SW Engineering CSC 648-848 Fall 2024

StudyGator

Team 08

Team Lead: Bryan Lee (blee37@sfsu.edu)

Front-end lead: Min Ye Thway Khaing

Back-end lead: Nishi Suratia

GitHub master: Kenneth Wen

Milestone 2 Part 1

October 20, 2024

Date Submitted	Date Revised
10/20/2024	10/26/2024

1. <u>Executive Summary</u>

As the university semester progresses, it becomes increasingly demanding and hectic. Balancing the many coursework, project deadlines, and exam preparation can become overwhelming and stressful, especially with the limited time students typically have. Traditional options to seek academic assistance, like office hours, are often inadequate as they could conflict with other classes, extracurricular activities, or part-time jobs. In this rapid academic environment, students need flexible, easily accessible resources tailored not only to the specific course content but also to the unique teaching styles and expectations of each professor.

This is where our application, StudyGator, actively addresses the challenge of finding academic support tailored to specific courses or professors at SFSU. Users no longer have to spend hours looking through materials they don't understand, as StudyGator connects them directly with experienced SFSU tutors or alumni who are familiar with their specific coursework. This personalized approach not only streamlines efficient studying but also alleviates the academic burden, allowing students to tackle their coursework with confidence. StudyGator is built explicitly by and for SFSU students, offering the unique feature of allowing users to browse tutors offering services for specifically listed SFSU subjects and courses. StudyGator ensures that help is always just a click away, allowing SFSU students to focus on what matters most–their academic success.

Our team is a diverse group of students seeking to connect SFSU students with tutors more efficiently. Equipped with extensive backgrounds in software development and user-oriented design, we plan to deliver a tutoring application focused on simplicity and ease of use. Our focus is on creating a seamless tutor browsing experience and a way for SFSU tutors to put themselves out there. We believe that finding a tutor shouldn't feel like shopping for a new outfit—it should be quick, straightforward, and flexible enough to fit into students' hectic schedules.

2. Main Data Items and Entities

- **User type**
  - **Unregistered user—**The user is not registered on the website. They can browse/search tutor listings, register for an account, or log into an existing account. If they want to message a tutor, the system will prompt them to register or log in.
  - **Registered user** - Browse and message a tutor—inherits permissions of unregistered users. Registered users can fill out a form and attach their resume/CV to create a tutor listing, although it requires admin approval. They can view their messages, listings, and ratings. They can also delete a listing in their dashboard.
  - **Admin -** Has permissions for both registered and non-registered users. Can approve or deny all tutor listing applications (videos, CV, tutor's bio, or form). They can edit user's accounts and moderate user-generated content like reviews and ratings.
- **Tutor Listings** - Details submitted through the registered user's listing form to promote a tutor's services.
  - Tutor image
  - Tutor name
  - Tutor CV in PDF format (optional)
  - Tutor video (optional)
  - Listing title
  - Availability/time slots (optional)
  - Tutor description/information
  - SFSU-specific Subjects/Classes
  - Pricing
  - Reviews/Ratings (optional)
- **Dashboard** - The registered user's dashboard. They can view their inbound messages, public listings, and ratings.

- ○ View listings with a status displaying either public or pending approval from an admin.
- ○ Delete listing(s)
- ○ View booked sessions (optional)
- ○ View received messages
- ○ View tutor ratings (optional)
- **Messages -** A message sent by a registered user to a listing containing the body of the message, the name and email of the sender, and the listing to which it originated.
  - ○ Sender email
  - ○ Date created
  - ○ Id of the recipient
  - ○ Id of the post
  - ○ Body of the message
- **User Login Information** - Created during registration
  - ○ SFSU-specific email
  - ○ Password
  - ○ Name
- **Search categories -** These are categories a user can use to filter listings.
  - ○ Subjects
  - ○ Classes listed by tutors
  - ○ Tutor name/course name

3. Functional Requirements

- **Priority 1**
  - ○ <u>Unregistered user</u>
    - ■ Unregistered users shall be able to browse tutor listings.
    - ■ Unregistered users shall be able to view tutor listing details (description, images, more information, SFSU-specified courses, and subject).

- ■ Unregistered users shall be able to search for courses and tutor descriptions.
- ■ Unregistered users shall be able to filter tutor listings by price.
- ■ Unregistered users shall be able to register an account using an SFSU email.
- ■ Unregistered users shall be able to sign in to an existing account.
- ■ Unregistered users shall be able to click the message tutor button; however, when they attempt to send a message, the application prompts them to sign in or create an account.
  - ○ Registered user
    - ■ Registered users shall inherit all previous privileges.
    - ■ Registered users shall be able to message a tutor for their services from the listing.
    - ■ Registered users shall be able to fill out an application to create a listing.
    - ■ Registered users shall be able to delete their message(s) in the dashboard.
    - ■ Registered users shall be able to view their listing(s) in the dashboard.
    - ■ Registered users shall be able to delete their listing(s) in the dashboard.
  - ○ Admin
    - ■ Admin users shall inherit all previous privileges.
    - ■ Admin users shall be <u>required</u> to verify all tutor listing applications before they are public.
    - ■ Admin users shall be able to remove users or listings from the website.
    - ■ Admin users shall be able to view and manage all user data.
- **Priority 2**
  - ○ <u>Unregistered user:</u> No further requirements are needed. All specified requirements are already listed.
  - ○ <u>Registered user</u>
    - ■ Registered users shall be able to leave reviews and ratings for tutors after sessions.
    - ■ Registered users shall be able to upload their CV/Resume.

