

Final Project for SW Engineering Class 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>ashrestha@sfsu.edu</i>

Milestone 5 History

Version	Date	Notes
<i>Submission Date</i>	<i>12/17/2025</i>	<i>Submitted for m5 grading</i>

URL: <http://3.133.58.251/>

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/>

SW Engineering CSC 648-848 Fall 2025

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

Team 06

<i>Role(s)</i>	<i>Members</i>	<i>Email</i>
<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

<i>Version</i>	<i>Date</i>	<i>Notes</i>
<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

SW Engineering CSC648/848 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>ashrestha@sfsu.edu</i>

Milestone 2 History

Version	Date	Notes
<i>Submission Date</i>	<i>10/31/2025</i>	<i>Submitted for m2 p1 grading</i>
<i>Revision Date</i>	<i>11/05/2025</i>	<i>Revised for m2 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.

List of Main Data Items & Entities

List of entities:

- 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

High level Functional Requirements.

User: a person who uses the tutor application with different levels of permission

- Guest user: A user who has not register 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

Tutor Entry: A record that is made when a registered user applies as a tutor for a specific course

- Course Name: Name of course, eg., Software Engineering
- Class Number: eg., CSC648
- Subject: The category that the course belongs in, eg., Computer Science
- Availability: The times that the tutor is available

Message Entry: A record that is made when a registered user messages another registered user.

- Sender: The user who is sending the message
- Receiver: The user who is receiving the message
- Message: Alphanumeric message sent between the users.

Post Entry: A record made when a post is made by a registered user

- Image: An image post
- Video: A video post
- Text: A text post
- Approved Post: A post that is approved can be posted

Functional Requirements - Prioritized

Priority 1

- Guest User
 - 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)
 - 4) Guest users shall be able to register for a user account (with a valid SFSU email)
 - 5) Guest users shall be able to log in to an existing user account
 - 3) Guest users shall be able to view the profiles of tutors and their details

- 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
 - x1) Registered users shall be able to register as a tutor for specific subjects/courses
 - x2) Registered users shall be able to view their dashboard to check messages and postings
- Admin
 - 12) Admins shall inherit all the functionalities available to registered users
 - 16) Admins shall be able to verify registered accounts applying as a tutor
 - 18) Admins shall be able to suspend registered users
 - x3) Admins shall be able to approve posts before they go live

Priority 2

- Guest User
 - 1) Guest users shall be able to browse the tutoring website
- Registered User
 - 10) Registered users shall be able to update their profile information
 - 17) Admins shall be able to approve posts before they go live
- Admin
 - 14) Admins shall be able to view profiles of registered users
 - 15) Admins shall be able to remove inappropriate content

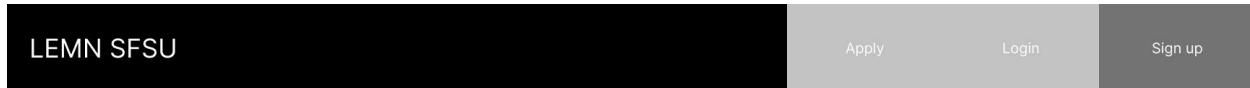
Priority 3

- Guest User
- Registered User
 - 11) Registered users shall be able to post videos
- Admin

UI Storyboards

Register

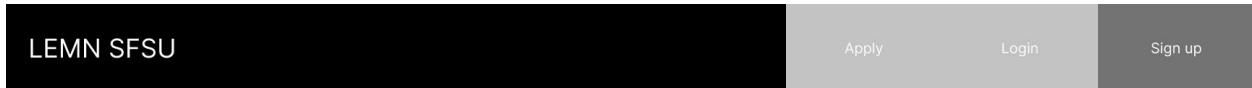
Unregistered users click on Sign Up on the navigation bar. It asks for username, password, SFSU email and phone number. It also has the ‘already have an account’ for registered users.



A registration form consisting of four input fields and one action button. The fields are labeled "Name", "Password" (with an eye icon), "SFSU Email", and "+1 Phone Number". Below these fields is a black rectangular button labeled "Next". At the bottom center of the page is a small, faint link: "Already have an account?"

The unregistered user enters the information username, password, SFSU email (@sfsu.edu)

and phone number to register and click on the Next button to proceed. Once the information is verified the page lists the email.

A screenshot of a registration form. It consists of four input fields stacked vertically: 1. A text input field containing "John". 2. A password input field containing "....." with an eye icon to its right. 3. An email input field containing "922859239@Sfsu.Edu". 4. A phone number input field containing "+925-484-9543". Below these fields is a dark grey "Next" button. At the bottom of the page, there is a small, faint link that says "Already have an account?"

Unregistered users click on the sign up button to complete the registration process.

LEMN SFSU

Apply

Login

Sign up

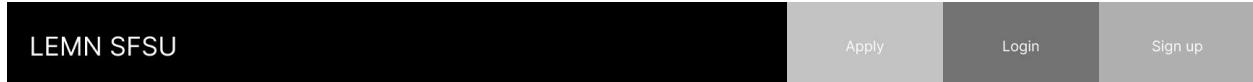
922859239@Sfsu.Edu

Sign up

Login

Registered users click on Login on the navigation bar.

The registered user enters the SFSU
email and password to Login / Sign in.



A login form consisting of two input fields and a "Sign in" button. The first field is labeled "SFSU Email" and the second is labeled "Password" with an eye icon indicating it's a password field. Below the fields is a black "Sign in" button. At the bottom right of the form area, there is a small link "Forgot password?"

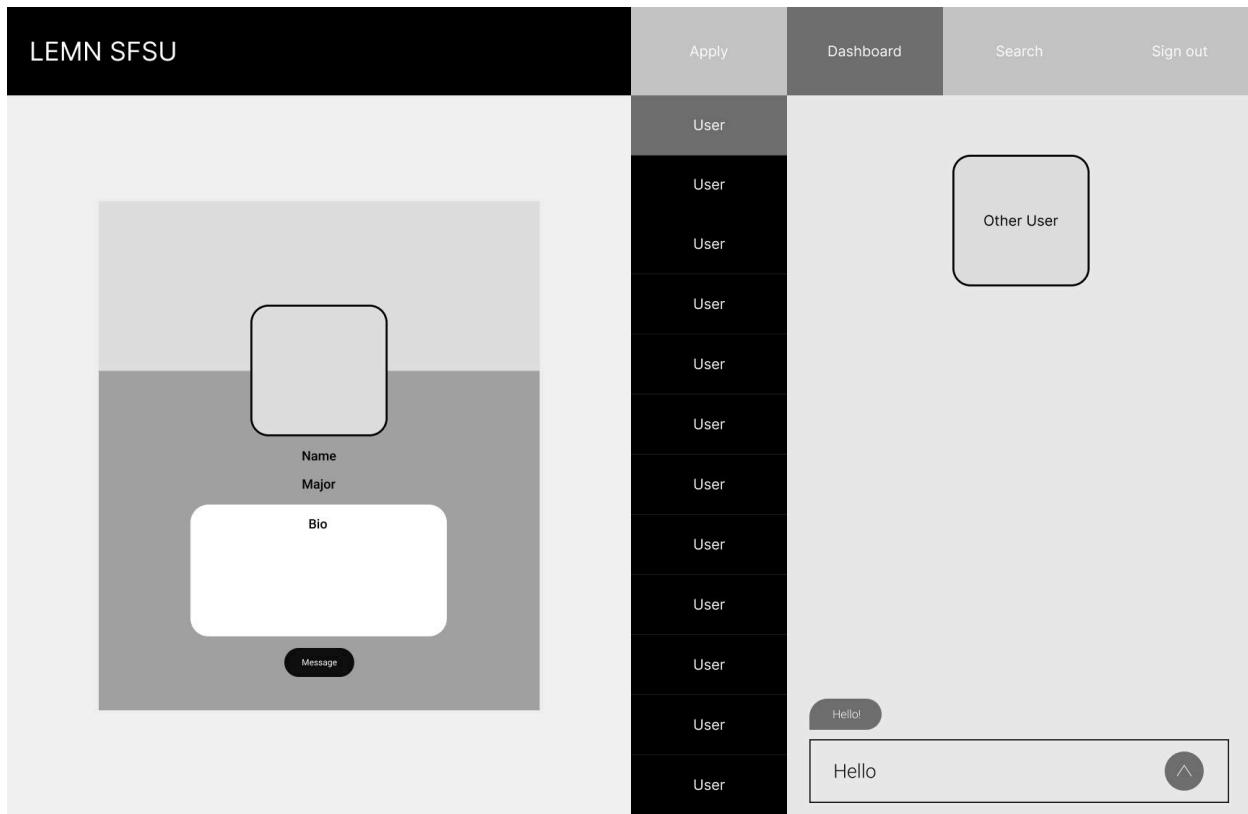
Once the information is entered the registered user

clicks on the Sign in button to proceed. It also has the forgot password option. The registered user enters the information to Sign in/ Login.



