SW Engineering CSC648-848 Fall 2025
Team 08

Kojiro Miura (Team Lead)
Kmiura@sfsu.edu
Atharva Walawalkar (Backend Lead)
Addhyan Kohli (Frontend Lead)
Krinjal Basnet (Frontend Dev)
Sonam Tobgyal (Github Master)
Aketzali Zeledon (Backend Dev)

Initial Submission	10/16/2025
Revised submission 1	10/25/2025

Table of Contents

1. Executive Summary

The SFSU Tutoring Portal is an exclusive academic platform developed to connect San Francisco State University students with trusted and verified tutors in a secure and professional environment. Designed to make academic support simple, reliable, and accessible, the platform allows students to easily search for tutors based on course, subject, or language, and schedule sessions that fit their needs. Each user registers with an official SFSU email address, ensuring that the system remains exclusive to the university community and maintains academic credibility. Tutors can manage their profiles, share learning resources, and set availability, while administrators review and approve all listings to uphold quality and integrity.

Students using the platform can search for tutors by course, subject, or language, making it easier than ever to find help that directly matches their academic goals. The system's user-friendly interface allows for smooth navigation, quick scheduling, and instant access to support that fits any learning style or schedule. Tutors can manage their profiles, showcase their expertise, and upload learning materials to enhance student understanding. The platform ensures that all listings are verified and approved by administrators, maintaining a high standard of quality and compliance with university policies. Communication between tutors and students is streamlined through an in-site messaging system that promotes professionalism and protects privacy, ensuring that every interaction stays productive and secure.

What makes the SFSU Tutoring Portal truly stand out is its commitment to inclusivity, convenience, and academic growth. Its modern and mobile-friendly design enables students to access tutoring anytime, anywhere, turning academic help into a seamless part of campus life. Whether students need support for challenging coursework, test preparation, or ongoing mentoring, the portal offers an environment that encourages continuous learning and personal improvement. More than just a website, it functions as a trusted academic ecosystem that empowers every SFSU student to take ownership of their education, build confidence, and achieve excellence through guided collaboration.

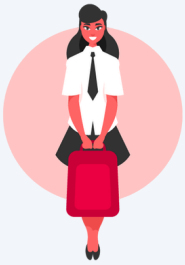
Behind this impactful initiative stands a passionate and highly skilled team dedicated to transforming the way academic support is delivered. The project is led by **Kojiro Miura**, who serves as the Team Lead and guides the team with vision and direction. **Atharva Walawalkar** oversees the backend development as Backend Lead, while **Addhyan Kohli** leads the frontend

design and user experience. **Krinjal Basnet** contributes as Frontend Developer, ensuring functionality and aesthetic balance. **Sonam Tobgyal** manages version control and coordination as GitHub Master, and **Aketzali Zeledon** strengthens backend performance and reliability as Backend Developer. Together, this talented group has created more than just a platform as they have built a digital foundation that strengthens academic connections, supports every student's learning journey, and redefines what it means to learn and grow within the SFSU community.

2. Personae

User Categories: Tutor, Student, Tutor & Student, and Admin.

<table>
<tr><td rowspan="2">Akemi - Student

  </td><td>Goals:
Improve Business Calculus understanding to get a better grade.
Get detailed explanations on calculus concepts.
Take advantage of time in between classes with structured accountability to study.</td></tr>
<tr><td>Skills:
Experienced with SFSU portal, Canvas, and Zoom as a student.
Comfortable web browsing on laptop or mobile devices.</td></tr>
<tr><td>General Characteristics:
SFSU Undergrad Student

A commuter student that has gaps between classes. Struggles with Business Calculus.</td><td>Pain Points:
Difficulty finding subject specific tutors.
Limited Time Constraints due to class schedule.
Poor at keeping track of commitments.
Distrustful of meeting people from third party websites.</td></tr>
</table>

Alex - Tutor 	**Goals:** • Earn income while tutoring students on campus. • Schedule tutoring sessions ahead of time with the general topic in mind in order to review before meeting. • Limit tutoring to specific classes and topics.
	Skills: • Experienced with SFSU portal and Zoom. • Comfortable web browsing on laptop or mobile devices.
General Characteristics: SFSU Alumni Good communicator, with time on certain days. Bachelor's degree in Business Accounting and Minor in Mathematics.	**Pain Points:** • Would like to avoid tutoring requests for unrelated topics or classes they are not familiar with. • Prefer to not have contact information available to non-students.