- ■ Registered users shall be able to manage their account information (name, email, password, and account deletion).
  - ○ <u>Admin</u>
    - ■ Admin users shall be able to delete user's ratings.
- **Priority 3**
  - ○ <u>Unregistered user:</u> No further requirements are needed. All specified requirements are already listed.
  - ○ <u>Registered user</u>
    - ■ Registered users shall be able to upload their "sale pitch" video.
    - ■ Registered users shall be able to book tutoring sessions based on tutor availability (scheduling feature).
    - ■ Registered users shall be able to reschedule or cancel a tutoring session.
  - ○ <u>Admin</u>
    - ■ Admin users shall be able to view website analytics.

4. UI Storyboards

Use Case 1: John is a new StudyGator user (Student)

John is an SFSU student looking for a tutor to help him prepare for his upcoming organic chemistry midterm. When he first visited our site, he found our landing page.
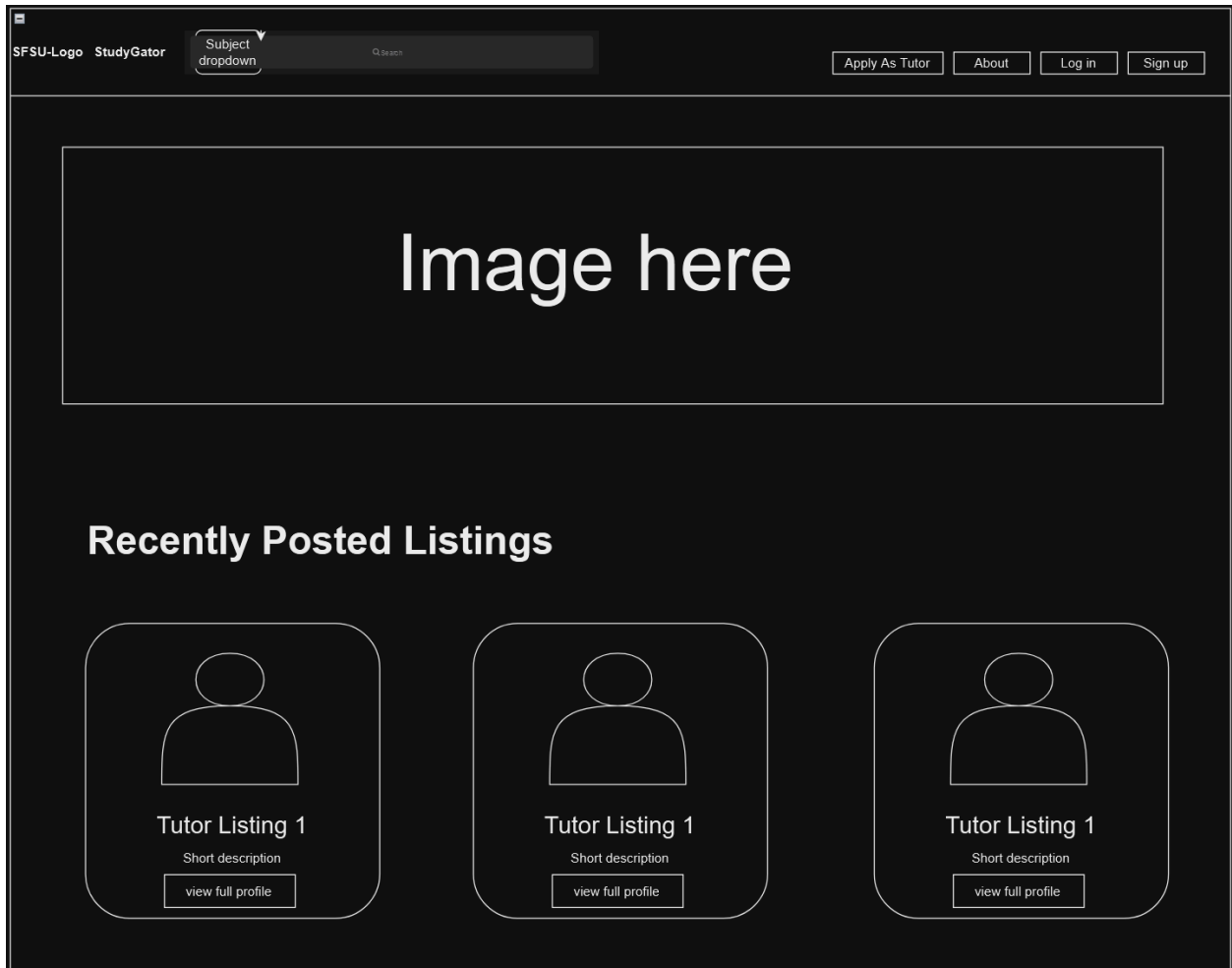
John immediately searches "organic chemistry" in the search bar, and the application redirects him to the tutor listings page, which shows tutors who specialize in organic chemistry. Here, he can browse and filter the listings using our underline unique feature, which allows searching by SFSU-related subjects, courses (e.g., Math 100, CSC 648), and prices.