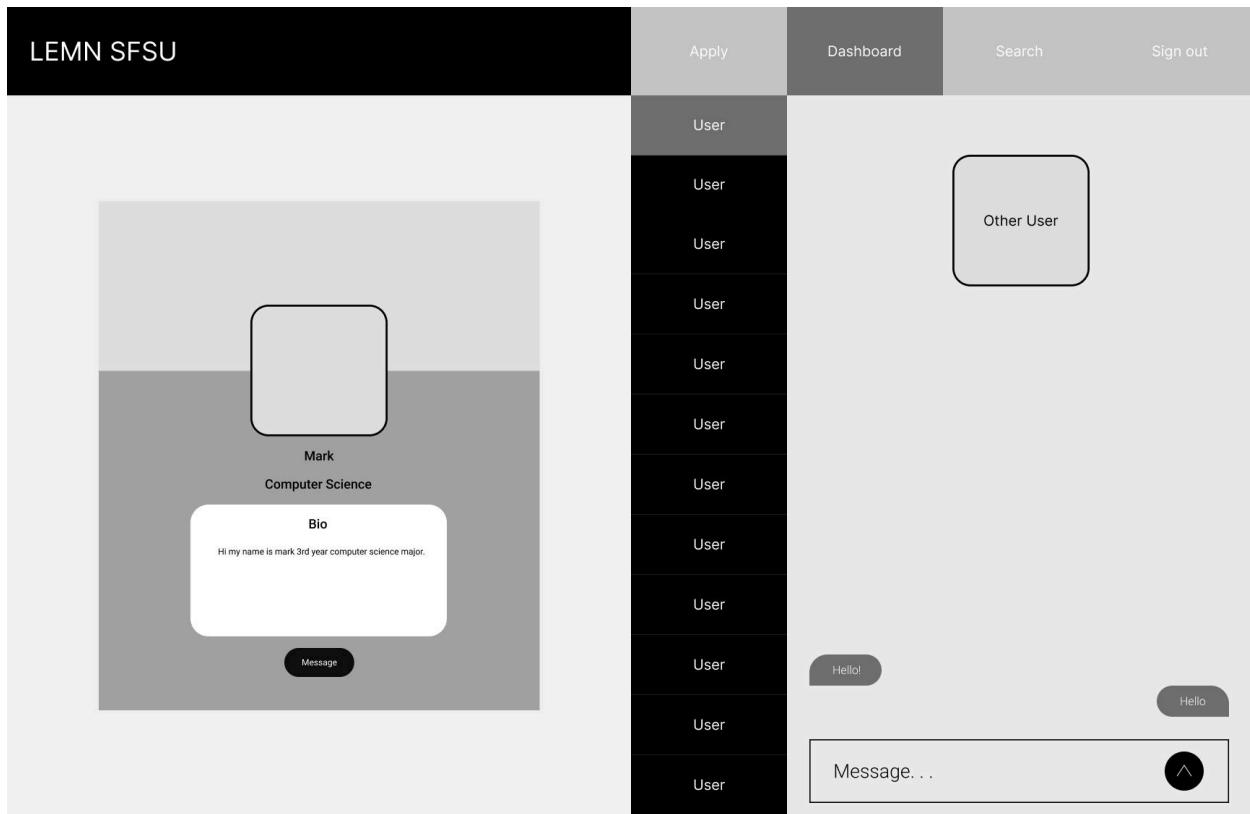
Messaging

Once the user is logged in, it displays the dashboard. Dashboard portraits the users name, major, bio and Message button on the left hand side.



Once the user clicks on the Message

button it gives the option to select the other users to send message (depicted on right hand side).The user types in the message and clicks on send button to send the message.



Tutor Application Form

The registered user clicks on Apply on the navigation bar. It initiates the subject and course.

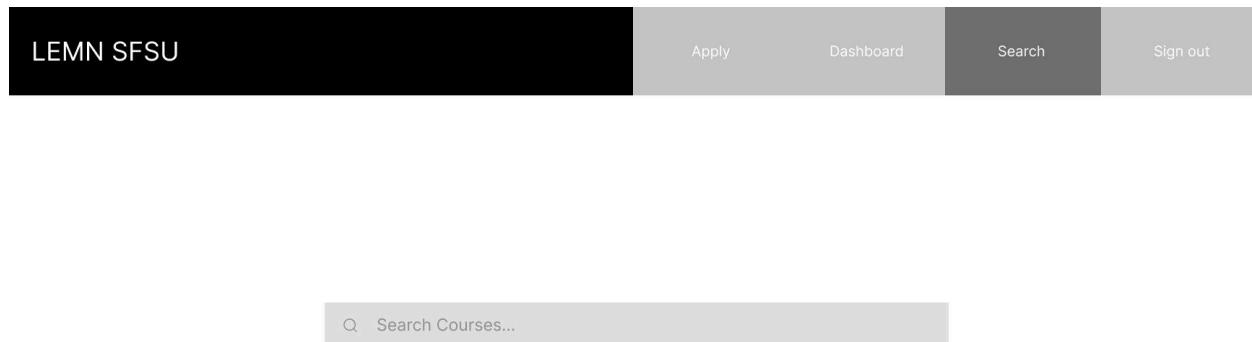
A form interface showing two input fields stacked vertically. The top field is labeled 'Subject' and the bottom field is labeled 'Course'. Below these fields is a solid black rectangular button labeled 'Apply' in white text.

The user enters the subject and course. User clicks on the Apply button to submit the form.

A screenshot of a search form. It consists of three stacked input fields. The top field contains the text "Computer Sciene". The middle field contains the text "CSC 648 SOFTWARE ENGINEERING". Below these is a dark grey button with the word "Apply" in white.

Browsing and Search

User clicks on Search on the navigation bar. It opens up the page for browse and search (to search courses, tutors, etc.)



User types in to search. It gives a drop down list of related search results for the user to select.2.

A search interface. At the top is a search bar with the placeholder "Search Courses...". Below it is a table with six rows, each containing a "Subject" column and a "Course..." column. All entries in the "Course..." column are identical.

Search Courses...	
Subject	Course...

High Level Architecture, Database Organization Summary

Registered User:

- Stores info about each user
- User id (PK): Numeric
- First name: Alphabetic
- Last name: Alphabetic
- Major: Alphabetic
- Minor: Alphabetic

Course: Represents each course within a subject

- Course id (PK): Numeric
- Class number (unique): Alphanumeric
- Course name (unique): Alphabetic
- Subject id (FK) -> Subject(Subject id)

User Course (junction table): links users to courses they are in

- User id (FK) -> Registered User(User id)
- Course id (FK) -> Course(Course id)
- (User id, Course id) (PK)

Subject: General academic subject area

- Subject id (PK): Numeric
- Subject name (unique): Alphabetic

Tutor Entry: Represents a tutor offering tutoring in a specific course

- Tutor entry id (PK): Numeric
- Tutor id (FK) -> Registered User(User id)
- Course id (FK) -> Course(Course id)

Message Entry: Represents a message between two users

- Message id (PK): Numeric
- Sender id (FK) -> Registered User(User id)
- Receiver id (FK) -> Registered User(User id)
- Message: Text

My suggestion is to only make the database 1NF compliant. Other normal forms are not required.

For the “Registered User” entity, the candidate key is {User_id, Course_id}

The table will look something like:

User_id	Course_id	First_name	Last_name	Major	Minor
1	1	Anurag	Sanadi	CS	None
1	2	Anurag	Sanadi	CS	None
2	1	Mukisa	Lubega		None
2	1	Mukisa	Lubega		None

Course_id	Class_num	Course_name
1	CSC-648	Software Engineering
2	CSC-675	Intro. To Database Systems

Key Risks

Skills Risk:

- Some team members are still developing their proficiency with React and Node.js, which may slow progress on complex features.
- *Resolution:* We plan to mitigate this by assigning tasks based on individual strengths, using tutorials and documentation for skill gaps, and pairing experienced members with those less familiar to accelerate learning.

Schedule Risk:

- Poor time management will cause delays in meetings and internal deadlines.
- *Resolution:* We will use Trello to visualize and track all tasks with clear due dates, hold weekly check-ins on Discord, and set mid-week checkpoints to ensure consistent progress.

Technical Risk:

- Integrating the front-end and back-end smoothly and managing deployment on AWS present potential technical challenges.
- *Resolution:* We will document all API endpoints early, test connections incrementally, and maintain shared debugging sessions between front-end and back-end leads.

Teamwork Risk:

- Occasional communication lags or uneven task distribution could affect workflow.
- *Resolution:* We will reinforce accountability by tracking task ownership on Trello, maintaining open communication through iMessage and Discord, and redistributing tasks when needed to balance workload.

Legal/Content Risk:

- Use of icons, images, or code snippets from external sources may pose copyright issues.
- *Resolution:* We will ensure all assets and code libraries are open-source or properly licensed and maintain documentation of all external resources used.

Project Management

For Milestone 2 and future deliverables, our team has maintained a consistent communication and coordination structure to stay organized and on schedule. We hold **weekly team meetings on Discord**, where we review progress, discuss upcoming goals, and address any blockers. Our work and documentation are stored within our **Discord server**, which serves as our central hub for file sharing, notes, and updates. For day-to-day communication and quick coordination, we use **iMessage** as our main channel for immediate responses.

As we move into Milestone 2, we plan to implement **Trello** to enhance our task management and accountability. Trello will allow us to assign specific tasks to team members, track deadlines, and visualize progress across both the front-end and back-end subteams. This will help ensure that all tasks are clearly defined, assigned, and completed on time.