Jean - Tutor & Student 	**Goals:** • Earn income while tutoring students on campus. • Gain experience teaching • Get affordable tutoring on upper division courses. • Keep track of previous completed tutoring sessions.
	Skills: • Experienced with SFSU portal, Canvas, and Zoom as a student. • Comfortable web browsing on laptop or mobile devices.
General Characteristics: SFSU Undergrad Student IR Major Enjoys explaining concepts as it helps reinforce learning.	**Pain Points:** • I would prefer to have time in between sessions. • Clarity and differentiation between hosting tutoring vs being tutored. • Open to appointment time flexibility outside of posted hours.

Chris - Admin 	**Goals:** ● Manage the tutoring posts for appropriateness. ● Update Available Class Lists **Skills:** ● Management and Organization Tools ● Experienced with SFSU portal, Canvas, and Zoom as SFSU Staff.
General Characteristics: SFSU Admin Staff	**Pain Points:** ● Not knowing the difference between current semester listings vs previous semester listings.

3. High-level Use Cases

Student schedules a tutoring meeting

Akemi is a **student** who is an SFSU undergrad student who is not doing so well in Business Calculus class. They want tutoring and login to the tutoring website to find specific Business Calculus tutoring sessions on Zoom. After their class they browse and choose a tutor that is teaching specifically Business Calculus that aligns with their schedule. The **tutoring website portal** then shows a list of available times and different classes for Zoom tutoring, the student confirms the time slot that fits their busy schedule for business calculus **tutoring meeting** and adds a **reminder** for the student to join the Zoom through the Zoom link on the selected time slot through the tutoring website on their SFSU portal.

Tutor inserts their own time availability

Alex is a **tutor** who is an SFSU alumni and teaches both Business Accounting and mathematics. Alex the tutor first logs into the **tutoring website portal**, after the tutor logs in using his SFSU login, the tutor checks his **tutor schedule** to check his monthly **available time slots** and selects and limits what courses the tutor wants to offer and at what time availability. The tutor receives a request that matches with what the tutor wants to offer. The tutor accepts the offer and the tutoring website portal automatically sends that the tutor has accepted the offer to the student and

adds the **tutoring meeting** to both tutor and student's email and calendar notification with a **reminder** to both and the tutor's **Zoom link**.

Student and tutor receive notification along with a zoom link

Jean is both a **tutor & student** who is an SFSU undergraduate student majoring in IR. Jean logins into the **SFSU portal** and is prompted for a **tutoring meeting** with a student who wants a **tutoring meeting**. Jean accepts the **tutoring meeting** and checks her **available time slots** as she also needs tutoring in IR between her classes. Jean searches for an upper division tutoring meeting that fits her available session and requests for a **tutoring meeting** that fits both her tutor and student schedules.

Chris, who is the **admin**, needs to monitor the tutoring website for outdated or irrelevant class posts. Chris sets up the **Class Options** using the **School Catalog** for the semester. He accesses the **Admin Dashboard** and reviews existing course listings, tutor profiles, and student requests. Using the available **Management Tools**, he updates tutor availability and ensures that each **Tutor Profile** includes accurate subjects, schedules, and contact information. Chris uses the **Scheduling Module** to assign tutors to the correct courses based on their areas of expertise and availability. When new tutors are hired or existing ones leave, Chris updates the database accordingly through the **Tutor Management Panel**. He monitors **Session Reports** to track student attendance, tutor activity, and overall platform usage. If issues arise such as duplicate course listings or inactive tutor accounts he resolves them using the **Maintenance Tools** provided by the system.

