

## **Final Project for SW Engineering Class CSC 648-848 Fall 2024**

Team 08

**StudyGator**

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

Front-end lead: Min Ye Thway Khaing

Back-end lead: Nishi Suratia

GitHub master: Kenneth Wen

### **Milestone 5**

December 3, 2024

URL: <https://eclipseskakura.online/>

## **1. Product Summary**

**Name of product:** StudyGator

**The itemized list of ALL primary committed functions:**

- Unregistered user
  1. Unregistered users shall be able to browse tutor listings.
  2. Unregistered users shall be able to view more details about the tutor listing.
  3. Unregistered users shall be able to search for a tutor.
  4. Unregistered users shall be able to filter tutor listings by price.
  5. Unregistered users shall be able to register an account using an SFSU email.
  6. Unregistered users shall be able to sign in to an existing account.
  7. Unregistered users shall be able to write a message but must sign in to send it.
  8. Unregistered users shall be able to fill out the tutor application form but must sign in to submit.
- Registered user
  1. Registered users shall inherit all previous privileges.
  2. Registered users shall be able to message tutors about their services.
  3. Registered users shall be able to submit a tutor application to create a new listing.
  4. Registered users shall be able to view their message(s) in the dashboard.
  5. Registered users shall be able to delete their message(s) in the dashboard.
  6. Registered users shall be able to view their listing(s) in the dashboard.
  7. Registered users shall be able to delete their listing(s) in the dashboard.
- Admin
  1. Admin users shall inherit all previous privileges.
  2. Admin users shall be required to verify all tutor listing applications before they are public.
  3. Admin users shall be able to remove users or listings from the website.
  4. Admin users shall be able to view and manage all user data.

**URL to product:** <https://eclipseskura.online/>

## **2. Milestone Documents (M1 - M4)**

Starting below.

# **SW Engineering CSC 648-848 Fall 2024**

## **StudyGator**

Team 08

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

Front-end lead: Min Ye Thway Khaing

Back-end lead: Nishi Suratia

GitHub master: Kenneth Wen

## **Milestone 1**

Date Submitted	Date Revised
10/11/2024	10/13/2024

## **1. Executive Summary**

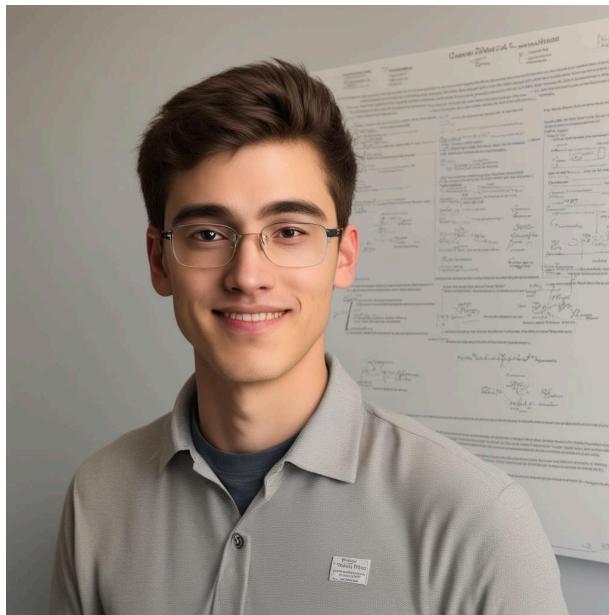
As the university semester progresses, it becomes increasingly demanding and hectic. Balancing the many coursework, project deadlines, and exam preparation can become overwhelming and stressful, especially with the limited time students typically have. Traditional options to seek academic assistance, 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. Personae**

John Johnson (User)



John Johnson is an SFSU student who is currently struggling in his organic chemistry class. He is proficient with computers and plays video games with his friends in his free time. He has a part-time job, so he has some money to afford tutoring; however, his budget is tight.

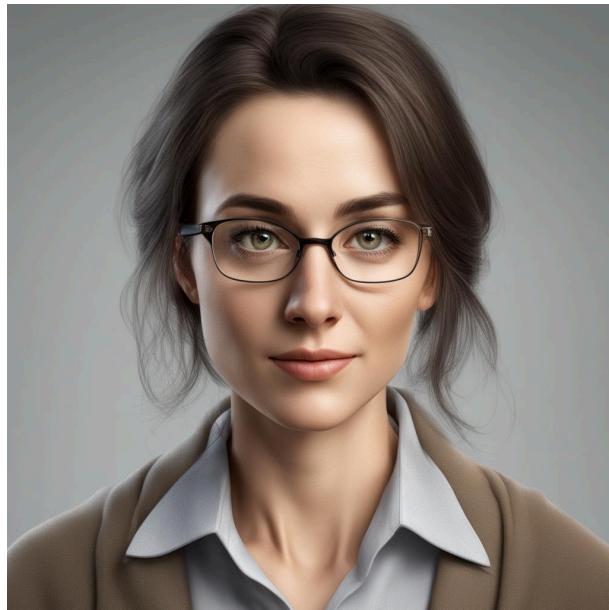
Goals:

- Passing his organic chemistry class
- Easily be able to access help from knowledgeable people

Pain Points:

- Budget-conscious
- Does not have time to sift through reviews and browse endlessly for tutors
- Stressed with little free time

Mary Merriam (Admin)



Mary Merriam has been a computer science professor at SFSU for over ten years. She is happy to help students and is willing to spend her free time helping them improve. Professor Merriam enjoys reading books and spending time with her two dogs. She is also familiar with MySQL workbench and has taught it in her lectures.

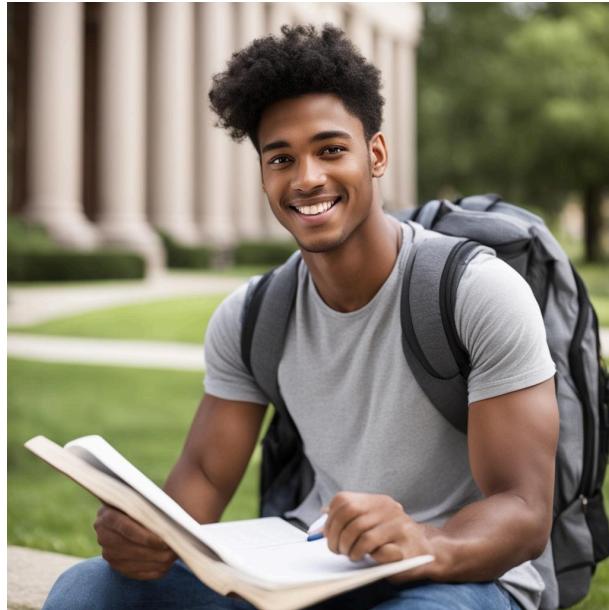
Goals:

- Help students improve academically
- Prevent inappropriate content from being posted

Pain points

- She does not want to lose too much of her free time
- Aware that not everyone is available during her office hours

Benjamin Benson (Tutor)



Benjamin is a senior at SFSU. He excels in his classes and helps out his friends during their study sessions. Benjamin realized the potential to earn some money through tutoring and decided to start at StudyGator. He has sufficient skills in navigating the internet.

Goals:

- Earn Money
- Improve his understanding by teaching
- Help others improve

Pain points:

- He does not know how to advertise his services

### **3. Use Cases**

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

John is looking for a tutor to help him prepare for the upcoming organic chemistry midterm. His friend recommended the StudyGator website, which advertised specifically to SFSU students. John uses the site's search and price filter option, entering the specific course name/ID. After browsing through a list of available tutors, John selects Benjamin. John clicks a button to message Benjamin about his services. As John clicks "send message," he is prompted to log in or create an account (lazy registration). After registering and logging in, the message is sent with the content he has written out.

#### **Use case 2: Mary (admin) approves tutor listings.**

Mary is a computer science professor proficient in MySQL Workbench, which is why she was hired to administer the StudyGator website. Through Workbench, she checks the listings of users who have filled out the tutor application form. After checking for malicious and inappropriate content, she approves or denies the application.

#### **Use Case 3: Benjamin (aspiring tutor) fills out the tutor listing form and receives approval.**

Benjamin wants to become a math tutor for Math100 as a side hustle. After logging into his account, he clicks on the "Apply As Tutor" button and fills out the form. He will have to wait 24 to 48 hours for an administrator to approve his application. The administrator approved his application the next day, so Benjamin can now see that his listing is available for other users to view.

#### **Use Case 4: Benjamin (tutor) receives a tutoring request (message).**

Benjamin checks his user dashboard for any new messages from potential students. He sees one message from John, who sent a message containing his contact information (email). Benjamin is able to see which tutor listing the message was sent from and contacts John through third-party services (outside of StudyGator).

## **4. Main Data Items and Entities**

- **User type**

- **Unregistered user**—The user is not registered to the website. They can browse/search tutor listings, register for an account, or log into an existing account. If they want to schedule a meeting with a tutor, they will be prompted to register/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 applications and listings (videos, CV, tutor's bio). They can edit user's accounts. Moderate user-generated content like reviews and ratings.

- **Tutor Listings** - Information inputted by the tutor about their 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

- **Dashboard**

- Profile
- Notifications
- User listings

- Booked sessions
- Session requests
- View messages
- **User Login Information** - Created during registration
  - SFSU-specific email
  - Password
  - Name
- **Search categories** - These are categories that a user can use to filter listings.
  - Subjects
  - Classes listed by tutors
  - Tutor name/course name

## **5. High-Level Functional Requirements**

### **Unregistered users**

1. Unregistered users shall be able to browse tutor listings.
2. Unregistered users shall be able to search and filter listings by subject, rating, and availability.
3. Unregistered users shall be able to register an account using an SFSU email.
4. Unregistered users shall be able to view tutor profiles, bios, and availability.

### **Registered users**

5. Registered users shall inherit all privileges of unregistered users.
6. Registered users shall be able to apply to become a tutor.
7. Registered users shall be able to message a tutor for their services from the listing.
8. Registered users shall be able to manage their account information (name, email, password, and account deletion).
9. Registered users shall be able to book tutoring sessions based on tutor availability.
10. Registered users shall receive notifications for messages or cancellations.
11. Registered users shall be able to leave reviews and ratings for tutors after sessions.
12. Registered users shall be able to reschedule or cancel a tutoring session.
13. Registered users shall be able to upload their CV/Resume.
14. Registered users shall be able to edit their tutor profiles (subjects, courses, availability, or bio)
15. Registered users shall be able to upload their bio videos.

### **Admin**

16. Admin users shall inherit all privileges registered users
17. Admin users shall be required to verify all tutor listings before they are posted.
18. Admin users shall be able to verify the prospecting tutor's application.
19. Admin users shall be able to remove users or listings from the website.
20. Admin users shall be able to view and manage all user data.

## **6. List of non-functional requirements**

1. Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0
2. Application shall be optimized for standard desktop/laptop browsers, e.g., must render correctly on the two latest versions of two major browsers
3. All or selected application functions shall 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 simple to use and intuitive
10. The application shall follow established architectural patterns
11. Application code and its repository shall be simple to inspect and maintain
12. Google Analytics shall be used
13. No email clients shall be allowed. Interested users can only message to sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application
14. Pay functionality, if any (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 2024. For Demonstration Only" at the top of the WWW page Nav bar. (Important so as to not confuse this with a real application).

## **7. Competitive analysis (functions/features only, not business or marketing)**

Feature	Our website <b>StudyGator</b>	Chegg Tutor	Tutor.com	Khan Academy	Wyzant
User Profile	++	+	+	+	-
Search Functionality	++	+	+	+	+
Tutor Matching	++	+	++	+	-
Session Scheduling	++	+	+	+	-
Campus focus (SFSU specific-features)	++	-	-	-	-

*Feature exists; ++ superior; - does not exist*

Our tutoring website offers unique value by focusing on SFSU-specific features, such as matching students and tutors within the SFSU community and integration with the university's scheduling. Competitors like Chegg, Tutors, and Wyzant provide robust matching and search functionality but lack the personalized, university-specific focus that our platform will offer. Additionally, the advanced filtering and customized scheduling features will make it easier for SFSU students to find and book sessions with tutors who have already taken the course, ensuring they receive support tailored to their specific academic needs.

## **8. High-Level System Architecture and Technologies used**

- Deployment cloud service we are using:
  - AWS EC2
  - Cloudflare and GoDaddy for custom domain
- Database:
  - AWS RDS and MySQL v8.0.39
- Web Server:
  - Nginx/1.24.0 with PM2 process manager
- Operating System:
  - Ubuntu 24.04
- Frontend frameworks:
  - ReactJS
  - Tailwind CSS
- Backend:
  - Node/Express v20.17.0
- Browsers we will support:
  - Google Chrome
  - Microsoft Edge
- Additional open source resources/APIS:
  - Flowbite for UI components and Google Analytics

## **9. Use of GenAI tools for Milestone 1**

genAI tool	ChatGPT-4o
Task	Grammar check/fix and brainstorming ideas for use cases
Useful rating	Medium

- Prompt: Can you fix the grammar/clarify the following text? [Insert fragment of the milestone one writing]

Although we do not have the prompt's history anymore, we basically had GPT perform grammar/spell check for our writing as well as to get considerable ideas for our use cases. One noticeable help it provided is in the last paragraph of the Executive summary. It helped clarify and get the point across to the reader more concisely and expanded on existing ideas.

Alternatively, GPT spewed a lot of random jargon and added unnecessary fluff to the writing, but that was filtered out and removed manually. Overall, it was helpful with grammar/spelling, but it still required a lot of intervention and correction to get the writing quality we desired.

genAI tool	DeepAi
Task	Image Generation
Useful rating	High

We also used AI to generate images for our personae. While the pictures are not perfectly realistic, they serve their purpose well by providing a face for our several personalities. Generating all three images took around five minutes. Overall, they are acceptable and were generated quickly despite their slight uncanniness.

## **10. Team and Roles**

Student Names	Email	Role
Bryan Lee	blee37@sfsu.edu	Team Lead
Min Ye Thway Khaing	mkhaing1@sfsu.edu	Front-end Lead
Kenneth Wen	kwen@sfsu.edu	GitHub Master
Nishi Suratia	nsuratia@sfsu.edu	Back-end Lead

## **11. Team Lead Checklist**

- So far, all team members are fully engaged and attending team sessions when required [OK]
- The team found a time slot to meet outside of the class [OK]
  - It is a challenge to accommodate everyone's hectic schedule. Some of us work over the weekend, lecture during others' free time, and so on. We have decided to resolve this with flexible online meetings and mandatory email checkpoints every end of Friday.
- Team ready and able to use the chosen back and front-end frameworks, and those who need to learn are working on learning and practicing [DONE]
- The team reviewed class slides on requirements and used cases before drafting Milestone 1 [OK]
- The team reviewed non-functional requirements from the “How to start...” document and developed Milestone 1 consistently. [DONE]
- The team lead checked the Milestone 1 document for quality, completeness, formatting, and compliance with instructions before the submission [DONE]
- The team lead ensured that all team members read the final M1 and agreed/understood it before submission [DONE]
- The team shared and discussed the experience with genAI tools among themselves [DONE]

- Github is organized as discussed in class (e.g., master branch, development branch, folder for milestone documents, etc.) **[OK]**

**SW Engineering CSC 648-848 Fall 2024**

**StudyGator**

Team 08

Team Lead: Bryan Lee ([blee37@sfsu.edu](mailto: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. Executive Summary**

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**
  - Unregistered user
    - 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).

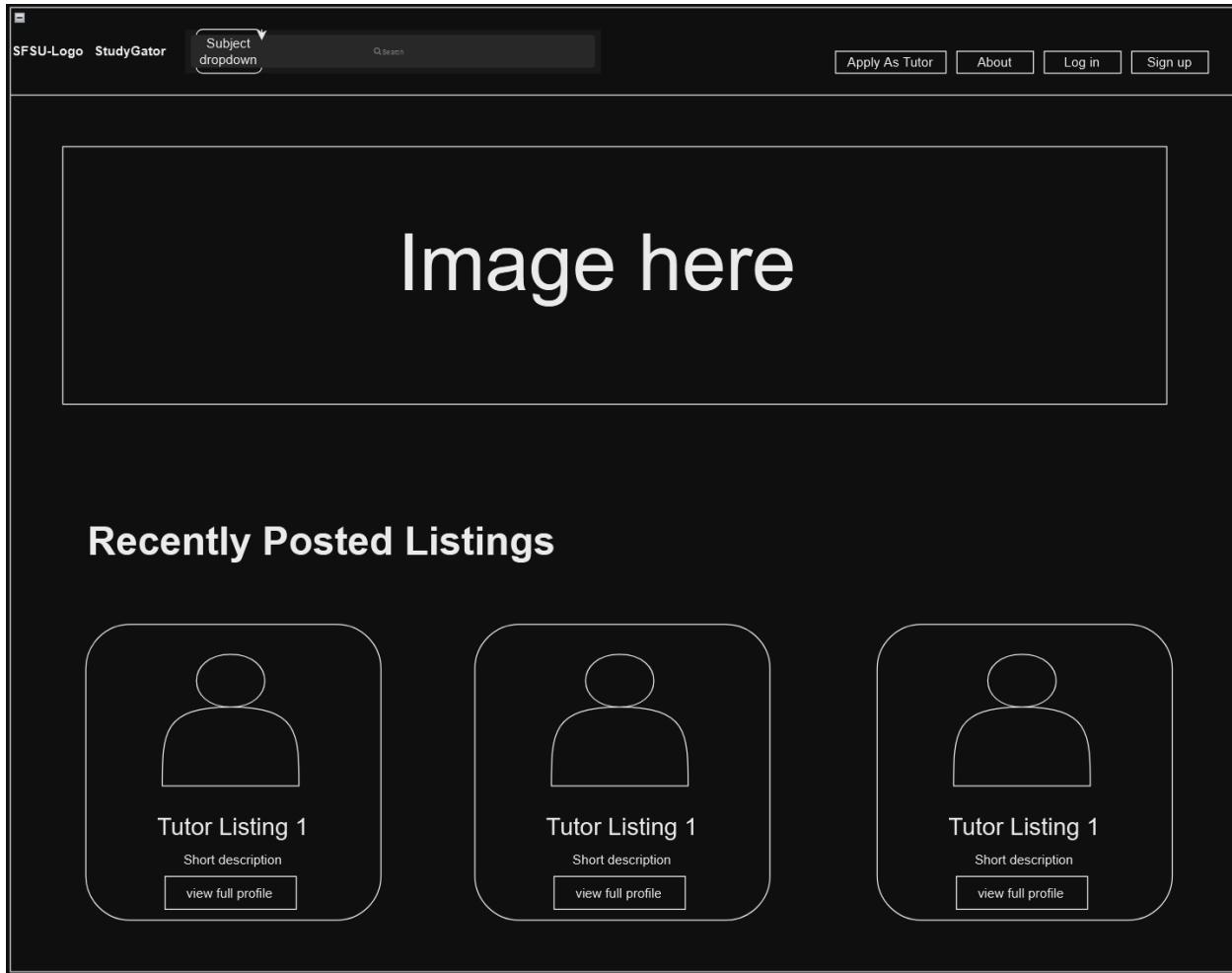
- Unregistered users shall be able to search for courses and tutor descriptions.
- Unregistered users shall be able to filter tutor listings by price.
- Unregistered users shall be able to register an account using an SFSU email.
- Unregistered users shall be able to sign in to an existing account.
- 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 required 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**
  - Unregistered user: No further requirements are needed. All specified requirements are already listed.
  - Registered user
    - 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).
- Admin
  - Admin users shall be able to delete user's ratings.
- **Priority 3**
  - Unregistered user: No further requirements are needed. All specified requirements are already listed.
  - Registered user
    - 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.
  - Admin
    - Admin users shall be able to view website analytics.

