Final Project for SW Engineering Class CSC 648-848 Fall 2022

Team 4

GatorExchange

(WWW site for Buy/Sell/Share of Digital media exclusively for SFSU students and faculty)

Mahisha Patel	mpatel17@mail.sfsu.edu	*Team Lead*
Sudhanshu Kulkarni	skulkarni@sfsu.edu	*GitHub Master*
Sophia Chu	jliu36@mail.sfsu.edu	*Front-end Lead*
Jerry Liu	ebuddharuksa@sfsu.edu	*Back-end Lead*
Ekarat Buddharuksa	schu5@mail.sfsu.edu	
Ruben Ponce	rponce3@sfsu.edu	

URL of the demo: http://54.200.101.218/

December 17, 2022

Table Of Contents

Product Summary

- Name of the Product: **GatorExchange**

- Major Functions:
    1. Login: One can log in to GatorExchange Website to be able to message sellers or upload any media for sharing/selling.
    2. Registration: One can register to become a GatorExchange member to be able to post media and contact sellers.
    3. Browsing: One can browse through different pages from the navigation bar.
    4. Homepage: One can view recently approved media postings from the home page.
    5. Search: One can search the list of digital media posted by SFSU students and staff based on different categories and search queries.
    6. Post: One can post digital media for sharing or selling.
    7. Post Detail: One can view any digital media post details.
    8. Contact Seller: One can contact the seller by sending a message along with his contact details.
    9. Dashboard: One can view their own posted media on the dashboard.
    10. Inbox: One can see the list of messages received by other users.
    11. Admin Approval: The admin approves the posts using SQL Workbench to be seen by the other users.
    12. Google Analytics: The admin can view the analytics of the GatorExchange Website.
    13. About: One can see the brief details of the developers of GatorExchange.

- Uniqueness:
    1. GatorExchange is exclusive to SFSU; it allows searching from limited and most relevant materials posted by someone from the same community.
    2. The buyer can send a message to the seller for any digital media they are interested in.
    3. The Homepage will show the most recent ten posts for the latest updates.
    4. Developed backend API for admin to view the uploaded post on the browser.
    5. User automatically logs in after registration; no need to log in again.

- URL to the product: http://54.200.101.218/

SW Engineering CSC648/848 Fall 2022

GatorExchange

(WWW site for Buy/Sell/Share of Digital media exclusively for SFSU students and faculty)

Team 4

Mahisha Patel	mpatel17@mail.sfsu.edu	*Team Lead*
Sudhanshu Kulkarni	skulkarni@sfsu.edu	*GitHub Master*
Sophia Chu	jliu36@mail.sfsu.edu	*Front-end Lead*
Jerry Liu	ebuddharuksa@sfsu.edu	*Back-end Lead*
Ekarat Buddharuksa	schu5@mail.sfsu.edu	
Ruben Ponce	rponce3@sfsu.edu	

Milestone 1

October 8, 2022

Date submitted for review	Date revised
October 8, 2022	**October 17, 2022**

1. Executive Summary

Nowadays, in a digital era, we are well aware that most communication and collaboration occur through digital devices and primarily through media, whether formal or informal interaction. This inflated the existence and the use of digital media like digital photos, videos, audio, soft copy of documents, etc. As students, we empathize with how difficult it is to search for the relevant documents and files for our projects, assignments, presentations, and self-study. Not only for students, but sometimes the professors, instructors, and teaching assistants also require some document or file related to their subject. Searching for the required and relevant files from the whopping internet is time-consuming and tiresome, especially when we have to work with an approaching deadline; locating the supporting material is more burdensome than completing the work. So, we came up with the idea of building a platform called "GatorExchange" that provides an easy-to-share, sell, and buy interface for exchanging digital media among SFSU students and staff members.

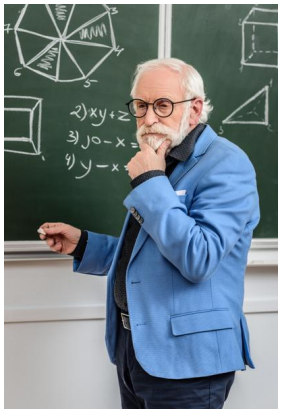
GatorExchange enables the students and staff members of the SFSU to look around for digital materials related to their interests without any login and sign-up process. It also allows them to upload their original media like handwritten notes, presentation, supplementary textbooks, musical sequence, etc. Moreover, these media can be shared freely or posted to sell. It works two ways, i.e., one can get an appreciation for their efforts, and the other can easily access the material either free of cost or with nominal pay. As our project is exclusive to SFSU, it allows searching from limited and most relevant materials posted by someone from the same community rather than futile efforts to explore the entire internet. Apart from that, the uniqueness involved in the project is the posts being associated with one or more tags that allow users to search by filtering tags.

We, as a team, are a group of students from Computer Science and Business major. Our vision is to build an interface that connects people with similar interests, hobbies, or majors to sell, buy and share digital media among SFSU. Also, we aim to design this application to be very user-friendly, enabling the people from SFSU to access the digital content easily.

2. Personae and main Use cases

PERSONAES:

- **Bob (A Professor at SFSU)**



- About Bob**:**
  - Very Busy
  - Wants a place to put media used in class
  - Not patient with technology
  - Has very Basic Internet skills
  - Prefers to use a computer; dislike using mobile apps
  - Has trouble seeing small text
  - Prefers reading and looking at diagrams instead of watching videos

Image Link (1)

  - Goals and Scenario: Decides to start posting his slides online for any student to use and look up even if they are not in his course currently.

- **Jane (A Student at SFSU)**

- About Jane**:**
  - Unemployed
  - Produces music in her free time
  - Able to use the Internet with no problem
  - Prefers mobile
  - Enjoys listening to music
  - Wants to sell her music

Image Link (2)

- Goals and Scenario: Decides to post some of her music. She wants a place to sell music to fellow SFSU students to make extra money and gain a following.

- **Josh (A Student at SFSU)**

- About Josh**:**
  - Works part-time
  - Biology major
  - Enjoys photography
  - Interests include exploring different kinds of plants and flowers
  - Sufficient computer skills
  - Enjoys exploring different kinds of photographs online
  - Wants to sell his photographs

Image Link (3)

- Goals and Scenario: Josh is Biology major at SFSU. He wants to earn some extra cash from his photography hobby. Josh needs a marketplace to sell his photographs online to interested buyers. He also wants to view and buy photographs and images of plants and flowers.

- **Ellie (A Teacher Assistant at SFSU)**

- About**:**
  - Helping teacher
  - Busy (Works, TA, and Student)
  - Has no Preference between mobile and desktop
  - Skilled with WWW
  - Very patient
  - Color Blind
  - Enjoys browsing through music and videos of his interests

Image Link (4)

- Goals and Scenario:
  - Ellie is a teacher assistant for a Physics Professor at SFSU. The professor tasked her to find videos, images, etc., for his course. Ellie wants to find these as she does not have time quickly.

- **Mark (An Employee at SFSU)**

Image Link (5)

- About:
  - Works at SFSU
  - Patient
  - Skilled with WWW
  - Enjoys browsing the internet
  - Knowledge of all SFSU guidelines and rules
  - Available at the same time the school is open

- Goals and Scenario:
  - Mark is employed by SFSU and tasked to be the admin of the app. He is trying to keep GatorExchange a safe place and ensure that there are appropriate posts and that users follow guidelines.

USE CASES:

1. **Post: Bob,** a CS professor at SFSU and a **guest user, previews** the website, decides to **register,** and goes through that process. He then wants to **post** slides from his class. When posting, he then gives his **post** a tag to his class, subject, and type of media. It then informs him that the **post** will be **reviewed**.

2. **Post: Jane**, an unemployed student at SFSU and a **guest user,** then goes and tries to **post** her music. She goes through the process of creating a **post**. When trying to **post,** she is prompted to **register**, goes through the **registration** process, and finally can **post** her music with **post tags** and make it **sell**. She is then informed that her **post** will be under **review** and take up to 24 hours to be **posted.**

3. **Buy: Josh**, a Biology major at SFSU, is a **registered** user who decides to use the **search** to find other photographs that other registered users have posted. He then finds a post he likes and decides to **buy**. He then messages the seller with the contact details to further communicate outside the website. If both agree, they will use a 3rd party app to complete the payment transaction. He then can **rate** the post he bought.

4. **Search: Ellie**, a teacher assistant at SFSU, decides to use the **search** and **filters** the search to narrow the field. She then **previews** the results as a **guest user**. She then finds some media her teacher needs and goes through the process of getting the media. This

prompts her to **register** to get the media. Now that she is a **registered user,** she finds some media she **bookmarks** to show the professor later.

5. **Approving post: Mark**, an employee at SFSU, as an administrator, decides to go in and view **posts** that **registered users** want to **post**. He then decides whether the **posts** are appropriate and then **approves** the post. If a **post** does not follow the guidelines, it will **not be approved,** and he may **ban** the user.

---

3. List of main data items and entities – data glossary/description

3.1 User:

The user can be of 3 types, as listed below.

- Admin: Admin can approve the post of the registered user to be displayed on the web.
- Registered user: Registered users can buy, sell and share the media.
- Guest user: Guest users can search but must register to buy, sell or share the media.

3.2 Post: The posted items have 1 or more tags associated with them and can be bought, sold, or shared exclusively with the people of SFSU.

The post can be of multiple types as shown below:

- Audio
- Video
- Image
- Document

3.3 Tags/Categories:

The tags can be attached to each post, making it easier for the user to search.

- Lecture Notes
- Presentation
- Lecture Recording
- Music
- Poetry
- Art
- Computer Science
- Philosophy
- Sports
- Photography
- Travel
- Gaming

- Food
- Technology

3.4 Transaction Record:
The transaction records the information when a buyer can send a message to the seller but without any actual payment involved. The registered user can only send the message or receive the message.
- Buyer
- Seller
- Message
- Status

---

4. The initial list of functional requirements

1. Guest users

1.1. Shall be allowed to register for an account.

1.2. Shall be able to search content by using specific tags.

1.3. Shall be able to preview media.

2. Registered user

2.1. Shall have all the functional requirements of the Guest user.

2.2. Shall be allowed to create a post with uploaded media.

2.3. Shall be allowed to message sellers to inquire about listed content.

2.4. Shall be allowed to bookmark posted content.

2.5. Shall be allowed to rate posted content.

2.6. Shall see notifications when a buyer is interested in posted content.

2.7. Shall be able to download purchased content after approval from the seller.

2.8. Shall be allowed to follow other users to get notified of new posts.

3. Administrator

3.1. Shall have all functional requirements of Registered users.

3.2. Shall be able to approve or reject user-submitted media for sharing and marketplace.

3.3. Shall be able to delete posts that break community guidelines.

3.4. Shall be able to ban user accounts.

3.5. Shall be able to view the website's analytics, showing overall users, interactions, and the number of posts.

---

5. List of non-functional requirements

- The application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0
- The application shall be optimized for standard desktop/laptop browsers, e.g., it must render correctly on the two latest versions of two major browsers
- All or selected application functions must render well on mobile devices
- Data shall be stored in the database on the team’s deployment server.
- No more than 50 concurrent users shall access the application at any time
- Privacy of users shall be protected
- The language used shall be English (no localization needed)
- The application shall be straightforward to use, and intuitive
- The application should follow established architecture patterns
- Application code, and its repository shall be easy to inspect and maintain
- Google Analytics shall be used
- No email 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

- Pay functionality, if any (e.g., paying for goods and services), shall not be implemented nor simulated in UI.
- Site security: basic best practices shall be applied (as covered in the class) for main data items
- Media formats shall be standard as used in the market today
- Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
- The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2022. For Demonstration Only" at the top of the WWW page nav bar. (Important to not confuse this with a real application).

---

6. Competitive analysis

Feature	Pexels	YouTube	Fiverr	GatorExchange
Text Search	+	+	+	+
Media Marketplace	+	-	++	++
Media Share	-	+	-	++
Preview Media	+	+	-	+
SFSU Exclusivity	-	-	-	++
Tag Media Based on Community Groups	-	-	-	++
Buy and Sell Media	+	-	+	++

GatorExchange shall bridge the gap between a media marketplace and a media sharing website. Only persons with an SFSU email will be able to register for an account. By being exclusive to SFSU students and faculty, our product aims to forge new communities through the sharing of various forms of media, as well as give SFSU content creators a space to build a foundation of support through personalized shops. We shall further magnify this aspect by

allowing users to tag and search for media by community groups such as class, majors, or interests. This shall create connections among users with similar interests and users in similar fields of study.

---

7. High-level system architecture and technologies used

Server Host: AWS EC2
OS: Ubuntu 18.04

Database: MySQL 8.0

Web Server: Nginx 1.22.0

Server-side Language: Python

Additional Technologies

- Web Framework: Flask, React, Node, React Native
- IDE: VS Code/PyCharm
- Web Analytics: Google Analytics

---

8. Team and roles

Team Lead:	Mahisha Patel
Frontend Lead:	Sophia Chu
Frontend Team Members:	Mahisha Patel, Ruben Ponce
Backend Lead:	Jerry Liu
Backend Team Members:	Sudhanshu Kulkarni, Ekarat Buddharuksa
Database Master:	Sudhanshu Kulkarni
GitHub Master:	Sudhanshu Kulkarni
Document Editor:	Ruben Ponce

---

9. Checklist

- So far, all team members are engaged and attending ZOOM sessions when required
  - **DONE**
- Team found a time slot to meet outside of the class
  - **DONE**
- Back end, Front end leads and Github master chosen
  - **DONE**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
  - **DONE**
- Team lead ensured that all team members read the final M1 and agree/understand it before submission

    - **DONE**
- Github organized as discussed in class (e.g., master branch, development branch, a folder for milestone documents, etc.)

  DONE

SW Engineering CSC648/848 Fall 2022

GatorExchange

(WWW site for Buy/Sell/Share of Digital media exclusively for SFSU students and faculty)

Team 4

Mahisha Patel	mpatel17@mail.sfsu.edu	*Team Lead*
Sudhanshu Kulkarni	skulkarni@sfsu.edu	*GitHub Master*
Sophia Chu	jliu36@mail.sfsu.edu	*Front-end Lead*
Jerry Liu	ebuddharuksa@sfsu.edu	*Back-end Lead*
Ekarat Buddharuksa	schu5@mail.sfsu.edu	
Ruben Ponce	rponce3@sfsu.edu	

Milestone 2

October 29, 2022

Date submitted for review	Date revised
October 29, 2022	**November 6, 2022**

1. Executive Summary

Nowadays, in a digital era, we are well aware that most communication and collaboration occur through digital devices and primarily through media, whether formal or informal interaction. This inflated the existence and the use of digital media like digital photos, videos, audio, soft copy of documents, etc. As students, we empathize with how difficult it is for us to search for the relevant documents and files for our projects, assignments, presentations, and self-study. Not only for students, but sometimes the professors, instructors, and teaching assistants also require some document or file related to their subject. Searching for the required and relevant files from the whopping internet is a time-consuming and tiresome process, especially when we have to work with an approaching deadline; locating the supporting material is more burdensome than completing the work. So, we came up with the idea of building a platform called "GatorExchange" that provides an easy-to-share, sell, and buy interface for exchanging digital media among SFSU students and staff members.

GatorExchange enables the students and staff members of the SFSU to share digital materials related to their interests without any login and sign-up process. It also allows them to upload their original media like handwritten notes, presentation, supplementary textbooks, musical sequence, etc. Moreover, these media can be shared freely or posted to sell. It works two ways, i.e., one can get an appreciation for their efforts, and the other can easily access the material either free of cost or with nominal pay. As our project is exclusive to SFSU, it allows searching from limited and most relevant materials posted by someone from the same community rather than futile efforts to explore the entire internet. Apart from that, the uniqueness involved in the project is the posts being associated with categories that allow users to search by filtering categories.

We, as a team, are a group of students from Computer Science and Business major. Our vision is to build an interface that connects people with similar interests, hobbies, or majors to sell, buy and share digital media among SFSU. Also, we aim to design this application to be very user-friendly, enabling the people from SFSU to access the digital content easily.

2. List of main data items and entities

2.1 User:

The user can be of 3 types, as listed below.

- Admin: Admin can approve the post of the registered user to b displayed on the web.
- Registered user: Registered users can buy, sell and share the media.
- Guest user: Guest users can search but must register to buy, sell or share the media. The user who is not registered in the database is guest users.

- **id**: Unique integer id of each user
- **first_name**: First name of the user
- **last_name**: Last name of the user
- **email**: Email will be used as the username of the user
- **password**: Password of the user *(encrypted)*
- **preferences:** Preference of the user such as interests or major

2.2 Post:

The post contains the data that is included in a post done by the user. It will have all the information shown in the post, including the user who posted, the type of the post, the tile, the description, the price, and the category.

- **post_id**: Unique integer id of each post
- **uploader_id**: User who posted (Foreign key from the User table)
- **post_type**: Type of the post (image/video/audio/document)
- **title**: Title of the post
- **file**: Media file of the post
- **description**: Description of the post
- **price**: Price of the post (0 if Free)
- **approved**: Post approval flag (True if approved by admin otherwise False)
- **category**: Category of the post (Foreign key from the Category Table)
- **created_timestamp**: Post time

2.3 Category:

Each Post is attached with some category. This category will be the foreign key in the Post table. Some of the examples of categories are:

- Presentation
- Lecture Recording
- Music
- Poetry

- Art
- Computer Science
- Software Engineering
- Philosophy
- Sports
- Photography
- Travel
- Gaming
- Food
- Technology

- **category_name**: Name of the category as listed above

2.4 Message:

This table will store the information about who sends a message to whom and for which post, it does not include any payment activity.

- **message_id**: Unique integer id of each transaction
- **post_id**: Post for which transaction occurred (Foreign key from the Post table)
- **buyer**: User id of the user who sends a message (Foreign key from the User table)
- **seller**: User id of the user who receives the message and owner of the post (Foreign key from the User table)
- **message**: Message content send by the buyer to seller
- **status**: Status of the activity
- **timestamp**: Timestamp when the message is sent

3. Functional Requirements

Priority 1:

Guest users

1.1. Shall be allowed to register for an account: Guest user needs to register to buy, sell, or share any digital media.

1.2. Shall be able to search: Guest users can still search for the media without registration.

1.4: Shall be able to browse.

1.5: Shall be able to view item details.

Registered users

2.1. Inherit all the functional requirements of the Guest user.

2.2. Shall be able to create a post with uploaded media: Can create a post for selling/sharing digital media along with the title, description, and category.