John clicks on <u>View More</u> to see a tutor listing in more detail. Here, he can see short pitches from the tutor, the tutor's introduction/bio, images/videos, the tutor's resume, and he can message the tutor.
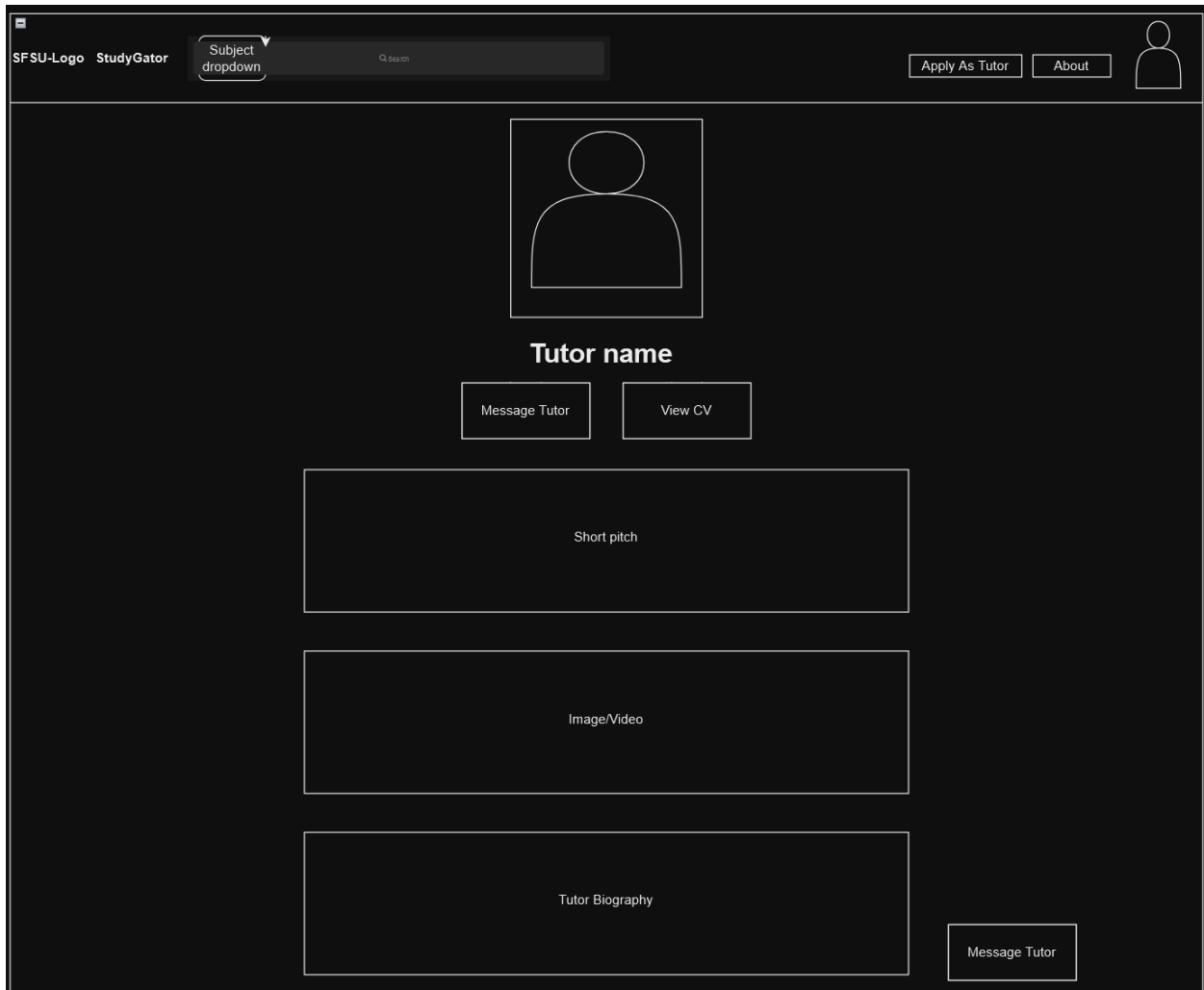
John is satisfied with the tutor he has chosen. He then clicks "message tutor," and a message form pops up. John writes his message and clicks "Send message," which prompts him to sign in (lazy registration).

SFSU-Logo StudyGator

Subject dropdown

Q Search

Apply As Tutor   About

email@sfsu.com

Tutor Subject request   [Subject line as title of tutor page prefilled]

Date: MONTH/DAY/YEAR TIME [Prefilled]

Hello, I am interested in your tutor services. Please contact me at aaa@sfsu.edu. (prefilled message)

Cancel   Send message

Tutor Biography

Message Tutor

Subject dropdown

Q Search

Explore   Apply As Tutor   About   Log in   Sign up

Sign in

Email address

email@example.com

Password

Password

☐ Remember me

Sign in

Don't have an account? Sign up

Forgot password?

Since John does not have an account, he clicks <u>Sign up</u> and creates a new account.

SFSU-Logo StudyGator

Subject dropdown

🔍 Search

Explore | Apply As Tutor | About | Log in | Sign up

Create An Account

Name

Name

Email address

email@example.com
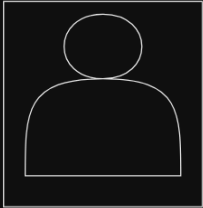
Password

Password

Confirm Password

Password

☐ Agree to our terms and services

Sign Up

Already have an account? Log in

After creating an account, John returns to the tutor's listing detail page and can message the tutor. The tutor will then contact the student and arrange an external appointment.

SFSU-Logo StudyGator

Subject dropdown

Q Search

Apply As Tutor   About

email@sfsu.com

Tutor Subject request   [Subject line as title of tutor page prefilled]

Date: MONTH/DAY/YEAR TIME [Prefilled]

Hello, I am interested in your tutor services. Please contact me at aaa@sfsu.edu.
(prefilled message)

