SW Engineering CSC648/848 Fall 2022

SFSU MEDIA STORE

TEAM 3

Himani Varshney (Team Lead) hvarshney1@sfsu.edu

Donnovan Jiles (Back-end Lead)

Olimpia Aguillon (Front-end Lead)

Josef Fiedler (Github Master)

Yasaman Pakdel

Robert Peter Swanson

Milestone 4

12/09/2022

Revisions History

Date Submitted	Date Revised
12/09/2022	

1) PRODUCT SUMMARY:

Name of the product – SFSU Media Store

When we joined this college, we needed to get access to the previous year recordings of the courses we had enrolled for. We went and checked on iLearn which is a go to platform for any course related contents. But unfortunately, it contains the content of the present semester only. We then had to ask my peers in college to share any recordings they had. This is when we decided to develop and launch a web platform "**SFSU Media Store**" for sharing media which is used only by SFSU students and faculty. Users can buy and sell any kind of digital media such as images, audios, videos, etc. on our web platform. It will be like a one stop store for all SFSU students and faculty to gain access to digital media. Our application will make searching for tutorials and class recordings/ presentations a lot easier since it will all be in one place. It will make the hassle of trying to find different information easier.

Functionalities:

1. Unregistered users will be able to browse items.
2. Unregistered users will be able to search items using media categories.
3. Unregistered users will be able to view item details.
4. Unregistered users will be able to search media using the item's title, description.
5. Unregistered users will be able to register to the website.
6. Registered Users will be able to download the free item.
7. Registered Users will be able to login into their personal account.
8. Registered Users will be able to contact sellers to buy paid media item.
9. Registered Users will be able to post an item.
10. Registered Users will be able to view the dashboard for their posts.
11. Registered Users will be able to view the dashboard for their messages.
12. Admin will approve the appropriate posts (before they go live).
13. Admin will reject the posts (from going live) which are inappropriate.

URL: http://35.88.249.97:3000/

2) Usability test plan:

Test Objectives:
To test the usability of the SEARCH functionality. It will help to measure the efficiency, effectiveness and satisfaction of the search usage for the users. Feedback of the same will help in designing a more user centred design for the product.

Test background and setup –
The user needs to have a Windows/Mac PC. He / She can use Windows OS / Linux OS. The user needs to have a web browser like Google Chrome / Microsoft Edge. The URL of the system to be tested is http://35.88.249.97:3000/Home
The intended users of the system are the SFSU students and the faculty who can have zero to advanced level of computer knowledge. This usability testing is done to measure the effectiveness, efficiency and satisfaction of users using Likert test.

Usability Task description:
The tester should access the URL in the web browser and go to the home page of the website. The user needs to test usability of the search bar by giving different input to the search bar.

Evaluation of Effectiveness:
To measure the effectiveness of the search functionality, we will measure the percentage of people who were able to search for the desired items within 2 minutes. We will also record the count of errors per task by the users.

Evaluation of efficiency:
To measure the efficiency of the search functionality, we will measure the average time users took to search for an item. We will also measure efficiency by the average time in which the image is loaded on click of search.

Evaluation of user satisfaction:
To measure the user satisfaction, we will use the results of the Likert survey.
Likert Survey:

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
It is easy to change category in the search bar.					
I had no problem in finding the search bar.					
I can easily see the number of results					

returned on click of search					

3) QA test plan

Test objectives:
To test the Search functionality of the SFSU Mediastore as per specs defined

HW and SW setup (including URL):
Windows OS
Google Chrome Version 108.0.5359.98
Microsoft Edge Version 108.0.1462.46
URL : http://35.88.249.97:3000/Home

Feature to be tested:
Search Bar

QA Test plan:

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS / FAIL)
0001	Search_test_like	Test % like in search for the search field	Go to the specified URL. On the home page, click on search bar. Type “gator” in the search bar. Click on Search button.	2 items should be displayed having gator in their title	
0002	Search_test_null	Test the search field by giving blank input	Go to the specified URL. On the home page, click on search bar. Do not type	8 items should be displayed.	

			anything in the search bar and click on Search button.		
0003	Search_test_dropdown	Test for the category dropdown in the search bar	Go to the specified URL. On the home page, select ‘Image’ from the Search dropdown menu and enter ‘gator’ in the search bar. Click on Search button.	Only 1 item should be displayed having gator in their title	