2.3. Shall be allowed to message sellers to inquire about listed content: Can send the message to the seller for sharing contact details to buy the digital media

2.4. Shall be allowed to bookmark posted content: Can save the post for future reference

2.9. Shall be able to view their posts using Dashboard.

2.10. Shall be able to view their messages using Dashboard.

Administrator

3.1. Inherit all functional requirements of Registered users.

3.2. Shall be required to approve or reject user-submitted media for sharing and marketplace.

3.3. Shall be able to delete posts that break community guidelines.

3.5. Shall be able to view the website's analytics, showing overall users, interactions, and the number of posts.

3.6 Shall be able to remove users.

Priority 2:

Registered user

2.6. Shall see notifications when a buyer is interested in posted content.

Priority 3:

Registered user

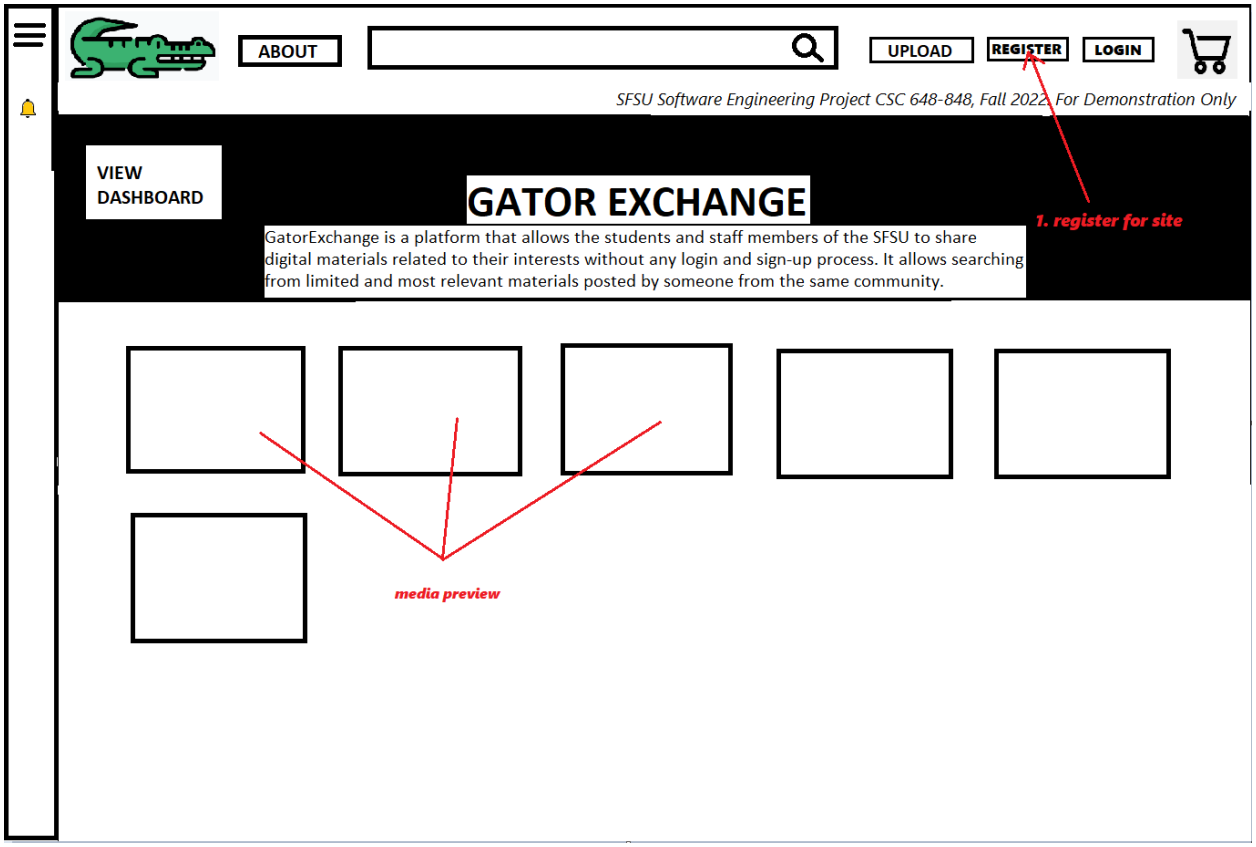
2.5. Shall be allowed to rate posted content

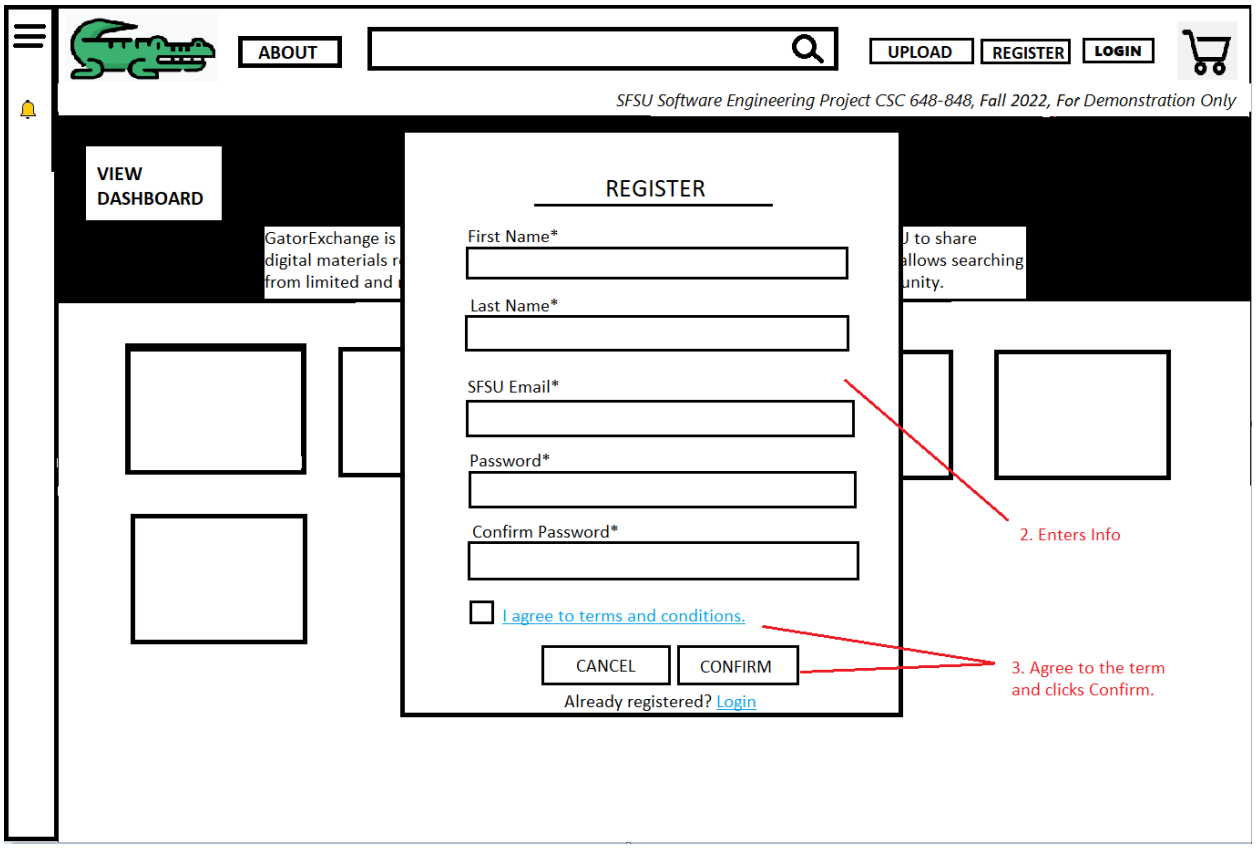
2.8. Shall be allowed to follow other users to get notified of new posts.

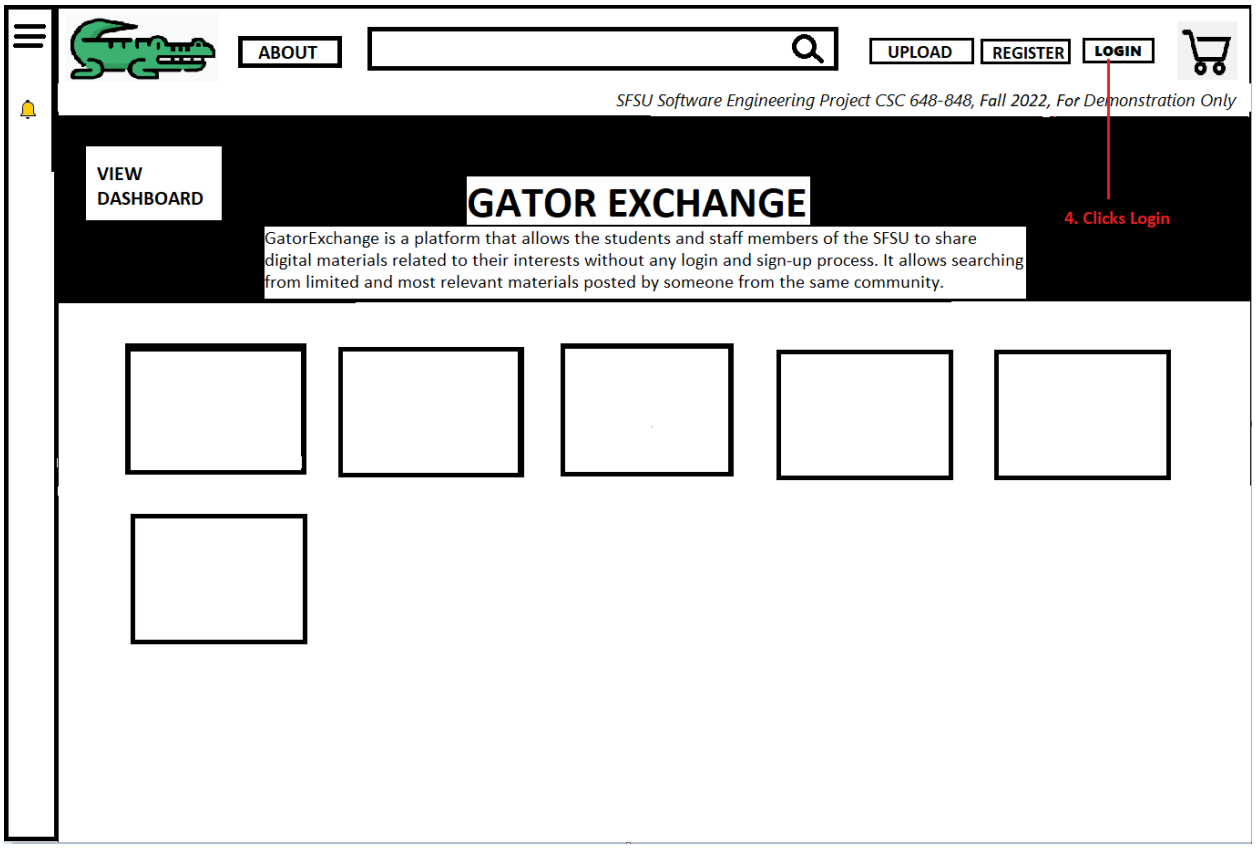
4. UI Storyboards for each main use case (low-fidelity B&W wire diagrams only)

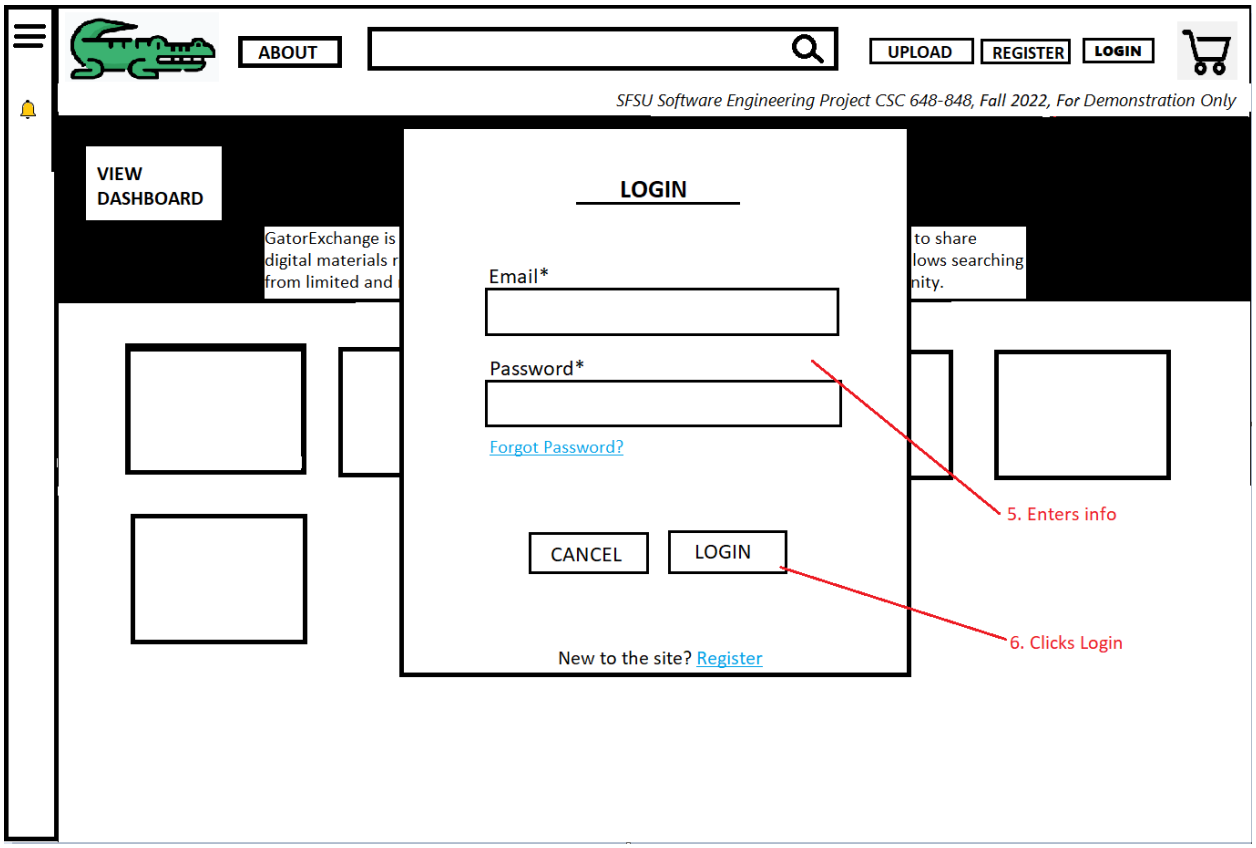
USE CASES:

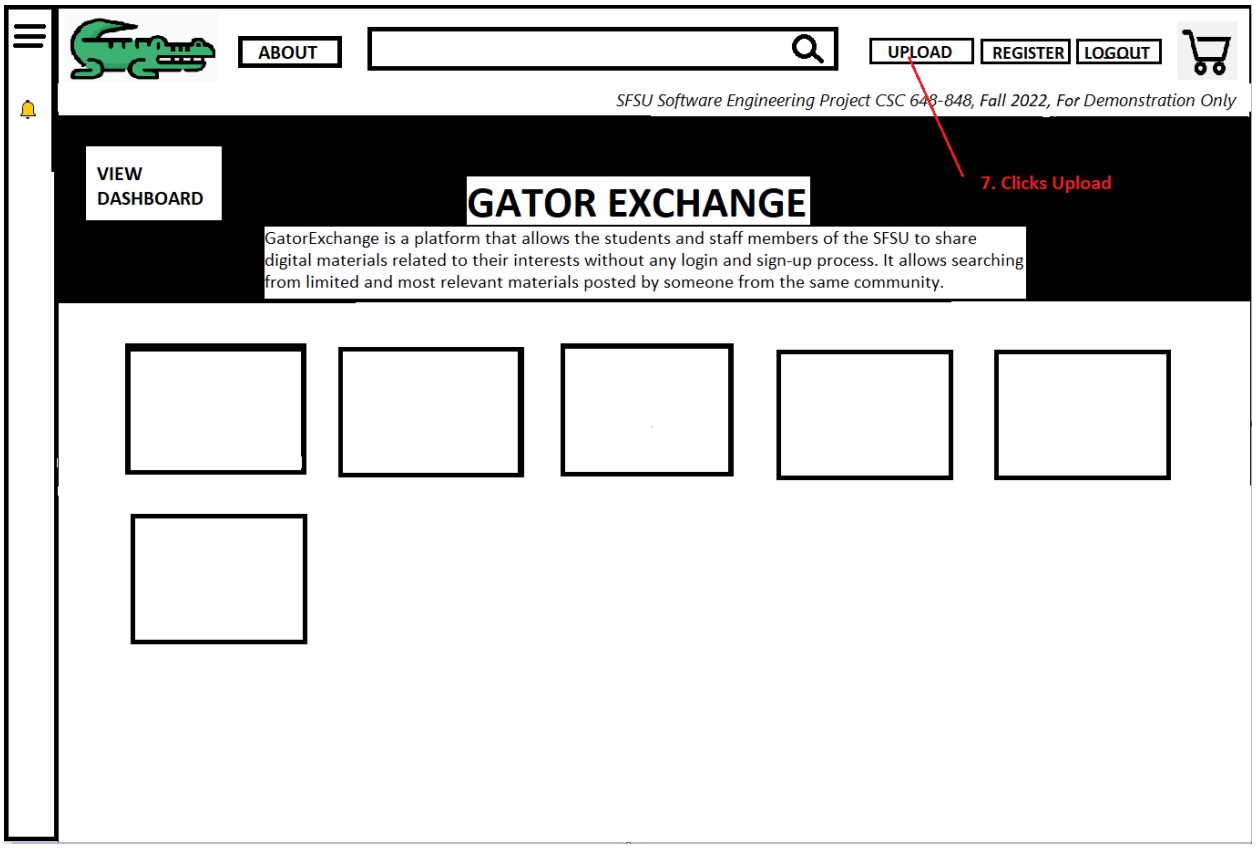
1. **Post:** A guest user, previews the website, decides to register, and goes through that process. He then login and creates a post. It then informs him that the post will be reviewed soon by the admin and will be available on the website.

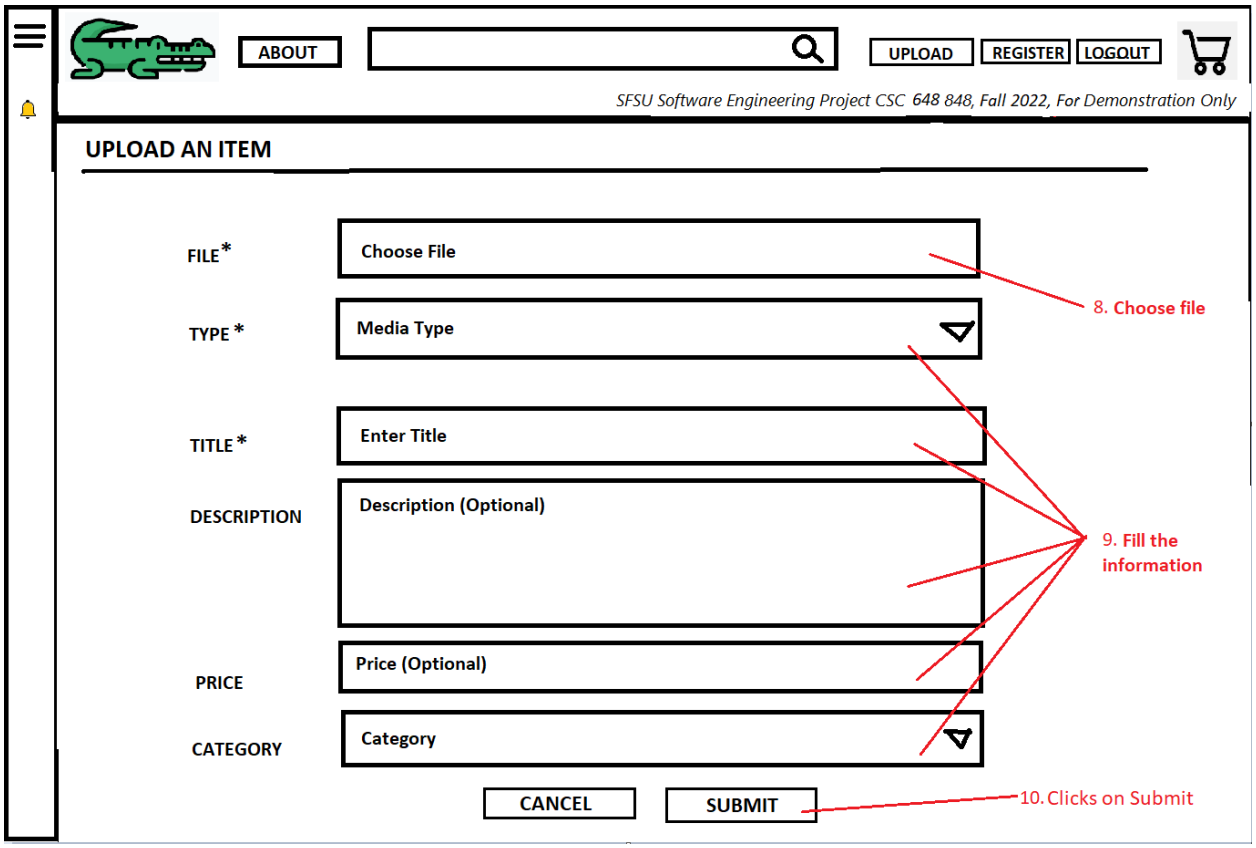


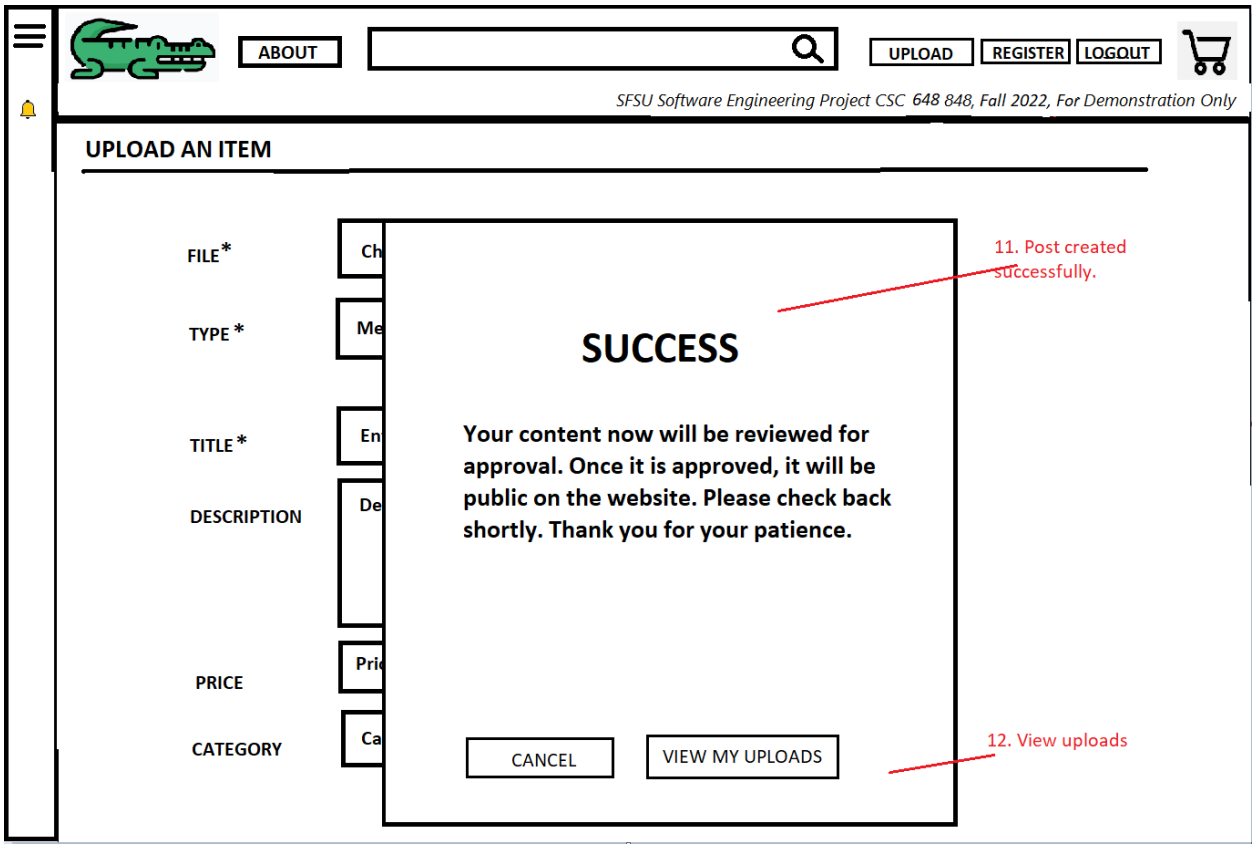

ABOUT
UPLOAD
REGISTER
LOGIN
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
VIEW DASHBOARD
REGISTER
First Name*
Last Name*
SFSU Email*
Password*
Confirm Password*
I agree to terms and conditions.
CANCEL
CONFIRM
Already registered? Login
2. Enters Info
3. Agree to the term and clicks Confirm.

