SW Engineering CSC648/848 Fall 2021

TwoHelpYou

Project Application And Name: Provide tutoring to SFSU students – "TwoHelpYou."

Team Number: Team 01

Team Members and Roles:

1. Justin Lam (Team Lead) - jlam18@mail.sfsu.edu
2. Wesley Xu (Front-End Lead) - wxu3@mail.sfsu.edu
3. Dinesh Thapa (Back-End Lead) - dthapa@mail.sfsu.edu
4. Aviral Puri (Github Master) - apuri2@mail.sfsu.edu
5. Chung Hei Fong
6. Kurt Resayo

Milestone: Milestone 4

Date: December 10, 2021

History:

Version	**Date**
M4V1	December 10, 2021

Table of Contents

1. Product Summary

Name of the product: TwoHelpYou - A tutoring site was developed to help SFSU students.

Product summary

1. Registration
   - Users are able to register an account with the website.
2. Login
   - Users are able to log in with their accounts and utilize website features.
3. Search
   - Users can use the search bar function to search for offered courses.
4. Search results
   - Executing the search show posts regarding the query to the user.
5. Post
   - Registered users are allowed to post offers to provide tutoring to other users.
6. Messaging
   - Registered users can send messages to other users offering to tutor.
7. Dashboard
   - Logged in users will see a dashboard that displays incoming messages and their posts.

URL of our website: http://3.17.38.143:3000/

2. Usability Test Plan

Test objectives:
One primary function that will be tested is the post form function, in which registered users can post their availability and which subjects they can help other users by providing tutoring sessions. This is being tested because we want to see if registered users can access the page, fill out the necessary information, submit a post. Following the submission, a moderator will review their submitted information and accept or deny their post. If a non-registered user accesses the page, fills out information, and hits the submit button, they will be met with an alert saying they either need to be logged in or register an account for the post to be posted submitted and reviewed. This function must be usable and bug-free since it is one of the critical core functions of the website for registered users to use and make posts when they need it. The post is crucial to the overall functionality of the website. It is the one feature that has to work to populate the website with the required content to publicize to peruse and make decisions on further interaction. Without the post's functionality, no tutoring sessions would be offered on the website. Thus if the post function does not work, users on the site will go to other competitor tutoring sites leading to a domino effect of the tutoring site losing web traffic. Eventually, it will get shut down to no one going to it, then everyone who works on the site will be out of the job due to lack of budget. We want to avoid that from happening, so the post function must be tested thoroughly to see if it is usable and check for any issues. Such issues consist of bugs that need to be ironed out, any exploits that can be stopped, and any quick improvements that need to be implemented before the official release.

Test background and setup:
For the system setup for the tester to use, it is recommended that they use a working computer that can connect to the internet. The operating systems that can be used are Linux, Windows, or macOS; if it's usable. Then they need to choose at least two website browsers to test the site on to see if it works on multiple browsers, i.e., Google Chrome, Firefox, Edge, to name a few.
At the starting point, the user will see the website and a button for posting availability in a subject they can offer to tutor. They press the button and see the information they need to fill out, and once they submit all the required information, they then hit the submit button. An alert will pop up saying they need to be signed in or create an account to post availability for tutoring. Once the user signs in or registers for an account, they can go back to the post form page with all their information saved then hit the submit button. After the user hits the submit button, the information will then be reviewed by the site's moderator, and the request will be either approved or denied based on the user's qualification to be a tutor, their availability and the course they want to tutor.
The intended users of this site are SFSU students who need help with homework and topics in specific courses they are struggling with. Also, it is intended for SFSU students who want to tutor others in said homework and topics in particular courses.
The URL of the system to be tested is 3.17.38.143:3000/post_form.
What needs to be measured is to see if the post form function can process and send information to the website moderator, how accessible it is, and if the user is able to navigate through it easily.

Usability Task description:
Instructions for the tester: Follow the link to the website, go to the post form page by clicking on the post button, fill out the requirements, upload a document, then hit the submit button.

Evaluation of Effectiveness:
Effectiveness would be measured by if the user can fill out the post form and submit it, how many users make mistakes filling it out, and if users can upload a document.

Evaluation of efficiency:
Efficiency is measured by how long it takes to fill out the post form, how long it takes for users to find it, and to see the number of clicks it takes to fill it out and submit it.