QA test results on Google Chrome Version 108.0.5359.98

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS / FAIL)
0001	Search_test_like	Test % like in search for the search field	Go to the specified URL. On the home page, click on search bar. Type “gator” in the search bar. Click on Search button.	2 items should be displayed having gator in their title	PASS
0002	Search_test_null	Test the search field	Go to the specified URL. On	8 items should be displayed.	PASS

		by giving blank input	the home page, click on search bar. Do not type anything in the search bar and click on Search button.		
0003	Search_test_dropdown	Test for the category dropdown in the search bar	Go to the specified URL. On the home page, select 'Image' from the Search dropdown menu and enter 'gator' in the search bar. Click on Search button.	Only 1 item should be displayed having gator in their title	PASS

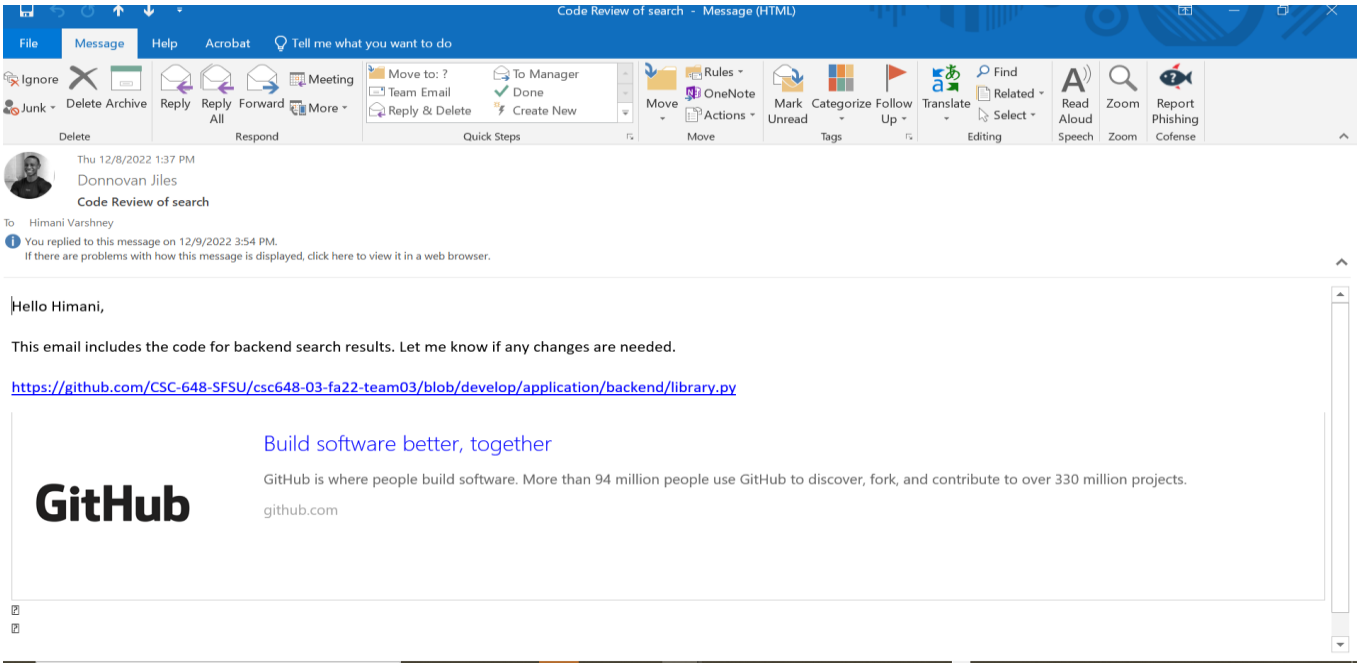
QA test results on Microsoft Edge Version 108.0.1462.46