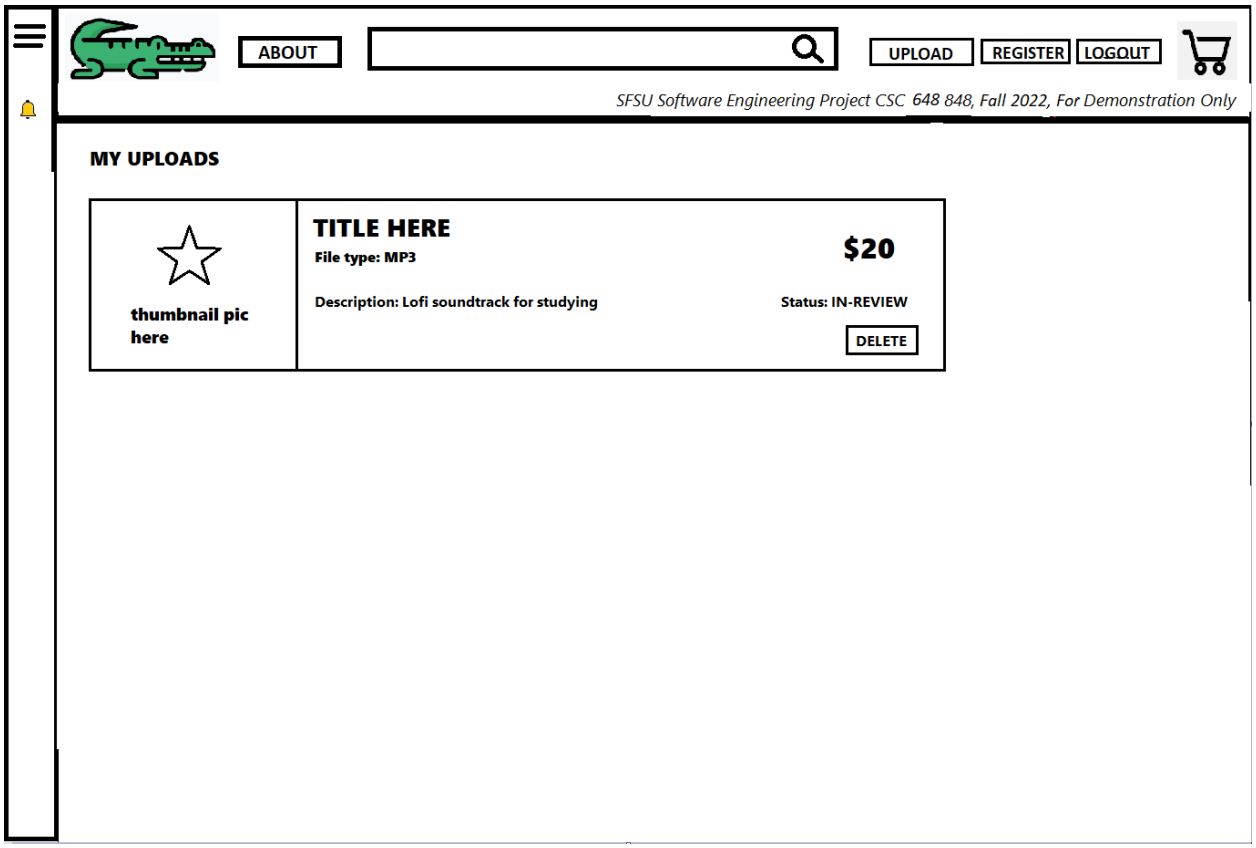

ABOUT
UPLOAD
REGISTER
LOGIN
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
VIEW DASHBOARD
GATOR EXCHANGE
GatorExchange is a platform that allows the students and staff members of the SFSU to share digital materials related to their interests without any login and sign-up process. It allows searching from limited and most relevant materials posted by someone from the same community.
4. Clicks Login


ABOUT
UPLOAD
REGISTER
LOGIN
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
VIEW DASHBOARD
LOGIN
Email*
Password*
Forgot Password?
CANCEL
LOGIN
New to the site? Register
5. Enters info
6. Clicks Login


ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
VIEW DASHBOARD
GATOR EXCHANGE
GatorExchange is a platform that allows the students and staff members of the SFSU to share digital materials related to their interests without any login and sign-up process. It allows searching from limited and most relevant materials posted by someone from the same community.
7. Clicks Upload

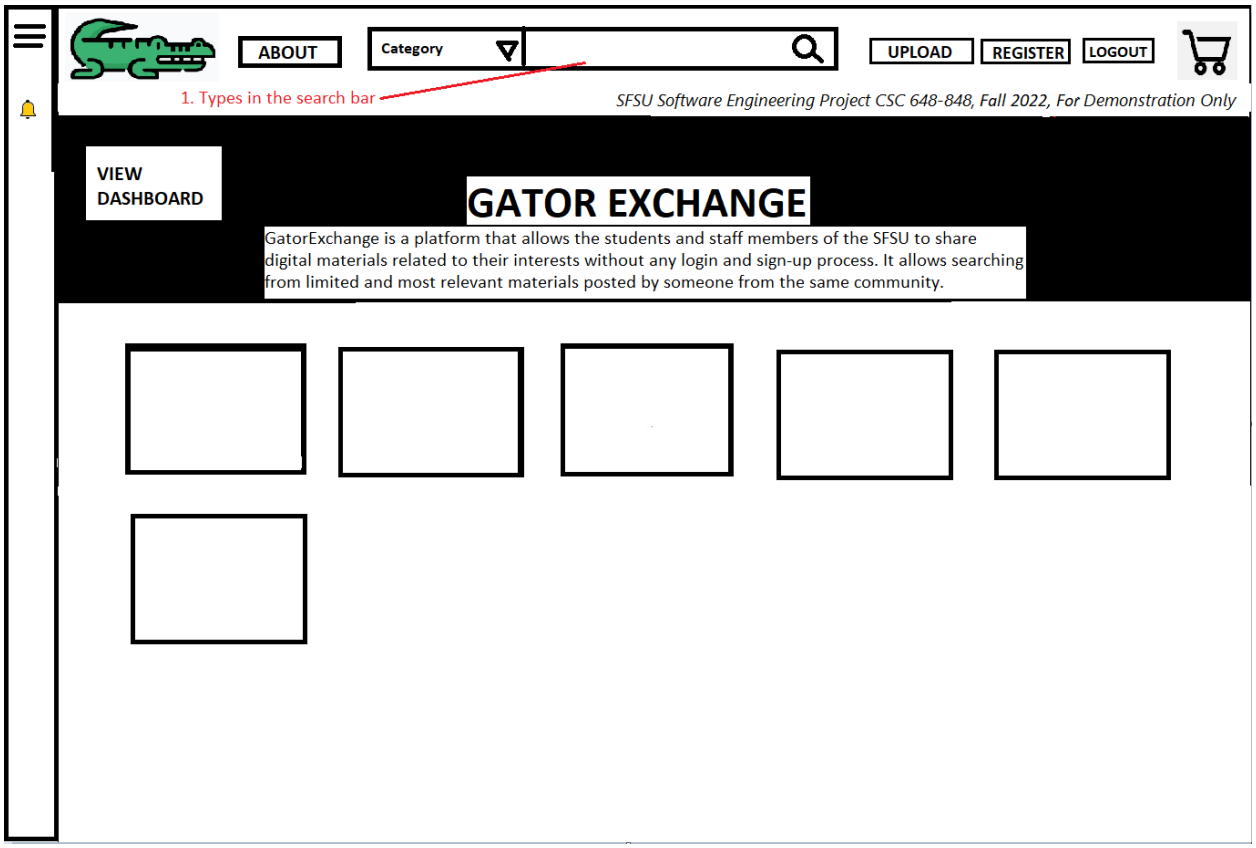

ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648 848, Fall 2022, For Demonstration Only
UPLOAD AN ITEM
FILE*
Choose File
TYPE *
Media Type
TITLE *
Enter Title
DESCRIPTION
Description (Optional)
PRICE
Price (Optional)
CATEGORY
Category
CANCEL
SUBMIT
8. Choose file
9. Fill the information
10. Clicks on Submit

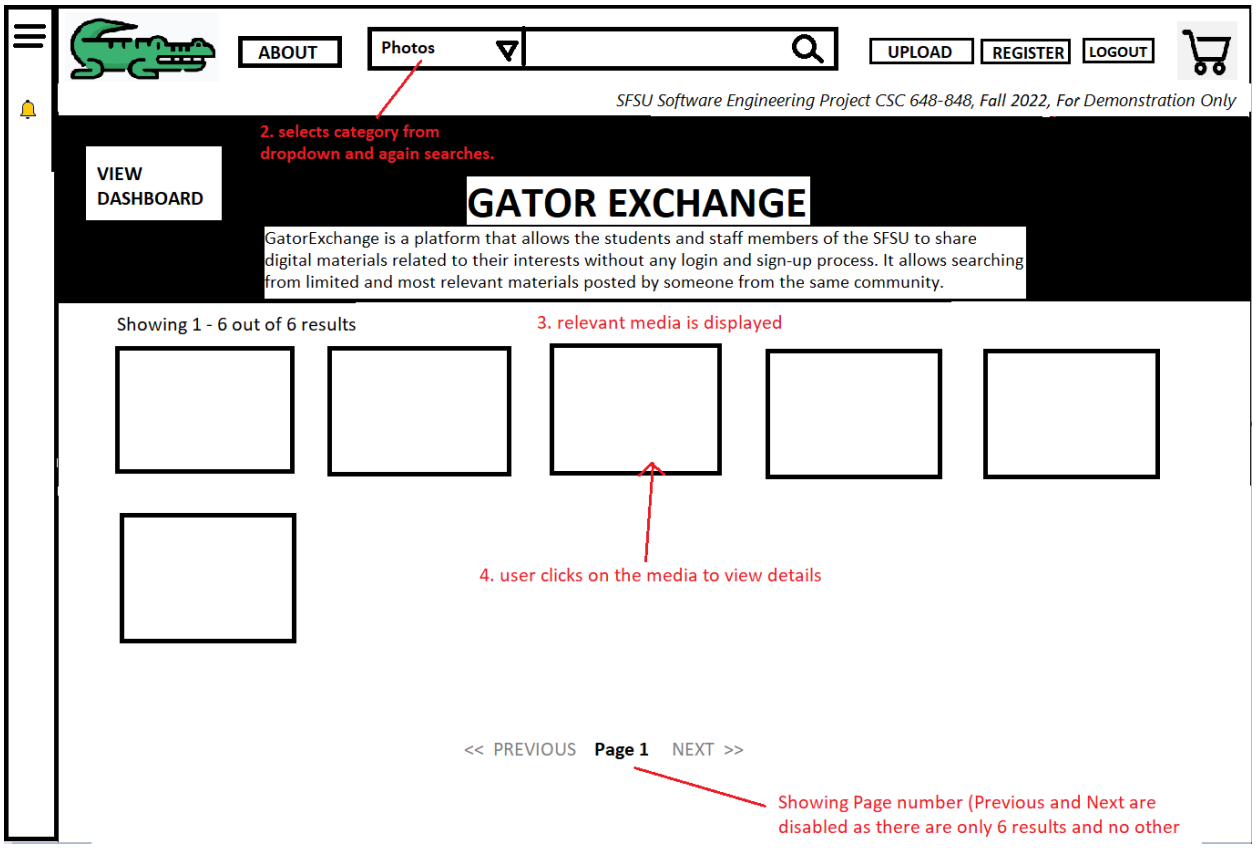

ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648 848, Fall 2022, For Demonstration Only
UPLOAD AN ITEM
FILE*
TYPE *
TITLE *
DESCRIPTION
PRICE
CATEGORY
SUCCESS
Your content now will be reviewed for approval. Once it is approved, it will be public on the website. Please check back shortly. Thank you for your patience.
CANCEL
VIEW MY UPLOADS
11. Post created successfully.
12. View uploads

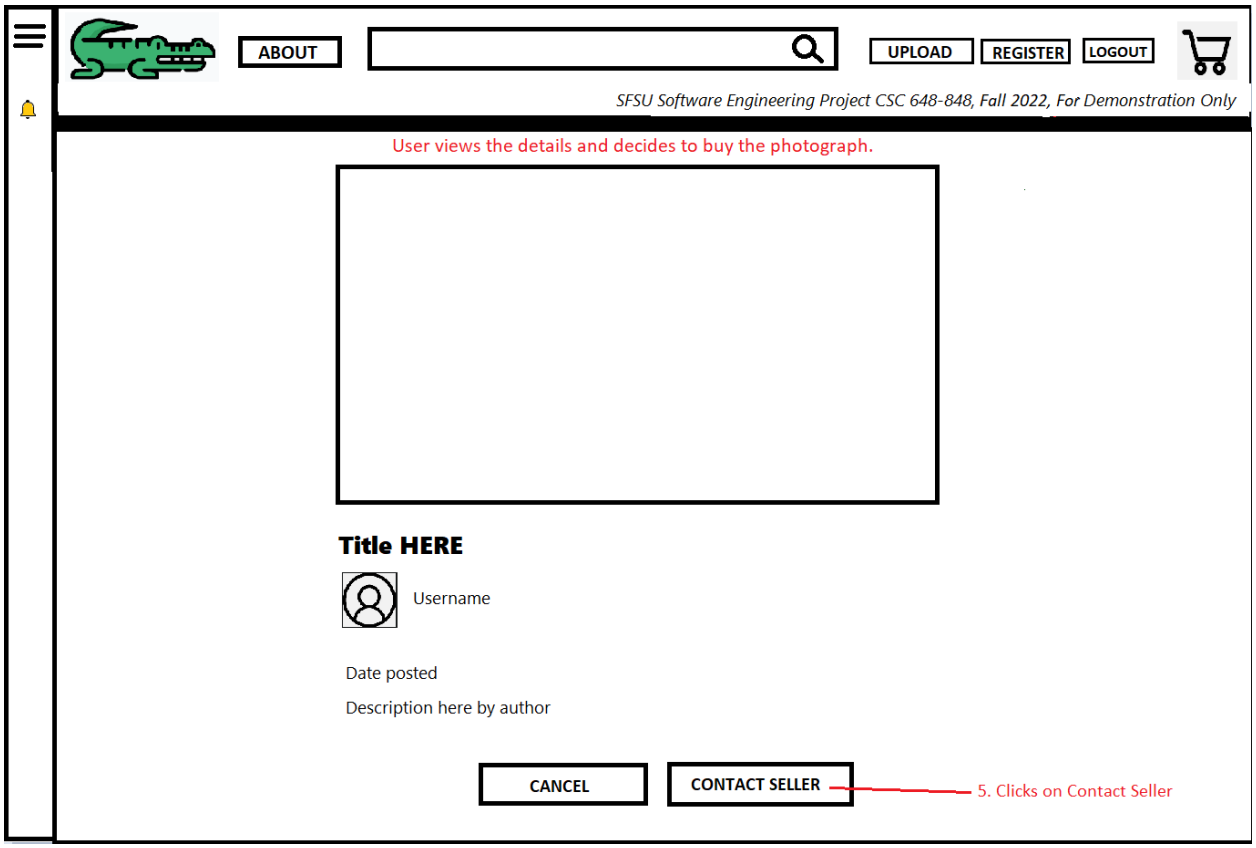

ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648 848, Fall 2022, For Demonstration Only
MY UPLOADS
thumbnail pic here
TITLE HERE
File type: MP3
$20
Description: Lofi soundtrack for studying
Status: IN-REVIEW
DELETE

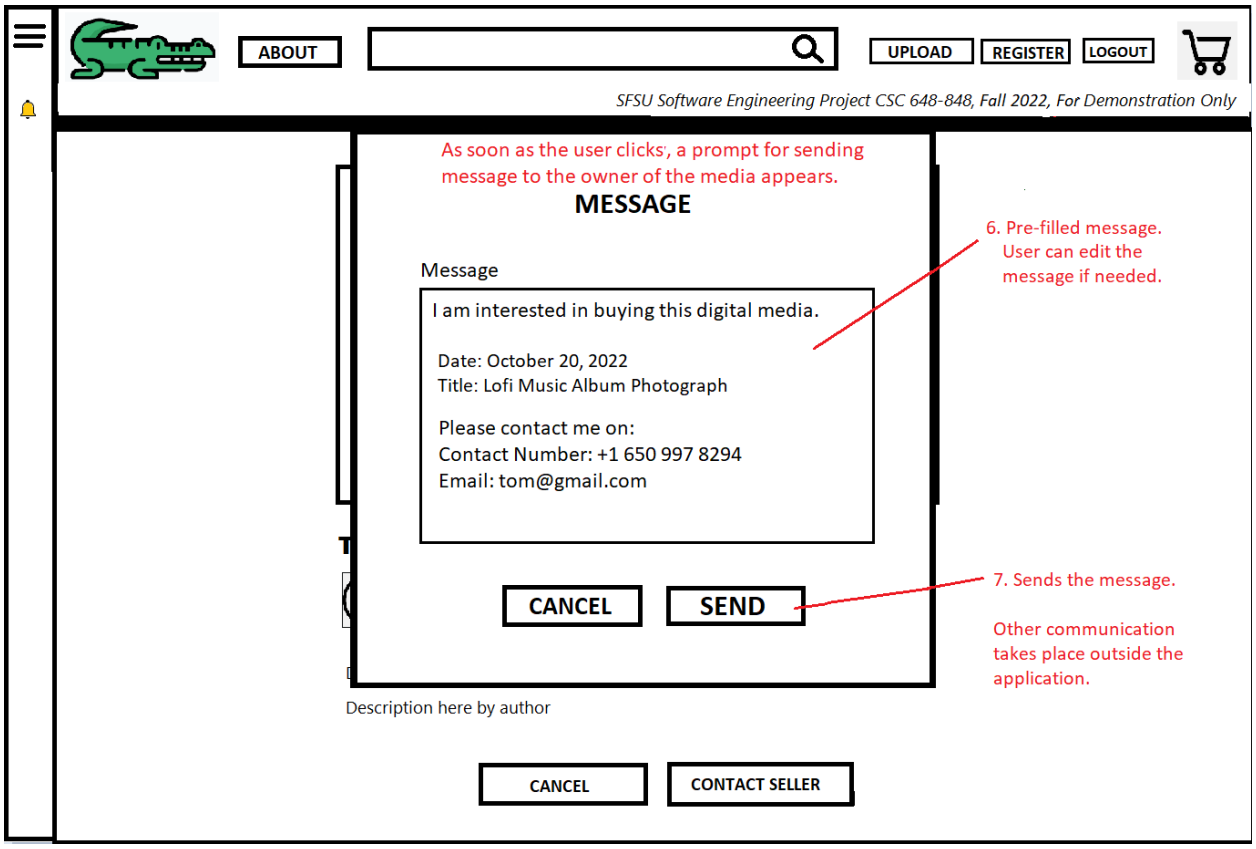
2. **Search and Contact Seller:** A registered user decides to use the search to find photographs that have been posted by other registered users. He then finds a post he likes and decides to contact the seller. He then messages the seller with the contact details so they can further communicate outside of the website. If both agree, they will use a 3rd party app to complete the payment transaction.

A user is already registered and logged in to his account.