To improve efficiency, we will continue to let the **front-end and back-end teams operate semi-independently** while coordinating on shared components such as API integration and database connectivity. Checkpoints and mid-week follow-ups will help us stay on track and maintain strong communication between both sides of development.

Use of GenAI

TASK	GenAI Tool	Usefulness	How It Was Used / Benefit
1. Drafting Milestone 2 Documentation	—	None	Not used.
2. Writing Project Descriptions and Objectives	ChatGPT (GPT-5)		Used to polish wording and organize the Project Management and Risk Management sections. Helped improve clarity and professional tone while keeping the team's original ideas intact.
3. Front-End Development (React / Tailwind)	—	None	No GenAI assistance used; all code written manually by the front-end team.
4. Back-End Development (Node.js / Express / MySQL)	—	None	Not used; code was developed and debugged internally by the back-end team.
5. Database / EER Design	—	None	Completed using manual diagramming and class materials only.
6. UI / UX Wireframes	—	None	Created manually by the front-end team without GenAI support.
7. Task Planning / Team Coordination	—	None	Managed through Discord and iMessage;

			Trello setup done manually.
--	--	--	-----------------------------

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 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	—
Team reviewed suggested resources before drafting Milestone 2	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 M2 and agree/understand it before submission	DONE	—
Team shared and discussed experience with GenAI tools among themselves	DONE	—

SW Engineering CSC648/848 Fall 2025

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

Team 06

<i>Role(s)</i>	<i>Members</i>	<i>Email</i>
<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>ashrestha@sfsu.edu</i>

Milestone 3 History

<i>Version</i>	<i>Date</i>	<i>Notes</i>
<i>Submission Date</i>	<i>10/31/2025</i>	<i>Submitted for m2 p1 grading</i>
<i>Revision Date</i>	<i>11/19/2025</i>	<i>Revised for m3 grading</i>

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

Purpose of Appendix II

This appendix documents how feedback from the Milestone 3 (M3) review was analyzed and translated into a concrete execution plan for the final phase of the project. The objective is to ensure completion of all **P1 features**, maintain **feature freeze discipline**, and deliver a fully deployed, production-ready system by Milestone 5 (M5).

This plan reflects industry-aligned SCRUM practices, emphasizing task ownership, checkpoints, and visible progress tracking.

Feature Freeze Statement

Effective **November 19**, the team entered **Feature Freeze**. From this point forward, only **P1 features** were permitted for discussion or implementation. No new features, enhancements, or exploratory work were allowed.

P1 Feature Scope Includes:

- Login and Registration
- Forgot Password (dummy implementation acceptable)
- Tutor Posting (one course per posting)
- Search and Search Results
- Messaging (context-aware, tied to postings)
- Dashboard (manage postings and messages only)

SCRUM Sprint Calendar (Final Sprint Phase)

Sprint 0 – Post-M3 Planning & Backlog Setup

Dates: Nov 19 – Nov 22

Objectives:

- Review and synthesize M3 feedback
- Complete Appendix II draft (internal)
- Establish SCRUM backlog in a visible PM tool

Key Tasks:

- Summarize M3 feedback into actionable items
- Create SCRUM backlog with owners and deadlines
- Assign checkpoints for each task

Deliverables:

- Internal Appendix II draft
- Active SCRUM board (GitHub Projects / Trello)

Sprint 1 – Core P1 Feature Completion

Dates: Nov 23 – Nov 30

Objectives:

- Complete all remaining P1 feature implementations
- Normalize UI/UX consistency across pages

Key Tasks:

- Implement persistent navigation bar with search
- Enforce required fields in tutor posting
- Add Forgot Password flow (dummy)
- Simplify dashboard to core functionality
- Improve commit messages and inline code comments

Checkpoints:

- Mid-sprint review (Nov 26)
- Sprint review (Nov 30)

Deliverables:

- All P1 features functionally complete
- Updated codebase with improved documentation

Sprint 2 – Integration, QA, and Deployment

Dates: Dec 1 – Dec 10

Objectives:

- Deploy system to AWS deployment server
- Validate end-to-end workflows

Key Tasks:

- Deploy frontend and backend to AWS
- Verify homepage and search run from server
- Resolve deployment/IP stability issues
- Conduct QA testing across all P1 flows

Checkpoints:

- Deployment verification checkpoint
- QA validation checkpoint

Deliverables:

- Fully deployed system
- Stable server-hosted homepage and search

Sprint 3 – Final QA & Documentation

Dates: Dec 11 – Dec 16

Objectives:

- Finalize system quality
- Complete all documentation for M5

Key Tasks:

- Fix remaining bugs
- Perform final regression testing
- Finalize Appendix II and M5 documentation
- Confirm all checkpoints are met

Deliverables:

- Final, stable build
- Completed M5 documentation package

Team Lead Responsibilities

Throughout the final sprint phase, the Team Lead was responsible for:

- Maintaining the SCRUM backlog and task assignments
- Monitoring progress and enforcing deadlines
- Ensuring full team engagement
- Communicating risks or blockers immediately
- Ensuring all checkpoints were met on time

Conclusion

This execution plan demonstrates a structured response to M3 feedback, emphasizing focus, discipline, and delivery. By enforcing feature freeze, managing work through SCRUM practices, and prioritizing deployment readiness, the team positioned itself to meet all M5 requirements within the given timeline.

Final Submission Date: December 17

SW Engineering CSC 648-848 Fall 2025

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

Team 06

<i>Role(s)</i>	<i>Members</i>	<i>Email</i>
<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 4 History

<i>Version</i>	<i>Date</i>	<i>Notes</i>
<i>Submission Date</i>	<i>12/17/2025</i>	<i>Submitted for m4 grading</i>

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:

The screenshot shows an email interface with the following details:

- From:** Zixiong Huang (ZH)
- To:** Grishma Maheshbhai Thumar
- Date:** Wed 12/17/2025 2:10 PM
- Actions:** Reply, Reply all, Forward, ...

The email body contains the following text:

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](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

[Summarize](#)

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

...

[Reply](#) [Forward](#)

The inline comments in GitHub for peer review:

Line: 27-32, 44,45 and 50

```

14 export const searchTutors = async (req, res) => {
15   const { q } = req.query;
16
17   let query = `SELECT te.tutor_entry_id, ru.First_name, ru.Last_name, ru.Portrait_path,
18   c.Course_name, c.Class_num
19   FROM Tutor_Entry te
20   JOIN Registered_User ru ON te.tutor_ID = ru.User_id
21   JOIN Course c ON te.course_id = c.Course_id
22   `;
23
24   const params = [];
25
26   /*The query structure is clean and parameterized
27   *
28   *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)
29   *and extend the WHERE clause instead of relying on text matching.
30   */
31
32   if (q) {
33     const searchTerm = `%${q}%`;
34     query += ` WHERE ru.First_name LIKE ?
35       OR ru.Last_name LIKE ?
36       OR c.Course_name LIKE ?
37       OR c.Class_num LIKE ?`;
38     params.push(searchTerm, searchTerm, searchTerm, searchTerm);
39   }
40
41   params.push(searchTerm, searchTerm, searchTerm, searchTerm);
42 }
43

```

The screenshot shows a GitHub file viewer for the file `search.controller.js`. The left sidebar lists the project's directory structure. The right pane shows the code for the `search.controller.js` file.

```
14  export const searchTutors = async (req, res) => {
39      OR c.Class_num LIKE ?
40      `;
41      params.push(searchTerm, searchTerm, searchTerm, searchTerm);
42  }
43
44 // If a tutor teaches multiple courses, this query may return multiple rows for the same tutor.
45 // Depending on frontend expectations, we may want to group results by tutor and aggregate Course_name values into a single subjects array.
46 const [rows] = await pool.query(query, params);
47
48 // Map rows to frontend-friendly format
49
50 // The mapping to a frontend-friendly format is clear and helpful.
51 const tutors = rows.map(row => ({
52     id: row.tutor_entry_id,
53     name: `${row.First_name} ${row.Last_name}`,
54     image: row.Portrait_path
55     ? `${req.protocol}://${req.get('host')}/images/${row.Portrait_path}`
56     : "", // fallback if no image
57     sessionsCompleted: 0, // you can calculate if needed
58     subjects: row.Course_name ? [row.Course_name] : [], // string array for frontend
59 }));
60
61 res.json(tutors);
62 } catch (err) {
63     console.error(err);
64     res.status(500).json({ error: "Database query failed" });
65 }
66 };
```