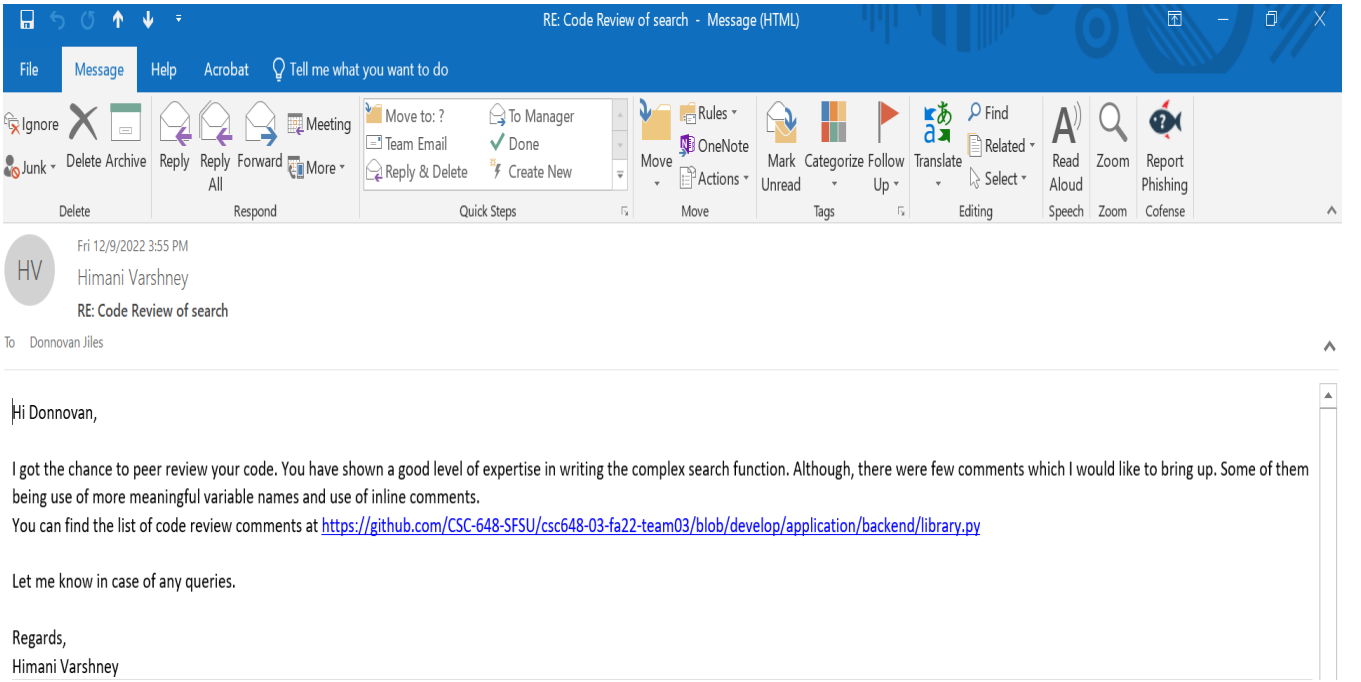
Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS / FAIL)
0001	Search_test_like	Test % like in search for the search field	Go to the specified URL. On the home page, click on search bar. Type "gator" in the search bar. Click	2 items should be displayed having gator in their title	PASS

			on Search button.		
0002	Search_test_null	Test the search field by giving blank input	Go to the specified URL. On the home page, click on search bar. Do not type anything in the search bar and click on Search button.	8 items should be displayed.	PASS
0003	Search_test_dropdown	Test for the category dropdown in the search bar	Go to the specified URL. On the home page, select ‘Image’ from the Search dropdown menu and enter ‘gator’ in the search bar. Click on Search button.	Only 1 item should be displayed having gator in their title	PASS

4) Code Review:

Back end code for Search -



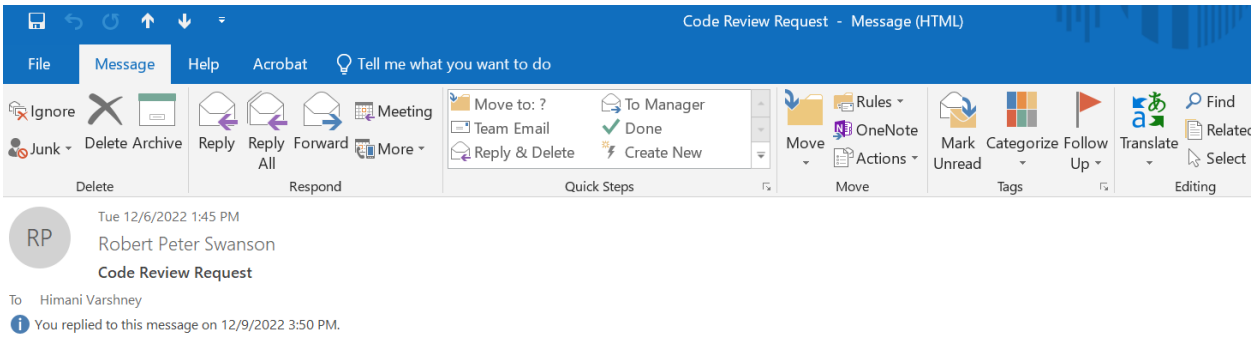


Github Repo screenshot:

```
'''
    Code reviewed by - Himani Varshney
    Comments - 1) Remove print statements before deploying the code on server.
               2) Use more meaningful variable names.
               3) Use more inline comments for each if else condition.

'''
#endpoint for search
@app.route('/search', methods=['GET', 'POST'])
def search():
    print("Hello World")
    if request.method == "POST":
        print(request)
        # request user input from <input> with name="book"
        print(dir(request))
        print("########################")
        #request.json['book']
        #print(request.body)
        #book = request.form['book']
        book = request.get_json()
        print(request.get_json())
        print(book['book'])
        print("########################")
        conn = mysql.connect()
        cursor = conn.cursor()
        if(book['book']=="" and book['Category'] == 'all' ):
            cursor.execute("SELECT item_title,item_description, user_username, item_path, item_price, item_category FROM item JOIN user_records ON item_creator_id = u
            conn.commit()
            data = cursor.fetchall()
            print(data)
```

