on UI:

1. Navbar should have a larger font and a distinct color (suggested: light yellow) to differentiate it from the rest of the page.

2. "Sort by" should be placed next to search results, not in the navbar.

3. Reduce text on the front page and increase the font size. Stop after "buy and sell" in the description.

4. Correct the trademark text by removing vertical lines and ensuring proper formatting.

5. Pre-fill recipient and item fields in the messaging page.

6. Provide guideline text for message fields, suggesting users to include contact info.
7. Add a "Cancel" button on the message page and consider opening a new message in a separate tab.

8. Increase the "2 of 2 items" text size in search results and remove "All Items" text.

9. Suggest formatting for book search with examples like "Enter book title or class number, e.g. CSC 648."

Summary of Feedback on Code and Architecture:

1. Ensure the architecture adheres to the MVC pattern.

2. Follow consistent coding styles and include minimal agreed-upon documentation.

3. Conduct code reviews to verify adherence to standards.

Summary of Feedback on GitHub Usage:

1. Include more descriptive comments in GitHub and inline code comments.
2. Use concise and consistent commit messages.

Summary of Feedback on Database:

Ensure passwords are encrypted for security.

Summary of Feedback on Teamwork and Risk Management:

1. Teamwork is solid with no significant risks identified.
2. Continue following P1 features as previously agreed upon.
3. Consider adding more items to the database for demonstration purposes.

Architecture Review Check:

We are confirming that the architecture follows the MVC pattern and has acceptable coding styles.

List of Agreed P1 Features:

1. An unregistered User shall be able to search for specific items.
2. An unregistered User shall be able to browse the website's items.
3. An unregistered User shall be able to see images for items/services with accurate description.
4. An unregistered user shall be able to register.
5. A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale.
6. A Registered User shall be able to post items and/or services(tutoring,delivery,etc) for sale with approval of admin
7. A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item.
8. A Registered User shall be able to message sellers about items for sale.
9. Only Registered Users shall be able to message other Registered Users.
10. A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc.
11. A Registered User shall be informed if another user wants one of their items via in-site messaging.
12. A registered User shall be able to login.
13. An admin shall be required to approve or deny user's posts to go online.

Other Comments and Issues:

1. Review all remaining P1 features to ensure a bug-free final product.
2. Keep documentation updated for the final submission.

Check Point (CP) Due: May 6th 2024

Important Considerations:

Prioritize P1 features based on product/user importance and ease of implementation. If we have time, work on P2.

SW Engineering CSC648-848

Spring 2024

Campus Buy/Sell Application: SwiftSell
Team 04

Team Members:
Aymane Arfaoui, Team Lead (aarfaoui@sfsu.edu)
Markus Reyer, Github Master
Amandeep Singh, Front End Lead
Alexis Alvarez, Back End
David Daly, Back End Lead

Milestone 4

05/22/2024

1. Executive Summary

Product Name: SwiftSell

Description:

SwiftSell is a dynamic marketplace platform developed by a team of passionate undergraduate, graduate, and postgraduate students, aimed at revolutionizing the way students buy and sell goods within the San Francisco State University (SFSU) community. Our project is motivated by the need to create a safe, convenient, and student-friendly marketplace that offers an alternative to generic platforms like Facebook Marketplace and Craigslist. By providing a platform specifically tailored to the needs of college students, SwiftSell seeks to enhance the overall student experience at SFSU.

Major Committed Functions (P1):

1. An unregistered User shall be able to search for specific items.
2. An unregistered User shall be able to browse the website's items
3. An unregistered User shall be able to see images for items/services with accurate description.
4. An unregistered User shall be able to register.
5. A Registered User shall have a dashboard with information containing order history, messages, notifications, and the items they posted for sale.
6. A Registered User shall be able to post items and/or services(tutoring,delivery, etc) for sale with approval of admin
7. A Registered User who posts items for sale shall upload at least one image, a brief description of said item, and a reasonable price for the item.
8. A Registered User shall be able to message sellers about items for sale.
9. Only Registered Users shall be able to message other Registered Users.
10. A Registered User shall be able to post small jobs such as tutoring, moving furniture, etc.
11. A Registered User shall be informed if another user wants one of their items via in-site messaging.
12. A Registered User shall be able to login.
13. Verified Users shall have profiles via SFSU email authentication.
14. A User shall be able to search items by course number.
15. An Admin shall be required to approve or deny user's posts to go online

Unique Features:

1. Exclusively tailored to the SFSU community with verification through SFSU email addresses.
2. Local listings and **job services** specifically for students, faculty, and staff members.
3. Enhanced search capabilities for finding class-specific textbooks and other items.

Access URL:

http://ec2-34-228-231-71.compute-1.amazonaws.com/

2. Usability Test Plan for Search Functionality

Test Objectives

The objective of this usability test is to evaluate the effectiveness, efficiency, and user satisfaction of the search functionality within our web application. We aim to ensure that users can easily and accurately find the information they need.

Test Background and Setup

System Setup:

The web application should be hosted on a live server, accessible via the internet.

The search functionality must be fully implemented and operational.

Required hardware for testers includes a computer or mobile device with internet access.

Starting Point:

Testers should start from the homepage of the web application.

Hardware Needed:

A computer or mobile device with an internet connection.

Screen recording software to capture the tester's interactions, if applicable.

Intended Users:

Users for this functionality are mainly SFSU students, faculty, and staff.

URL of the System to be Tested:

http://ec2-34-228-231-71.compute-1.amazonaws.com/

Test Environment:

The test can be conducted anywhere..

No technical equipment is required.

No formal training is required before the test.

Plan for Evaluation of Effectiveness

Plan for Evaluation of Effectiveness: Effectiveness will be measured by the accuracy and completeness of the search results. This will be evaluated by tracking the number of correct results the user finds, comparing the user's search results with a predefined set of correct results, and measuring the percentage of successful searches where users find the required information within the first three attempts.

Plan for Evaluation of Efficiency

Efficiency will be measured by the time it takes for users to complete search tasks and the number of steps they take. This can be evaluated by recording the time from when the user initiates the search until they find the correct information, counting the number of clicks or interactions required to complete the search task, and evaluating the user's ability to refine their search queries to achieve better results.

Plan for Evaluation of User Satisfaction

We will instruct users to complete the following tasks: Search for specific items or services on our website and contact a seller or inquire about a service. After completing these tasks, users will fill out a survey that includes Likert scale questions to assess their subjective thoughts on the website's effectiveness, efficiency, and overall satisfaction.

Usability Task Description:

Users will be instructed to perform the following tasks before filling out the Likert scale questionnaire:

1. Navigate to the homepage of the web application.
2. Use the search bar to find information on a specific topic (e.g., "search CSC 415 textbook").
3. Note down the first three results and assess their relevance to the search query.
4. Refine the search query to get more accurate results if necessary.
5. Spend no more than 3 minutes on this task.

Likert Scale Evaluation Entries:

Ease of Use:

The search functionality was easy to use (circle one).

Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree

Relevance of Results:

The search results were relevant to my query (circle one).

Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree

Overall Satisfaction:

I am satisfied with the overall performance of the search functionality (circle one).

Strongly Disagree | Disagree | Neutral | Agree | Strongly Agree

3. QA test plan and QA testing

Test Objectives

The objective of this QA test plan is to ensure the reliability, correctness, and overall quality of the search functionality within the SwiftSell platform. This involves verifying that users can successfully search for items, including by course number, ensuring search results are accurate and relevant, the search input persists, and the user interface operates smoothly across different browsers.

HW and SW Setup

Hardware:

- A computer or mobile device with internet access.

Software:

- Three major web browsers: Google Chrome, Mozilla Firefox, and Safari.
- SwiftSell platform URL: http://ec2-34-228-231-71.compute-1.amazonaws.com/

Feature to be Tested

- The search functionality, including searching by item name and course number, with persistent search input and filters.**QA Test Plan**

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL)
1	Basic Search Query	Enter "Book" in the search field and verify that the results are relevant to textbooks	On the home page of SwiftSell, go to the search bar located at the top of the page. Enter "Books" in the search field, and then click the "Search" button to the right of the search bar.	2 items related to books are displayed and the search input persists	Chrome: PASS / Safari: PASS
2	Advanced Search Query	Enter "Book" in the search field and verify that the results are relevant and filtered correctly from lowest to highest price	On the home page of SwiftSell, go to the search bar located at the top of the page. Enter "Book" in the search field, and then click the "Search" button to the right of the search bar. Next, locate the filter under the "Sign Up" page on the right-hand side of the website. In the dropdown menu, select "Sort by price."	Items related to books are sorted from the lowest to highest price and are displayed and the search input persists	Chrome: PASS / Safari: PASS
3	Empty Search	Do not enter anything in the search field and verify that all the results are displayed	On the home page of SwiftSell, go to the search bar located at the top of the page. Do not enter anything in the search field, and then click the "Search" button to the right of the search bar. Next, locate the filter under the "Sign Up" page on the right-hand side of the website. In the	All the items will be returned, up to a maximum of 10 items.	Chrome: PASS / Safari: PASS

			dropdown menu, select "Sort by price."		
4	Search by Course Number	Enter "CSC415" in the search field and verify that the results are relevant to CSC415 course materials	On the home page of SwiftSell, go to the search bar located at the top of the page. Enter "CSC415" in the search field, and then click the "Search" button to the right of the search bar.	2 Items related to CSC415 are displayed and the search input persists	Chrome: PASS / Safari: PASS
5	Persistent Search Functionality	Select the category "furniture" and then type "brown" to see if the category filter stays persistent and only furniture items that contain the keyword "brown" are displayed.	On the home page of SwiftSell, go to the search bar located at the top of the page. Select the category "furniture" and then type "brown" in the search field. Click the "Search" button to see if the category filter stays persistent and only furniture items that contain the keyword "brown" are displayed. Navigate to another page on the website, then return to the search results page to verify that the search input and filter persist.	Items related to the input "furniture" are displayed, showing only furniture that are brown, and the applied filter remains in the search field.	Chrome: PASS / Safari: PASS

4. Peer Code Review

CSC 648-848 Spring 2024 Team04 Peer Review

AS

Amandeep Singh <asingh51@sfsu.edu> Yesterday at 8:51 PM

To: Aymane Arfaoui; Markus Reyer; Alexis Alvarez; Dave Daly; +1 more

Hello Team,

Please review the code for the search functionality on SwiftSell. Here is the link to the code repository (lines 30-117): https://github.com/CSC-648-SFSU/csc648-sp24-03-team04/blob/featureAlexis/application/app.py

Please focus on the following areas:

- Basic header and in-line comments
- Naming conventions (class, method, variable names)
- Functionality and efficiency
- Commit comments

Thank you,
Amandeep Singh

Re: CSC 648-848 Spring 2024 Team04 Peer Review

AA

Aymane Arfaoui <aarfaoui@sfsu.edu> Today at 9:52 PM

To: Amandeep Singh; **Cc:** Dragutin Petkovic

Hi Aman !