Self-check on best practices for security

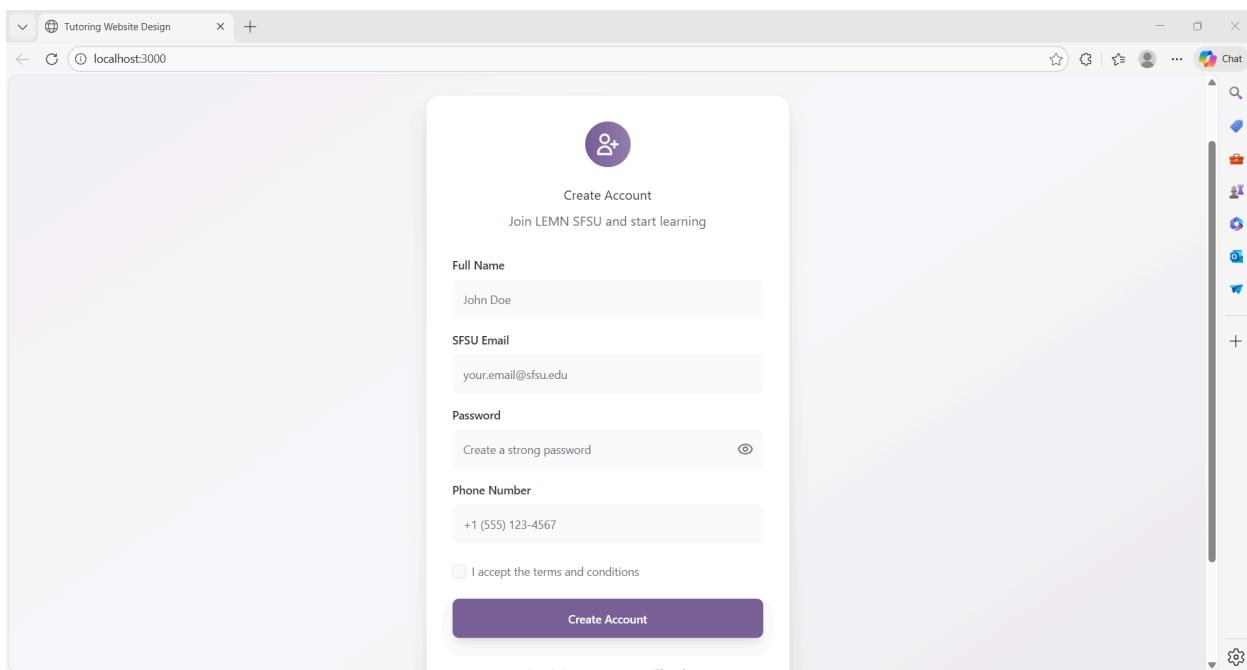
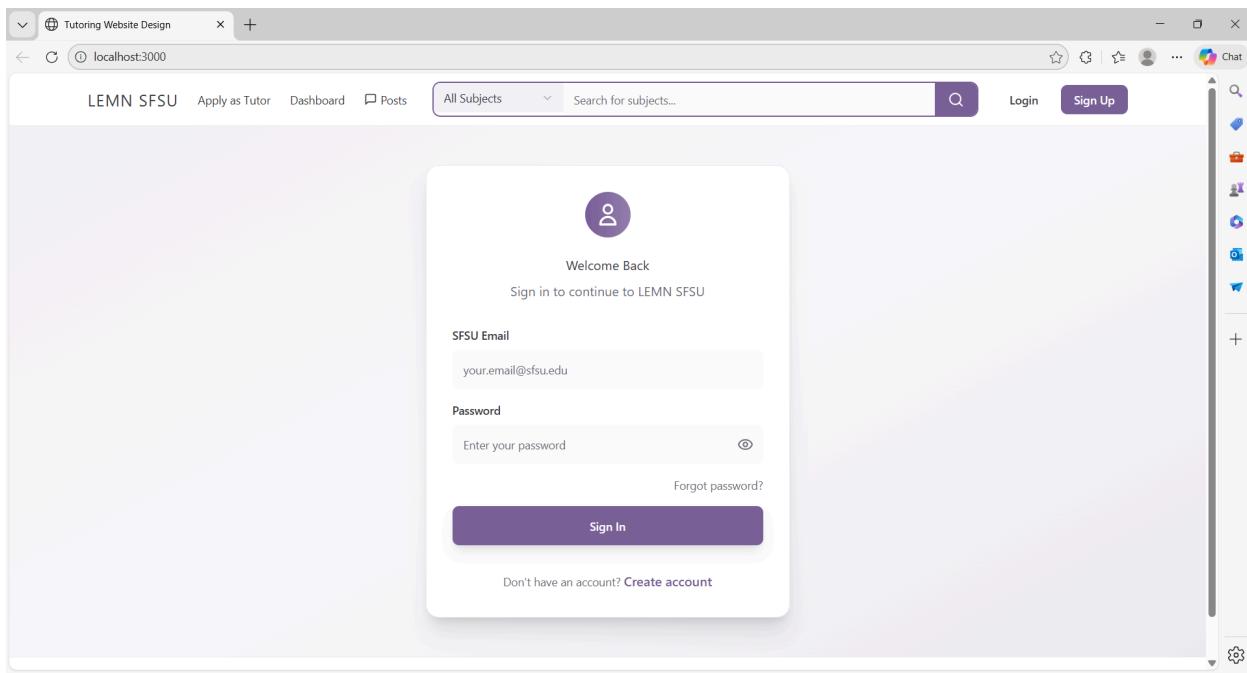
Assets to be protected	Types of possible attacks	Consequences of security breach	Strategy to protect asset
User data	SQL injection in register page	Sensitive data like email and name gets leaked	input validation for register form
Tutor listing data	SQL injection in apply as tutor form	Data like tutor 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.

Product Screen Shots:



Tutoring Website Design

localhost:3000

LEMN SFSU Apply as Tutor Dashboard Posts All Subjects Search for subjects... Login Sign Up Chat

Find Your Perfect Tutor at SFSU

Connect with experienced peer tutors across all subjects. Get personalized help and excel in your classes.

6 items found

Sarah Johnson
150 sessions completed
Calculus, Linear Algebra, Statistics
Message

Michael Chen
120 sessions completed
Computer Science, Python, Java
Message

Emily Rodriguez
180 sessions completed
Biology, Chemistry, Anatomy
Message

A screenshot of a web browser showing a search results page for tutors at SFSU. The page features a header with navigation links like 'LEMN SFSU', 'Apply as Tutor', 'Dashboard', 'Posts', 'All Subjects', and search bar. Below the header is a main title 'Find Your Perfect Tutor at SFSU' with a subtitle about connecting with experienced peer tutors. A message '6 items found' is displayed above three cards, each representing a tutor with a profile picture, name, session count, subjects taught, and a 'Message' button.

Tutoring Website Design

localhost:3000

your classes.

6 items found

Sarah Johnson
Calculus, Linear Algebra, Statistics

Hi! I'm Sarah Johnson. How can I help you today?
03:16 PM

Emily Rodriguez
180 sessions completed
Biology, Chemistry, Anatomy
Message

A screenshot of a web browser showing a detailed view of a tutor profile. The profile card for 'Sarah Johnson' is shown with her name, subjects, and a message window. The message window contains a greeting and a timestamp. The background shows other tutor profiles and a sidebar with various icons.

Welcome back, sf!

Track your tutoring sessions, view your courses, and connect with tutors.

Tutors Posts

Current Tutors

Mark Johnson Computer Science
Hi my name is Mark, 3rd year computer science major. I specialize in data structures and algorithms.
Message

Sarah Chen Mathematics
Math tutor with 2+ years experience. Love helping students understand calculus and linear algebra!
Message

Alex Rodriguez Physics
Physics grad student. Passionate about making complex concepts simple and fun to learn.
Message

Announcements

Course updates, discussions, and announcements

+ New Post

All Courses

PR Prof. Johnson Tutor 3 days ago
Project Milestone 1 Due Date Extended
Due to the feedback I've received, I'm extending the Milestone 1 deadline to Friday, Nov 22nd at 11:59 PM. Make sure to submit via Canvas and include all required documentation.
Posted 42 2

PR Prof. Johnson Tutor 5 days ago
Welcome to CSC 648 - Software Engineering!
Welcome everyone! This semester we'll be working on exciting team projects. Please review the syllabus and reach out if you have any questions. Looking forward to a great semester!
Posted

The screenshot shows a web browser window titled "Tutoring Website Design" at "localhost:3000". The header includes navigation links for "LEMN SFSU", "Apply as Tutor", "Dashboard", and "Posts". A search bar says "All Subjects" and "Search for subjects...". The user is logged in as "sf@sfsu.edu".

A purple banner at the top says "Welcome back, sf!" and "Track your tutoring sessions, view your courses, and connect with tutors."

The main content area has tabs for "Tutors" and "Posts", with "Posts" being active. It displays two recent posts:

- Looking for study partner for CSC 648 final project**
Hey! I need someone to collaborate with on the final project. Preferably someone strong in backend.
CSC 648 Posted 2 days ago
Edit Delete
- Question about React Hooks**
Can anyone explain useEffect dependencies? I'm having trouble understanding when it re-runs.
CSC 648 Pending 5 days ago
Edit Delete

A "Create New Post" button is located below the posts. The footer says "CSC 648 SEC 03 Fall 2025 Team 06". On the right side, there's a sidebar with various icons and a gear icon.

Tutoring Website Design

localhost:3000

LEMN SFSU Apply as Tutor Dashboard Posts All Subjects Search for subjects... Chat

Become a Tutor
Share your knowledge and help others succeed

Applications will be reviewed within 24-48 hours. We'll get back to you soon!

