

SW Engineering CSC 648-848 Fall 2025

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

Team 06

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

Milestone 1 History

Version	Date	Notes
<i>Submission Date</i>	<i>10/20/2025</i>	<i>Submitted for m1 grading</i>
<i>Revision Date</i>	<i>10/25/2025</i>	<i>Revised for m1 grading</i>

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

Executive Summary

Access to quality academic support can make the difference between students struggling or thriving in their courses. At San Francisco State University (SFSU), many students seek tutoring but face barriers such as unreliable external sites, high costs, or scattered communication tools. **LEMN SFSU** is designed to address this gap by offering a centralized, secure, and student-exclusive platform for peer tutoring. By creating a trusted campus network, this platform exclusively for SFSU supports the schools mission to promote academic success, equity, and community building.

LEMN SFSU provides a seamless way for students to **browse and search for SFSU specific 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.

Unlike generic public tutoring platforms, LEMN SFSU is **exclusively for SFSU students**, requiring a verified @sfsu.edu email for registration. This builds a trusted peer-to-peer learning space aligned with SFSU's academic culture. The platform integrates **SFSU-specific course filters and class codes**, making it easy for students to find help tailored to their exact classes. LEMN SFSU is **completely free**, with no ads or payment systems, ensuring that academic support is accessible to all students. Administrative oversight ensures compliance with CSU IT and privacy policies, protecting both students and tutors.

LEMN SFSU delivers **academic value** by improving student performance and retention through accessible, peer-driven tutoring. It provides **social value** by strengthening the campus community, encouraging inclusivity, and fostering connections between students. It also offers **career value** by allowing tutors to develop teaching and leadership skills, showcase their experience on résumés, and open doors to future opportunities in education and mentoring.

LEMN SFSU is developed by a team of passionate SFSU software engineering students committed to solving real campus challenges. This project represents not only a technical solution but also a student-driven initiative to build a stronger academic community. With your support, we aim to make LEMN SFSU a lasting resource that enhances student success across SFSU.

Personae

Persona 1 – The Guest User (“Curious Student”)



About:

A first-time or occasional visitor to the tutoring platform. Often a freshman exploring SFSU's academic resources or an upperclassman seeking quick homework help.

Attitude:

Open-minded but cautious about signing up; wants quick answers without commitment.

Skills:

Has basic web and smartphone skills (browsing, simple search).

Limitations:

Unfamiliar with the tutoring system; may not understand navigation or access limitations without registration.

Pain Points:

- Unclear filters or search options.
- Can't view all tutor details before registering.
- Unsure about tutor credibility or ratings.

Goals and Scenario:

- I want to browse subjects, view tutor profiles and availability, and decide whether registration is worth it.

Example Scenario:

- While doing homework at home, searches “Calculus 1 tutor SFSU,” visits the site, filters by subject, and checks tutor bios before deciding to create an account.

Persona 2 – The Student User (“Active Learner”)



About:

A student who actively uses the platform for consistent academic support.

Attitude:

Goal-oriented and self-motivated; values efficiency and reliable communication.

Skills:

Comfortable navigating dashboards, scheduling tools, and messaging features.

Limitations:

Limited time; must balance studies and scheduling conflicts.

Pain Points:

- Some subjects lack available tutors.
- Tutors may reply slowly.
- Schedule mismatches cause missed sessions.

Goals and Scenario:

- I want to find qualified tutors quickly, book sessions, and track appointments.

Example Scenario:

- Logs in between classes, filters by “CSC 240,” messages a tutor, and books a video session that fits both schedules.

Persona 3 – The Tutor (“Peer Mentor”)



About:

An SFSU student confident in specific subjects who tutors peers to build experience and earn income.

Attitude:

Supportive and professional; enjoys helping classmates succeed.

Skills:

Strong subject knowledge, communication, and basic familiarity with teaching tools.

Limitations:

Managing multiple messages or sessions while studying; waiting for admin approval of content.

Pain Points:

- Slow approval of video demos or profile updates.
- Limited visibility due to search-ranking issues.
- Repetitive student questions.

Goals and Scenario:

- I want to maintain a clear schedule, highlight expertise, and reach more students.

Example Scenario:

- Updates profile with new “Data Structures” video, checks dashboard for pending session requests, and confirms appointments with two students.

Persona 4 – The Admin (“Site Moderator”)



About:

SFSU staff or project-team members ensuring safety, verification, and quality control.