Hope you are doing well, thanks for submitting your code, I spent some time looking at it and here's my feedback.

1. Security:

   First thing is I noticed how you use the parametrization of queries for preventing SQL injection attacks which is excellent.

   One thing I would suggest however is to not hardcode database credentials and secret keys (like it was done starting from line 20 (after this comment)).
   # Database connection info. Note that this is not a secure connection.

2. Comments and Documentation:

   The code includes comments, but more detailed explanations would be beneficial.

   I believe that adding a header comment at the beginning of the file and more detailed comments within the code would be helpful.

3. Modularity principle:

   The code for the search functionality is contained within a single route, search, which includes database connection, query execution, and rendering the template.

   To be consistent with the modularity concept, having helper functions for database connection and query execution would improve readability, maintainability and would help in debugging scenarios.

Besides, everything looks great ! Very good work Aman !

Regards,
Aymane Arfaoui

```
from werkzeug.utils import secure_filename
import uuid as uuid

app = Flask(__name__,
            template_folder='templates',
            static_folder='src/static',
            static_url_path='/static')

# file path to store user images
# Absolute path to the directory where images should be saved
BASE_DIR = os.path.abspath(os.path.dirname(__file__))  # Absolute directory of this script
app.config['UPLOAD_FOLDER'] = os.path.join(BASE_DIR, 'src', 'static', 'images')


# Database connection info. Note that this is not a secure connection.
db_config = {
    'user': 'root',
    'password': [redacted],
    'host': '127.0.0.1',
    'database': 'swiftselldb'
}

# Setting the secret key to a random collection of characters. Tell no-one!
app.secret_key = '[redacted]'

# Secure cookie settings
# app.config['SESSION_COOKIE_SECURE'] = True
# app.config['SESSION_COOKIE_HTTPONLY'] = True
# app.config['SESSION_COOKIE_SAMESITE'] = 'Lax'
```

5. Security Best Practices

Major Assets and Their Protection

Asset	Major Threats	Protection Strategy
User confidential data such as passwords and images	Unauthorized access, data breaches, identity theft	Encrypt data at rest and in transit, implement authentication mechanisms, relative paths for images.
Transaction information	Unauthorized access, data manipulation	Use secure connections (HTTPS), validate and sanitize all inputs, and log and monitor transactions for unusual activity (admin)
Application code	SQL injection	Validate all inputs, use parameterized queries, and perform regular code reviews and security testing Search bar also prevent form submission if non-alphanumeric characters are present and only allows up to 40 characters.
User sessions	Unauthorized access	Use secure cookies, implement session timeouts.
Trademarked and Copyrighted Content	Stolen content	Add our own copyright and/or trademark to our original materials; apply for trademark protection, if appropriate

User Data Protection

1. **Password Encryption**
    - generate_password_hash and check_password_hash from werkzeug.security for hashing passwords.
2. **Authentication**
    - **Method**: Implement strong password policies, including requirements for complexity and regular updates.

Transaction Information Protection

1. **Secure Transmission**
    - **Method**: Ensure all transactions are conducted over HTTPS to protect data in transit.
2. **Input Validation**
    - **Method**: Validate and sanitize all transaction inputs to prevent injection attacks.

Application Code Protection

1. **Code Reviews**
    - **Method**: Perform regular code reviews and use static analysis tools to detect security vulnerabilities.
2. **Queries with Placeholders**
    - **Method**: parameterized queries with placeholders (e.g., %s) for all database queries

User Session Protection

1. **Secure Cookies**
    - **Method**: Use secure and HttpOnly cookies to protect session data.
2. **Session Management**
    - **Method**: Implement session timeouts and invalidate sessions after a period of inactivity.

6. Adherence to Original Non-functional Specs

1. Copy of All Original 17 Non-functional Requirements

1. Tools and Servers: Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0.

2. Desktop Optimization: Application shall be optimized for standard desktop/laptop browsers, e.g., must render correctly on the two latest versions of two major browsers.

3. Mobile Optimization: All or selected application functions shall render well on mobile devices.

4. Data Storage: Data shall be stored in the database on the team's deployment server.

5. Concurrent Users: No more than 50 concurrent users shall be accessing the application at any time.

6. Privacy Protection: Privacy of users shall be protected.

7. Language: The language used shall be English (no localization needed).

8. Usability: Application shall be very easy to use and intuitive.

9. Architecture Patterns: Application shall follow established architecture patterns.

10. Code Inspection and Maintenance: Application code and its repository shall be easy to inspect and maintain.

11. Google Analytics: Google Analytics shall be used.

12. Messaging: No e-mail clients shall be allowed. Interested users can only message sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application.

13. Payment Functionality: 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: Media formats shall be standard as used in the market today.

16. Modern SE Processes and Tools: Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools.

17. UI Text: The application UI (WWW and mobile) shall prominently display the following exact text on all pages: "SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only" at the top of the WWW page Nav bar.

2. Status of Each Requirement

Tools and Servers: Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0.
Status: DONE

Desktop Optimization: Application shall be optimized for standard desktop/laptop browsers, e.g., must render correctly on the two latest versions of two major browsers.
Status: DONE

Mobile Optimization: All or selected application functions shall render well on mobile devices.
Status: DONE

Data Storage: Data shall be stored in the database on the team’s deployment server.
Status: DONE

Concurrent Users: No more than 50 concurrent users shall be accessing the application at any time.
Status: DONE

Privacy Protection: Privacy of users shall be protected.
Status: DONE

Language: The language used shall be English (no localization needed).
Status: DONE

Usability: Application shall be very easy to use and intuitive.
Status: DONE

Architecture Patterns: Application shall follow established architecture patterns.
Status: DONE

Code Inspection and Maintenance: Application code and its repository shall be easy to inspect and maintain.
Status: DONE

Google Analytics: Google Analytics shall be used.
Status: DONE

Messaging: No e-mail clients shall be allowed. Interested users can only message sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application.
Status: DONE

Payment Functionality: Pay functionality, if any (e.g., paying for goods and services) shall not be implemented nor simulated in UI.
Status: DONE

Site Security: Basic best practices shall be applied (as covered in the class) for main data items.
Status: DONE

Media Formats: Media formats shall be standard as used in the market today.
Status: DONE

Modern SE Processes and Tools: Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools.
Status: DONE

UI Text: The application UI (WWW and mobile) shall prominently display the following exact text on all pages: "SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only" at the top of the WWW page Nav bar.
Status: DONE

7. Use of genAI tools like ChatGPT and copilot

In Milestone 4, we utilized GenAI tools such as ChatGPT, focusing on securing our website and enhancing various aspects of our project involved in QA and usability. These tools were invaluable in guiding our code review process and providing key insights. Additionally, they helped us find effective metrics to evaluate user satisfaction, efficiency, and effectiveness. Below is a detailed description of our usage:

Tools Used: ChatGPT (version 4)

Tasks and Usefulness:

Securing the Website: HIGH

Description: We used ChatGPT to identify potential security vulnerabilities and suggest best practices for securing our website. It provided us with detailed explanations and actionable steps to improve our security measures.

Benefit: ChatGPT's insights helped us strengthen our security protocols, ensuring a safer environment for our users.

Guiding Code Review: MEDIUM

Description: ChatGPT and Copilot were used to review our code, highlighting areas that needed improvement and suggesting optimizations.

Benefit: These tools facilitated a more thorough and efficient code review process, though some recommendations required additional verification.

Evaluating Metrics for Satisfaction, Efficiency, and Effectiveness: HIGH

Description: ChatGPT provided us with relevant metrics and methods to measure user satisfaction, efficiency, and effectiveness. It suggested specific survey questions, evaluation techniques, and key performance indicators.

Benefit: This guidance allowed us to create a comprehensive evaluation plan that accurately measures our project's success.

Examples of Key Prompts and Responses:

Prompt for Security: "What are the best practices for securing a web application built with flask?"

Response: ChatGPT provided a detailed list of practices, including implementing HTTPS, using environment variables for sensitive data, validating user inputs, and setting up proper authentication and authorization mechanisms.