First Name *
e.g., John

Last Name *
e.g., Doe

Email *
e.g., john.doe@sfsu.edu

Phone Number *
e.g., (123) 456-7890

Availability *
e.g., Mon-Fri 3pm-6pm, Sat 10am-2pm

Subject *

Tutoring Website Design

localhost:3000

Personal Information

Full Name
John Doe

Email Address
sf@sfsu.edu

Major
Computer Science

Year
3rd Year

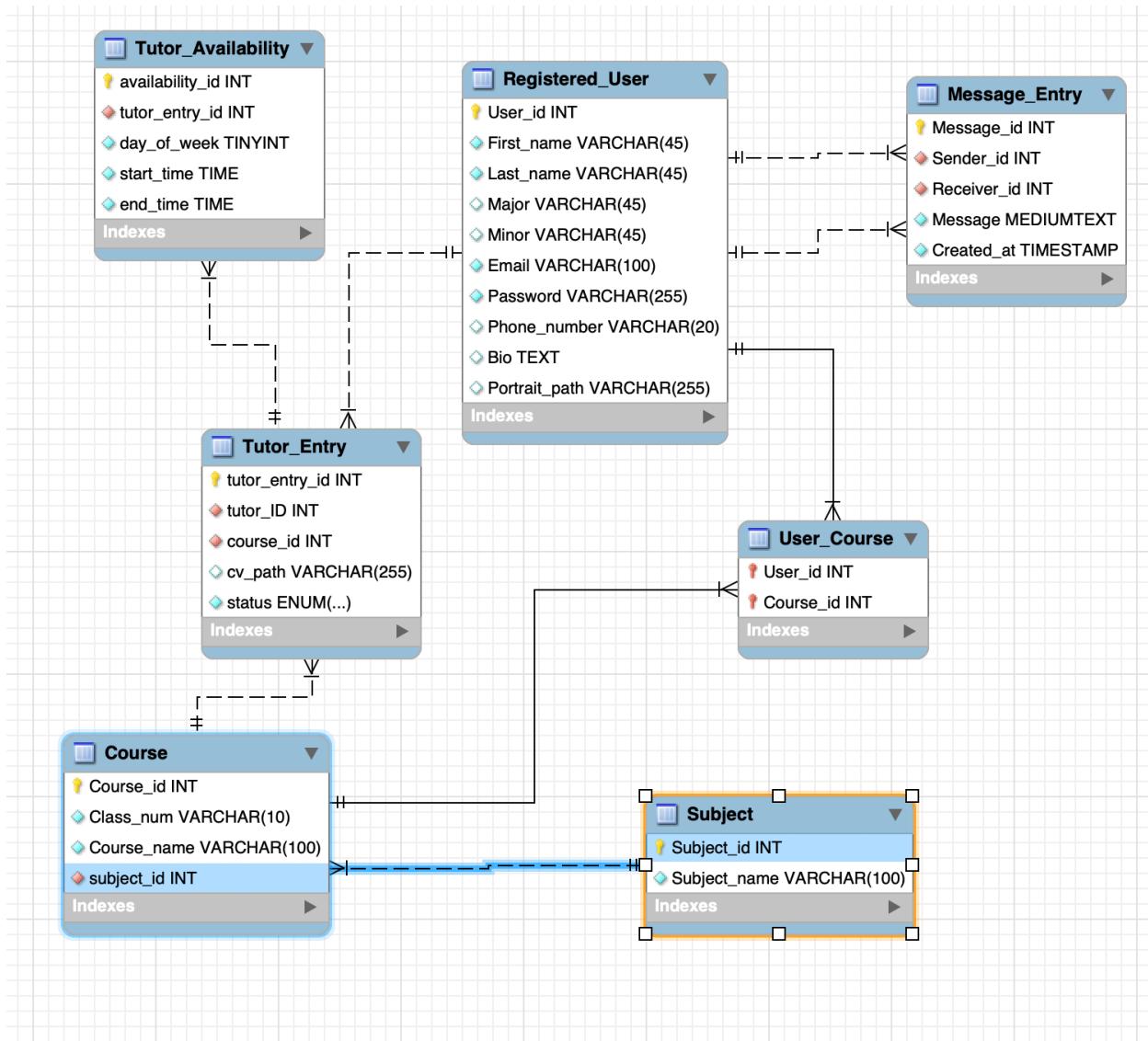
Phone Number (Optional)
Enter your phone number

Bio
Hi, I'm a computer science major passionate about software development and helping others learn.

96/500 characters

Save Changes Cancel

Database Organization:



Github Organization:

Main Branches in Team GitHub Repository

Our team used a **multi-branch Git workflow** to separate concerns and manage development effectively.

The main branches in our GitHub repository are:

- **main** – Primary production branch and default branch
- **frontend** – Front-end feature development and UI changes
- **backend** – Server-side logic, API development, and database integration
- **development** – Integration and testing branch used earlier in the project lifecycle

This branching strategy allowed parallel development while reducing merge conflicts and maintaining stability in the main branch.

(Screenshot 1: GitHub → Branches page showing main, frontend, backend, and development branches)

The screenshot shows the GitHub 'Branches' page for the repository 'CSC-648-SFSU / csc648-fa25-03-team06'. The page displays four main branches: 'main', 'frontend', 'backend', and 'development'. The 'main' branch is the default branch. The 'frontend' branch was updated 4 hours ago, 'backend' was updated last month, and 'development' was updated 2 months ago. Each branch row includes a 'Pull request' button and a three-dot menu icon.

Branch	Updated	Check status	Behind	Ahead	Pull request
main	22 minutes ago	Default			
frontend	4 hours ago	57 0			
backend	last month	86 0			
development	2 months ago	93 0			

Access to Master/Main Branch

Access to the **main (master) branch** was restricted to **team leads and administrators** to preserve code stability and prevent unauthorized or unreviewed changes.

From the repository access settings:

- **Admins** (full access to main branch):
 - Andrew Lubega (Team Lead)
 - Anurag Sanadi
- **Maintainers** (limited permissions, no direct main branch control):
 - Abinash Shrestha
 - Grishma
 - Zhuang

Contributors without admin rights worked on feature branches and submitted changes through pull requests when applicable.

(Screenshot 2: GitHub → Settings → Collaborators and Teams showing roles and access levels)

The screenshot shows the GitHub repository settings for 'CSC-648-SFSU/csc648-fa25-03-team06'. The 'Access' tab is selected under 'Collaborators and teams'. The interface displays three main sections: 'Base role' (318 members), 'Direct access' (5 entities), and 'Organization access' (0 users and 0 teams). Below this, the 'Manage access' section lists five collaborators with their roles: Abinash Shrestha (maintain), Andrew Lubega (admin), Anurag Sanadi (admin), grishma2305 (maintain), and zhuang231 (maintain). The sidebar on the left provides navigation links for General, Access, Collaborators and teams, Code and automation, Security, and Integrations.

Project Management:

Tools Used

Our team primarily used **Discord** as our project management and coordination platform. Discord served as a centralized hub for **task assignment, progress tracking, team communication, and sprint coordination** throughout the project.

Key tools:

- **Discord** – project management, task tracking, sprint coordination, and communication
- **GitHub** – version control, issue tracking, and code collaboration
- **AWS EC2** – deployment and infrastructure management

How Discord Was Used for Project Management

We organized our Discord server using **dedicated channels** to support an Agile-style workflow and maintain visibility across the team:

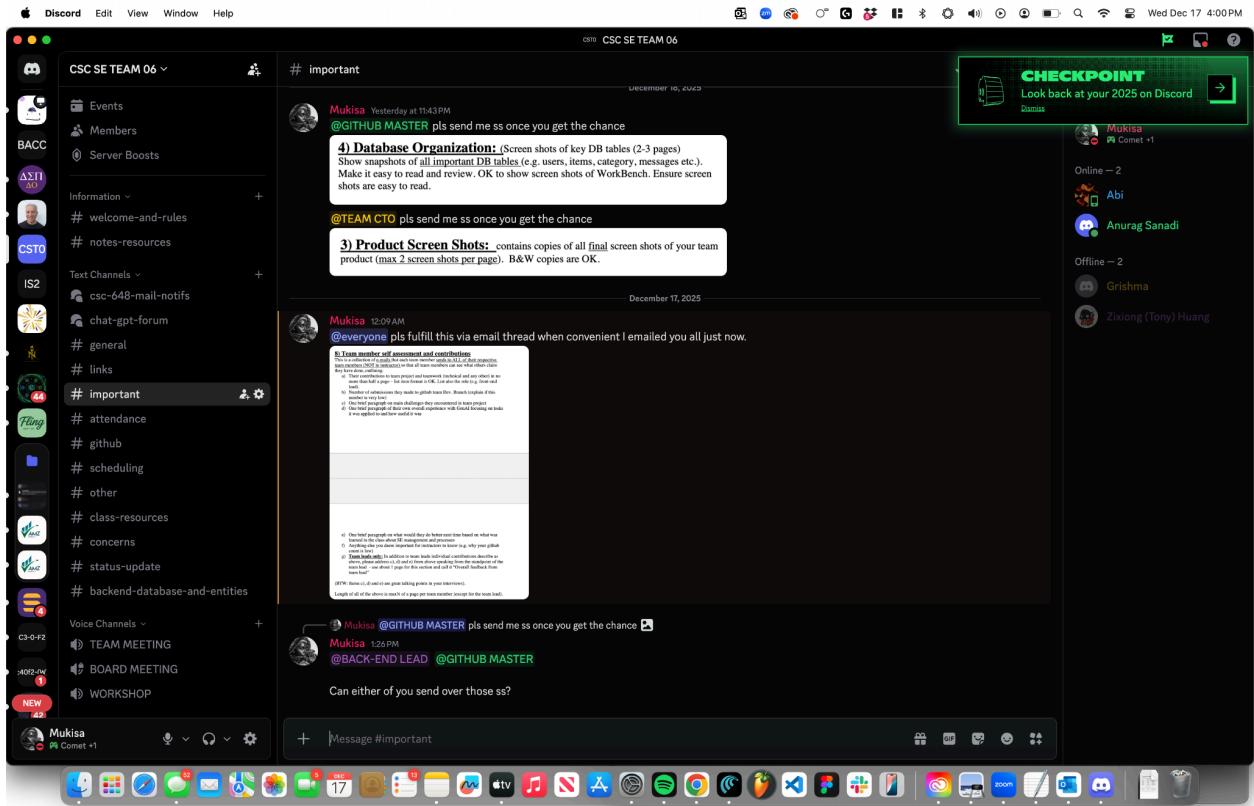
- **#announcements** – milestone updates, deadlines, and professor communications
- **#task-assignments** – clearly defined tasks with owners and deadlines
- **#standups / #progress-updates** – daily or weekly status updates and blockers
- **#dev-backend / #dev-frontend** – technical discussions and troubleshooting
- **#qa-testing** – bug reports and verification before merges