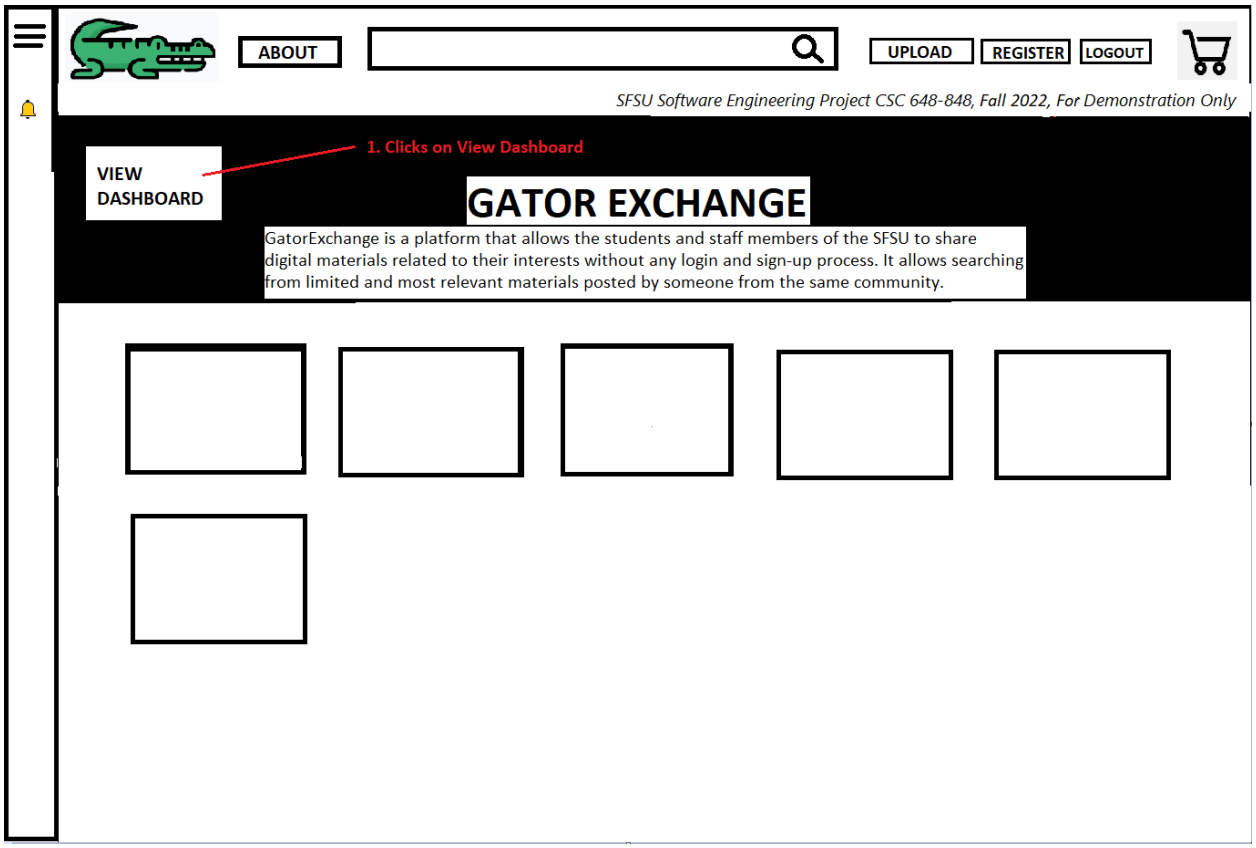


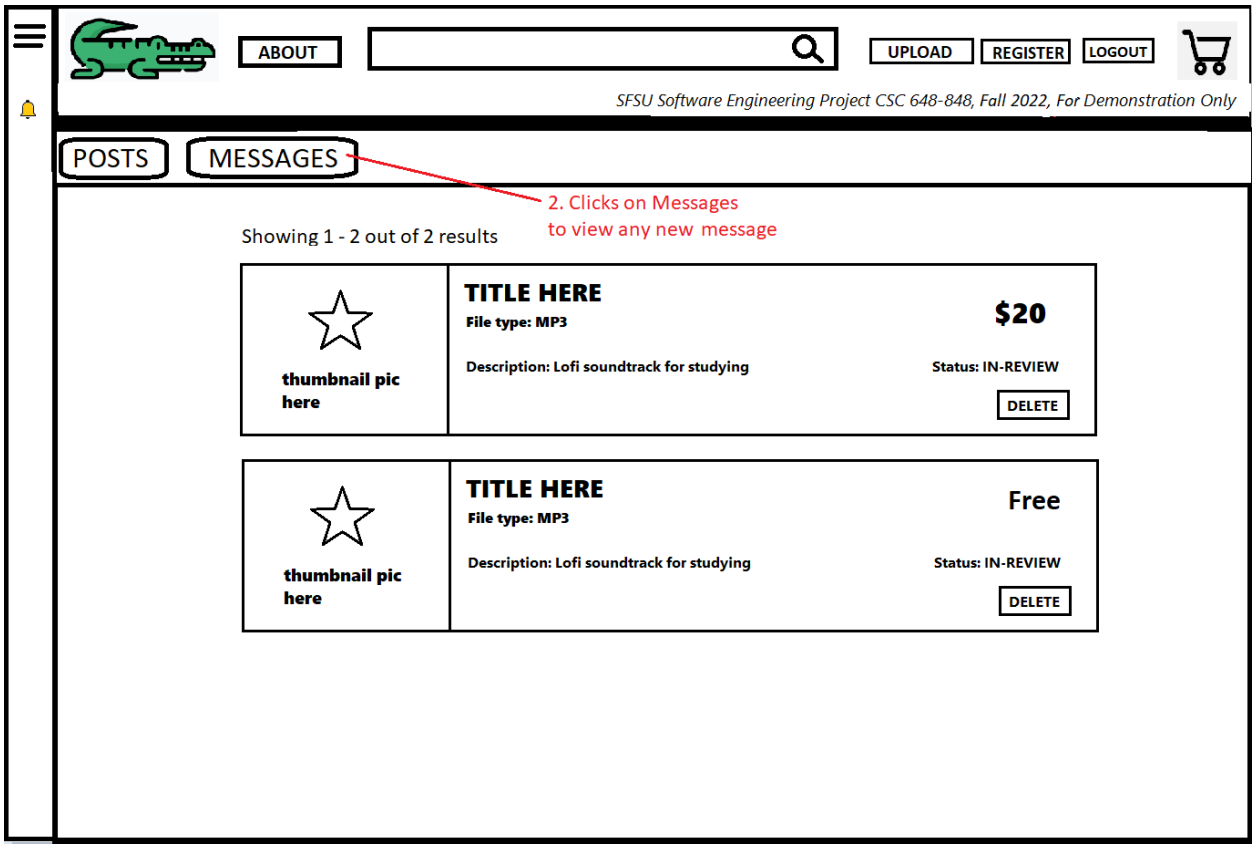

ABOUT
Photos
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
2. selects category from dropdown and again searches.
VIEW DASHBOARD
GATOR EXCHANGE
GatorExchange is a platform that allows the students and staff members of the SFSU to share digital materials related to their interests without any login and sign-up process. It allows searching from limited and most relevant materials posted by someone from the same community.
Showing 1 - 6 out of 6 results
3. relevant media is displayed
4. user clicks on the media to view details
<< PREVIOUS
Page 1
NEXT >>
Showing Page number (Previous and Next are disabled as there are only 6 results and no other

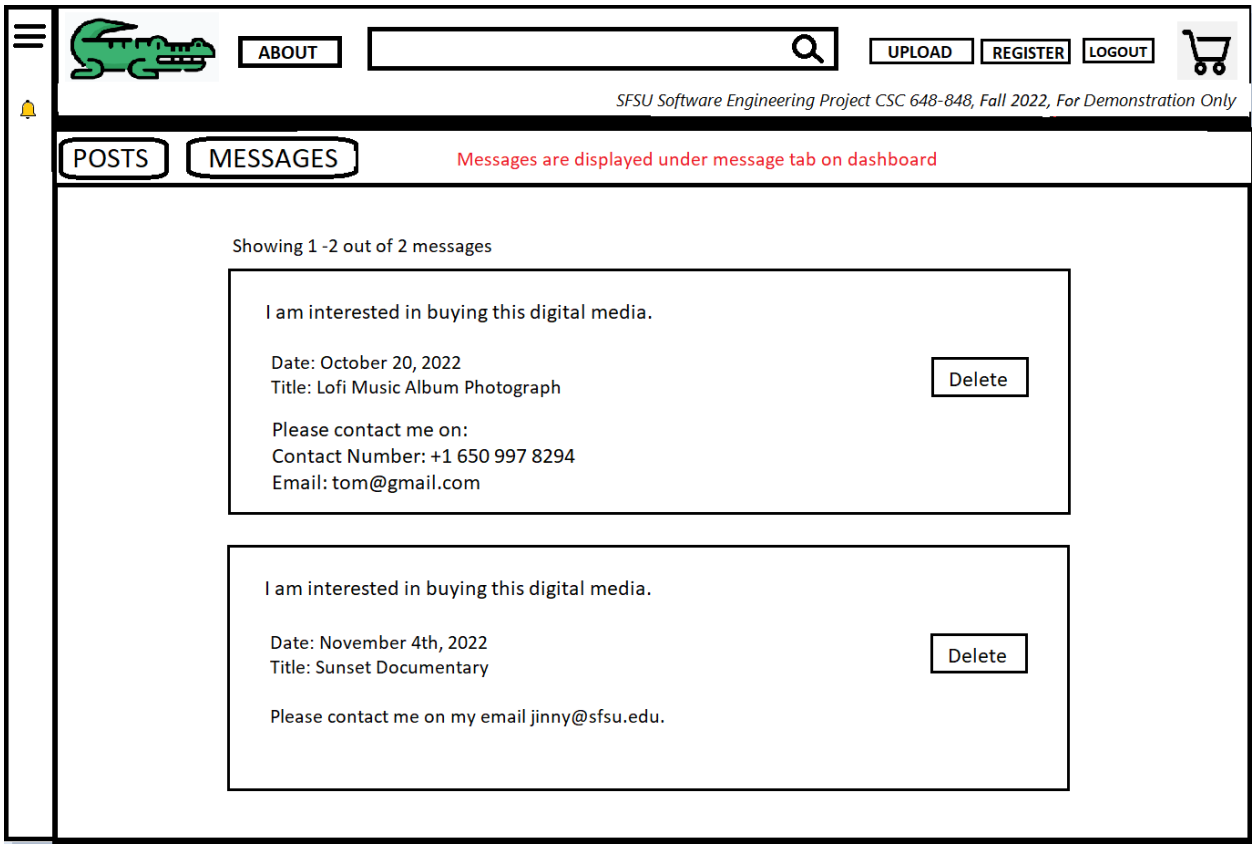

ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
User views the details and decides to buy the photograph.
Title HERE
Username
Date posted
Description here by author
CANCEL
CONTACT SELLER
5. Clicks on Contact Seller



3. **View Messages:** A user who posted his digital media for sale, receives a message. He then views the message on the dashboard and notes the contact details sent by the interested buyer.


ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
VIEW DASHBOARD
1. Clicks on View Dashboard
GATOR EXCHANGE
GatorExchange is a platform that allows the students and staff members of the SFSU to share digital materials related to their interests without any login and sign-up process. It allows searching from limited and most relevant materials posted by someone from the same community.


ABOUT
UPLOAD
REGISTER
LOGOUT
SFSU Software Engineering Project CSC 648-848, Fall 2022, For Demonstration Only
POSTS
MESSAGES
2. Clicks on Messages to view any new message
Showing 1 - 2 out of 2 results
thumbnail pic here
TITLE HERE
File type: MP3
$20
Description: Lofi soundtrack for studying
Status: IN-REVIEW
DELETE
thumbnail pic here
TITLE HERE
File type: MP3
Free
Description: Lofi soundtrack for studying
Status: IN-REVIEW
DELETE



4. **Approving post:** The admin approves the post by going to the workbench and editing the field approved in Post Table after the post is created by the user. The post won't be displayed on the website until the admin approves it.

   The admin can also delete any post or user if any rules or regulations are being violated. That banned user won't be able to access the website anymore.

5. High-level Architecture, Database Organization summary only

1. **DB Organization**

   User:

   - **id**: INT
   - **first_name**: VARCHAR
   - **last_name**: VARCHAR
   - **email**: VARCHAR
   - **password**: VARCHAR
   - **preferences**: VARCHAR

Post:

- **post_id**: INT
- **uploader_id**: INT (Foreign key from the User Table)
- **post_type**: VARCHAR
- **title**: VARCHAR(45)
- **file**: VARCHAR (255)
- **description**: VARCHAR(255)
- **price**: FLOAT
- **approved**: VARCHAR
- **category**: VARCHAR (Foreign key from the Category Table)
- **created_timestamp**: DATETIME

Category:

- **category_name**: VARCHAR

Message:

- **message_id**: INT
- **post_id**: INT (Foreign key from the Post Table)
- **buyer**: INT (Foreign key from the User Table)
- **seller**: INT (Foreign key from the User Table)
- **message**: VARCHAR(255)
- **status**: VARCHAR
- **timestamp**: DATETIME

2. **Media Storage**
   - Our application will be storing different types of media files such as documents, videos, images, and audio files on the server. The location (file path) of the media will be stored in the database.

3. **Search/filter architecture and implementation**
   - We will be using the '*%like'* feature provided by SQL, which will search on the category and title.
   - Also items will also be organized by categories. Our search function will search posts by categories. If the post has the requested category, all posts with that category will be shown in the results.

4. **API**
   - login(email, password): *create a session entry for the user*
   - register(details): *get all details and save a new user + login*

- home(user): *get main screen data (relevant posts) for the logged-in user/guest user.*
- search_posts(keywords): *search posts and categories based on keywords entered.*
- get_my_posts(user): *get all the posts posted by and bought by the logged-in user.*
- upload_post(details): *get all details and save the post.*
- get_post_details(post): *get all the details of a post.*
- send_message(buyer, seller, post): *send a message to the buyer from the seller about a post.*

6 . Risk

- Skill risks:

The skill risk that might occur is that not every member of the team is well aware of the technologies and tools that we are going to use. (Python, Reactjs, SQL, Flask). To address this issue, the team will watch videos either the ones provided by the professor or ones we find on our own. As well as we will help each other learn through personal experience or what we found and limit the scope to features in Priority 1.

- Schedule risks:

The risk is not having time to meet to discuss our progress or upcoming deadlines. We will address this issue by finding a time that a majority of the group can meet even if it is a 30 min time slot either in person or virtually via zoom meeting.

- Technical risks:

The technical risk is the admin not being able to review the media from the Workbench. We will address this issue by training the admin to view the media file by directly copying the link and playing it into the local system.

- Teamwork risks:

There can be a disagreement when it comes to this project. To address this issue, we all will have a voice and create a way to communicate our ideas and have an open mind.

Another teamwork risk is having everyone on the same page and doing a task at an appropriate time. To address this issue, we will create a task list and set deadlines for these tasks to be completed.

- Legal/content risks:

The legal risk includes finding media that will not cause us legal trouble in the future. We will address this issue by making our own media. Also, we will find free media online that have no copyright issues to use for the project.

7 . Project Management

GitHub Issues: The team will use GitHub Issues for Project Management.

As we are using GitHub for our Project, it is easier to maintain our tasks in Github itself to avoid managing different tools. GitHub provides an easy-to-use portal for managing GitHub Issues where we can assign the task with a description, assignee, labels, and deadline.

- Team leads (Frontend/Backend) will assign the task through GitHub Issue to the team members along with the deadline, title, description, etc. The team member will be able to push the relevant files to GitHub, and it will be reviewed and approved by the respective team lead, and then finally, it will be approved by the Team Lead of the Project.
- Changes done to the project can be linked to the issues, with appropriate labels and milestones to categorize them.
- Push Requests will be linked to the issues to be tracked.

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

Team number: 04

Meeting date: 11/16/22 (Wednesday 5:00 pm)

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

1. **Navigation Bar**
    - No Home Tab in the navigation bar
    - Disclaimer goes on the top right corner above everything
    - Move all the buttons (About, Upload, Register, log in) to the right
    - Add a line below the navbar that separates the page from the navbar

2. **Home Page**
    - The description on the home page should be about the website and not the team
    - Home page will show the most recent 12 posts and add some text like “Some recent uploads” and then show the posts

3. **Dashboard**
    - Add title
    - Address as “Welcome, username”
    - When a particular post is clicked, the post detail page should open in a new window

4. **Upload Post**
    - File should be uploaded at last
    - Thumbnail will not be uploaded; it should be created on the backend

5. **Message**
    - The messages should have a date, sender, and text. (If possible, can be sorted by date)

6. **Search Results**
    - Display “Showing 1 - x out of y images”

Common UI Changes:

- Buttons should be larger and colored (Delete/Cancel - Red, Submit/Confirm - Green, Send/Purchase - Yellow/Blue)
- Two buttons should have some space between both
- Homepage postcode should be the same as the search results
- Mention “*” are mandatory fields
- Make everything uniform (Black lines under the navbar - same for all the page)
- All titles of the pages should be uniform, big font, bold.

- **Summary of feedback on code and architecture**
    - Include Header Comments: Error, Input, Output, Owner, Coder, Date
    - Include Inline Comments
    - Fix the code, Document it, and put it on Master Branch

- **Summary of feedback on GitHub usage**
    - Write a detailed commit explaining what feature and file are changed and the reason of change

- **Summary of feedback on DB**
    - Database is okay.

- **Summary of feedback on teamwork and risk management**
    - The teamwork is good but the team is behind in UI so work on the feedback.
    - Last-minute building on the server should be avoided. The code must be tested and pushed in the master branch and the server immediately after testing.

List of features for final delivery: (Priority 1 Features):

- Login
- Register
- Homepage
- Search
- Post
- Post Detail
- Message
- Dashboard
- Inbox
- Uploaded Post

Any other comments and issues Check Point (CP), DUE 11-22-22:

- Develop all UI pages for the application as per your P1 list, based on feedback from M2 and M3
- Connect at a minimum: home, search results, search details and have it run from the cloud server
- Other pages you can develop as design with no connection
- All pages must be developed using your framework
- Organize your github and SW development to follow best practices

SW Engineering CSC648/848 Fall 2022

GatorExchange

(WWW site for Buy/Sell/Share of Digital media exclusively for SFSU students and faculty)

Team 4

Mahisha Patel	mpatel17@mail.sfsu.edu	*Team Lead*
Sudhanshu Kulkarni	skulkarni@sfsu.edu	*GitHub Master*
Sophia Chu	jliu36@mail.sfsu.edu	*Front-end Lead*
Jerry Liu	ebuddharuksa@sfsu.edu	*Back-end Lead*
Ekarat Buddharuksa	schu5@mail.sfsu.edu	
Ruben Ponce	rponce3@sfsu.edu	

Milestone 4

December 9 , 2022

Date submitted for review	Date revised
December 9, 2022	Created as the first version

Table Of Contents

1. Product Summary

- Name of the Product: **GatorExchange**

- Description: GatorExchange provides an easy-to-share, sell, and buy website for exchanging digital media among SFSU students and staff members. The website users can search digital materials related to their interests without any login and sign-up process. It enables the students and staff members of the SFSU to contact the owner of the media for buying or using it for free.

- Major Functions:
    1. Login: One can log in to GatorExchange Website to be able to message sellers or upload any media for sharing/selling.
    2. Registration: One can register to become a GatorExchange member to be able to post media and contact sellers.
    3. Browsing: One can browse through different pages from the navigation bar.
    4. Homepage: One can view recently approved media postings from the home page.
    5. Search: One can search the list of digital media posted by SFSU students and staff based on different categories and search queries.
    6. Post: One can post digital media for sharing or selling.
    7. Post Detail: One can view the post details of any digital media.
    8. Contact Seller: One can contact the seller by sending a message along with his contact details.
    9. Dashboard: One can view his/her own posted media on the dashboard.
    10. Inbox: One can see the list of messages received by other users.
    11. Admin Approval: The admin approves the posts using SQL Workbench to be seen by the other users.
    12. Google Analytics: The admin can view the analytics of the GatorExchange Website.
    13. About: One can see the brief details of the developers of GatorExchange.

- Uniqueness:
    1. GatorExchange is exclusive to SFSU; it allows searching from limited and most relevant materials posted by someone from the same community.
    2. The buyer can send a message to the seller for any digital media that he/she is interested in.
    3. The Homepage will show the most recent 10 posts for the latest updates.

- URL to the product: http://54.200.101.218/

2. Usability Test Plan

Usability Test: **Search Functionality**

- Test Objectives:
    1. The main objective is to test search functionality and whether it returns results relevant to the search query.
    2. The test objective is to check if the user interfaces for this functionality is easy to access and use.
    3. The time for search query execution is to be tested.

- Test background and setup:

  System Setup:

  The users are required to have a device, i.e., (Computer / Laptop) with a stable internet connection and go to GatorExchange URL from Google Chrome or Mozilla Firefox browser.

  Starting Point:

  The user has to go to GatorExchange URL, enter the search query in the search bar and select a category from the dropdown and the relevant list of digital media should appear. The length of the search query should not exceed 40 characters.