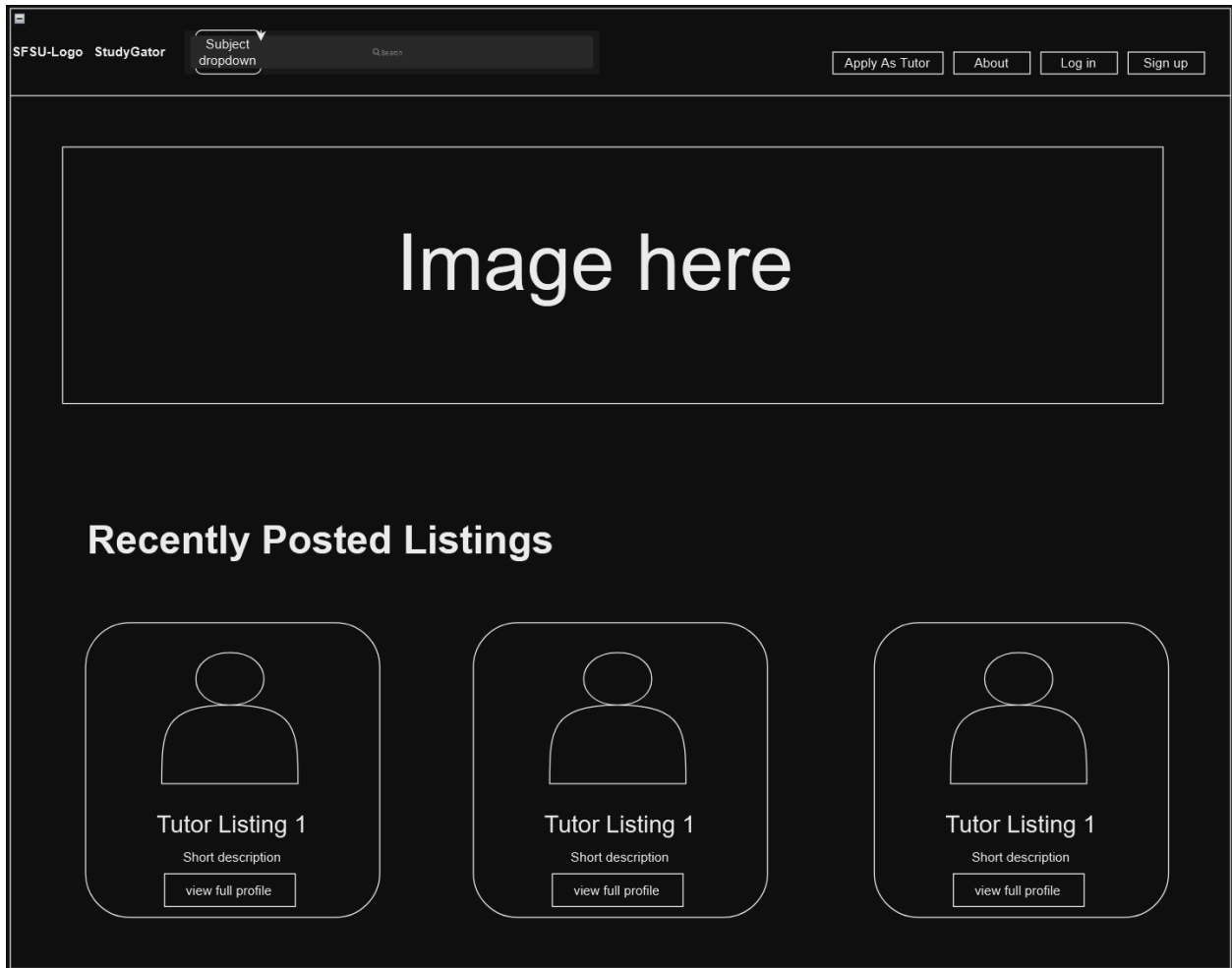Cancel   Send message

Tutor Biography

Message Tutor

Use Case 2: Benjamin applies to create a tutor listing.

Benjamin wants to create a tutor listing and opens the application, which brings him to the landing page.

*Benjamin signs into his existing **StudyGator** account.*

SFSU-Logo StudyGator

Subject dropdown

🔍 Search

Explore | Apply As Tutor | About | Log in | Sign up

Sign in

Email address

email@example.com

Password

Password

☐ Remember me

Sign in

Don't have an account? Sign up

Forgot password?

Once he signs in, he is brought back to the previous page he was on (landing page in this use case) and clicks the <u>Apply as Tutor listing</u> button on the navigation bar. The application will redirect him to the listing form, which he can fill out.

Once Benjamin submits the form, he must wait for admin approval. Once he receives approval, he will be able to see his new tutor listing on his dashboard and the search page.

SFSU-Logo StudyGator

Subject dropdown

🔍 Search

Explore   Apply As Tutor   About

Welcome StudyGator's Tutor Benjamin!

User icon replaces login button after logging in. Click on it causes dropdown to appear. Dropdown links to dashboard or option to sign out