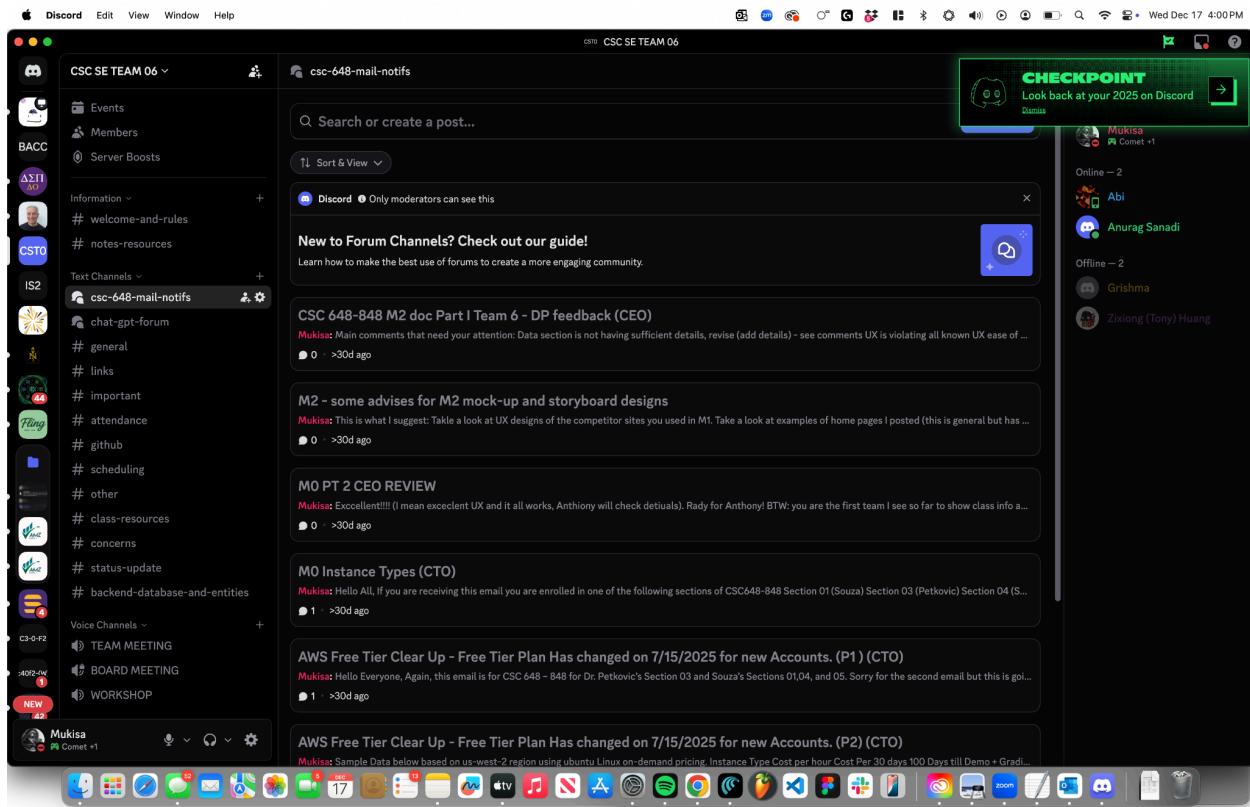
Tasks were assigned through structured messages and pinned posts, which included:

- Task descriptions and acceptance criteria
- Assigned team member(s)
- Target completion dates
- Links to relevant GitHub issues, commits, or pull requests

Pinned messages and threaded discussions allowed the team to **track progress, reduce miscommunication, and quickly reference active tasks**. GitHub links were used to validate task completion and ensure alignment between communication and code changes.

Screenshots Provided





Team Member Self Assessment & Contributions

Mukisa Lubega

Role: Team Lead / Project Manager

a) Contributions & Role

- Led overall project planning, coordination, and milestone execution
- Delegated tasks, tracked progress, and followed up on blockers
- Compiled and edited Milestones (M0–M2), including architecture, risk analysis, and GenAI usage logs
- Led backend/database planning and front-end integration discussions
- Launched and configured the AWS EC2 deployment environment
- Reviewed pull requests and monitored repository activity
- Oversaw system implementation using the LEMN stack:
 - Linux (AWS EC2)
 - Express.js / Node.js (backend API)
 - MySQL (relational database)
 - React (front-end)

b) GitHub Contributions

- 11 submissions to the team GitHub development branch
- As Team Lead, a significant portion of my work focused on planning, documentation, coordination, reviews, and deployment ownership, which does not always translate directly into a high commit count but was essential to project delivery.

c) Main Challenges

Key challenges included coordinating across different schedules, maintaining consistent engagement, and managing deployment and system integration complexity. Balancing leadership responsibilities with technical oversight required continuous communication and prioritization.

d) Use of GenAI

GenAI was used to support documentation drafting, architectural clarification, ideation, and conceptual troubleshooting. All outputs were reviewed and validated before inclusion.

e) What I Would Do Better Next Time

I would adopt a dedicated project management platform (such as Trello) earlier instead of relying primarily on Discord, in order to better track tasks, deadlines, ownership, and sprint progress.

f) Additional Notes

Several of my contributions were leadership- and process-focused and may not be fully reflected by GitHub metrics alone.

g) Overall Feedback from Team Lead

From the perspective of the team lead, this project closely mirrored real-world software engineering in both its technical and organizational challenges. One of the most significant challenges was coordinating a team with different schedules, experience levels, and comfort with the technology stack. While the team was capable, progress sometimes slowed due to inconsistent communication and unclear short-term priorities. This reinforced the importance of strong leadership, clear task ownership, and early alignment on expectations.

From a technical standpoint, system integration and deployment introduced the most complexity. Launching and managing the AWS EC2 environment required careful configuration and coordination across the backend, database, and front-end layers of the LEMN stack (Linux, Express/Node, MySQL, React). Small misalignments between components often caused downstream issues, highlighting the importance of early integration testing rather than waiting until features are “complete” in isolation.

GenAI was used primarily as a support tool throughout the project. As team lead, I found it most effective for refining documentation, clarifying architectural decisions, generating alternative design approaches, and assisting with conceptual troubleshooting. However, GenAI outputs required careful review, as they were not always context-aware or fully aligned with project constraints. This reinforced that GenAI should complement—not replace—engineering judgment, especially in collaborative team environments.

In terms of process improvements, one of the biggest lessons learned was the need for more formal project management tooling and structure earlier in the semester. While Discord was effective for quick communication, it was not ideal for tracking tasks, deadlines, or accountability. In future projects, adopting a dedicated project management platform such as Trello would provide clearer visibility into task ownership, sprint progress, and dependencies. This would help reduce confusion, prevent last-minute rushes, and improve overall workflow consistency.

Overall, this project emphasized that successful software engineering is not only about writing code, but also about communication, planning, and leadership. Early risk identification, structured task tracking, and consistent integration checkpoints would significantly improve outcomes. Serving as team lead provided valuable experience in balancing technical oversight with team coordination, and reinforced key software engineering management principles taught in the course.

