SW Engineering CSC 648-848 Fall 2025

Project Title: *LEMN SFSU — Learn Easily, Mentor Naturally*

Team 06

Role(s)	***Members***	***Email***
Team Lead	*Mukisa Lubega*	*922859259@sfsu.edu*
Technical Officer	*Grishma Thumar*	*gthumar@sfsu.edu*
Github Master	*Anurag Sanadi*	*asanadi@sfsu.edu*
Back-End	*Tony Haung*	*zhuang23@mail.sfsu.edu*
Front-End Lead	*Abinash Shrestha*	*asanadi@sfsu.edu*

Milestone 4 History

Version	***Date***	***Notes***
Submission Date	*12/17/2025*	*Submitted for m4 grading*

Prepared by Team 06 for CSC 648-848 Fall 2025 — San Francisco State University

Product Summary

LEMN SFSU provides a seamless way for students to browse and search for tutors by subject, class, or availability, ensuring they find the right peer support quickly. Each tutor and student has a verified SFSU profile, including course expertise and short introductions, creating a reliable academic environment. A built-in messaging system enables secure communication without relying on external email. Students can schedule meetings easily and manage their tutoring activity through a personalized dashboard that includes scheduling tools, video postings, and messaging. For oversight and quality assurance, administrators manage user approvals, monitor content, and maintain community standards, ensuring the platform remains safe and professional. An About Page transparently shares the project's purpose and team, reinforcing trust and clarity.

Major committed functions:

1. Guest users shall be able to use the search functionality to find appropriate tutors
2. Guest users shall be able to register for a user account (with a valid SFSU email)
3. Guest users shall be able to log in to an existing user account
4. Guest users shall be able to view the profiles of tutors and their details
5. Registered users shall inherit all the functionalities available to guest users
6. Registered users shall be able to log into the system
7. Registered users shall be able to message other registered users
8. Registered users shall be able to declare their class, and subjects they can tutor
9. Registered users shall be able to register as a tutor for specific courses
10. Registered users shall be able to view their dashboard to check messages and postings
11. Admins shall inherit all the functionalities available to registered users
12. Admins shall be able to verify registered accounts applying as a tutor
13. Admins shall be able to suspend registered users
14. Admins shall be able to approve posts before they go live

URL of product: http://3.133.58.251/

Usability Test Plan for Search

Test objectives

- Assess how easily users can locate and use the tutor search feature
- Determine whether users can successfully find tutors that meet specific criteria
- Measure task completion time and error rate to evaluate efficiency

Test background and setup

- Testers will use the build of the tutor app with access to the search function. The database will be pre-populated with sample tutor profiles that vary in courses and subjects.
- The intended users are students who are looking for tutors.
- Testers will start on the home page with the subject dropdown on all subjects.
- The testers will require either a smartphone or laptop/desktop with internet connection.
- The test environment will be at home with camera monitoring and no training.

Usability task description

- Find a tutor for computer science
- Find a tutor for the course CSC648
- Find what classes Alice is tutoring

Plan for evaluation of efficiency

- Measure how long it takes for testers to complete task
- Determine if users can complete tasks without assistance
- Note incorrect selections, backtracking, or confusion
- Count unnecessary clicks/taps or repeated actions
- Compare performance across participants to identify common bottlenecks

Plan for evaluation of user satisfaction:
It was easy to understand how to use the search bar
_Strongly disagree _Disagree _Neutral _Agree _Strongly Agree

The subject dropdown bar was useful when looking for a tutor
_Strongly disagree _Disagree _Neutral _Agree _Strongly Agree

The information shown for each tutor was easy to understand
_Strongly disagree _Disagree _Neutral _Agree _Strongly Agree

QA Test Plan for Search

Test Objectives

- Verify that the search feature returns accurate and relevant results
- Confirm that subject dropdown filters entries properly
- Ensure search input is validated and edge cases are handled
- Verify consistent behavior across platforms