Dashboard

Sign out

Messages	Listings	Ratings
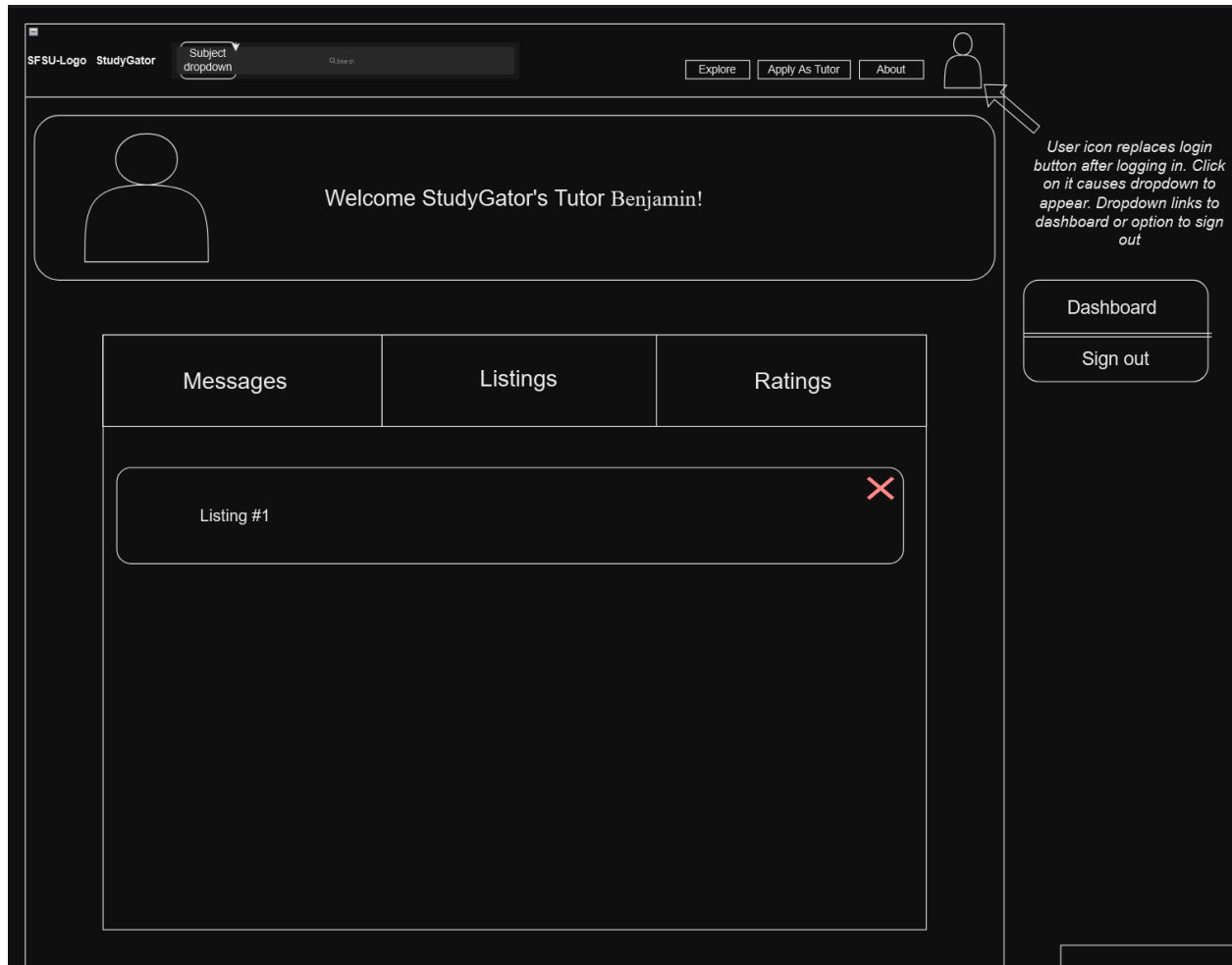
Listing #1
✕

Use Case 3: Benjamin uses his dashboard.

*Benjamin wants to check his new messages and delete a listing on **StudyGator**. He first opens up the site and is greeted with the landing page.*
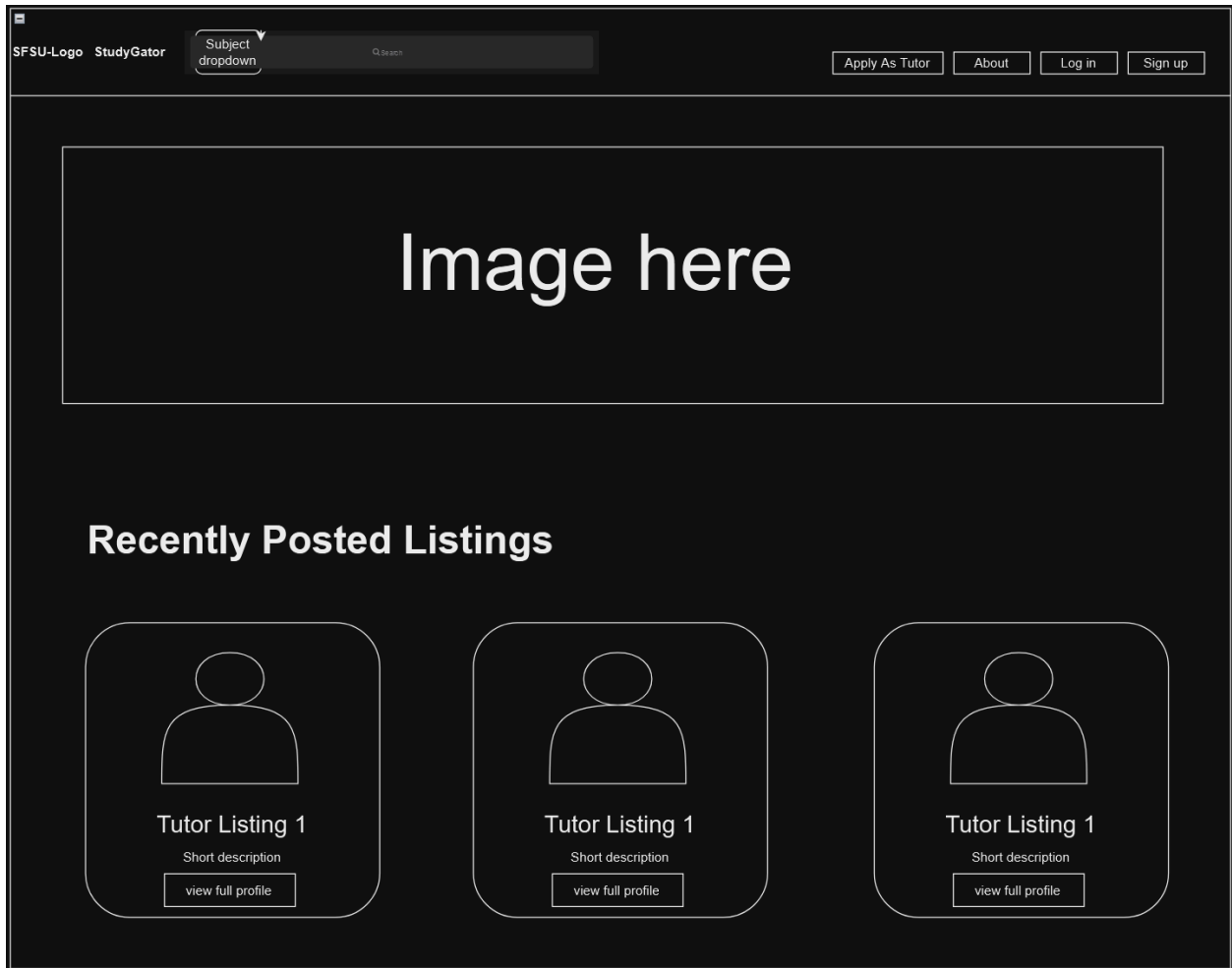
Benjamin goes to sign in to his existing account.