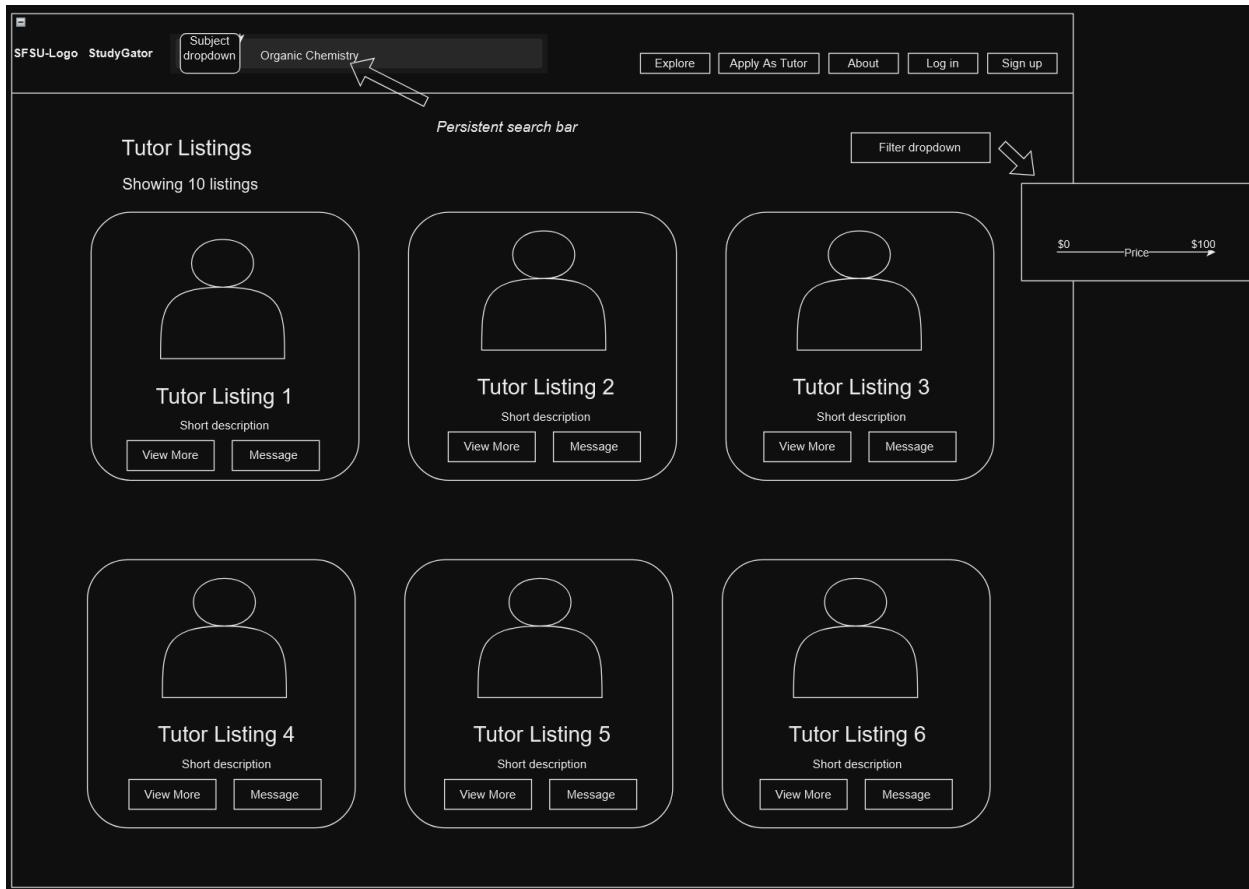
## 4. UI Storyboards

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

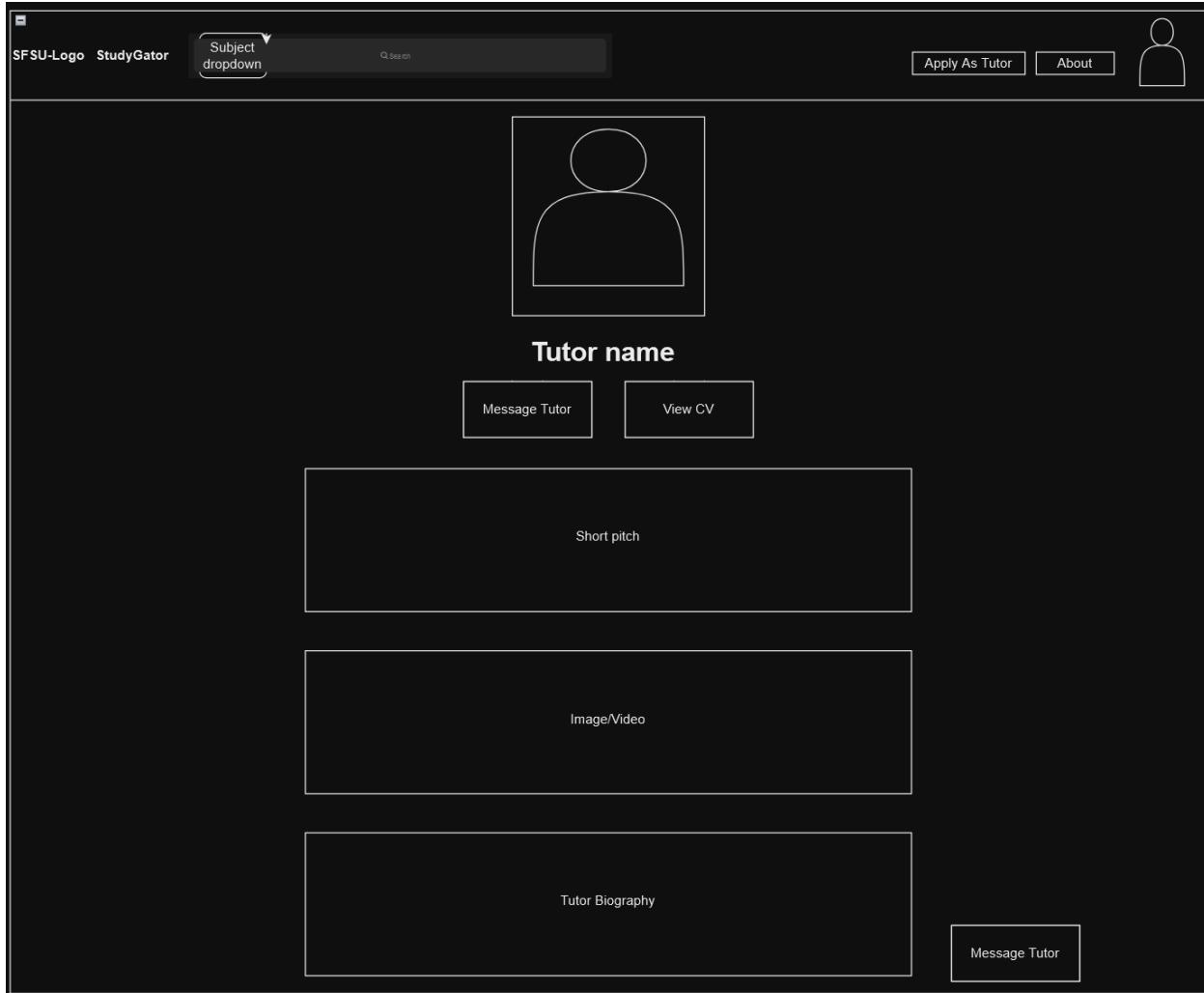
*John is an SFSU student looking for a tutor to help him prepare for his upcoming organic chemistry midterm. When he first visited our site, he found our landing page.*



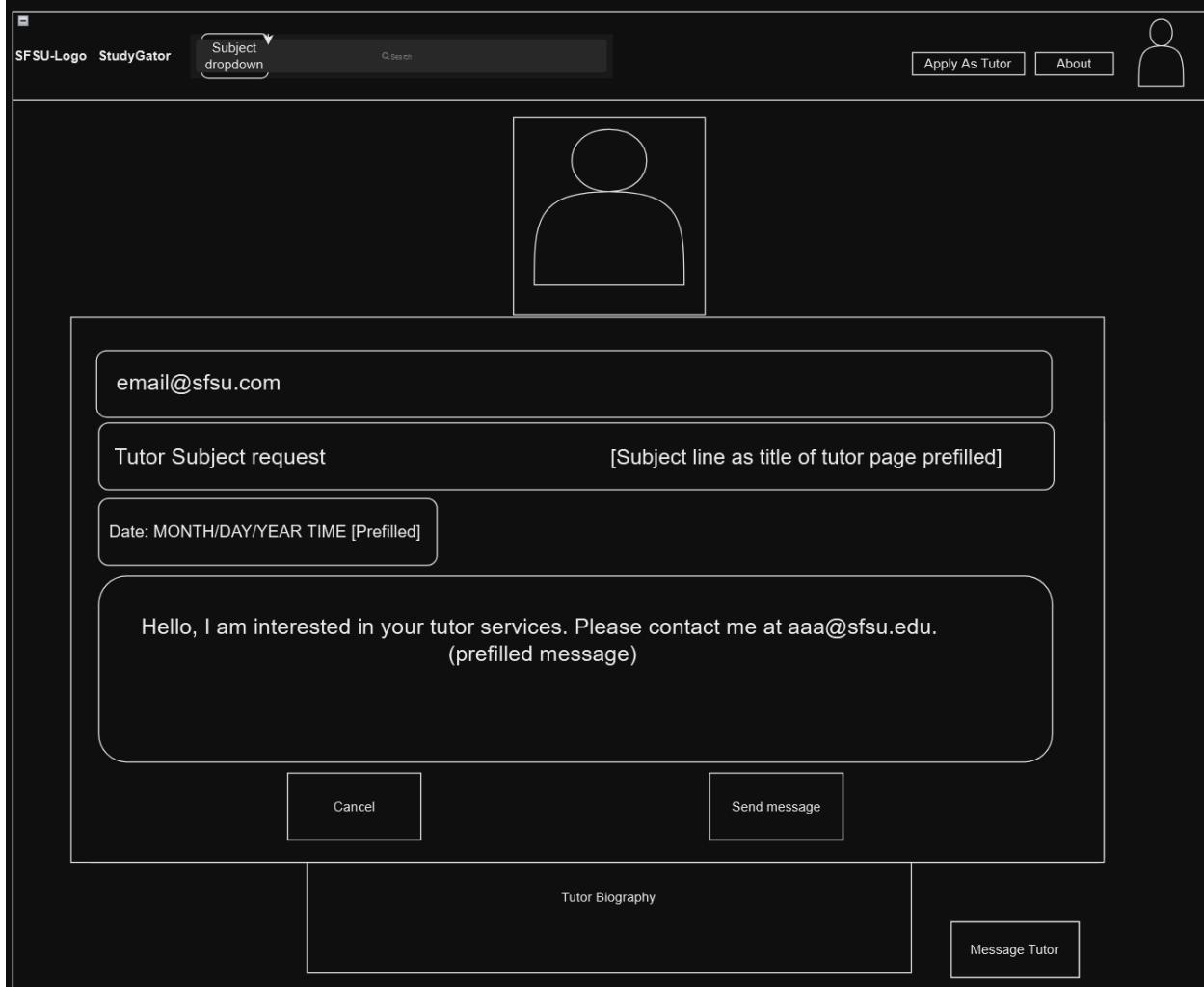
*John immediately searches “organic chemistry” in the search bar, and the application redirects him to the tutor listings page, which shows tutors who specialize in organic chemistry. Here, he can browse and filter the listings using our unique feature, which allows searching by SFSU-related subjects, courses (e.g., Math 100, CSC 648), and prices.*

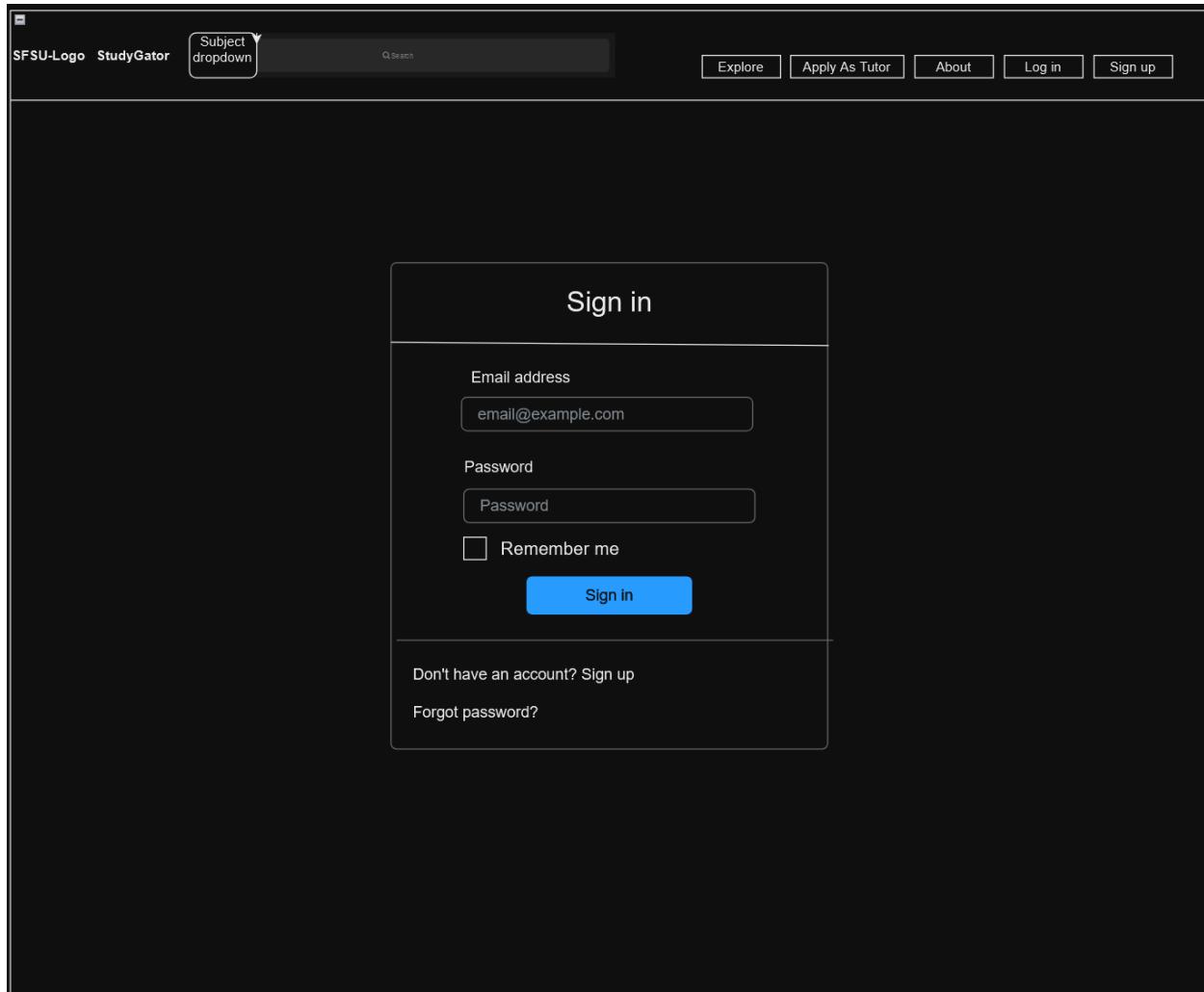


*John clicks on View More to see a tutor listing in more detail. Here, he can see short pitches from the tutor, the tutor's introduction/bio, images/videos, the tutor's resume, and he can message the tutor.*