HW and SW setup

- Testers will use the build of the tutor app with access to the search function. The database will be pre-populated with sample tutor profiles that vary in courses and subjects.
- Testers will use a smartphone and laptop/desktop to access the site on Google Chrome, and Safari

QA Test Plan (Google Chrome)

Test #	Test Title	Test Description	Test Input	Expected Output	Test Results
1	empty search bar	Ensure empty search bar returns proper results	Input nothing Click on search button		
2	search tutor by course	Ensure that search by course number returns proper results	Input “csc648” Click on search button		
3	subject dropdown	Ensure that the dropdown shows tutors for those course	Click subjects dropdown Select computer science Click on search button		

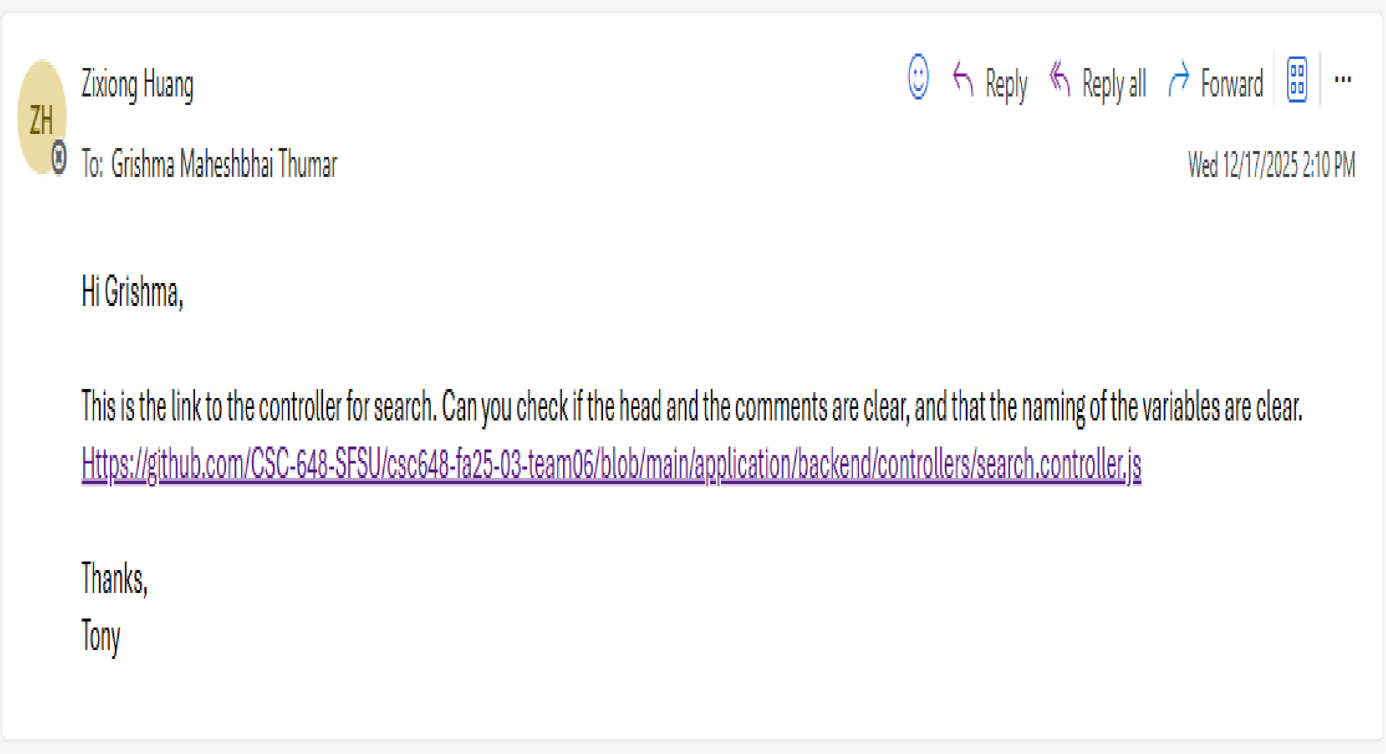
QA Test Plan (Safari)