SFSU-Logo StudyGator   Subject dropdown   🔍 Search   Home   Explore   Services   About   Log in   Sign up

Sign in

Email address

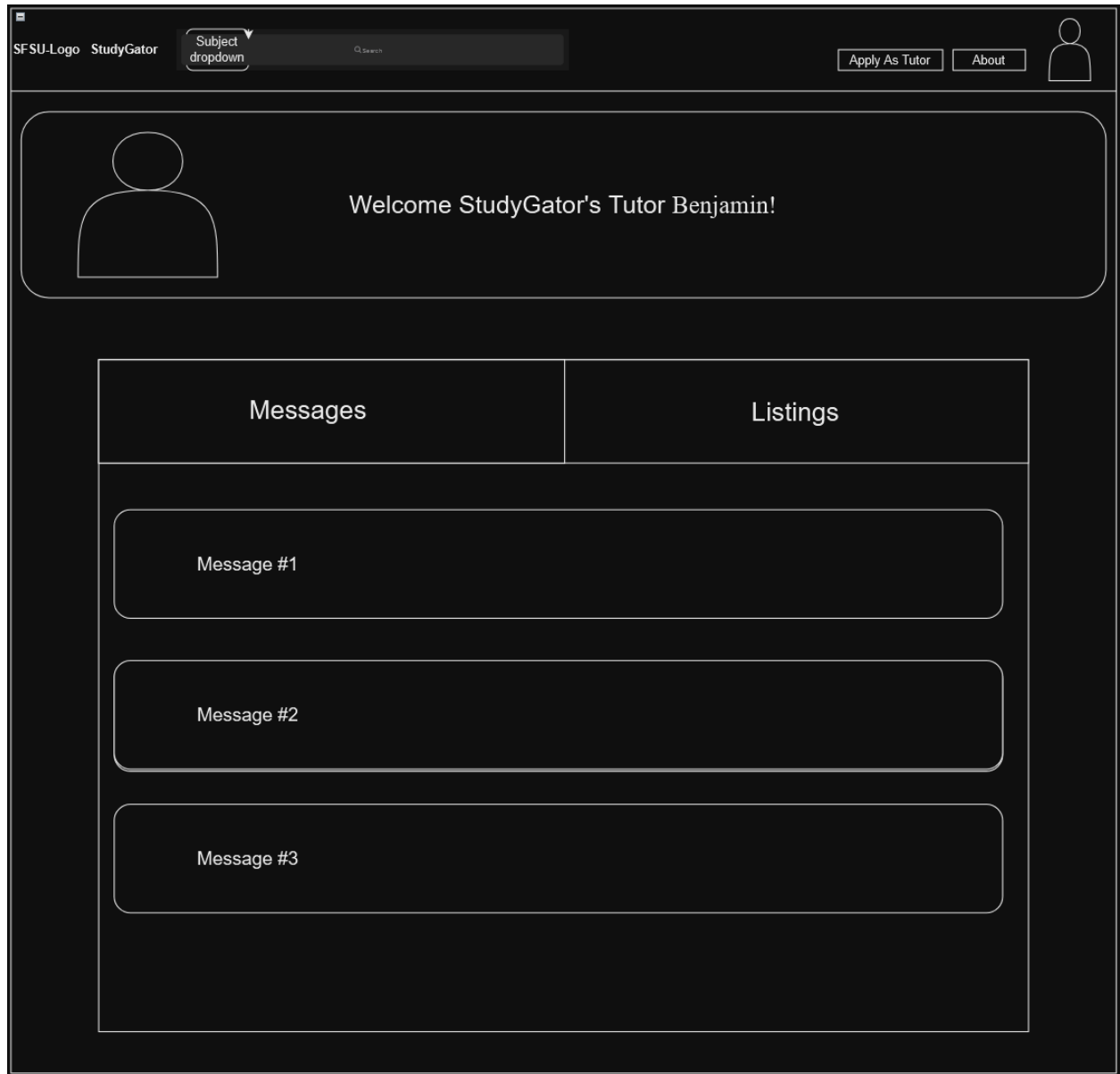email@example.com

Password

Password

☐ Remember me

Sign in

Don't have an account? Sign up

Forgot password?

Benjamin then uses the navigation bar to access his dashboard, where he can view his messages and listings.

After clicking one of the messages, Benjamin sees a message popup from a potential client who wants his tutor services. He will then contact the potential client externally.

