GatorTutoring - Milestone 4

SW Engineering CSC 648-848 Fall 2024 Team 7

Thomas Bercasio <Team Lead> - tbercasio@sfsu.edu
Zarko Cosovic <Backend Lead> - zcosovic@sfsu.edu
Jaylin Jack <Backend Developer> - jjack1@mail.sfsu.edu
Suzanna Li <Frontend Lead> - sli47@sfsu.edu
Ken Chika <GitHub Master and QA Engineer > - kchika@sfsu.edu

Document History:
Submitted Draft: 12/10/24

Table of Contents

Product Summary

Gator Tutoring

This website is designed to help SFSU students find tutors that can help them with whatever subject they are studying. The website has a clean user interface and is easy to navigate. The website also allows new tutors to register as service providers.

The tutoring website allows all visitors to search for tutors either by subject and/or tutor's values like name, description, and even SFSU courses. The visitor can then browse through the selection of tutors and see whatever information they have posted and find one they feel will best fit their needs. If a student would like to contact a tutor they will click our Register link and create an account using their SFSU email. Students using services and tutors providing services are SFSU students.

As for those looking to offer tutor help to SFSU students, follow the same registration steps above and visit our tutor application link. From there you will be able to apply as a tutor. This application is then approved or denied by our DataBase Admin, if approved their information will be added to the database of tutors available on the website.

Major Committed Function List

All site visitors shall be able to:

- Search for tutors by SFSU subject, SFSU course name, tutor's bio and tutor's name.
- View images and videos from tutor profile.

All Registered Users shall be able to:

- Apply as a tutor.
  - Upload a short Biography, Video and Images to the website.
  - Any uploaded content shall be approved by an administrator before being posted to the website.
- Message any tutor.
- Have a dashboard showing all received messages.

Other Functions

- Messages, Tutor Applications, and all data uploads shall be approved/rejected by Admin.

Our Website http://18.118.85.170/

Usability Test Plan

- Test Objectives: Test the effectiveness and the efficiency of the search on the website, as well as the user satisfaction with the search performance. The goals of this test are finding the accuracy and completeness of searching for a tutor, finding the resources expanded in relation to the accuracy and completeness of searching for a tutor, and receiving feedback on the comfort and acceptability of performing a search.
- Test Background and Setup:
    - System setup - The tester should either be on a laptop or mobile device with an internet connection.
    - Intended Testers - Usability testers are SFSU students who are seeking tutoring services.
    - URL - http://18.118.85.170/
    - Test Environment – Home, office, library.
- Usability Task Description: The tester will attempt to search for a tutor on the website using a subject, using a tutor's name, using a tutor's bio, and using all combinations of these options.
- Plan for Evaluation of Effectiveness: Effectiveness will be measured by the percentage of users who are able to successfully search for a tutor.
- Plan for Evaluation of Efficiency: Efficiency will be measured by the number of searches performed to obtain the desired result, and the time it takes to do them.
- Plan for Evaluation of User Satisfaction: The tester will be given a google form to fill out after their attempt at searching for a tutor on the website. This form will have a Likert Questionnaire rating their experience on a 1 - 5 scale. Additionally, this form also provides fields for additional feedback about their experience using the website.
- Link to Google Form for Usability Test: https://forms.gle/m1KHLG68jCJPkUGZ7

QA Test Plan

- Test objectives: - QA Testing will test the functionality of the search function. This test is to ensure that the website is returning the correct output. Test is to be performed using 2 major WWW browsers.
- HW and SW setup (including URL):

- System setup - The tester should either be on a laptop or mobile device with an internet connection.
- Intended Tester - Zarko Cosovic
- URL - http://18.118.85.170/
- Test Environment – Home, office, library.
- URL - http://18.118.85.170/

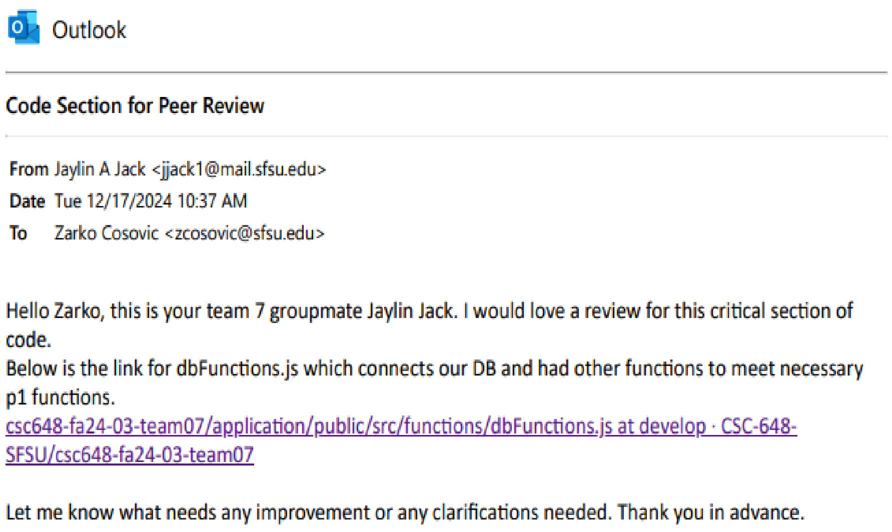
- Feature to be tested: Validation tools to limit the kinds of input a user has.
- QA Test plan:

Test Number	Test Title	Test Description	Test Input	Expected Correct Output	Test Results (PASS/FAIL for each testing browser)
1	Search by Name Test	Test searching for a registered tutor by name	Type “Kim Mari” in the search field and click submit	Get one result of a math tutor, whose name is “Kim Mari”, and the bio states “i love math”	PASS on Chrome Version 131.0.6778.140 (Official Build) (64-bit). PASS on Microsoft Edge Version 131.0.2903.99 (Official build) (64-bit)
2	Search by Subject Test	Test searching for a registered tutor by subject	Type “Math” in the search field and click submit	Get four results of math tutors, whose names are “Kim Mari”, “Ned Flanders”, and “Jerry Shaw”, and “Steve Clifford”	PASS on Chrome Version 131.0.6778.140 (Official Build) (64-bit). PASS on Microsoft Edge Version 131.0.2903.99 (Official build) (64-bit)
3	Search by Name Test	Test search using a name which is not in the database	Type “John Smith” in the search field and click	Get a a list of all 10 tutors available	PASS on Chrome Version 131.0.6778.140

			submit		(Official Build) (64-bit). PASS on Microsoft Edge Version 131.0.2903.99 (Official build) (64-bit)
4	Search by Name and Subject Test	Test search using a Name and Subject at the same time	Type “Ned Flanders” in the name field, and type “Math” in the subject field and click submit	Get one result of a math tutor, whose name is “Ned Flanders, and the bio states “Math major”	PASS on Chrome Version 131.0.6778.140 (Official Build) (64-bit). PASS on Microsoft Edge Version 131.0.2903.99 (Official build) (64-bit)

Report: The website initially failed the first test, which helped locate a bug involving the regex pattern used to validate alphanumeric inputs. After resolving the issue, all tests passed.

Peer Code Reviews

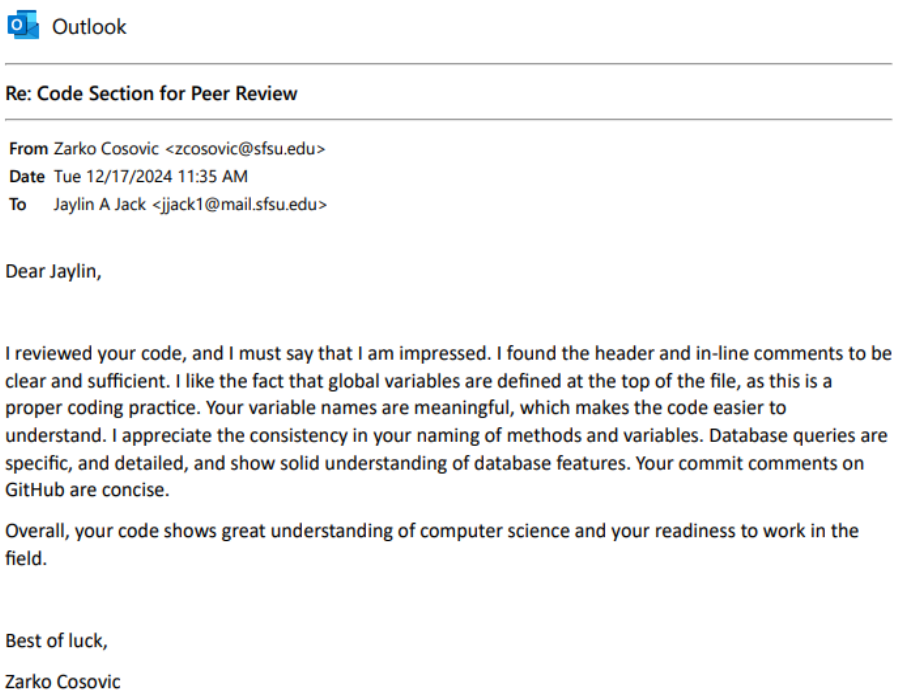


Code Section for Peer Review

From Jaylin A Jack <jjack1@mail.sfsu.edu>
Date Tue 12/17/2024 10:37 AM
To Zarko Cosovic <zcosovic@sfsu.edu>

Hello Zarko, this is your team 7 groupmate Jaylin Jack. I would love a review for this critical section of code.
Below is the link for dbFunctions.js which connects our DB and had other functions to meet necessary p1 functions.
csc648-fa24-03-team07/application/public/src/functions/dbFunctions.js at develop · CSC-648-SFSU/csc648-fa24-03-team07

Let me know what needs any improvement or any clarifications needed. Thank you in advance.