Test #	Test Title	Test Description	Test Input	Expected Output	Test Results
1	empty search bar	Ensure empty search bar returns proper results	Input nothing Click on search button		
2	search tutor by course	Ensure that search by course number returns proper results	Input “csc648” Click on search button		
3	subject dropdown	Ensure that the dropdown shows tutors for those course	Click subjects dropdown Select computer science Click on search button		

Peer Code Review

Author: Zixiong (Tony) Huang, Back-end Lead
Peer Reviewer: Grishma Maheshbhai Thumar, Team CTO

Below are the email conversations:



Zixiong Huang

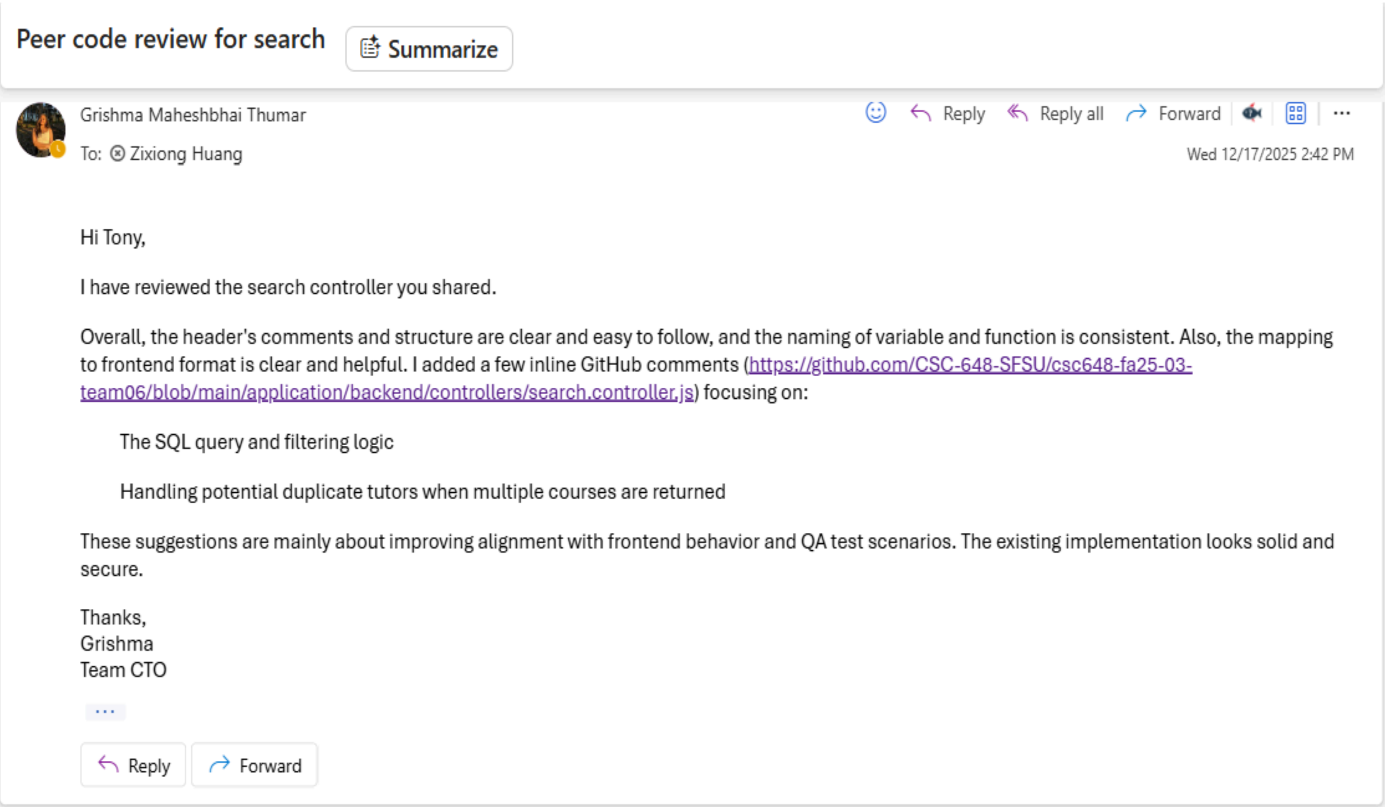
To: Grishma Maheshbhai Thumar

Wed 12/17/2025 2:10 PM

Hi Grishma,

This is the link to the controller for search. Can you check if the head and the comments are clear, and that the naming of the variables are clear.