*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).*

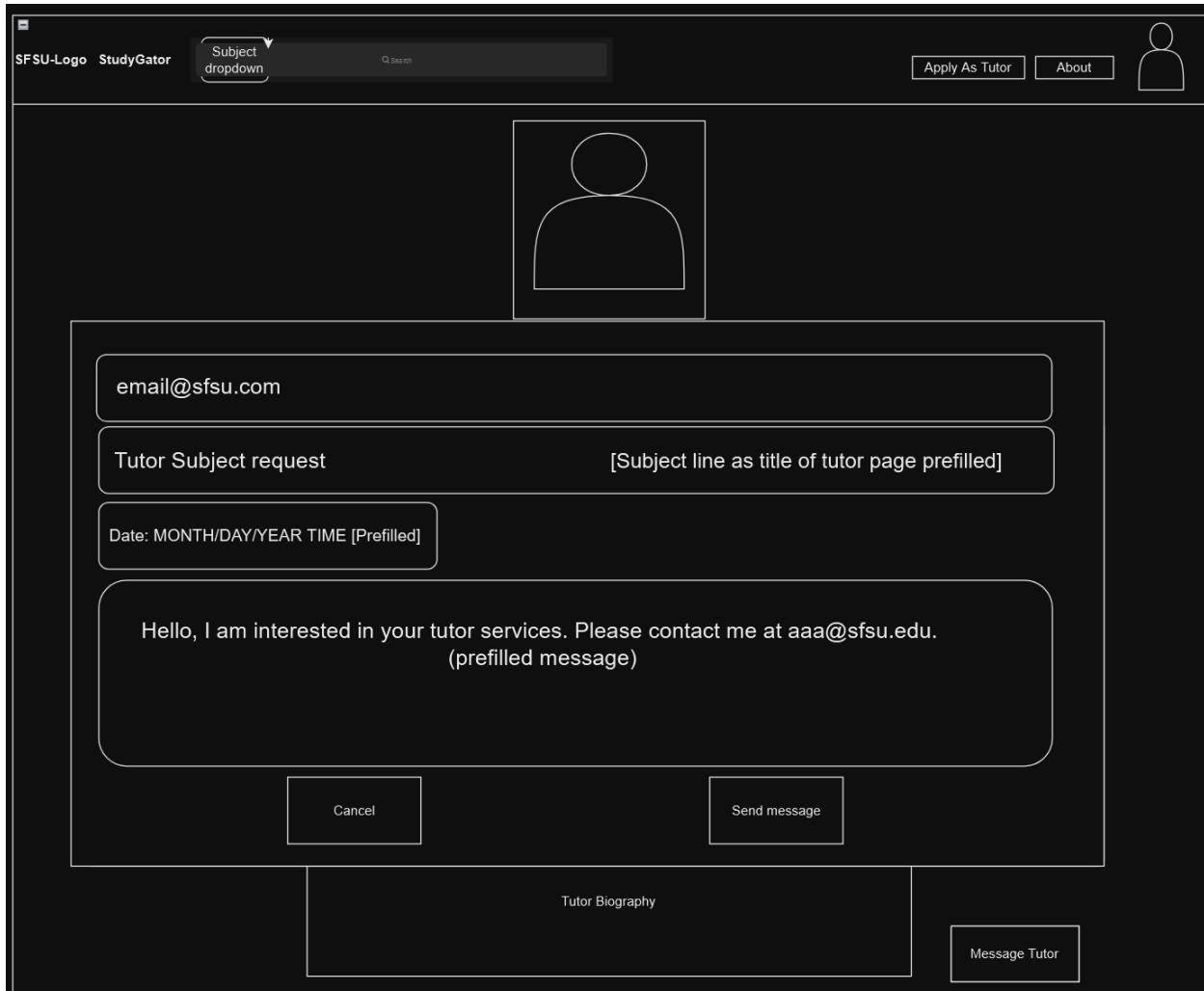




*Since John does not have an account, he clicks Sign up and creates a new account.*

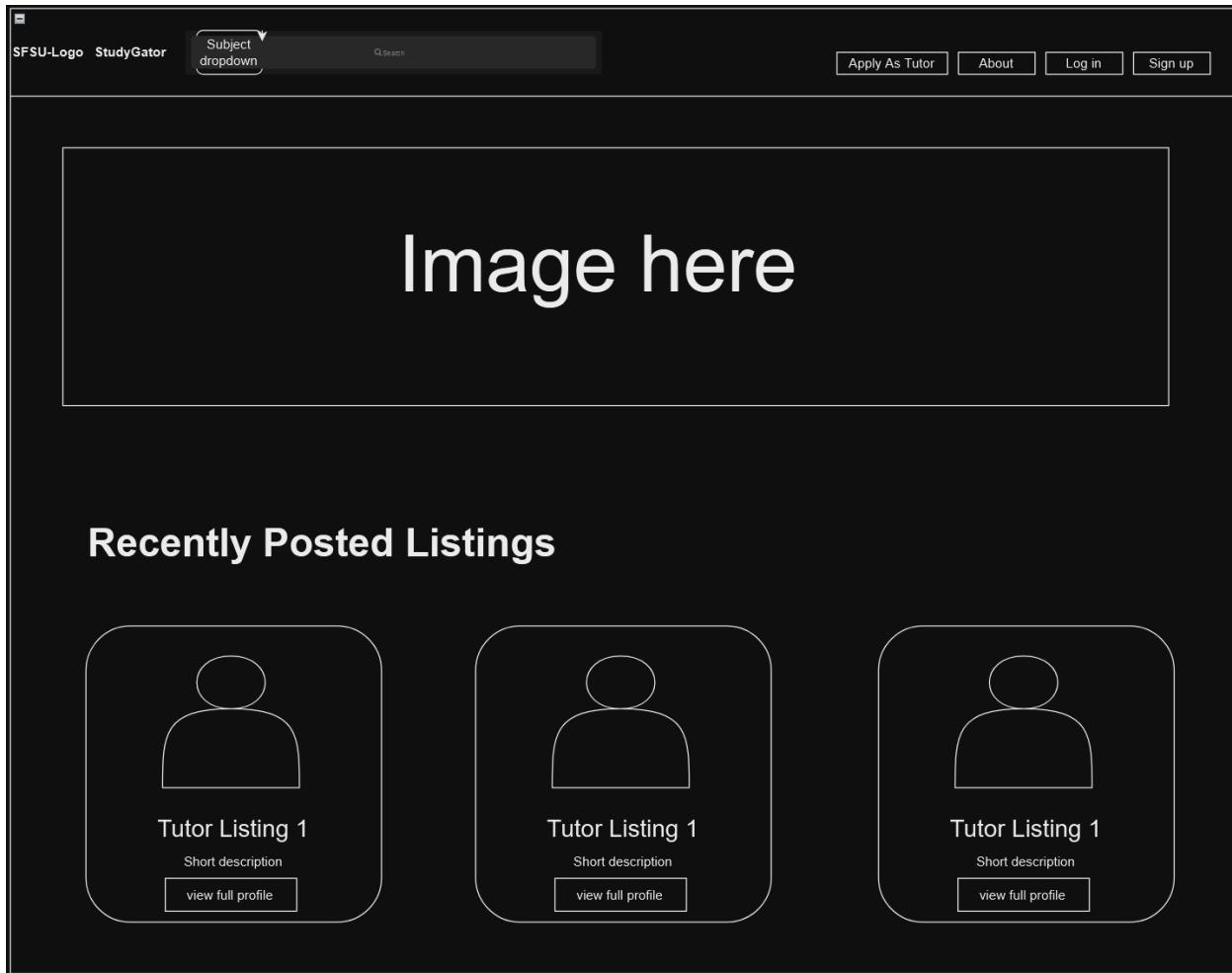
The screenshot shows a dark-themed web application interface. At the top, there is a header bar with the SFSU Logo, the text "StudyGator", a "Subject dropdown" button, a search bar with the placeholder "Search", and several navigation links: "Explore", "Apply As Tutor", "About", "Log in", and "Sign up". Below the header is a large central form titled "Create An Account". The form contains four input fields labeled "Name", "Email address", "Password", and "Confirm Password", each with a corresponding text input box. Below these fields is a checkbox labeled "Agree to our terms and services". A blue "Sign Up" button is positioned below the checkbox. At the bottom of the form, there is a link "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.*

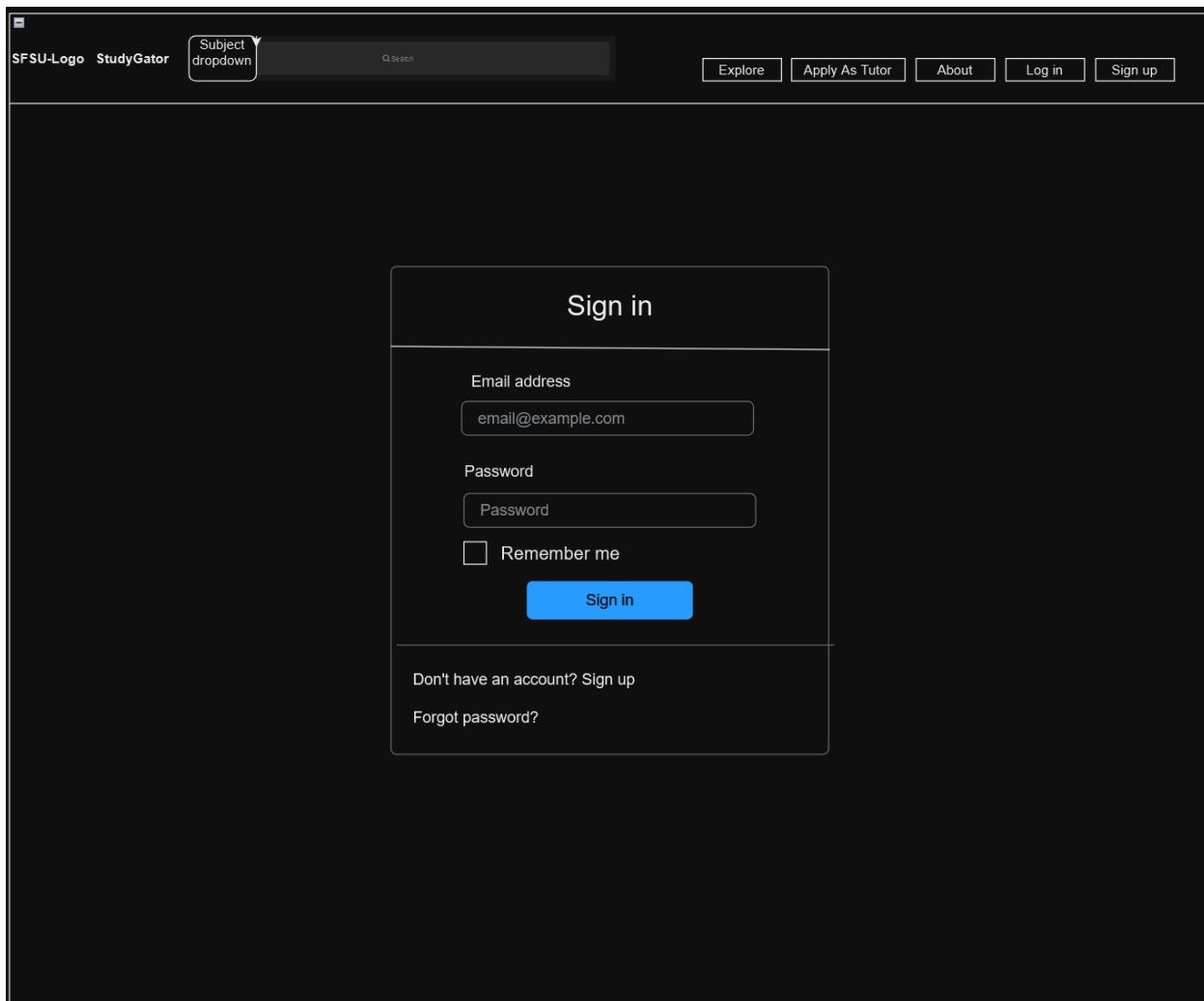


## Use Case 2: Benjamin applies to create a tutor listing.

*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.*



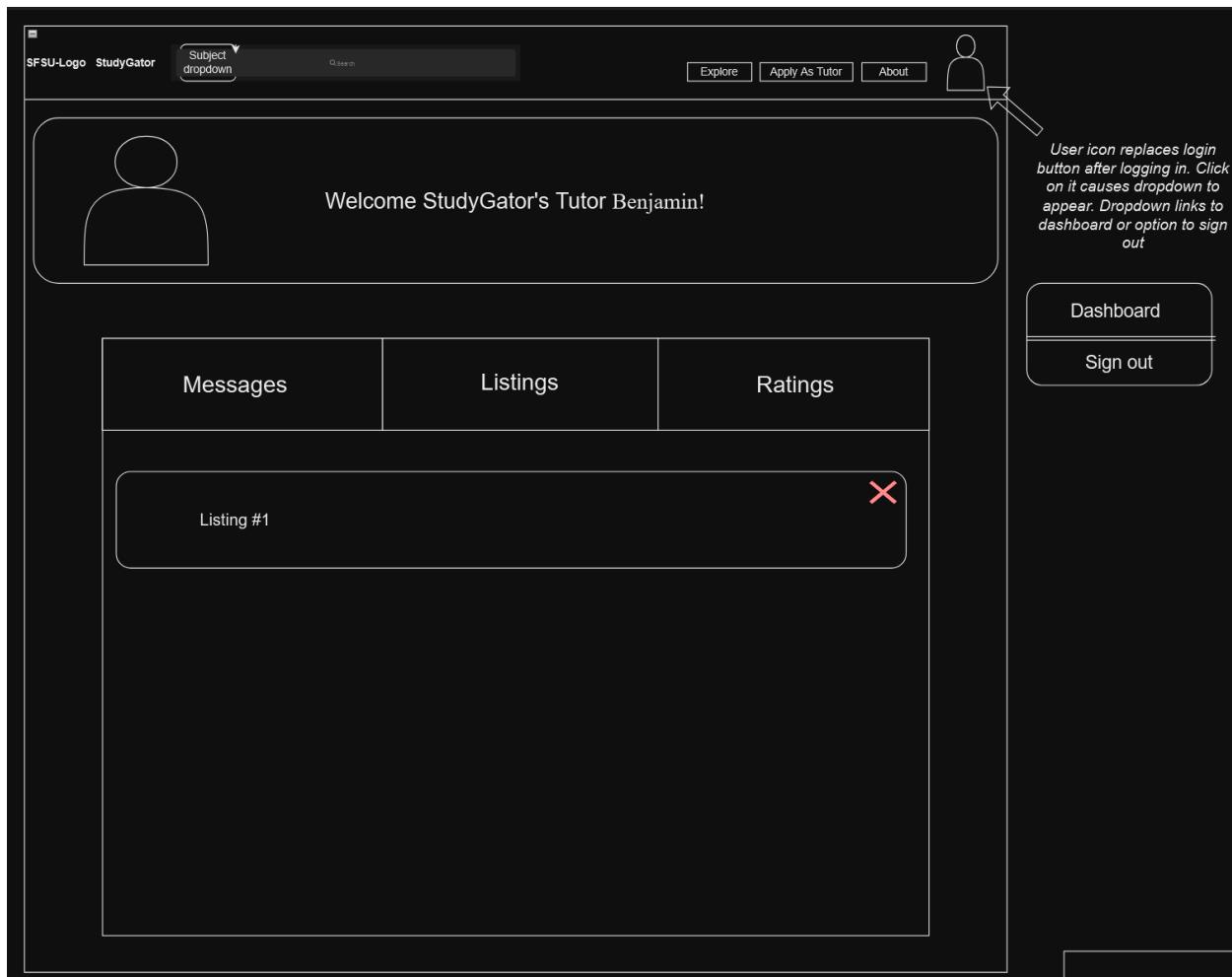
Once he signs in, he is brought back to the previous page he was on (landing page in this use case) and clicks the Apply as Tutor listing button on the navigation bar. The application will redirect him to the listing form, which he can fill out.

The screenshot shows a dark-themed web application interface. At the top, there is a header with the SFSU Logo, the text "StudyGator", a "Subject dropdown" menu, a search bar, and navigation links for "Explore", "Apply As Tutor", and "About". A user profile icon is also present in the top right corner. Below the header, the main content area is titled "Apply As Tutor Form". The form fields include:

- Tutor's Name \* (input field)
- Email address \* (input field containing "email@sfsu.edu")
- Price \$/hr (input field containing "Input price")
- Which Subject are you tutoring? (dropdown menu labeled "Dropdown for subject")
- Upload Your Image (button labeled "Drag or drop image here")
- Upload Your Resume/CV (button labeled "Drag or drop pdf here")
- A large text area for comments with placeholder text: "Provide an introduction or comments about yourself here!"

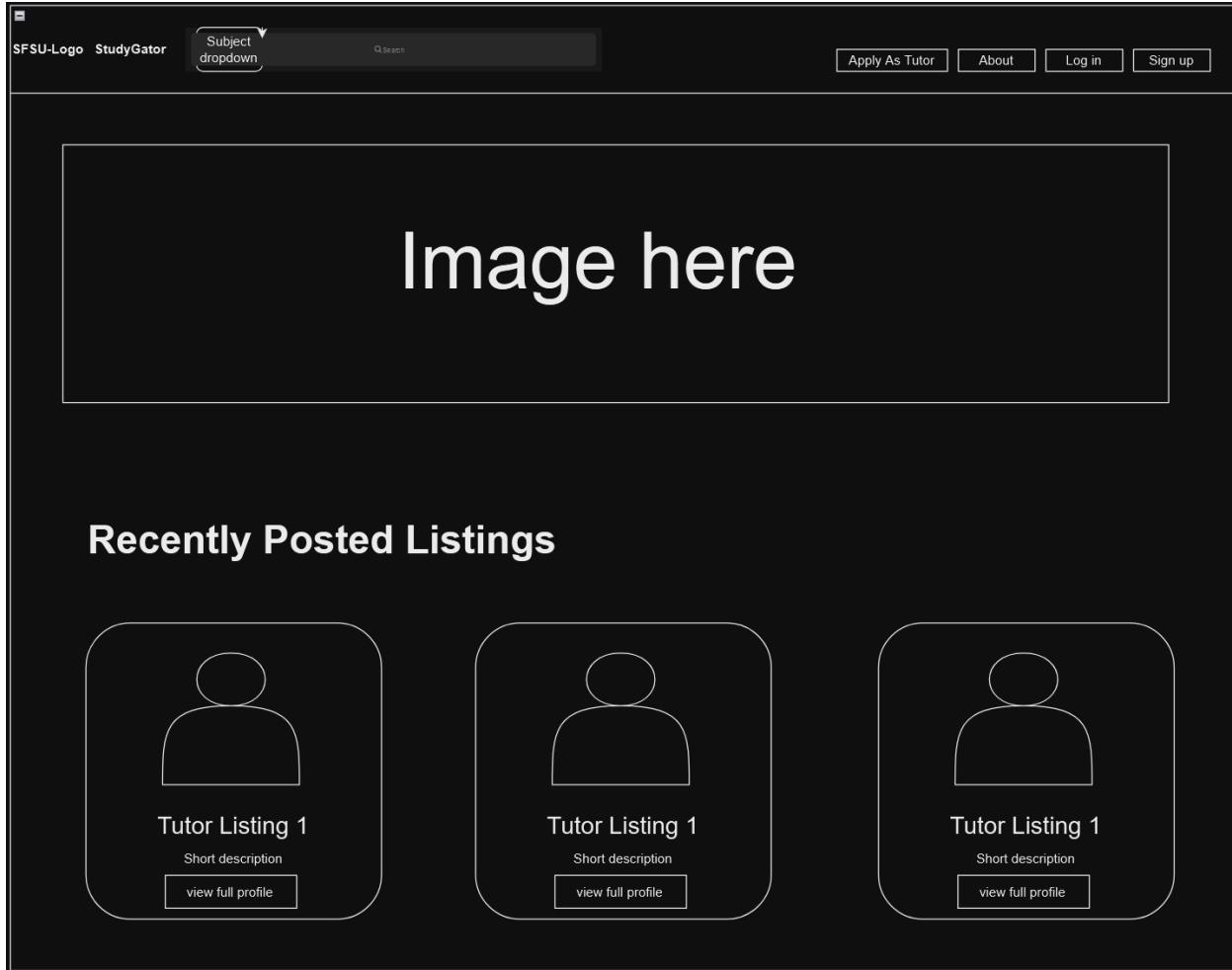
At the bottom of the form, a note states: "Listing may take up to 24 to 48 hours to be approved by an admin before going public." A blue "Submit" button is located at the very bottom center of the form area.

