SW Engineering CSC 648-848 Fall 2024

GatorTutor

Team 6

Team Lead, Alex Hoff - (ahoff2@mail.sfsu.edu)
Backend Lead 1, Dylan Faulder - (dfaulder@mail.sfsu.edu)
Backend Lead 2, Austin Ng - (Ang@sfsu.edu)
FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)
Github Master, Jack Richards - (jrichards7@sfsu.edu)

https://gatortutor.net/

History

Date Submitted:	**December 17th, 2024**

Product Summary

Product Name and Description: "*GatorTutor*"

A site for San Francisco State University students to find academic help from other students who've been in their position. They are able to find and search for tutors that specialize in the courses they need assistance in.

GatorTutor is a tutoring service created by San Francisco State University students for SFSU students. We aim to provide a place that facilitates the matching between all SFSU students with their best fit SFSU tutor. We feel that the dividing factor between a successful student and a struggling student is gaps in their knowledge and our goal is to bridge that gap. Our service helps match tutors with experience in those subjects and the students seeking assistance. Since our application is institution specific, our tutors are able to provide a level of service unique to SFSU that foreign services can not compete with.

Itemized List of P1 Functions:

Browsing: Users shall be able to browse and easily navigate the website and most of what it has to offer.
Searching: Users shall be able to use the search bar in order to search for SFSU-specific classes and professors to find hyper-relevant material.
Posting: Users shall be able to create a tutor profile and advertise their skills on the website.
Dashboard: Users shall have a functioning dashboard that displays their postings and messages received.
Messaging: One way messaging system from student to tutor.
Approving Post: Site Admin shall be able to delete inappropriate items or users based on the criteria of the post.
Deployment URL:
https://gatortutor.net/

SW Engineering CSC 648-848 Fall 2024

GatorTutor

Team 6

Team Lead, Alex Hoff - (ahoff2@mail.sfsu.edu)
Backend Lead 1, Dylan Faulder - (dfaulder@mail.sfsu.edu)
Backend Lead 2, Austin Ng - (Ang@sfsu.edu)
FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)
Github Master, Jack Richards - (jrichards7@sfsu.edu)

Milestone 1

History

Date Submitted:	**October 12th, 2024**
Date Revised:	**October 15th, 2024**

1. Executive Summary:

GatorTutor is a tutoring service created by San Francisco State University students for SFSU students. We aim to provide a place that facilitates the matching between all SFSU students with their best fit SFSU tutor. We feel that the dividing factor between a successful student and a struggling student is gaps in their knowledge and our goal is to bridge that gap. Our service helps match tutors with experience in those subjects and the students seeking assistance. Since our application is institution specific, our tutors are able to provide a level of service unique to SFSU that foreign services can not compete with.

Since our team is composed of SFSU students, we are able to target institution specific problem areas that's stopping students from achieving their potential through tutoring. Our application allows students to search for tutors by university specific classes. Professors don't always teach the same curriculum as others, so tutors are able to specify which ones they have experience with, allowing students to select them based on their issues. Tutors are able to create profiles unique between individuals through multimedia to customize their impressions, introduce themselves through text descriptions, show interested students their rates along with their availability, and appear on searches by course identifiers. Our platform also facilitates communication between interested students and tutors with our built-in messaging service.

2. CSC 648 Personae:

Amy - Student

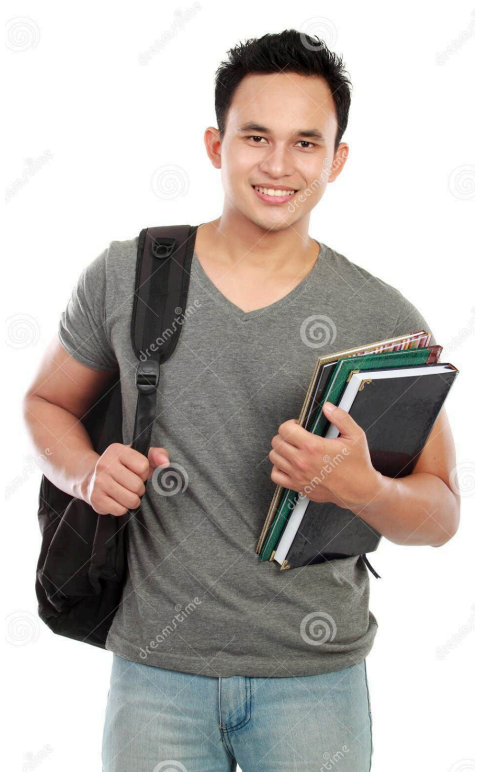
Amy is a student at SFSU

- Has basic website navigation skills
- Has strong math Knowledge
- Needs $$
- Wants to become a tutor

Goal And scenario -
Amy wants to be able to easily sign up for GatorTutor and place all her information so that any students who need math help can easily find and contact her.

- **Name**: Amy
- **Demographics**: [20, Female, Student]
- **Goals**: [Wants to raise money for tuition by tutoring on gator tutor]
- **Skills**: [Strong Math skills, Basic WWW experience]
- **Pain Points**: [Lacks skills to function on using the mobile version]
- **Motivation**: [Raise money for tuition]

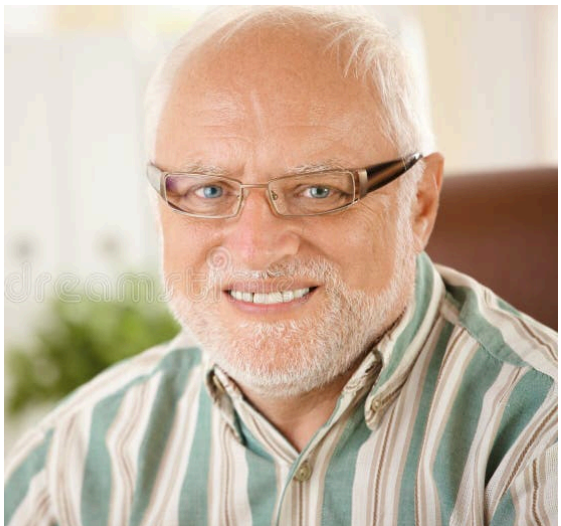
John - Struggling Student



- Is Studying nursing
- Struggles in Biology class
- Does not have a lot of time
- Wants to schedule a tutor easily

Goal And scenario -
John Wants to schedule an appointment with a tutor for his BIO class. He wants to look to see if any of his classes have tutors. If he is able to find one he wants to be able to contact and set up an appointment.

- **Name**: John
- **Demographics**: [18, Male, Student]
- **Goals**: [Wants to pass BIO class]
- **Skills**: [Knows how to use mobile]
- **Pain Points**:[Cannot find his class listed on the site]
- **Motivation**: [Wants to pass BIO and needs a tutor to do s



Herald - Adult Continuing student looking for help

- Came back to SFSU to study writing
- Has very bad writing skills and wants to get a tutor to ask how they write better essay for their class

- Is a Adult student who does not use technology at all
    - Needs to be able to simply sign up and look for a tutor

Goal And scenario -
An older student who has bad technology skills wants to find a tutor in as little steps as possible. Wants to first browse the tutors to see if there is anyone he would like.

- **Name**: Herald
- **Demographics**: [65, Male, Student]
- **Goals**: [Wants to improve his writing skills and needs a tutor to do so]
- **Skills**: [Very little experience in using websites]
- **Pain Points**:[Is hard to navigate the website]
- **Motivation**: [Wants to improve their writing skills for their class and wants a tutor who is good at essays to help them outline a better paper]

Sarah- Admin

- A site admin
    - Has to approve submissions and posts on Gator Tutor
    - Is skilled and knowledge about the site
    - Monitors the site to make sure there are no problems

Goal And scenario -
Sarah is a site admin who has control over the site and is in charge of taking care of monitoring the website and approving new tutors and messages.

- **Name**: Sarah
- **Demographics**: [26, Female, Admin]
- **Goals**: [Able to maintain the site and allow/take down posts]
- **Skills**: [Strong skills in understanding the site]
- **Pain Points**:[Has trouble overlooking the site]
- **Motivation**: [Wants to make sure the website is well maintained and there are no bad faith tutors or posts getting made without proper overview so that users can have a good safe area to find tutors]

3. SFSU Tutoring Web Application High Level Use Cases:

Use Case 1: Amy Explores Tutoring Options
Case Number: 1

Actors: Amy

Narrative: Amy, a student at SFSU, visits the tutoring app to explore help for her calculus class. Without requiring her to log in or create an account, the system allows her to browse available tutors by subject and filter by availability. After reviewing a few profiles, Amy decides to contact a tutor who seems like a good fit. The system then prompts her to complete a registration (lazy registration) before she can send a message or arrange a meeting time through the built-in messaging system. Once registered, she proceeds to arrange the session and later rates the tutor based on her experience.

Postconditions: Amy successfully connects with a tutor, attends the session, and leaves a rating.

Exceptions: If no tutors match her search criteria, the system suggests alternative options or prompts Amy to broaden her search

---

Use Case 2: John Finds a Specific Tutor
Case Number: 2

Actors: John

Narrative: John, another student at SFSU, hears about a specific tutor from a friend and wants to find them on the platform. John logs in and enters the tutor's name in the search bar. The system returns a list of tutors with matching or similar names. John clicks on the tutor's profile to view more information and contacts them via the messaging feature to arrange a session. The tutor then messages John outside of the website and they schedule a time to meet.

Postconditions: John successfully finds the tutor he was looking for and schedules a session.

Exceptions: If the tutor is unavailable, the system suggests similar tutors or conveys to John that the specific tutor he is looking for is not available at any time.

Use Case 3: Amy Becomes a Tutor
Case Number: 3

Actors: Amy

Narrative: Amy, who has previously used the platform to find tutoring help, now wants to become a tutor herself. Amy fills out the required information, including subjects she can tutor, her rates, availability, a nd a brief bio. She reviews her post and submits it for approval. After submitting, Amy is notified that her tutor post is pending approval. Once approved, her post goes live, and she receives notifications when students contact her.

Postconditions: Amy successfully submits her tutor post, and it goes live once approved.

Exceptions: If her post is rejected, Amy receives a notification. She can then try to make a new post.

Use Case 4: Amy Receives a Message from a Student
Case Number: 4

Actors: Amy, John

Narrative: After Amy's tutor post goes live, John, a student, sends her a message asking about her availability for a tutoring session. Amy receives a notification in the app, indicating that she has a new message. She reads the message with John's contact info and reaches out to him outside of the site. They arrange for an appropriate time to meet on an alternate platform such as sms message or facebook.

Postconditions: Amy successfully communicates with John and schedules a tutoring session.

Exceptions: If either party is offline, messages remain in the inbox until the user returns.

Use Case 5: John Reports a Tutor
Case Number: 5

Actors: John

Narrative: John, after a negative experience with a tutor, decides to report the issue. He logs into the app, navigates to his message history, and selects the tutor's profile. John clicks the 'Report' button (may change) and is redirected to a form where he can describe the issue. He fills out the details, explaining the problem, and submits the report. The system informs John that the report has been submitted and will be reviewed within 24 hours.

Postconditions: John submits a report, and the system confirms it will be reviewed by an admin.

Exceptions: If the report does not meet the platform's criteria for action, John is informed and provided with relevant information about the reporting process.

Admin Use Case: Sarah Approves Tutor Posts

Case Number: 6

Actors: Sarah

Narrative: Sarah, the admin responsible for approving tutor posts, logs into the admin interface. She is notified of new pending tutor posts. Sarah reviews each post to ensure it meets the platform's guidelines, including valid subject matter, appropriate pricing, and a clear bio. If a post is acceptable, Sarah approves it, and the tutor is notified. If it doesn't meet the standards, Sarah rejects it. Once approved, the post is live on the platform, and tutors can start receiving messages from students.

Postconditions: Tutor posts are reviewed, approved, or rejected. Approved posts go live, and tutors are notified.

Exceptions: If Sarah rejects a post, the tutor receives feedback? (possibly just deleted with no context) and is asked to revise and resubmit.

Admin Use Case: Sarah Reviews Pending Reports

Case Number: 7

Actors: Sarah

Narrative: Sarah, the admin, receives a notification about a new report submitted by a user. She logs into the workbench system and checks the list of pending reports. Sarah reviews the details of the report and cross-references it with the tutor post in question. If the tutor's post violates the platform's guidelines, Sarah deletes the post and marks the report as resolved. If the report is not valid, Sarah dismisses it.

Postconditions: Sarah resolves the report, either by removing the offending tutor post or dismissing the report if it is invalid.

Exceptions: If additional information is required, Sarah contacts the reporting user for clarification before taking further action (TODO, may change).

4. Main Data and Entities:

Guest **User**

Entity Name: Guest User

Description: A general term for anyone who interacts with the system.

Usage: All users must register with SFSU credentials to use the platform. They can create accounts, search for tutors, and communicate with others. Some users may create tutor posts.

Registered User

Entity Name: Registered User

Description: A user who has successfully registered on the platform.

Usage: Registered users can create posts, search for tutoring services, send messages, and interact with other users. They can also associate themselves with tutor posts.

Tutor Posts

Entity Name: Tutor Posts

Description: Posts created by registered users who offer tutoring services.

Usage: These posts contain details such as subjects offered, availability, and hourly rate. Registered users can create these posts and students can search for and contact tutors through them. Tutor posts are visible to other users, and interactions are initiated through them.

Admin

Entity Name: Admin

Description: An individual designated to manage the platform and handle database operations.

Usage: Admins do not interact with the platform as regular users do. They only use tools such as an MQL workbench to manage the system's data. They handle tasks such as profile approvals, database queries, and maintenance. Admins do not have standard user profiles.

Message

Entity Name: Message

Description: A communication between registered users, typically between students and tutors.

Usage: Messages are exchanged through the platform for inquiries about tutoring services and general communication.

Subject

Entity Name: Subject

Description: A specific academic subject at SFSU.

Usage: Subjects are associated with tutor posts. Students search for tutors based on these subjects.

5. High Level Functional Requirements:

1.1 Registered Users - Creating Posts: Users shall be able to create a tutor profile and advertise their skills on the website.

1.2 Users shall be able to choose their desired categories of study to advertise (CSC, MATH, PHYS, ETC.)

2. Registered Users - Dashboard: Users shall have a functioning dashboard that displays their postings and messages received.

3. Unregistered/Registered Users - Browsing: Users shall be able to browse and easily navigate the website and most of what it has to offer.

4. Unregistered/Registered Users - Searching: Users shall be able to use the search bar in order to search for SFSU-specific classes and professors to find hyper-relevant material.

5. Admin - Approval of Tutor Posts: Site Admin shall be able to approve or deny any tutor post through the MySQL Workbench.

6. Admin - Deleting Inappropriate Content: Site Admin shall be able to delete inappropriate items or users based on the criteria of the post.

6. Nonfunctional Requirements:

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers

3. All or selected application functions shall render well on mobile devices (no native app to be developed)

4. Posting of tutor information and messaging to tutors shall be limited only to SFSU students

5. Critical data shall be stored in the database on the team's deployment server.

6. No more than 50 concurrent users shall be accessing the application at any time

7. Privacy of users shall be protected

8. The language used shall be English (no localization needed)

9. Application shall be very easy to use and intuitive

10. Application shall follow established architecture patterns

11. Application code and its repository shall be easy to inspect and maintain

12. Google analytics shall be used

13. No 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:

GatorTutor aims to provide tutoring services tailored towards the needs of students enrolled at San Francisco State University. Unlike its competitors, this platform connects students to tutors who are familiar with the university's curriculum and can help those who are struggling with a particular course. However, this objective does not come at the expense of the simple and intuitive interface that users are accustomed to working with on other platforms. GatorTutor

makes connecting with a tutor an easy process through a search filter. Additionally, the platform also provides support for both online and in-person tutoring which ensures anyone will find a tutor that meets their needs. Tutors are also given full control over their profiles. They can set their own hourly rates, availability, and subjects. These features will collectively make GatorTutor the primary destination for all SF State students looking to further their learning and academic growth.

	GatorTutor	Varsity Tutor	Preply	Wyzant
SFSU Specialized Tutoring	✔✔			
Course-specific tutoring	✔✔			
Filtered Searching	✔✔		✔✔	✔✔
User-Friendly Interface	✔✔	✔	✔	✔✔
Online and In-Person Tutoring	✔✔	✔		✔✔
Review	✔✔	✔✔	✔✔	✔

8. High-Level System Architecture and Technologies Used:

Main Software Components: Node.js and MySQL

Deployment Cloud Service: AWS

Front-End Frameworks: ExpressJS

Supported Browsers: Chrome and Firefox

External APIs: Google Analytics

9. Use of GenAI tools like ChatGPT and copilot for Milestone 1:

Our team used GenAI in order to:

- Create a structure for the Executive Summary
- Give us ideas for Personas about what a student might be looking for in a website
- Help us detail the non-functional requirements and what their purposes might be
- Aide us in the competitive analysis, thinking of what features a competitive tutoring website may have

Was it useful?

GenAI was quite useful in spitballing ideas to us that we, as a group, could then further explore and refine. Although we only used it for a few items in M1, we found it to be very helpful where we needed it and used it more as a tool of inspiration than anything.

10. Team and Roles:

Role	Github Master	Front-end Lead	Back-End Lead	Programming Lead	Team Leader
Name	Dylan	Dalan	Jack	Austin	Alex

11. Team Lead Checklist

- So far all team members are fully engaged and attending team sessions when required - **DONE/OK**
- Team found a time slot to meet outside of the class - **DONE/OK**

- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing - **DONE/OK**

- Team reviewed class slides on requirements and use cases before drafting Milestone 1 consistently - **DONE/OK**

- Team lead checked Milestone 1 document for quality, completeness, formatting and compliance with instructions before the submission - **DONE/OK**

- Team lead ensured that all team members read the final M1 and agree/understand it before submission - **DONE/OK**

- Team shared and discussed experience with genAI tools among themselves - **DONE/OK**

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

SW Engineering CSC 648-848 Fall 2024

GatorTutor

Team 6

Team Lead, Alex Hoff - (ahoff2@mail.sfsu.edu)
Backend Lead 1, Dylan Faulder - (dfaulder@mail.sfsu.edu)
Backend Lead 2, Austin Ng - (Ang@sfsu.edu)
FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)
Github Master, Jack Richards - (jrichards7@sfsu.edu)

Milestone 2

History

Date Submitted:	**October 20th, 2024**
Date Revised:	**October 26th, 2024**

1. Executive Summary:

GatorTutor is a tutoring service created by San Francisco State University students for SFSU students. We aim to provide a place that facilitates the matching between all SFSU students with their best fit SFSU tutor. We feel that the dividing factor between a successful student and a struggling student is gaps in their knowledge and our goal is to bridge that gap. Our service helps match tutors with experience in those subjects and the students seeking assistance. Since our application is institution specific, our tutors are able to provide a level of service unique to SFSU that foreign services can not compete with.

Since our team is composed of SFSU students, we are able to target institution specific problem areas that's stopping students from achieving their potential through tutoring. Our application allows students to search for tutors by university specific classes. Professors don't always teach the same curriculum as others, so tutors are able to specify which ones they have experience with, allowing students to select them based on their issues. Tutors are able to create profiles unique between individuals through multimedia to customize their impressions, introduce themselves through text descriptions, show interested students their rates along with their availability, and appear on searches by course identifiers. Our platform also facilitates communication between interested students and tutors with our built-in messaging service.

2. List of main data items and entities:

Guest User - Mandatory

Entity Name: Guest User

Description: A general term for anyone who interacts with the system without an account.

Usage: They may create an account if they so choose to but without an account they are still able to browse the website, search for tutors, learn about the creators, and view tutors profiles.

Registered User - Mandatory

Entity Name: Registered User

Description: A user who has successfully registered on the platform.

Usage: Once a user has registered, they gain the ability to create a tutor post, customize their tutor profile, as well as message tutors they might be interested in meeting with.

Tutor Posts - Mandatory

Entity Name: Tutor Posts

Description: Posts created by registered users who offer tutoring services.

Usage: These posts contain details such as:

Subjects Offered: SFSU specific subjects and classes. i.e. “CSC-648”

Availability: Days of the week and hours of those days in which the tutors are regularly available to meet. i.e. “MON - 12pm-3pm, WED - 9am-1pm, SAT 2pm-5pm”

Hourly rate: A flat hourly rate which the tutor will charge the student for their services. i.e. “$16.25/HR”

Any user may search for a tutor and filter their results based off of these 3 categories in order to find the perfect tutor for them.

Admin - Mandatory

Entity Name: Admin

Description: An individual designated to the management of the platform and handling of database operations.

Usage: Admins do not interact with the platform as regular users do. They only use tools such as an SQL workbench to manage the system’s data. They handle tasks such as:

Tutor Post Approvals: Every post from a registered user regarding their tutoring skills needs to be reviewed through the workbench by an admin before that post can go live onto the website. If the post is not deemed appropriate to the websites standards, it will not be approved and be deleted from the database.

User Management: If a user submits an inappropriate tutor post (Nudity, Harassment, or anything that is irrelevant to the website's original purpose), the user will be banned from the website through the removal of their account from the database.

Admins do not have standard user profiles.

Message - Mandatory

Entity Name: Message

Description: A communication between registered users, typically between students and tutors.

Usage: Messages are exchanged through the platform for inquiries about tutoring services and general communication.

Subject - Mandatory

Entity Name: Subject
Description: A specific academic subject at SFSU.
Usage: Subjects are associated with tutor posts. Students search for tutors based on these subjects.

Reviews - Optional

Entity Name: Reviews
Description: Reviews for Tutors
Usage: Stores reviews for tutors including details about the reviewer and the content of the review. It helps provide feedback and ratings for tutors.

3. Functional Requirements - Prioritized:

Priority 1:

1.1 Unregistered/Registered Users - Browsing: Users shall be able to browse and easily navigate the website and most of what it has to offer.

1.2 Unregistered/Registered Users - Searching: Users shall be able to use the search bar in order to search for SFSU-specific classes and professors to find hyper-relevant material.

1.3 Registered Users - Creating Posts: Users shall be able to create a tutor profile and advertise their skills on the website.

1.4 Registered Users - Dashboard: Users shall have a functioning dashboard that displays their postings and messages received.

1.5 Registered User - Messaging: One way messaging system from student to tutor.

1.6 Admin - Deleting Inappropriate Content: Site Admin shall be able to delete inappropriate items or users based on the criteria of the post.

Priority 2:

2.2 Registered User - Login: User should be able to login into their account to see their dashboard

2.3 Admin - Approval of Tutor Posts: Site Admin shall be able to approve or deny any tutor post through the MySQL Workbench

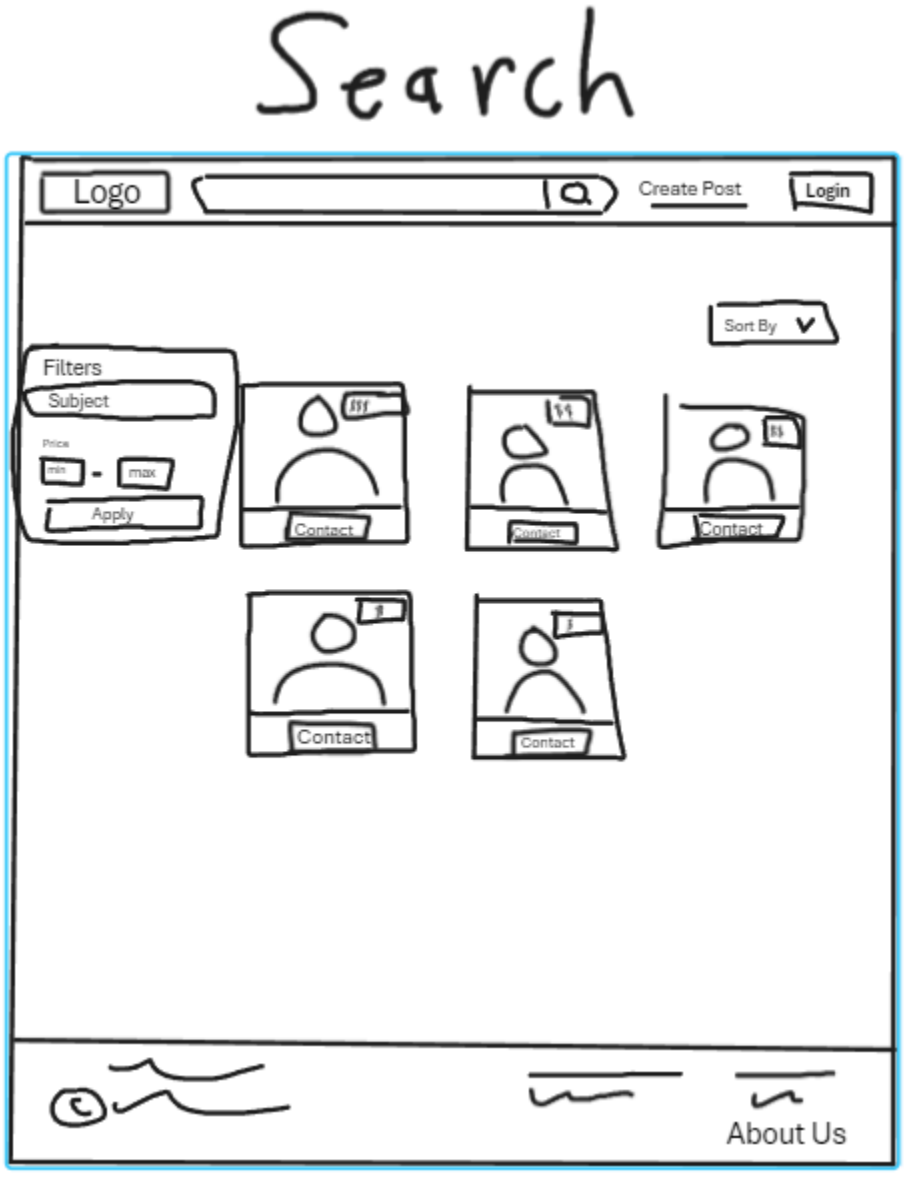
Priority 3:

3.1 Registered User - Interactive Chat: A two way messaging between tutor and student for real-time contact.

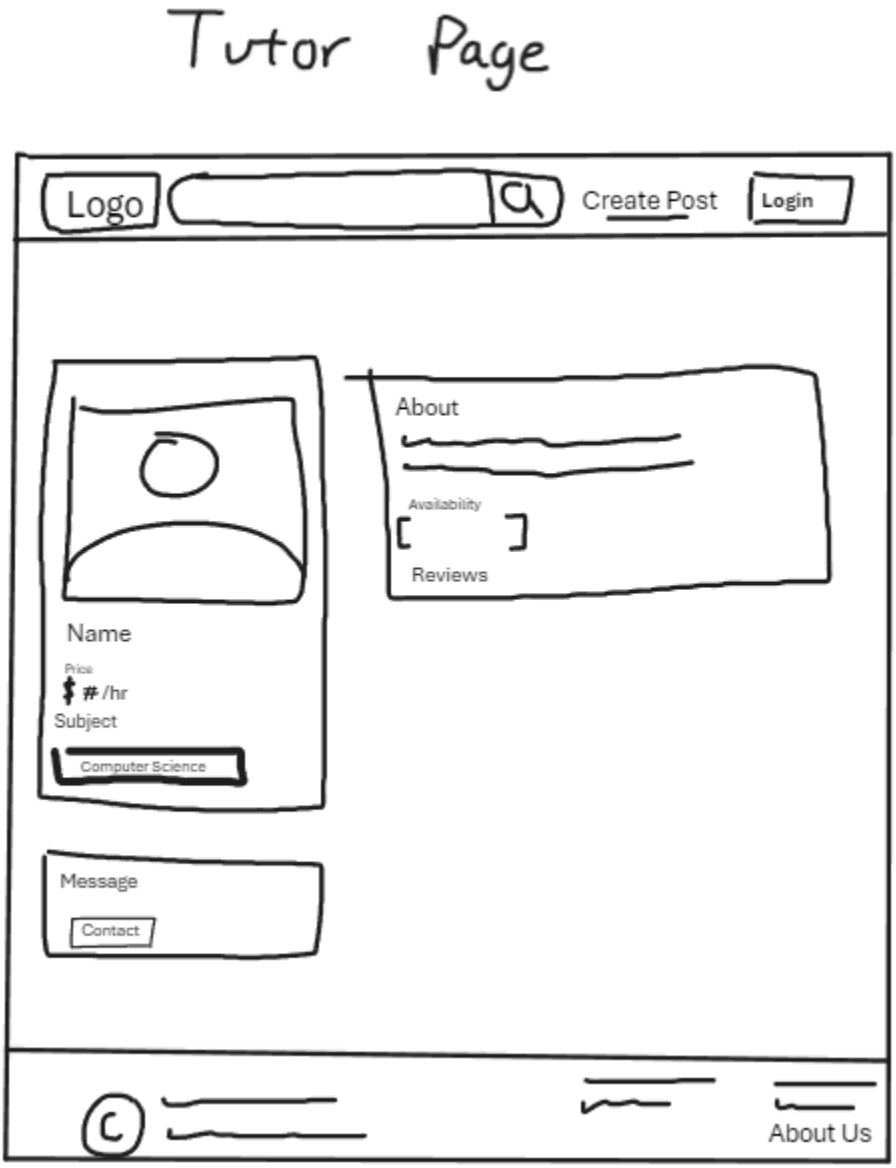
3.2 Student Reviews for Tutors: Students can leave detailed feedback and ratings for tutors, providing additional transparency and accountability.

4. UI Storyboards For Each Main Use Case:

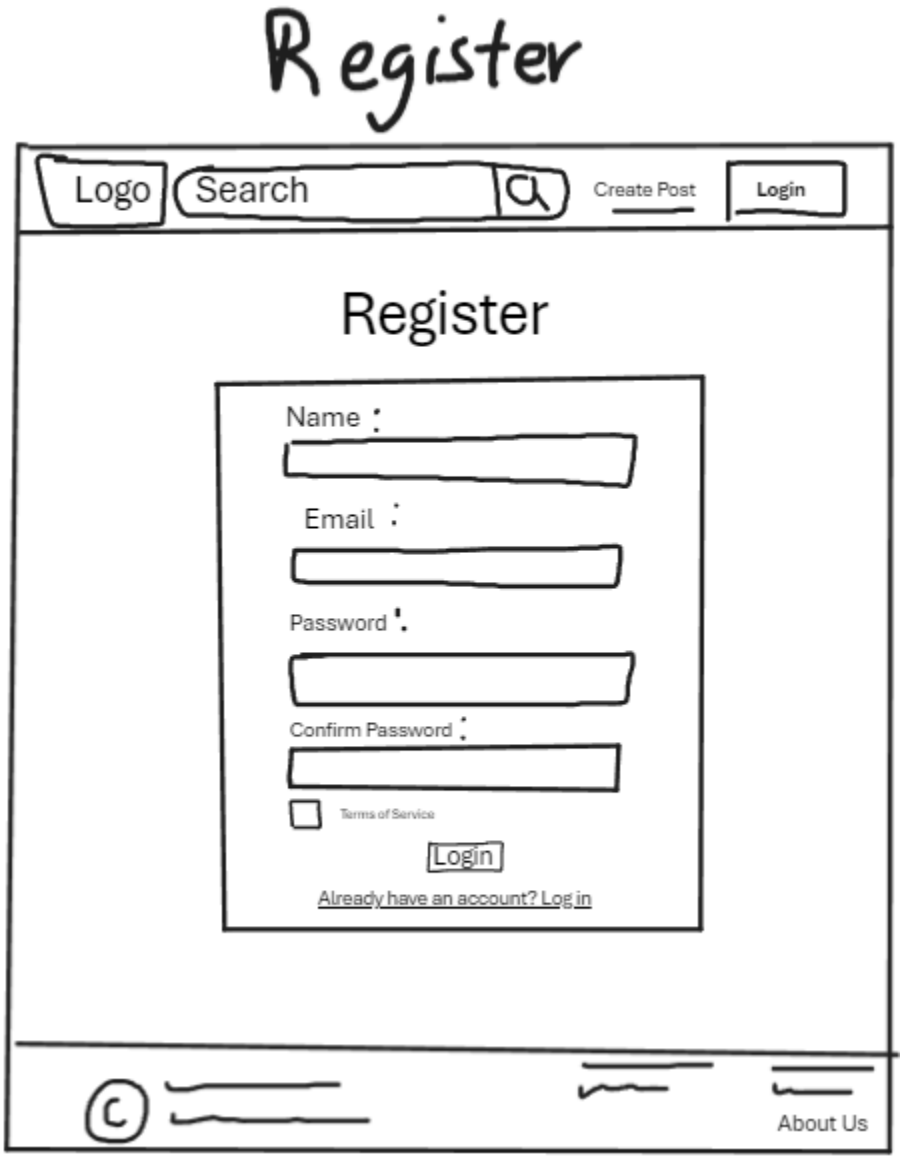
The search page includes all of the tutors' contact-cards on the page along with a filter box to the top left of all of the cards. In this box you can filter things like “Subjects Taught”, and “Price per Hour”. Once you find a tutor you want you can click on their contact card and be brought to the **Tutors page**.



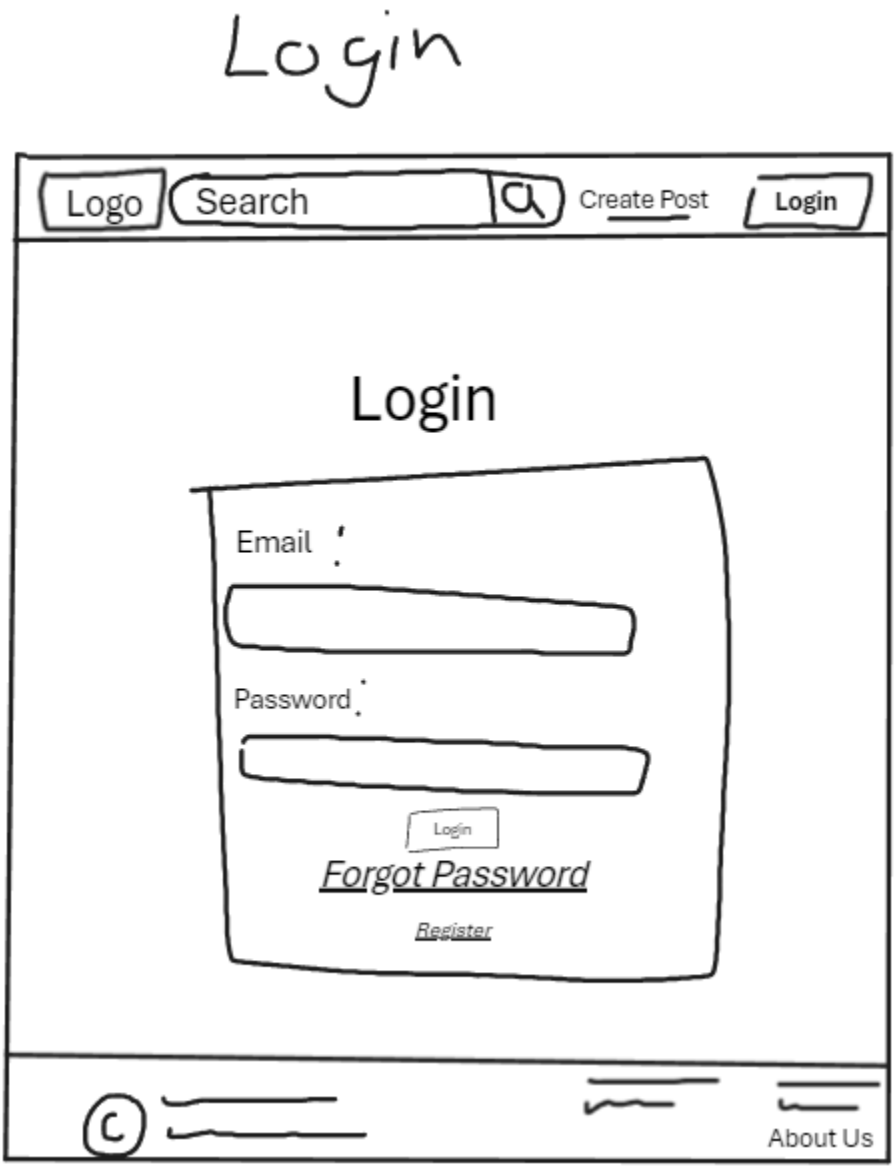
The tutor page includes information about the specific tutor that you would like to contact such as “Price Per Hour”, SFSU specific “Subjects Taught”, along with more personal information such as a description of themselves, a profile picture, and a video used as an example of the material they might teach. Before you can contact a tutor you must **register** for an account.



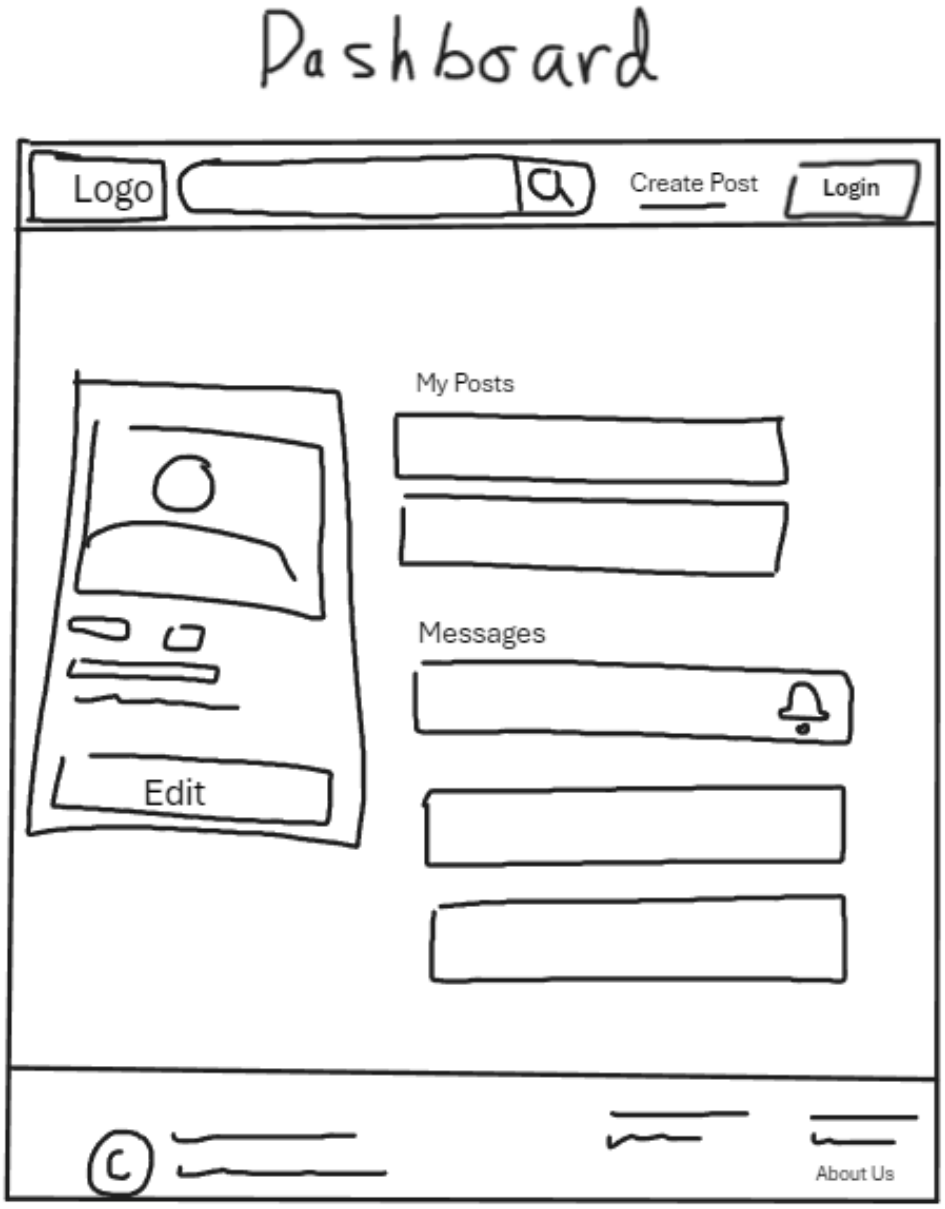
The register page includes a table with boxes where you can enter your Username , SFSU email, and Password, once you create your password you will need to confirm it by typing it in again. Once you have filled in these three fields you then can agree to the terms of service and create your account. If you already have an account you can use the "Already have an account?" link in order to navigate to the **login page**.



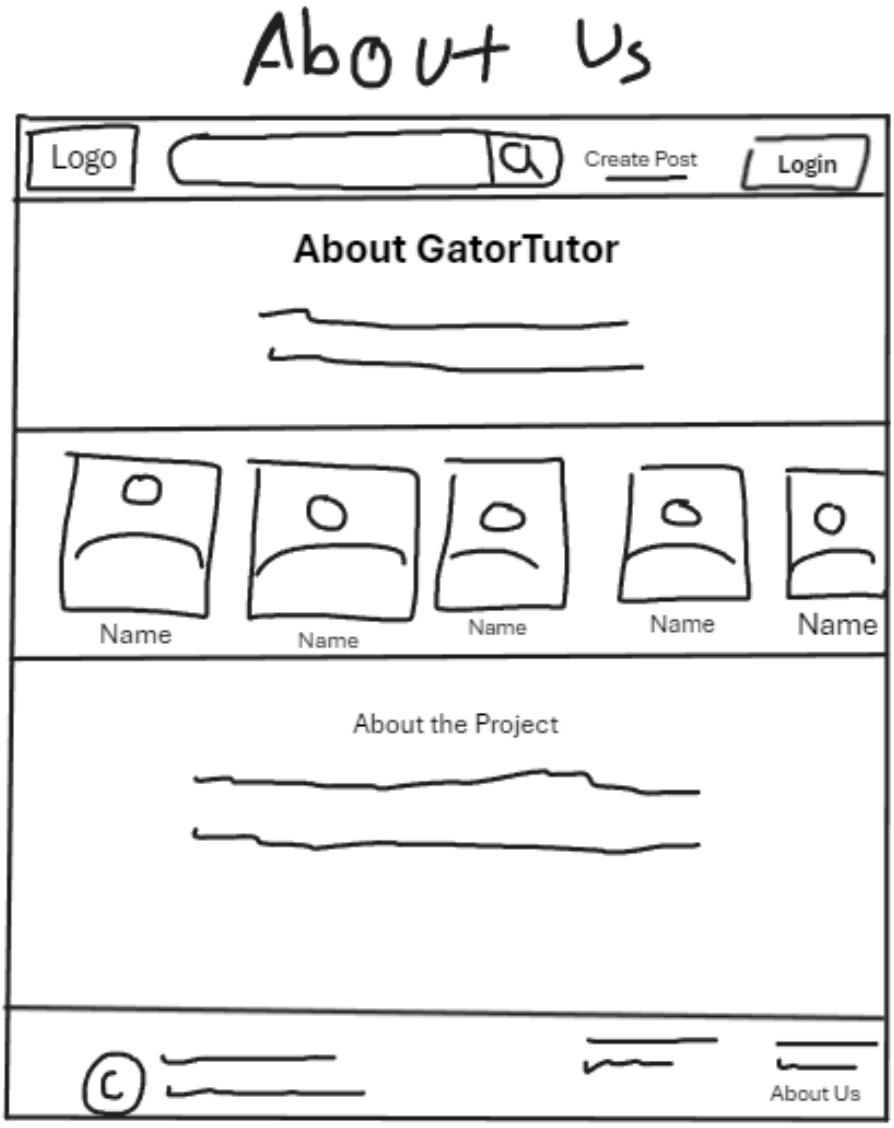
The login page is much like the registration page except for the fact that you don't have to type your password in twice like when you are creating an account. You just type in your username, password, and click "Login". There is also a link labeled "Register" that will redirect you to the registration page. There is also a link that you can click in the instance you forget your password. Once you login you will either be redirected to the page you were previously trying to reach or, by default you will be redirected to the **landing page**.



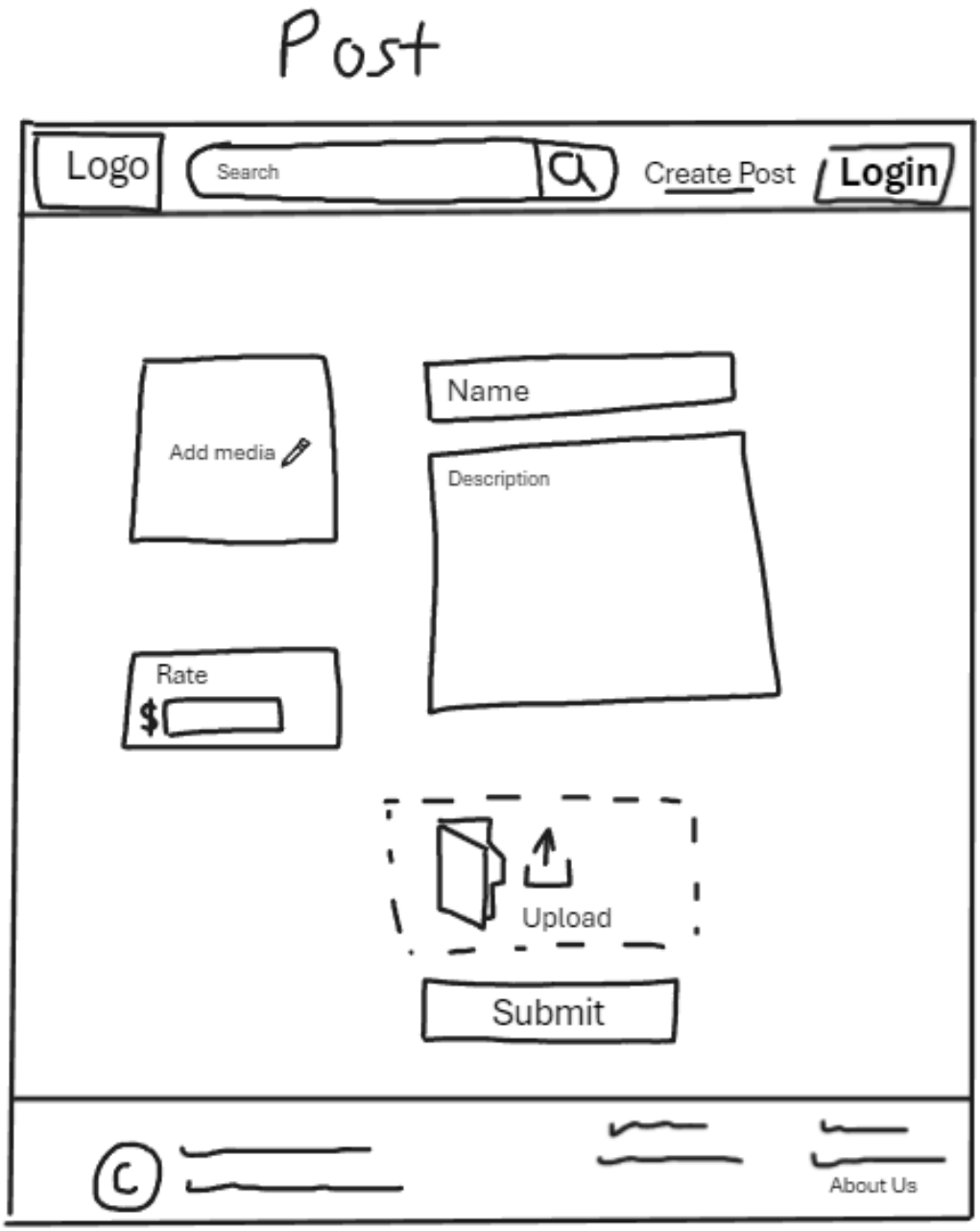
The Dashboard will display the students profile information as well as any messages they have been sent if they have created a tutor post. This page also contains cards that display posts that they created.



The About Us Page displays our creators of the website proudly, along with a description of each student and their role in the project.



The Post Page will include features that need to be filled in by the user in order to create a tutor post such as “Profile Picture”, “Name”, “Description”, “Price per Hour”, and “Video of Example Material”.



5. High level Architecture, Database Organization Summary:

MVC design pattern:
The core logic resides in the src directory, which includes all the MVC components and routes. The **Models** handle data and business logic, **Controllers** manage user requests. **Views** define the user interface. **Routes** link user actions to the appropriate controllers. Additionally, we have a **public** folder for static assets, including a uploads directory for media files. This setup allows us to serve media efficiently from the file system.

Mock File Structure:

```
project-root/
├── src/
│   ├── controllers/   # Handles user requests
│   ├── models/        # Manages data and logic
│   ├── views/         # Defines user interface
│   └── routes/        # Maps URLs to controllers
├── public/
│   ├── css/           # Stylesheets
│   ├── images/        # Static images
│   └── uploads/       # Media files
└── app.js             # Main application setup
```

DB organization:

Users	**Messages**	**Tutor Posts**
Stores user information like username and email.. It's essential for managing user accounts and authentication. id: Unique identifier for each user username: User's chosen name email: User's email address password: Encrypted password created_at: Timestamp of account creation last_login: Timestamp of last login updated_at: Timestamp of last update	Contains messages between users, including sender and recipient details. It enables communication on the platform. Messages will only be one directional as specified, however this approach allows for easy expansion of the messages feature in the future. id: Unique identifier for each message sender_id: ID of the user sending the message recipient_id: ID of the user receiving the message message: Content of the message created_at: Timestamp of when the message was sent read_at: Timestamp of when the message was read	Holds tutor profile details such as bio, availability, and rates. It helps tutors showcase their services. id: Unique identifier for each post user_id: ID of the tutor creating the post bio: Short biography of the tutor availability: Tutor's available times hourly_rate: Rate charged per hour contact_info: Tutor's contact information created_at: Timestamp of post creation updated_at: Timestamp of last update profile_photo: File for the tutor's profile photo profile_video: File for the tutor's profile video

		additional_images: Files for multiple images uploaded to the tutor's profile resume_pdf: File for the uploaded PDF
Subjects Lists all available subjects. It organizes the subjects that tutors can teach and students can learn. id: Unique identifier for each subject subject_name: Name of the subject	**Classes** Represents different classes linked to subjects. It helps categorize educational content. id: Unique identifier for each class class_name: Name of the class	**Tutor Subjects** Links tutors to the subjects they can teach. It ensures students find the right tutor for their needs. tutor_id: ID of the tutor subject_id: ID of the subject
Class Subjects Connects classes to their relevant subjects. It maintains an organized educational framework. class_id: ID of the class subject_id: ID of the subject	**Reviews** Stores reviews for tutors, including details about the reviewer and the content of the review. It helps provide feedback and ratings for tutors. id: Unique identifier for each review tutor_id: ID of the tutor being reviewed user_id: ID of the user writing the review rating: Numeric rating given by the user comment: Content of the review created_at: Timestamp of when the review was created	An **Appointments** table and its corresponding linker table could be useful if we decide to implement scheduling features, but it will not be a priority right now. It would help manage session bookings between tutors and students, storing details like date, time, and participants. If user demand for organized scheduling grows, this feature could become more relevant. (we know it is not in the scope of this project)

Search/filter architecture and implementation:
We will implement a search feature that allows users to find tutors based on specific criteria such as name, subject expertise, and hourly rate. This will be achieved by using a method that involves querying our database with filters applied to these criteria. The search will utilize SQL's **%LIKE%** operator to match partial text entries, ensuring that users can find relevant tutors even if they only remember part of a name or subject. Additionally, we will allow sorting of results by different parameters like price or recency, providing a flexible and user-friendly experience. This approach will ensure that users can easily navigate and find the right tutor for their needs.

The **%LIKE%** operator will be applied to text-based fields such as username, subject_name, class_name, short_bio, experience, availability allowing users to search for partial matches in names, descriptions, ect.

Non-trivial algorithms:
Dynamic Query Building for Tutor Search: We will have a system that dynamically constructs SQL queries based on user-provided filters and sorting options. This allows users to search for tutors using various criteria such as subjects, price range, and availability, providing a tailored search experience.

Tutor Profile Management: We will manage tutor profiles by handling complex data inputs, including availability, subjects, and multimedia uploads. This ensures that tutors can maintain comprehensive and up-to-date profiles, enhancing their visibility to potential students.

Subject and Tutor Association: We will implement a mechanism to associate tutors with specific subjects, allowing for efficient retrieval and display of tutors based on their expertise. This involves managing relationships between tutors and subjects in the database.

Recommendation System: We might develop a recommendation engine to suggest tutors to students based on their past interactions, preferences, and similar user behaviors. This would enhance user engagement by providing personalized tutor suggestions.

Dynamic Pricing Model: We could implement a dynamic pricing algorithm that adjusts tutor rates based on demand, time of day, or tutor experience. This would optimize both tutor earnings and student affordability, ensuring a balanced marketplace.

Changed or added SW tools and frameworks

- **Multer:** Our choice for handling file uploads, essential for multipart/form-data.
- **Express-Validator:** Keeps our input clean and secure, preventing SQL injection and XSS.
- **Express-Session:** Manages user sessions seamlessly, crucial for authentication.
- **Bcrypt:** Ensures password security, trusted by LinkedIn and Dropbox.
- **EJS:** Our templating engine for dynamic HTML, simple and effective.
- **TailwindCSS**: We love its utility-first approach for quick, responsive UI building.
- **Autoprefixer:** Saves us time by adding vendor prefixes for cross-browser CSS.
- **PostCSS:** Automates CSS tasks like linting and minifying, streamlining our workflow.

Development Tools

Nodemon + Browser-Sync: Nodemon automatically restarts our server on file changes, while Browser-Sync refreshes the browser, making development faster and more efficient.

Docker

Docker allows us to containerize our application, making it easy to run consistently on AWS servers. It simplifies sharing our environment across the team, ensuring everyone works with the same setup, reducing "it works on my machine" issues.

6. Key Risks:

In order to be proactive and avoid future Skill and Schedule risks our team is aiming to achieve a "Minimal Viable Product (MVP)". This will ensure that despite any challenges we may face, we have the skills to, and will be on schedule to turn in a finished project by mid-December. If we happen to finish our product earlier than expected, then and ONLY THEN will we attempt to implement further features to enhance the product.

Skills Risks

Q: What if a team member lacks the knowledge or skills to accomplish their task?

A: Our team agreed on a "no shame" environment where everyone is encouraged to ask each other questions, no matter how obvious they may be. There are no bad questions.

Q: What if a team member is not happy with the task they are assigned with?

A: The team allows for flexibility. Team members are allowed to switch tasks with another person if they are uncomfortable. Everyone should be competent in the task they were assigned.

Schedule Risks

Q: What if a team member is unable to finish their task on time?

A: Our team will help accommodate another member by having others help work on that specific task. This will be discussed through our Discord server or even through direct messaging in more urgent situations.

Q: What if an idea is too ambitious to finish?

A: The team has to be reminded that the goal is to push out a project that meets the minimum requirements. If the idea does not directly contribute to completing the main goal, it shouldn't be included.

Technical Risks

Q: Are there any unknown technical challenges that could affect progress?
A: Unknown technical challenges like integration issues, unexpected bugs, or tool limitations (e.g., with Trello or Discord) could impact progress.
Q: What if a team member makes changes or additions to the code that others are confused about?

A: Code should be well-documented using comments and a member should explain their code if others need explanation.

Teamwork Risks

Q: What if a specific team member is not communicating or being active on the project?

A: Our team will first reach out to the team member directly. Continued silence will lead the team to address the professor directly about the situation.

Q: What if the entire team lacks communication with each other?

A: Weekly online and in-person meetings are required every week to ensure everyone stays updated on the current progress of the project.

Legal/Content Risks

Q: What if copyrighted images are used?

A: All material needs to be reviewed by the team to ensure that every image is original or royalty-free.

7. Project Management:

This workflow generally represents the team's process, from milestone discussions to task assignments, progress check-ins, and tracking using Trello.

[Discussion] --> [Assign Tasks] --> [Task Details (Discord)] --> [Check-ins] --> [Trello Updates]

For each milestone, our team will discuss the various tasks each milestone contains to ensure that everyone understands what needs to be done and how we can accomplish it.

After discussion, our team lead will assign us the task that he believes will best fit our roles and skills. The tasks will be divided up in such a way where everyone will have a similar workload. Each team member is allowed to give input on which task they would prefer before and after the roles have been assigned.

Once everyone has agreed on the task they have been assigned, our team lead will then create documents for each member highlighting each specific task in more detail. These documents are then sent through discord with a deadline to get them in by.

Each week, our team will decide on a day to have a weekly discord meeting. This meeting acts as a mid-point check in for all of us to discuss our current progress and what else needs to be done to complete the assignment.

However, constant communication about the milestone progress is also maintained in between meetings through the discord text channel. The team lead will occasionally bring up progress checks and any other updates relating to the completion of the current milestone. More meetings are added if required for the milestone.

This system has allowed our front-end and back-end teams to maintain a solid form of communication. In order to further improve the efficiency of managing the project, we are incorporating the usage of Trello for this and future milestones.

With the use of Trello, we will have an easier time keeping track of task deadlines and updates. Instead of asking or looking back at the discord server for status updates, we are able to view these updates directly on the platform.

8. Use of genAI tools like ChatGPT and copilot:

Our team used ChatGPT 4o, the latest free model from openAI as our main Generative AI tool.

Our team used GenAI in order to:

- Figure out how to format our main data items and prioritize them based on the System the class CEO gave to us - LOW, you will find a trend in this section regarding AI and its abilities when it comes to workloads that are highly-tailored to a specific party. Since GenAI takes advantage of so many works across the internet, the answers are very general and well-balanced which can be a good thing but is not necessarily helpful when we are working in an environment that only WE know. For example, WE know that a search feature, although useful, is not the #1 priority when it comes to developing a tutoring website but chatGPT may see it as a high priority because it is trained on data all around the internet that might see a search feature as a #1 priority. For this reason, genAI is not great with prioritizing features.
- Divide up the workload of milestone 2 evenly between group members as well as assign relevant topics to the same group members who might have worked with them before - HIGH, as an experiment, I fed chatGPT the entire Milestone 2 instruction document that was provided to us by the class CEO. I prompted it to read all requirements and then divide the workload into even sections. It did this surprisingly well and even remembered which sections I gave my group members in Milestone 1 and matched some of the same topics up to the same students. This really impressed me and went beyond what I expected.

- Give us an idea of how to prioritize our high-functional requirements - LOW, once again, having genAI prioritize information that only WE really know how to prioritize is futile and can give you some idea of how to structure the section, but ultimately will not be accurate to what is expected when we turn in our document.
- Help us understand what "key risks" in a working environment might look like as well as what they would look like unique to our project - MEDIUM, I labeled the efficiency and accuracy of this task as medium because although it gave some good examples of what some key risks might look like in a working environment, it wasn't able to go the extra step and customize the potential risks to our situation of being students, emulating a working environment, working on a group project together,.
- Structure our project management section - HIGH, Our group as a whole didn't really have a solidified idea of what we wanted for our "Project Management" section. Using GenAI and re-reading the original document helped us understand what we were expected to do. This was one of the more simple requirements but it was very important that we include good, relevant information of how we are maintaining our group's organization and lines of communication, which will in turn result in a great end-product for our team.
- After viewing another team's use of chatGPT to create what a SFSU specific tutoring platform might look like, I took the liberty of prompting it similarly and it spit out this photo:
- 

- We actually enjoyed the look of this layout so much we took a good amount of inspiration for our own website

9. Team Lead Checklist:

- So far all team members are fully engaged and attending team sessions when required **DONE/OK**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing **DONE/OK**
- Team reviewed suggested resources before drafting Milestone 2 **DONE/OK**
- Team lead checked Milestone 2 document for quality, completeness, formatting and compliance with instructions before the submission **DONE/OK**
- Team lead ensured that all team members read the final Milestone 2 document and agree/understand it before submission **DONE/OK**
- Team shared and discussed experience with genAI tools among themselves **DONE/OK**

SW Engineering CSC 648-848 Fall 2024

GatorTutor

Team 6

Team Lead, Alex Hoff - (ahoff2@mail.sfsu.edu)
Backend Lead 1, Dylan Faulder - (dfaulder@mail.sfsu.edu)
Backend Lead 2, Austin Ng - (Ang@sfsu.edu)
FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)
Github Master, Jack Richards - (jrichards7@sfsu.edu)

Milestone 3 - Meeting Review

History

Date Submitted:	**November 15th, 2024**

Summary of Milestone 3 meeting review with Prof. Petkovic and plans for further development

Team number: 6
Meeting date: 11/13/24

- Summary of feedback on UI (record all pages that need revision)
    - Instead of displaying “Create Tutor Post”, display “Become a Tutor”, more clear and interactive
    - Keep buttons that do the same thing consistent in what they are labelled
    - Don’t have two different buttons for search, just use search bar
    - For the tutor post form have it only be one column
    - Sort the form by importance from top down, put a red star next to mandatory information
    - Have a part of the form where you can enter the title of the post
    - Change bio to comments
    - Include cancel button within tutor form
    - Have button to open tutor page read “More Info” instead of “View Profile”
    - Include the “Send Message” button on the search page where tutors are listed, not just when you open an individual profile
    - Include a “Found x results” under the search bar
    - Change “Create Your Tutor Profile” title at the top of the tutor form to “Become a Tutor”
    - On the search page show how many results and how it is being sorted, have a sorting feature by price, dates available, classes, etc.
    - Messages received should have date, who sent them, and which tutor post of yours they message you from
- Summary of feedback on code and architecture
    - Search functionality works
    - Layout of search page needs work as filter panel is off screen
    - Nav bar is not responsive enough to where a horizontal bar is triggered
    - Some of the button text in the nav bar is too verbose
    - Short simple words that are clear should be used
- Summary of feedback on github usage
    - Commits look good
    - Code looks clean and well described
    - Commits Unbalanced
- Summary of feedback on DB
    - No feedback
- Summary of feedback on teamwork and risk management
    - Teamwork is good, all team members are participating

- Confirm that you have done architecture review to check that developers adhere to MVC pattern, coding style, minimal agreed documentation etc. Record if OK or list the issues found. Request developers follow up on corrections and follow up later by doing code reviews
    - This has been OK
- List below agreed upon P1 list of features for final delivery which constitute product plan.
    - Search
    - Create Tutor Post
    - Login
    - Registration
    - Message
- Any other comments and issues
    - N/A

SW Engineering CSC 648-848 Fall 2024

GatorTutor

Team 6

Team Lead, Alex Hoff - (ahoff2@mail.sfsu.edu)
Backend Lead 1, Dylan Faulder - (dfaulder@mail.sfsu.edu)
Backend Lead 2, Austin Ng - (Ang@sfsu.edu)
FrontEnd Lead, Dalan Choy - (dchoy3@mail.sfsu.edu)
Github Master, Jack Richards - (jrichards7@sfsu.edu)

Milestone 4

History

Date Submitted:	**December 12th, 2024**

Product Summary

Product Name and Description: "*GatorTutor*"

A site for San Francisco State University students to find academic help from other students who've been in their position. They are able to find and search for tutors that specialize in the courses they need assistance in.

GatorTutor is a tutoring service created by San Francisco State University students for SFSU students. We aim to provide a place that facilitates the matching between all SFSU students with their best fit SFSU tutor. We feel that the dividing factor between a successful student and a struggling student is gaps in their knowledge and our goal is to bridge that gap. Our service helps match tutors with experience in those subjects and the students seeking assistance. Since our application is institution specific, our tutors are able to provide a level of service unique to SFSU that foreign services can not compete with.

Itemized List of P1 Functions:

Browsing: Users shall be able to browse and easily navigate the website and most of what it has to offer.
Searching: Users shall be able to use the search bar in order to search for SFSU-specific classes and professors to find hyper-relevant material.
Posting: Users shall be able to create a tutor profile and advertise their skills on the website.
Dashboard: Users shall have a functioning dashboard that displays their postings and messages received.
Messaging: One way messaging system from student to tutor.
Approving Post: Site Admin shall be able to delete inappropriate items or users based on the criteria of the post.
Deployment URL:
https://gatortutor.net/

Usability Test

1. **Test Objectives**:
   The objective of this usability test is to evaluate how easily users can create a tutor post on GatorTutor. This function is critical to the platform's success, as it enables tutors to share their information and students to find suitable tutors. Ensuring a seamless posting experience is essential for maintaining the platform's functionality and user satisfaction.

2. **Test Background and Setup**:

System Setup
A standard personal computer with a modern operating system capable of running web browsers such as Google Chrome and Mozilla Firefox. No specialized hardware or software is required beyond these browsers and a stable internet connection to access the specified URL.

Intended Users
GatorTutor is intended for students and tutors with varying levels of experience in navigating online with a browser. This means that users with the minimal experience should still be able to navigate through the website without any difficulties. The navigation should feel intuitive to people of all levels of experience.

URL of the System
Testing of tutor post creation functions on GatorTutor will be conducted on https://gatortutor.net/

Test Environment
The test will take place in a home setting with no cameras present. Additionally, training will not be done because GatorTutor aims to have a simplistic navigation experience.

3. **Usability Task Description**:
    1. Navigate to Create Tutor Post section
    2. Fill out the Basic Information section with the following details
        a. Set the Name to "John Doe"
        b. Set the Subject to "Finance"
        c. Set the Hourly Rate to "$20"
        d. Set Contact Information to "test@email.com"
        e. Set Bio to "I successfully completed and filled out the bio section."
    3. Set the availability to Monday and Friday
    4. Upload a given sample MP4, PDF CV, and PNG profile picture in the Additional Documents section.
    5. Submit the post

4. **Plan for Evaluation of Effectiveness**:
Effectiveness will be determined by the number of people who were able to complete the task within a specific amount of time. This will be compared to the number of people who weren't able to successfully complete the task. The second observation that will be made is looking at each field to see if it was filled out correctly. The fields must be filled out exactly as the task description mentions and must not be missing. Lastly, viewing the

post to observe if it was submitted and uploaded successfully will be taken into consideration.

5. **Plan for Evaluation of Efficiency**:
   Efficiency will be determined by two main factors. The first factor will be how long it took to complete the task. The tester's skill level will be taken into consideration when observing the completion time. The second factor that will be observed is the number of clicks it took to create and submit the post.

Major Assets & Their Protection

- User Credentials
  - Threats: Password theft, unauthorized access
  - Protection: Passwords encrypted using bcrypt with salt rounds of 10, secure HTTP-only cookies for sessions
- User Personal Data (SFSU emails, profiles)
  - Threats: Data breach, identity theft
  - Protection: Data access restricted through authentication middleware, sensitive data filtered from API responses
- Tutor-Student Messages
  - Threats: Unauthorized message access, impersonation
  - Protection: Message access verified against user session, sender/recipient validation on server

Passwords are encrypted in database using bcrypt

```
const hashedPassword = await bcrypt.hash(data.password, 10);
```

Search Bar Input (api also restricts > 40 char)

```
maxLength={40}
```

SFSU Email Validation

```
email: z.string()
  .email("Invalid email format")
  .endsWith("@sfsu.edu", "Must be an SFSU email address")
```

Terms Acceptance

```
acceptTerms: z.boolean().refine((val) => val === true, {
  message: "You must accept the terms and conditions"
})
```

6. **Plan for Evaluation of User Satisfaction**:
    1. Creating a tutor post was a straightforward process.
       1) Strongly Disagree 2) Disagree 3) Neutral 4) Agree 5) Strongly Agree

    2. The fields were easy to understand and fill out.
       1) Strongly Disagree 2) Disagree 3) Neutral 4) Agree 5) Strongly Agree

    3. I am satisfied with the experience of creating a tutor post.
       1) Strongly Disagree 2) Disagree 3) Neutral 4) Agree 5) Strongly Agree

QA Test Plan and QA Testing

1. Test Objective,

    The Goal of these tests are to verify we have a working create tutor post functions.

2. Specify all hardware and software requirements, including the URL

    Basic hardware capability, enough to give access to basic web browser abilities.

https://gatortutor.net/

3. Feature to be tested

    The create tutor post function, along with working Title, Name,Hourly Rate, Bio and Availability.

4. QA Test plan Table

Test #	1	2	3
Test Title	Test of successful tutor post creation	Test of successful tutor post creation	Test of successful tutor post creation
Test Description	Testing if when given input that the create tutor post functions successfully create a working tutor	Testing if when given input that the create tutor post functions successfully create a working tutor post	Testing if when given input that the create tutor post functions successfully create a working tutor post visible to users.

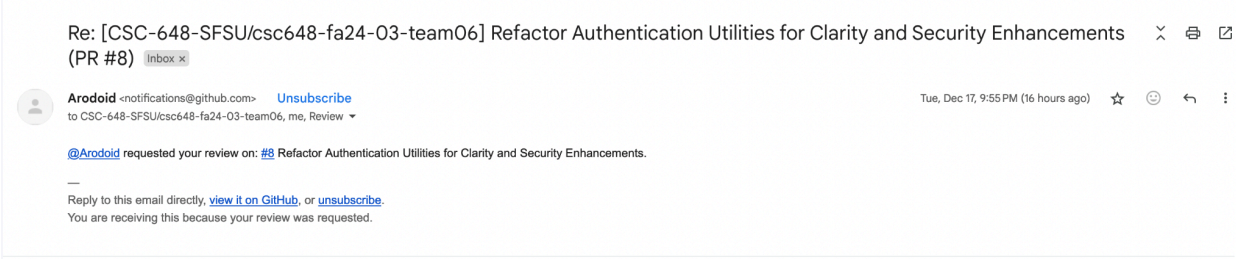
	post visible to users.	visible to users.	
Test Input	Title: Native Japanese tutor Hello I am a Native Japanese Speaker who specializes in teaching Japanese to students Name: Test Name #1 Hourly Rate: 21.00 Subject: Japanese Bio: Hello I am a Native Japanese Speaker who specializes in teaching Japanese to students Availability: Monday Tuesday Wednesday	Title: Highly Skilled Computer Scientist Name: Test Name #2 Hourly Rate: 100 Subject: Computer Science Bio: This is a test for checking if the bio will produce exactly what is given 18001222 Availability: Monday Tuesday Wednesday Thursday Friday Saturday Sunday	Title: 123456789 Name: Test Name #3 Hourly Rate: 1 Subject: Art History Bio: Art History is a interesting major that expands one's knowledge of art throughout history Availability: Saturday Sunday
Expected Output	Title: Native Japanese tutor Hello I am a Native Japanese Speaker who specializes in teaching Japanese to students Name: Test Name #1 Hourly Rate: 21.00 Subject: Japanese Bio: Hello I am a Native Japanese Speaker who specializes in teaching Japanese to students Availability: Monday Tuesday Wednesday	Title: Highly Skilled Computer Scientist Name: Test Name #2 Hourly Rate: 100 Subject: Computer Science Bio: This is a test for checking if the bio will produce exactly what is given 18001222 Availability: Monday Tuesday Wednesday Thursday Friday Saturday Sunday	Title: 123456789 Name: Test Name #3 Hourly Rate: 1 Subject: Art History Bio: Art History is a interesting major that expands one's knowledge of art throughout history Availability: Saturday Sunday
Test results:	Correct output given	Correct output given	Correct output given

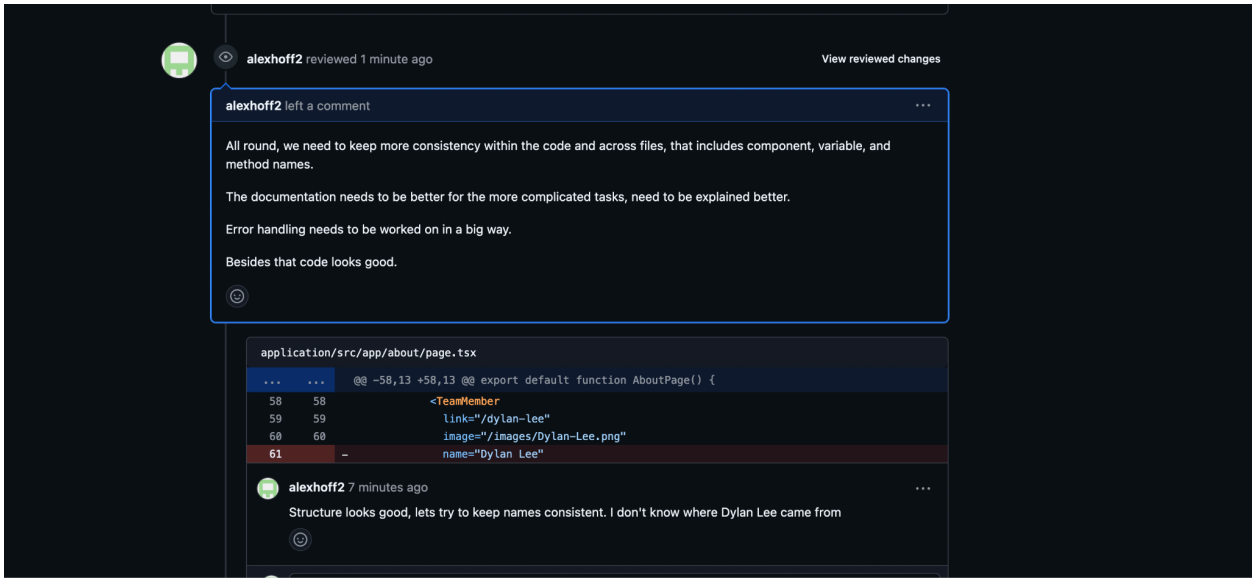
Browser Test	Chrome: Successful:PASS Firefox: Successful:PASS	Chrome: Successful:PASS Firefox: Successful:PASS	Chrome: Successful:PASS Firefox: Successful:PASS

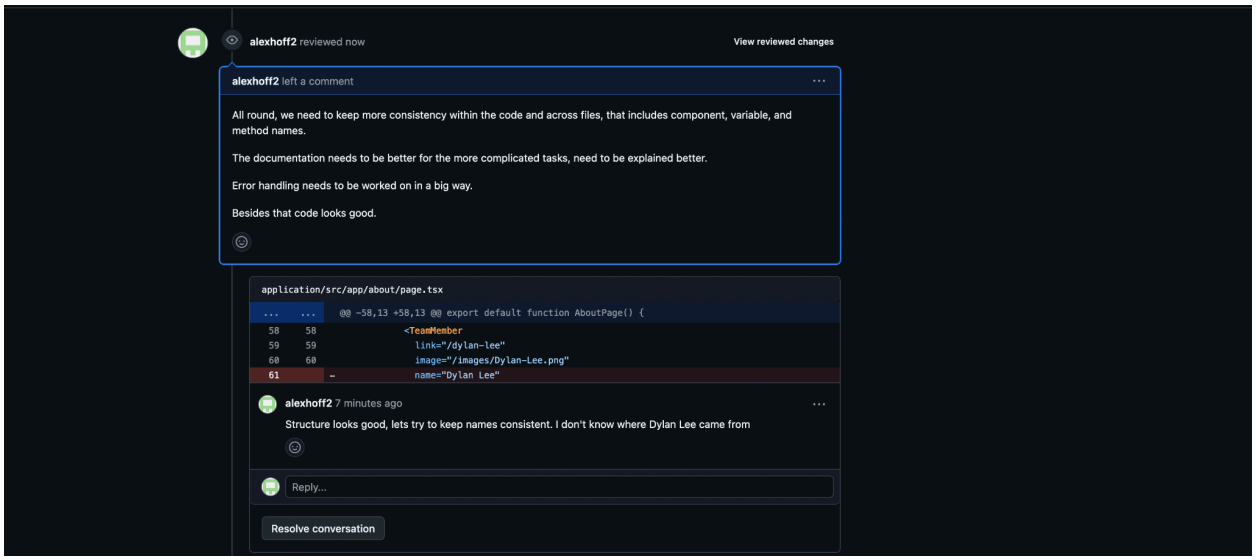
5. Perform tests on two different browsers
Brave (Chromium) and Firefox

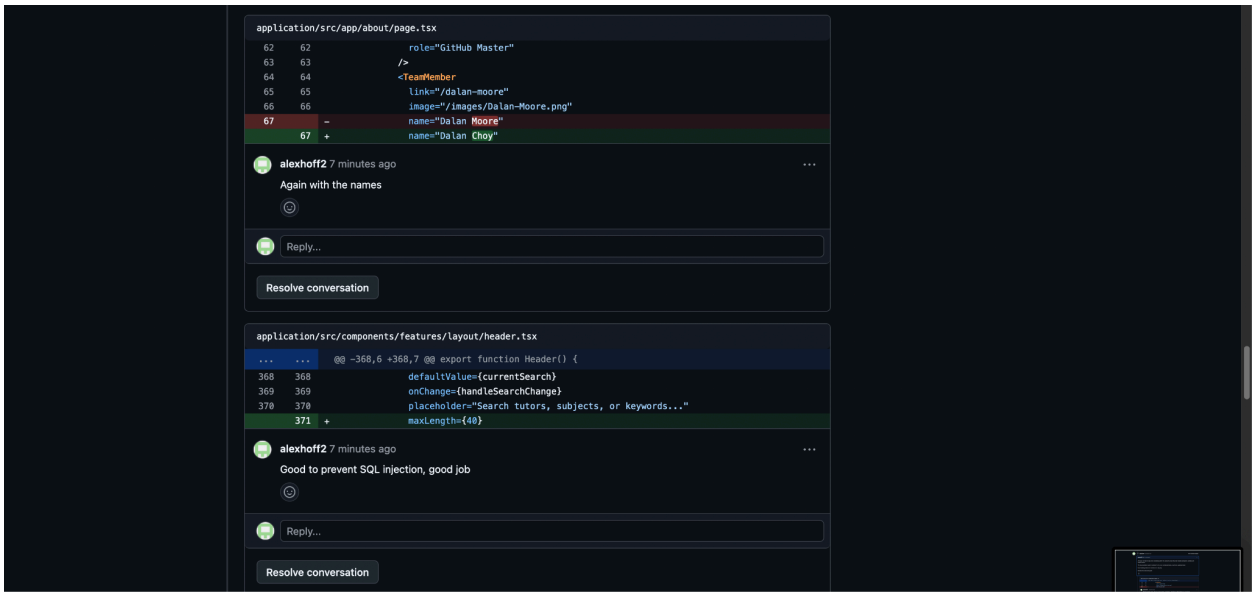
Peer Review Code

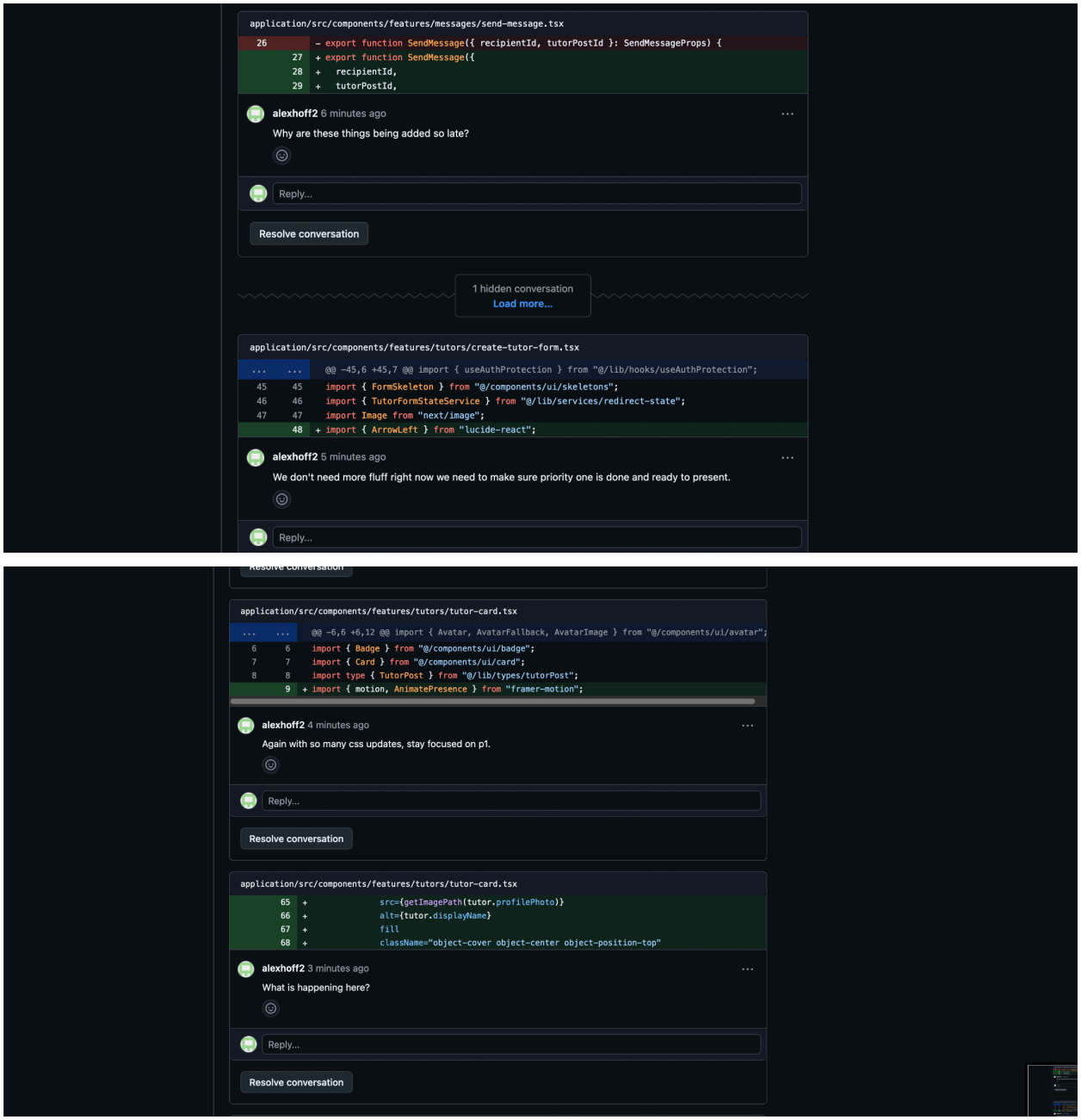
This peer review was done with Dylan's and Jack's code reviewed by Alex.




alexhoff2 reviewed 1 minute ago
View reviewed changes
alexhoff2 left a comment
All round, we need to keep more consistency within the code and across files, that includes component, variable, and method names.
The documentation needs to be better for the more complicated tasks, need to be explained better.
Error handling needs to be worked on in a big way.
Besides that code looks good.
application/src/app/about/page.tsx
@@ -58,13 +58,13 @@ export default function AboutPage() {
58 58 <TeamMember
59 59 link="/dylan-lee"
60 60 image="/images/Dylan-Lee.png"
61 - name="Dylan Lee"
alexhoff2 7 minutes ago
Structure looks good, lets try to keep names consistent. I don't know where Dylan Lee came from


alexhoff2 reviewed now
View reviewed changes
alexhoff2 left a comment
All round, we need to keep more consistency within the code and across files, that includes component, variable, and method names.
The documentation needs to be better for the more complicated tasks, need to be explained better.
Error handling needs to be worked on in a big way.
Besides that code looks good.
application/src/app/about/page.tsx
@@ -58,13 +58,13 @@ export default function AboutPage() {
58 58 <TeamMember
59 59 link="/dylan-lee"
60 60 image="/images/Dylan-Lee.png"
61 - name="Dylan Lee"
alexhoff2 7 minutes ago
Structure looks good, lets try to keep names consistent. I don't know where Dylan Lee came from
Reply...
Resolve conversation


application/src/app/about/page.tsx
role="GitHub Master"
/>
<TeamMember
link="/dalan-moore"
image="/images/Dalan-Moore.png"
- name="Dalan Moore"
+ name="Dalan Choy"
alexhoff2 7 minutes ago
Again with the names
Reply...
Resolve conversation
application/src/components/features/layout/header.tsx
@@ -368,6 +368,7 @@ export function Header() {
defaultValue={currentSearch}
onChange={handleSearchChange}
placeholder="Search tutors, subjects, or keywords..."
+ maxLength={40}
alexhoff2 7 minutes ago
Good to prevent SQL injection, good job
Reply...
Resolve conversation



Non-Functional Specs Self-Check

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 - **IN PROGRESS**

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 - **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. (Important so as to not confuse this with a real application). - **IN PROGRESS**

GenAI Usage

***CHATGPT-4o mini* WAS USED FOR ALL TASKS LISTED BELOW**

Code Review Assistance
Usefulness: High
How We Used It: ChatGPT was used to assist in conducting code reviews by generating suggestions for improving variable naming consistency, function comments, and error-handling checks. For example, when reviewing our messaging feature, ChatGPT suggested adding input validation and improving feedback messages for failed message sends.

Prompt Example:"Review this JavaScript function for potential bugs and suggest improvements for better readability and performance."

Bug Identification & Debugging
Usefulness: Medium
How We Used It: GitHub Copilot was used during the debugging phase to detect logical issues. It helped catch edge cases in our availability filtering system and suggested fixes for incomplete SQL queries in our Prisma models.

Prompt Example:"Fix SQL query that fetches tutor availability from the database based on matching criteria."

UI/UX Recommendations
Usefulness: Medium
How We Used It: ChatGPT provided recommendations for improving usability in our Create Tutor Post form. It suggested adding placeholder text and form validation messages to enhance user experience.

Prompt Example:"Suggest improvements for a React form collecting tutor registration data."

Documentation Assistance
Usefulness: High

How We Used It: ChatGPT helped draft clear, concise comments for complex functions, such as the tutor card rendering logic. This saved time in documenting intricate logic in our codebase.

Prompt Example:"Explain how this React component renders tutor profiles with interactive effects."

Usability Testing Support
Usefulness: Low
How We Used It: ChatGPT was consulted for creating usability test scenarios and evaluation metrics. While some suggestions were relevant, others were too generic for our specific needs.

Prompt Example:"Generate usability testing scenarios for an online tutor-matching platform."

Additional Insights:

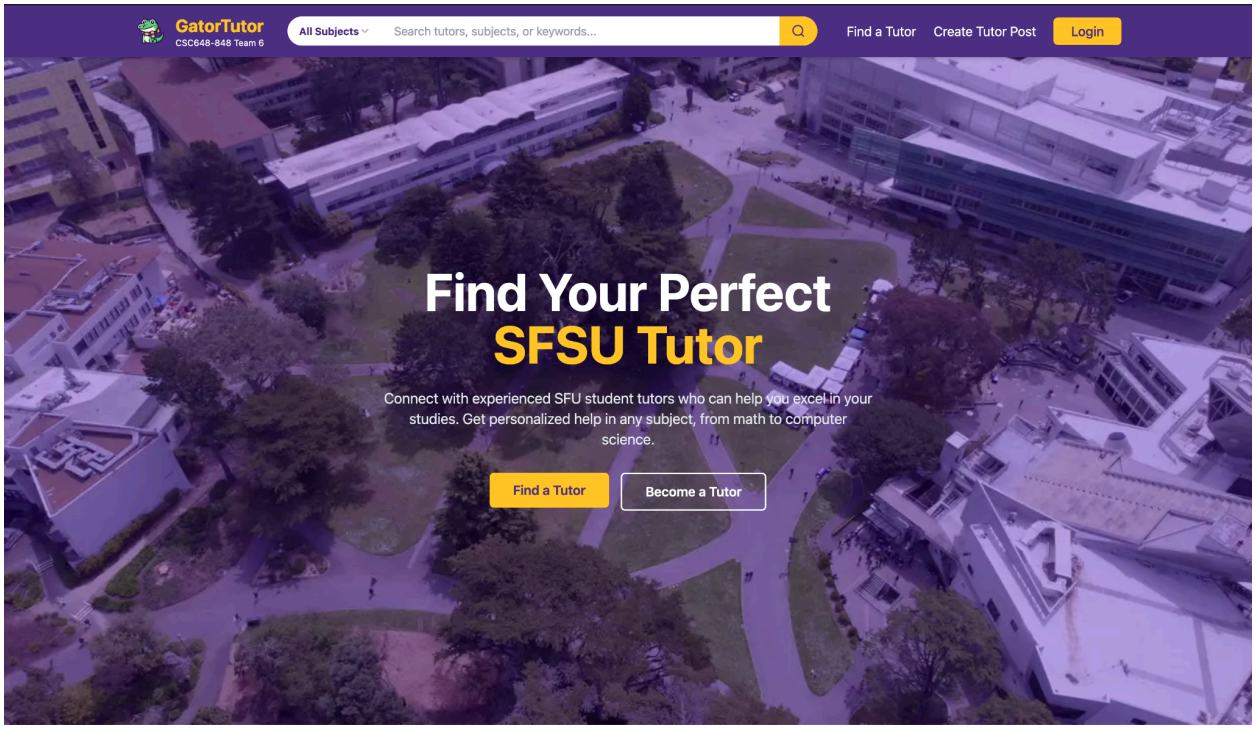
Best Use Cases: We found ChatGPT particularly useful for code reviews and generating documentation, which helped maintain high development standards.

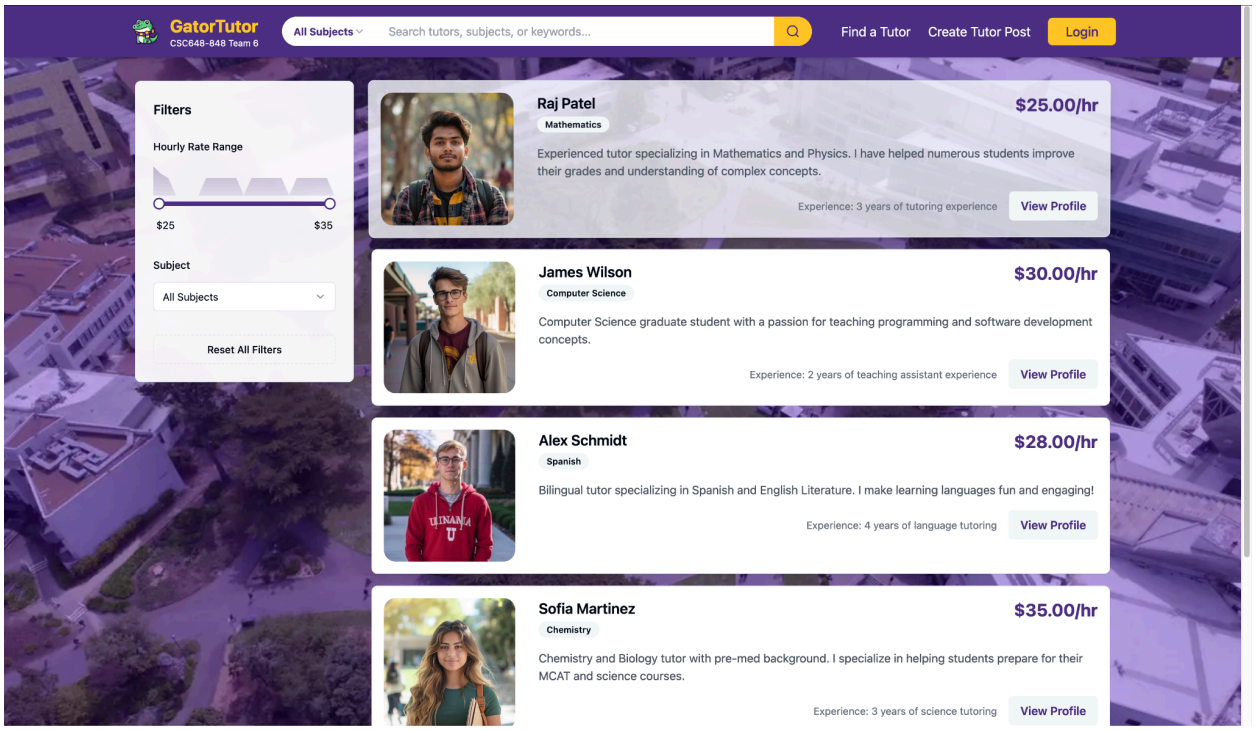
Challenges: Some prompts required rephrasing to receive accurate responses, and specific technical queries occasionally yielded incomplete answers.

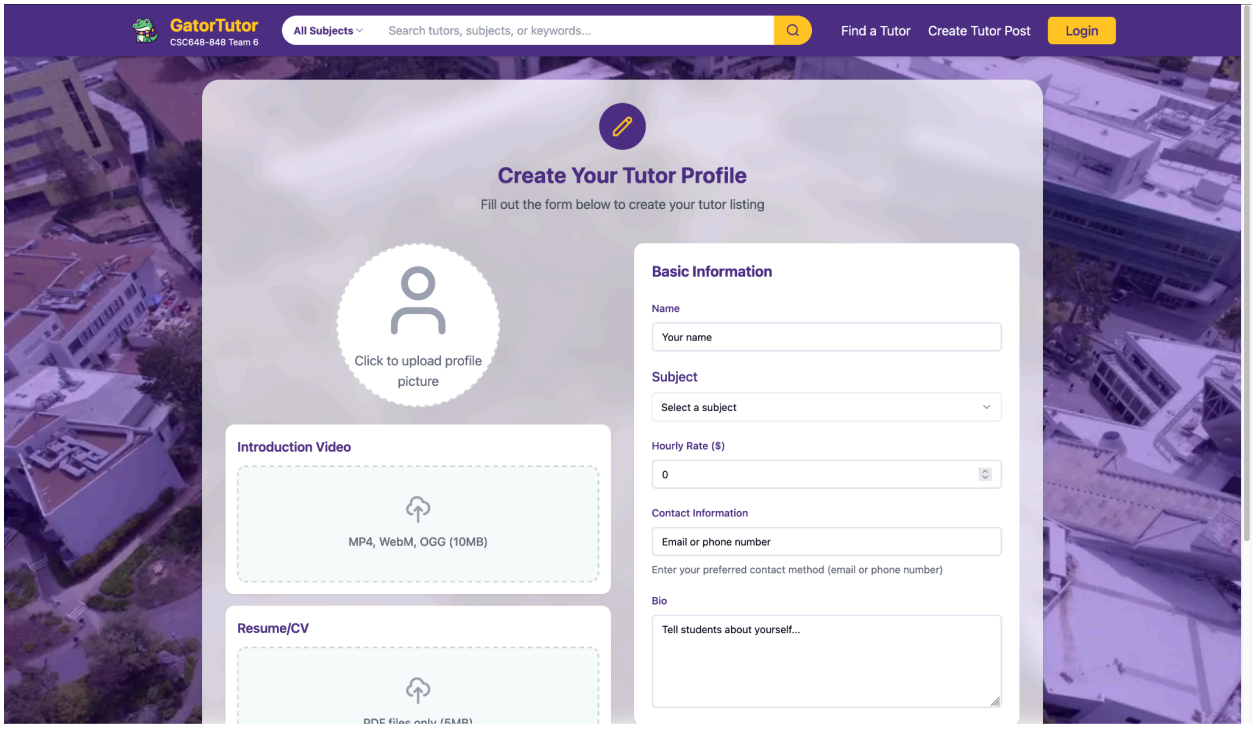
Conclusion:

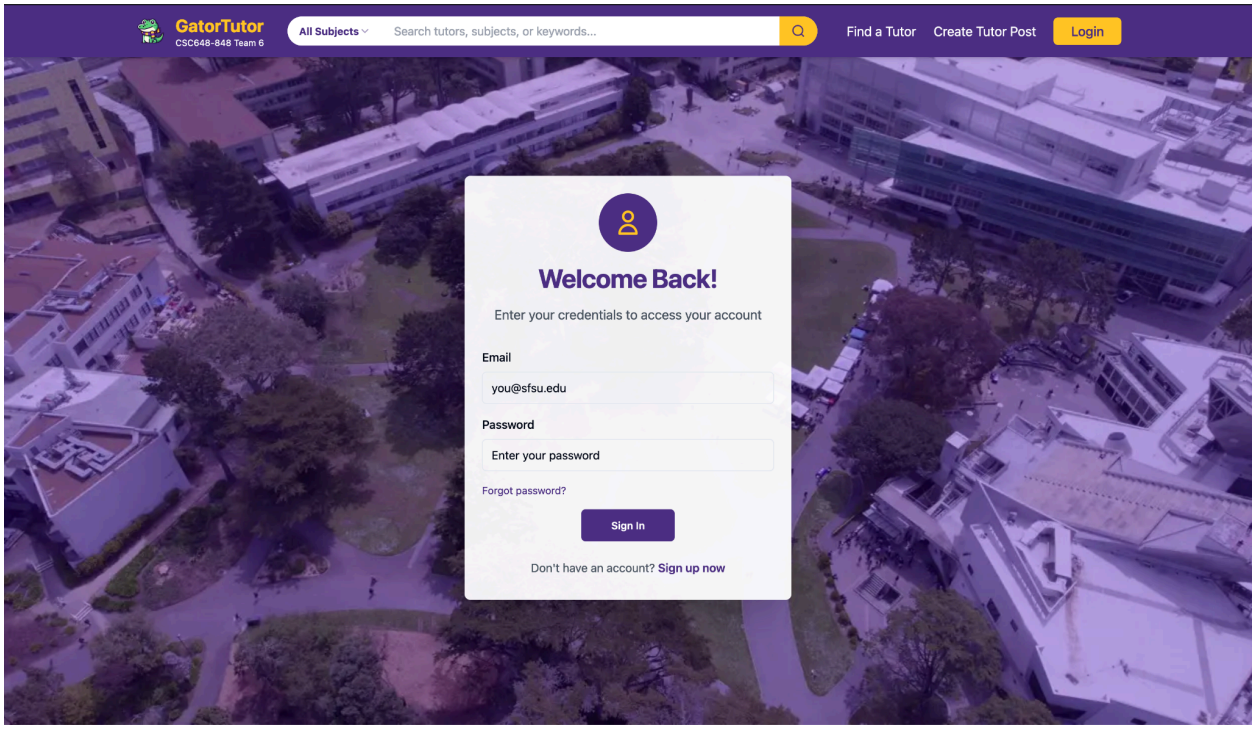
Using GenAI tools in Milestone 4 proved beneficial, particularly in code reviews, debugging, and documentation. These tools supported our development process, improved our codebase, and saved significant time while allowing us to focus on core development tasks.

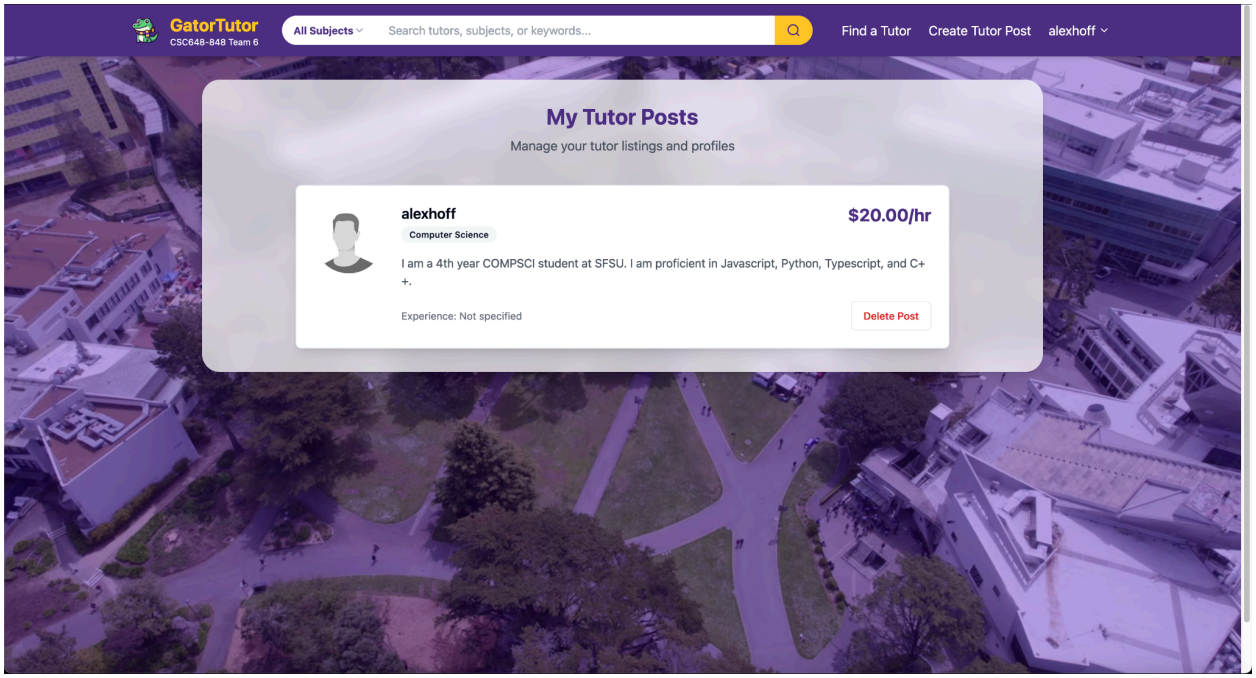
Product Screenshots:

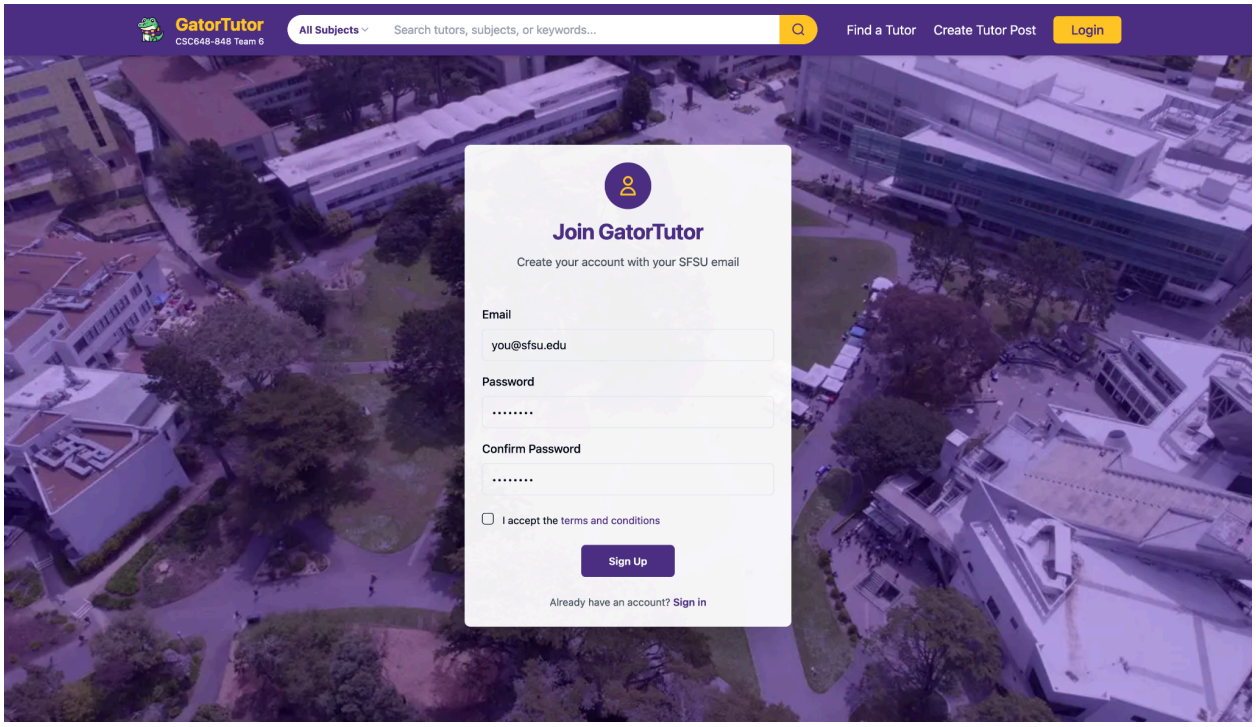


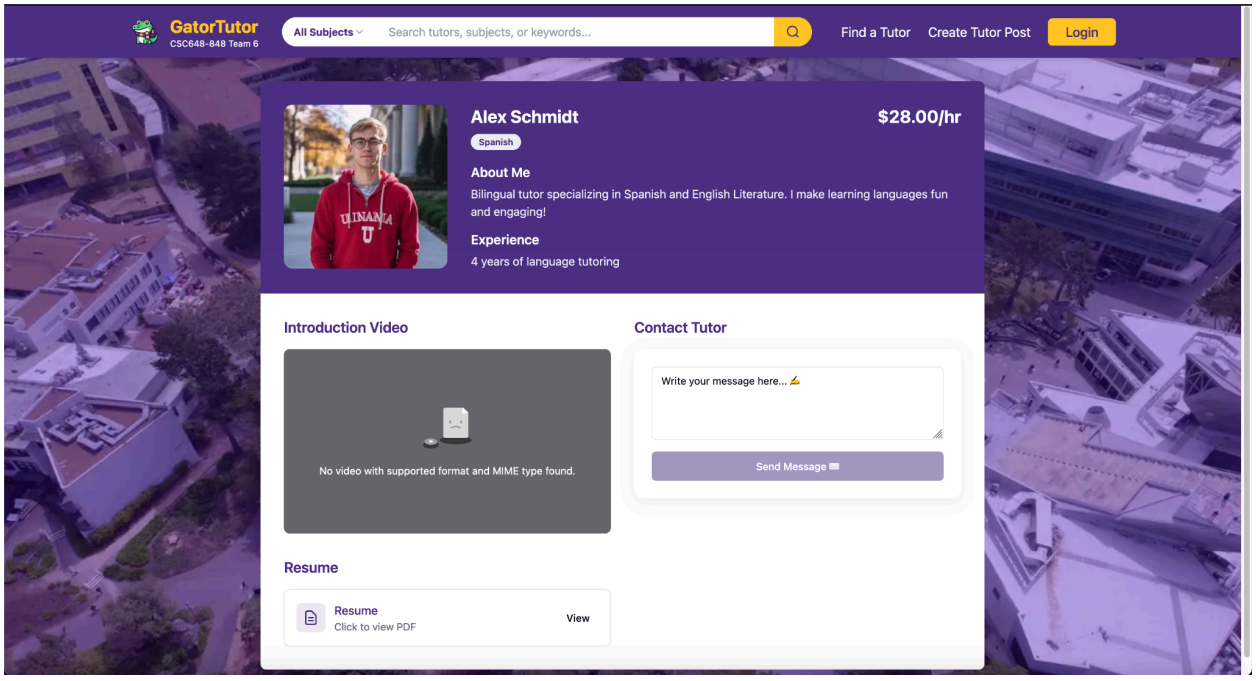



GatorTutor
CSC648-848 Team 6
All Subjects
Search tutors, subjects, or keywords...
Find a Tutor
Create Tutor Post
Login
Create Your Tutor Profile
Fill out the form below to create your tutor listing
Click to upload profile picture
Basic Information
Name
Your name
Subject
Select a subject
Hourly Rate ($)
0
Contact Information
Email or phone number
Enter your preferred contact method (email or phone number)
Bio
Tell students about yourself...
Introduction Video
MP4, WebM, OGG (10MB)
Resume/CV

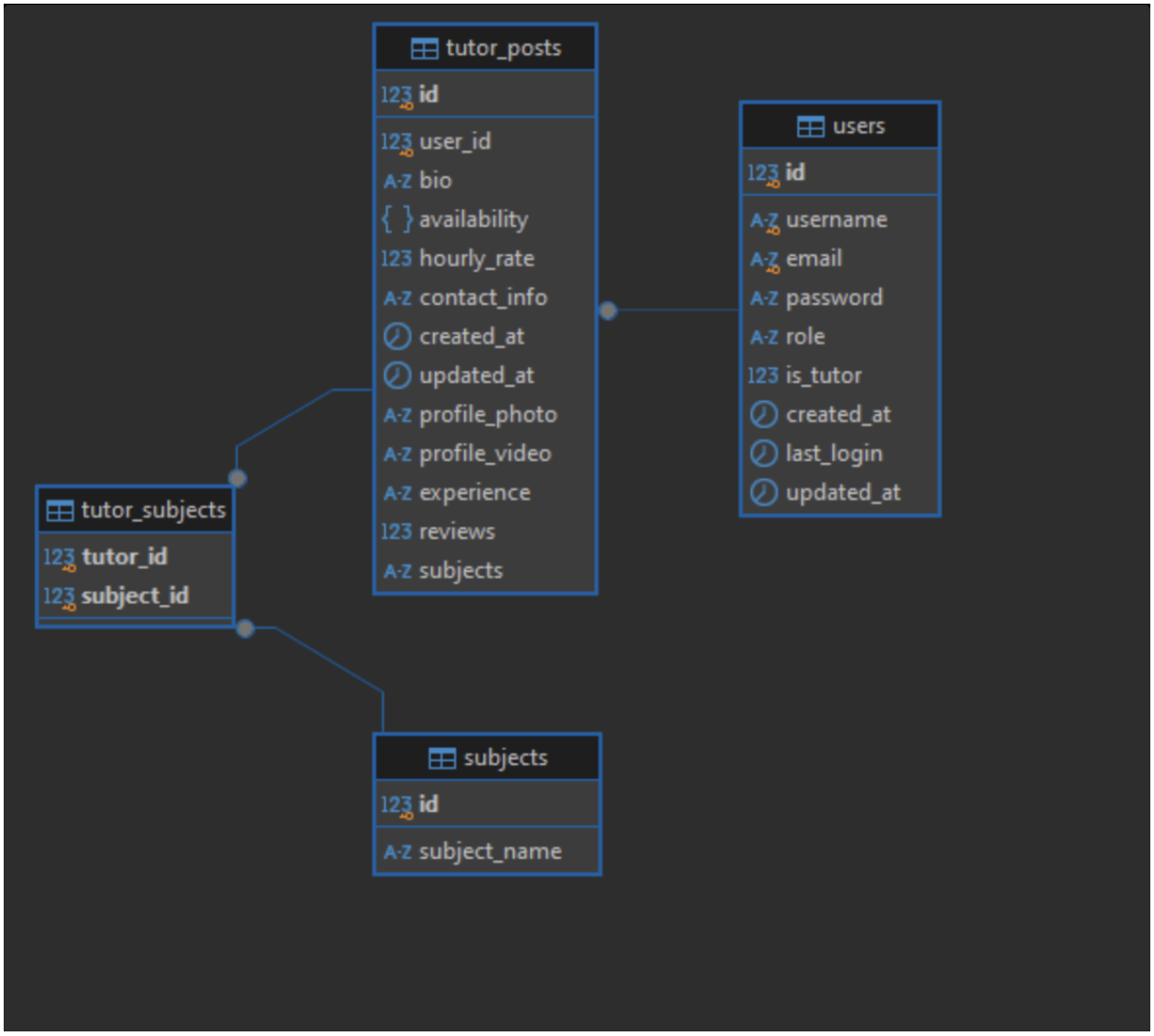

GatorTutor
CSC648-848 Team 6
All Subjects
Search tutors, subjects, or keywords...
Find a Tutor
Create Tutor Post
Login
Welcome Back!
Enter your credentials to access your account
Email
you@sfsu.edu
Password
Enter your password
Forgot password?
Sign In
Don't have an account? Sign up now


GatorTutor
CSC648-848 Team 6
All Subjects
Search tutors, subjects, or keywords...
Find a Tutor
Create Tutor Post
alexhoff
My Tutor Posts
Manage your tutor listings and profiles
alexhoff
Computer Science
$20.00/hr
I am a 4th year COMPSCI student at SFSU. I am proficient in Javascript, Python, Typescript, and C++.
Experience: Not specified
Delete Post


GatorTutor
CSC648-848 Team 6
All Subjects
Search tutors, subjects, or keywords...
Find a Tutor
Create Tutor Post
Login
Join GatorTutor
Create your account with your SFSU email
Email
you@sfsu.edu
Password
Confirm Password
I accept the terms and conditions
Sign Up
Already have an account? Sign in

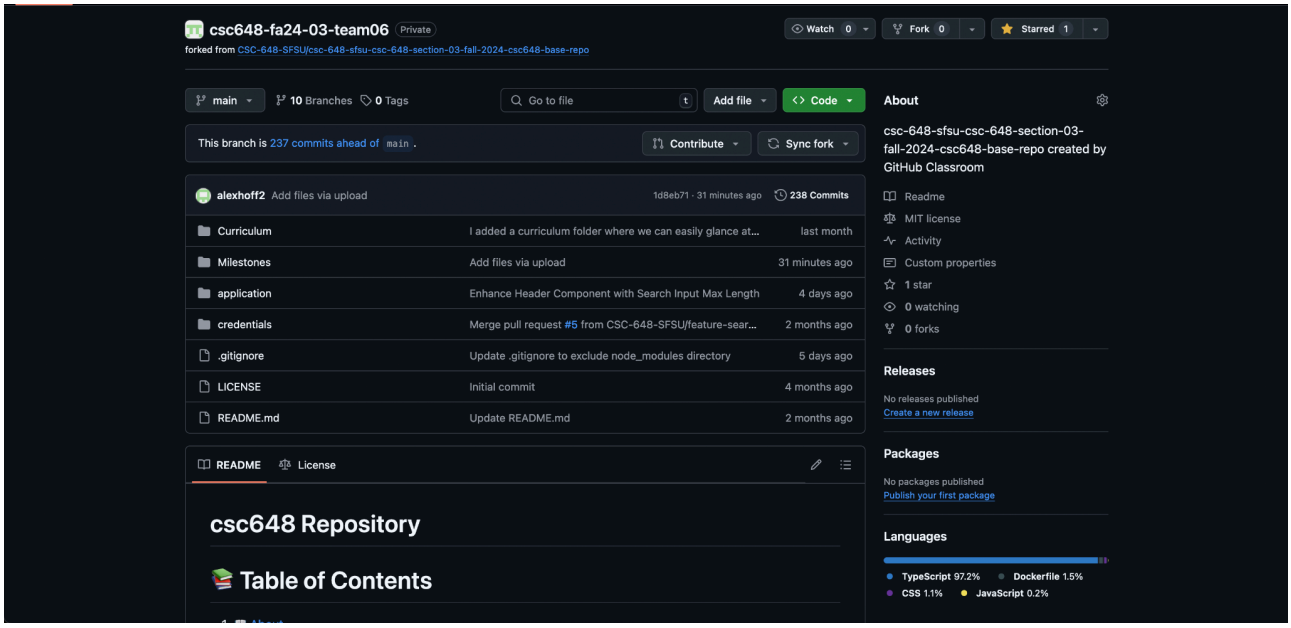

GatorTutor
CSC648-848 Team 6
All Subjects
Search tutors, subjects, or keywords...
Find a Tutor
Create Tutor Post
Login
Alex Schmidt
$28.00/hr
Spanish
About Me
Bilingual tutor specializing in Spanish and English Literature. I make learning languages fun and engaging!
Experience
4 years of language tutoring
Introduction Video
No video with supported format and MIME type found.
Contact Tutor
Write your message here...
Send Message
Resume
Resume
Click to view PDF
View

Database Organization:



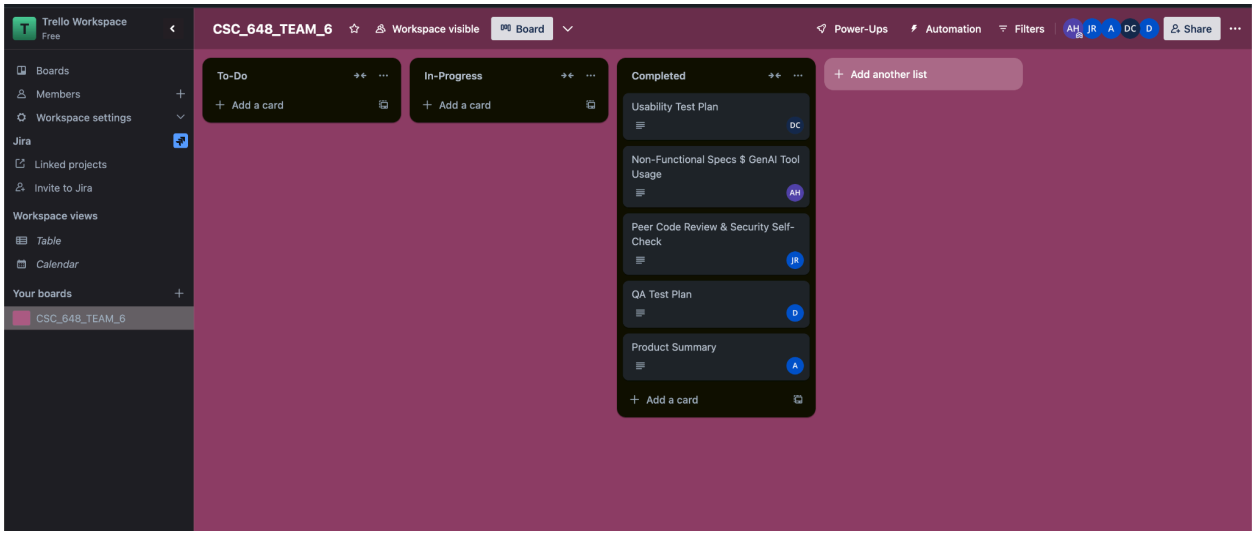
Github Organization:

- Our main branches for this project were the:
    - Development
    - Main
    - newFeature
- We used development and new feature as experimental branches while keeping main as the branch we have live on AWS
- Only I am the team lead, Dylan, the Github Master, and Jack, one of our Frontend Leads.



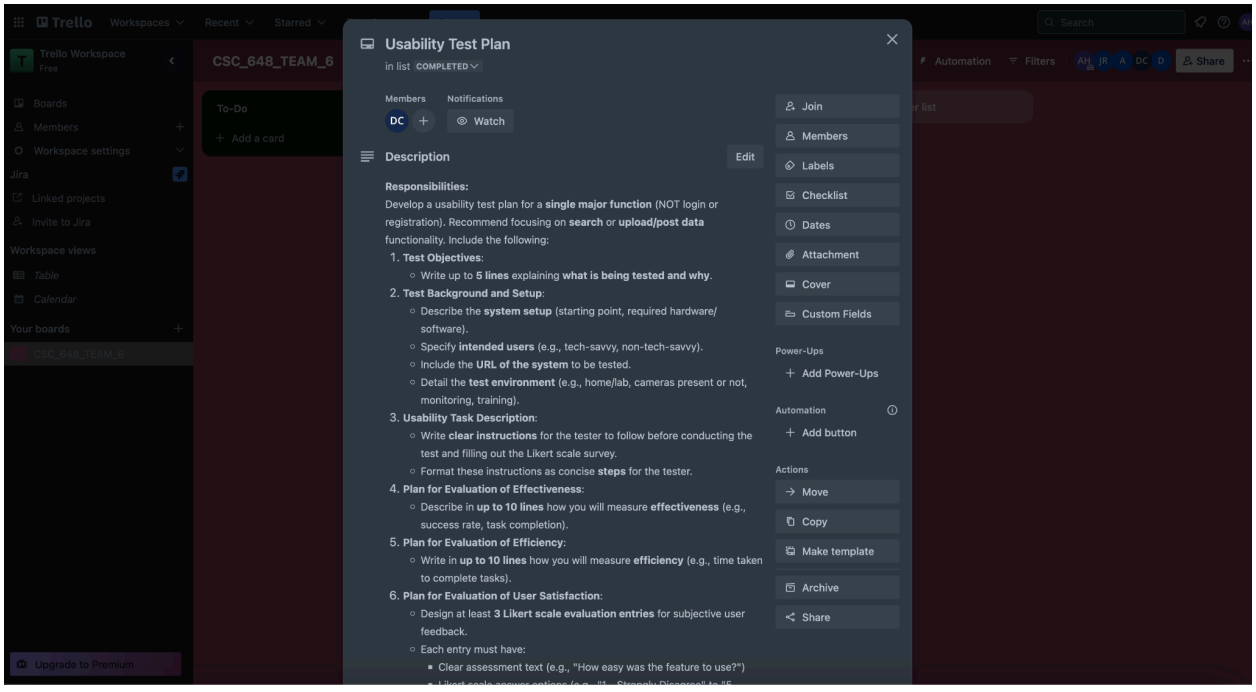
Project Management:

We took advantage of Trello's services for Milestone 4.



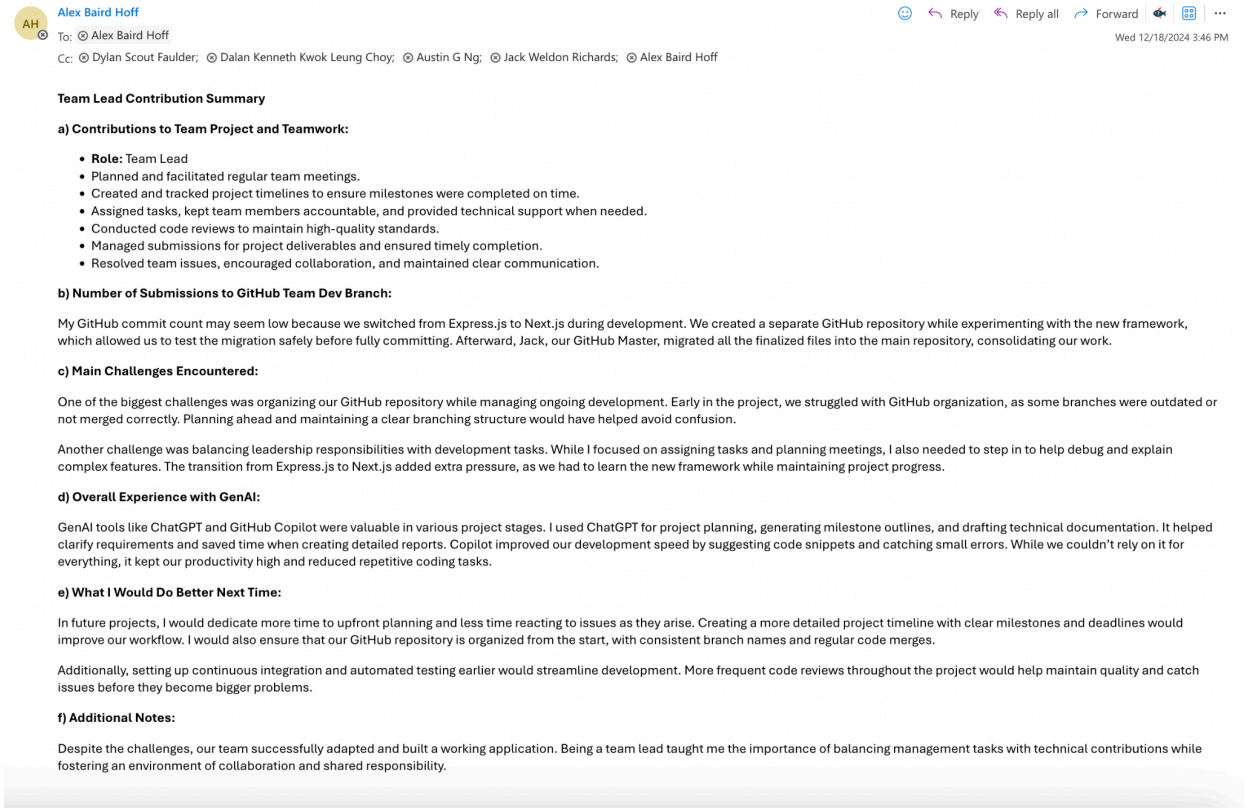
Each member had a card and within that card was a brief description of what is expected of them. All team members could then move their card depending on if it was To-Do, In-Progress, or Completed.

This is an example from our milestone 4 trello relating to Dalan's work:



Team member self assessment and contributions:

Alex Hoff, Team Lead:



Alex Baird Hoff

Reply | Reply all | Forward

To: Alex Baird Hoff

Wed 12/18/2024 3:46 PM

Cc: Dylan Scout Faulder; Dalan Kenneth Kwok Leung Choy; Austin G Ng; Jack Weldon Richards; Alex Baird Hoff

Team Lead Contribution Summary

a) Contributions to Team Project and Teamwork:

- **Role:** Team Lead
- Planned and facilitated regular team meetings.
- Created and tracked project timelines to ensure milestones were completed on time.
- Assigned tasks, kept team members accountable, and provided technical support when needed.
- Conducted code reviews to maintain high-quality standards.
- Managed submissions for project deliverables and ensured timely completion.
- Resolved team issues, encouraged collaboration, and maintained clear communication.

b) Number of Submissions to GitHub Team Dev Branch:

My GitHub commit count may seem low because we switched from Express.js to Next.js during development. We created a separate GitHub repository while experimenting with the new framework, which allowed us to test the migration safely before fully committing. Afterward, Jack, our GitHub Master, migrated all the finalized files into the main repository, consolidating our work.

c) Main Challenges Encountered:

One of the biggest challenges was organizing our GitHub repository while managing ongoing development. Early in the project, we struggled with GitHub organization, as some branches were outdated or not merged correctly. Planning ahead and maintaining a clear branching structure would have helped avoid confusion.

Another challenge was balancing leadership responsibilities with development tasks. While I focused on assigning tasks and planning meetings, I also needed to step in to help debug and explain complex features. The transition from Express.js to Next.js added extra pressure, as we had to learn the new framework while maintaining project progress.

d) Overall Experience with GenAI:

GenAI tools like ChatGPT and GitHub Copilot were valuable in various project stages. I used ChatGPT for project planning, generating milestone outlines, and drafting technical documentation. It helped clarify requirements and saved time when creating detailed reports. Copilot improved our development speed by suggesting code snippets and catching small errors. While we couldn't rely on it for everything, it kept our productivity high and reduced repetitive coding tasks.

e) What I Would Do Better Next Time:

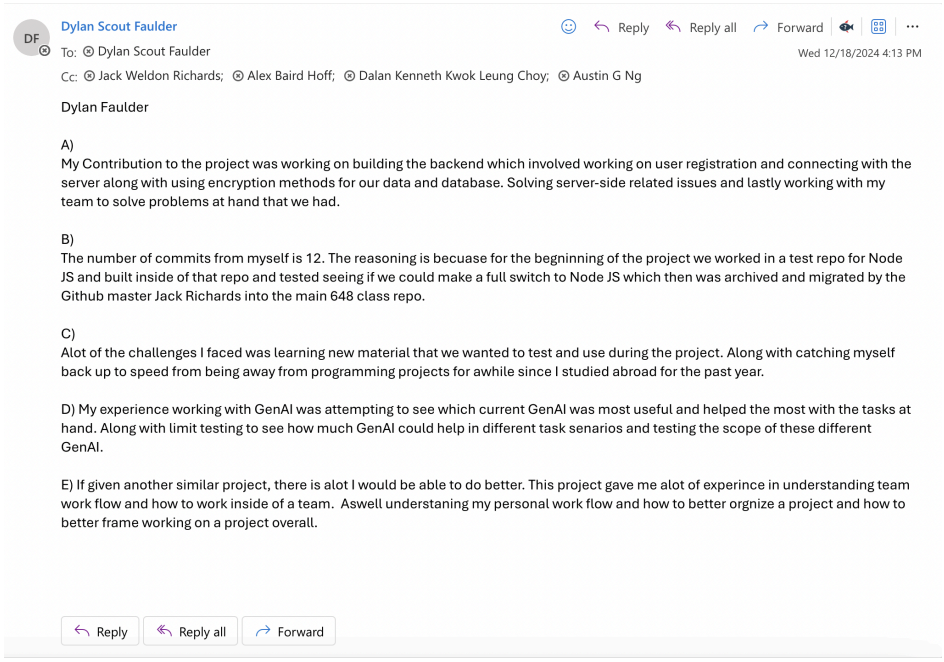
In future projects, I would dedicate more time to upfront planning and less time reacting to issues as they arise. Creating a more detailed project timeline with clear milestones and deadlines would improve our workflow. I would also ensure that our GitHub repository is organized from the start, with consistent branch names and regular code merges.

Additionally, setting up continuous integration and automated testing earlier would streamline development. More frequent code reviews throughout the project would help maintain quality and catch issues before they become bigger problems.

f) Additional Notes:

Despite the challenges, our team successfully adapted and built a working application. Being a team lead taught me the importance of balancing management tasks with technical contributions while fostering an environment of collaboration and shared responsibility.

Dylan Faulder, Backend Lead 1:



Dylan Scout Faulder

Reply | Reply all | Forward

To: Dylan Scout Faulder

Wed 12/18/2024 4:13 PM

Cc: Jack Weldon Richards; Alex Baird Hoff; Dalan Kenneth Kwok Leung Choy; Austin G Ng

Dylan Faulder

A)
My Contribution to the project was working on building the backend which involved working on user registration and connecting with the server along with using encryption methods for our data and database. Solving server-side related issues and lastly working with my team to solve problems at hand that we had.

B)
The number of commits from myself is 12. The reasoning is becuase for the begninning of the project we worked in a test repo for Node JS and built inside of that repo and tested seeing if we could make a full switch to Node JS which then was archived and migrated by the Github master Jack Richards into the main 648 class repo.

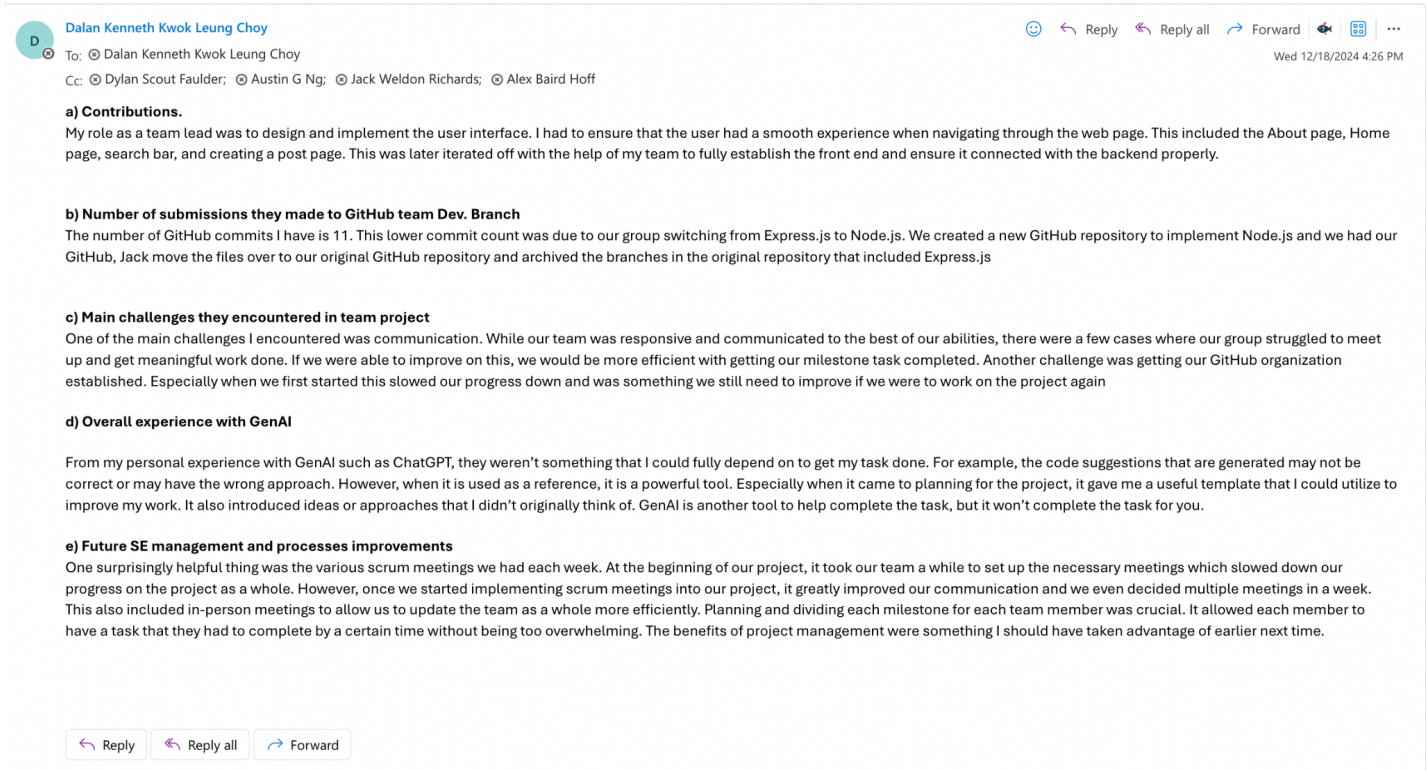
C)
Alot of the challenges I faced was learning new material that we wanted to test and use during the project. Along with catching myself back up to speed from being away from programming projects for awhile since I studied abroad for the past year.

D) My experience working with GenAI was attempting to see which current GenAI was most useful and helped the most with the tasks at hand. Along with limit testing to see how much GenAI could help in different task senarios and testing the scope of these different GenAI.

E) If given another similar project, there is alot I would be able to do better. This project gave me alot of experince in understanding team work flow and how to work inside of a team. Aswell understaning my personal work flow and how to better orgnize a project and how to better frame working on a project overall.

Reply | Reply all | Forward

Dalan Choy, Frontend Lead:



Dalan Kenneth Kwok Leung Choy

Reply | Reply all | Forward

To: Dalan Kenneth Kwok Leung Choy

Wed 12/18/2024 4:26 PM

Cc: Dylan Scout Faulder; Austin G Ng; Jack Weldon Richards; Alex Baird Hoff

a) Contributions.

My role as a team lead was to design and implement the user interface. I had to ensure that the user had a smooth experience when navigating through the web page. This included the About page, Home page, search bar, and creating a post page. This was later iterated off with the help of my team to fully establish the front end and ensure it connected with the backend properly.

b) Number of submissions they made to GitHub team Dev. Branch

The number of GitHub commits I have is 11. This lower commit count was due to our group switching from Express.js to Node.js. We created a new GitHub repository to implement Node.js and we had our GitHub, Jack move the files over to our original GitHub repository and archived the branches in the original repository that included Express.js

c) Main challenges they encountered in team project

One of the main challenges I encountered was communication. While our team was responsive and communicated to the best of our abilities, there were a few cases where our group struggled to meet up and get meaningful work done. If we were able to improve on this, we would be more efficient with getting our milestone task completed. Another challenge was getting our GitHub organization established. Especially when we first started this slowed our progress down and was something we still need to improve if we were to work on the project again

d) Overall experience with GenAI

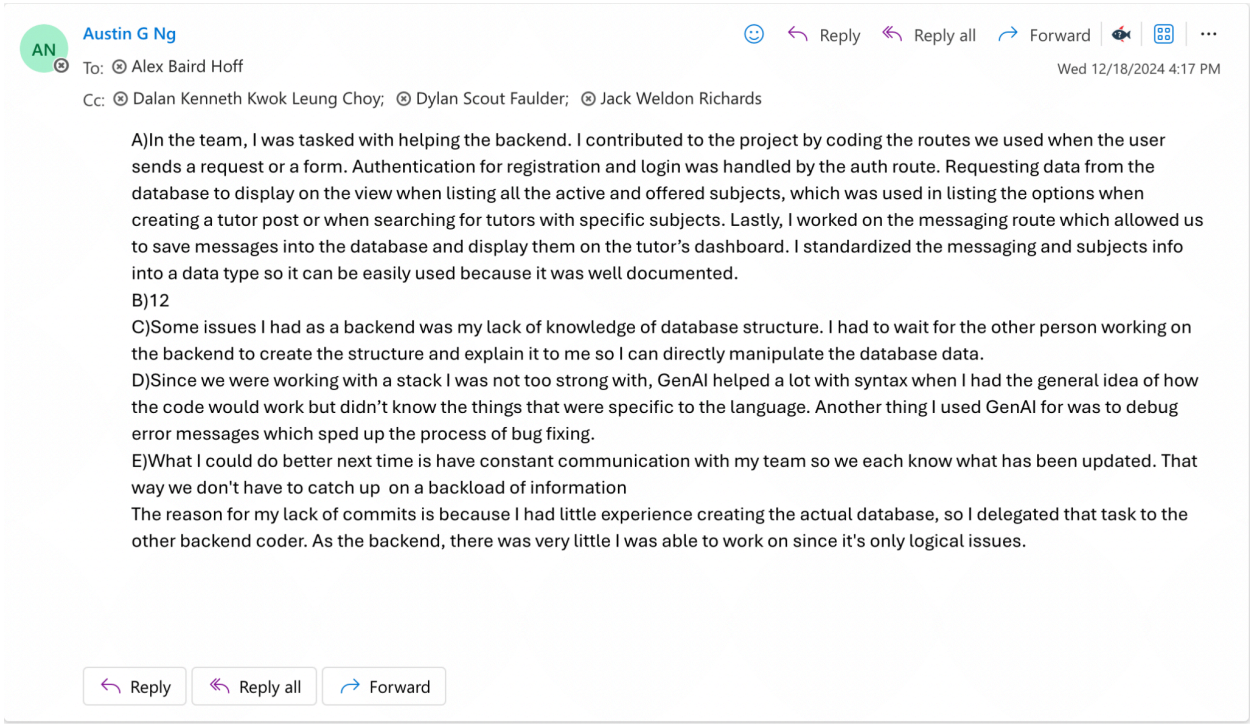
From my personal experience with GenAI such as ChatGPT, they weren't something that I could fully depend on to get my task done. For example, the code suggestions that are generated may not be correct or may have the wrong approach. However, when it is used as a reference, it is a powerful tool. Especially when it came to planning for the project, it gave me a useful template that I could utilize to improve my work. It also introduced ideas or approaches that I didn't originally think of. GenAI is another tool to help complete the task, but it won't complete the task for you.

e) Future SE management and processes improvements

One surprisingly helpful thing was the various scrum meetings we had each week. At the beginning of our project, it took our team a while to set up the necessary meetings which slowed down our progress on the project as a whole. However, once we started implementing scrum meetings into our project, it greatly improved our communication and we even decided multiple meetings in a week. This also included in-person meetings to allow us to update the team as a whole more efficiently. Planning and dividing each milestone for each team member was crucial. It allowed each member to have a task that they had to complete by a certain time without being too overwhelming. The benefits of project management were something I should have taken advantage of earlier next time.

Reply | Reply all | Forward

Austin Ng, Backend Lead 2:



Austin G Ng

Reply | Reply all | Forward

To: Alex Baird Hoff

Wed 12/18/2024 4:17 PM

Cc: Dalan Kenneth Kwok Leung Choy; Dylan Scout Faulder; Jack Weldon Richards

A)In the team, I was tasked with helping the backend. I contributed to the project by coding the routes we used when the user sends a request or a form. Authentication for registration and login was handled by the auth route. Requesting data from the database to display on the view when listing all the active and offered subjects, which was used in listing the options when creating a tutor post or when searching for tutors with specific subjects. Lastly, I worked on the messaging route which allowed us to save messages into the database and display them on the tutor's dashboard. I standardized the messaging and subjects info into a data type so it can be easily used because it was well documented.

B)12

C)Some issues I had as a backend was my lack of knowledge of database structure. I had to wait for the other person working on the backend to create the structure and explain it to me so I can directly manipulate the database data.

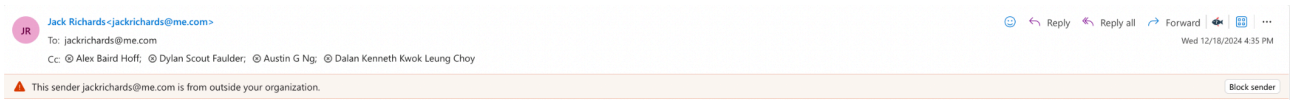
D)Since we were working with a stack I was not too strong with, GenAI helped a lot with syntax when I had the general idea of how the code would work but didn't know the things that were specific to the language. Another thing I used GenAI for was to debug error messages which sped up the process of bug fixing.

E)What I could do better next time is have constant communication with my team so we each know what has been updated. That way we don't have to catch up on a backload of information

The reason for my lack of commits is because I had little experience creating the actual database, so I delegated that task to the other backend coder. As the backend, there was very little I was able to work on since it's only logical issues.

Reply | Reply all | Forward

Jack Richards, Github Master:



Team Member Self-Assessment and Contributions

a) Contributions to Team Project and Teamwork:

- **Role:** GitHub Master
  - Led the organization and maintenance of the GitHub repository for our application.
  - Managed branch structures, merges, and resolved conflicts to ensure smooth development workflows.
  - Developed core components alongside the team and integrated them seamlessly into the shared repository.
  - Ensured consistent commit practices and provided support for team members facing GitHub-related challenges.
  - Collaborated with the team to maintain clear version control and regular updates to the repository.
  - Conducted periodic repository cleanups to avoid duplication and confusion.

b) Number of Submissions to GitHub Team Dev Branch:
I made a moderate number of commits to the GitHub Dev Branch. My contributions included significant organizational and technical work, but the count appears low due to two reasons:

1. Several structural prototypes and experimental branches were completed in separate repositories or branches to prevent disruptions in the main repository during transitions. These were later merged as consolidated features.
2. A portion of my work, such as repository organization and merging, often did not involve creating new code but was crucial for the workflow.

c) Main Challenges Encountered:
The biggest challenge I faced was maintaining an organized GitHub repository while managing ongoing development and transitions. As the team transitioned from Express.js to Next.js, I had to refactor parts of the repository structure and ensure compatibility with the new framework.

Another challenge was ensuring all team members adhered to consistent GitHub practices. As the GitHub Master, I had to resolve conflicts and ensure that all contributions were merged correctly without disrupting progress.

d) Overall Experience with GenAI:
GenAI tools like ChatGPT and GitHub Copilot proved invaluable during this project. I used ChatGPT to generate scripts for automating repetitive repository tasks and to debug Git-related issues. It also helped clarify best practices for version control and suggested efficient workflows. Copilot was particularly useful in suggesting code snippets when resolving conflicts or merging features. While these tools saved time and effort, they required oversight to ensure quality and adherence to project requirements.

e) What I Would Do Better Next Time:
In future projects, I would allocate more time for upfront planning of repository structure and branch organization. Setting up detailed guidelines for commit messages and branching strategies early on would improve clarity and reduce confusion.

Additionally, I would maintain a more detailed changelog and documentation for repository changes. This would improve transparency and streamline collaboration across the team.

Lastly, better communication with the development teams early in the project could have minimized delays caused by mismatched expectations regarding branch workflows and repository organization.

f) Additional Notes:
Despite challenges, our team delivered a functional and visually appealing application. As the GitHub Master, I gained valuable experience in balancing technical leadership with collaboration. The lessons learned will significantly enhance my ability to contribute effectively in future projects.