Https://github.com/CSC-648-SFSU/csc648-fa25-03-team06/blob/main/application/backend/controllers/search.controller.js

Thanks,
Tony



Peer code review for search

Grishma Maheshbhai Thumar

To: Zixiong Huang

Wed 12/17/2025 2:42 PM

Hi Tony,

I have reviewed the search controller you shared.

Overall, the header's comments and structure are clear and easy to follow, and the naming of variable and function is consistent. Also, the mapping to frontend format is clear and helpful. I added a few inline GitHub comments (https://github.com/CSC-648-SFSU/csc648-fa25-03-team06/blob/main/application/backend/controllers/search.controller.js) focusing on:

The SQL query and filtering logic

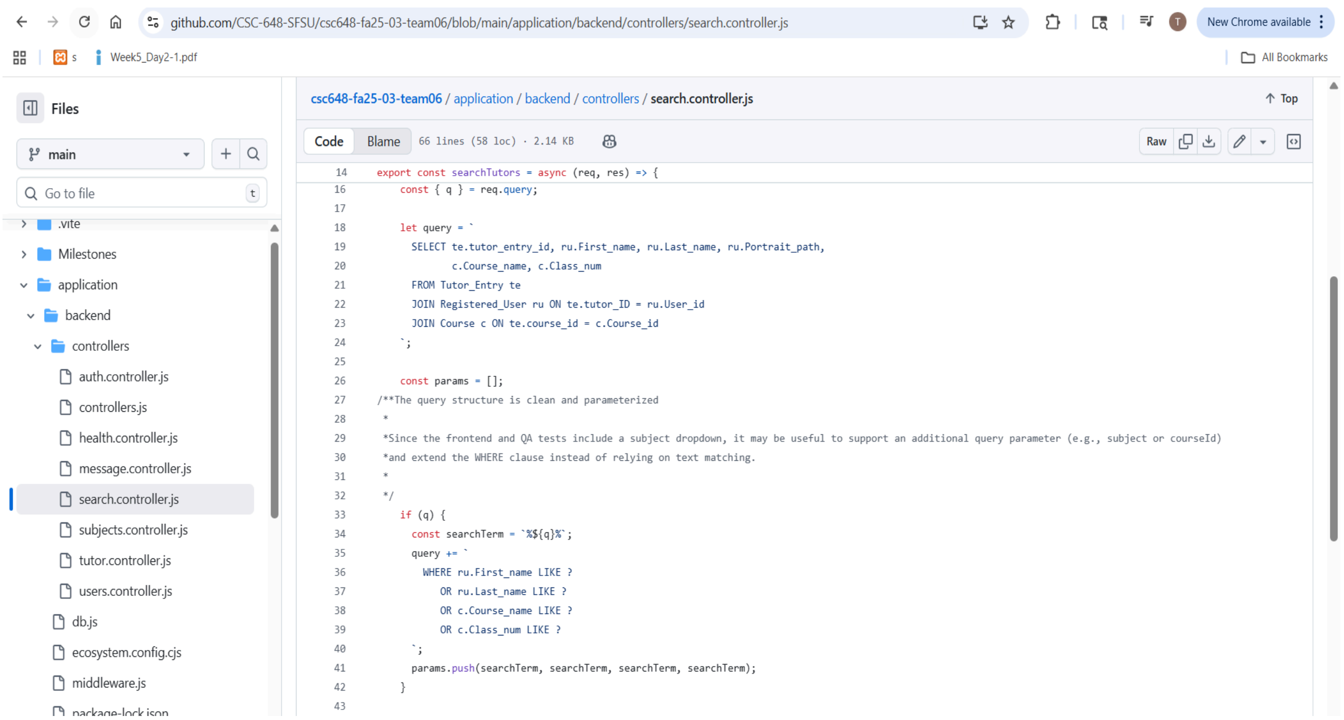
Handling potential duplicate tutors when multiple courses are returned

