SW Engineering CSC648-848 Fall 2023

SFSU TutorLink

Team 02

Jeremy Woodling - Team Lead - jwoodling@sfsu.edu

Abel Seyoum - Github Manager

Brandon Watanabe - Front End Lead

Ankita Mukherjee - Backend Lead

Guillermo Villar

Lars Severson

Milestone 1

Submitted	**Revised**
9/30/23	

Table of Contents

Executive Summary

In our SFSU community, students sometimes find challenges that extend beyond the classroom. Different timetables, working on the side or managing a school-life balance are problems many students struggle with, and no one knows how to manage time and courses better than those that have already done it. That is why SFSU Tutorlink appears as a platform to connect students with tutors who have gone through the same problems earlier, and are best equipped to help students learn and pass their courses swiftly.

We are a web-based platform bridging SFSU students in need with tutors with expertise, tailored to the students courses and needs. Students can schedule their tutoring sessions matching their own timetables and goals through our messaging system linking them with their tutors. This all can seem not very tied into SFSU, but we are tied to our community through various different unique offerings like the ability to search through your Courses and Professors, an adherence to all SFSU guidelines, and our focus on development of connections within the community, making our university a more supportive and collaborative learning environment.

Finally, we believe that investing into our student-run startup is an investment on academic success and the well-being of students, as well as an investment into the sense of community here at SFSU. SFSU TutorLink is not just a project; it's a promoter for academic excellence and a testament to your commitment to student success at SFSU.

Competition Analysis

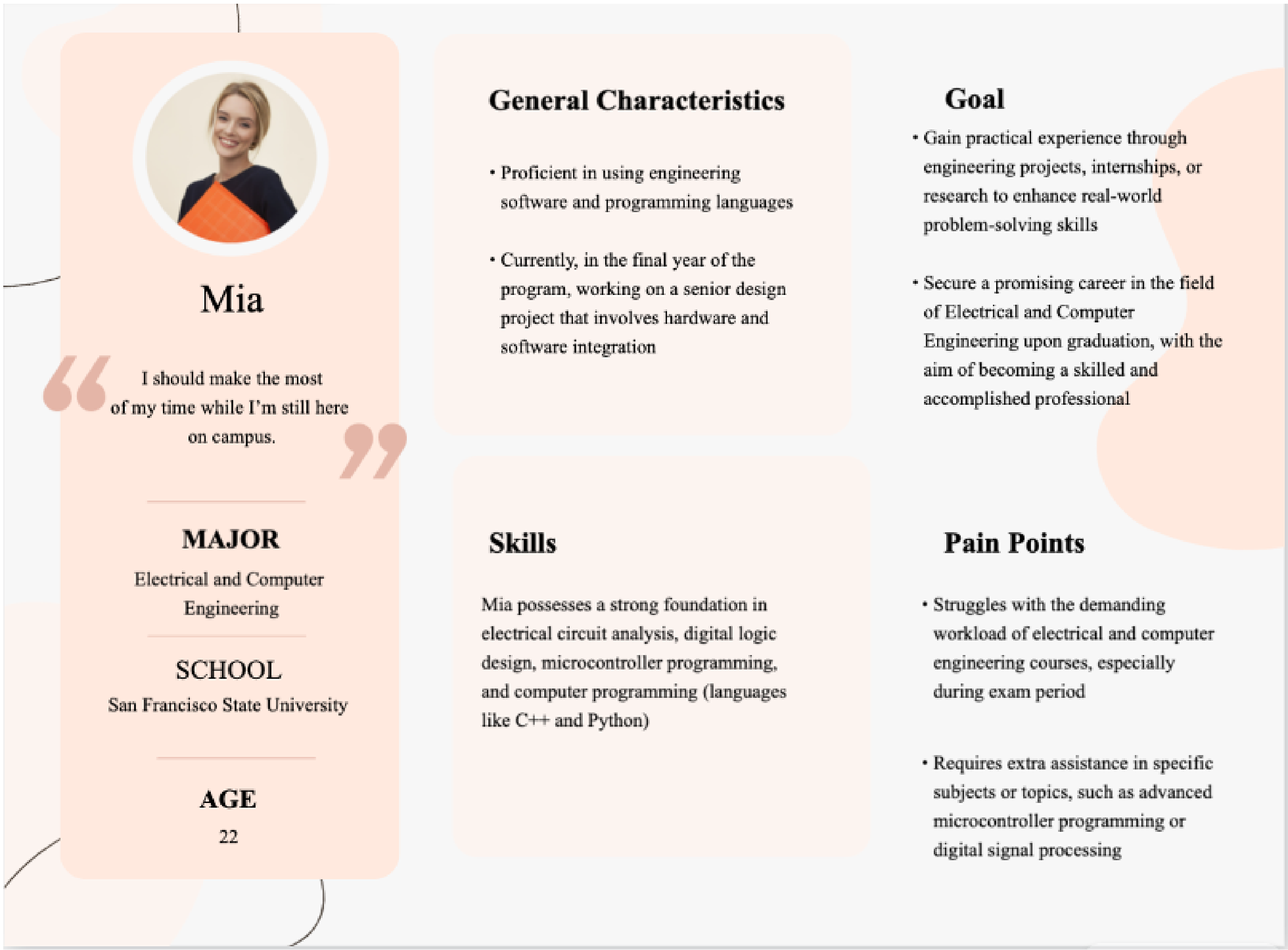
	Chegg	Wyzant	Varsity Tutor	Team 02
Search	++	++	++	+
Support	++	++	++	+
Session Format	++	++	++	+
Feedback and Reviews	-	+	+	+
Ease of Use	-	+	+	+
New User Enticement	+	++	-	+
New User Onboarding	- -	+	-	+

++ superior, + present, - inferior, - - Not Present

Our planned product shares core features commonly found in existing competitors, creating a sense of familiarity and ease of use for users. These include a standard search bar for easy tutor discovery, efficient email-based customer support channels to address user inquiries, online video sessions as a learning format, and a robust feedback and review system to empower users to provide input on their tutoring experiences. This foundation enables us to focus our efforts on not only refining and improving these core elements but also innovating and expanding upon them to deliver an exceptional and highly personalized learning journey that sets us apart from the competition.

Personas

1. Student - Mia



Mia

“I should make the most of my time while I’m still here on campus.”

MAJOR
Electrical and Computer Engineering

SCHOOL
San Francisco State University

AGE
22

General Characteristics

- Proficient in using engineering software and programming languages
- Currently, in the final year of the program, working on a senior design project that involves hardware and software integration

Goal

- Gain practical experience through engineering projects, internships, or research to enhance real-world problem-solving skills
- Secure a promising career in the field of Electrical and Computer Engineering upon graduation, with the aim of becoming a skilled and accomplished professional

Skills

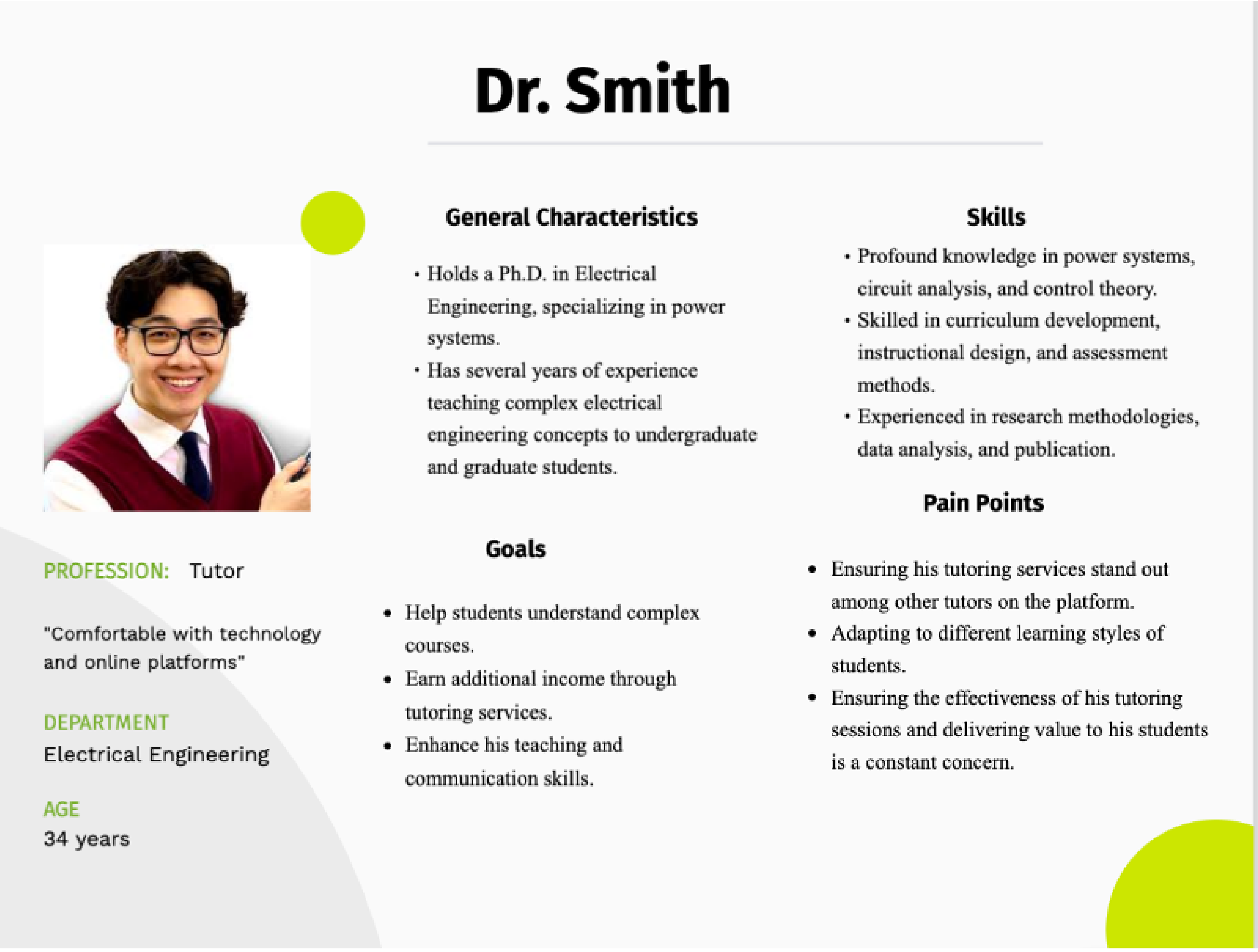
Mia possesses a strong foundation in electrical circuit analysis, digital logic design, microcontroller programming, and computer programming (languages like C++ and Python)

Pain Points

- Struggles with the demanding workload of electrical and computer engineering courses, especially during exam period
- Requires extra assistance in specific subjects or topics, such as advanced microcontroller programming or digital signal processing

@Source: https://www.visme.co/templates/

2. Tutor: Dr. Smith



Dr. Smith

PROFESSION: Tutor

"Comfortable with technology and online platforms"

DEPARTMENT
Electrical Engineering

AGE
34 years

General Characteristics

- Holds a Ph.D. in Electrical Engineering, specializing in power systems.
- Has several years of experience teaching complex electrical engineering concepts to undergraduate and graduate students.

Goals

- Help students understand complex courses.
- Earn additional income through tutoring services.
- Enhance his teaching and communication skills.

Skills

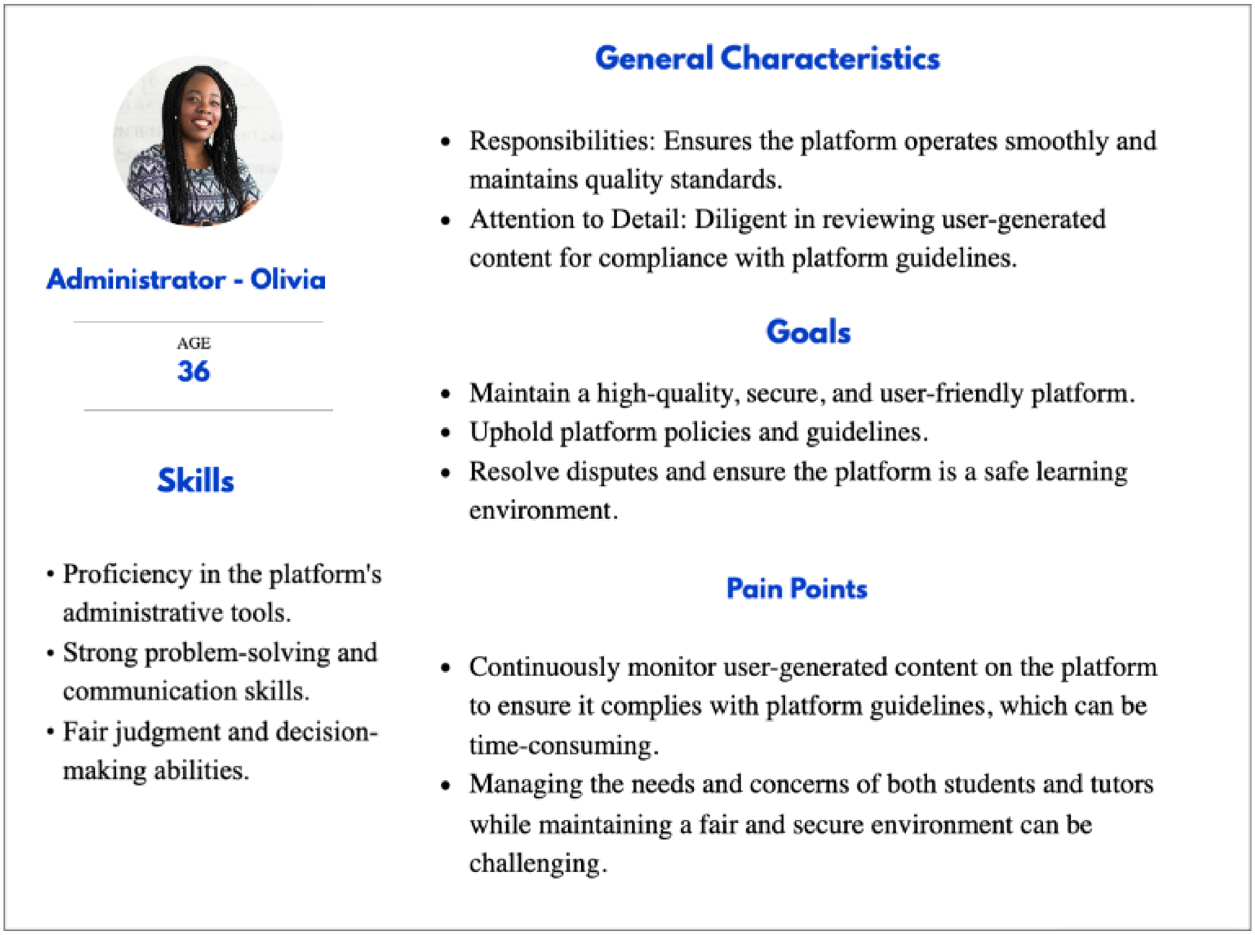
- Profound knowledge in power systems, circuit analysis, and control theory.
- Skilled in curriculum development, instructional design, and assessment methods.
- Experienced in research methodologies, data analysis, and publication.

Pain Points

- Ensuring his tutoring services stand out among other tutors on the platform.
- Adapting to different learning styles of students.
- Ensuring the effectiveness of his tutoring sessions and delivering value to his students is a constant concern.

@Source: https://www.visme.co/templates/

3. Admin: Olivia



Administrator - Olivia

AGE
36

Skills

- Proficiency in the platform's administrative tools.
- Strong problem-solving and communication skills.
- Fair judgment and decision-making abilities.

General Characteristics

- Responsibilities: Ensures the platform operates smoothly and maintains quality standards.
- Attention to Detail: Diligent in reviewing user-generated content for compliance with platform guidelines.

Goals

- Maintain a high-quality, secure, and user-friendly platform.
- Uphold platform policies and guidelines.
- Resolve disputes and ensure the platform is a safe learning environment.

Pain Points

- Continuously monitor user-generated content on the platform to ensure it complies with platform guidelines, which can be time-consuming.
- Managing the needs and concerns of both students and tutors while maintaining a fair and secure environment can be challenging.

@Source: https://www.visme.co/templates/

Use Cases:

1. Persona 1: Student: Mia

1. Browsing and Searching Tutors
    - Mia logs into the tutoring service.
    - She uses the search by subject category to find tutors specializing in her programming languages and topics.
    - Mia reviews tutor profiles, focusing on their qualifications and expertise in her specific areas of interest.
2. Reviewing Tutors
    - Mia reads in-depth reviews and ratings from other students who have received tutoring from prospective tutors.
    - She pays close attention to reviews highlighting tutors' effectiveness in explaining complex programming concepts.
3. Contacting Tutors
    - After selecting a tutor with expertise matching her needs, Mia contacts the tutor to discuss her specific challenges and schedule tutoring sessions.
    - She awaits a prompt response from the tutor.

2. Persona 2: Tutor - Dr. Smith

1. Uploading Tutor Information

- Dr. Smith creates a comprehensive tutor profile, including his CV, professional photo, and a video introducing himself.
- He can search for the electrical engineering subjects he can teach.
- Sets competitive pricing, and specifies his availability after signup.

2. Managing Profile

- Dr. Smith frequently updates his profile to showcase his expertise and qualifications.
- He monitors and promptly responds to messages from students, addressing inquiries, and scheduling tutoring sessions.

3. Persona 3: Admin - Olivia

1. Approving Tutor Profiles

- Olivia receives notifications when tutors upload their information.
- She reviews each tutor's profile meticulously, ensuring it complies with platform guidelines, and approves them for publication when appropriate.

2. Managing User Complaints

- Olivia addresses user complaints and inquiries, including disputes between students and tutors or technical issues.
- She conducts thorough investigations and resolves issues while maintaining a fair and secure environment for all users.

High-Level Requirements

User Registration and Profile Management:

Requirement Name: User Account Management

Description: Users should be able to create, edit, and manage their profiles. This includes personal information, contact details, and preferences. Some functions tbd for "guest" users.

Course Search and Listing:

Requirement Name: Course Search and Listing

Description: Users should be able to search for specific courses, view course details, and see a list of available tutors for each course.

Tutor Matching:

Requirement Name: Tutor Matching

Description: Users should be able to request a tutor for a specific course, and the system should match them with an available tutor (based on course compatibility and availability?).

Appointment Scheduling:

Requirement Name: Appointment Scheduling

Description: Users should be able to schedule tutoring sessions with their matched tutors, specifying date, time, and location.

Messaging System:

Requirement Name: Messaging System

Description: Users should have a messaging platform to communicate with tutors for inquiries, session coordination, and support.

User Authentication and Security:

Requirement Name: User Authentication and Security

Description: The system should ensure secure user authentication and protect user data using encryption and secure login practices.

Notification System:

Requirement Name: Notification System

Description: Users should receive notifications for important events, such as tutor matches, appointment confirmations, and new messages.

User Support and Help Center:

Requirement Name: User Support and Help Center

Description: The system should provide a help center with FAQs, guides, and support options to assist users.

Admin Dashboard:

Requirement Name: Admin Dashboard

Description: Administrators should have access to a dashboard for managing user accounts, tutor listings, resolving disputes, and monitoring system activities.

Privacy and Data Protection:

Requirement Name: Privacy and Data Protection

Description: The system should comply with data protection regulations and ensure user privacy by allowing users to control their data and providing a privacy policy.

Payment Integration (Optional):

Requirement Name: Payment Integration (Optional)

Description: If applicable, users should have the option to make payments for tutoring services securely through the webpage.

Functional Requirements

1. Unregistered User:

   1.1. A user shall be able to search for a tutor by class or topic

   1.2. A user shall be required to register prior to contacting a tutor

   1.3. A user shall be able to register for a single account by using their "@sfsu.edu" email

   1.4. A user shall be able to start the application for becoming a tutor without prior registration

2. Registered User:

   2.1. A registered user has all the permissions of an unregistered user

   2.2. A registered user shall be able to submit an application for becoming a tutor

   2.3. A registered user shall be able to upload a one and only one profile picture

   2.4. A registered user shall be able to remove their profile picture

   2.5. A registered user shall be able to leave one and only one review on a tutor's profile

   2.6. A registered user shall have at most one role

   2.7. A registered user shall be able to start a chat with a tutor

   2.8. A registered user shall be able to respond to a message

3. Profile Pictures:

   3.1. A profile picture shall be uploaded by one and only one user

   3.2. A profile picture shall be viewed by many users

4. Reviews:

4.1. A review shall be written by one and only one registered user

4.2. A review shall be associated with one and only one tutor

5. Role:

5.1. A role shall be linked to many registered users

6. Tutor:

6.1. A tutor shall upload one and only one resume

6.2. A tutor shall have many reviews

6.3. A tutor shall be able to upload many photos

6.4. A tutor shall be able to change their posted photos

6.5. A tutor shall be able to upload one and only one video

6.6. A tutor shall be able to change their posted video

7. Resume:

7.1. A resume shall be added by a tutor

7.2. A resume shall be viewed by many users

8. Video:

8.1. A video shall be added by a tutor

8.2. A video shall be viewed by many users

9. Photo:

9.1. A photo shall be added by a tutor

9.2. A photo shall be viewed by many users

10. Admin:

10.1. Admin shall be required to approve tutor applications prior to going live

10.2. Admin shall be required to deny inappropriate tutor applications

11. Message:

11.1. A message shall be able to be created by a user

11.2. A message shall be able to be received by one and only one user

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
4. Data shall be stored in the database on the team's deployment server.
5. No more than 50 concurrent users shall be accessing the application at any time
6. Privacy of users shall be protected
7. The language used shall be English (no localization needed)
8. Application shall be very easy to use and intuitive
9. Application shall follow established architecture patterns
10. Application code and its repository shall be easy to inspect and maintain
11. Google analytics shall be used
12. No e-mail 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
13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
15. Media formats shall be standard as used in the market today
16. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development

17. The application UI (WWW and mobile) shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2023. For Demonstration Only” at the top of the WWW page nav bar. (Important so as to not confuse this with a real application).

Architecture and Technologies

Server Host	AWS EC2, t2.micro ● 1 CPU Core ● 1 GB RAM
Operating System	Ubuntu 20.04.3 LTS
Server Database	MySQL v8.0.34
Web Server	Apache2 2.4.52
Server-Side Language	Python 3.10.12
Web Framework	Flask
Additional Technologies	● SQLAlchemy ● Flask-login ● Auth0 (possibly, may extend scope of project)
IDE	PyCharm, VSCode, Webstorm Web
Analytics	Google Analytics
SSL Cert	Lets Encrypt (Cert Bot)

ChatGPT

We have used the September 25th version.

We did review the policy and found it useful to go through some requirements text and as a spell checking tool for our executive summary. In addition to this, the team lead will be using ChatGPT for email fluff as it is effective with some minor editing.

We would say it definitely is helpful (Does a bit more in depth grammar checking than google docs word directory matching). It would be classified towards medium usefulness.

We did draft our assignment first, as mentioned it was used for spell checking and more general grammar adjustments, but our contents and ideas were already laid out.

Data Glossary

Roles	
Student	Unverified role given to new users and the main user base who use the service to find tutors
Tutor	Verified role of tutors offering services to students
Admin	Manually added role for staff managing the website

Program Structure (folders/root files)	
__init__.py	Index page and app entrypoint.
db	Database specific python files.
mod	General purpose modules (mainly routes and route helper functions
static	Static files (non-user uploaded)

Routes	
Index (__init__.py)	Main app setup and handing the index page.
about.py	Handles the about page and subpages.
search.py	Handles the search and displaying of search results.
profile.py	Handles displaying and editing of user profiles (for both student and tutor).
chat.py	Handles conversations with tutors on the service.
error.py	Handles error pages.
login.py	Handles user logging in and verification. API handles all technical account based operations.
register.py	Handles account creation (mainly in terms of front end, used login API).
apply.py	Handles tutor applications.
home.py	Home page showing the highest rated tutors and their bios. Tutor subject ignored if user is not logged in or is logged in but hasn't declared a major. Shows major specific tutors if logged in and major is declared.

Route Structure	
[route].py	Handles the page and routes for the website relevant to a given purpose that the user interacts with (see routes section).
[route]_fe.py	Hold classes to be populated by the backend and given to the front end for ease of asynchronous development and communication between FE and BE.
[route]_api.py	Hold api routes and helper functions that may need to be used by other parts of the program.