Re: Code Section for Peer Review

From Zarko Cosovic <zcosovic@sfsu.edu>
Date Tue 12/17/2024 11:35 AM
To Jaylin A Jack <jjack1@mail.sfsu.edu>

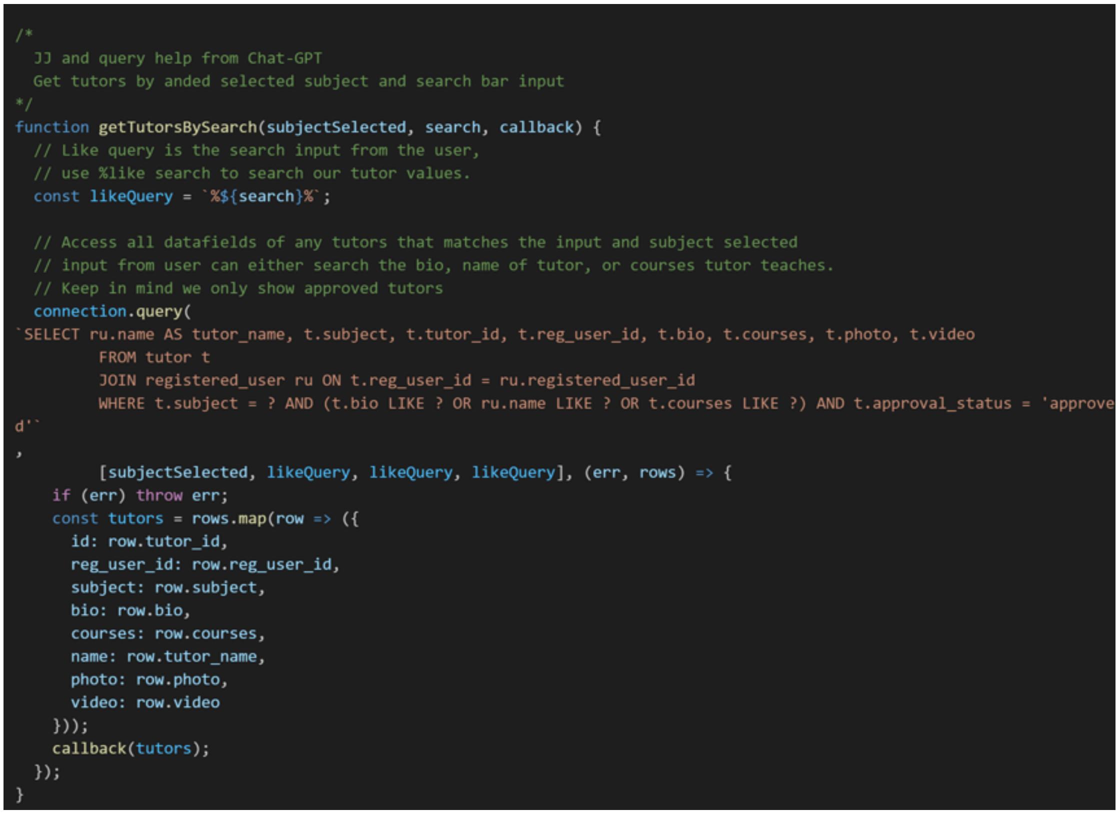
Dear Jaylin,

I reviewed your code, and I must say that I am impressed. I found the header and in-line comments to be clear and sufficient. I like the fact that global variables are defined at the top of the file, as this is a proper coding practice. Your variable names are meaningful, which makes the code easier to understand. I appreciate the consistency in your naming of methods and variables. Database queries are specific, and detailed, and show solid understanding of database features. Your commit comments on GitHub are concise.