Grishma

Role: Backend / Integration Support

a) Contributions & Role

- Assisted with backend development and API implementation
- Supported integration between frontend and backend components
- Helped debug backend issues and resolve integration errors
- Assisted with repository organization and code updates as needed
- Provided technical support during deployment and testing phases

b) GitHub Contributions

- Contributed commits primarily to backend-related branches
- Commit distribution reflects backend-focused responsibilities and collaborative development workflow rather than work concentrated in a single branch

c) Main Challenges

One of the main challenges was ensuring smooth integration between frontend and backend components. Coordinating changes across branches and keeping APIs aligned with frontend expectations required continuous communication and testing.

d) Use of GenAI

GenAI tools were used to assist with understanding backend logic, debugging errors, and clarifying implementation approaches. AI-generated suggestions were reviewed and validated before being applied.

e) What I Would Do Better Next Time

In future projects, I would focus on earlier and more frequent integration testing with the frontend team to identify issues sooner and reduce last-minute debugging.

f) Additional Notes

N/A

Anurag Sanadi

Role: Full-Stack / Deployment Support

a) Contributions & Role

- Created and managed branches in the team GitHub repository
- Deployed both frontend and backend code to the AWS server
- Edited frontend and backend code to ensure proper connectivity and integration

Most active development occurred in the frontend and backend branches, which I initially set up. As a result, the majority of my contributions are reflected there rather than in the main development branch.

b) GitHub Contributions

Most commits were made to the frontend and backend branches, consistent with the project's branching strategy rather than a lack of contribution.

c) Main Challenges

A major challenge was learning how to correctly switch and merge Git branches. Git requires precise sequencing of commands, and mistakes often resulted in aborted operations. Through trial and error, documentation, and practice, I learned to manage branches more smoothly.

d) Use of GenAI

I used GenAI extensively to guide me through Git commands, including creating, switching, and merging branches, managing pull requests, and resolving conflicts. This significantly accelerated my learning process.

e) What I Would Do Better Next Time

I would avoid switching branches without committing changes and apply best Git practices earlier in the project to improve workflow efficiency.

f) Additional Notes

N/A

Zixiong (Tony) Huang

Role: Backend Lead

a) Contributions & Role

- Helped decide and design the majority of functional requirements
- Designed and refined the database throughout the project
- Organized the backend code structure
- Implemented controllers for most backend endpoints
- Contributed to Milestone 4 documentation

b) GitHub Contributions

- 33 commits to the GitHub repository
- Most commits were made directly to the main branch at the discretion of the GitHub master

c) Main Challenges

One of the biggest challenges was aligning frontend and backend development. Establishing a clear interface between the two systems required consistent communication.

d) Use of GenAI

GenAI was used to suggest backend code solutions, which I carefully reviewed line-by-line. I frequently challenged and validated AI-generated suggestions to identify flaws. The AI-generated frontend structure increased backend integration complexity.

e) What I Would Do Better Next Time

I would communicate more frequently with the frontend team to ensure early alignment on data transfer and API expectations.

f) Additional Notes

N/A

Abinash Shrestha

Role: Frontend Lead

a) Contributions & Role

- Served as Frontend Lead for the project
- Designed and implemented the front-end using React
- Used Figma AI to create and refine UI/UX designs and prototypes
- Ensured consistency between design mockups and implemented components
- Coordinated with backend developers to align APIs with frontend needs
- Reviewed frontend-related changes and resolved UI issues

b) GitHub Contributions

- 9 commits to the frontend branch
- Contribution distribution reflects the project's branching strategy and frontend-focused responsibilities

c) Main Challenges

Integrating frontend changes with backend updates—especially near deadlines—was challenging. Maintaining UI consistency while backend functionality evolved required careful coordination.

d) Use of GenAI

Figma AI was used extensively for UI design and layout generation, allowing rapid prototyping. All AI-assisted designs were manually reviewed and refined before implementation.

e) What I Would Do Better Next Time

I would complete major frontend components earlier to allow additional time for testing, refinement, and smoother integration.

f) Additional Notes

This project strengthened my frontend development skills and reinforced the importance of early integration, planning, and cross-team collaboration in real-world software engineering.

New Mail

- Favorites
- All Accounts
 - 922859259@sfsu.edu
 - Inbox 3888
 - Archive 1
 - Drafts 30
 - Sent
 - Deleted Items 1
 - AI-STAARS
 - Conversation History
 - Hw
 - Junk Email 14
 - Work
 - alubega@aritzia.com
 - Saved Searches

Sent

Today

AS Abinash Shrestha... CSC 648 – Team Member Self-Assessment & Contributions.pdf 12/12/AM Hello Team Members, Att... 3888

MM To: Abinash Shrestha; CSC 648 – Team Member Self-Assessment & Contributions.pdf 73.9 KB

Re: CSC 648 – Team Member Self-Assessment & Contributions.pdf

Mukisa Martin Muwan...

Download · Preview

Hello Team Members,

Attached is my expanded overview.

Best,
Mukisa Lubaga
Team Lead – CSC 648

From: Mukisa Martin Mowanguzi-Lubaga <...>
Date: Wednesday, December 17, 2025 at 1:10:41 AM (PDT)
To: Abinash Shrestha <ashrestha@sfsu.edu>
Subject: CSC 648 – Team Member Self-Assessment & Contributions.pdf

Hi Team,

Below is my self-assessment and summary of contributions for the CSC 648 team project, shared with all team members as required.

a) Contributions & Role

Role: Team Lead / Project Manager

- Led overall project planning, coordination, and milestone execution
- Delegated tasks, tracked progress, and followed up on blockers
- Compiled and edited Milestones (M0–M2), including architecture, risk analysis, and GenAI usage logs
- Led backend/database planning and front-end integration discussions
- Launched and configured the AWS EC2 deployment environment
- Reviewed pull requests and monitored repository activity
- Oversaw system implementation using the LEMN stack:
 - Linux (AWS EC2)
 - Express.js / Node.js (backend API)
 - MySQL (relational database)
 - React (front-end)

b) GitHub Submissions

- 11 submissions to the team GitHub Dev branch
- As Team Lead, much of my work focused on planning, documentation, coordination, reviews, and deployment ownership, which does not always translate to a high commit count but was essential to project delivery

c) Main Challenges

Key challenges included coordinating across different schedules, maintaining consistent engagement, and managing deployment and integration complexity. Balancing leadership responsibilities with technical oversight required ongoing communication and prioritization.

d) Use of GenAI

I used GenAI to support documentation drafting, architecture clarification, ideation, and conceptual troubleshooting. It was useful for improving efficiency and clarity, but all outputs were reviewed and validated before inclusion.

e) What I Would Do Better Next Time

I would shift from using Discord as the primary coordination tool to a dedicated project management platform such as Trello to better track tasks, ownership, deadlines, and sprint progress.

f) Additional Notes

Several of my contributions were leadership-related.

New Mail

- Favorites
- All Accounts
 - 922859259@sfsu.edu
 - Inbox 3888
 - Archive 1
 - Drafts 30
 - Sent
 - Deleted Items 1
 - AI-STAARS
 - Conversation History
 - Hw
 - Junk Email 14
 - Work
 - alubega@aritzia.com
 - Saved Searches

Sent

Today

AS Abinash Shrestha... CSC 648 – Team Member Self-Assessment & Contributions.pdf 12/12/AM Hello Team Members, Att... 3888

MM To: Abinash Shrestha; CSC 648 – Team Member Self-Assessment & Contributions.pdf 73.9 KB

Re: CSC 648 – Team Member Self-Assessment & Contributions.pdf

Mukisa Martin Muwan...

Download · Preview

Hello Team Members,

Attached is my expanded overview.

Best,
Mukisa Lubaga
Team Lead – CSC 648

From: Mukisa Martin Mowanguzi-Lubaga <...>
Date: Wednesday, December 17, 2025 at 1:10:41 AM (PDT)
To: Abinash Shrestha <ashrestha@sfsu.edu>
Subject: CSC 648 – Team Member Self-Assessment & Contributions.pdf

Hi Team,

Below is my self-assessment and summary of contributions for the CSC 648 team project, shared with all team members as required.

a) Contributions & Role

Role: Team Lead / Project Manager

- Led overall project planning, coordination, and milestone execution
- Delegated tasks, tracked progress, and followed up on blockers
- Compiled and edited Milestones (M0–M2), including architecture, risk analysis, and GenAI usage logs
- Led backend/database planning and front-end integration discussions
- Launched and configured the AWS EC2 deployment environment
- Reviewed pull requests and monitored repository activity
- Oversaw system implementation using the LEMN stack:
 - Linux (AWS EC2)
 - Express.js / Node.js (backend API)
 - MySQL (relational database)
 - React (front-end)

b) GitHub Submissions

- 11 submissions to the team GitHub Dev branch
- As Team Lead, much of my work focused on planning, documentation, coordination, reviews, and deployment ownership, which does not always translate to a high commit count but was essential to project delivery

c) Main Challenges

Key challenges included coordinating across different schedules, maintaining consistent engagement, and managing deployment and integration complexity. Balancing leadership responsibilities with technical oversight required ongoing communication and prioritization.

d) Use of GenAI

I used GenAI to support documentation drafting, architecture clarification, ideation, and conceptual troubleshooting. It was useful for improving efficiency and clarity, but all outputs were reviewed and validated before inclusion.

e) What I Would Do Better Next Time

I would shift from using Discord as the primary coordination tool to a dedicated project management platform such as Trello to better track tasks, ownership, deadlines, and sprint progress.

f) Additional Notes

Several of my contributions were leadership-related.