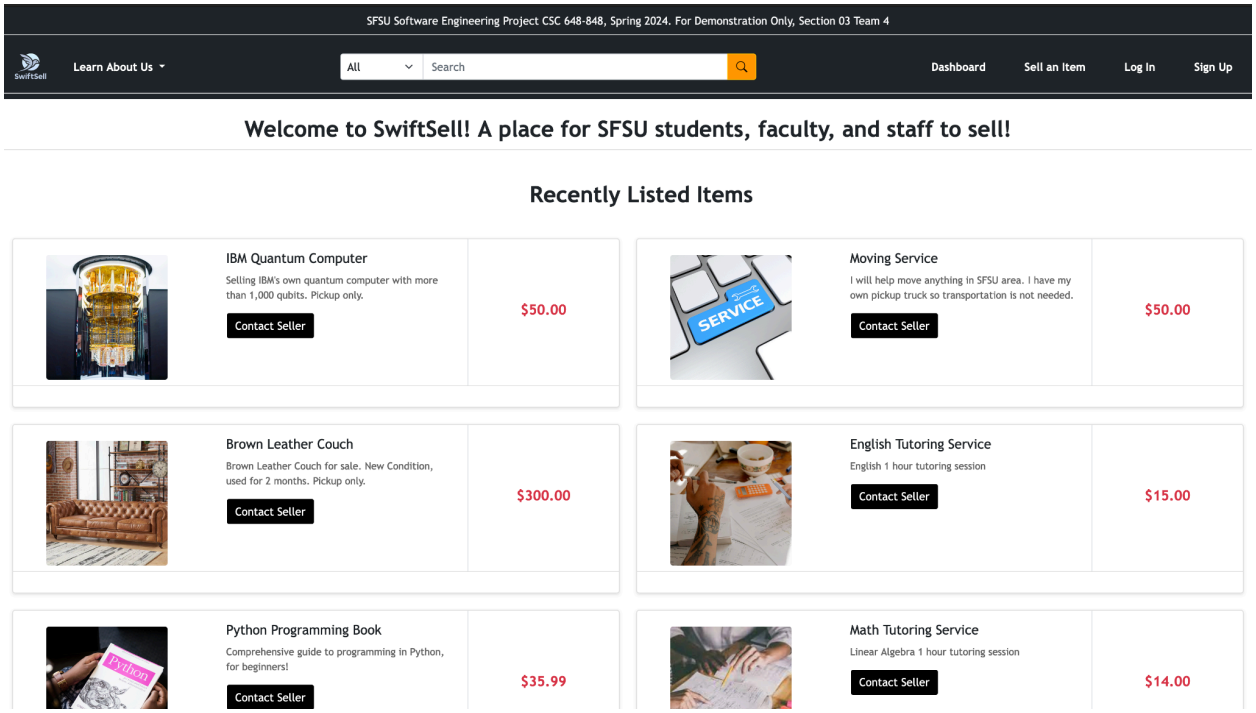
Prompt for Metrics: "What metrics should we use to evaluate user satisfaction and efficiency in a web application?"

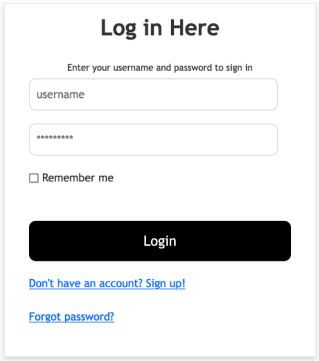
Response: ChatGPT suggested using Likert scale surveys for user satisfaction, tracking task completion times, counting the number of user interactions, and measuring the success rate of search queries.

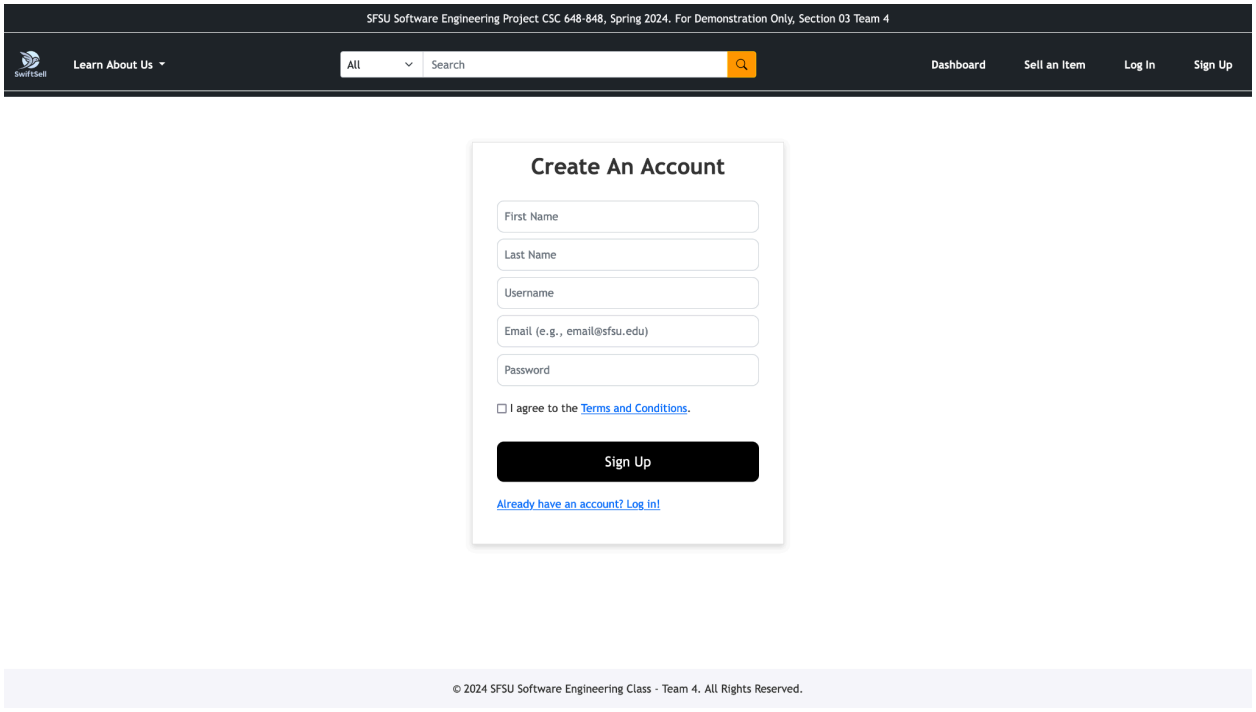
Additional Comments:

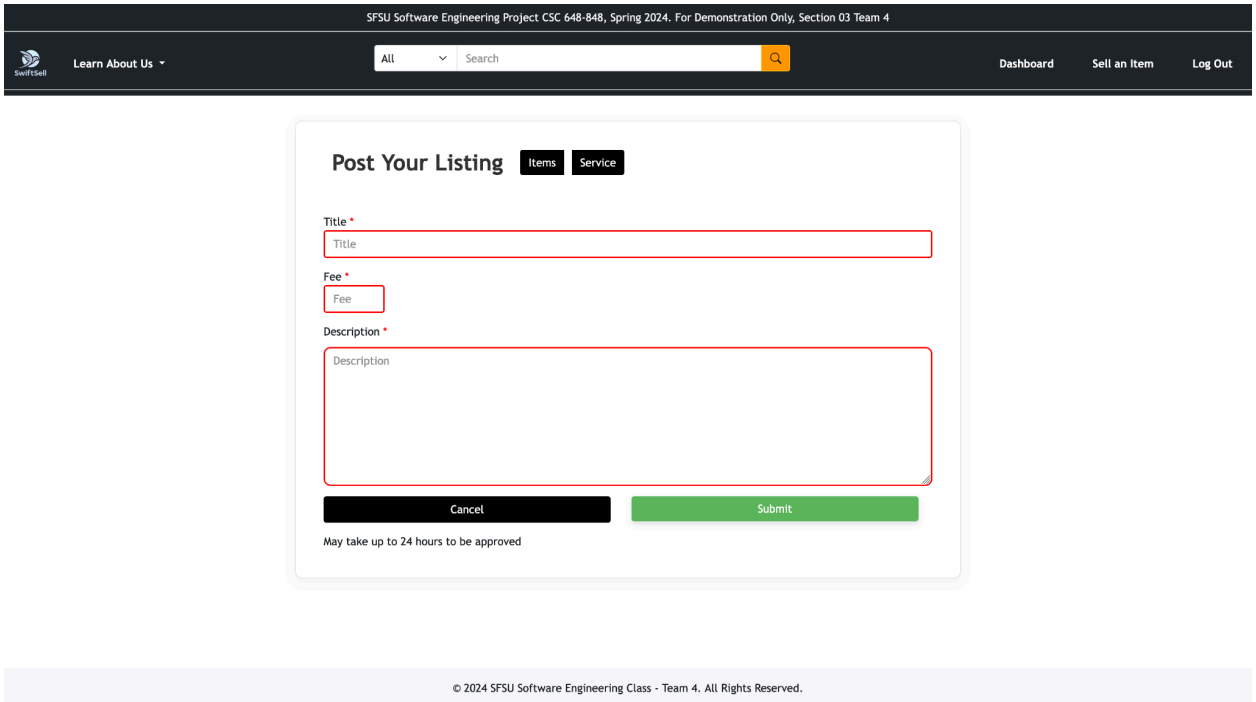
By integrating GenAI tools into our workflow, we were able to enhance the security, usability, and overall quality of our project, ensuring a better experience for our users.