Benjamin then moved to the "Listings" tab to check his current listings. He no longer wants to tutor for math, so he wants to remove Listing #2. He presses the "X" button to delete it and is prompted with a confirmation message.
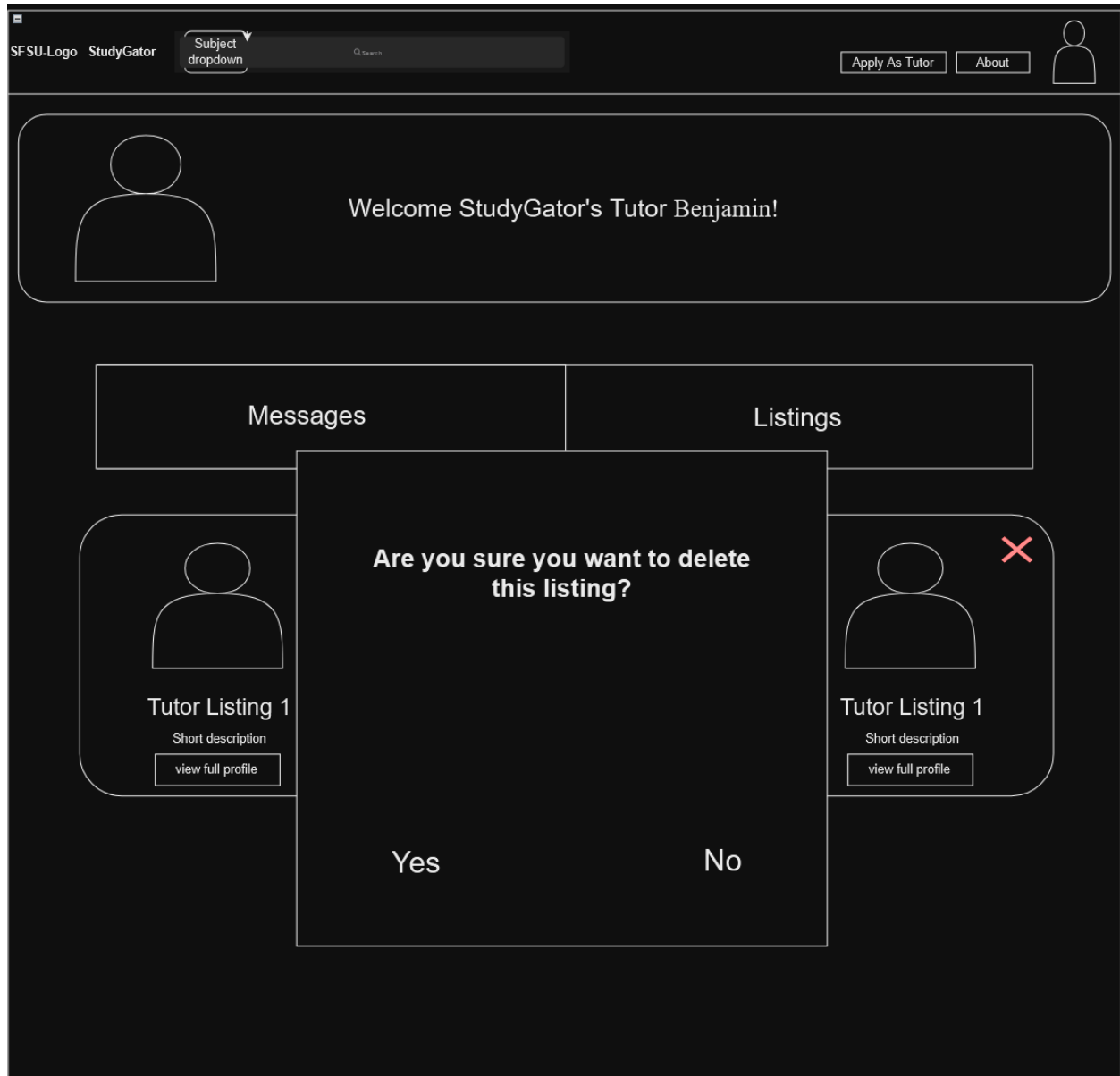
Use case 4: Mary approves tutors.
Admin will use MySQL Workbench (no storyboard for this one)
Mary checks the listings from users who filled out the tutor application form. After checking for malicious and inappropriate content, she would approve or deny the application.

5. <u>High-level Architecture, Database Organization summary only</u>

Database Organization:
- <u>Users (table)</u>
    - id
    - name
    - email
    - password
    - date_created
- <u>Tutor Listings (table)</u>
    - id
    - associated_user_id
    - image
    - description
    - date_created
    - subject
    - pricing
    - Potential priority two columns
        - attached_file (will be the CV)
        - attached_video
    - status (indicates approved or not)
- <u>Messages (table)</u>
    - id
    - recipient_user_id
    - sender_user_id
    - listing_id
    - date_created
    - content
- <u>Ratings (potential priority two table)</u>
    - id
    - target_user_id

- posted_user_id
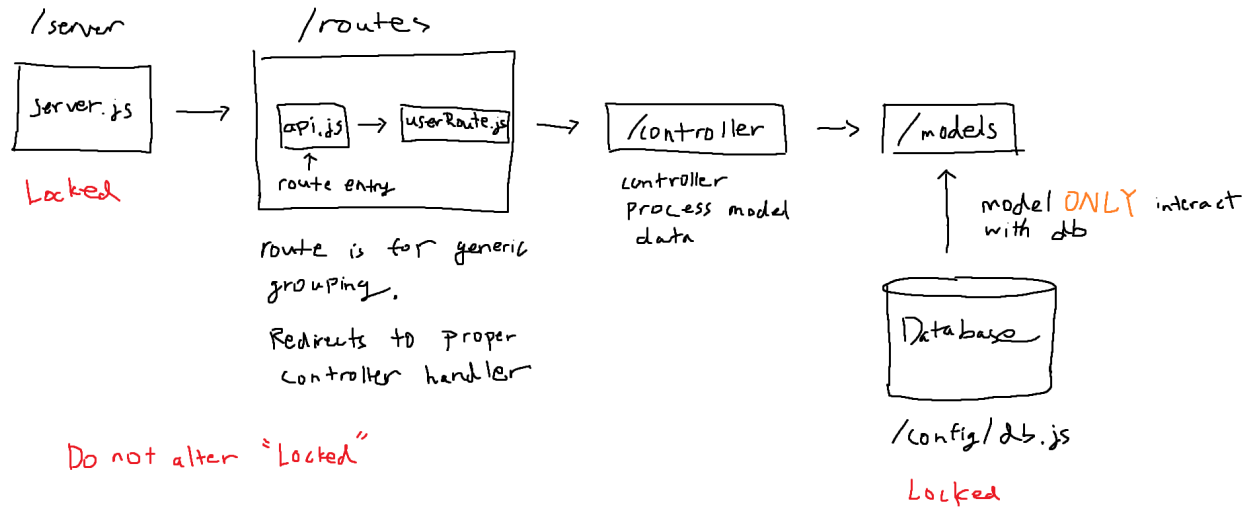- date_created
- rating_value
- comments