CSC-648 Team member contribution

From: Anurag Sanadi <asanadi@sfsu.edu>

To: Mukisa Martin Muwanguzi-Lubega; Grishma Maheshbhai Thumar; Zixiong Huang; Abinash Shrestha

Hi team,

- a. Some of my contributions during this team project included
 - a. creating and managing branches of the github repo
 - b. deploying both the frontend and backend code to the AWS server
 - c. Editing both the frontend and backend code to connect them to each other
- b. This team project made full use of contributions from all members of the branch, but most of our development was done in the frontend and backend branch and so (since that's how I setup the repo), most of my contributions are in these branches.
- c. One main challenge I encountered was switching and merging branches. Git requires us to both switch and merge branches in a specific way otherwise, it aborts the requested command. Learning to do this was a challenge. I learnt to do this smoothly after lots of trial and error and extensive use of chatgpt and google overview.

d. I had very good experience using GenAI to complete my tasks. I primarily used GenAI to guide me through the various git commands used to main a git repository like creating, switching, merging branches, managing pull requests and resolving conflicts.

e. Now that I have a fairly good idea about managing github repositories, I will use my experience to not make the same mistakes I committed here like switching branches without committing changes in the present branch.

I.N.A

Thank you,
Anurag

Frontend Self-Assessment.pdf

Self-Assessment and Summary of Contributions – CSC 648 Team Project

Contributions & Role

Role: Frontend Lead

- I served as the **Frontend Lead** for the CSC 648 team project.
- I designed and implemented the front-end using **React**.
- I used **Figma AI** to create and refine UI/UX designs and prototypes.
- I ensured consistency between design mockups and implemented components.
- I coordinated with the backend team to align front-end functionality with APIs.
- I contributed **9 commits** to the **frontend** branch of the team GitHub repository.

GitHub Contributions

I pushed nine commits to the frontend branch, which reflects my primary responsibility for front-end development. My contributions were focused on UI components and design-related implementation, and the commit distribution aligns with the project's branching strategy.

Main Challenges

One of the main challenges was integrating front-end work with backend changes, especially near deadlines. Managing time effectively while ensuring UI consistency during ongoing development required clear communication and flexibility.

Use of GenAI

I used **Figma AI** for UI design and layout generation, which helped speed up the prototyping process. I also used GenAI tools in a limited, supportive role for design ideas and clarification, and all final work was reviewed and implemented by me manually.

What I Would Do Better Next Time (Paragraph)

In future projects, I would complete major front-end components earlier to allow more time for testing, refinement, and smoother integration with backend features. This would help reduce last-minute pressure and improve overall quality.

Overall Reflection

This project provided valuable experience working as a **Frontend Lead** in a team-based software engineering environment. It strengthened my front-end development skills and reinforced the importance of collaboration, planning, and early integration in real-world projects.

Sent

Today

Re: CSC 648 – Team Member Self-Assessment & Contributions

ZH **Abinash Shrestha...** CSC 648 – Te... 12/12AM Hello Team Members, Att...
Abinash Shrestha Hello team, This... Inbox
Zixiong Huang Hello team, This... Inbox
Grishma Mahesh... Hey Team Memb... Inbox
Mukisa Martin M... Hello Team Me... 12/12AM
Mukisa Martin M... Hi Team, Below... 12/03AM

Last Week

BL **Bing Luo** Follow-Up Reg... 12/11/25 Hi Professor Luo, I hope...
Emily Duenas MCS FOLLOW... 12/9/25 Good Afternoon, Thank y...
AK **Altura, Kimberley** (S... Register for Spr... 12/4/25 Hi Altura, Yes, I'm aware...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 Hi Altura, I'm in touch wit...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 I'm aware, I have pending...
KA **Kimberley Altura** (Sh... Register for Spr... 12/2/25 Hi Kimberly, The cause o...
DP **Dragutin Petkovic** [DPRC] CSC 64... 12/1/25 Hi professor, I believe I w...

This Month

AK **Altura, Kimberley** (S... Register for Spr... 12/4/25 Hi Altura, Yes, I'm aware...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 Hi Altura, I'm in touch wit...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 I'm aware, I have pending...
KA **Kimberley Altura** (Sh... Register for Spr... 12/2/25 Hi Kimberly, The cause o...
DP **Dragutin Petkovic** [DPRC] CSC 64... 12/1/25 Hi professor, I believe I w...

Last Month

AK **Altura, Kimberley** (S... Register for Spr... 12/4/25 Hi Altura, Yes, I'm aware...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 Hi Altura, I'm in touch wit...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 I'm aware, I have pending...
KA **Kimberley Altura** (Sh... Register for Spr... 12/2/25 Hi Kimberly, The cause o...
DP **Dragutin Petkovic** [DPRC] CSC 64... 12/1/25 Hi professor, I believe I w...

Re: CSC 648 Backend Lead.pdf

To: **Mukisa Martin Muwannguzi-Lubega;** **Abinash Shrestha;** **Anurag Saradi;** **Zixiong Huang**

Download • **Preview** **46.6 KB**

Team Member Self Assessment and Contributions

Name: Zixiong (Tony) Huang
Role: Backend Lead

a) Contributions:

- Helped decide and design the majority of the functional requirements.
- Designed and refined the database over the course of the project.
- Organized the structure of the backend code.
- Wrote the controllers for most of the endpoints in the backend.
- Wrote milestones 4

b) I have made 33 commits to the github. Most of these commits were made to main at the discretion of the github master.

c) One of the main challenges of this project was aligning the frontend with the backend. There needed to be a clear interface to connect the frontend to the backend and that required a lot of communication.

d) I used gptn to give me suggestions for the backend code, which I distrust and verified by reading through it line by line and having it explain the code and challenge itself. It usually finds most of the flaws in the code through challenging itself. Due to the use of use of a mostly ai generated frontend, the file structure was really complicated and made it very difficult to implement the backend.

e) I will try to communicate as much as possible to the frontend to make sure we are on the same page when it comes to what gets transferred between the backend to make both sides' jobs much easier.

f) Additional Notes

Sent

Today

Re: CSC 648 – Team Member Self-Assessment & Contributions

ZH **Abinash Shrestha...** CSC 648 – Te... 12/12AM Hello Team Members, Att...
Abinash Shrestha Hello team, This... Inbox
Zixiong Huang Hello team, This... Inbox
Grishma Mahesh... Hey Team Memb... Inbox
Mukisa Martin M... Hello Team Me... 12/12AM
Mukisa Martin M... Hi Team, Below... 12/03AM

Last Week

BL **Bing Luo** Follow-Up Reg... 12/11/25 Hi Professor Luo, I hope...
Emily Duenas MCS FOLLOW... 12/9/25 Good Afternoon, Thank y...
AK **Altura, Kimberley** (S... Register for Spr... 12/4/25 Hi Altura, Yes, I'm aware...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 Hi Altura, I'm in touch wit...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 I'm aware, I have pending...
KA **Kimberley Altura** (Sh... Register for Spr... 12/2/25 Hi Kimberly, The cause o...
DP **Dragutin Petkovic** [DPRC] CSC 64... 12/1/25 Hi professor, I believe I w...

This Month

AK **Altura, Kimberley** (S... Register for Spr... 12/4/25 Hi Altura, Yes, I'm aware...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 Hi Altura, I'm in touch wit...
AK **Altura, Kimberley** (S... Register for Spr... 12/3/25 I'm aware, I have pending...
KA **Kimberley Altura** (Sh... Register for Spr... 12/2/25 Hi Kimberly, The cause o...
DP **Dragutin Petkovic** [DPRC] CSC 64... 12/1/25 Hi professor, I believe I w...

Re: CSC 648 Backend Lead.pdf

To: **Mukisa Martin Muwannguzi-Lubega;** **Abinash Shrestha;** **Anurag Saradi;** **Zixiong Huang**

Download • **Preview** **53.6 KB**

Team Member Self Assessment and Contributions

Today at 12:31PM

Name: Zixiong (Tony) Huang
Role: Backend Lead

a) Contributions:

- Oversaw overall front-end architecture and design decisions
- Designed and refined UI/UX using Figma AI for faster prototyping
- Ensured consistency between design and implemented React components
- Coordinated with backend team to align APIs with front-end requirements
- Reviewed front-end related changes and helped resolve UI issues
- Supported integration of front-end features into the main application

b) GitHub Submissions:

- 6 submissions to the team GitHub Dev branch
- Most of my commits were made to the front-end branch, since my work primarily focused on UI and front-end development. As a result, my Dev branch commit count is lower but reflects the project's branching strategy rather than lack of contribution.

c) Main Challenges:

One of the main challenges was managing time near deadlines and integrating front-end work smoothly with backend changes. Coordinating updates across branches while ensuring the UI stayed consistent and functional required careful communication.

d) Use of GenAI:

We used Figma AI extensively for front-end design and layout generation. It was very useful for quickly exploring UI ideas, improving design quality, and saving time during the prototyping phase. All designs were reviewed and adjusted manually before implementation.

e) What I Would Do Better Next Time:

Next time, I would aim to complete major front-end work earlier to avoid last-minute submission. Finishing ahead of time would allow more room for testing, refinement, and smoother integration.

f) Additional Notes:

From: Mukisa Martin Muwannguzi-Lubega <azimwannguzilubega@sfsu.edu>