  Who are the intended users?

  The intended users for the website are SFSU students and staff, and they can be a guest, registered, or admin users. The guest users can browse and search but have to get registered if they wish to contact sellers or post-digital media via lazy registration.

  URL of the system to be tested:

  http://54.200.101.218/

- Usability Task Description:

  The user has to search for digital media related to some specific keyword or interest and choose the category. The search results should display relevant results based on the search query and category field.

  The following 4 cases should be tested before filling out the above survey:

1. User can directly click on the search button without typing or selecting anything.

2. User can select the category and click on search.

3. User can simply type some search query and click on search.

4. User can select the category from the dropdown and type the search query and then click on search.

- Evaluation of Effectiveness:

  The effectiveness will be measured as follows:

Test/Use Cases	% Completed	Errors	Comments
Search Query			
Search Category			
Contact Seller			

- Evaluation of Efficiency:

  The evaluation of efficiency can be measured by the time taken to execute the search query, Number of clicks, Number of screens and Number of pages of instructions.

- Evaluation of User Satisfaction:

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
The search results were useful					
The category dropdown was helpful					
The time taken for the search was less					
The search process was smooth and easy					
The UI was convenient to use					

3. QA Test Plan

- Test Objectives:

  The objective is to test search, post, and contact the seller as a guest/registered user.

- HW and SW setup:

  HW Setup: A computer or laptop device with a stable internet connection.

  SW Setup: The below software setup is required while testing on the local machine, otherwise a browser with internet supportability (Chrome/Firefox) is sufficient.

  Server Host: AWS EC2
  Operating System: Ubuntu 18.04
  Database: MySQL 8.0
  Web-Server: Nginx 1.22.0
  Server-Side Language: Python
  Browser: Chrome/ Firefox
  Additional Technologies:
  Web Framework: Flask, React, Node
  IDE: VS Code/PyCharm
  Web Analytics: Google Analytics

- Feature to be tested:

  Test Sequence

  Search:

  Go to GatorExchange via http://54.200.101.218/, locate the search bar, type the search query and select the category from category dropdown beside the search bar and finally click on “search” button.

  Upload Post:

  Go to GatorExchange via http://54.200.101.218/, Click on Upload in the navigation bar, fill in the upload post form (title, description, price, media type, media file) and click on the “post” button.

Contact Seller:

First, perform the steps of Search as written above, then from the search results, click on any post detail, locate the “Contact Seller” button at the end of the media detail page and click on it. A pop-up will appear with a pre-filled text message, edit and click “Send”.

- QA Test Plan:

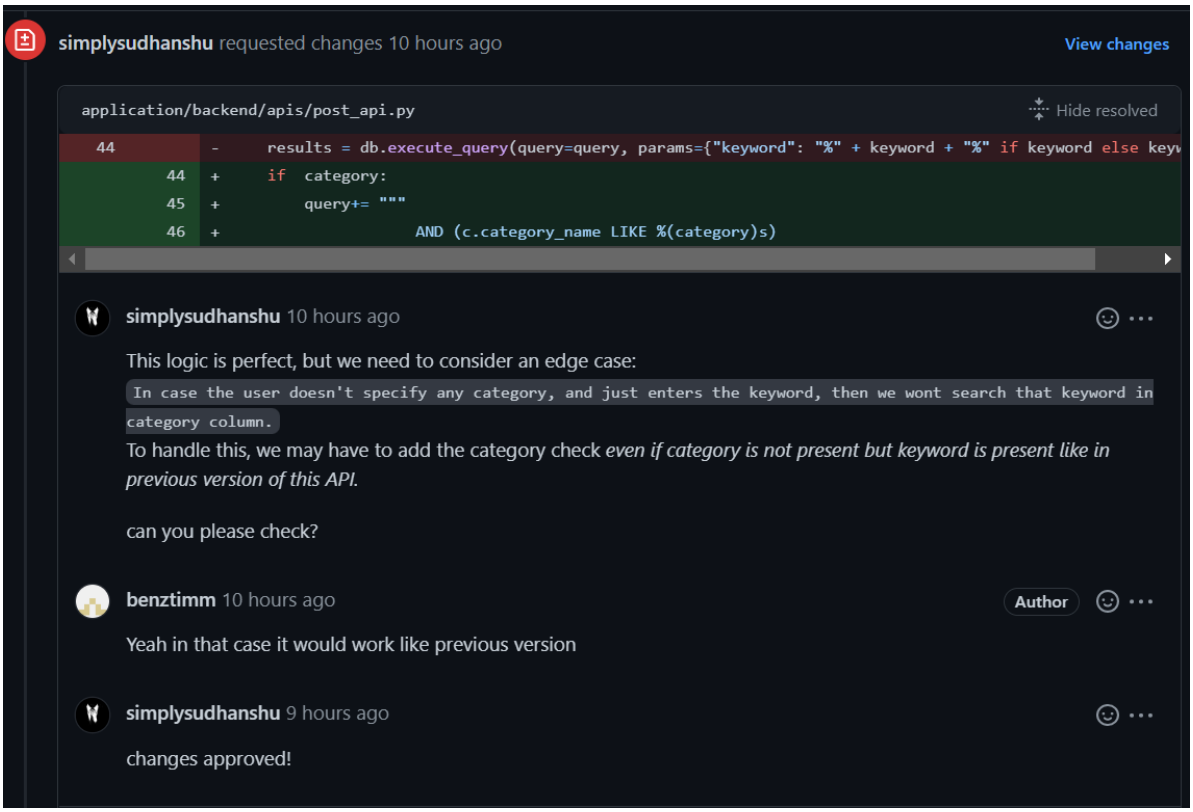
Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Result (Pass / Fail)
1.	Search result	Test search results (%like logic shall be tested)	Type “CS” in the search field and click search	List of 3 media that has “CS” either in the description or title	Pass in Chrome/ Pass in Firefox
2.	Search Input Validation	Test the search input field validation, i.e., no special characters should be allowed	Type any special character ($, %, ^, &, *, etc.) in the search field	An alert message pops up saying “Search input must be alphanumeric”	Pass in Chrome/ Pass in Firefox
3.	Search query length validation	Test the search query length, i.e., the search query should not exceed 40 characters	Type any search query that exceeds 40 characters in total	More than 40 characters are not allowed and one can not type more than 40. A message pops up saying the maximum number of allowed characters is 40.	Pass in Chrome/ Pass in Firefox
4.	Upload Post	Registered and logged-in User is uploading the media for selling	Submit the media file, title, description, category, and price for posting it on the website.	An alert message pops up saying “Post will be approved within 24 hours.”	Pass in Chrome/ Pass in Firefox
5.	Contact Seller	Guest User tries to contact	Opens the post details and	Lazy Registration (Registration form	Pass in Chrome/ Pass

		seller but lazy registration happens.	clicks on contact seller button	pops us)	in Firefox
6.	Test Post Data	Admin will test the Media post data on the workbench.	The admin opens the workbench and has login credentials to access the database.	Checks all the data (media file, category, title, price, and description) from the Post Table in Workbench.	Pass in Workbench

4. Code Review

Code review was done in the following way:

1. The code was submitted by a backend team member for review to the GitHub master and team lead.
2. Github Master asked for some clarification which was followed by a reply from the backend team member.
3. Github Master reviewed and approved.
4. Team Lead reviewed the entire code and made some comments and approved.



Review by Github Master

Review by Team Lead

5. Self-check on best practices for security

Asset to be protected	**Types of possible/expected attacks**	**Strategy to mitigate/protect the asset**
The user's email/password in the database (High value)	Unauthorized user gains access to the database (High probability)	The password of the users will be encrypted in the database. The database is accessible by the admin only.
The user's email/password in the database (High value)	SQL Injection through login/registration forms (Medium probability)	Input Validation on the field to have only alphanumeric characters. The email of the user should be the SFSU school email, i.e., sfsu.edu at the end.
The media/category database (Low value)	SQL Injection through search bar (Medium probability)	Input Validation on the search bar to have only alphanumeric characters with a maximum

		total length of 40 characters. The backend server has connected to the database via the 'pymysql' library and can handle SQL injection vulnerabilities.
The application's hosting service credentials	Unauthorized user makes application unavailable (Low probability)	Strict management to credentials and keys to the web hosting server.
An individual user's account (Low value)	Unauthorized access to the user's account (Medium probability)	The admin can use the database-stored e-mail for verification in order to restore or reset a user's login credentials.

6. Self-check of the adherence to original Non-functional specs

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

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

3. All or selected application functions must render well on mobile devices. **ON TRACK**

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

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

6. The privacy of users shall be protected. **DONE**

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

8. The application shall be very easy to use and intuitive. **DONE**

9. The application should follow established architecture patterns. **DONE**

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

11. Google Analytics shall be used. **ON TRACK**

12. No 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. **DONE**

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

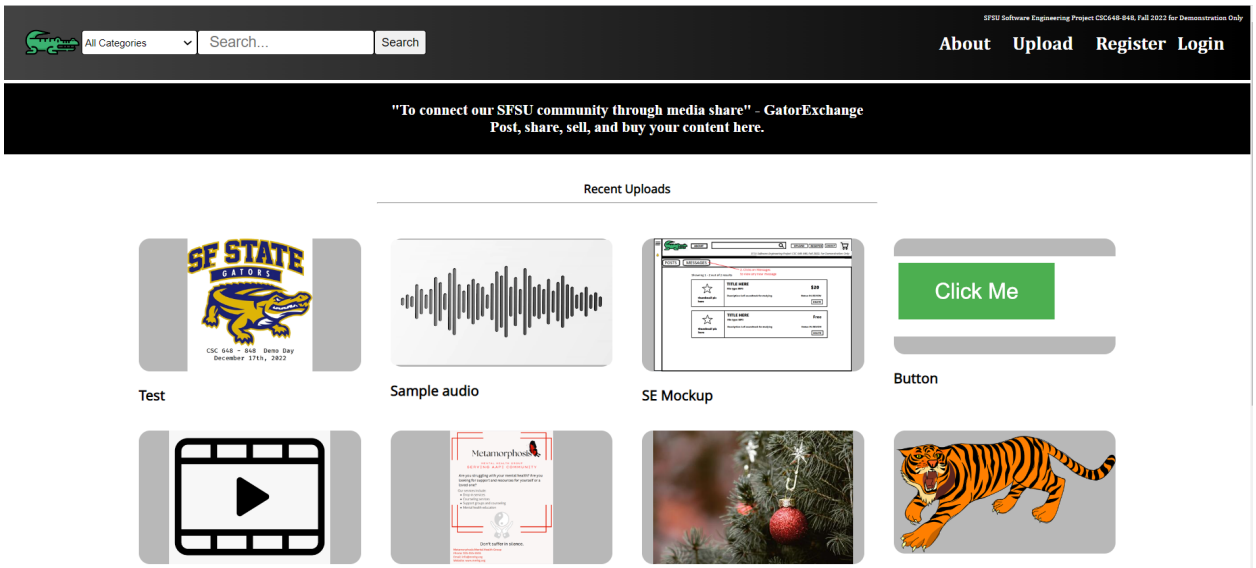
14. Site security: basic best practices shall be applied (as covered in the class) for main data items. **DONE**

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

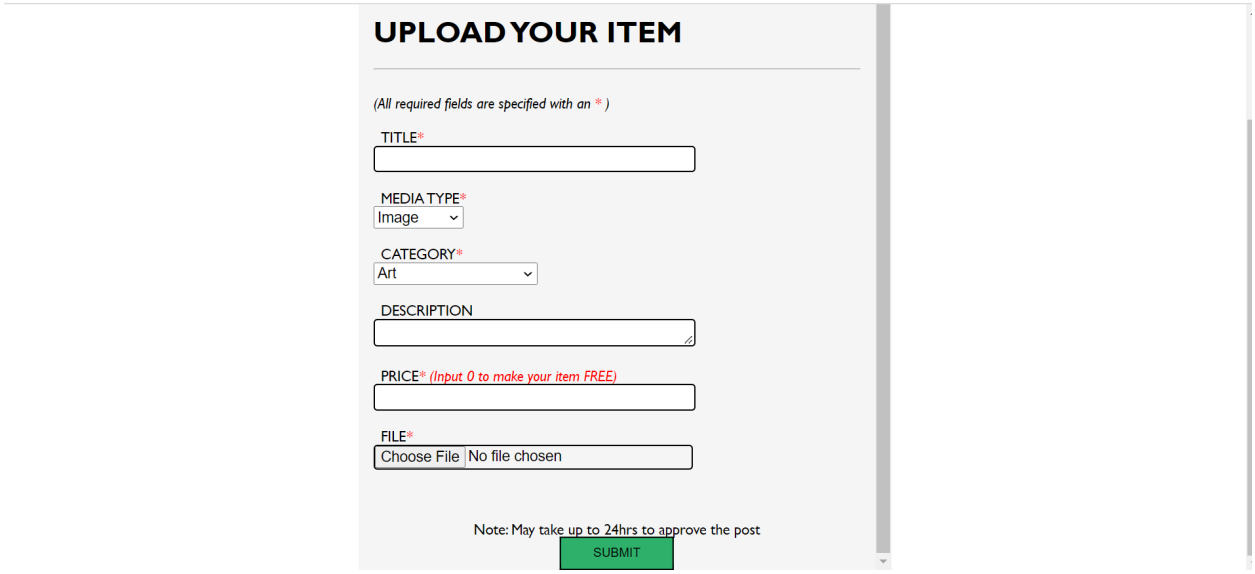
16. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development. **DONE**

17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2022. For Demonstration Only" at the top of the WWW page nav bar. **DONE**