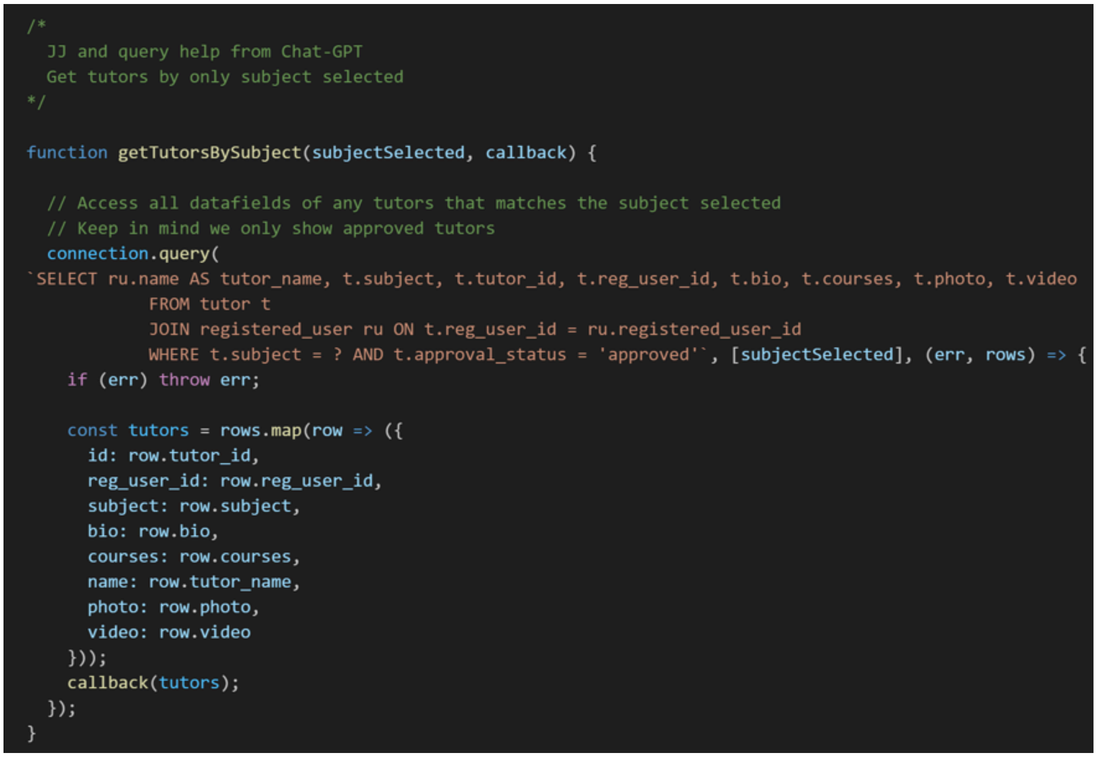
Overall, your code shows great understanding of computer science and your readiness to work in the field.