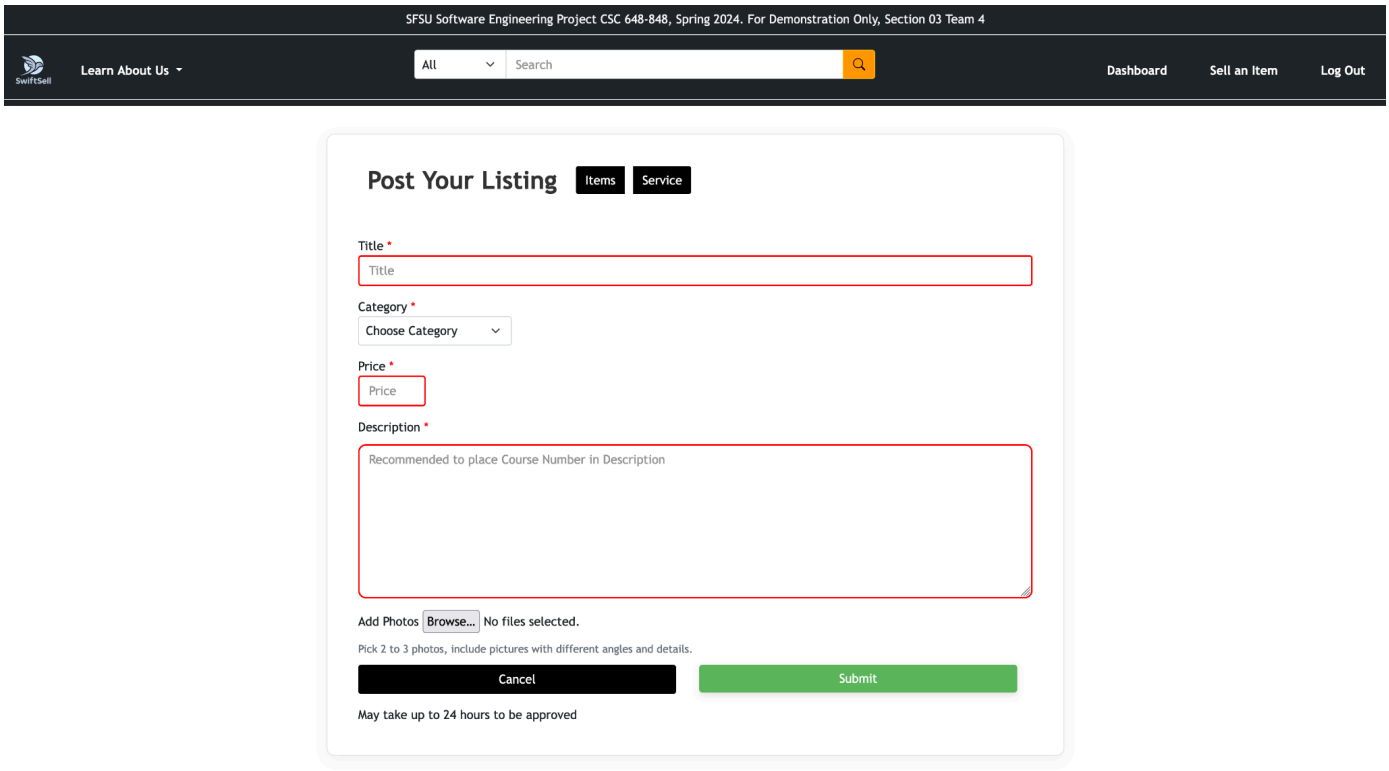
3) Product Screen Shots:






SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4
SwiftSell
Learn About Us
All
Search
Dashboard
Sell an Item
Log In
Sign Up
Create An Account
First Name
Last Name
Username
Email (e.g., email@sfsu.edu)
Password
I agree to the Terms and Conditions.
Sign Up
Already have an account? Log in!
© 2024 SFSU Software Engineering Class - Team 4. All Rights Reserved.


SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4
SwiftSell
Learn About Us
All
Search
Dashboard
Sell an Item
Log Out
Post Your Listing
Items
Service
Title *
Title
Fee *
Fee
Description *
Description
Cancel
Submit
May take up to 24 hours to be approved
© 2024 SFSU Software Engineering Class - Team 4. All Rights Reserved.


SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4
SwiftSell
Learn About Us
All
Search
Dashboard
Sell an Item
Log Out
Post Your Listing
Items
Service
Title *
Title
Category *
Choose Category
Price *
Price
Description *
Recommended to place Course Number in Description
Add Photos
Browse...
No files selected.
Pick 2 to 3 photos, include pictures with different angles and details.
Cancel
Submit
May take up to 24 hours to be approved

SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4

SwiftSell | Learn About Us | All | Search | Dashboard | Sell an Item | Log Out

Welcome, Aman24!

Messages | Items

Headphones

Aman2423

Great headphones! I'd really like to check them out, contact me at asingh@sfsu.edu!

May 22, 2024

SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4

SwiftSell | Learn About Us | All | Search | Dashboard | Sell an Item | Log Out

Welcome, Aman24!

Messages | Items

Headphones

Great headphones to workout with or to jam out to some music!

$50.00

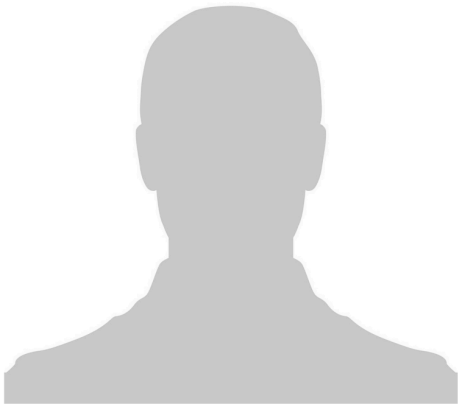
View | Delete

© 2024 SFSU Software Engineering Class - Team 4. All Rights Reserved.

SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4

SwiftSell | Learn About Us | All | Search | Dashboard | Sell an Item | Log Out

Amandeep Singh's About Me



Hello, nice to meet you!

I am a computer science student at San Francisco State University, currently pursuing my bachelor's degree. I am in my senior year and am looking to break into the industry soon enough!

Contact me

SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4

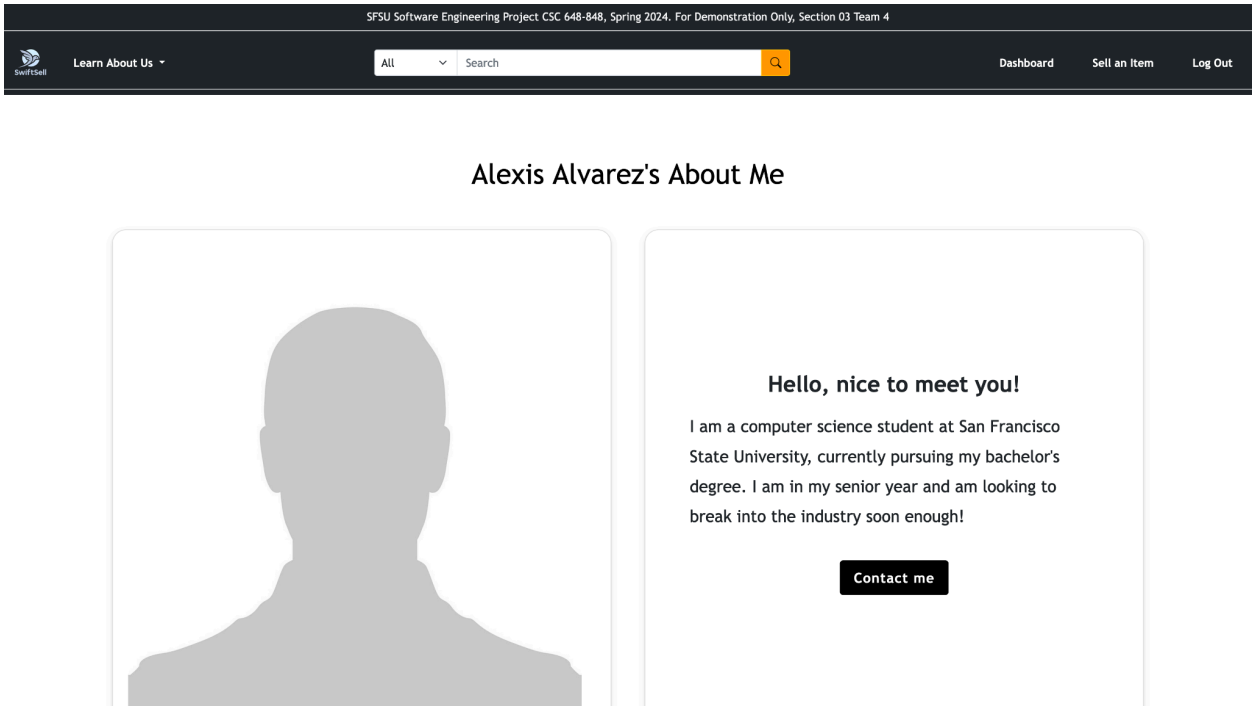
SwiftSell | Learn About Us | All | Search | Dashboard | Sell an Item | Log Out