*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.*



### **Use Case 3: Benjamin uses his dashboard.**

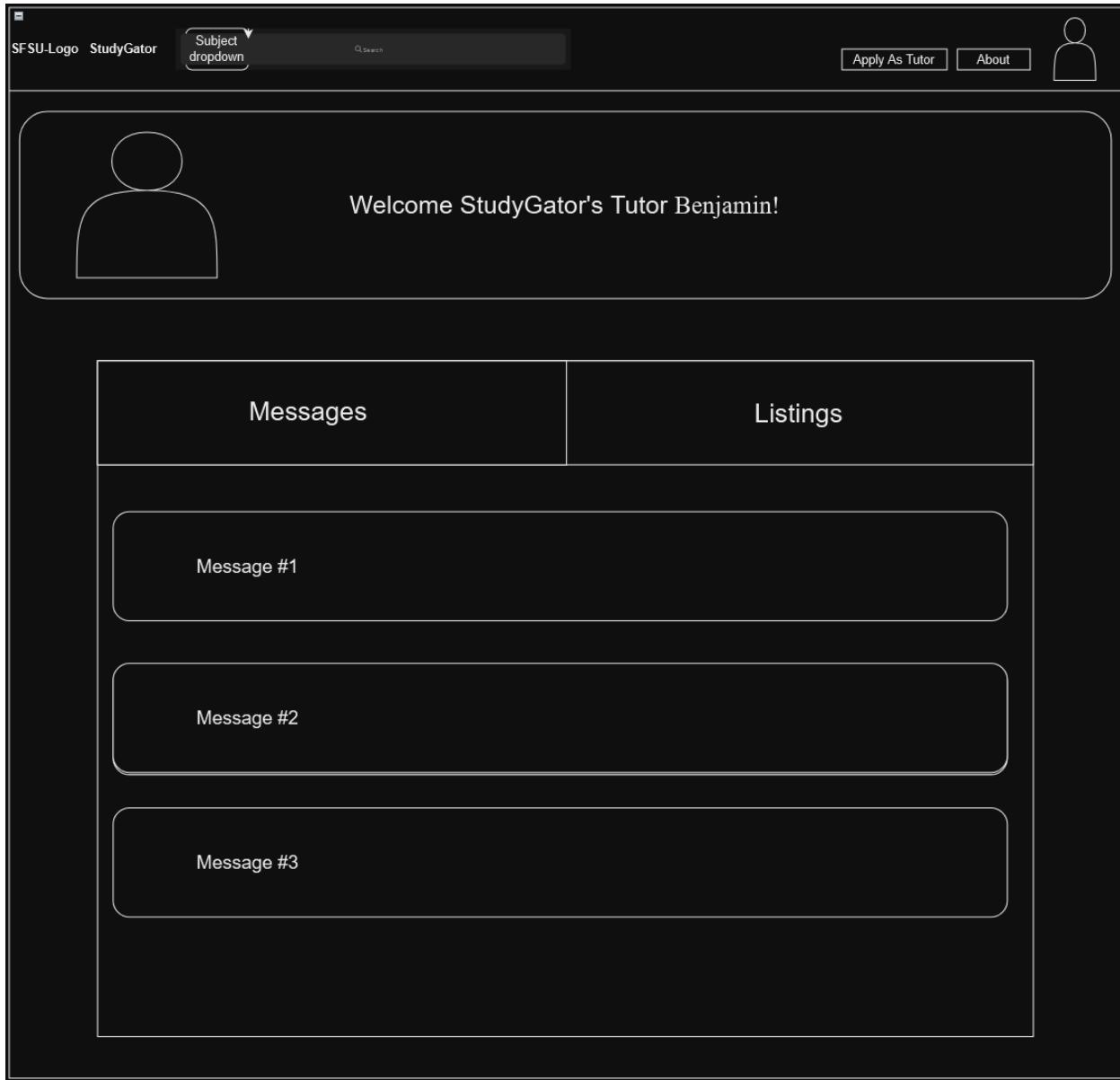
*Benjamin wants to check his new messages and delete a listing on **StudyGator**. He first opens up the site and is greeted with the landing page.*



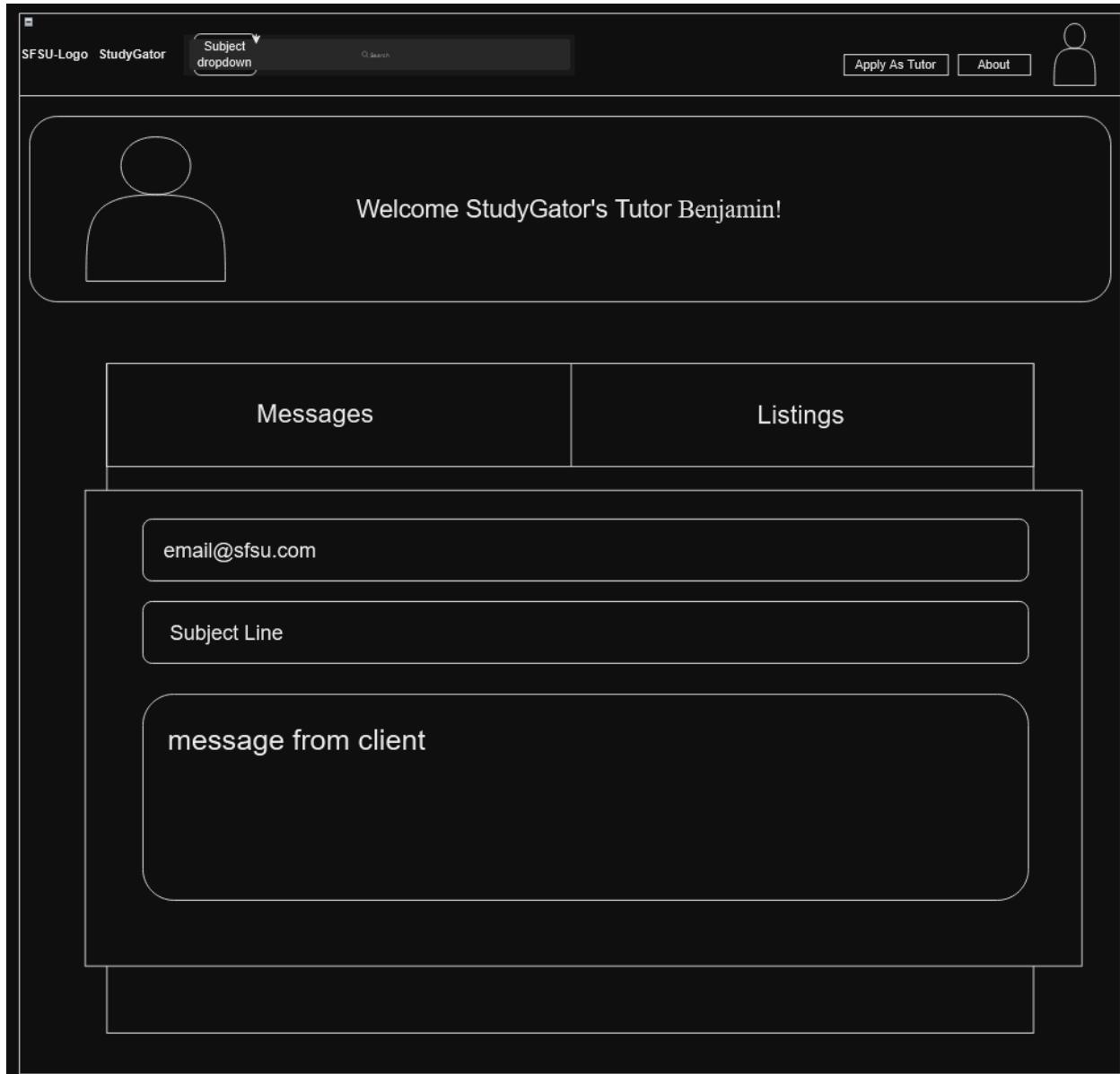
*Benjamin goes to sign in to his existing account.*

The screenshot shows a web browser window with a dark theme. At the top, there is a header bar with the SF SU-Logo, the text "StudyGator", a "Subject dropdown" button, a search bar with the placeholder "Search", and navigation links for "Home", "Explore", "Services", "About", "Log in", and "Sign up". Below the header is a large, centered "Sign in" form. The form contains fields for "Email address" (with the value "email@example.com") and "Password", both enclosed in input boxes. There is also a "Remember me" checkbox and a blue "Sign in" button. At the bottom of the form, there are links for "Don't have an account? Sign up" and "Forgot password?".

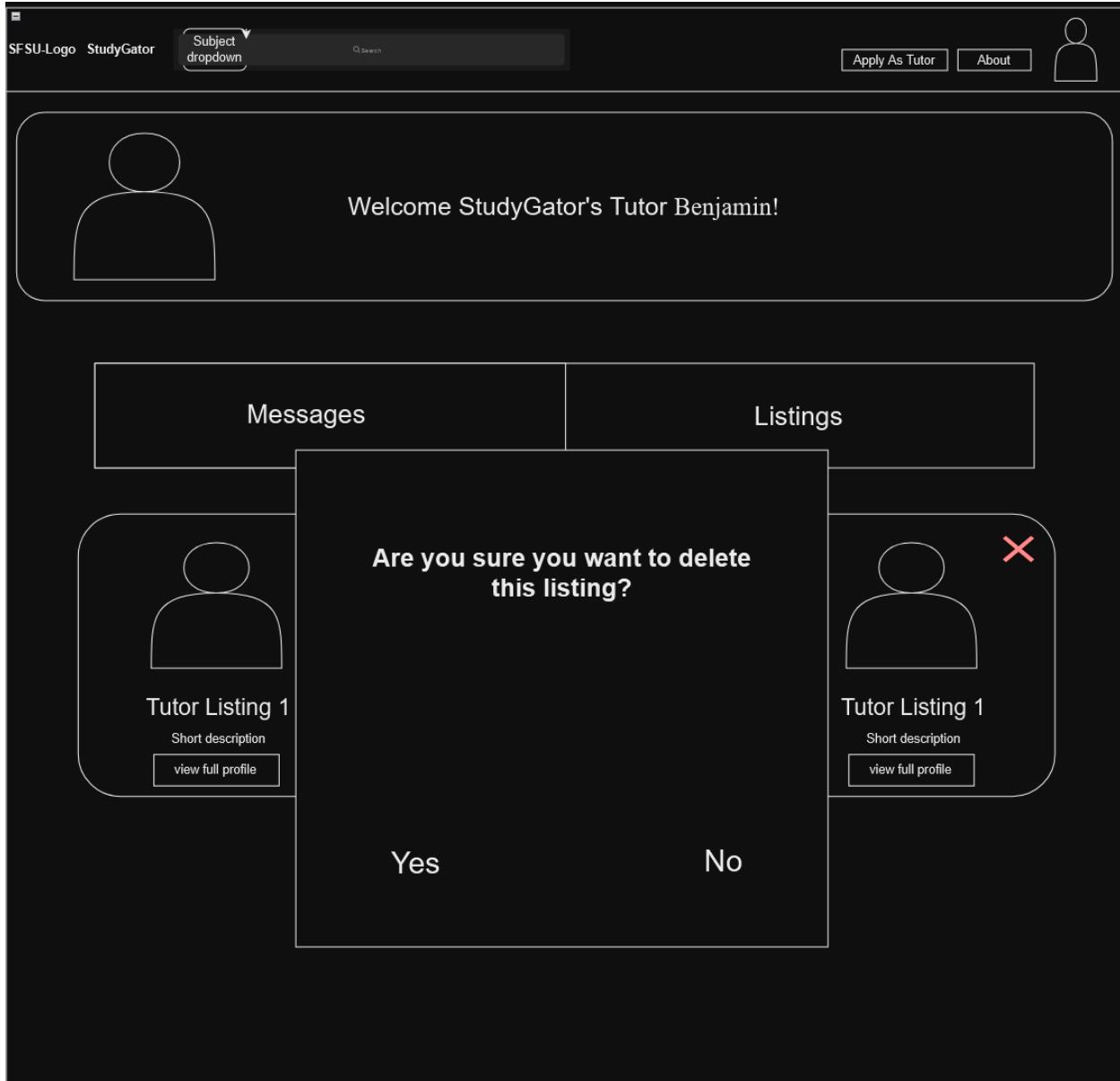
*Benjamin then uses the navigation bar to access his dashboard, where he can view his messages and listings.*



*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. High-level Architecture, Database Organization summary only**

### **Database Organization:**

#### ➤ Users (table)

- id
- name
- email
- password
- date\_created

#### ➤ Tutor Listings (table)

- 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)

#### ➤ Messages (table)

- id
- recipient\_user\_id
- sender\_user\_id
- listing\_id
- date\_created
- content

#### ➤ Ratings (potential priority two table)

- id
- target\_user\_id

- posted\_user\_id
- date\_created
- rating\_value
- comments

### **Media Storage:**

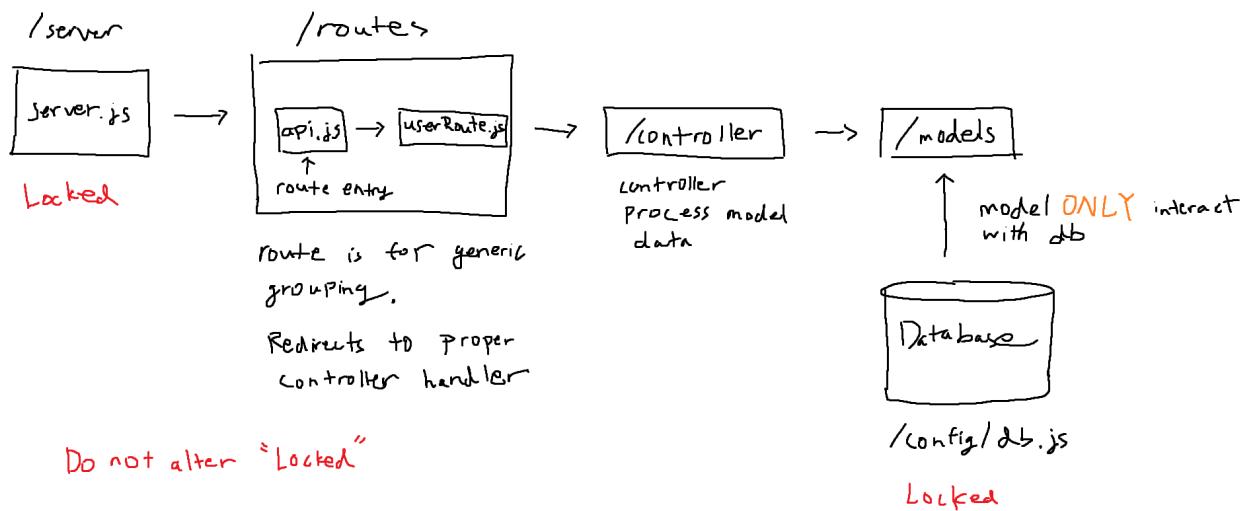
- We will store media content via blobs. Some specific requirements for images are the following extensions: jpg, jpeg, png, and “webp.” For potential priority two, the CV upload will only take PDF. Video uploads will also only take MP4, MOV, or “webm.”

### **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):



## 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. Project Management**

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]**

## **SW Engineering CSC 648-848 Fall 2024**

**StudyGator**

Team 08

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

Front-end lead: Min Ye Thway Khaing

Back-end lead: Nishi Suratia

GitHub master: Kenneth Wen

### **Milestone 3 Meeting Review Summary**

November 13, 2024

Date Submitted	Date Revised
12/18/2024	n/a

## **1. Summary of feedback on UI**

For small bits of here-and-there, the first thing we need to fix is to remove the “explore” or “find a tutor” button on the navbar. We do not want multiple ways to access the tutor listing browsing page; we can simply use the search functionality. Another thing to add to the navbar is a legal disclaimer that states the application is a university project. Moving on to the landing page, we want to have a smaller and single static image. There should be no need for any animation or image rotation carousel, especially the buttons. The recent tutor cards on the landing page should also be slightly more pronounced without the need to scroll down on the page. Additionally, we should add a second message button at the bottom of the “more detail” page about a particular tutor’s listing.

For the tutor listing card, we will need to add a title to the top of the tutor cards (above the image). The title should be customizable by the user and added to the tutor listing’s database table. Another necessary change we need to make to the listing card is to have the message button on the left-hand side of the “more detail” button. Possibly, we can switch the message from an icon to the text “contact.”

One important aspect we must correct is the lazy registration, which only prompts the sign-in when the user clicks “submit” on the tutor application. The lazy registration should also be used for the upcoming message feature. On a side note, the application form’s header should simply be “Apply as tutor” (without the form), and we will need a new input field for the tutor listing card’s title.

The other UI we will need to work with is to add a greeting to the dashboard’s header. The button for switching between the messages and “Manage my tutor listings” (emphasis on changing your to my) should have a darker shade when hovered or selected. Lastly, as an additional precaution, we should keep all the “close” buttons as literal text rather than an “x” icon to maintain consistency and ease of use.

## **2. Summary of feedback on code and architecture**

The only feedback on the code would be the comments. Overall, the code and architecture are good.

## **3. Summary of feedback on GitHub usage**

The feedback we got on our GitHub usage is to phrase our commits a tad bit better. They should be more concise, with a verb followed by a noun, such as “i.e., created resume pop up.” Overall, our GitHub usage is acceptable, but we will have to work on the Git commits.

## **4. Summary of feedback on DB**

n/a. All good.

## **5. Summary of feedback on teamwork and risk management**

Due to an incorporated teammate who failed to complete their task, I, the team lead, will provide an email report on the situation. With our smaller group size of four people, teamwork is critical; however, three of us are feeling the burdensome liability of said incorporative teammate. More details are provided in the email.

## **6. Architecture review**

All good.

## **7. Agreed upon P1 list of features for final delivery**

- Unregistered user
  - Unregistered users shall be able to browse tutor listings.
  - Unregistered users shall be able to view tutor listing details (description, images, more information, title, sales pitch, and subject).
  - Unregistered users shall be able to search for a tutor by the tutor's name, the descriptions, the sales pitch, the subject, and the listing title.
  - 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 (lazy registration).
  - Unregistered users shall be able to fill out the tutor application form; however, when they attempt to submit it, the application prompts them to sign in or create an account (lazy registration).
- 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 and submit a listing application.
  - Registered users shall be able to view their message(s) in the dashboard.
  - 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 required 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.