Best of luck,

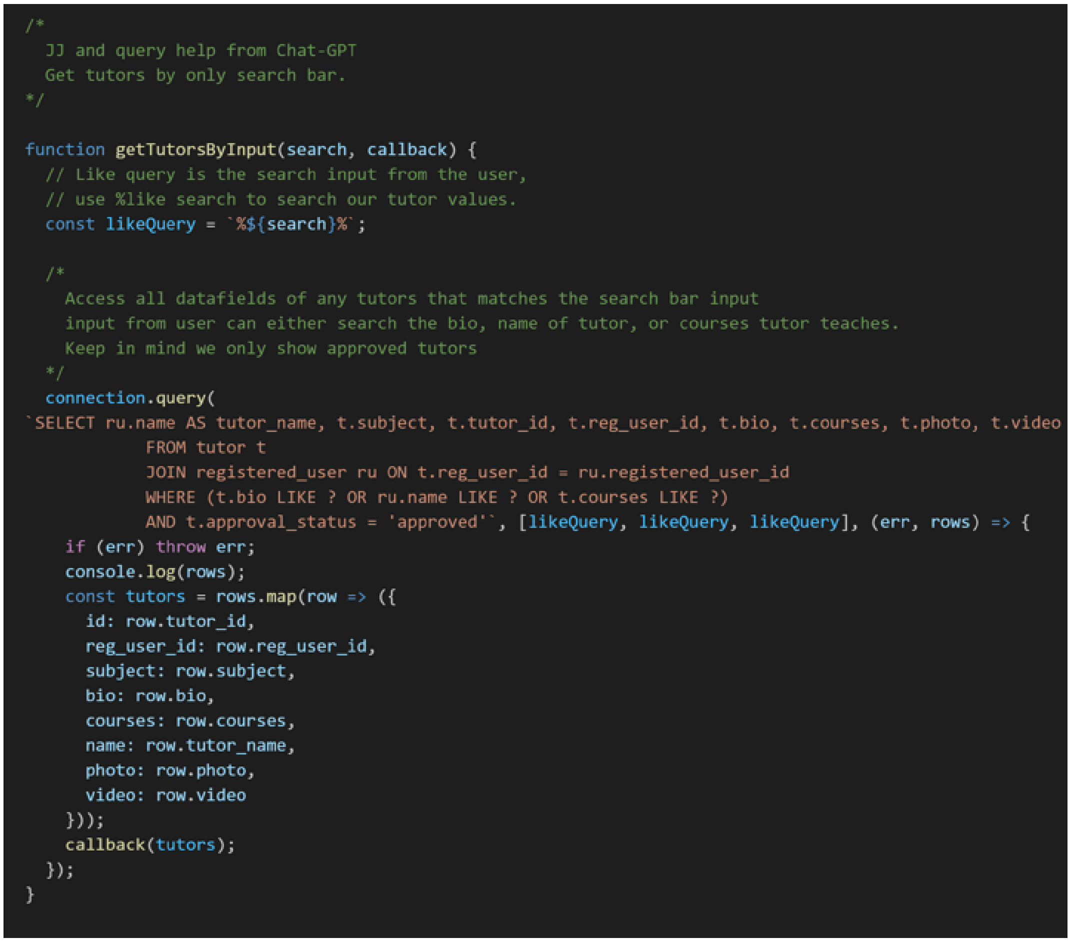
Zarko Cosovic



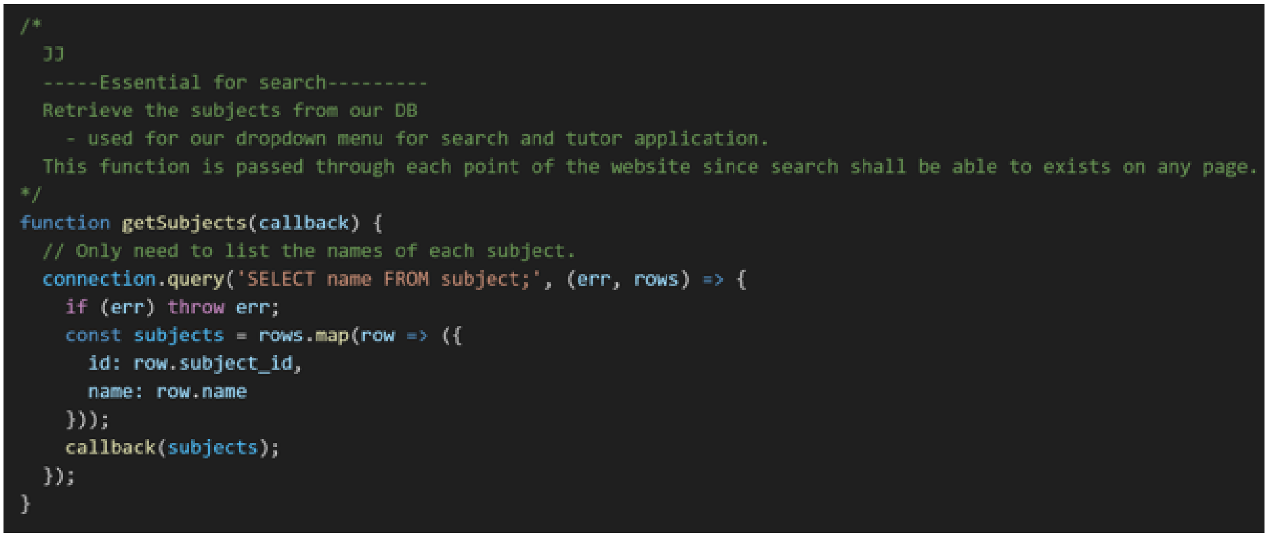
```
/*
  JJ and query help from Chat-GPT
  Get tutors by anded selected subject and search bar input
*/
function getTutorsBySearch(subjectSelected, search, callback) {
  // Like query is the search input from the user,
  // use %like search to search our tutor values.
  const likeQuery = `%${search}%`;

  // Access all datafields of any tutors that matches the input and subject selected
  // input from user can either search the bio, name of tutor, or courses tutor teaches.
  // Keep in mind we only show approved tutors
  connection.query(
`SELECT ru.name AS tutor_name, t.subject, t.tutor_id, t.reg_user_id, t.bio, t.courses, t.photo, t.video
        FROM tutor t
        JOIN registered_user ru ON t.reg_user_id = ru.registered_user_id
        WHERE t.subject = ? AND (t.bio LIKE ? OR ru.name LIKE ? OR t.courses LIKE ?) AND t.approval_status = 'approve
d'`
,
        [subjectSelected, likeQuery, likeQuery, likeQuery], (err, rows) => {
    if (err) throw err;
    const tutors = rows.map(row => ({
      id: row.tutor_id,
      reg_user_id: row.reg_user_id,
      subject: row.subject,
      bio: row.bio,
      courses: row.courses,
      name: row.tutor_name,
      photo: row.photo,
      video: row.video
    }));
    callback(tutors);
  });
}
```