Evaluation of user satisfaction:

	Strongly agree	Agree	Neutral	Disagree	Strongly Disagree
Filling out the post form is easy					
The instructions are easy to understand					
The post form is visually organized					

3. QA Test Plan

Test objectives: The ability of a registered and logged-in user to find, fill out and submit the tutor post form. The security against non-registered users submitting forms. The security against invalid inputs in form submission.

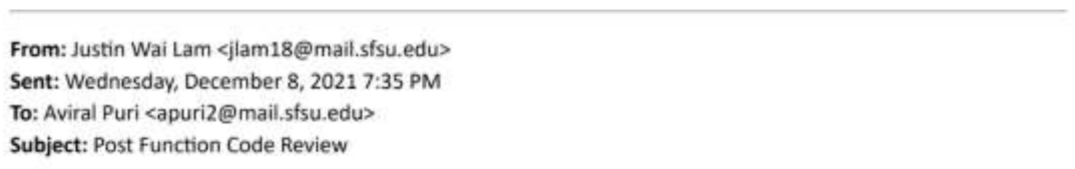
HW and SW setup (including URL):
- Access to an Internet connection
- Requires the installation of at least 2 web browsers.
    - Google Chrome: https://www.google.com/chrome/
    - Microsoft Edge: https://www.microsoft.com/en-us/edge
- Website URL: http://3.17.38.143:3000/
- Required Files: Using this link - https://constitutioncenter.org/media/files/constitution.pdf. Download the pdf.
- After installing the above-listed web browsers, click on the website URL, which will navigate you to the homepage of TWOHELPYOU. This will be the starting point of all tests.
- Expected Time: 30 minutes

Feature to be tested: Tutor Post Feature

Test#	Title	Description	Input	Expected Output	Result
1	Non-Registered User	Main Objective: confirm that non-registered users are not able to submit post forms.	Click the Post button on the navbar.	Tutor Application Page is displayed	PASS
			In the first the input field select "ACCT"	No errors should be displayed	PASS
			In the second input field enter "101"	Confirmation message should change to green	PASS
			Using the third input field enter "Mon-Fri 9am-5pm"	The input field labeled should gain a green border.	PASS
			In the fourth input box choose the constitution.pdf	Constitution.pdf should be displayed next to the choose file button.	PASS
			Click the submit button near the page bottom	A pop-up to login or register should display	PASS
2	Invalid Input	Main Objective: Confirm that forms cannot be submitted with invalid inputs	Click the login button on the right side of the navbar.	The website opens a pop-up window for login	PASS
			Input "tuser" into the input field labeled "Username"	A green message is displayed above the input field	PASS
			Input "Password!123" into the input field labelled "Password"	A green message is displayed above the input field	PASS
			Click the Login button	The pop up disappears and displays a green confirmation message.	PASS

			Click the Post button on the navbar.	Tutor Application Page is displayed	PASS
			In the first the input field select "ACCT"	No errors should be displayed	PASS
			Using the second input field labeled enter "11"	Message under the input label should change color to a red message	PASS
			In the third input field labeled enter "Mon"	The input field should gain a red border.	PASS
			Using the fourth input box choose the constitution.pdf file	Constitution.pdf should be displayed next to the choose file button.	PASS
			Click the submit button at the bottom of the page	The input box should warn the user and not submit.	PASS
3	Valid Input	Main Objective: Confirm that forms with valid input are properly submitted and able to be reviewed by the moderators.	Click the login button on the right side of the navbar.	The website opens a pop-up window for login	PASS
			Input "tuser" into the input field labeled "Username"	A green message is displayed above the input field	PASS
			Input "Password!123" into the input field labelled "Password"	A green message is displayed above the input field	PASS
			Click the Login button	Pop up disappears and shows a green message.	PASS
			Click the Post button on the navbar.	Tutor Application Page is displayed	PASS
			In the drop down on the first input field select "ACCT"	No errors should be displayed	PASS
			In the second input field enter "101"	Input label should change color and display a green message	PASS
			Enter "Mon-Fri 9am-5pm" into third input field.	The input field should gain a green border.	PASS
			In the fourth input box choose the constitution.pdf	Constitution.pdf should be displayed next to the choose file button.	PASS
			Click the submit button at the bottom of the page	Should display the home page with a green confirmation message	PASS
			Await moderator to approve post	Post should appear on the home page	PASS

4. Code Review



From: Justin Wai Lam <jlam18@mail.sfsu.edu>
Sent: Wednesday, December 8, 2021 7:35 PM
To: Aviral Puri <apuri2@mail.sfsu.edu>
Subject: Post Function Code Review

Good Evening Avi,

I hope things are going well this semester for you.