Attitude:

Responsible and cautious about maintaining trust and compliance.

Skills:

Proficient in platform-management tools, data review, and moderation workflows.

Limitations:

High workload; must process many verifications manually.

Pain Points:

- Time-consuming content approvals.
- Balancing fast response with careful verification.
- Handling inappropriate messages or reports.

Goals and Scenario:

- I want a smooth moderation system and trusted user community.

Example Scenario:

- Reviews new tutor applications, checks uploaded videos for accuracy, and flags inappropriate chat messages for follow-up.

Use Cases

1. Browsing and Search

A new guest user opens the tutoring application. They browse through the list of available tutors. They view a tutor's profile, and details like subject, class, and time availability. They evaluate the tutor and decide that the tutor does not match. The guest user decides to search for their class and subject to find a tutor.

2. Register and log in

The guest user found a matching tutor for the help they need. The guest user sees that they can message the tutor to book a session. They see that they can also apply as a tutor. The guest user registers their account using their school email and sets a password. They can now log in with their email and password.

3. Dashboard

The registered user logs in and sees their dashboard. They see that they can apply as a tutor there. They edit their profile through the dashboard to introduce themselves. They can now message the tutor through the messaging section of the dashboard or the tutor's page. They write the subject in the subject section of the message and ask for a tutoring session along with their time availability and wait for the tutor to respond.

4. Tutoring

A registered user wants to become a tutor. They apply as a tutor on their dashboard. They list their time availability, classes, and subjects that they can tutor. After some time, an admin verifies the application. The registered user goes to their dashboard and sees that they can post videos and uses it as an introduction. The video gets verified and gets posted. They receive a message requesting a tutoring session. They reply to it to finalize the booking.

5. Admin Management

An admin sees on their dashboard that a registered user is trying to apply as a tutor. They check their application and verify them. After a while, the user posts a video. Before the video goes up, the admin checks the video and sees that it is appropriate and lets it get posted. The admin then checks the profile and messages of the user and sees that it is very inappropriate and suspends the user with a warning.

High Level Initial Function Specifications

- Video: Tutoring videos posted by registered users
- Subject: Branch of knowledge that is being tutored by the registered user
- Class: The specific class that a registered user can cover
- Guest User: A user who has not registered an account and does not have registered user privileges
- Registered User: A user who has registered an account and has registered user privileges
- Admin: A user who has registered user privileges and manages registered accounts
- Message: An asynchronous message sending system between registered users

Functional Requirements

Guest User

1. Guest users shall be able to browse the tutoring website
2. Guest users shall be able to use the search functionality to find appropriate tutors based on their availability, class, and subject (SFSU unique search)
3. Guest users shall be able to view the profiles of tutors and their details
4. Guest users shall be able to register for a user account (with a valid SFSU email) that allows them to act as both a student and a tutor on the platform
5. Guest users shall be able to log in to an existing user account

Registered User

6. Registered users shall inherit all the functionalities available to guest users
7. Registered users shall be able to log into the system
8. Registered users shall be able to message other registered users
9. Registered users shall be able declared their time availability, class, and subjects they can tutor
10. Registered users shall be able to update their profile information
11. Registered users shall be able to post videos

Admin

12. Admins shall inherit all the functionalities available to registered users.
13. Admins shall be able to monitor messages between registered users
14. Admins shall be able to view profiles of registered users
15. Admins shall be able to remove inappropriate content
16. Admins shall be able to verify newly registered accounts
17. Admins shall be able to approve posts before they go live

18. Admins shall be able to suspend registered users

Nonfunctional 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 render 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 to 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).

Competitive Analysis (Functions/Features ONLY)

The table below compares key functionality among our planned product and leading tutoring platforms. Focus is on features, not pricing or business models. Our column is shaded.

Feature	LEMN SFSU (Our Product)	Wyzant	Varsity Tutors	Knack (Campus Tutoring)	TutorOcean
SFSU-only access (requires @sfsu.edu)	Yes	No	No	Yes	Configurable (schools)
Filter by SFSU course codes & sections	Yes	Subject-level only	Subject-level only	Campus & course filters	Campus/program filters
In-site messaging only (no external email for comms)	Yes	Yes	Yes	Yes	Yes
Admin moderation (account & content approval)	Yes	Limited	Limited	Yes	Yes
Tutor intro videos on profiles	Yes	Varies	Varies	Yes	Yes
Free to students (no payments/ads)	Yes	No	No	Institution-paid	Institution-paid

Summary

LEMN SFSU is differentiated by campus-only access tied to @sfsu.edu identity, precise filtering by SFSU course codes, and a free, ad-free experience with in-site messaging and active admin moderation of accounts and posts. General tutoring marketplaces (Wyzant, Varsity Tutors) offer broad reach but lack SFSU-specific filters and closed campus gating, while campus-focused platforms (Knack, TutorOcean) approximate some institutional controls but do not target SFSU's exact class taxonomy and project constraints (e.g., no external email, no payments). Our features set is intentionally scoped to SFSU needs and class requirements.