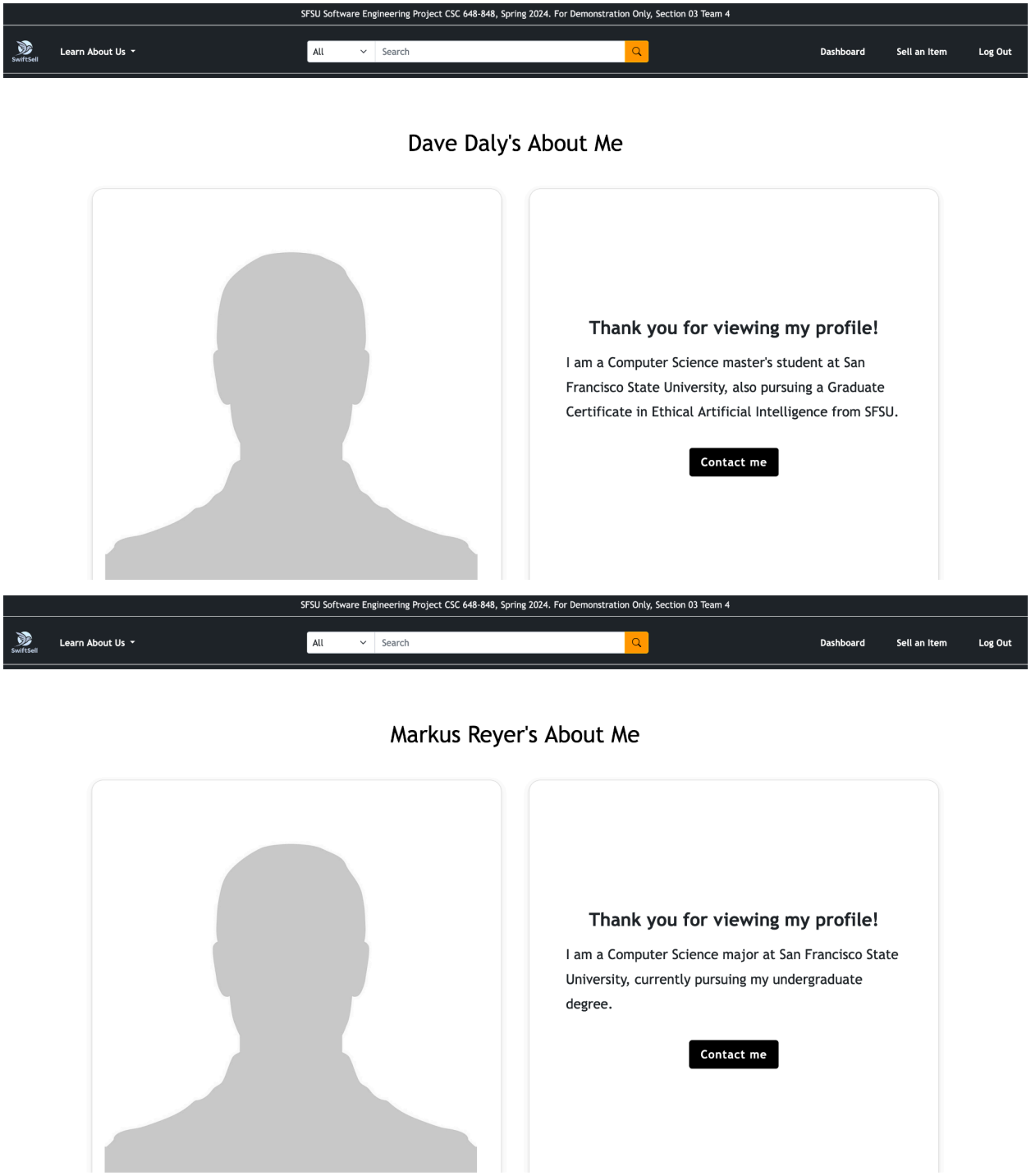
Aymane Arfaoui's About Me

Hi ! Great Meeting you

Hi, My name is Aymane Arfaoui. I am an exchange student from Montreal Canada. I am really happy to be doing an exchange term here at San Francisco State University. Our goal as a team is to deliver an excellent project for SFSU students where they can find a place to exchange textbooks, access tutoring lessons and much more !

Contact me


SFSU Software Engineering Project CSC 648-848, Spring 2024. For Demonstration Only, Section 03 Team 4
SwiftSell
Learn About Us
All
Search
Dashboard
Sell an Item
Log Out
Alexis Alvarez's About Me
Hello, nice to meet you!
I am a computer science student at San Francisco State University, currently pursuing my bachelor's degree. I am in my senior year and am looking to break into the industry soon enough!
Contact me



4) Database Organization: (Screen shots of key DB tables (1-3 pages)

```
1 • SELECT * FROM swiftselldb.categories;
```

Limit to 1000 rows

100% 38:1

Result Grid Filter Rows: Search Edit: Export/Import:

categories_...	category_name
1	Electronics
2	Furniture
3	Books
4	Service
NULL	NULL

categories 1 Apply

```
1 • SELECT * FROM swiftselldb.items_for_sale;
```

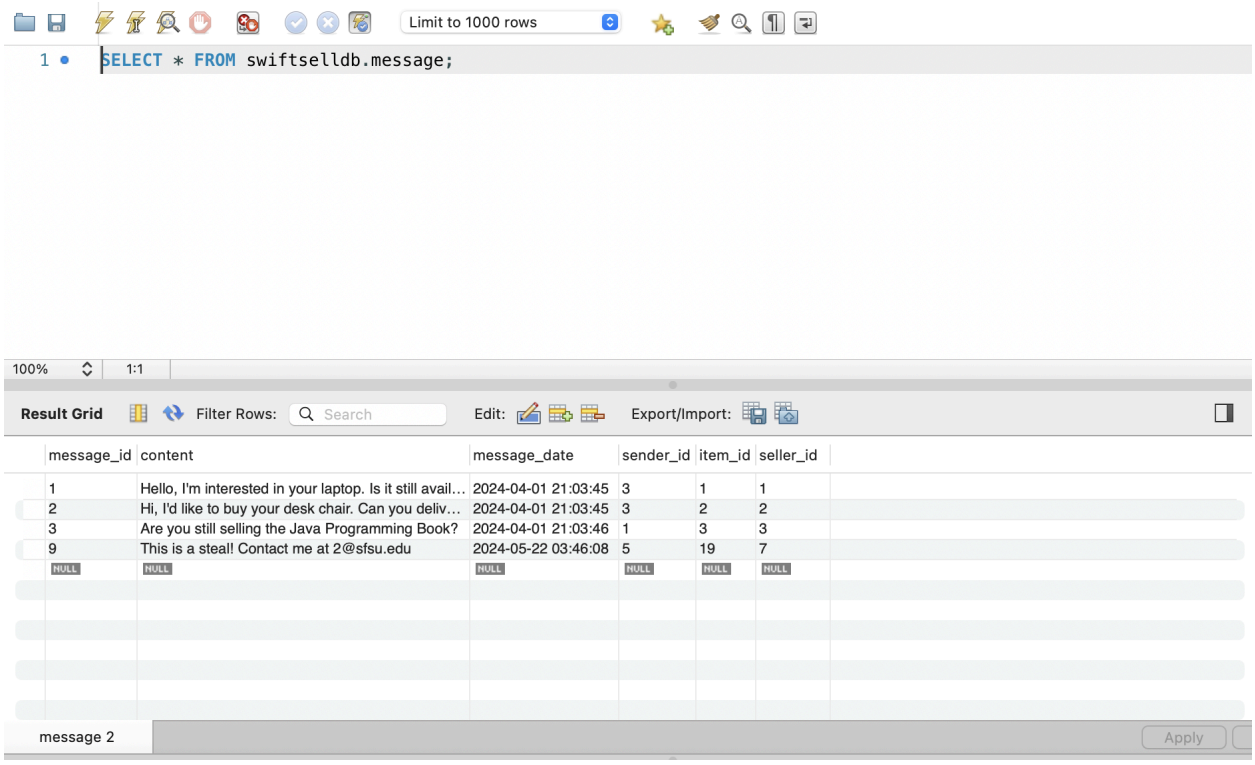
Limit to 1000 rows

100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import:

item_id	title	description	price	live	listed_date	availability	seller	category_id	photo_path
1	Laptop	High-performance laptop with SSD storage.	999.99	1	2024-04-01 20:48:09	In stock	1	1	images/bestlaptops-2(
2	Desk Chair	Ergonomic desk chair with adjustable armrests.	149.99	1	2024-04-01 20:48:09	In stock	2	2	images/deskchair.jpeg
3	Java Programming Book	Comprehensive guide to Java programming lan…	49.99	1	2024-04-01 20:48:09	In stock	3	3	images/javaTextbook.j
4	Math Tutoring Service	Linear Algebra 1 hour tutoring session	14.00	1	2024-04-01 21:08:01	In stock	3	4	images/tutorImage.jpe
5	English Tutoring Service	English 1 hour tutoring session	15.00	1	2024-04-01 21:35:01	In stock	3	4	images/english_tutor.j
6	Sofa	A comfortable sofa to relax on.	200.00	1	2024-04-01 20:08:01	In stock	2	2	images/sofa.jpg
7	Python Programming Book	Comprehensive guide to programming in Pytho…	35.99	1	2024-04-01 21:21:01	In stock	3	3	images/python.jpg
8	Speaker	Lightweight speaker with bluetooth connection	120.00	1	2024-04-01 15:14:01	In stock	1	1	images/speaker.jpg
15	Brown Leather Couch	Brown Leather Couch for sale. New Condition,…	300.00	1	2024-05-22 03:22:42	NULL	7	2	images/6a2fb137-33a
16	Moving Service	I will help move anything in SFSU area. I have…	50.00	1	2024-05-22 03:23:43	NULL	7	4	images/Service.jpg
19	IBM Quantum Computer	Selling IBM's own quantum computer with more…	50.00	1	2024-05-22 03:42:45	NULL	7	1	images/1eed01b6-ba6
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