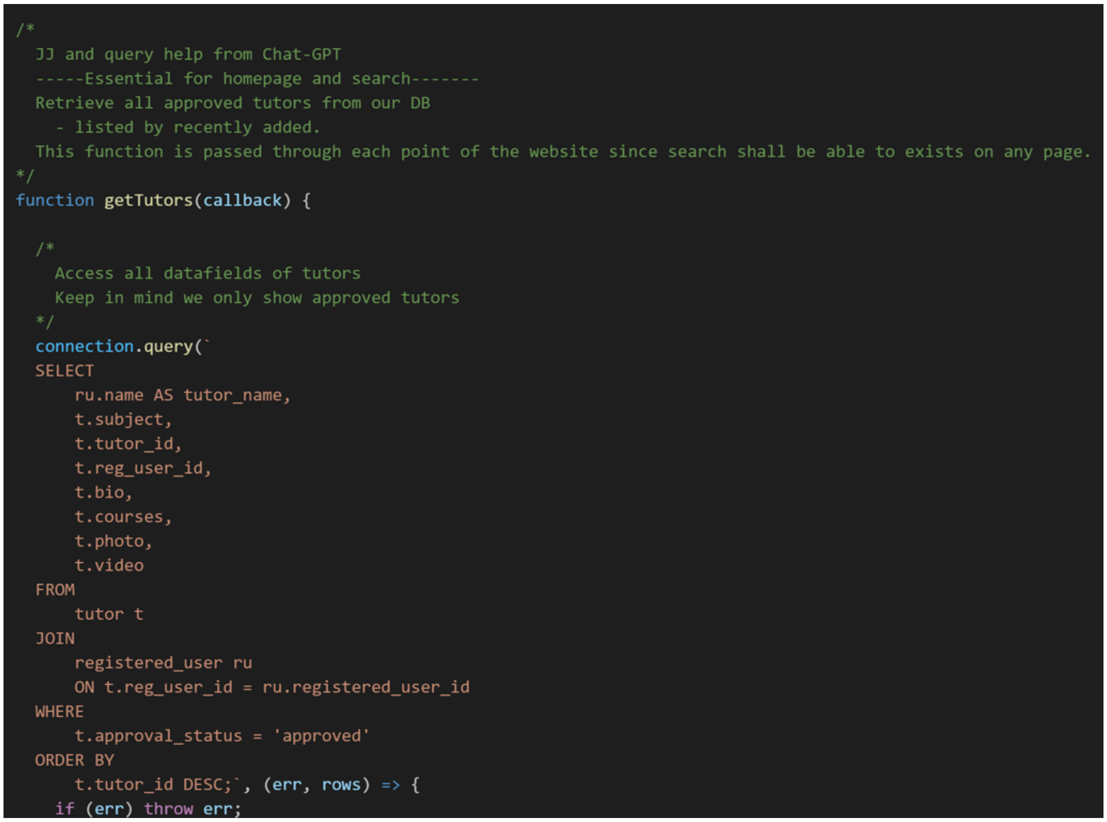
```
/*
  JJ and query help from Chat-GPT
  Get tutors by only subject selected
*/

function getTutorsBySubject(subjectSelected, callback) {

  // Access all datafields of any tutors that matches the subject selected
  // Keep in mind we only show approved tutors
  connection.query(
`SELECT ru.name AS tutor_name, t.subject, t.tutor_id, t.reg_user_id, t.bio, t.courses, t.photo, t.video
            FROM tutor t
            JOIN registered_user ru ON t.reg_user_id = ru.registered_user_id
            WHERE t.subject = ? AND t.approval_status = 'approved'`, [subjectSelected], (err, rows) => {
    if (err) throw err;

    const tutors = rows.map(row => ({
      id: row.tutor_id,
      reg_user_id: row.reg_user_id,
      subject: row.subject,
      bio: row.bio,
      courses: row.courses,
      name: row.tutor_name,
      photo: row.photo,
      video: row.video
    }));
    callback(tutors);
  });
}
```