High-Level System Architecture & Technologies Used

Below is the concise, itemized list (same as M0 unless noted):

Main software components (with versions):

Category	Details
Main Software Components (with Versions)	<ul style="list-style-type: none"> Backend: Node.js 22 LTS, Express 4.x Database: MySQL 8.0 Web Server / Reverse Proxy: Nginx 1.24 (Ubuntu 22.04 LTS)
Deployment Cloud Service	AWS (EC2 for app, optional RDS for DB, S3 for media assets)
Front-End Framework	React 18 with Vite build and Tailwind CSS
Supported Browsers (Last Two Versions)	<ul style="list-style-type: none"> Google Chrome Mozilla Firefox
Major Additional Open-Source APIs / Libraries	<ul style="list-style-type: none"> Google Analytics 4 (basic usage metrics, no PII) Sharp (Node image processing) for thumbnails / avatars AWS SDK (S3 object storage for media uploads) reCAPTCHA v3 (signup / submit abuse protection)

Use of GenAI for M1

GenAI Tool Used: ChatGPT (GPT-5, OpenAI Platform)

Tasks and Usefulness Ratings:

Task	Description of Use	Usefulness
Drafting Executive Summary & Personae	<i>Used ChatGPT to generate a clear and concise executive summary and outline the four core personae (Guest User, Registered User, Tutor, Admin). This helped standardize tone and flow for non-technical readers.</i>	High
Structuring Functional & Non-Functional Requirements	<i>Assisted in grouping functional requirements by user level (Guest, Registered, Admin) and formatting non-functional requirements to match the class document template.</i>	Medium-High
Competitive Analysis (Table & Summary)	<i>Used ChatGPT to research and compare tutoring platforms (Wyzant, Varsity Tutors, Knack, TutorOcean) and generate a comparison table focused on features only. Manual validation ensured accuracy.</i>	High
High-Level Architecture List	<i>Leveraged ChatGPT to summarize approved stack from Milestone 0 and format it into an itemized list per professor's template. No architectural decisions were made by AI.</i>	Medium
Editing & Formatting	<i>Used ChatGPT to generate clean tables and consistent headings in .docx and PDF formats to ensure compliance with submission guidelines and Canvas standards.</i>	High

Example Prompts Used:

- “Write an executive summary for a peer-tutoring web app exclusive to SFSU students called LEMN SFSU.”
- “Create four user personae for a university tutoring platform (Guest User, Registered User, Tutor, Admin).”
- “Generate a competitive feature comparison table between LEMN SFSU and Wyzant, Varsity Tutors, Knack, TutorOcean.”
- “List high-level software components and technologies for a Node.js-React-MySQL web application hosted on AWS.”

Reflections and Lessons Learned:

ChatGPT proved most useful for formatting and communication-oriented tasks (summaries, tables, structure) rather than for technical decision-making. Its use accelerated documentation and improved clarity without replacing team judgment or research. The team ensured that all AI-assisted outputs were manually reviewed and edited for accuracy and originality, consistent with SFSU academic honor and GenAI policy.

Team Lead Checklist

For each item below, the team lead must answer with one of the following: DONE/OK (completed successfully), ON TRACK (in progress and expected to be completed on time), or ISSUE (problem encountered, with short explanation).

Item	Status	Comments (if ISSUE)
So far all team members are fully engaged and attending team sessions when required	OK	—
Team found a time slot to meet outside of the class	DONE	Weekly team sync established 1 night at 9pm
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	ON TRACK	Some members still refining React familiarity through internal mini-projects
Team reviewed class slides on requirements and use cases before drafting Milestone 1	DONE	—
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	OK	—
Team shared and discussed experience with GenAI tools among themselves	DONE	Used ChatGPT collaboratively for section drafting and formatting consistency
Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)	ON TRACK	Repo structured with /application/, /docs/, and /Milestones/ folders per guidelines