These suggestions are mainly about improving alignment with frontend behavior and QA test scenarios. The existing implementation looks solid and secure.

Thanks,
Grishma
Team CTO

The inline comments inGitHub for peer review:

Line: 27-32, 44,45 and 50

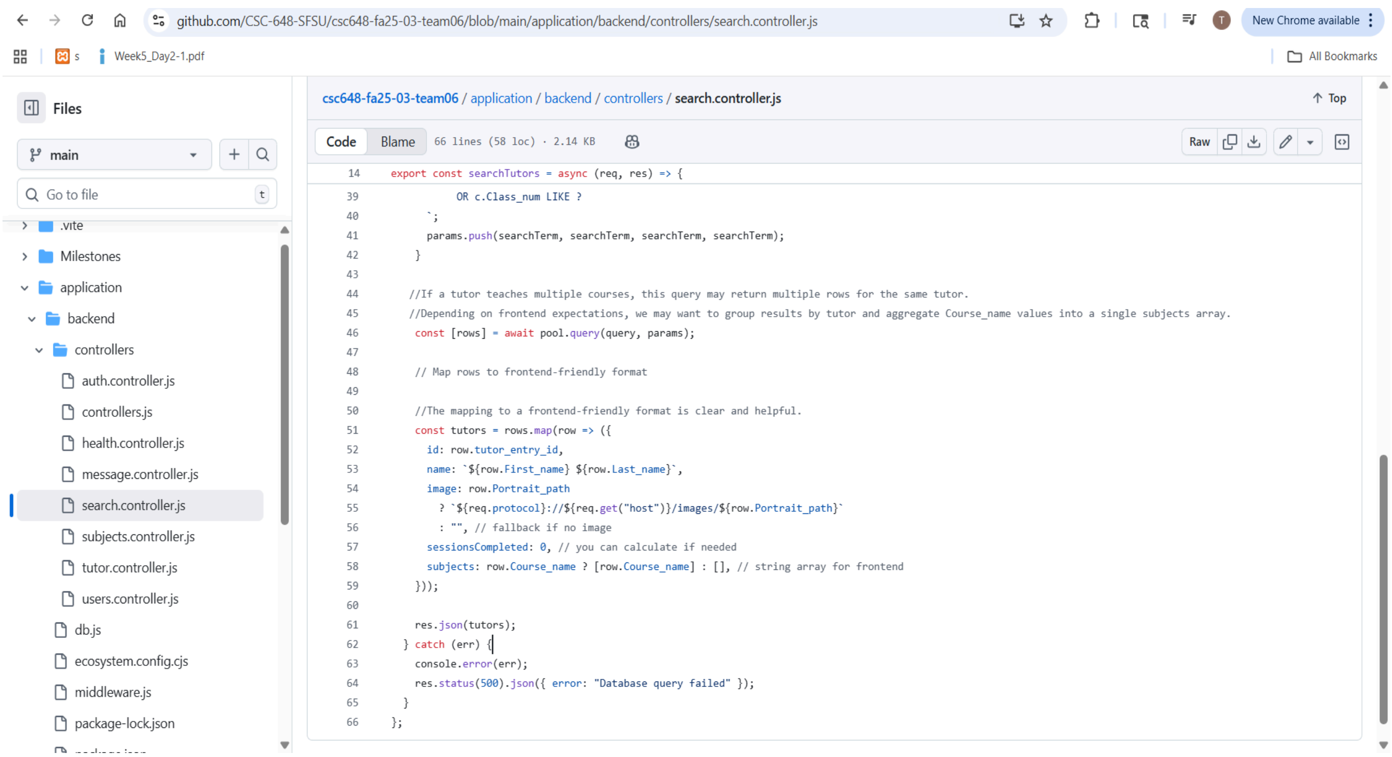


```javascript
export const searchTutors = async (req, res) => {
    const { q } = req.query;

    let query = `
      SELECT te.tutor_entry_id, ru.First_name, ru.Last_name, ru.Portrait_path,
             c.Course_name, c.Class_num
      FROM Tutor_Entry te
      JOIN Registered_User ru ON te.tutor_ID = ru.User_id
      JOIN Course c ON te.course_id = c.Course_id
    `;

    const params = [];
/**The query structure is clean and parameterized
 *
 *Since the frontend and QA tests include a subject dropdown, it may be useful to support an additional query parameter (e.g., subject or courseId)
 *and extend the WHERE clause instead of relying on text matching.
 *
 */
    if (q) {
      const searchTerm = `%${q}%`;
      query += `
        WHERE ru.First_name LIKE ?
           OR ru.Last_name LIKE ?
           OR c.Course_name LIKE ?
           OR c.Class_num LIKE ?
      `;
      params.push(searchTerm, searchTerm, searchTerm, searchTerm);
    }