items_for_sale 1 Apply



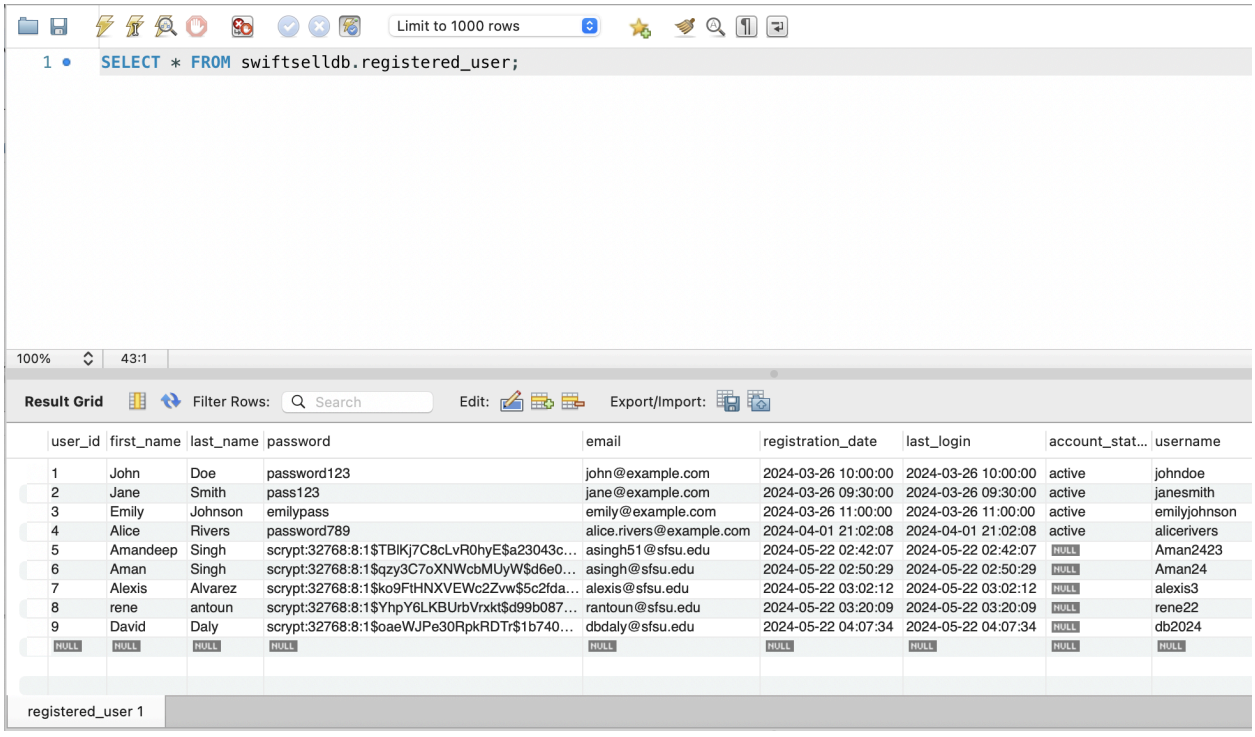
```
SELECT * FROM swiftselldb.message;
```

100% 1:1

Result Grid Filter Rows: Search Edit: Export/Import:

message_id	content	message_date	sender_id	item_id	seller_id
1	Hello, I'm interested in your laptop. Is it still avail…	2024-04-01 21:03:45	3	1	1
2	Hi, I'd like to buy your desk chair. Can you deliv…	2024-04-01 21:03:45	3	2	2
3	Are you still selling the Java Programming Book?	2024-04-01 21:03:46	1	3	3
9	This is a steal! Contact me at 2@sfsu.edu	2024-05-22 03:46:08	5	19	7
NULL	NULL	NULL	NULL	NULL	NULL

message 2 Apply



```
SELECT * FROM swiftselldb.registered_user;
```

100% 43:1

Result Grid Filter Rows: Search Edit: Export/Import:

user_id	first_name	last_name	password	email	registration_date	last_login	account_stat…	username
1	John	Doe	password123	john@example.com	2024-03-26 10:00:00	2024-03-26 10:00:00	active	johndoe
2	Jane	Smith	pass123	jane@example.com	2024-03-26 09:30:00	2024-03-26 09:30:00	active	janesmith
3	Emily	Johnson	emilypass	emily@example.com	2024-03-26 11:00:00	2024-03-26 11:00:00	active	emilyjohnson
4	Alice	Rivers	password789	alice.rivers@example.com	2024-04-01 21:02:08	2024-04-01 21:02:08	active	alicerivers
5	Amandeep	Singh	scrypt:32768:8:1$TBlKj7C8cLvR0hyE$a23043c…	asingh51@sfsu.edu	2024-05-22 02:42:07	2024-05-22 02:42:07	NULL	Aman2423
6	Aman	Singh	scrypt:32768:8:1$qzy3C7oXNWcbMUyW$d6e0…	asingh@sfsu.edu	2024-05-22 02:50:29	2024-05-22 02:50:29	NULL	Aman24
7	Alexis	Alvarez	scrypt:32768:8:1$ko9FtHNXVEWc2Zvw$5c2fda…	alexis@sfsu.edu	2024-05-22 03:02:12	2024-05-22 03:02:12	NULL	alexis3
8	rene	antoun	scrypt:32768:8:1$YhpY6LKBUrbVrxkt$d99b087…	rantoun@sfsu.edu	2024-05-22 03:20:09	2024-05-22 03:20:09	NULL	rene22
9	David	Daly	scrypt:32768:8:1$oaeWJPe30RpkRDTr$1b740…	dbdaly@sfsu.edu	2024-05-22 04:07:34	2024-05-22 04:07:34	NULL	db2024
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

registered_user 1

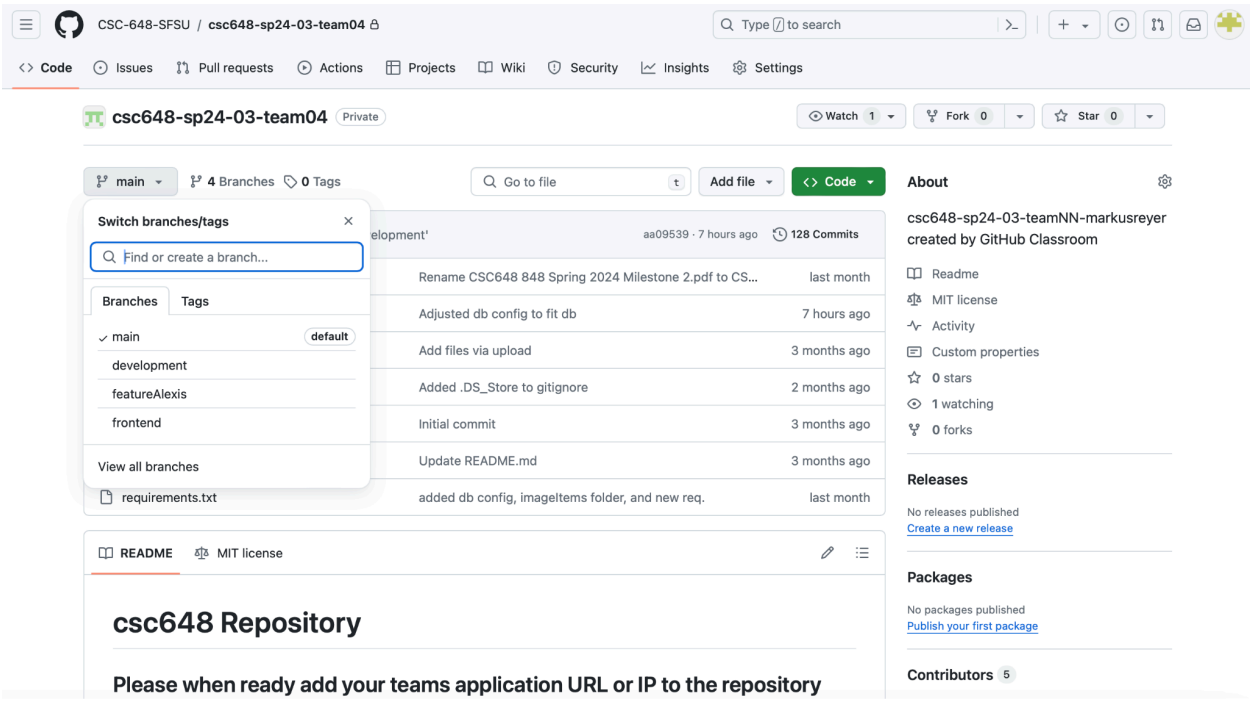
5) Github organization:

Branches:

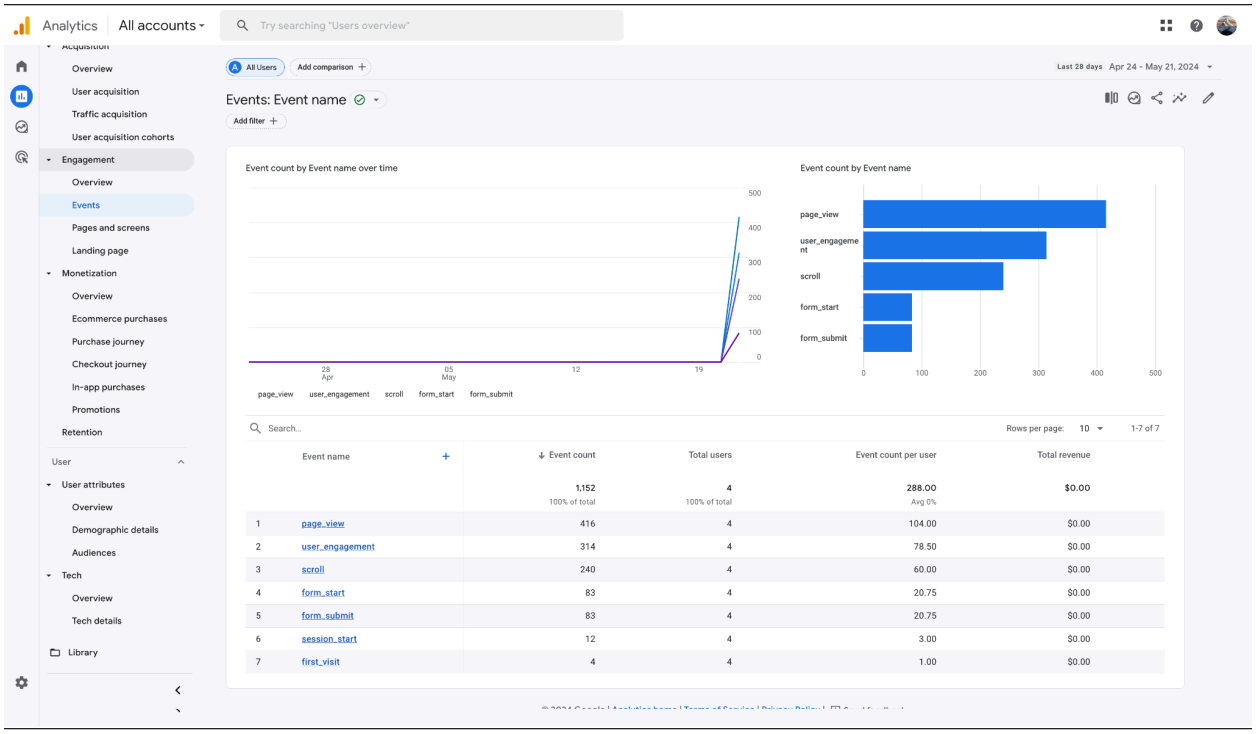
- Main (Final delivery)
- Development (Main branch used to push updates and milestones)
- featureAlexis (Branch used for backend testing before merging to development)
- Frontend (Branch used to test Frontend features before merging to development)