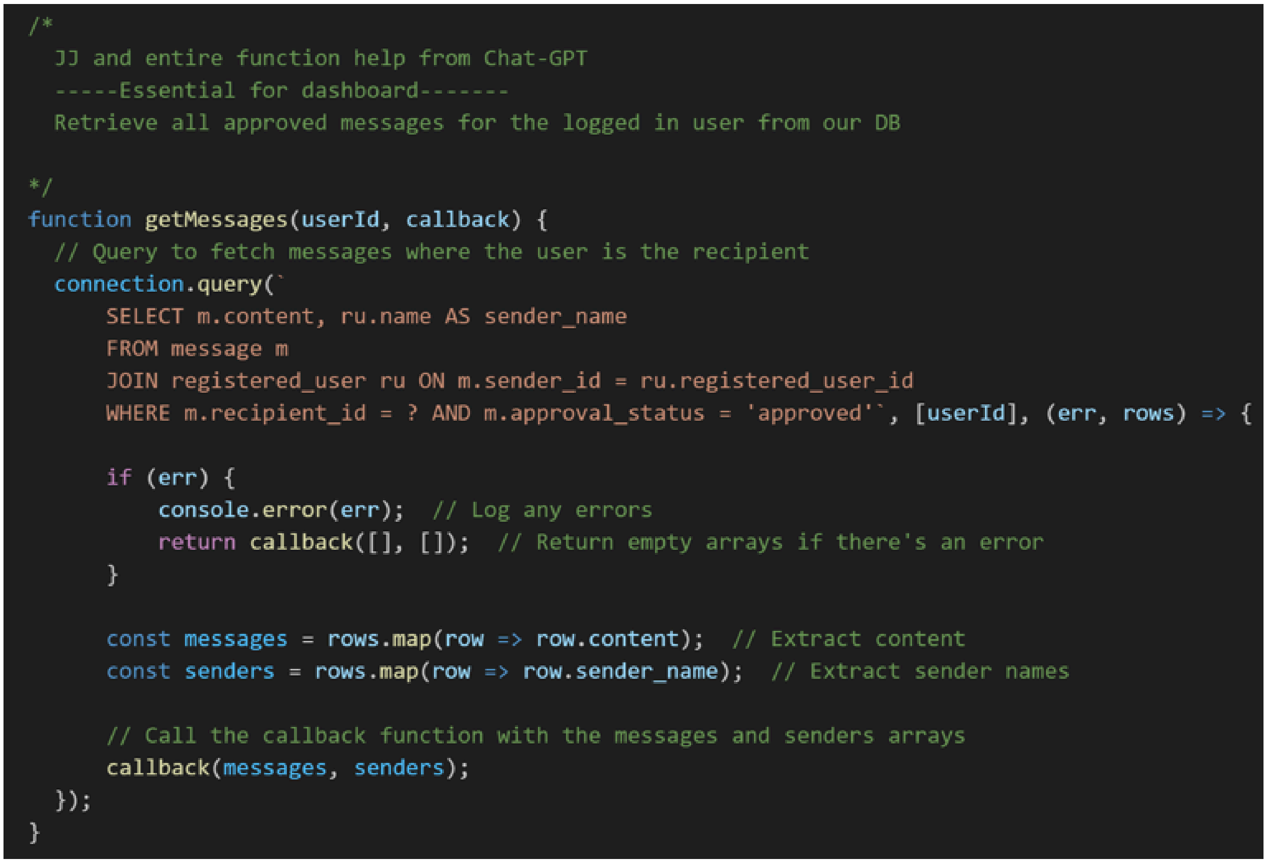
```
/*
  JJ and query help from Chat-GPT
  Get tutors by only search bar.
*/

function getTutorsByInput(search, callback) {
  // Like query is the search input from the user,
  // use %like search to search our tutor values.
  const likeQuery = `%${search}%`;

  /*
    Access all datafields of any tutors that matches the search bar input
    input from user can either search the bio, name of tutor, or courses tutor teaches.
    Keep in mind we only show approved tutors
  */
  connection.query(
`SELECT ru.name AS tutor_name, t.subject, t.tutor_id, t.reg_user_id, t.bio, t.courses, t.photo, t.video
            FROM tutor t
            JOIN registered_user ru ON t.reg_user_id = ru.registered_user_id
            WHERE (t.bio LIKE ? OR ru.name LIKE ? OR t.courses LIKE ?)
            AND t.approval_status = 'approved'`, [likeQuery, likeQuery, likeQuery], (err, rows) => {
    if (err) throw err;
    console.log(rows);
    const tutors = rows.map(row => ({
      id: row.tutor_id,
      reg_user_id: row.reg_user_id,
      subject: row.subject,
      bio: row.bio,
      courses: row.courses,
      name: row.tutor_name,
      photo: row.photo,
      video: row.video
    }));
    callback(tutors);
  });
}
```



```
/*
  JJ
  -----Essential for search---------
  Retrieve the subjects from our DB
    - used for our dropdown menu for search and tutor application.
  This function is passed through each point of the website since search shall be able to exists on any page.
*/
function getSubjects(callback) {
  // Only need to list the names of each subject.
  connection.query('SELECT name FROM subject;', (err, rows) => {
    if (err) throw err;
    const subjects = rows.map(row => ({
      id: row.subject_id,
      name: row.name
    }));
    callback(subjects);
  });
}
```



```
/*
  JJ and query help from Chat-GPT
  -----Essential for homepage and search-------
  Retrieve all approved tutors from our DB
    - listed by recently added.
  This function is passed through each point of the website since search shall be able to exists on any page.
*/
function getTutors(callback) {

  /*
    Access all datafields of tutors
    Keep in mind we only show approved tutors
  */
  connection.query(`
  SELECT
      ru.name AS tutor_name,
      t.subject,
      t.tutor_id,
      t.reg_user_id,
      t.bio,
      t.courses,
      t.photo,
      t.video
  FROM
      tutor t
  JOIN
      registered_user ru
      ON t.reg_user_id = ru.registered_user_id
  WHERE
      t.approval_status = 'approved'
  ORDER BY
      t.tutor_id DESC;`, (err, rows) => {
    if (err) throw err;
```



```
/*
  JJ and entire function help from Chat-GPT
  -----Essential for dashboard-------
  Retrieve all approved messages for the logged in user from our DB

*/
function getMessages(userId, callback) {
  // Query to fetch messages where the user is the recipient
  connection.query(`
      SELECT m.content, ru.name AS sender_name
      FROM message m
      JOIN registered_user ru ON m.sender_id = ru.registered_user_id
      WHERE m.recipient_id = ? AND m.approval_status = 'approved'`, [userId], (err, rows) => {

      if (err) {
          console.error(err);  // Log any errors
          return callback([], []);  // Return empty arrays if there's an error
      }

      const messages = rows.map(row => row.content);  // Extract content
      const senders = rows.map(row => row.sender_name);  // Extract sender names

      // Call the callback function with the messages and senders arrays
      callback(messages, senders);
  });
}
```