Student can request for specific course to be tutored for if not offered

Akemi, who is a **student, searches** the SFSU tutoring portal for help with ECON 301 (Intermediate Microeconomics) but finds no **Tutors** currently offering this course. Instead of leaving the platform, the Student submits a **Course Coverage Request** specifying the SFSU course number, topics needed (consumer theory and market equilibrium), preferred rate range, and availability between classes. The system identifies Tutors with expertise in related courses like ECON 101 or ECON 200 and sends them notifications about the request. A Tutor teaching ECON 200 reviews the request, decides they can help, and accepts. The system notifies the Student, they schedule their first **Tutoring Session**, and the Tutor's profile is updated to show ECON 301 as an available course for future students.

4. Data Glossary/ Description

User: Any person who logs into the tutoring website through the SFSU portal.

Student: A person enrolled at SFSU, using the website.

Tutor: An SFSU enrolled person (or student) who offers tutoring services.

Tutor & Student: A user who serves both as a tutor and as a student.

Admin: SFSU staff who monitor the tutoring website

Class Options: The list of available SFSU courses for the semester as configured by the Admin.

Course Listings: Courses offered by SFSU, added by the admin.

Tutor Profile: Contains information about tutor name, subjects, schedule, and credentials.

Student Profile: Contains information about enrolled classes, requests

Tutoring Session: A scheduled meeting between student and tutor including course, time, location or meeting link.

Dynamic Waitlist: A feature that groups students requesting the same topic and time range for small-group tutoring when sessions are full.

Course Coverage Request: A request submitted by a student when a desired course is not currently offered for tutoring.

Meeting Link: The meeting link automatically sent to both tutor and student when a session is confirmed.

Session Feedback: Ratings and written comments exchanged between student and tutor after completing a session.

Tutor Management Panel: Admin interface used to update tutor information, assign courses, and remove accounts.

Maintenance Tools: System utilities allowing the Admin to resolve issues such as duplicate course listings or inactive tutor accounts.

Session Reports: Records that show student attendance, tutor activity, and overall platform usage, used by Admin for monitoring.

Management Tools: General admin utilities used to update and maintain tutor listings, schedules, and system content.

SFSU Portal: Authentication gateway for all users (students, tutors, admins) to log in securely.

5. High Level Functional Requirements

Unregistered Users

1. Unregistered users can browse and search tutor profiles without logging in.
2. They can filter search results by course number, subject tag, or language.
3. They can open and view tutor profiles from the search results.

Registered Students

4. Registered students can send one in-site message to a selected tutor through the contact form.
5. Students can view their sent message in their personal dashboard.
6. Each student is limited to a single in-site message per tutor to maintain privacy and professionalism.
7. Students can submit a “Reported Item” form to flag inappropriate or inaccurate tutor profiles.
8. Request courses to be tutored that are not yet available
9. Use the chat functionality to send and receive real time messages, share files.
10. Students can schedule tutoring sessions with available tutors and automatically receive a confirmation notification containing a Zoom link and calendar reminder.

Registered Tutors

8. Registered tutors can create and manage their tutor profile, which includes courses covered, subjects, hourly rate, languages spoken, and availability summary.
9. All newly created or edited tutor profiles are automatically placed in **Pending** status for administrative review.
10. Tutor profiles become visible in search results only after admin approval.
11. Tutors can edit their profiles and resubmit them for approval when needed.
12. Tutors can view received in-site messages through their dashboard.
13. Tutors can view the current approval status of their profile (pending, approved, or rejected).
14. Use the chat functionality to send and receive real time message, share files.
15. Tutors can manage and update their time availability.
16. Tutors receive notifications when a student books a confirmed session.

Administrators

14. Administrators can review pending tutor profiles and approve or reject them.
15. Administrators can delete inappropriate listings, manage reported items, and update class or course lists.
16. Administrators can monitor overall activity, including tutor availability, student requests, and message reports.

