

SW Engineering CSC 648/848 Fall 2023

Team 3

Dan Tahir - Team Lead

dtahir@sfsu.edu

Cleveland Plonsey

Ava Albert

Michael Mathews

Griffin Evans

Bryan Maldonado

Milestone 4

12/13/23

History

Date Submitted	Date Revised
12/13/2023	

1. Product summary

Name:

SF State Tutors.Tech

Description:

The modern college educational experience is more complex than ever, with students expected to take on ever-greater challenges in order to earn their degrees. Students struggle to make the key connections they need to make in order to succeed. But what if there was a service that helped bridge the gap between students to enable them to help each other?

A tool that equips students with the expertise and practical know-how to provide assistance to fellow students requiring support. And what if that service were geared explicitly towards the students of one university, with services unique to their campus? Enter SF State Tutors.Tech.

SF State Tutors.Tech intends to be THE place where SFSU tutors and SFSU clients connect on the SF State campus. We intend to have an easy onboarding process for tutors whereby they can register and sign up as tutors in one click, or register as students and then onboard as tutors later. We intend searching for tutors to be easy for anyone, with an intuitive display of each tutor's credentials, rates and other information. We plan to enable messaging between tutors and students for registered students. We plan to have SFSU-specific features, such as only allowing students and tutors to register if they have an SFSU email address, and including maps of campus to enable students to meet up with tutors more easily. We also plan to have reviews of tutors by fellow SF State students so students will be able to trust the reputation of their tutor. With these great features we believe we can capture the market for tutors on the SFSU campus.

Final list of P1 Items:

Unregistered Users

1. User shall be able to view the home page
2. User shall be able to browse all tutors that are verified by admin
3. User shall be able to search for tutors
4. User shall be able to view information on a list of tutors
5. User shall be able to view a page for each tutor including the tutor's description and media
6. User shall be able to view reviews of tutors
7. User shall be able to view an average score for tutors
8. User shall be able to sign up

Registered Users

9. User shall inherit privileges from unregistered users
10. User shall be able to sign in
11. User shall be able to message tutors
12. User shall be able to become a tutor
 - 13.1 User shall be able to submit data including photo, subject, classes taught, bio, video, and pdfs
13. User shall be able to view an account page containing messages sent to them and by them
14. User shall be able to exchange new messages with people who have messaged them or whom they have already messaged
15. User shall be able to review tutors

Admins

16. Admin shall be able to view entries in the database in MySQL Workbench
17. Admin shall be required to approve tutors before their results are visible
18. Admin shall be required to delete accounts that violate terms of service
19. Admin shall only be permitted to approve and delete tutors
20. Admin shall be able to edit the database in MySQL Workbench

2. Usability test plan for selected function

Test Objectives:

We are testing the search function to ensure that users can effectively and efficiently locate tutors for various subjects. Our specific instructions to users are: Search SF State Tutors.Tech for a tutor for Math 225.

Test Background and Setup:

User needs: A laptop or phone with the latest version of Chrome or Firefox that is capable of connecting to the internet.

Intended users: Students at San Francisco State University.

URL of the system to be tested: <https://tutors.tech>

Test environment: In an office, with cameras, with monitoring, with no training before the test.

Plan for Evaluation of Effectiveness:

We will determine:

- How many users found the search bar
- How many users used the dropdown menu on the search bar
- How many users used the text field in the search bar
- How many users successfully searched for a tutor and viewed results
- How many users clicked through to the tutor's information page
- How many users attempted to send a message
- How many users created an account to send their message

Plan for Evaluation of Efficiency:

We will determine:

- How long it took users to find the search bar
- How long it took users to click the dropdown menu (if they clicked it)
- How long it took users to access the text search (if they used it)
- How long it took them to search and view results
- How long it took them to click through to a tutor
- How long it took them to find a tutor in their subject
- How long it took them to send a message
- How long it took them to sign up to send the message

Plan for Evaluation of User Satisfaction:

Usability Task Description: Search SF State Tutors.Tech for a tutor for Math 225.

We will ask users to evaluate the following Likert scale evaluation entries:

1. I found SF State Tutors.Tech easy to use
Strongly Disagree Disagree Neutral Agree Strongly Agree
2. I found SF State Tutors.Tech pleasant to use
Strongly Disagree Disagree Neutral Agree Strongly Agree
3. I was successful in using SF State Tutors.Tech
Strongly Disagree Disagree Neutral Agree Strongly Agree

3. QA test plan and QA testing

Test Objectives:

To QA test the search function

HW and SW Setup:

Dell Inspiron 5515 Windows 11 laptop running Chrome and Firefox on <https://tutors.tech/>

What is being tested:

Search

QA Test Plan:

Number	Description	Test input	Expected output	Pass/Fail
1	Test the seach bar	Enter "Math" in the search bar and click the search button	5 posts showing the Math subject	PASS
2	Test empty search	Leave the dropdown set to "All" and the search field empty and click the search button	15 posts comprising all site tutors	PASS
3	Test partial matches in the name field	Leave the dropdown set to "All" and enter "Ja" in the search bar	1 result for tutor "Jason Vance"	PASS
4	Test the search sort	Leave the dropdown set to All and leave the search bar empty. Click the search button. Use the sort dropdown to change the sort to Review.	15 results with tutors going from the highest rating to the lowest rating	PASS
5	Test the subject filter	Set the dropdown to "Math" and click Search	Five posts for subject Math	PASS

Test Case: Search Function

- 1) Open the Application
- 2) Locate the Search Bar (Centered at the Top of the Site Page)
- 3) Enter "Math" in the Search Bar
- 4) Initiate the Search
- 5) Search Yields 5 Results Showing the Posts with the 'Math' Subject

Test Case: Empty Search

- 1) Open the Application
- 2) Locate the Search Bar (Centered at the Top of the Site Page)
- 3) Leave the Search Bar Empty
- 4) Initiate the Search
- 5) Search Yields 15 Results Consisting of all User Posts

Test Case: Partial Search

- 1) Open the Application
- 2) Locate the Search Bar (Centered at the Top of the Site Page)
- 3) Enter 'Ja' in the Search Bar
- 4) Initiate the Search
- 5) Search Yields 1 Result Consisting of Tutor 'Jason Vance'

Test Case: Sort Search

- 1) Open the Application
- 2) Locate the Search Bar (Centered at the Top of the Site Page)
- 3) Leave the Search Bar Empty
- 4) Initiate the Search
- 5) Click the Drop Down Next to 'Sort by: "
- 6) Choose Review and Apply Sort
- 7) Click the Up Arrow next to Apply Sort and then Apply Sort
- 8) Search Yields 15 Results with the Tutors having the Highest Rating Going towards Lowest Rating



Test Case: Filter Search Using Subject Filter


- 1) Open the Application
- 2) Locate the Search Bar (Centered at the Top of the Site Page)
- 3) Click on the Drop Down Button Labeled 'All'
- 4) Change the filter to 'Math'
- 5) Leave the Search Bar Empty
- 6) Initiate the Search
- 7) Search Yields 5 Results with all the Posts Having 'Math' as the Subject


4. Peer code review


9

10 + //TODO: pass Conversation component time_stamp prop
which is when the conversation was last active.

 **rubbur** 10 minutes ago 



Make sure to remove the TODO comments when they are implemented. 





 Reply...


104

105 + const sortConvos = () => {
106 + console.log("sorting results");

 **rubbur** 12 minutes ago 

You can remove this console.log since the feature is working. 



 Reply...

107 +
setConvoList([...convoList.sort(comparators[sortType])])
;
108 + }
109 +


```
sortConvos();  
console.log(JSON.stringify(convolist)))}}>Apply  
sort</button>
```



rubbur 12 minutes ago



Remove this console.log too.



Reply...

```
25 + //determine the sender and recipient ids  
26 + const senderId = cookie.get("userId");  
27 + const recipientId =  
    msgs[0].thread_id.split('_')[0] == senderId ?  
    msgs[0].thread_id.split('_')[1] :  
    msgs[0].thread_id.split('_')[0];
```



rubbur 6 minutes ago



Creating more functions to get recipient ID and
the date could make the useEffect more
readable.



Reply...

```
62 + let formattedDate =  
    msg.date_stamp.split(/[- :]/);  
  
63 + formattedDate = formattedDate[1] +  
    "/" + formattedDate[2].split("T")[0] + "/" +  
    formattedDate[0]
```



rubbur 5 minutes ago



formattedDate could also use its own function.



Reply...

1 comment on commit `d475717` Lock conversation

rubbur commented on `d475717` 1 minute ago ...

MessageView.jsx could use more comments to help someone else understand the code.

code review

Inbox x



Cleveland Plonsey

1:58 AM (34 minutes ago)



to me ▾

This commit concerns the finished messaging feature. It spans frontend and backend files...

The feature has been tested, however I'm concerned with the code quality. Please review

[finish message feature - CSC-648-SFSU/csc648-03-fa23-team03@d475717 \(github.com\)](#)

This commit concerns searching through conversations alphabetically (as well as by other criteria). It has been tested but concerned with quality again. Let me know if it looks good as is and I can merge it.

[conversations can be sorted alphabetically - CSC-648-SFSU/csc648-03-fa23-team03@6f80e34 \(github.com\)](#)



Virus-free www.avast.com



Michael Mathews

2:19 AM (12 minutes ago)



to Cleveland ▾

Hey Cleveland,

I reviewed your code and it looks good. There are some console.logs for debugging still present and the files could use some more comments. I left some comments on Github attached to the commits.