Self-Check on Best Practices for Security

Asset to be protected	Types of possible/ expected attacks	Consequence of Security Breach	Strategy to mitigate/protect the asset
User's passwords	Registered User's accounts being hacked.	Unauthorized access to our users will be given.	All passwords are encrypted when entered to our DB.
Terms of service agreement	Lawsuit.	Leaves you liable to be sued.	Registration Page requires TOS agreement.
Communication between users	Unauthorized users contacting our tutors	Inappropriate messages can be sent to our tutors	Messaging requires the user to be logged in.
Data Upload and Message Content	Inappropriate content being posted on our website	Integrity of our site will be lost.	Starts in a pending state and needs Admin approval to go live on the website.
Search Input	SQL Injection	Unauthorized access to certain DB values	Allow up to 40 alphanumeric characters for search.
Registration	SQL Injection and more	Unauthorized users allowed to use SFSU exclusive functions	Registration Page validates only emails ending in "@sfsu.edu"

Self Check of the Adherence to Original Non-Functional Specs

Done

1. Application shall be developed, tested, and developed using tools and servers approved by Class CTO and as agreed in M0.
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers.
3. All or selected application functions shall render well on mobile devices (no native app to be developed)
4. Posting of tutor information and sending messages to tutors shall be limited only to SFSU students.
5. Critical data shall be stored in the database on the team's deployment server.
6. No more than 50 concurrent users shall be accessing the application at the same time.
7. Privacy of users shall be protected.
8. The language used shall be English (no localization needed).
9. Web application shall be easy to use and initialize.
10. Web application shall follow established architecture patterns.
11. Web application code and its repository shall be easy to inspect and maintain.
12. Google Analytics shall be used.
13. No email clients shall be allowed. Interested users shall only be able to message sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application.

14. Pay functionality, if any (e.g. paying for goods) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in class) for main data items.
16. Media formats shall be standard as used in the market today
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools.
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 2024. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application).

Use of GenAI Tools

Jaylin Jack - Backend Team Member

I used ChatGPT 40 mini to try to help my team meet the < 50 concurrent users on our site at once requirement. I asked ChatGPT how to make sure that my website wouldn't allow more than 50 users at one time using node and I was prompted to use ioredis I believe and it did not work whatsoever. I would say the use of ChatGPT was not helpful since I was not able to make this security requirement work using that method.

Thomas Bercasio - Team Leader

I used ChatGPT 40 mini to generate a list of questions to ask when conducting the usability test. It seemed for the most part to follow a lot of what was covered in the class lectures when it generated the questions and the categories they should belong to like ease of use, efficiency, satisfaction, etc. Overall for this use case, I would rate GenAI very highly.

Additionally, I also used this version of ChatGPT to generate a QA test plan for the feature we were going to test. It delivered an entire article covering a number of things we should go over when it came to creating tests for that function of the application. For each test ChatGPT also recommended test cases to go with them. Once again, I would rate GenAI very highly for this use case.

Zarko Cosovic - Backend Lead

I used ChatGPT 40 mini when working on planning and documentation of Usability testing and QA testing. I asked questions in general terms, and ChatGPT returned very detailed and well structured information on writing documentation for Usability and QA testing. The results were helpful, but there were too many suggestions, possibly because my questions were too general.

Suzanna Li - Frontend Lead

I used ChatGPT to help with finalizing the designs we did. It was not very helpful. The answers it gave were too general. I wanted to know if the colors on the site was good enough for colorblind people to view. It may not be a requirement but I believe it is important to create a site that is easily viewed by everyone. I consulted a colorblind test site instead of the results I got from chatgpt.