For Milestone 4, we have the code review portion. I believe we are ready for QA and Usability testing regarding the code related to the Post functionality. The functions associated with Post need to be reviewed as they will be the focus of the testing. These are the files that represent most of the backend code relating to Post.

Please review these files, check if they adhere to coding conventions and readability standards:

- csc648-03-fa21-team01/post.js at main · CSC-648-SFSU/csc648-03-fa21-team01 (github.com)
- csc648-03-fa21-team01/post_model.js at main · CSC-648-SFSU/csc648-03-fa21-team01 (github.com)
- csc648-03-fa21-team01/post_middleware.js at main · CSC-648-SFSU/csc648-03-fa21-team01 (github.com)

Thank you for the hard work!

Regards,
Justin Lam

Re: Post Function Code Review

Aviral Puri <apuri2@mail.sfsu.edu>

Wed 12/8/2021 7:39 PM

To: Justin Wai Lam <jlam18@mail.sfsu.edu>

Hi Justin,

I have completed code review of the files related to backend post functionality. My comments and feedback have been posted to the code-review-post branch. Below is a brief overview of my findings followed by a detailed checklist of conventions for each file reviewed.

Overview:
Overall the code looks great! It is very readable and functions to specs. The github comments are also very descriptive and make it very easy to follow the development process. Some of the header and function descriptions could use a little more detail but it not absolutely necessary. The only issues found were a few variable names across several files that may have been written before coding conventions were set up and will need to be updated to match the current convention.

Breakdown by file:
Post_Middleware.js:

- Header is present and contains the relevant information about the file
- All functions have comments with parameters, return and descriptions
- Variable and function names are consistent and follow conventions
- Github commit comments are descriptive and accurate

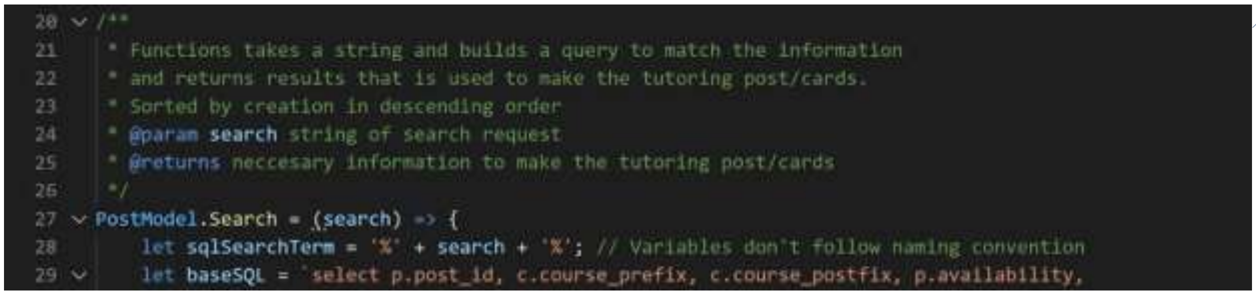
Post_Model.js:

- Header is present and contains the relevant information about the file
- A few variable names don't follow coding conventions such as: sqlSearchTerm and baseSQL
- All functions have comments with parameters, return and descriptions
- Github commit comments are descriptive and accurate

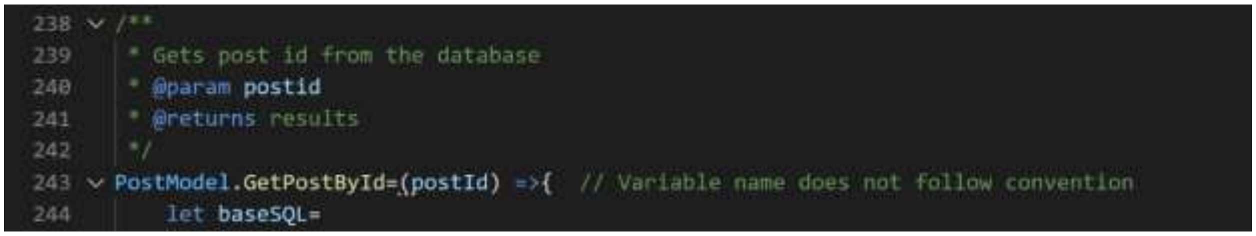
Post.js:

- Header is present and contains the relevant information about the file
- A few variable names don't follow coding conventions such as: searchQuery and searchTerm
- Variable and function names for the rest of the file are consistent and follow conventions
- All functions have comments with parameters, return and descriptions
- Github commit comments are descriptive and accurate

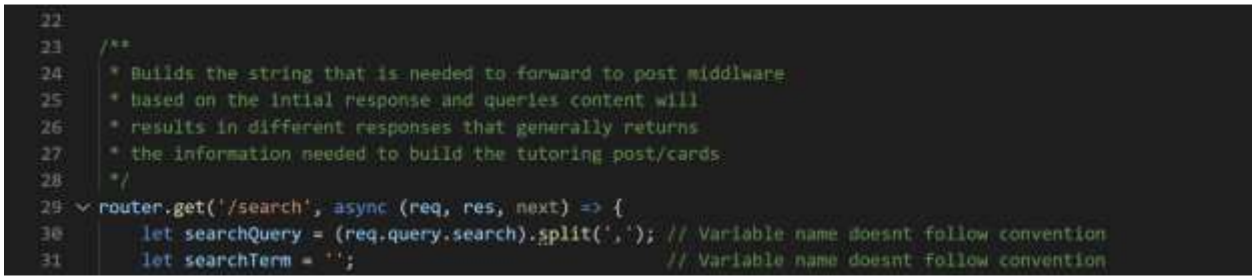
Thanks and regards,
Aviral Puri



```
/**
 * Functions takes a string and builds a query to match the information
 * and returns results that is used to make the tutoring post/cards.
 * Sorted by creation in descending order
 * @param search string of search request
 * @returns neccesary information to make the tutoring post/cards
 */
PostModel.Search = (search) => {
    let sqlSearchTerm = '%' + search + '%'; // Variables don't follow naming convention
    let baseSQL = `select p.post_id, c.course_prefix, c.course_postfix, p.availability,
```



```
/**
 * Gets post id from the database
 * @param postid
 * @returns results
 */
PostModel.GetPostById=(postId) =>{  // Variable name does not follow convention
    let baseSQL=
```



```

/**
 * Builds the string that is needed to forward to post middlware
 * based on the intial response and queries content will
 * results in different responses that generally returns
 * the information needed to build the tutoring post/cards
 */
router.get('/search', async (req, res, next) => {
    let searchQuery = (req.query.search).split(','); // Variable name doesnt follow convention
    let searchTerm = '';                             // Variable name doesnt follow convention
```

5. Self-check on best practices for security

Asset to be protected	**Types of possible/expected attacks**	**Your Strategy to mitigate/protect the asset**
Website Server (AWS)	DDOS attack	• AWS built-in security service.
	Unauthorized access to AWS.	• Strong password protected. • An SSH key is required to access AWS.
The database	Unauthorized attempt to access the database.	• The default password has been changed to a more secure password. • Current access does not use root as user.
	SQL injection.	• Search request does not permit requests longer than 40 characters. Any attempt to do so will provide a message failing to execute. • Most of the input fields utilize frontend and backend validation before execution.
User account info within the database	Password theft.	• All password within the database is encrypted using bycrypt. • Special format for password registration.
	Unauthorized user registration.	• User email is required to include "sfsu.edu" or "mail.sfsu.edu" in registration.
Tutor post data	Corruption and attempts to access the user accounts to post inappropriate content.	• Before the public can see a user post offering tutoring, the admin will need to review and approve all post.

6. Self-check: Adherence to original Non-functional specs

#	Non-Functional Requirement	Status
1	Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in Milestone 0. Application delivery shall be from chosen cloud server.	DONE
2	Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.	DONE
3	All or selected application functions must render well on mobile devices.	DONE
4	Data shall be stored in the database on the team's deployment cloud server.	DONE
5	No more than 50 concurrent users shall be accessing the application at any time	DONE
6	Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.	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	DONE
11	Google analytics shall be used	DONE
12	No email clients shall be allowed.	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	Application shall be media rich (images, video etc.). 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	For code development and management, as well as documentation like formal milestones required in the class, each team shall use their own github to be setup by class instructors and started by each team during Milestone 0	DONE

18	The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2021 For Demonstration Only" at the top of the WWW page. (Important so as to not confuse this with a real application)	DONE