No other additional comments, issues, or checkpoints are required.

**SW Engineering CSC 648/848 Fall 2024**

Team 08

**StudyGator**

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

Front-end lead: Min Ye Thway Khaing

Back-end lead: Nishi Suratia

GitHub master: Kenneth Wen

**Milestone 4**

November 24, 2024

Date Submitted	Date Revised
12/18/2024	n/a

## **1. Product Summary**

**Name of product:** StudyGator

**Product Description:**

**StudyGator** is a tutoring platform allowing SFSU students to tutor and receive tutoring from other SFSU students. While there are numerous other tutoring websites to choose from, StudyGator stands out by connecting students with tutors who are also SFSU students, offering firsthand knowledge of the same course materials and challenges. Users can easily find tutors by searching subjects and keywords or filtering by price to fit their budget. For those looking to offer tutoring services, **StudyGator** provides a seamless process to create and promote listings. **StudyGator** ensures that academic help is accessible and is always just a click away to empower SFSU students to focus on what matters most—their educational success.

**The itemized list of ALL primary committed functions:**

- Unregistered user
  1. Unregistered users shall be able to browse tutor listings.
  2. Unregistered users shall be able to view more details about the tutor listing.
  3. Unregistered users shall be able to search for a tutor.
  4. Unregistered users shall be able to filter tutor listings by price.
  5. Unregistered users shall be able to register an account using an SFSU email.
  6. Unregistered users shall be able to sign in to an existing account.
  7. Unregistered users shall be able to write a message but must sign in to send it.
  8. Unregistered users shall be able to fill out the tutor application form but must sign in to submit.
- Registered user
  1. Registered users shall inherit all previous privileges.
  2. Registered users shall be able to message tutors about their services.
  3. Registered users shall be able to submit a tutor application to create a new listing.
  4. Registered users shall be able to view their message(s) in the dashboard.
  5. Registered users shall be able to delete their message(s) in the dashboard.

6. Registered users shall be able to view their listing(s) in the dashboard.
  7. Registered users shall be able to delete their listing(s) in the dashboard.
- **Admin**
    1. Admin users shall inherit all previous privileges.
    2. Admin users shall be required to verify all tutor listing applications before they are public.
    3. Admin users shall be able to remove users or listings from the website.
    4. Admin users shall be able to view and manage all user data.

**URL to product:** <https://eclipseskura.online/>

## **2. Usability test plan for the selected function**

- 1) **Test objectives:** This usability test evaluates our application's search functionality, focusing on its effectiveness in finding relevant tutor listings and measuring user satisfaction. The evaluation will gather feedback on the relevance of search results to user queries and chosen subjects, as the search feature is the backbone of our product's advertised services.
- 2) **Test background and setup:**
  - a) **System setup and starting point:** The testers will need an electronic device with internet access, such as a computer, laptop, tablet, or mobile device. They will start on the website's landing page and access the search bar via the navigation bar at the top.
  - b) **Hardware Requirements:** Testers will need a working electronic device with access to the internet.
  - c) **Intended User:** Our product's intended users are SFSU students, whether they are seeking a tutor for a specific course or looking to offer their tutoring services.
  - d) **URL:** <https://eclipseskura.online/>
  - e) **Test Environment:** The usability test can occur in any location with internet access. Ideally, no monitoring or training is necessary before testing. The usability

test simulates the role of a stressed student (who has some technological skills) with limited patience and who is a new user of the application.

**3) Usability Task description:** The specific instructions given to the tester prior to the testing and filling out our Likert survey will be as follows: Your task is to play the role of a stressed student with upcoming exams. Given a short list (about three items to not drain the tester's patience) of specific tutors and courses (related to some tutor, of course) to search for, you will open the website and then complete the following:

- Search for a tutor by the course they teach.
- Locate the desired tutor from the search results.
- Modify the query if needed to find the desired tutor.
- Repeat the process for the remaining items on the list.

**4) Plan for Evaluation of Effectiveness:** We plan to measure effectiveness by having the tester fill out a brief questionnaire after each search. The questionnaire is not part of the Likert scale and serves as a means to consolidate a more in-depth report or document the steps to recreate any issues. The questionnaire will ask the tester to provide the number of search attempts to find the tutor successfully, what queries they used, the time taken, and any frustrations, issues, or possible comments. A lower number of search queries indicates a more streamlined search process. Additionally, a success rate metric will then classify the percentage of successful searches within the predefined three queries and quantify effectiveness. Any patterns in failed searches or searches that require substantial effort could then easily be traced and identified for resolution.

**5) Plan for Evaluation of Efficiency:** We will measure efficiency by evaluating the time and effort testers spend completing each search task. As specified in the questionnaire above, testers will report how long it takes them to locate a specific tutor and the queries they use. We will document instances where testers need to modify their searches multiple times, as these may indicate inefficiencies or usability issues. To provide a comprehensive evaluation, we will analyze task completion times and calculate averages to determine the overall speed of navigating and using the search function. These data will help identify specific areas for improvement, such as query responsiveness and result relevance.

**6) Plan for Evaluation of User Satisfaction (Likert scale):** We will provide this at the very end after all the tests. Example:

Statement	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I was able to quickly find the tutor I was looking for using the search function.	<input type="checkbox"/>				
I was satisfied with the amount of information available about the tutor.	<input type="checkbox"/>				
I found the search experience user-friendly.	<input type="checkbox"/>				

There will also be a section for any additional comments that the tester has.

### **3. QA test plan and QA testing**

- 1) Test objectives:** The objective is to ensure our application's search functionality functions as expected, focusing on its effectiveness in finding the relevant tutor listings.
- 2) Hardware and Software setup:**
  - a) Hardware:** An electronic device with internet access, like a laptop or desktop.
  - b) Software:** Google Chrome and Microsoft Edge.
  - c) URL:** <https://eclipsesakura.online/>
- 3) Feature to test:** The primary feature under test is the **search bar functionality**, which allows users to find tutors based on input queries.
- 4) QA Test plan:**

Test #	Test Title	Test Description	Test Input	Expected Correct Output	Test results (Chrome/Microsoft Edge)
1	Search with Valid Query	Verify the search returns relevant results for a valid query.	Query: "Math 100"	Relevant tutor profiles for "Math 100" are displayed within 2-3 seconds.	Chrome: Pass Edge: Pass
2	Search for Non-Existent Subject	Verify the fallback message when no results are found.	Query: "Astrology"	Message: "We couldn't find your desired tutor listing, but here are some tutors that might interest you."	Chrome: Pass Edge: Pass
3	Search empty query	Verify the ability to see all listings when searching with an empty query.	Query: N/A	Displays 15 tutor listings (All tutor listings are currently in the database).	Chrome: Pass Edge: Pass

### Google Chrome and Microsoft Edge test results:

Test #	Test Title	Google chrome result	Microsoft Edge result	Notes
1	Search with a valid query	PASS	PASS	Relevant tutor profiles (e.g., for "Math 100 Tutor") displayed within 2-3 seconds.
2	Search for nonexistent query	PASS	PASS	Correctly displayed the error message when the search results showed no listings found. This message appeared when searching for a subject not listed (e.g., "Astrology").
3	Search empty query	PASS	PASS	Searching with an empty query displays all tutor listings.

#### Test 1: Search with a valid query

Description: Verify the search returns relevant results based on the input in the search bar.

Input: Enter "Math 100"

Expected Output: 1 result containing "Math 100" in their tutor listing details.

Results (both Chrome & Edge): 1 result containing “Math 100” in their tutor listing details.

### **Test 2: Search with a query with no results**

Description: Verify the search returns relevant results based on the input in the search bar.

Input: Enter “Astrology”.

Expected Output: We couldn't find your desired tutor listing, but here are some tutors that might interest you. Please refine your search or select a subject for a more precise result.

Results (both Chrome & Edge): Displayed “We couldn't find your desired tutor listing, but here are some tutors that might interest you. Please refine your search or select a subject for a more precise result.” and also displayed other tutors the user might be interested in.

### **Test 3: Search with an empty query**

Description: Pressing the search button without inputting anything in the search bar.

Input: Nothing. Just press the search button.

Expected Output: Displays 15 tutor listings.

Results (both Chrome & Edge): Displays 15 tutor listings (All tutor listings currently in the database).

#### 4. Peer Code Review:

##### Milestone 4 Code Review



Kenneth Wen

To: Nishi Rushirajsinh Suratia



Sun 12/8/2024 3:37 AM

Hello Nishi,

Can you review my code in SearchPage.jsx? The function of the search page is to show the tutor listings by generating tutor cards after retrieving the data from the database.

Link: <https://github.com/CSC-648-SFSU/csc648-fa24-03-team08/blob/main/application/client/src/SearchPage/SearchPage.jsx>

Thanks,  
Kenneth



Nishi Rushirajsinh Suratia

To: Kenneth Wen



Sun 12/8/2024 3:54 AM

Hello kenneth, sure.

Here is my feedback for your SearchPage.jsx that shows tutor listings:

- 1) Emphasis on reviewing comments, variable names, and alignment with established naming conventions is correct such that it ensures code readability.
- 2) Highlighting the importance of an intuitive user experience, especially for search and filter features is necessary ensuring actions like price filtering are simple for users so the placement of search functionality in the page aligns well with this.
- 3) Error messages and feedback mechanisms during server loads can be made more visible to the users.
- 4) Confirms password encryption, input sanitization, and database protection to secure user data and prevent vulnerabilities.
- 5) When specific results are unavailable, random tutor profiles are presented as a helpful alternative, encouraging users to explore.

Further detailed comments have been included in the github review ,Kindly check that.  
Thank you.

## Github code review for SearchPage.jsx:

```
// State variables for tutor listings- naming conventions align with data-Nishi
const [listings, setListings] = useState([]);
const [filteredListings, setFilteredListings] = useState([]);
const [isFilterApplied, setIsFilterApplied] = useState(false);
const [randomListing, setRandomListing] = useState(false);
const [loading, setLoading] = useState(true);
const [serverError, setServerError] = useState(false);

// State variables for filter drop down
const [minPrice, setMinPrice] = useState(0);
const [maxPrice, setMaxPrice] = useState(200);
const [isPriceDropdownOpen, setIsPriceDropdownOpen] = useState(false);
```

```
useEffect(() => {
  ReactGA.send({ hitType: "pageview", page: window.location.pathname, title: "Search page" });
}, []);

return (
  <div
    className="top-0 flex items-center justify-center sm:min-h-screen bg-gray-100 bg-fixed relative"
    //Background image adds a good touch although test it with high contrast settings to confirm accessibility.-Nishi
    style={{
      backgroundImage: "url('/GatorSearchBackground.png')",
      backgroundRepeat: "no-repeat",
      backgroundSize: "cover",
    }}
  >

  {/* Server error warning */}
  {serverError &&
```

Code 55% faster with GitHub Copilot

Spaces 4 No wrap

```
144
145
146          >
147            Price
148            <svg className="w-2.5 h-2.5 ms-3" aria-hidden="true" xmlns="http://www.w3.org/2000/svg" fill="none" viewBox="0 0 10 6">
149              <path stroke="currentColor" strokeLinecap="round" strokeLinejoin="round" strokeWidth="2" d="M1 1 4 4 4-4" />
150            </svg>
151          </button>
152        {isPriceDropdownOpen && (
153          <div className="absolute z-10 border bg-white divide-y divide-gray-200 rounded-lg shadow-md w-60 mt-1 p-4 right-0">
154            <label className="text-sm font-semibold mb-2">Price Range:</label>
155            <div className="flex flex-col py-3">
156              <input
157                type="range"
158                min="0"
159                max="200"
160                value={minPrice}
161                onChange={(e) => setMinPrice(Math.min(Number(e.target.value), maxPrice))}
162                className="slider"
163              />
164              <input
165                type="range"
166                min="0"
167                max="200"
168                value={maxPrice}
169                onChange={(e) => setMaxPrice(Math.max(Number(e.target.value), minPrice))}
170                className="slider"
171              />
172              <div className="flex justify-between text-sm text-gray-700 mt-2">
173                <span className="w-10 text-right">${minPrice}</span>
174                <span>${maxPrice}</span>
175              </div>
176            </div>
177          </div>
178        )}
```

## **5. Self-check on best practices for security**

Yes, we have more than half a page of security self-check, but all is considered and implemented.

Asset to be protected	Types of possible/expected attacks	Consequence of security breach	Our Strategy to Mitigate/Protect the Asset
User Passwords	Brute force attacks, data breaches, unauthorized access	Exposure of sensitive user accounts, unauthorized logins	Encrypt passwords in the database using a secure hashing algorithm like bcrypt. Server-sided verifications.
User Personal Data	Data breaches, phishing attacks	Leakage of private user data, identity theft	Ensure HTTPS is used for data transfer. Implement access control for sensitive user data at both the backend and database layers. Encrypt cookies using JWT.
Tutor Images/Resumes	Unauthorized downloads, tampering with files	Misuse of personal branding material, spreading false data	Validate uploaded files for type and size before saving them. Use signed URLs for accessing sensitive documents.
Search Bar Input	SQL Injection, XSS (Cross-Site Scripting)	Database Corruption, malicious script execution	Validate search bar input to accept only alphanumeric characters of up to 40 characters. Use parameterized queries to prevent SQL injection. Escape special characters to avoid XSS.
Registration Email	Spam accounts, invalid email submissions	System clutter, reduced credibility, inability to contact	Using a regex pattern, validate email addresses by including "sfsu.edu" at the end of the registration. Reject invalid emails before processing.
Acceptance of Terms	Users bypassing terms, lack of accountability	Legal vulnerabilities, user disputes	Ensure the acceptance of the terms checkbox is mandatory before registration.

## **6. Self-check of the adherence to original Non-functional specs – performed by team leads.**

1. Application shall be developed, tested, and deployed using tools and servers approved by Class CTO and as agreed in M0 **[DONE]**
2. Application shall be optimized for standard desktop/laptop browsers, e.g., must render correctly on the two latest versions of two major browsers **[DONE]**
3. All or selected application functions shall render well on mobile devices (no native app to be developed) **[DONE]**

4. Posting of tutor information and messaging to tutors shall be limited only to SFSU students **[DONE]**
5. Critical data shall be stored in the database on the team's deployment server. **[DONE]**
6. No more than 50 concurrent users shall be accessing the application at any time **[DONE]**
7. Privacy of users shall be protected **[DONE]**
8. The language used shall be English (no localization needed) **[DONE]**
9. Application shall be very easy to use and intuitive **[DONE]**
10. Application shall follow established architecture patterns **[DONE]**
11. Application code and its repository shall be easy to inspect and maintain **[DONE]**
12. Google Analytics shall be used **[DONE]**
13. No e-mail clients shall be allowed. Interested users can only message sellers via in-site messaging. One round of messaging (from user to seller) is enough for this application **[DONE]**
14. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **[DONE]**
15. Site security: basic best practices shall be applied (as covered in the class) for main data items **[DONE]**
16. Media formats shall be standard as used in the market today **[DONE]**
17. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development and GenAI tools **[DONE]**
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 2024. For Demonstration Only" at the top of the WWW page Nav bar. (This is important so as not to confuse this with a real application). **[DONE]**

## **7. Use of genAI tools like ChatGPT and Copilot (mandatory)**

genAI tool	ChatGPT-4o
Task	Structure/idea on usability evaluation plan for

	effectiveness and efficiency.
Useful rating	Medium

- Prompt: Can you provide a structure/idea on how I can create a usability evaluation plan for effectiveness and efficiency for my tutoring website?



Certainly! Here's how you can structure the evaluation plans for **Effectiveness** and **User Satisfaction (Efficiency)** separately, including Likert scale entries:

#### **Plan for Evaluation of Effectiveness:**

Effectiveness will be assessed by measuring the relevance and accuracy of the search results based on tester input. This includes tracking how easily testers can find the desired tutor listings, how many attempts or query modifications are required, and the overall success rate in locating the correct results. Feedback will be gathered on the alignment between the search results and the intended query, using direct observations during testing.

#### **Plan for Evaluation of User Satisfaction (Efficiency):**

Efficiency will be evaluated by recording the time it takes for testers to locate the desired tutor listings and the number of search modifications needed to refine results. Post-test, testers will fill out a Likert scale questionnaire to assess their satisfaction with the speed, ease of use, and overall experience with the search feature.

#### **Sample Likert Scale Entries:**

1. **How easy was it to find the desired tutor listings using the search feature?**  
(1 - Very difficult, 5 - Very easy)
2. **How satisfied are you with the accuracy of the search results relative to your query?**  
(1 - Very dissatisfied, 5 - Very satisfied)
3. **How efficient was the search process in terms of finding the desired tutor listings without excessive effort or time?**  
(1 - Very inefficient, 5 - Very efficient)

For this milestone, one way we use genAI, particularly ChatGPT, is to generate ideas for the usability test plan. It was helpful in providing insightful ideas but could have been more accurate. However, ChatGpt inspired us to create a separate questionnaire besides the Likert scale to acquire more in-depth data about the tester's number of search attempts to find the tutor successfully, what queries they used, the time taken, and any frustrations, issues, or possible comments. Overall, ChatGpt provided valuable insights but required a brute-force approach to prompting. It would repeatedly give the same idea and offer no new insight besides empty fluff.

genAI tool	ChatGPT-4o
Task	Ideas for QA test planning
Useful rating	Medium

- Prompt:

"Can you create a usability and QA test plan for a website's search functionality that caters to stressed students searching for tutors?"

### QA Test Plan

1. Objective:

- Verify the core functionality, responsiveness, and error-handling of the search feature.

2. Feature Scope:

- Test search bar visibility and accessibility.
- Test handling of valid, partial, and invalid queries.
- Check the system's behavior with unlisted subjects or empty searches.

3. Tests to Include:

- **Visibility Test:** Is the search bar prominently placed and functional?
- **Query Test:** Does the search bar return relevant results for a valid query (e.g., "Math 101 Tutor")?
- **Fallback Test:** When no matches are found (e.g., for "Astrology"), does the system display a helpful message like "We couldn't find your desired tutor listing, but here are some tutors that might interest you"?
- **Modification Test:** Can the user refine their query easily and get more specific results?

4. Testing Tools:

- Two major browsers (e.g., Google Chrome, Mozilla Firefox).
- Devices with varied screen sizes to check responsiveness.

5. Pass/Fail Criteria:

- A test passes if the feature behaves as expected (e.g., relevant results are shown or fallback messages appear).
- Failures are documented with observed issues and potential causes.