Front end code for Search :



RP

Tue 12/6/2022 1:45 PM

Robert Peter Swanson

Code Review Request

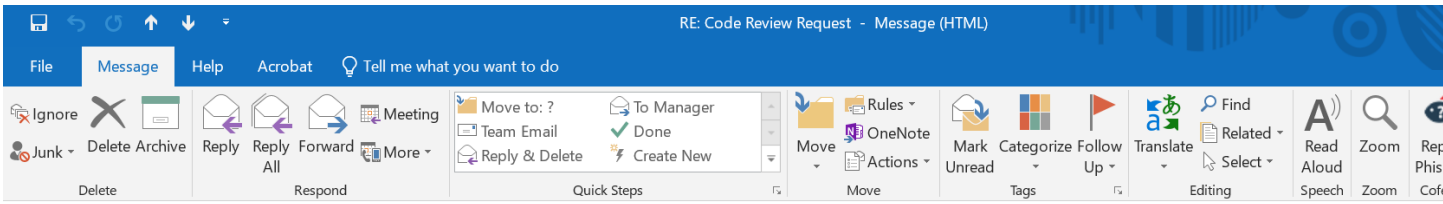
To Himani Varshney

You replied to this message on 12/9/2022 3:50 PM.

Hello Himani,

Please review the code for search results and let me know if any changes are needed.
https://github.com/CSC-648-SFSU/csc648-03-fa22-team03/blob/develop/application/client/src/NavBar.js

Best,
Robert Swanson



HV

Fri 12/9/2022 3:51 PM

Himani Varshney

RE: Code Review Request

To Robert Peter Swanson

Hi Robert,

I got the chance to peer review your code. I must say you have written very good quality code by showing good use of header comments and inline comments.
You can find the code review comments on the file at https://github.com/CSC-648-SFSU/csc648-03-fa22-team03/blob/develop/application/client/src/NavBar.js
Please remove commented code blocks before deployment and use a separate file to store static urls.
Rest everything looks good.

Regards,
Himani Varshney

Github Repo screenshot:

```
1  /*
2   * File: NavBar.js
3   * Author: Robert Swanson
4   * Description: NavBar for webapp with search functionality and links to About Us, Post,
5   *              Dashboard, Sign in, and register. And contains disclaimer that this is student
6   *              project
7   */
8
9  /*
10 Code reviewed by - Himani Varshney
11 Date - 12/09/2022
12 Comments - 1) Please remove commented block of code before moving to deployment server.
13            2) Good use of header comments and inline comments
14            3) All URL can be moved to one place and imported from there.
15 */
16 import { Link, useNavigate } from "react-router-dom"
17 import { useEffect,useState } from 'react';
18
19 const NavBar = () => {
20     const [category, setCategory] = useState("all");
21     const [searchText, setSearchText] = useState("");
22     const navigate = useNavigate();
23 /*
24     function setCategories() {
25         //Categories return from DB goes here
26         const categories = { 'cat': ['Audio', 'Video','Image', 'Class'] }; //Hard coded for testing
27         /*
28         const options = []; //Array of <option> to be returned to dropdown
```

5) Self - check on best practices for security –

Asset to be protected	Types of possible / expected attacks	Strategy to protect / mitigate the asset
Media items	Loss / Theft of media	All the media items are securely stored on the file disk of server, where in only admin has the access to the server and no one else
User Information	Theft of user's personal data, SQL Injection	DB is on the server with only admin having the access. All the confidential data like password is encrypted. Input Validation (like limiting input length to 40 alpha numeric characters) is done to prevent SQL injection. Only users with sfsu.edu can access the website.

Encrypt PW in the DB – DONE using md5 encryption algorithm
Input data validation –
search bar input for up to 40 alphanumeric characters – ON TRACK
registration e-mail to include "sfsu.edu" at the end – DONE

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

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 – On track
2. 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 – On track
6. Privacy of users shall be protected - Done
7. The language used shall be English (no localization needed) - Done
8. Application shall be very easy to use and intuitive – Done
9. Application should follow established architecture patterns – Done
10. Application code and its repository shall be easy to inspect and maintain – On track
11. Google analytics shall be used – On track