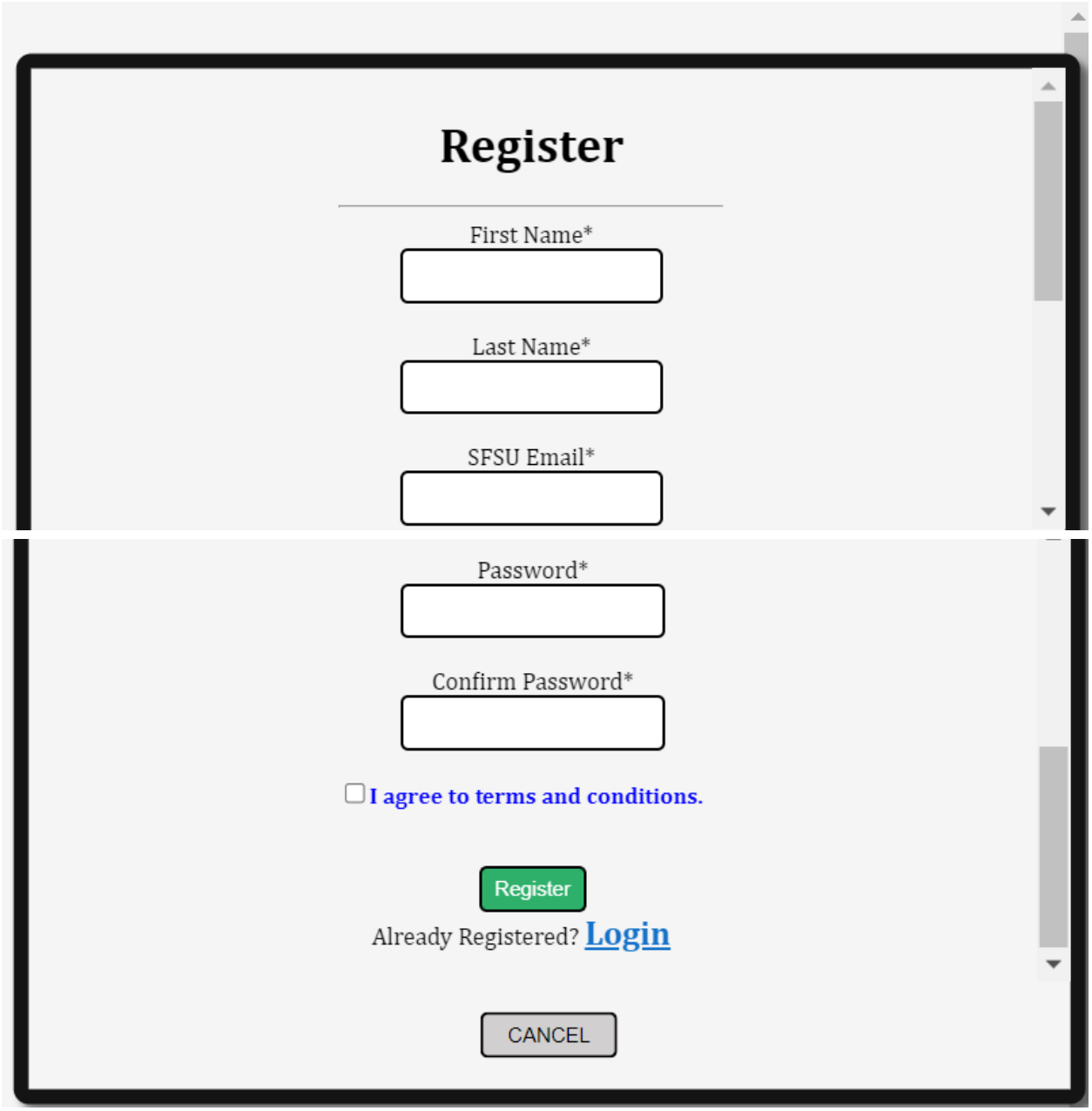
Product Screenshots



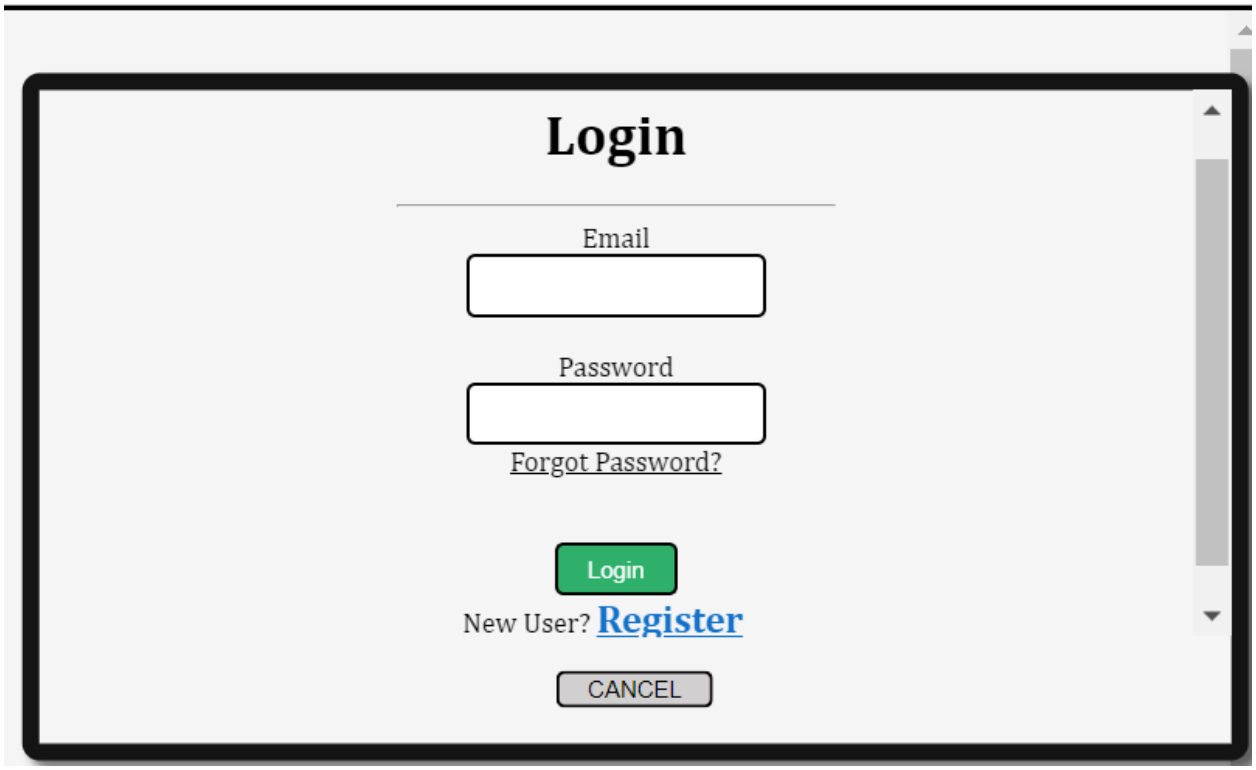
Homepage



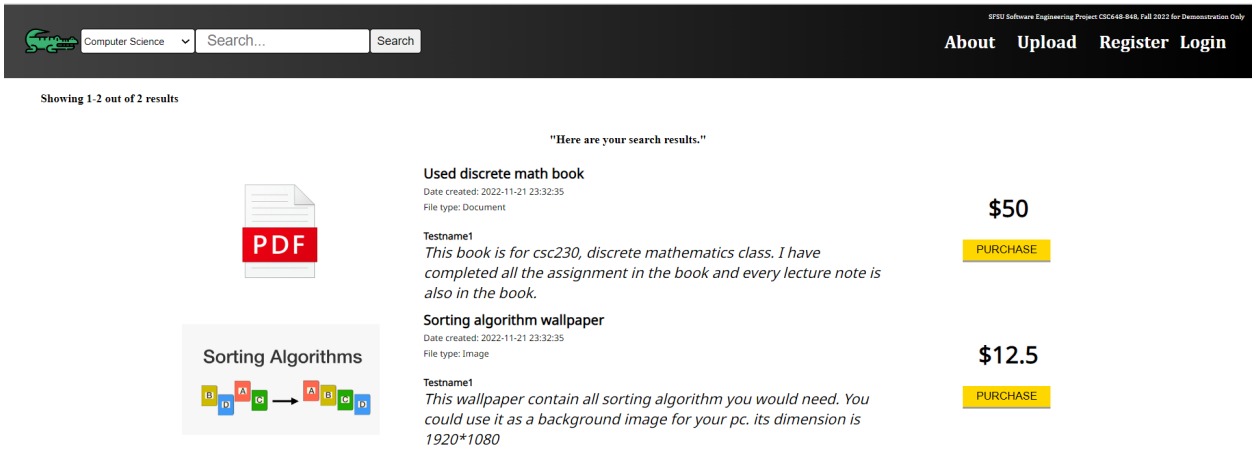
Upload Post



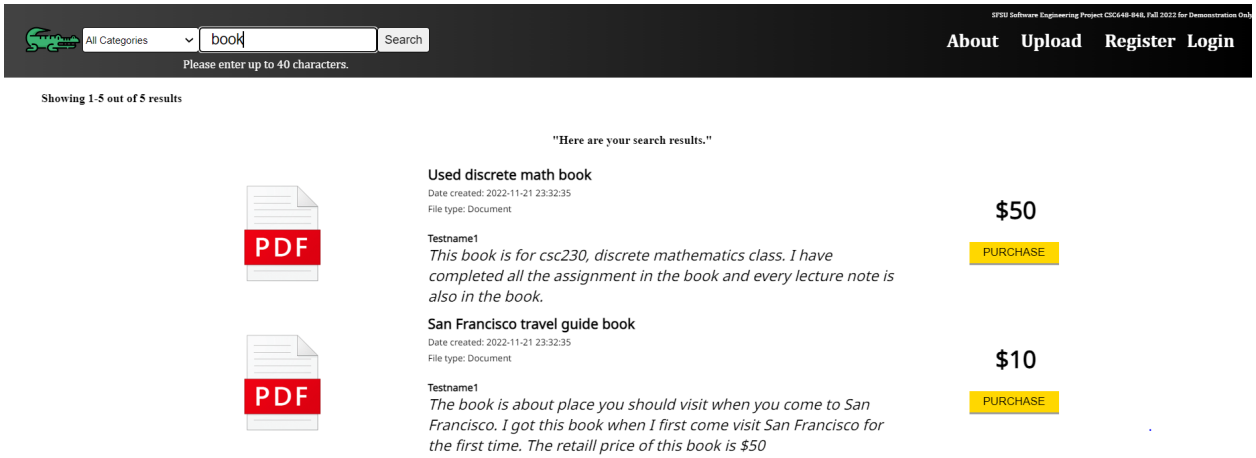
Registration



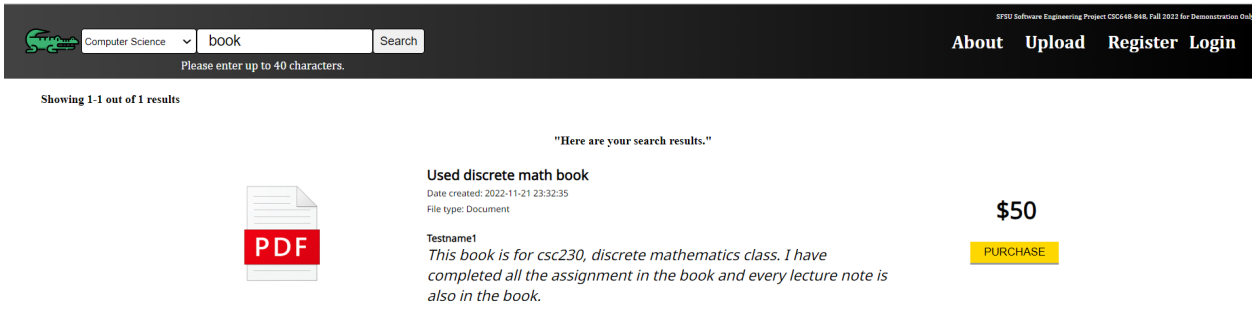
Login



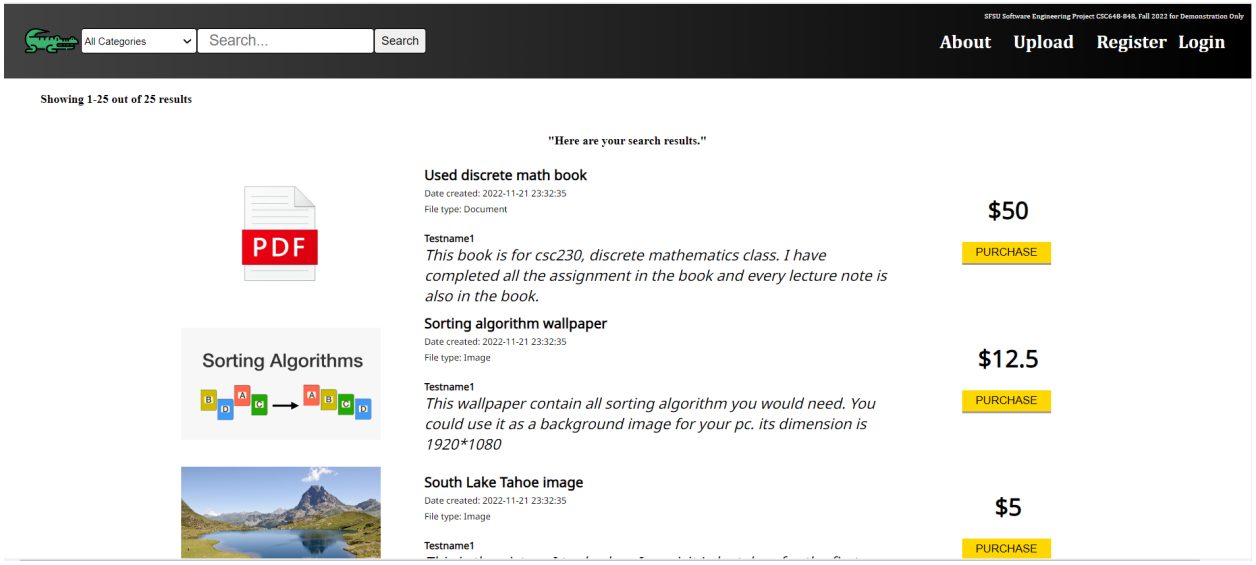
Search with Category: “Computer Science”



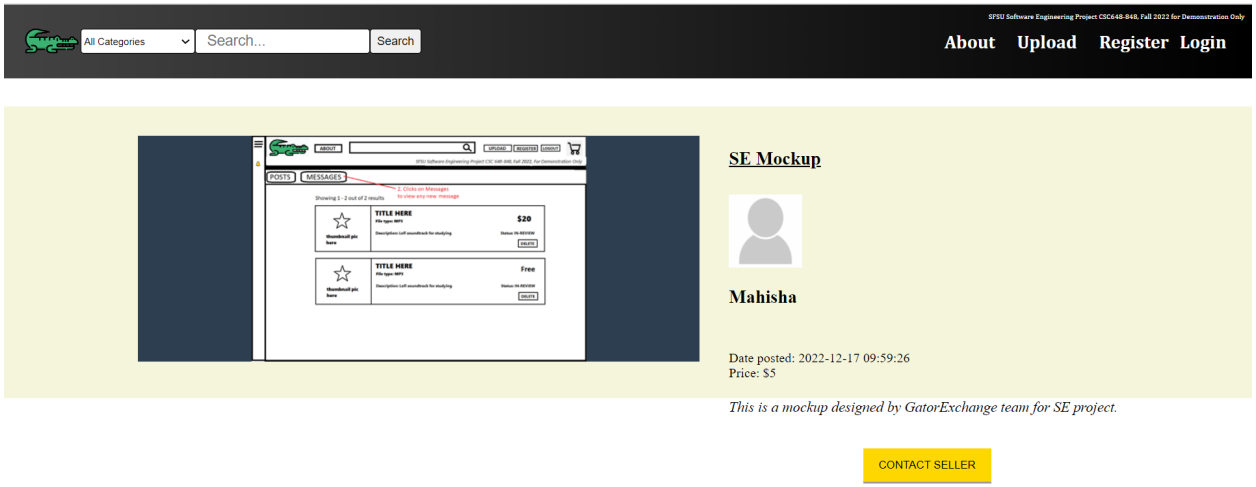
Search with search query: “book”



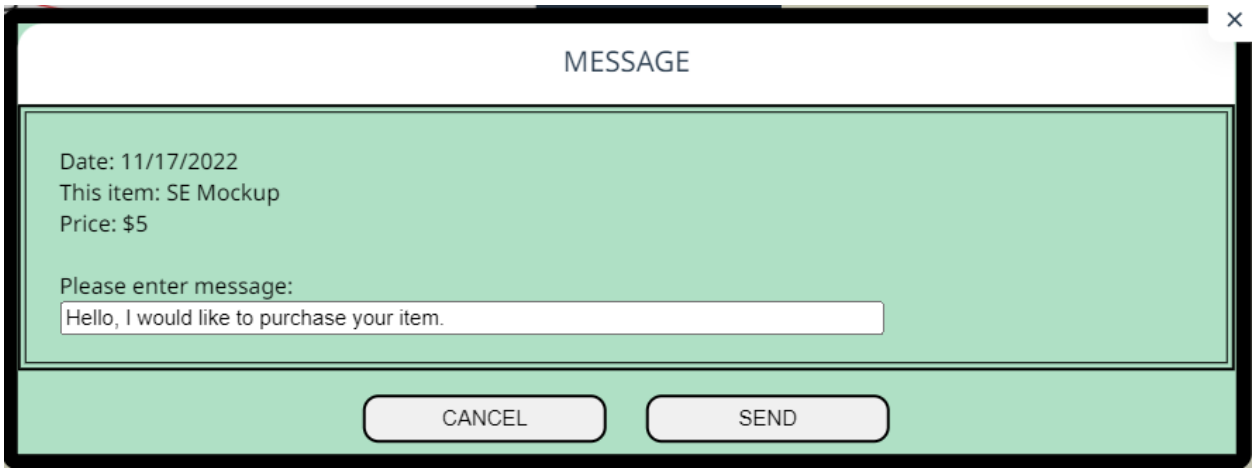
Search with Category “Computer Science” and Search Query “Book”



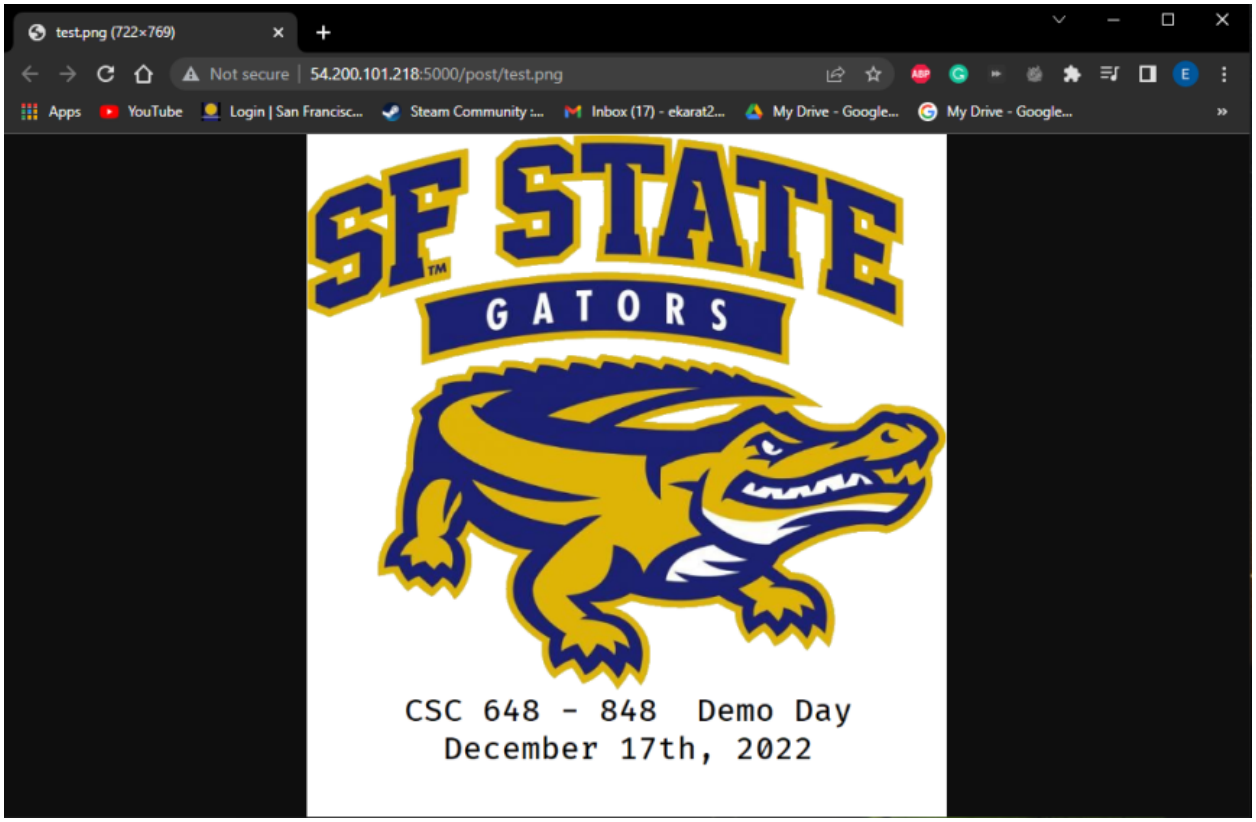
Empty search with “All Categories” and no search query