```



```javascript
export const searchTutors = async (req, res) => {
           OR c.Class_num LIKE ?
      `;
      params.push(searchTerm, searchTerm, searchTerm, searchTerm);
    }

   //If a tutor teaches multiple courses, this query may return multiple rows for the same tutor.
   //Depending on frontend expectations, we may want to group results by tutor and aggregate Course_name values into a single subjects array.
    const [rows] = await pool.query(query, params);

    // Map rows to frontend-friendly format

    //The mapping to a frontend-friendly format is clear and helpful.
    const tutors = rows.map(row => ({
      id: row.tutor_entry_id,
      name: `${row.First_name} ${row.Last_name}`,
      image: row.Portrait_path
        ? `${req.protocol}://${req.get("host")}/images/${row.Portrait_path}`
        : "", // fallback if no image
      sessionsCompleted: 0, // you can calculate if needed
      subjects: row.Course_name ? [row.Course_name] : [], // string array for frontend
    }));

    res.json(tutors);
  } catch (err) {
    console.error(err);
    res.status(500).json({ error: "Database query failed" });
  }
};
```

Self-check on best practices for security

Assets to be protected	Types of possible attacks	Consequences of security breach	Strategy to protect asset
User data	SQL injection in register page	Sensitive data like email and name gets leaked	input validation for register form
Tutor listing data	SQL injection in apply as tutor form	Data like tutor schedule gets leaked	input validation for apply for tutor form
Message histories	SQL injection by inputting into chatbox	Privacy between users get breached and messages could be injected	parameterize message query so messages are not used for building sq strings
Database access credentials	Stealing the credentials posted on github	compromising the above 3 assets	change and then move credentials off of github

Gen AI Usage for M4

We used ChatGPT 5.1 for:

- Usability test plan
    - It was used to give input for all of the prompts in this section. Due to some missing inputs on our end, it gave some extra ideas that did not fit our project. We utilized some of its ideas with modifications.
    - HIGH. It gave a really good general idea of how to do the task, but it requires some more input from us.
- QA test plan
    - Same as the usability test plan.