6. Non-functional Requirements

1. Application shall be developed, tested and deployed 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 be rendered well on mobile devices (no native app to be developed)
4. Posting of tutor information and messaging 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 any time
7. Privacy of users shall be protected
8. The language used shall be English (no localization needed)
9. Application shall be very easy to use and intuitive
10. Application shall follow established architecture patterns
11. Application code and its repository shall be easy to inspect and maintain
12. Google analytics shall be used
13. No e-mail clients shall be allowed. Interested users (clients) can only message service providers via in-site messaging. One round of messaging (from client to service provider) is enough for this application. No chat functions shall be developed or integrated
14. Pay functionality (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
15. Site security: basic best practices shall be applied (as covered in the 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 2025. For Demonstration Only"at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application).

7. Competitive Analysis

Key Functions	**Wyzant**	**CSM**	**Tutor.com**	**SFSU (our product)**
Uni Access	❌ No university login	✅ Supports university access	❌ Individual login only	✅ integrates with student records
Adding Tutor listings by Admin	❌Tutors create own listings	✅Admins can add tutors manually	✅Admins can approve and post listings	✅Admins can add tutors ✅link to specific courses ✅manage approvals
Search tutor	✅Search by subject and price	✅Search by subject and department	✅Search by subject and availability	✅Advanced search by course/subject and specialization ✅Search by schedule
In-app Text	✅Supports messages	❌No messaging option	✅Supports messages	✅Supports messages ✅Offers real-time chat ✅File sharing
Mobile Responsive	✅Mobile responsive	❌Limited mobile layout	✅Mobile responsive	✅Fully responsive ✅ Optimized for chat, booking, and schedules

8. High-level System Architecture & Technologies

Amazon AWS EC2 (t2.micro, Free Tier)
Ubuntu Ubuntu 22.04 LTS
MySQL 8.0.43
Nginx 1.18.0
React 18.3.1

Python 3.11
FastAPI 0.116.2
SQLAlchemy 2.0.44
Pytest 8.4.2
Google Analytics
Compatible with Google Chrome 139-141 and Firefox 140.0-144.0

9. Use of GenAI

9.1 ChatGPT: High Level Use Case - Admin

ChatGPT was given the low level description of the admin use case as a rough draft as presented below and then asked to convert it into a 'high level' use case for the admin.

> "The admin needs to monitor the website for outdated or relevant class posts. The admin sets up the **class options** using the school catalog for the semester. The Admin reviews the existing **listings** by tutors, tutor profiles, student & tutor profiles, and student profiles. The admin uses the **admin tools** to update or archive out of date listings. "

The output was edited and reduced to be more relevant to the goals discussed by the group.

9.2 ChatGPT – Executive Summary

A rough draft was given to ChatGPT so that we have much polish, sales worthy executive summary.

10. Team Roles

Name	Role	School email
Kojiro Miura	Team Lead	kmiura@sfsu.edu
Atharva Walawalkar	Backend Lead	awalawalkar@sfsu.edu
Addy Kohli	Frontend Lead	akohli@sfsu.edu
Sonam Tobgyal	Github Master	stobgyal@mail.sfsu.edu
Krinjal Basnet	Frontend Dev	kbasnet1@sfsu.edu
Aketzali Zeledon	Backend Dev	azeledon@mail.sfsu.edu

11. Team Lead Checklist

- So far all team members are fully engaged and attending team sessions when required
    - On Track
- Team found a time slot to meet outside of the class
    - Done
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing
    - Done
- Team reviewed class slides on requirements and use cases before drafting
    - Done

Milestone 1

- The team reviewed non-functional requirements from “How to start...” document and developed Milestone 1 consistently
    - Done
- Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission
    - Done
- Team lead ensured that all team members read the final M1 and agree/understand it before submission
    - Done
- Team shared and discussed experience with GenAI tools among themselves
    - Done
- Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)
    - On Track