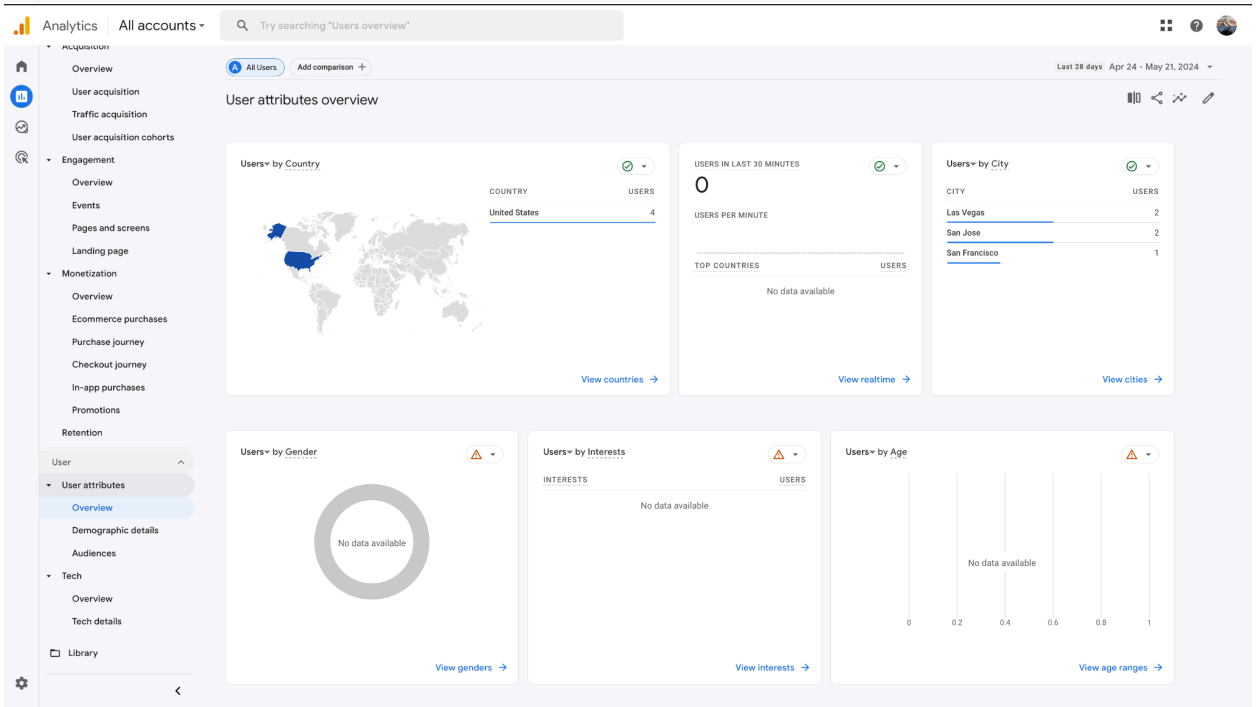
MasterBranch:

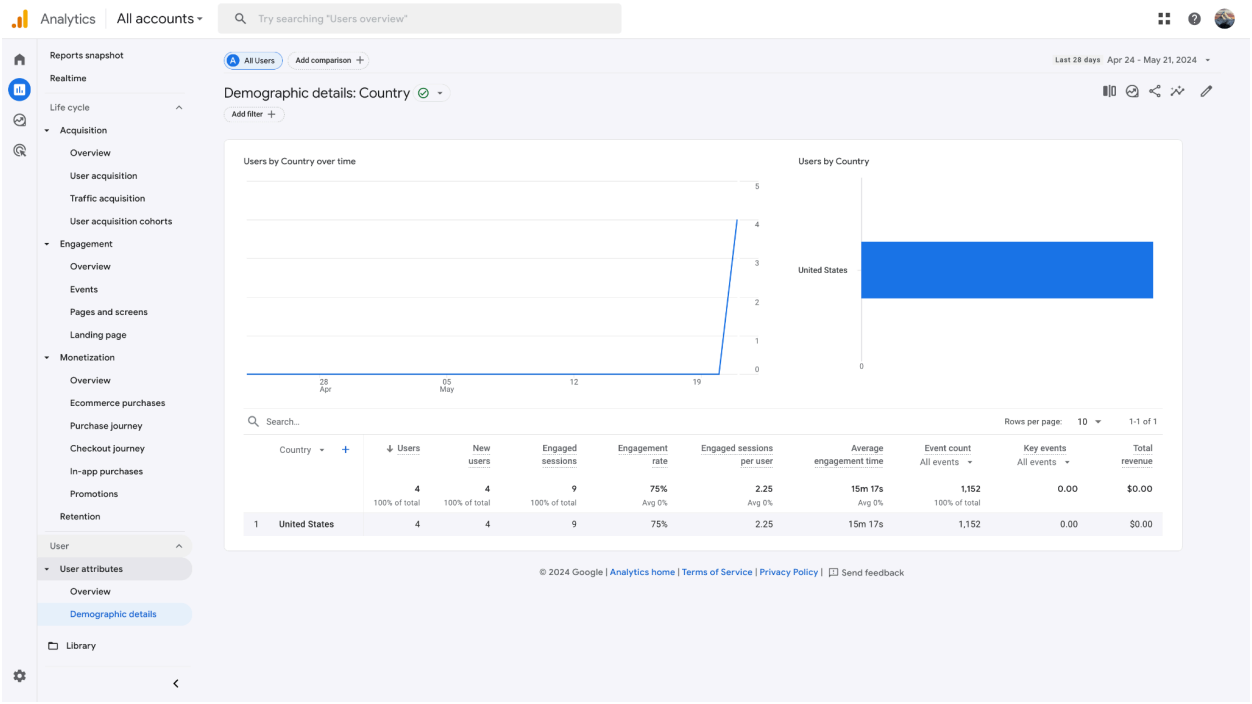
The Master branch was able to be accessed by Aymane, our team lead, and Markus, the Github master. Everyone else pushed to the development branch throughout the project's creation.



6) Google analytics stats plot for your WWW site







7) Project management

For project management, we briefly used Trello but did not find it particularly helpful for our team's workflow. Instead, we primarily relied on Discord and regular meetings to keep track of changes, updates, and to discuss any issues. Our approach was based on the SCRUM routine, which involved regular check-ins, sprint planning, and review sessions.

While we initially set up Trello boards for task tracking and progress monitoring, we quickly realized that Discord's real-time communication features better suited our needs for immediate collaboration and discussion. Here are some screenshots of our project management setup:

May 19, 2024

aymane4405 05/19/2024 4:14 PM
Hi Everyone! just a reminder that we are meeting with the teacher tomorrow at 9 am (edited)

aymane4405 05/19/2024 7:25 PM
milestone 3:
https://docs.google.com/document/d/1_Au9L1MkECiY7DSTCUIPCNiGBO_Jv_UoIqqiQKtcK38/edit?usp=sharing

Google Docs
MileStone 3

May 6, 2024

aymane4405 05/06/2024 11:05 AM
Meeting today guys at 6 pm
👍 1

Aman Singh 05/06/2024 5:37 PM
Bet

Aman Singh 05/06/2024 8:23 PM
pushed some changes to frontend right now, added details page (still workin on it), adjusted some of the other issues he had, made it bigger a bit, and fixed sign up so it goes into the db (local db for now) (edited)

> @Aman Singh pushed some changes to frontend right now, added details page (still workin...

Markus Reyer 05/06/2024 8:23 PM
That's awesome!

Aman Singh 05/06/2024 8:24 PM
yessirr
🙌 1

May 7, 2024

alexis 05/07/2024 2:02 PM
pushed to featureAlexis, fixed login. gonna work on post route now
check it out if anyone has a chance to make sure it works on your end

8) Team member self assessment and contributions

DD

Dave Daly

To: Aymane Arfaoui

Cc: Markus Reyer; Amandeep Singh; Alexis Alvarez

Tue 2024-05-21 2:30 PM

As back-end lead, my chief contributions were:

- I helped establish the database architecture for the back end.
- I worked to connect the horizontal and vertical prototypes.
- I helped implement login and signup functionality.
- For both the M3 audit session and the second testing session, I
  - recorded feedback,
  - identified the actionable tasks,
  - organized it by section of the project, and
  - shared the result with the team.
- I edited milestone documents for accuracy, style, and clarity.
- I met regularly with teammates throughout the process.

Commit count: 11.

Main challenges:
The biggest challenge was the initial set-up of the project on the server. Once the server was up and functioning, things ran more smoothly. MySQL also presented some challenges with the latest version of MacOS.