Media Storage:

➢ We will store media content via blobs. Some specific requirements for images are the following extensions: jpg, jpeg, png, and "webp." For potential priority two, the CV upload will only take PDF. Video uploads will also only take MP4, MOV, or "webm."

Search/filter architecture & implementation:

➢ We will simply use the search/filter architecture described in the lecture using the %like operator in MySQL. The listing's description, tutor's name, subject, and courses will be concatenated. The user can filter the items by only subject and courses (potentially pricing and ratings; however, these are priority 2).

MVC Design Pattern:

➢ Our StudyGator platform follows the Model-View-Controller (MVC) design pattern:
- Model: Manages data and database interactions.
- View: Handles the front end, built using ReactJs and Tailwind CSS.
- Controller: Processes HTTP requests and handles business logic with Node.js/Express.

➢ A rough plan of the design pattern (tentative and subject to change):

Diagram labels:
/server — server.js — Locked
/routes — api.js → userRoute.js
route entry
route is for generic grouping.
Redirects to proper controller handler
Do not alter "Locked"
/controller — controller process model data
/models — model ONLY interact with db
Database
/config/db.js
Locked

6. Key Risks

- **Skill risks**
  - A potential skill risk our team faces is working with unfamiliar tools and frameworks such as React, Tailwind CSS, and MySQL Workbench. We will resolve this issue by working together and assisting each other when trouble arises. We will/have assigned small practices/self-study in using these tools and frameworks to gain familiarity with them and mitigate future potential challenges. We will also keep our feature scope to the minimum to save time.

- **Schedule risk**
  - Conflicting schedules within the team are a potential schedule risk. We can only meet in person once a week and balance workloads from other classes/jobs. We plan to mitigate this by focusing on the basic high-priority functionalities and assisting each other when one of us gets too busy with different tasks. We will also be using Trello to stay up to date on project progress and tasks.

- **Technical risk**
  - Implementing the file system media storage is a potential technical risk. Since the team is unfamiliar with production-style development and MySQL, media storage will be the most significant challenge. To resolve this, the backend lead and

teammates who are more skilled in coding will do research. We will also try to make use of office hours.

- **Teamwork risk**
  - Due to the team's varying schedules, some teammates might be a bit more uncooperative with proposed deadlines. The quality of teamwork could also vary and not be up to standard with expectations. To mitigate this, we plan on having more frequent communication alongside a mandatory email/message status report on individual deadlines.

- **Legal/Content risks**
  - The use of the SFSU logo in the application's navigation bar poses a legal/content risk. However, the risk can be easily mitigated by explicitly implying that the project is a school demonstration project and not a separate private entity using this application for profit.

7. <u>Project Management</u>

In Milestone 2, we plan to manage the project's progress with Trello and continue using Discord for rapid communication. With Trello, we can organize tasks more efficiently and categorize them into to-dos, work-in-progress, and completed/done. Such management can help mitigate loose verbal proposed deadlines and allow for more flexible plan adjustments. It will enable the team to plan accordingly and adjust plans/schedules as necessary rather than the infrequent once-a-week meeting.

We will also continue to use Discord for rapid text communication. It offers flexible document/image uploads, discussions, announcement pings, and fast teamwork outside of campus. One additional proposal is a weekly progress/status report to ensure a smooth workflow. Overall, the Discord arrangement is going well and helped a lot with off-campus UI/UX design and milestone writing.

8. Use of genAI tools

genAI tool	ChatGPT-4o
Task	Grammar check/idea clarification
Useful rating	High

- Prompt: Can you fix the grammar/clarify the following text? An example:

> Fix fragmented sentence:
> Messages - Message sent by a registered user to a listing. It contains the body of the message, the name and email of the user who sent the message, and the listing it was sent

> Messages – A message sent by a registered user to a listing, containing the body of the message, the name and email of the sender, and the listing to which it was sent.

We did not use genAI tools much for this milestone. However, we used GPT to perform grammar/spell check for our writing. One noticeable help it provided is in the data item section. It helped clarify and get the point across to the reader more concisely and slightly expanded on existing ideas. It did spew some additional jargon, but that was minimal for this milestone and was easily filtered out manually.

9. Team Lead Checklist

- So far, all team members are fully engaged and attending team sessions when required. **[OK]**
- The team is ready and able to use the chosen back and front-end frameworks, and those who need to learn are working on learning and practicing. **[OK]**

- ○ The backend could use a little bit more practice, but we have not really started development yet.
- The team reviewed suggested resources before drafting Milestone 2. **[OK]**
- The team lead checked the Milestone 2 document for quality, completeness, formatting, and compliance with instructions before the submission. **[DONE]**
- The team lead ensured that all team members read the final Milestone 2 document and agreed/understood it before submission. **[DONE]**
- The team shared and discussed the experience with genAI tools among themselves. **[DONE]**