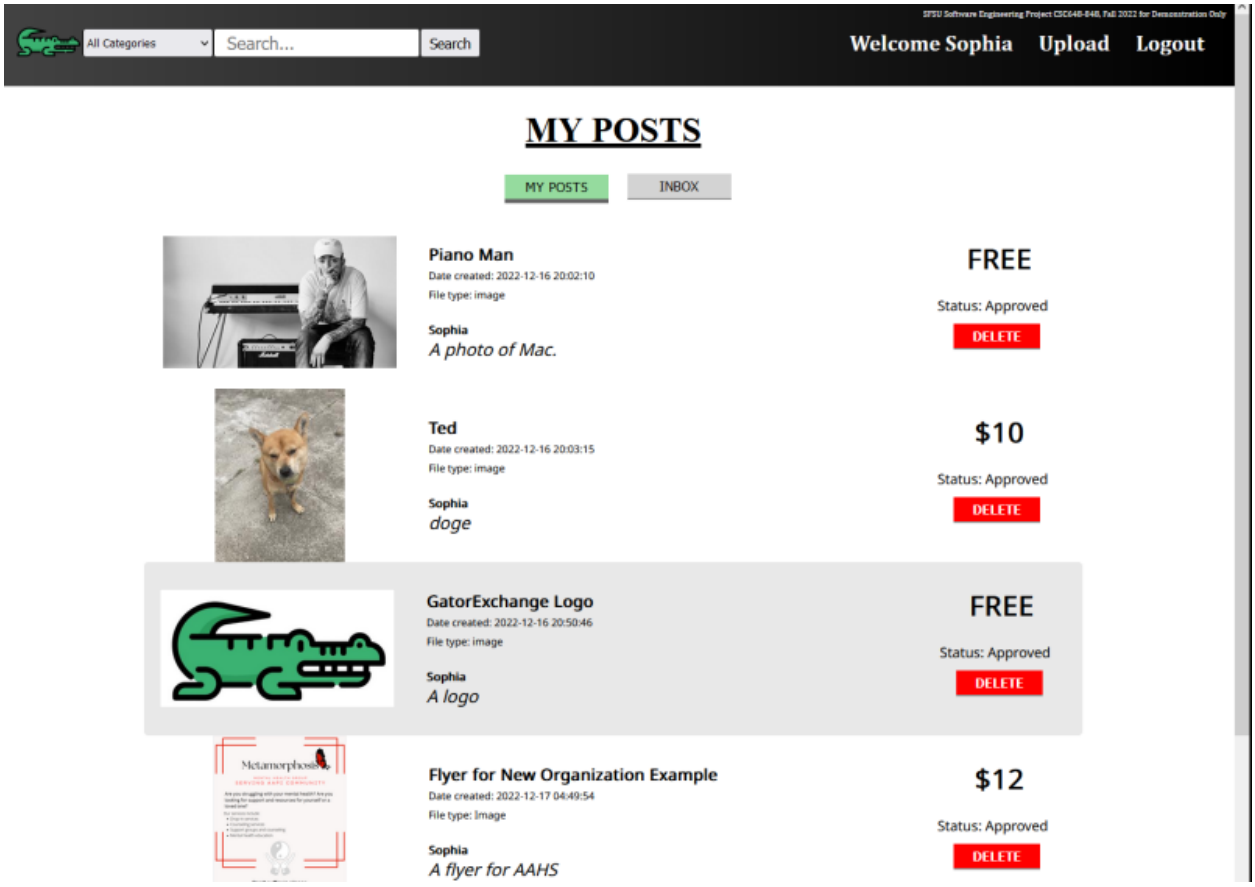
Post Detail



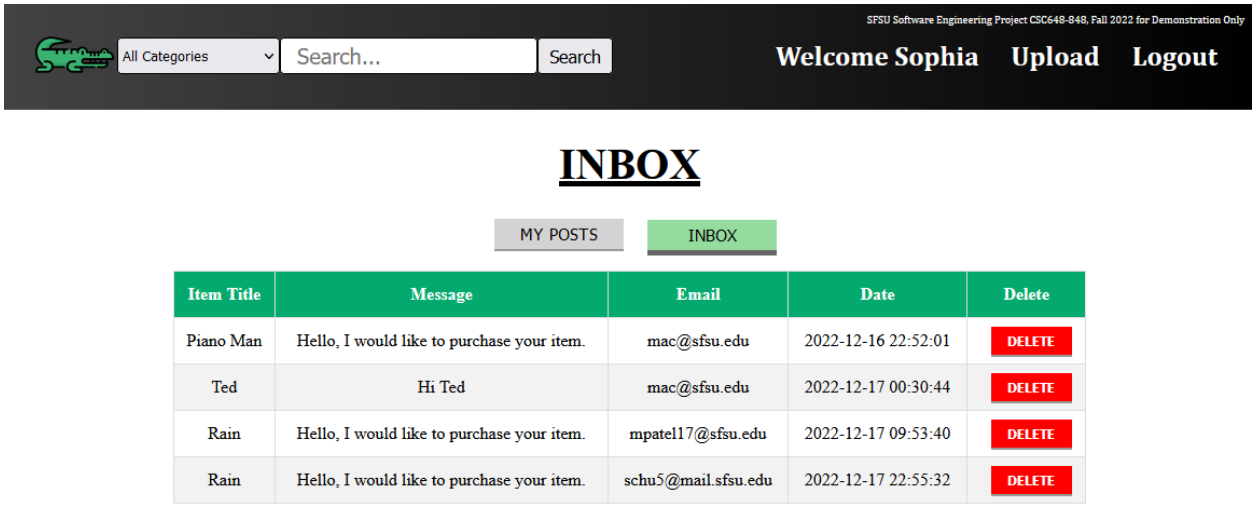
Contact Seller



Admin viewing post on the server and will then approve via Workbench

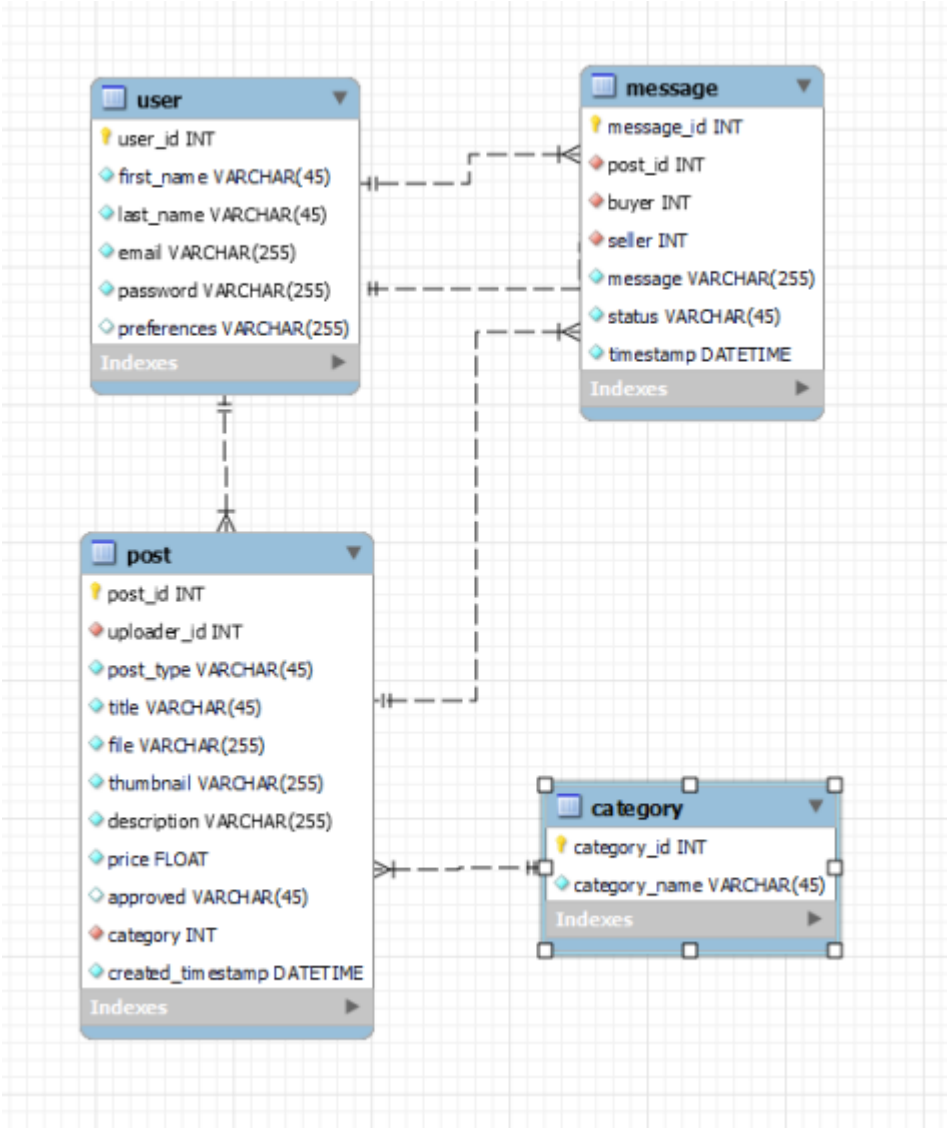


Dashboard - Post



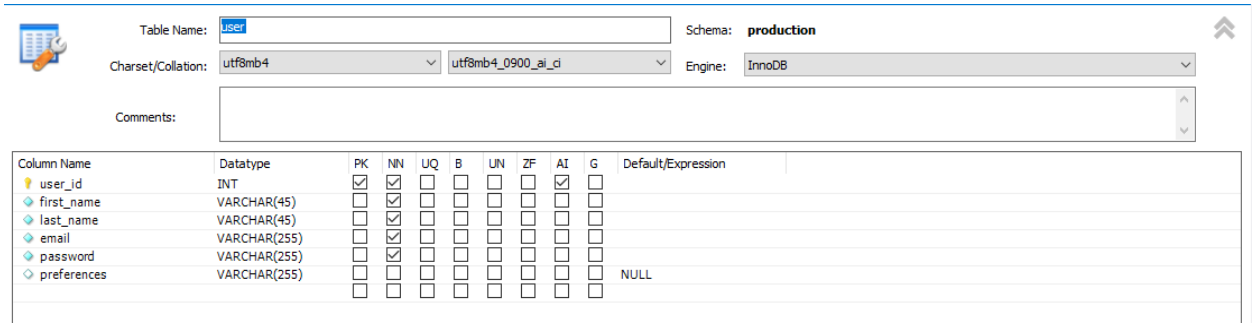
Inbox

Database Organization

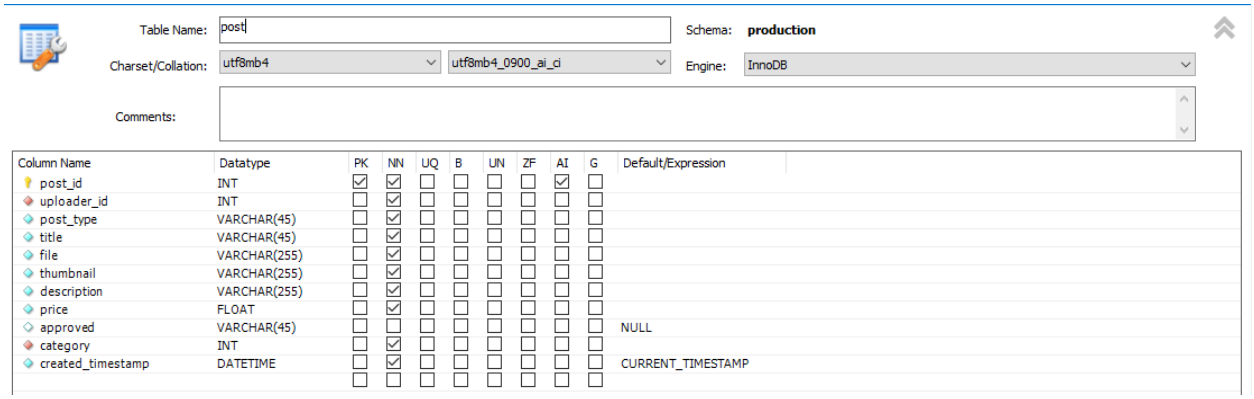


E-R Diagram

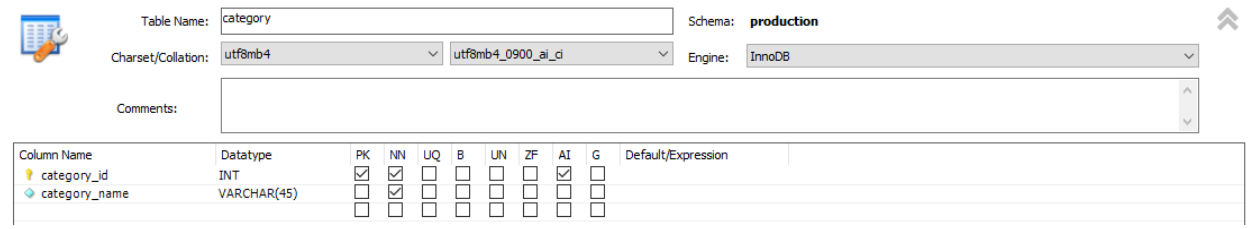
User Table



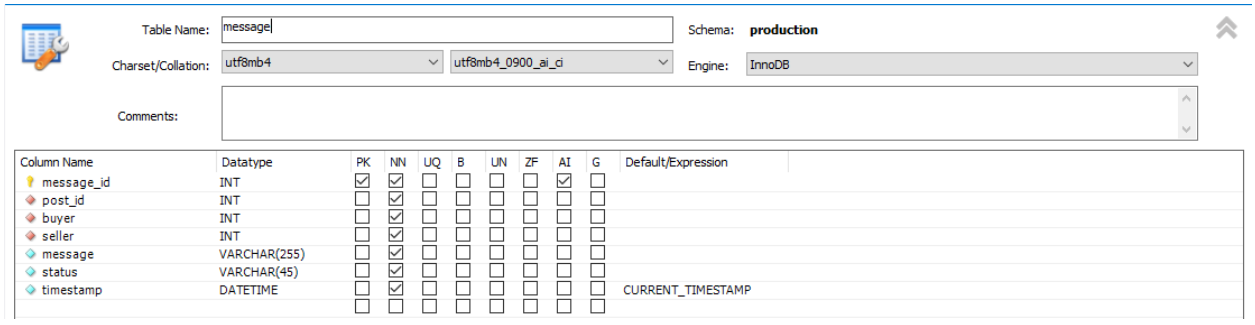
Post Table



Category Table



Message Table



Github Organization

On GitHub, we followed strict branching and merging policies. We followed a naming convention while creating a new branch, which strictly began with the type of dev that went into it, for example, *'feature/search-api'* or *'hotfix/search-query-fix'* The repository always held these protected branches:

- Master: This branch held the latest stable release of the app. We cannot push commits directly to the master. The code has to go via a pull request from a separate branch *(typically the release branch)* which needs to be approved by either the team lead or the GitHub master.

- Release: This branch holds the intermittent app version, which has all the required changes for the upcoming release. The code on this branch undergoes exhaustive testing by all the team members, and bug fixes are deployed if required. Merging the code on this branch requires code review and approval by at least two leads *(team lead, GitHub master, front-end lead, back-end lead)*.

- Dev: Dev branch has all the work that is recently developed, and unit tested. Each developer usually merges the code from a feature branch, but we can commit directly to the branch in case of a quick fix. Pull requests to the Dev branch goes through code reviews and are linked to their specific GitHub issues for tracking.

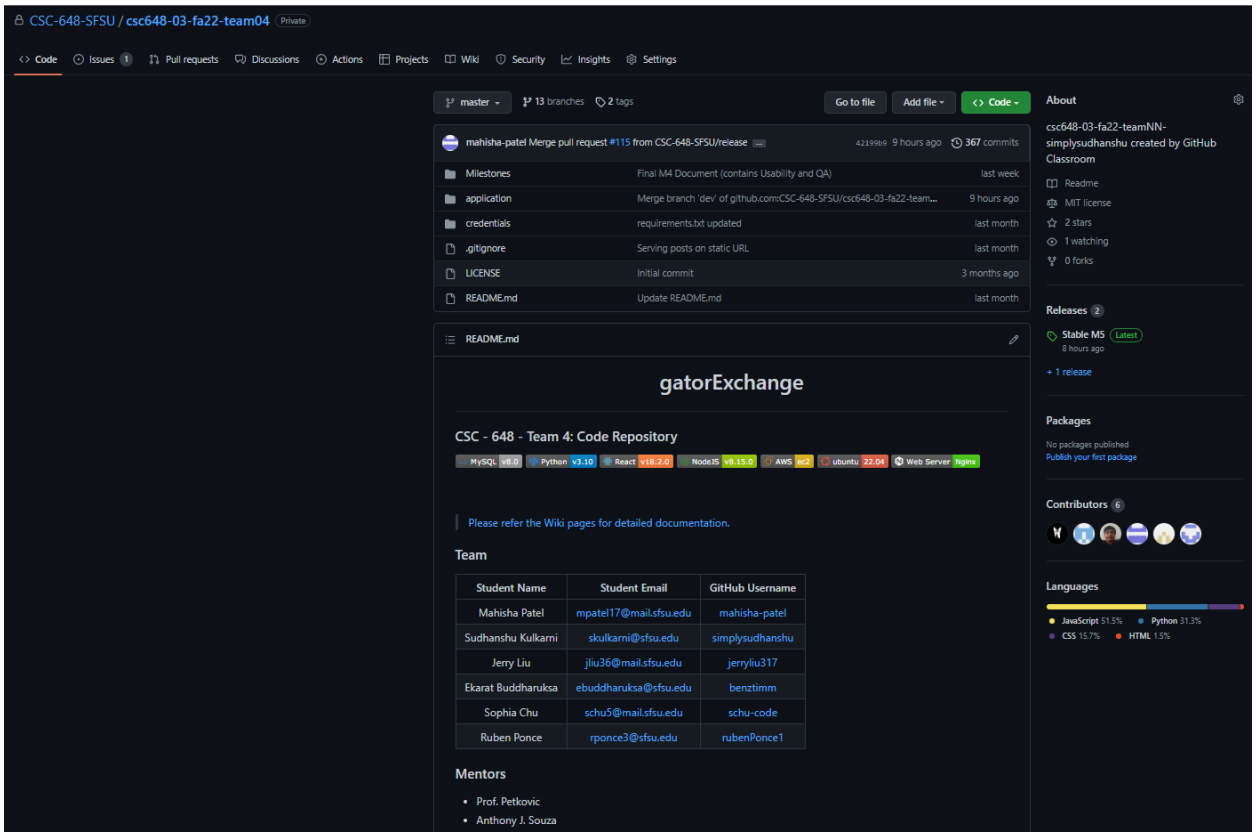
The repository had a clearly designed directory structure split into three main sections:

1. Milestones: This folder held the documentation for each milestone submitted for grading. Along with documents for each milestone, we have an overview of what each milestone consisted of in the readme file.

2. Application: The entire code was housed here, split into *Frontend and Backend.* The frontend folder had the react app, while the backend folder had the python-flask app. There were a few '.gitignore' files to manage file tracking.

3. Credentials: The readme file in this folder is the most important document, which contains technical details and credentials to the server and database. There is also a private key, which is used to connect with the server via ssh.

The home page of our repository looks like this; however, the WiKi pages have detailed instructions on installing the application *(both front-end and back-end frameworks),* documentation for each back-end API, and a detailed guide on how to use GitHub, branch, and merge protocols.



Home Page of Github Repository

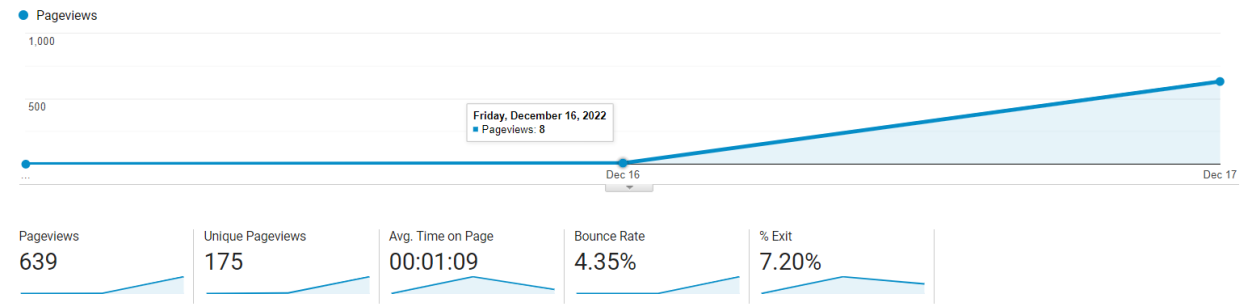
Google Analytics Stats plot

Primary Dimension: Page Page Title Other

Plot Rows | Secondary dimension | Sort Type: Default | advanced

Page	Pageviews	Unique Pageviews	Avg. Time on Page	Entrances	Bounce Rate	% Exit	Page Value
	639 % of Total: 100.00% (639)	175 % of Total: 100.00% (175)	00:01:09 Avg for View: 00:01:09 (0.00%)	46 % of Total: 100.00% (46)	4.35% Avg for View: 4.35% (0.00%)	7.20% Avg for View: 7.20% (0.00%)	$0.00 % of Total: 0.00% ($0.00)
1. /	188 (29.42%)	36 (20.57%)	00:01:20	31 (67.39%)	6.45%	12.23%	$0.00 (0.00%)
2. /dashboard	65 (10.17%)	11 (6.29%)	00:01:16	0 (0.00%)	0.00%	1.54%	$0.00 (0.00%)
3. /productpage/38	64 (10.02%)	17 (9.71%)	00:01:01	6 (13.04%)	0.00%	9.38%	$0.00 (0.00%)
4. /Upload	55 (8.61%)	15 (8.57%)	00:01:36	3 (6.52%)	0.00%	7.27%	$0.00 (0.00%)
5. /searchresults	53 (8.29%)	22 (12.57%)	00:01:42	2 (4.35%)	0.00%	5.66%	$0.00 (0.00%)
6. /inbox	34 (5.32%)	10 (5.71%)	00:00:08	0 (0.00%)	0.00%	0.00%	$0.00 (0.00%)
7. /productpage/32	26 (4.07%)	9 (5.14%)	00:00:31	1 (2.17%)	0.00%	3.85%	$0.00 (0.00%)
8. /productpage/36	26 (4.07%)	7 (4.00%)	00:02:27	2 (4.35%)	0.00%	7.69%	$0.00 (0.00%)

Page views from December 15 to December 17



Pageviews	Unique Pageviews	Avg. Time on Page	Bounce Rate	% Exit
639	175	00:01:09	4.35%	7.20%

Site Content
- Page
- Page Title

Site Search
- Search Term

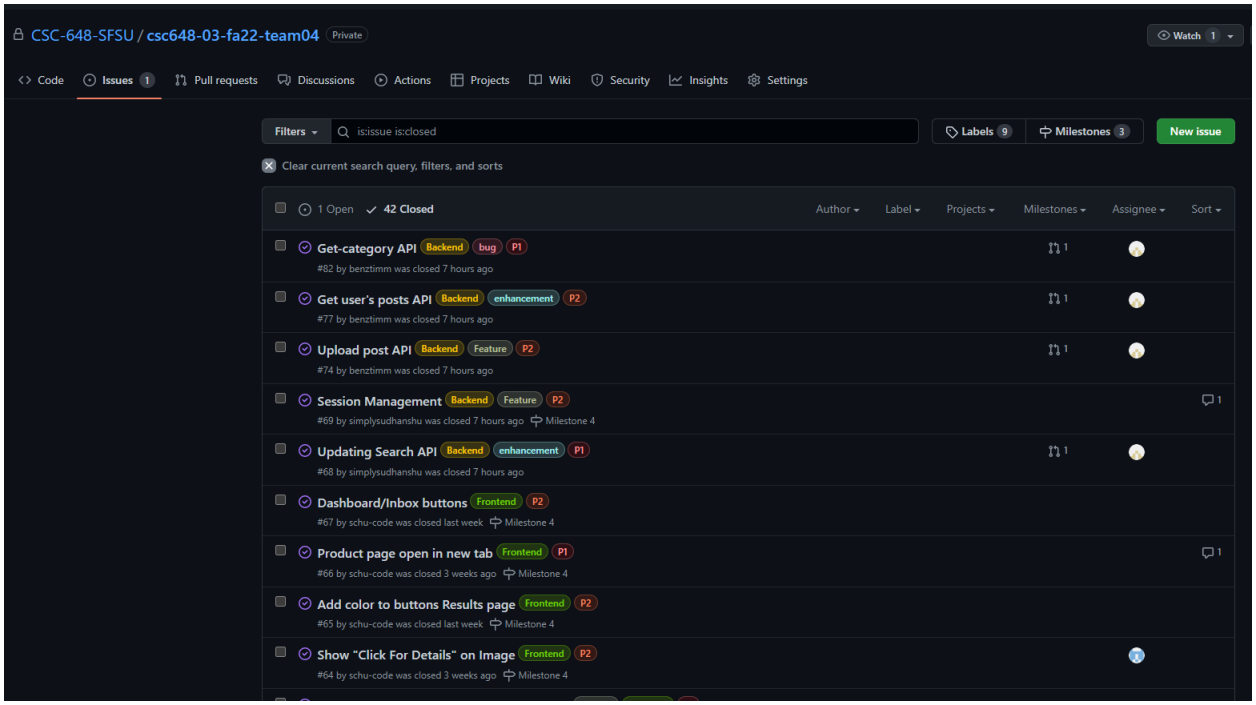
Events
- Event Category

Page	Pageviews	% Pageviews
1. /	188	29.42%
2. /dashboard	65	10.17%
3. /productpage/38	64	10.02%
4. /Upload	55	8.61%
5. /searchresults	53	8.29%
6. /inbox	34	5.32%
7. /productpage/32	26	4.07%
8. /productpage/36	26	4.07%
9. /productpage/33	17	2.66%

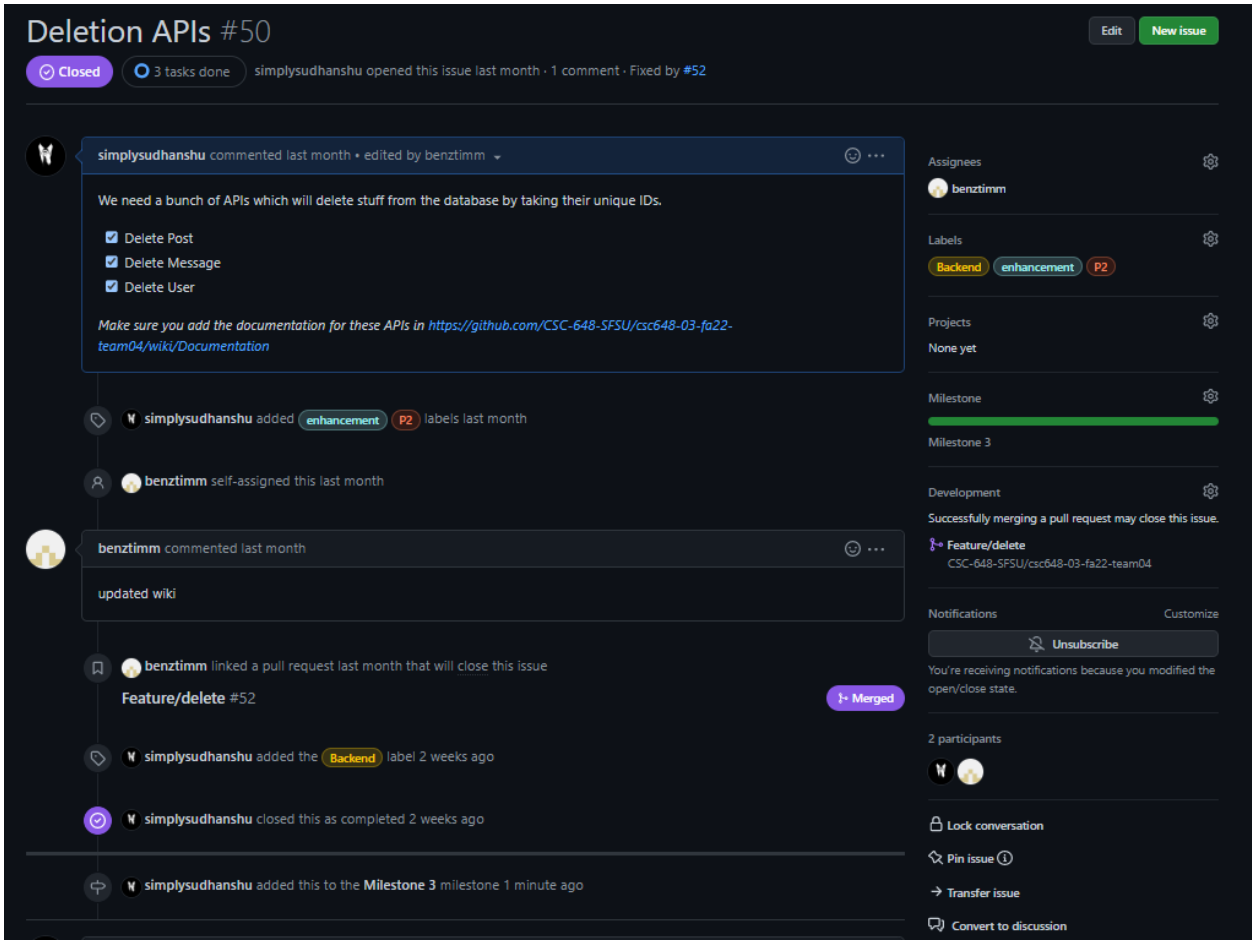
Unique Page views and Average Time per Page

Project Management

We used **GitHub issues** for project management. This was an efficient tool because we could directly link Pull Requests against the issues and have code reviews on the same thread. We can assign tasks to team members, set deadlines, and add checklists to keep track of an assignment. Further, we created different labels to categorize the tasks based on team *(backend, frontend)*, priority *(P1, P2),* and type *(feature, enhancement, bug, documentation)*. We set deadlines based on Milestones. Adding pull requests directly to an issue helps us maintain the code and cherry-pick the exact commit which contains the dev for that specific task. Once the concerned lead created the task, they were usually self-assigned or assigned to team members based on criticality and urgency. We have filtered views to display tasks based on status, type, milestone, team member, priority, etc., which gives us an overall view of how the project is progressing.