Experience with Gen AI:
I asked ChatGPT to explain various error messages and found that it could be on a par with Stackoverflow for clarity and utility of information. I also experimented with asking ChatGPT to design an SQL database. As expected, the process was an iterative one, requiring multiple prompts to address inefficiencies or errors. Gen AI is useful to provide material to edit, to summarize material, or to save on repetitive typing, but having an attentive human to review the product is essential.

What I would do differently/other important things for the instructors:
I would make an effort to push more and smaller commits, focussing on incremental changes.

AS

Amandeep Singh

To: Aymane Arfaoui

Cc: Alexis Alvarez; Markus Reyer; Dave Daly

Reply Reply all Forward

Tue 2024-05-21 2:38 AM

a.) Contributions to Team Project and Teamwork (Front-End Lead)

- Set up the AWS Instance, the server, and the overall connection to the database.
- Responsible for UI design and overall theme (style and color) for Swiftsell.
- Worked closely with the back-end team to implement all required code for database interactions.
- Communicated with team members to fix both front-end and back-end issues.
- Regularly adjusted UI design based on team suggestions.
- Created base templates for files to ensure clarity in design.
- Provided instructions and debugged issues with teammates setting up the local virtual environment.
- Maintained professional interactions, enhancing team chemistry.

b.) Commit Count

- 110 commits across all development branches (excluding main).

c.) Main Challenges

- The main challenges were setting up the server, connecting it to the database, and implementing proper Flask routing for rendering the website. Being the sole front-end developer was demanding, as others had their own tasks, making front-end adjustments heavily dependent on me. Implementing the search function and overall design of the search page required significant coding adjustments.

d.) GenAI Experience

- Using GPT-4 was instrumental in the project. It assisted greatly with understanding Bootstrap and Flask syntax, debugging, and solving various issues. It was my go-to resource, saving time on front-end code involving long class names, IDs, buttons, and more.

e.)Improvements

- Improvements could be made in time management, as balancing front-end tasks with other classes often pushed implementations close to deadlines. More focus on overall page design, better spacing, and cleaning up CSS files would enhance clarity for teammates. Additionally, better task delegation and seeking help when necessary would improve team efficiency.

f.) Additional Notes

- Though I led the UI design, the team was always willing to help. Commit counts may appear low due to poor commit practices on my part, often committing large changes instead of smaller, incremental updates.

MR Markus Reyer

To: Markus Reyer

Cc: Alexis Alvarez; Dave Daly; Amandeep Singh; Aymane Arfaoui

Reply | Reply all | Forward

Tue 2024-05-21 11:31 AM

a.) Contributions to Team Project and Teamwork (GitHub Master)

- Setup the project on GitHub and assigned roles to teammates.
- Educating teammates on the functionality of GitHub including how to clone, pull and commit.
- Responsible for observing all commits to GitHub branches and testing project to ensure functionality.
- Observed GitHub commit messages ensuring enough detail was added for referencing and debugging.
- Setup branches for front end and back-end needs.
- Created UI for login page and assisted in feedback of designs of other pages related to the project.
- Attended team meetings outside of class and regularly met with teammates in class to stay up to date with project progress.

b.) Commit Count

- 25 commits across all branches.

c.) Main Challenges

- The main challenge that I faced was being unfamiliar with the Python programming language at the beginning of this project.
- Difficulty in setting up my virtual environment for testing due to being on Windows while teammates were all on Mac.

d.) GenAI Experience

- Use of ChatGPT was very useful in creating our design concepts as well as creating skeleton code for the login page. It also helped me debug setting up my virtual environment.

e.)Improvements

- I felt that my teammates were very instrumental in the success of this project. I feel that several of my teammates took on a larger workload to myself which led to time constraints that I feel that I could have helped in mitigating.

f.) Additional Notes

- My teammates were very helpful, and I learned a lot with their assistance. I feel grateful for having been part of such an incredible team as they maintained professionalism and were ready to push the project to the next level.

AA Alexis Alvarez

To: Markus Reyer

Cc: Dave Daly; Amandeep Singh; Aymane Arfaoui

Reply | Reply all | Forward

Tue 2024-05-21 12:39 PM

Self-Assessment and Contributions

a. Contributions to team project and teamwork (Back-End)
   - Helped set up AWS Instance and Apache server
   - Developed All Functional requirements and set up their priorities
   - Helped set up back-end
   - Implemented templates for message.html and about_alexisalvarez.html
   - Implemented several UI changes on post.html, message.html
   - Implemented search functionality that connects to database
   - Implemented post functionality with items and services which connects to database
   - Helped implement login and signup functionality
   - Provided inline comments for code readability on search and post functionality
   - Met up with team regularly and contributed to project planning process
b. Commit Count
   - 36 commits
c. One brief paragraph on main challenges they encountered in team project
   - The most challenging part of this project in my opinion was setting up the infrastructure. Setting up the AWS Instance, Apache server, MySQL Database Server, necessary libraries and packages, and connecting everything together proved to be a daunting challenge. However, with the help of my teammates (and ChatGPT), we were able to figure out and develop a plan to correctly set our backend up without any faults.
d. One brief paragraph of their own overall experience with GenAI
   - For this project, I would consider GenAI to be the sixth member of our group. GenAI proved to be an extremely valuable tool that allowed us to solve any technical issues with relative ease. Of course, sending prompts can only take you so far in the development lifecycle. As developers, we had to: 1. Know what exactly to ask. 2. Know how to word the prompt. 3. Read and comprehend the answer we received from GenAI. 4. Understand how to apply the response to our project and fix any issues we might observe from the initial response.
e. One brief paragraph on what would they do better next time based on what was learned in the class about SE management and processes
   - I believe one thing I would do better next time is to make more commits with smaller changes. I believe I could have improved upon better applying CI/CD in my software development process. I noticed after reviewing that some of my commits are too large and do not contain small, gradual changes. Another task I could improve upon would involve organizing our app routes into a cleaner setting. Separating back-end routes into their own files would improve readability and comprehension.
f. Anything else you deem important for instructors to know
   - One reason I did not make so many commits on the development branch was due to the fact that I may have focused too much on the features instead of integrating them into the development branch.

AA Aymane Arfaoui

To: Dave Daly

Cc: Markus Reyer; Amandeep Singh; Alexis Alvarez

Wed 2024-05-22 3:17 AM

a) Contributions to Team Project and Teamwork:

- Ensured the product advanced and all problems were dealt with rapidly and efficiently.
- Scheduled and led regular SCRUM meetings.
- Developed mockups and designed the UI for the sign-up, sign-in, post a service, and post an item pages.
- Coded and developed the sign-up, sign-in, post an item, and post a service page with functionality for switching tabs.
- Created personas and use cases to guide the design process
- Continuously communicated with team members to ensure their well-being and project alignment.
- Reviewed and wrote the milestones.
- Clarified project requirements with the instructor to ensure team understanding.
- Involved in brainstorming ideas, strategy and debugging code .
- Conducted code reviews to ensure quality and resolve issues.
- Adjusted milestones after reviewing feedback from the instructor.
- Facilitated real-time collaboration and task tracking using Discord.
- Briefly set up Trello boards for initial project management before switching to Discord.
- Ensured all team members were on point with their tasks and made sure to show up every time we needed to.

b) Number of Submissions to GitHub Team Dev. Branch:

29 total commits.

c) Main Challenges Encountered in Team Project:

As the team lead, one of the main challenges I encountered was ensuring clear communication and understanding of project requirement. For instance, when the instructor mentioned that there should be no chat implementation on the website, but users should be able to contact sellers, it was initially unclear. As team lead, I had to seek clarification from the instructor, which helped the team stay on track. As team lead, it is important to always know exactly what the requirements are and if not, always clarify with the teacher to not waste time on building something that is not part of what is requested.

Time management was sometimes challenging, especially around midterms and finals, since everyone was busy. We had to make a little effort to meet, and certain team members could not attend every meeting. However, those who could not attend made an effort to stay updated and catch up later on Discord. This approach ensured that everyone remained informed and could contribute effectively.

We also decided to transition to SCRUM style meetings to keep it short and make it easier for everyone to attend which worked really well.

Despite these challenges, the team worked exceptionally well together, with everyone completing their tasks and consistently showing up when needed.

d) Overall Experience with GenAI:

Using GenAI tools like ChatGPT was a valuable experience for the team. These tools provided significant assistance by suggesting best practices. Additionally, they guided us in our code review process and helped us to debug our code. The insights provided by GenAI tools streamlined our development process and improved our project's overall quality. ChatGPT assisted greatly with understanding Bootstrap and Flask syntax, debugging, and solving various issues. It was my go-to resource, saving time on front-end code involving long class names, IDs, buttons, and more. We learned to use ChatGPT effectively by giving it specific roles to get better results and tailored our prompts to maximize the tool's usefulness. The insights provided by GenAI tools streamlined our development process and improved our project's overall quality.

e) Improvements for Next Time:

Based on what I learned in the class about software engineering management and processes, I would try to explore other management tools. While our use of Discord was effective for real-time communication, integrating a more detailed project management tool could enhance our ability to monitor progress more carefully and plan way ahead of time the tasks that needs to be done and deadlines.

I learned to respect deadlines and make sure that all the P1 functions remain priority #1 even if exploring other ideas might be interesting. I learned the importance of use cases and personae in the software development and it helped everyone have the right mindset about how contribute to building the website.

f) Anything Else Important for Instructors to Know:

One notable aspect of our project management was the flexibility and adaptability of the team. This adaptability was crucial in maintaining efficient communication and collaboration throughout the project. The team's dedication and excellent cooperation ensured that we met our goals effectively and efficiently. The team really worked well together !