We also use generative AI, particularly ChatGPT, to brainstorm ideas for our QA test plan. This tool provided valuable insights that enriched our plan, including suggestions for query testing, fallback testing, modification testing, cross-browser testing, and defining clear pass/fail criteria. While some recommendations, like visibility testing, were redundant, the overall contribution was highly beneficial.

### 3. Product Screen Shots



The screenshot shows the StudyGator homepage. At the top, there's a purple header bar with the text "SFSU Software Engineering Project CSC 648-848, Fall 2024. For Demonstration Only." Below the header is a search bar with the placeholder "Search tutors, subjects, or keywords..." and a magnifying glass icon. To the right of the search bar are buttons for "Apply As Tutor", "About Us", "Log In", and "Sign Up". The main background image is a photograph of a modern university campus with students walking and sitting on the grass.

**Find Your SFSU Tutor at StudyGator!**

Connect with skilled SFSU student tutors dedicated to helping you succeed.  
Receive personalized support tailored to your SFSU courses and academic goals.

#### Recent Tutor Listings



Conquer Organic Chemistry:  
Expert Tutoring for Success



Come learn math with me!



Master Computer Science:  
Expert Tutoring for All Levels



**Mariah Hayden**   **\$99 / hour**  
Chemistry

Struggling with reaction mechanisms or molecular structures? I provide clear, personalized tutoring to demystify organic chemistry and help you excel in your coursework and exams.

[Message](#) [More details →](#)



**Alice Li**   **\$30 / hour**  
Math

Struggling with algebra, calculus, or geometry? I offer tailored tutoring sessions to simplify math concepts, boost your confidence, and help you achieve academic excellence.

[Message](#) [More details →](#)



**Alice Li**   **\$30 / hour**  
Computer Science

Unlock the world of coding, algorithms, and data structures with personalized tutoring. Whether you're a beginner or tackling advanced topics, I'll help you grasp concepts and excel in computer science. I can help with classes such as CSC210, CSC648, and CSC667.

[Message](#) [More details →](#)

[Browse more](#)



**StudyGator**

FOLLOW US

Github  
Discord

LEGAL

Privacy Policy  
Terms & Conditions

LOCATION

HSS Building  
1600 Holloway Ave.  
San Francisco, CA 94132

CSC648-FA24-03-Team08 Demonstration Project

**StudyGator**

All Computer Science

Search Results: 4 items found

Browse Tutor Listings

Price ▾

**Master Computer Science: Expert Tutoring for All Levels**

**Alice Li** \$30 / hour

Computer Science

Unlock the world of coding, algorithms, and data structures with personalized tutoring. Whether you're a beginner or tackling advanced topics, I'll help you grasp concepts and excel in computer science. I can help with classes such as CSC210, CSC648, and CSC667.

**Master Math with Confidence!**

**James Jones** \$90 / hour

Computer Science

Overcome CS challenges with tailored tutoring sessions designed for your unique needs. Whether you're struggling with basics or advanced topics, I'll guide you step-by-step to build confidence and achieve success. Let's make CS easier and more enjoyable!

**Code Mastery Made Simple: Your Path to CS Success!**

**Bryan Lee** \$20 / hour

Computer Science

Struggling with computer science concepts or coding projects? Let me help you master them with ease! As an experienced computer science tutor, I specialize in breaking down complex topics into simple, digestible lessons tailored to your learning style.

**Personalized Computer Science Tutoring**

**Luna Brandt** \$65 / hour

Computer Science

Unlock the world of tech with tailored computer science tutoring! Having completed CSC 415 (Operating Systems), CSC 413 (Software Development), and CSC 648 (Software Engineering), I'm well-equipped to tutor these courses with a focus on practical understanding and success.

Message More details →

Follow Us

Github

Discord

Legal

Privacy Policy

Terms & Conditions

Location

HSS Building

1600 Holloway Ave.

San Francisco, CA 94132

© 2024 StudyGator™. All Rights Reserved.

CSC648-FA24-03-Team08 Demonstration Project

StudyGator All Search tutors, subjects, or keywords... Apply As Tutor About Us Log In Sign Up

**Mariah Hayden**  
Subject Expertise: Chemistry

**Short Pitch**  
Struggling with reaction mechanisms or molecular structures? I provide clear, personalized tutoring to demystify organic chemistry and help you excel in your coursework and exams.

**About Tutor**  
I'm a passionate organic chemistry tutor with a knack for simplifying complex topics like reaction mechanisms, stereochemistry, and spectroscopy. With a tailored approach, I help students build a deep understanding of concepts and develop effective problem-solving skills. Whether you're prepping for exams, tackling challenging homework, or aiming for mastery, I'm here to guide you every step of the way. Let's turn organic chemistry into a subject you can master with confidence!

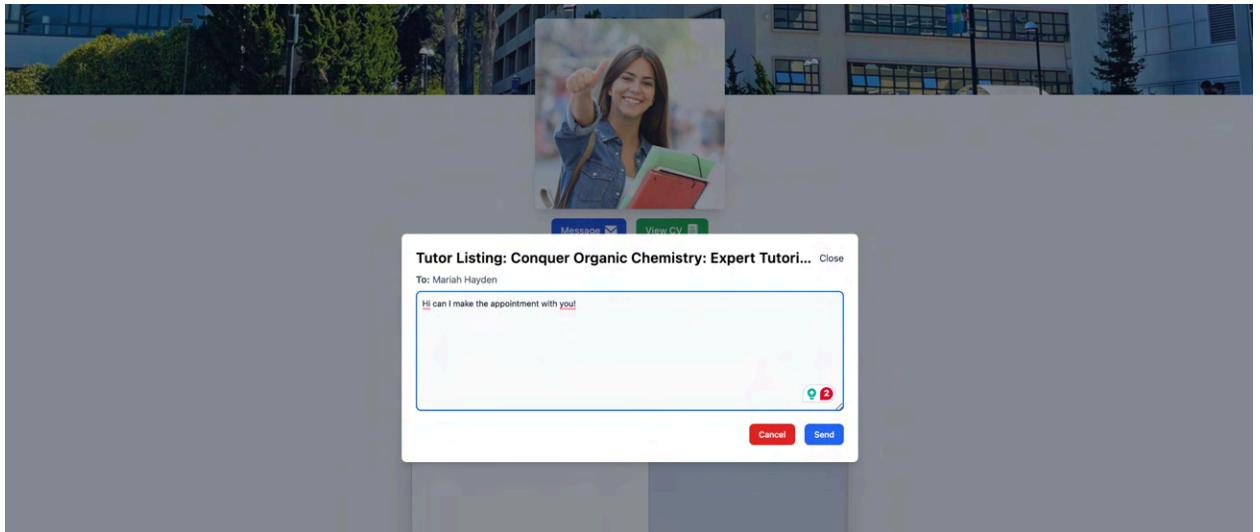
Message Back →

**StudyGator**

FOLLOW US  
Github  
Discord

LEGAL  
Privacy Policy  
Terms & Conditions

LOCATION  
HSS Building  
1600 Holloway Ave.  
San Francisco, CA 94132

A screenshot of a CV viewer. The title of the document is "Mariah Hayden's CV". The CV includes sections for "Professional Summary", "Skills", "Professional Experience", and "About Tutor".

**John Doe**  
1234 Elm Street  
Anytown, USA 12345  
(123) 456-7890  
john.doe@email.com  
[LinkedIn Profile](#) | [GitHub Profile](#)

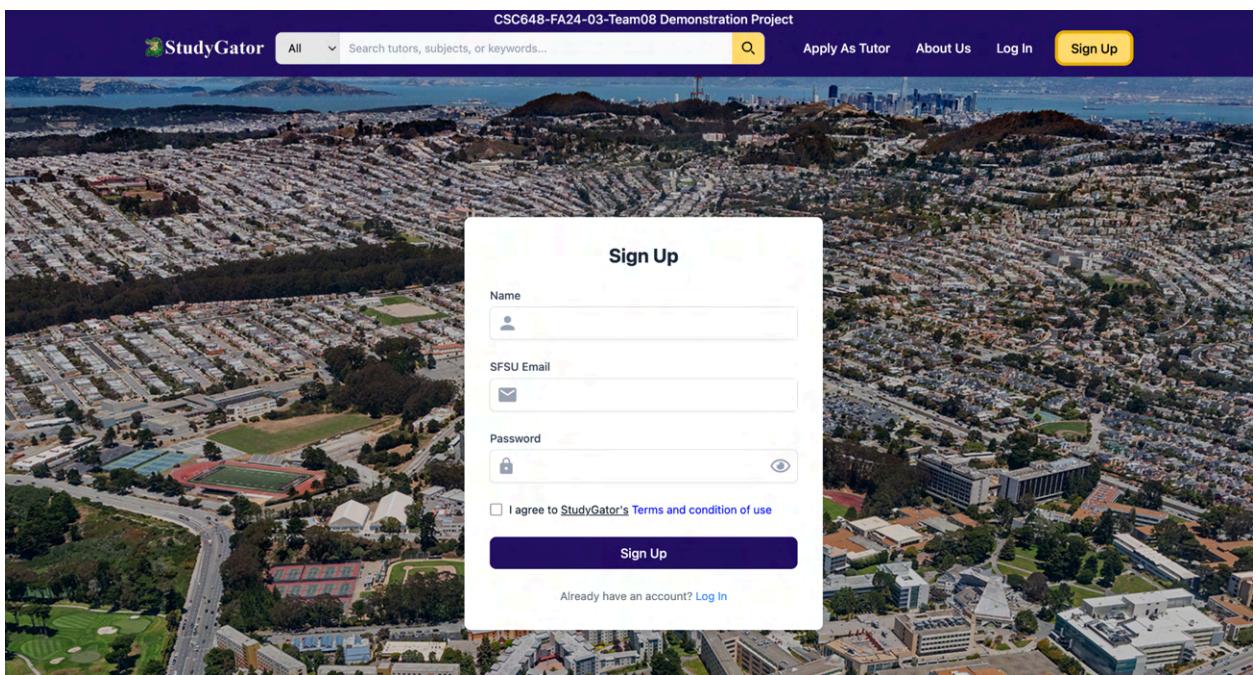
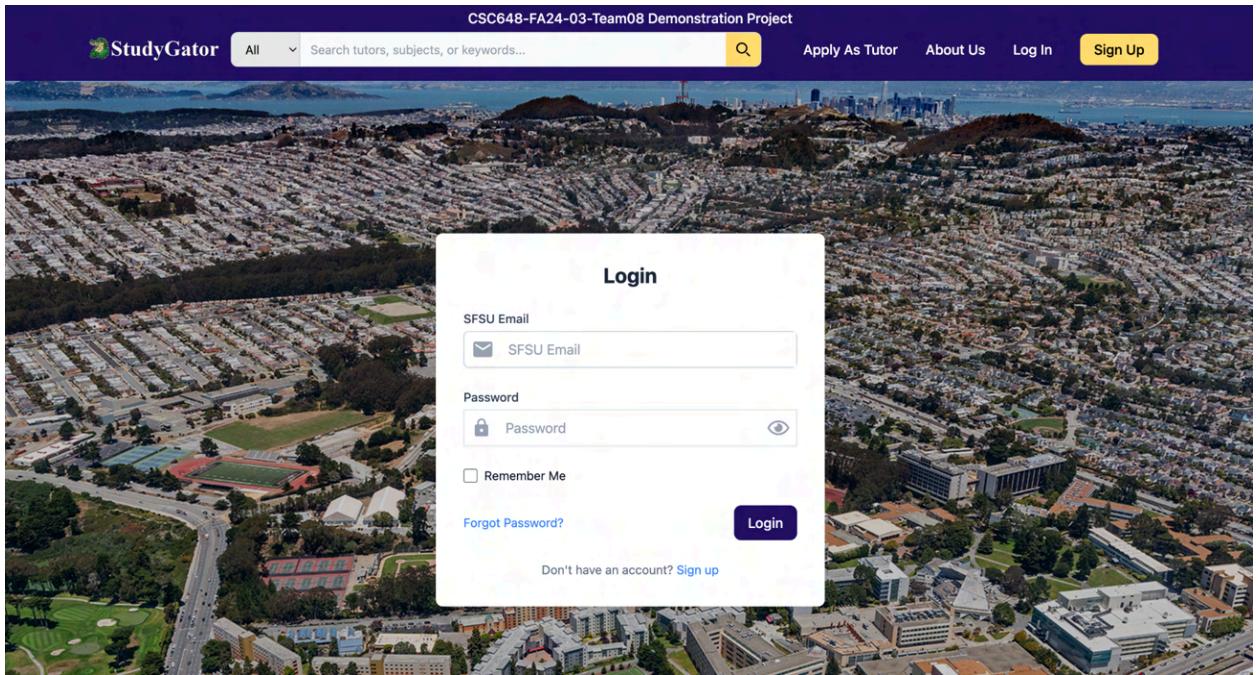
**Professional Summary**  
Detail-oriented and results-driven software developer with over 5 years of experience in designing, developing, and deploying scalable web applications. Proficient in modern programming languages, frameworks, and cloud-based architectures. Adept at collaborating in team environments and delivering solutions that align with business goals.

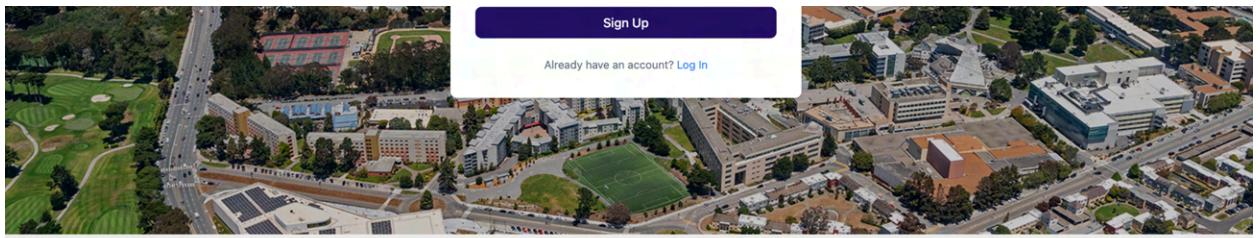
**Skills**

- Programming Languages: JavaScript, Python, Java, C++
- Web Development: React, Node.js, Angular, HTML, CSS, SASS
- Databases: MySQL, MongoDB, PostgreSQL
- Tools & Platforms: Docker, Kubernetes, Git, Jenkins, AWS
- Other Skills: Agile methodologies, CI/CD pipelines, RESTful API design

**Professional Experience**  
Software Developer  
XYZ Tech Solutions, San Francisco, CA

**About Tutor**  
I'm a passionate organic chemistry tutor with a knack for





[Sign Up](#)

Already have an account? [Log In](#)

 StudyGator

FOLLOW US

Github

Discord

LEGAL

Privacy Policy

Terms & Conditions

LOCATION

HSS Building

1600 Holloway Ave.

San Francisco, CA 94132

© 2024 StudyGator™. All Rights Reserved.



CSC648-FA24-03-Team08 Demonstration Project

 StudyGator [All](#)   [Apply As Tutor](#) [About Us](#) [Log In](#) [Sign Up](#)

**Apply As Tutor Form**

**Name**

**SFSU Email**

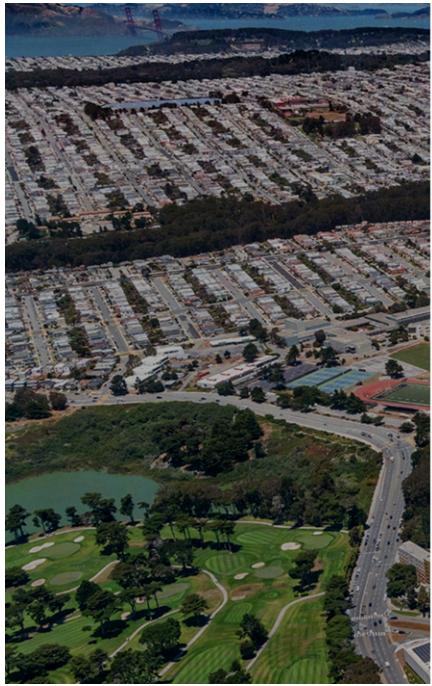
**Price \$/hr \***

**Which Subject Are You Tutoring? \***

**Creative title for listing \***

**Short pitch for listing \***

**About Me \***



**About Me \***

Write your tutor bio here

**Upload Tutor Listing Picture:** \* No file selected

Click to upload or drag and drop  
SVG, PNG, JPG, or JPEG (MAX. 5MB)

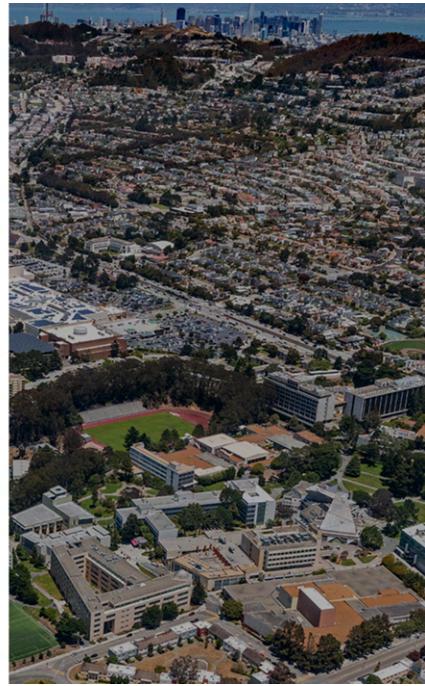
**Upload Resume/CV (optional):** No file selected

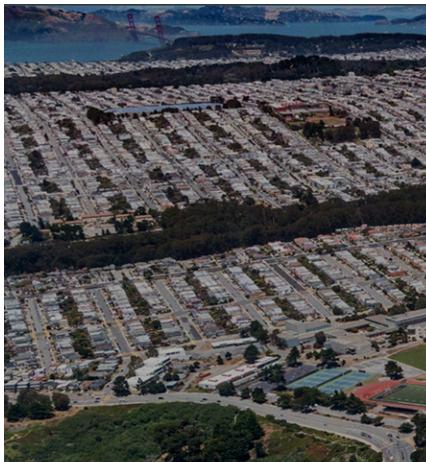
Click to upload or drag and drop  
PDF ONLY (MAX. 10MB)

**Upload Video (optional):** No file selected

Click to upload or drag and drop  
MP4, MOV, or WebM (MAX. 250MB)

Listing may take up to 24 to 48 hours to be approved by an admin  
before going public





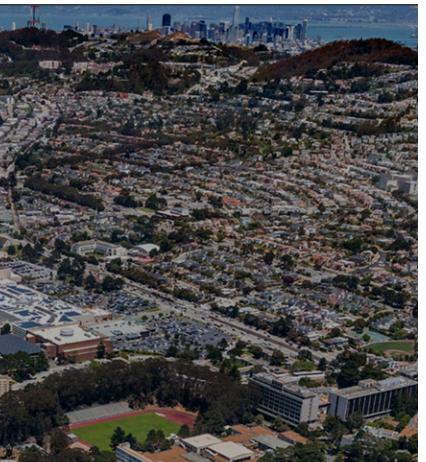
Click to upload or drag and drop  
SVG, PNG, JPG, or JPEG (MAX. 5MB)

Upload Resume/CV (optional): **No file selected**

Upload Video (optional): **No file selected**

Listing may take up to 24 to 48 hours to be approved by an admin before going public.

**Submit Form**
**Cancel**



**StudyGator**

[FOLLOW US](#)

[LEGAL](#)

[LOCATION](#)

[Github](#)

[Privacy Policy](#)

[HSS Building](#)

[Discord](#)

[Terms & Conditions](#)

[1600 Holloway Ave.](#)

[San Francisco, CA 94132](#)

© 2024 StudyGator™. All Rights Reserved.

**CSC648-FA24-03-Team08 Demonstration Project**

StudyGator

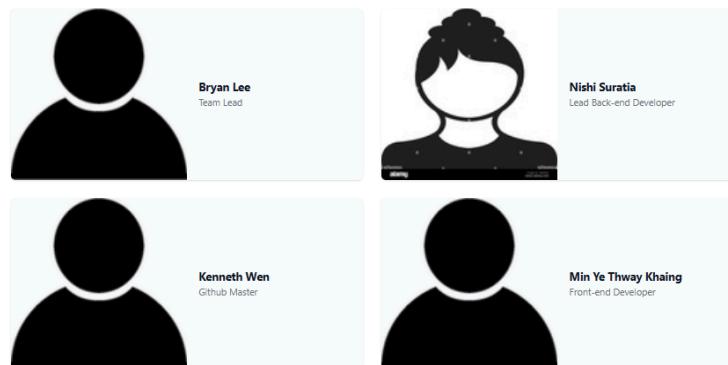
All
▼

Search tutors, subjects, or keywords...

Q
▼
Apply As Tutor
About Us
Log In
Sign Up

## Our Team

This is Team#8's group project for 648/848 Software Engineering class. As a team, innovating and delivering high-quality solutions for clients and users alike.



**StudyGator**

[FOLLOW US](#)

[LEGAL](#)

[LOCATION](#)

[Github](#)

[Privacy Policy](#)

[HSS Building](#)

[Discord](#)

[Terms & Conditions](#)

[1600 Holloway Ave.](#)

[San Francisco, CA 94132](#)

© 2024 StudyGator™. All Rights Reserved.

CSC648-FA24-03-Team08 Demonstration Project

StudyGator Computer Science Search tutors, subjects, or keywords... Apply As Tutor About Us

### Dashboard - Hello, Bryan Lee

**My Messages** View my messages here

**My Listings** Manage my tutor listings here

SENDER	TUTOR LISTING TITLE	CONTENT	DATE SENT
Bryan Lee blee@sfsu.edu	Code Mastery Made Si...	Testing	December 3, 2024

**StudyGator**

FOLLOW US    LEGAL    LOCATION

Github    Privacy Policy    HSS Building

Discord    Terms & Conditions    1600 Holloway Ave.

San Francisco, CA 94132

© 2024 StudyGator™. All Rights Reserved.

CSC648-FA24-03-Team08 Demonstration Project

StudyGator Computer Science Search tutors, subjects, or keywords... Apply As Tutor About Us

### Dashboard - Hello, Bryan Lee

**My Messages** View my messages here

**My Listings** Manage my tutor listings here

SENDER	TUTOR LISTING TITLE	CONTENT	DATE SENT
Bryan Lee blee@sfsu.edu	Code Mastery Made Si...	Testing	December 3, 2024

**Tutor Listing: Code Mastery Made Simple: Your Path to CS...** Close

From: Bryan Lee (blee@sfsu.edu)

Date Sent: December 3, 2024 at 11:42 AM

Testing

Delete

CSC648-FA24-03-Team08 Demonstration Project

StudyGator Computer Science Search tutors, subjects, or keywords... Q

Apply As Tutor About Us

Hello, Bryan Lee

Dashboard Sign out

## Dashboard - Hello, Bryan Lee

**My Messages**  
View my messages here

**My Listings**  
Manage my tutor listings here

### Manage Tutor Listings

**Code Mastery Made Simple:  
Your Path to CS Success!**



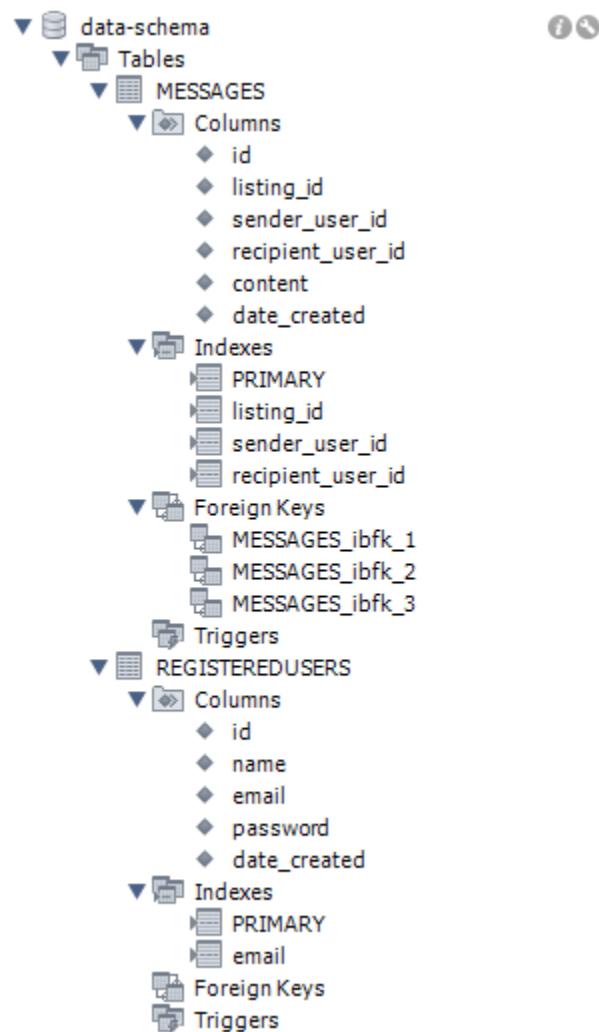
**Bryan Lee** **\$20 / hour**

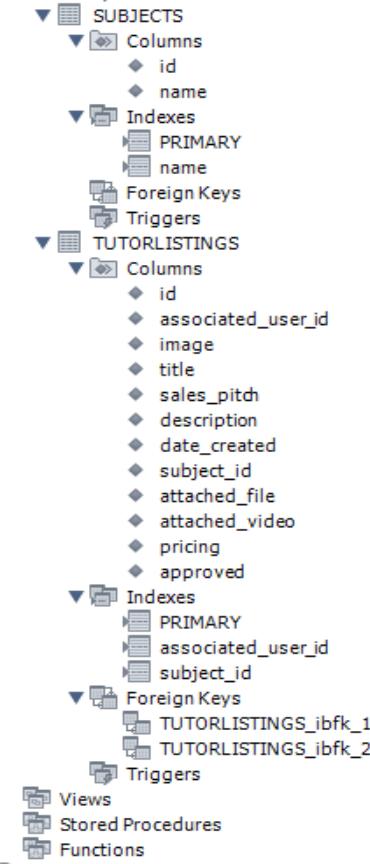
Computer Science

Struggling with computer science concepts or coding projects? Let me help you master them with ease! As an experienced computer science tutor, I specialize in breaking down complex topics into simple, digestible lessons tailored to your learning style.

More details →

#### 4. Database Organization





Messages table:

	id	listing_id	sender_user_id	recipient_user_id	content	date_created
▶	cb6e3263-b1ae-11ef-9e70-02fff31b5977	8d570e12-b08e-11ef-9e70-02fff31b5977	45bc98bb-926b-11ef-9e70-02fff31b5977	45bc98bb-926b-11ef-9e70-02fff31b5977	Testing	2024-12-03 19:42:56
*	NULL	NULL	NULL	NULL	NULL	NULL

Registered users table:

Result Grid					Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
	id	name	email	password	date_created			
	4b67b50c-b129-11ef-9e70-02fff31b5977	Luna Brandt	luna@sfsu.edu	\$2b\$10\$oKJLMNuzsgw/EPcqX1P4zOGzKhPKZ...	2024-12-03 03:47:18			
	55bf2e16-b08f-11ef-9e70-02fff31b5977	Jane Doe	jane@sfsu.edu	\$2b\$10\$ejJa18PQWnQRa8102AXKUef.lvwkhJ1...	2024-12-02 09:25:13			
	578014c2-b125-11ef-9e70-02fff31b5977	James Jones	james@sfsu.edu	\$2b\$10\$AhFWdNhlluKhSHYY0RkBeSz4olgixYC...	2024-12-03 03:19:01			
	65f972ed-b126-11ef-9e70-02fff31b5977	Emily Ventura	emily@sfsu.edu	\$2b\$10\$e5DrTzW4oNsFQ8/R/advOmHWS3cO....	2024-12-03 03:26:34			
	7e6e133e-9c22-11ef-9e70-02fff31b5977	Bryan Li	thisisatest@sfsu.edu	\$2b\$10\$Qc7Rg06xId127qTMX2CUoxTAFox5k...	2024-11-06 09:35:43			
	8c60004b-92b2-11ef-9e70-02fff31b5977	Kai	kai@sfsu.edu	\$2b\$10\$ysmc30bAKSZQFuDHjY/C5.Ft8.FKAcv0...	2024-10-25 09:21:42			
	96e426bd-b14c-11ef-9e70-02fff31b5977	Sophie Ramsey	sophie@sfsu.edu	\$2b\$10\$Q.7kSmJav.UDgalRL8TDv.YxBebmJMw...	2024-12-03 07:59:57			
	9b401fbca1a4-11ef-9e70-02fff31b5977	Bryan Test	bryan@sfsu.edu	\$2b\$10\$Z6tp4oHf8y1Tejhg7oEeYSg.CdmX6h...	2024-11-13 09:49:42			
	b22f28e7-a727-11ef-9e70-02fff31b5977	Testing1234	test123@sfsu.edu	\$2b\$10\$1kC3ZQaSPZI.okbCO8irXuKCuZS3HtQc...	2024-11-20 10:10:40			
	b6f13071-9d7a-11ef-9e70-02fff31b5977	Enjun	eli14@sfsu.edu	\$2b\$10\$86VAD.esUUmPcvnzGTU.hOJNH9Ho...	2024-11-08 02:39:45			
	bf4e4808-b139-11ef-9e70-02fff31b5977	Camden Clay	camden@sfsu.edu	\$2b\$10\$dpju70yVf/gNwPrn4zvt9DO9yHJtLqMM...	2024-12-03 05:45:05			
	cf030e95-9353-11ef-9e70-02fff31b5977	John Johnson	john@sfsu.edu	\$2b\$10\$899OpGiVIkIK2K0Z.UboxemR8SUbpvD...	2024-10-26 04:36:03			
	d4bf6e9c-b14b-11ef-9e70-02fff31b5977	Jason Fowler	jason@sfsu.edu	\$2b\$10\$.TDCDVXqhSE7vm.f/dqxx.pZ.o1bt6w4...	2024-12-03 07:54:32			
	dbf9eae5-9f15-11ef-9e70-02fff31b5977	Gabe	ruy1@sfsu.edu	\$2b\$10\$Y6ZMDByDCG8obM6cveCns.g1JaPqPK...	2024-11-10 03:42:50			
	dc8e7158-b122-11ef-9e70-02fff31b5977	Annie Annison	ann@sfsu.edu	\$2b\$10\$A/K3PeAZV8hbxGfA1edWeAuzqKGxm...	2024-12-03 03:01:15			
	de26d872-b125-11ef-9e70-02fff31b5977	Clark Woods	clark@sfsu.edu	\$2b\$10\$!Zrz3/TJxzaXpPuRE57/OaHcVEsQs3Q...	2024-12-03 03:22:47			
*	df227392-9c22-11ef-9e70-02fff31b5977	Testing123	Test@sfsu.edu	\$2b\$10\$PJBO6dPAOBmqplL4cahwR0u.Vae9zga...	2024-11-06 09:38:25			
*	NULL	NULL	NULL	NULL	NULL			

Subjects table:

	id	name
▶	96b5b71e-941b-11ef-9e70-02ffff31b5977	Chemistry
	96b5bf0-941b-11ef-9e70-02ffff31b5977	Computer Science
	96b5b33b-941b-11ef-9e70-02ffff31b5977	Math
*	NULL	NULL

Tutor listings table:

	id	associated_user_id	image	title
▶	0c7f31d1-b13a-11ef-9e70-02ffff31b5977	bf4e4808-b139-11ef-9e70-02ffff31b5977	BL0B	Excel in Math: Expert Tutoring Tailored to You
	0ce97e2f-b125-11ef-9e70-02ffff31b5977	dc8e7158-b122-11ef-9e70-02ffff31b5977	BL0B	Unlocking the Secrets of Chemistry
	1a379057-b08f-11ef-9e70-02ffff31b5977	cf030e95-9353-11ef-9e70-02ffff31b5977	BL0B	Unlock the Magic of Molecules: Expert Chemistry...
	37c3806d-b126-11ef-9e70-02ffff31b5977	de26d872-b125-11ef-9e70-02ffff31b5977	BL0B	Math Made Simple: Expert Guidance to Achieve ...
	5f994e37-b14e-11ef-9e70-02ffff31b5977	24bbc3e8-b14e-11ef-9e70-02ffff31b5977	BL0B	Conquer Organic Chemistry: Expert Tutoring fo...
	65d926c5-b14d-11ef-9e70-02ffff31b5977	196ca37b-b14d-11ef-9e70-02ffff31b5977	BL0B	Master Computer Science: Expert Tutoring for ...
	77a19075-b14c-11ef-9e70-02ffff31b5977	d4bf6e9c-b14b-11ef-9e70-02ffff31b5977	BL0B	Unlock the Secrets of Chemistry
	854241b9-b125-11ef-9e70-02ffff31b5977	578014c2-b125-11ef-9e70-02ffff31b5977	BL0B	Master Computer Science with Confidence!
	8d570e12-b08e-11ef-9e70-02ffff31b5977	45bc98bb-b926-b11ef-9e70-02ffff31b5977	BL0B	Code Mastery Made Simple: Your Path to CS Su...
	8dc96769-b126-11ef-9e70-02ffff31b5977	65f972ed-b126-11ef-9e70-02ffff31b5977	BL0B	Crack the Code of Math: Personalized Tutoring ...
	9d8ee76c-b08f-11ef-9e70-02ffff31b5977	55bf2e16-b08f-11ef-9e70-02ffff31b5977	BL0B	Math Made Easy: Expert Tutoring for All Levels!
	add3c761-b124-11ef-9e70-02ffff31b5977	dc8e7158-b122-11ef-9e70-02ffff31b5977	BL0B	Unlocking the Secrets of Math: Expert Tutoring ...
	b786acab-b14d-11ef-9e70-02ffff31b5977	196ca37b-b14d-11ef-9e70-02ffff31b5977	BL0B	Come learn math with me!
	edb88c46-b14c-11ef-9e70-02ffff31b5977	96e426bd-b14c-11ef-9e70-02ffff31b5977	BL0B	Chemistry Simplified!
	ffff9829-b12c-11ef-9e70-02ffff31b5977	4b67b50c-b129-11ef-9e70-02ffff31b5977	BL0B	Personalized Computer Science Tutoring
*	NULL	NULL	NULL	NULL

sales_pitch	description	date_created	subject_id	attached_file	attached_video	pricing	approved
Struggling with math? I provide clear, step-by-s...	As an experienced math tutor, I'm dedicated to ...	2024-12-03 05:47:14	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	100	1
Struggling with chemistry? I offer personalized t...	I'm a passionate chemistry tutor dedicated to h...	2024-12-03 03:16:56	96b5b71e-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	25	1
Discover the magic of chemistry with personaliz...	Hi, I'm John Johnson, your go-to guide for mast...	2024-12-02 09:23:33	96b5b71e-941b-11ef-9e70-02ffff31b5977	BL0B	NULL	10	1
Conquer math with clarity and confidence! I pro...	I'm a math tutor passionate about making math ...	2024-12-03 03:25:17	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	55	1
Struggling with reaction mechanisms or molecula...	I'm a passionate organic chemistry tutor with a ...	2024-12-03 08:12:44	96b5b71e-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	99	1
Unlock the world of coding, algorithms, and dat...	I'm a passionate computer science major with a ...	2024-12-03 08:05:45	96b5b5f0-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	30	1
Master chemistry concepts with ease! Whether ...	I am a passionate chemistry tutor with a deep u...	2024-12-03 07:59:05	96b5b71e-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	88	1
Overcome CS challenges with tailored tutoring s...	As a dedicated CS tutor, I specialize in helping s...	2024-12-03 03:20:17	96b5b5f0-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	90	1
Struggling with computer science concepts or co...	Hi, I'm Bryan, a passionate and experienced Co...	2024-12-02 09:19:37	96b5b5f0-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	20	1
Math doesn't have to be hard! I offer customiz...	As a math tutor with a passion for teaching, I s...	2024-12-03 03:27:41	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	70	1
Struggling with math? Let me help you conquer i...	Hi, I'm Jane Doe, a passionate math tutor dedic...	2024-12-02 09:27:14	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	50	1
Struggling with math? I offer personalized tutori...	I'm a passionate math tutor dedicated to helpin...	2024-12-03 03:14:16	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	25	1
Struggling with algebra, calculus, or geometry?...	I am a dedicated math tutor with a passion for ...	2024-12-03 08:08:02	96b5b3b-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	30	1
Struggling with chemical equations or the myste...	I am a dedicated chemistry tutor with a strong ...	2024-12-03 08:02:23	96b5b71e-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	5	1
Unleash the world of tech with tailored computer ...	I'm a computer science tutor passionate about ...	2024-12-03 04:13:50	96b5b5f0-941b-11ef-9e70-02ffff31b5977	BL0B	BL0B	65	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## 5. Github organization

## List main branches in your team's GitHub:

- 1) main
- 2) message
- 3) dashboard
- 4) Detail
- 5) Dev

Warning: To reduce conflicts and improve productivity, branches other than “main” are frequently squashed, merged, and deleted via pull request. The different major branches are listed but should not be classified under the term “main” branches.

## Clarify who had access to the main branch:

- a) Kenneth Wen (Github Master)
- b) Bryan Lee (Team lead)

## Screenshot of the team's GitHub home page:

The screenshot shows a GitHub repository page for 'csc648-fa24-03-team08'. The repository is private and forked from 'CSC-648-SFSU/csc-648-sfsu-csc-648-section-03-fall-2024-csc648-base-repo'. The main branch is selected, showing 5 branches and 0 tags. The repository has 200 commits ahead of the main branch. The commit list includes:

Author	Commit Message	Date
BryanL43	Fixed Google Analytics	4aee2fd · 5 days ago
	Add files via upload	2 months ago
	Fixed Google Analytics	5 days ago
	Credential instructions	2 months ago
	Initial commit	3 months ago
	Created the about page	3 months ago

The repository has a README and a LICENSE file. The 'About' section indicates it was created by GitHub Classroom. It features the URL [eclipseskakura.online/](https://eclipseskakura.online/) and tags for nodejs, nginx, reactjs, expressjs, pm2, ec2-instance, tailwindcss, and rds-mysql. The 'Releases' section shows no releases published. The 'Languages' section shows no packages published. The 'Languages' section also lists Java, Python, C, C++, C#, Go, Rust, and others.

**csc648 Repository**

Please when ready add your teams application URL or IP to the repository description. This will help with grading. Teams are expected to keep this value up to date.

Please do the following steps before completing Milestone 0.

1. Change the name of the repository. All that needs to change is the NN to your respective team number. Team

1. Change the name of the repository. All that needs to change is the NN to your respective team number. Team numbers whose value is less than 10, please pad with a 0. Ex team 1 is Team01 team 11 is Team11. Please make sure to also remove the username from the repository as well. Teams with incorrectly name repository will have points deducted from their milestone 0 grades.

- o Please follow the naming convention assigned by your instructor.

2. PLEASE REMOVE THE USERNAME FROM THE REPOSITORY NAME!!!

3. Add ALL members of your team to this repository. For it to count, they must ACCEPT the invite.

4. Fill out the table below

Student Name	Student Email	GitHub Username
Bryan Lee	<a href="mailto:blee37@sfsu.edu">blee37@sfsu.edu</a>	BryanL43
Kenneth Wen	<a href="mailto:kwen@sfsu.edu">kwen@sfsu.edu</a>	Kenwen1
Nishi Suratia	<a href="mailto:nsuratia@sfsu.edu">nsuratia@sfsu.edu</a>	nishisuratia
Min Ye Thway Khaing	<a href="mailto:mkhaing1@sfsu.edu">mkhaing1@sfsu.edu</a>	MinYeThway

**NO code needs to be stored in the root of your repository. You may rename the application folder if you like to your team's application name. But all source code related to your team's application should be stored inside the application folder.**

#### Languages



© 2024 GitHub, Inc.

[Terms](#)

[Privacy](#)

[Security](#)

[Status](#)

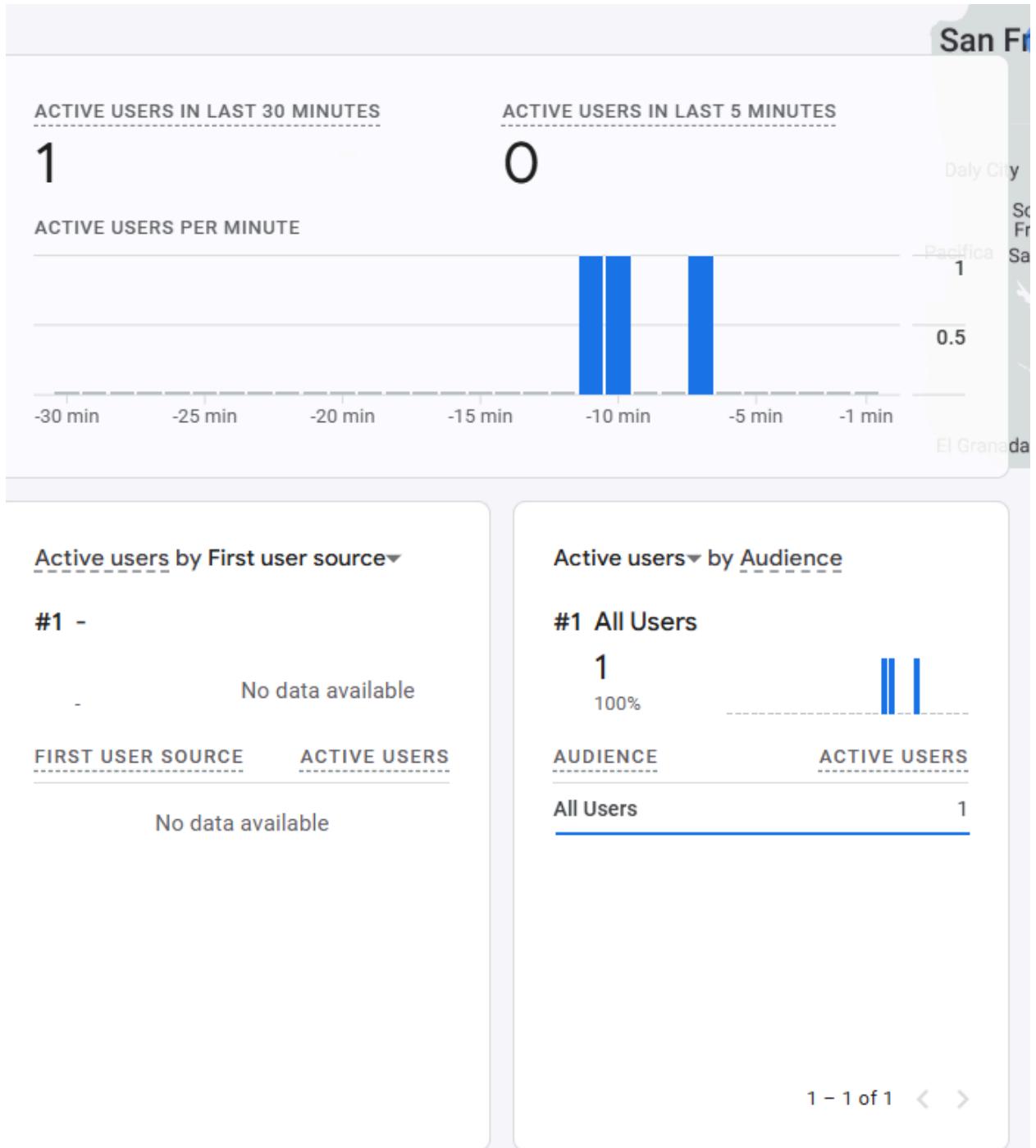
[Docs](#)

[Contact](#)

[Manage cookies](#)

[Do not share my personal information](#)

## 6. Google Analytics stats plot for our WWW site



### Views by Page title and screen name

#### #1 StudyGator



1 - 6 of 7 < >

### Event count by Event name

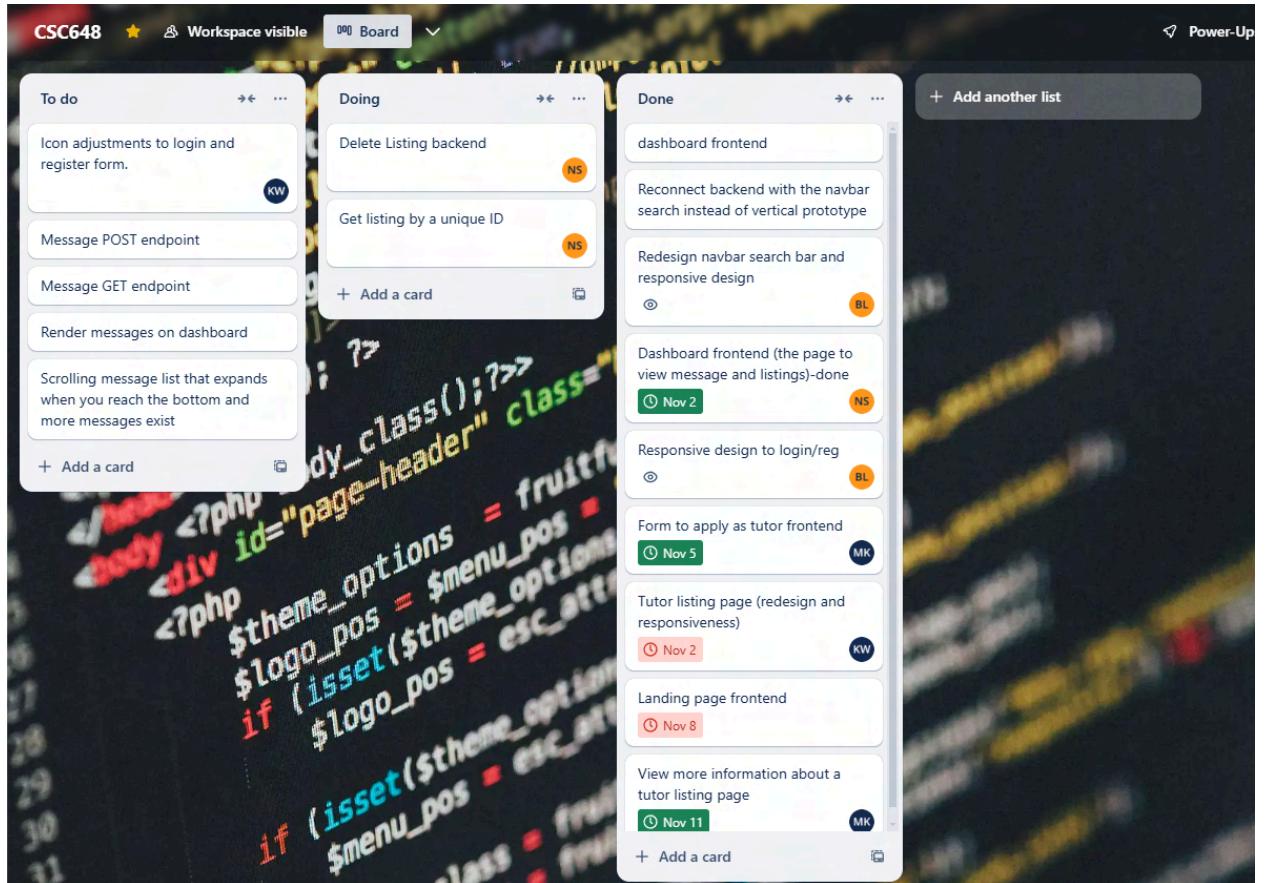
#### #1 page\_view



1 - 5 of 5 < >

## 7. Product Management

Trello:



## Discord:

Bryan 11/6/2024 2:31 AM

@everyone I think it is time for a team reset. Please free up your schedule. We will be staying until at least 7:40-8pm tomorrow after class. We will do some coding during our meeting.

Here are the simple task I want us to do and restart our teamwork:

- Nishi: You will create a backend code to fetch all the tutor listing associated with a **logged in** user and render it on the frontend (frontend is already implemented more help in class)
- Kai: You will create the “message” pop up UI
- Kenneth: You will rescale the Apply as tutor form (its too “skinny” looking right now and plain) **AND** you will fix the filter for browse page as the design is terribly clipped. I will provide more details in class.

I will act as QA and also provide help on refreshing the github and working environments so we are all on the updated version.

Non-negotiable coding session **MAKE TIME**

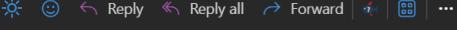
1

Rryan 11/6/2024 2:40 AM

## **8. Team member self-assessment and contributions**

Min Ye Thway Khaing - Self-Assessment and Contributions - Frontend Lead

 Min Ye Thway Khaing  
To: Bryan J Shen Lee; Kenneth Wen; Nishi Rushirajsinh Suratia

 Tue 12/17/2024 2:10 PM

**Contributions:**  
As the Frontend Lead, I contributed to the following:  
Login Page: Secure UI with email validation.  
Sign-Up Page: Ensured email validation for "@sfsu.edu" and terms acceptance.  
Tutor Listing Page: Implemented filters, display cards, and "View More" navigation.  
Tutor Detail Page: Detailed profiles with ratings, images, and resume links.  
Message Page & Pop-Up: Designed messaging UI and modal for better UX.  
About Page: Assisted with team member cards and layout.

For GitHub Submissions, My commit count shows only 19 submissions, but some were deleted by the team lead during cleanup. All my work can be verified in the final project deliverables.

The main challenge I encountered was designing the UI and choosing consistent styles for the application, which was the most challenging part—balancing aesthetics with functionality required careful decision-making and testing.

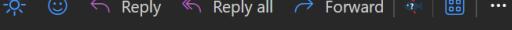
GenAI tools helped with debugging, CSS layouts, and input validation patterns, significantly improving my productivity and saving time.

Next time I would like to improve my collaboration early on UI design decisions to ensure consistency and maintain better local backup to avoid losing work and the last one is to improve time management for testing and refinements.

Thank you all for the teamwork and collaboration!  
Best Regards,  
Min Ye Thway Khaing (kai)  
Frontend Lead

Kenneth Wen Team Member Self Assessment

 Kenneth Wen  
④ To: ④ Bryan J Shen Lee; ④ Nishi Rushirajsinh Suratia; ④ Min Ye Thway Khaing

 Tue 12/17/2024 2:48 PM

- Github Master
- Created personae and use cases for website
- Created storyboards for the website
- Search function for the website
- Search page UI
- Price filter in the search page

I have 38 commits on GitHub. The number might be low due to my bad habit of not committing frequently, which I have gotten better at throughout the course of this project.

One of the main challenges I encountered during this project was to learn how to use Tailwind CSS and React. It was my first time working with these tools and the first time I interacted this much with the frontend. It took a while to get used to, but the experience felt rewarding in the end.

GenAI was useful for coming up with ideas for our project, such as when we needed to create personae and potential use cases. It is nice to have some quick inspiration or ideas to help kickstart the work.

After learning the SE management and processes, I think I can create better UI that is more user friendly and efficient next time. I would also have more tools and am more comfortable with the whole designing process.

## self assesment

NS

Nishi Rushirajsinh Suratia

To: Min Ye Thway Khaing; Kenneth Wen; Bryan J Shen Lee



Tue 12/17/2024 3:01 PM

As the Backend Lead for the StudyGator project, I wanted to provide an overview of my contributions to the development process. While my GitHub commit count stands at 32, I believe this reflects the nature of my role in managing the backend of a project that primarily focuses on the UI features.

Below is a detailed breakdown of my contributions:

**MessageModel.js:** Designed and implemented the messaging functionality for StudyGator which allows ability to send a message to tutor as well as delete it. Developed corresponding routes and controllers in the `routes` and `controller` folders.

**ListingModel.js:** Built the listing system for tutor based on his application defining its logic for fetching data from database.

I faced some challenges in writing query to fetch data accurately for a specific function in listing which was time consuming and in some time management tasks.

The usage of GenAI tools to identify optimization opportunities was extremely helpful that significantly improved performance of our model

This experience has been invaluable in honing both my technical and soft skills. It improved my backend expertise, while collaborating with the team taught me the importance of effective communication and teamwork.

Thank you for your guidance and support throughout this course.

Best,

Nishi Suratia

Backend lead

## Bryan Lee - Team Member Self Assessment - Team Lead



Bryan J Shen Lee

To: Bryan J Shen Lee

Cc: Min Ye Thway Khaing; Kenneth Wen; Nishi Rushirajsinh Suratia



Reply



Reply all



Forward



...

Wed 12/18/2024 12:32 PM

### Contribution to the project:

- Technical:
  - Landing page UI
  - Dashboard page UI
  - Connected a multitude of backend to frontend
  - Wrote most of the backend API POST services
  - Project architecting and DevOps
  - QA and PR manager
- Teamwork:
  - Scheduled in-person & online meetings
  - Organized task distributions
  - Resolved team conflict
  - Documentation work

I have made 109 commits to GitHub submissions, which is relatively high for many reasons. As the most proficient member of the team, I have a strong practice of committing frequently, uploading the milestone documentation after I verify, merging, and fixing bugs here and there. I was the one who had to do the initial deployment of our production base and project structure. Please also take a look at my previous emails about our application backend; they are related to my commit count.

One of the main challenges I encountered during the team project was task distribution. All of us had commitments, and that made it challenging to distribute work accordingly. There was a lot of frustration, especially when the task fell onto me due to such reasons. The other challenge was scheduling team meetings where none of us had a schedule that matched each other. The only time we could really meet was either online or in-class meeting. Due to the many scheduling conflicts, we have only been able to meet as a group in person twice outside of class time. This caused a lot of work distribution to clash, especially with other's work and effort being erased. Lastly, the most severe challenge that I encountered was a team member's lack of contribution. It resulted in a lot of work that fell upon me or distributed to the other group members. It reveals the importance of being stricter as a team leader.

and effort being erased. Lastly, the most severe challenge that I encountered was a team member's lack of contribution. It resulted in a lot of work that fell upon me or distributed to the other group members. It reveals the importance of being stricter as a team leader.

My experience with genAI was quite good. It wasn't good with any of the documentation or planning; however, it was extremely useful in giving ideas on how to structure the project technically. It convinced me to use the BLOB system for my complicated AWS ECS and RDS database system, provided MySQL assistance, and assisted with the initial development of the MVC architecture I have. I also used genAI to help out with designing UI, but it is quite terrible. It only gave me some ideas about the syntax for Tailwind CSS but is pretty much blind to designing. GenAI could have been helpful for planning; however, its ideas are primarily repetitive and typically lack the necessary "emotions" for such activities, especially in UX.

One thing that I would do better as a team lead next time is to be stricter. I was a bit too relaxed at the start, and that bit me back. It is critical that each team member does their part of the work or the work falls upon me, the team lead. Being too accommodating resulted in a lot of work being poorly distributed and causing more stress to the other team members. Additionally, I need to gather more points on what to discuss during my future team meetings. I felt that most of my meetings were done with last-minute plans, but they did reflect the nature of agile software development. There is just a lot of stuff you can't predict and must accommodate on the fly. Lastly, I should be doing more Scrum every half a week to get a status report for the team. It was not easy to accommodate everyone's schedule, so the better tactic would be Scrum, whether it is online or in-person; however, it needs the cooperation of all team members.

The last thing I want the instructor to know is to refer to my prior emails on team status. They are related to my larger commit count. Do note the team's git commit does not reflect the amount of work done and should consider such underlying factors (the reports) and git familiarity.