- Michael

```
171 + let q = "INSERT INTO tutor_posts (tutor_id, subject, description, flier_url, cv_url,
      video_url, hourly_rate) VALUES (?, ?, ?, ?, ?, ?, ?)";
```



CleveyP 1 minute ago



...

q is not a good variable name. Should be descriptive of the query like: insertNewPostQuery. This is imperative so that if a query breaks, we can control + f for that query. If every query is named q then we cannot easily search for the broken code.



Reply...

```
172 + try {
173 +   const result = await db.query(q, [tutor_id, subject, description, flier_url, cv_url,
      video_url, hourly_rate]);
174 +   console.log("post inserted successfully!");
```



CleveyP now



...

remove this once the controller is working.




Reply...

```
175 +   res.send({ success: true });
176 + }
177 + catch (err) {
178 +   console.log("error inserting post");
179 +   res.send({ success: false, error: err });
180 + }
181 + }
```

```

6
7 + const ConversationList = () => {
8 +   const [convolist, setConvolist] = useState([]);
9 +   const userId = cookie.get('userId');
10 +
11 +   useEffect(() => {
12 +     const fetchConversations = async () => {
13 +       try {
14 +         const response = await axios.post(

```

 ClevyP 1 minute ago
response is not a descriptive variable name. Maybe something more like conversationResult is better.




Reply...

```

15 +     `${process.env.REACT_APP_BACKEND_URL}/getConversations`,
16 +     { userId },
17 +     { withCredentials: true }
18 +   );
19 +   if (response.data.success) {
20 +     setConvolist(response.data.messages);
21 +   } else {
22 +     console.log("Error fetching conversations: " + response.data.errorMessage);

```

 ClevyP 2 minutes ago
instead of console.logging this, the error needs to be displayed on the browser page so that the user can see the error.




Reply...

```

23 +   }
24 +   } catch (error) {
25 +     console.error("Error fetching conversations:", error);

```

 ClevyP 1 minute ago
show this error on the browser page.




Reply...

```

26 +   }
27 +   };
28 +
29 +   fetchConversations();
30 +   }, [userId]);
31 +
32 +   return (
33 +     <div className="ConversationList">
34 +       <h1>Conversations</h1>
35 +       {convolist.map((convo) => (

```

 ClevyP now
should check that convolist is defined before calling a method on it.



Reply...



Please code review these two files Inbox x



Michael Mathews <michael.mathews1999@gmail.com>
to me ▾

1:42 AM (11 minutes ago) ☆ ↶ ⋮

This contains the "createPost" controller function, please review it.

<https://github.com/CSC-648-SFSU/csc648-03-fa23-team03/commit/ba89c1a3bcc61dbe8de935902f7b4da82ca1b19a>

Also, please review my code for the messageView component. It is for fetching data from our backend.

<https://github.com/CSC-648-SFSU/csc648-03-fa23-team03/commit/d4b8ed382930bf333105bbaf7fa9744d0c077a47>



Cleveland Plonsey
to Michael ▾

1:53 AM (0 minutes ago) ☆ ↶ ⋮

Done. The reviews for these commits are on github. Code looked good. The files had headers which is great! A few small checks should be performed on results of http requests to make sure that the object is defined before using the object. And variable names should be more descriptive even if they become kind of long.



Virus-free www.avast.com

↶ Reply

↷ Forward

5. Self-check on best practices for security

Asset to be protected	Types of possible/expected attacks	Your strategy to mitigate/protect the asset
User password	Sql injection, XSS	Express sanitizer middleware, auth middleware, query variables Hash passwords via Bcrypt and don't store salt in db
User email	Sql injection, XSS	Express sanitizer middleware, auth middleware, query variables
User images	Virus files, corruption	Frontend file type validation. Backend chmod against execute privileges (should store in S3 bucket but didnt have time)
User messages	Sql injection, XSS	Express sanitizer middleware, auth middleware, query variables
Pdfs	Virus files, corruption	Frontend file type validation. Backend chmod against execute privileges (should store in S3 bucket but didnt have time)
User videos	Virus files, corruption	Frontend file type validation. Backend chmod against execute privileges (should store in S3 bucket but didnt have time)

All forms where users can input information are validated in the frontend against regex for correctness. And then in the backend, the data is further sanitized and checked via authorization and express sanitization through middlewares that are attached to all routes that deliver user data that interacts with the database in any way. Therefore no malicious user input makes it into any controller. Once in the controller, the data is further verified for correctness via conditional logic.

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 shall render well on mobile devices - **Done**
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. 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 shall 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 e-mail clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application - **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 tools 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 2023. For Demonstration Only" at the top of the WWW page nav bar. (Important so as to not confuse this with a real application). - **Done**