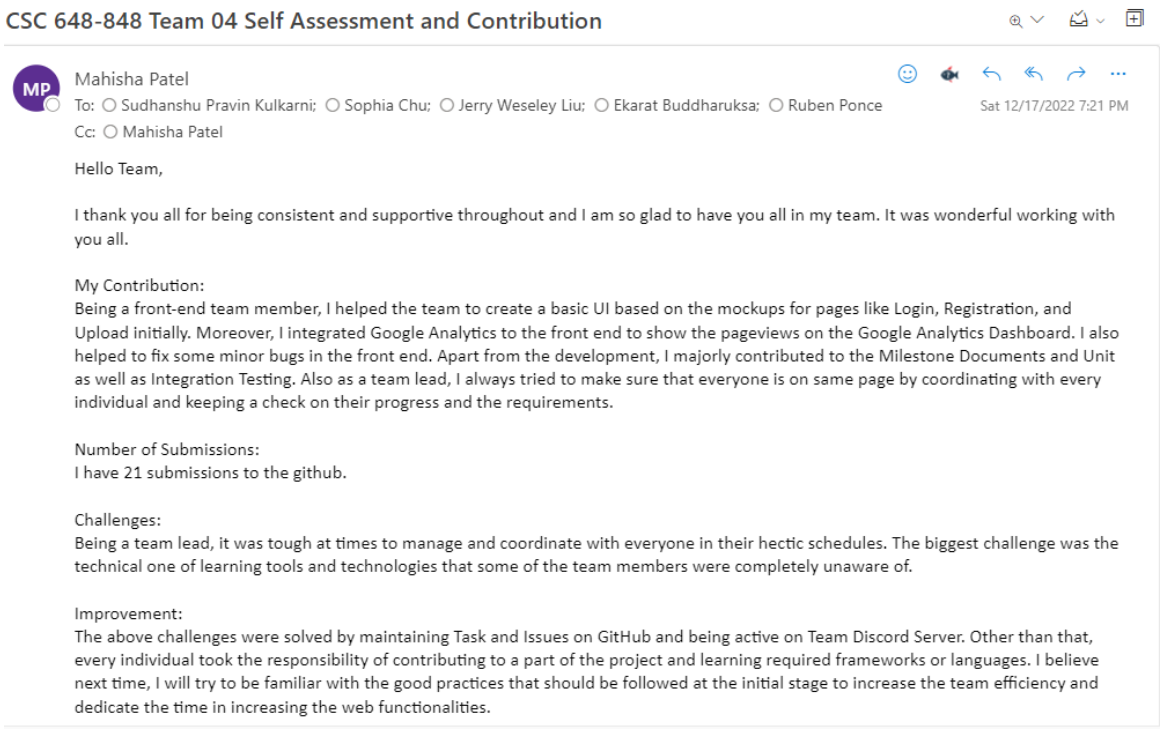
List of Assigned Tasks (Along with Priority, Team, Feature type, Milestone, and Assignee)



Example (Task creation, tracking, and progress)

Team Member Self Assessment and Contributions

1. Email by Mahisha (Team Lead, Front End Team Member)



CSC 648-848 Team 04 Self Assessment and Contribution

Mahisha Patel

To: Sudhanshu Pravin Kulkarni; Sophia Chu; Jerry Weseley Liu; Ekarat Buddharuksa; Ruben Ponce

Sat 12/17/2022 7:21 PM

Cc: Mahisha Patel

Hello Team,

I thank you all for being consistent and supportive throughout and I am so glad to have you all in my team. It was wonderful working with you all.

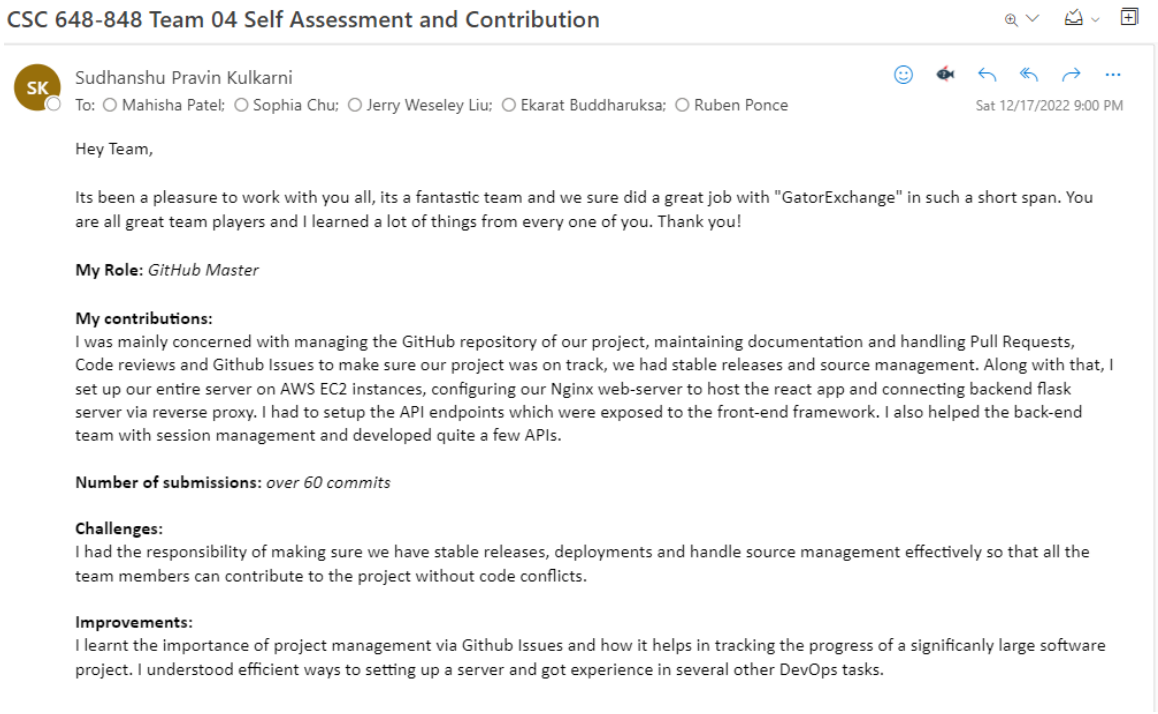
My Contribution:
Being a front-end team member, I helped the team to create a basic UI based on the mockups for pages like Login, Registration, and Upload initially. Moreover, I integrated Google Analytics to the front end to show the pageviews on the Google Analytics Dashboard. I also helped to fix some minor bugs in the front end. Apart from the development, I majorly contributed to the Milestone Documents and Unit as well as Integration Testing. Also as a team lead, I always tried to make sure that everyone is on same page by coordinating with every individual and keeping a check on their progress and the requirements.

Number of Submissions:
I have 21 submissions to the github.

Challenges:
Being a team lead, it was tough at times to manage and coordinate with everyone in their hectic schedules. The biggest challenge was the technical one of learning tools and technologies that some of the team members were completely unaware of.

Improvement:
The above challenges were solved by maintaining Task and Issues on GitHub and being active on Team Discord Server. Other than that, every individual took the responsibility of contributing to a part of the project and learning required frameworks or languages. I believe next time, I will try to be familiar with the good practices that should be followed at the initial stage to increase the team efficiency and dedicate the time in increasing the web functionalities.

2. Email by Sudhanshu (Github Master, Back End Team Member)



CSC 648-848 Team 04 Self Assessment and Contribution

Sudhanshu Pravin Kulkarni

To: Mahisha Patel; Sophia Chu; Jerry Weseley Liu; Ekarat Buddharuksa; Ruben Ponce

Sat 12/17/2022 9:00 PM

Hey Team,

Its been a pleasure to work with you all, its a fantastic team and we sure did a great job with "GatorExchange" in such a short span. You are all great team players and I learned a lot of things from every one of you. Thank you!

My Role: *GitHub Master*

My contributions:
I was mainly concerned with managing the GitHub repository of our project, maintaining documentation and handling Pull Requests, Code reviews and Github Issues to make sure our project was on track, we had stable releases and source management. Along with that, I set up our entire server on AWS EC2 instances, configuring our Nginx web-server to host the react app and connecting backend flask server via reverse proxy. I had to setup the API endpoints which were exposed to the front-end framework. I also helped the back-end team with session management and developed quite a few APIs.

Number of submissions: *over 60 commits*

Challenges:
I had the responsibility of making sure we have stable releases, deployments and handle source management effectively so that all the team members can contribute to the project without code conflicts.

Improvements:
I learnt the importance of project management via Github Issues and how it helps in tracking the progress of a significanly large software project. I understood efficient ways to setting up a server and got experience in several other DevOps tasks.

3. Email by Sophia (Front End Lead)

Sophia Chu

To: Ruben Ponce; Mahisha Patel; Sudhanshu Pravin Kulkarni; Jerry Weseley Liu; Ekarat Buddharuksa

Sat 12/17/2022 8:21 PM

Hello Team!

I had a great time working with you all. I really appreciate how much we communicated and how frequently everyone checked Discord. Shoutout to the backend team, thank you for quickly putting together all the Api's I asked for and making all the changes I requested!

My Contribution

- Designed and implemented base React website with all routes and pages
- Styled pages with css
- Implemented frontend tech that enabled display support for different file formats (videos, audio, pdfs, etc)
- Made sure all user inputs were sending the correct data
- Connected apis to frontend UI
- Implemented table for user inbox

Commits: 104

Main challenge

My main challenge was being frontend lead but this was (basically) my first time doing frontend. I've also never coded in javascript. I felt like picking React as part of our tech made some tasks more challenging due to the states and rendering. I've also never worked with apis in this manner, only for math related things like graphing. I was pretty much learning every step as each milestone was due.

Improvements

Next time, I will have a better understanding of how to set up the base website. We had some css issues related to

4. Email by Ruben (Front End Team Member, Document Editor)

Ruben Ponce

To: Mahisha Patel; Sudhanshu Pravin Kulkarni; Sophia Chu; Jerry Weseley Liu; Ekarat Buddharuksa

Sat 12/17/2022 7:55 PM

Hello team 4,

It was a great learning experience this year working with you all. Thank you all for all the hard work all semester. It was great meeting and working with you all.

My Contributions:

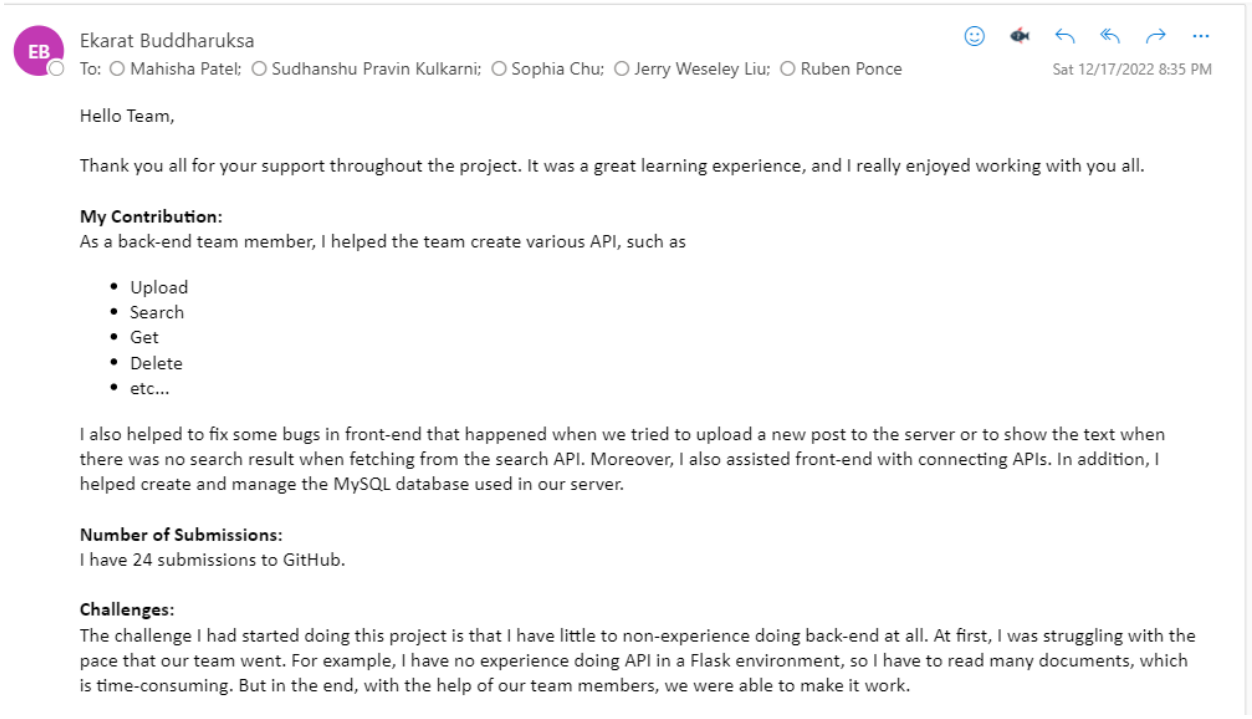
Frontend Team

1. Creating and editing the Navbar
2. About Us page
3. Modal for product message to send buyer (started initial helped by others)
4. Product detail page (started initial helped by others)
5. Helped maintain css uniformity
    a. Text and fonts being correct size and be uniform
    b. Helped create and keep buttons uniform in all different pages
    c. Helped with UI when it was resized
    d. Had some small contributions with UI of different pages

Number of Submissions:

21 submissions. Was somewhat low because in the beginning I was not pushing changes for every change waited and pushed large number of changes.

5. Email by Ekarat (Back End Team Member)



Ekarat Buddharuksa
To: Mahisha Patel; Sudhanshu Pravin Kulkarni; Sophia Chu; Jerry Weseley Liu; Ruben Ponce
Sat 12/17/2022 8:35 PM

Hello Team,

Thank you all for your support throughout the project. It was a great learning experience, and I really enjoyed working with you all.

My Contribution:
As a back-end team member, I helped the team create various API, such as

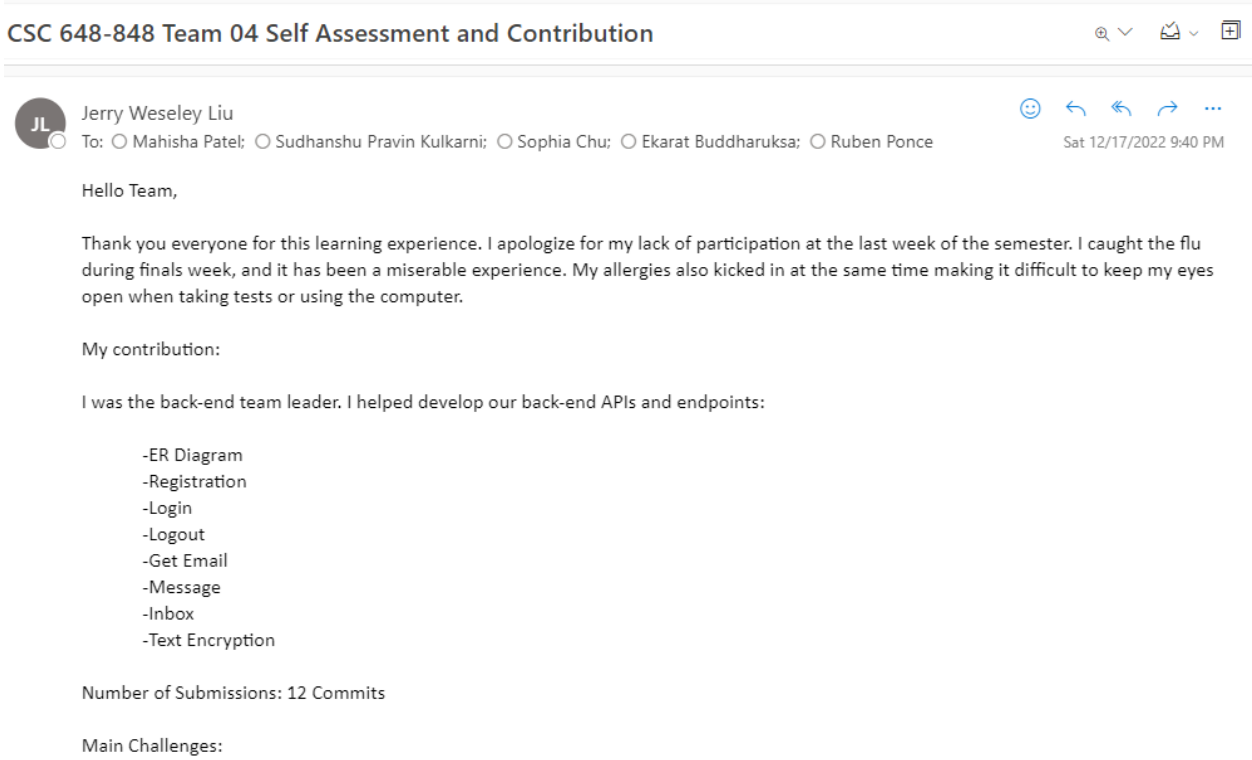
- Upload
- Search
- Get
- Delete
- etc...

I also helped to fix some bugs in front-end that happened when we tried to upload a new post to the server or to show the text when there was no search result when fetching from the search API. Moreover, I also assisted front-end with connecting APIs. In addition, I helped create and manage the MySQL database used in our server.

Number of Submissions:
I have 24 submissions to GitHub.

Challenges:
The challenge I had started doing this project is that I have little to non-experience doing back-end at all. At first, I was struggling with the pace that our team went. For example, I have no experience doing API in a Flask environment, so I have to read many documents, which is time-consuming. But in the end, with the help of our team members, we were able to make it work.

6. Email by Jerry (Back End Lead)



CSC 648-848 Team 04 Self Assessment and Contribution

Jerry Weseley Liu
To: Mahisha Patel; Sudhanshu Pravin Kulkarni; Sophia Chu; Ekarat Buddharuksa; Ruben Ponce
Sat 12/17/2022 9:40 PM

Hello Team,

Thank you everyone for this learning experience. I apologize for my lack of participation at the last week of the semester. I caught the flu during finals week, and it has been a miserable experience. My allergies also kicked in at the same time making it difficult to keep my eyes open when taking tests or using the computer.

My contribution:

I was the back-end team leader. I helped develop our back-end APIs and endpoints:

-ER Diagram
-Registration
-Login
-Logout
-Get Email
-Message
-Inbox
-Text Encryption

Number of Submissions: 12 Commits

Main Challenges: