Final Project for SW Engineering CSC 648-848 Fall 2023
Team 6
Wise Guidance And Tutoring Outreach Resource (wiseGATOR)

Role	Team Members
Team Lead	Joaquin Warren jwarren3@mail.sfsu.edu
Github Master	Ronnie Huang
Front-end Lead	Karl Moreno
Front-end Developer	Darien Banuelos
Back-end Lead	Sean Ryan
Back-end Developer	Philip Karnatsevich

http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/

12-13-2023

1) Product Summary:

- **Name of the Product:** WiseGATOR
- **List of Features:**
  Unregistered users shall be able to have access to view all tutors
  Unregistered users shall be able to view all available tutors as default
  Unregistered users shall be able to filter tutors based on topic/subject e.g. MATH, ENG
  Unregistered users shall be able to narrow search results by typing in search bar
  Unregistered users shall be able to register
  Unregistered users shall be able to apply to become a tutor, and will be prompted to register or log in
  Registered users shall be able to apply to be a tutor, submitting their information
  Registered users shall be able to upload a video
  Registered users shall be able to upload a photo
  Registered users shall be able to upload their CV, Flyer, or Resume
  Registered users shall be able to log in
  Registered users shall be able to log out of their account
  Registered users shall be able to message a tutor of their choice
  Registered users shall be able to access their messages and tutor posts via dashboard
  Registered users shall be prompted to choose what topic(s) they tutor when applying as tutor
  Registered users shall be able to write information about themselves inside of the description field
  Admin shall be able to view database
  Admin shall be required to approve registered users to become tutors before they go live
  Admin shall be able to reject user posts before they are published

Wise Guidance And Tutoring Outreach Resource (wiseGATOR), is a quick-to-learn SFSU exclusive tutoring service that aims to connect students that could learn from one another. Our platform offers convenient access to students in need of help in SFSU courses. What makes our site stand out is that we cater to the needs of SFSU students by hosting tutors only registered at SFSU and have taken courses offered at the college to provide much more relevant expertise and mentorship.

- **Link**: http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/

2) Milestone documents (M1-M4):

Milestone 1:

SW Engineering CSC 648-848 Fall 2023
Wise Guidance And Tutoring Outreach Resource (wiseGATOR)
Team 6

Team Member	Role
Joaquin Warren jwarren3@mail.sfsu.edu	Team Lead
Ronnie Huang	Github Master
Karl Moreno	Front-end Lead
Darien Banuelos	Front-end Developer
Sean Ryan	Back-end Lead
Philip Karnatsevich	Back-end Developer

Milestone 1

Revision History

Date Submitted	10/01/23
Date Revised	10/05/23

1.Executive Summary:

If you're a student, you're familiar with the struggle of juggling multiple classes and meeting overlapping deadlines. Time is money when you're in college, and with our new application, we can save our users both! We relate to the challenges students face when looking for convenient and affordable access to tutoring services on campus. That's why we started developing wiseGATOR, a brand new SFSU Guidance and Tutoring Outreach Resource. It's not only important to pass the classes, but to understand and retain the key concepts in each course. wiseGator allows SFSU students to find SFSU tutors relevant to their coursework, to get fast help and mentorship. SFSU students often depend on one another to get assistance with their classes and homework. This application will bridge the gap between students in need of tutoring services and experienced students that can offer specialized help.

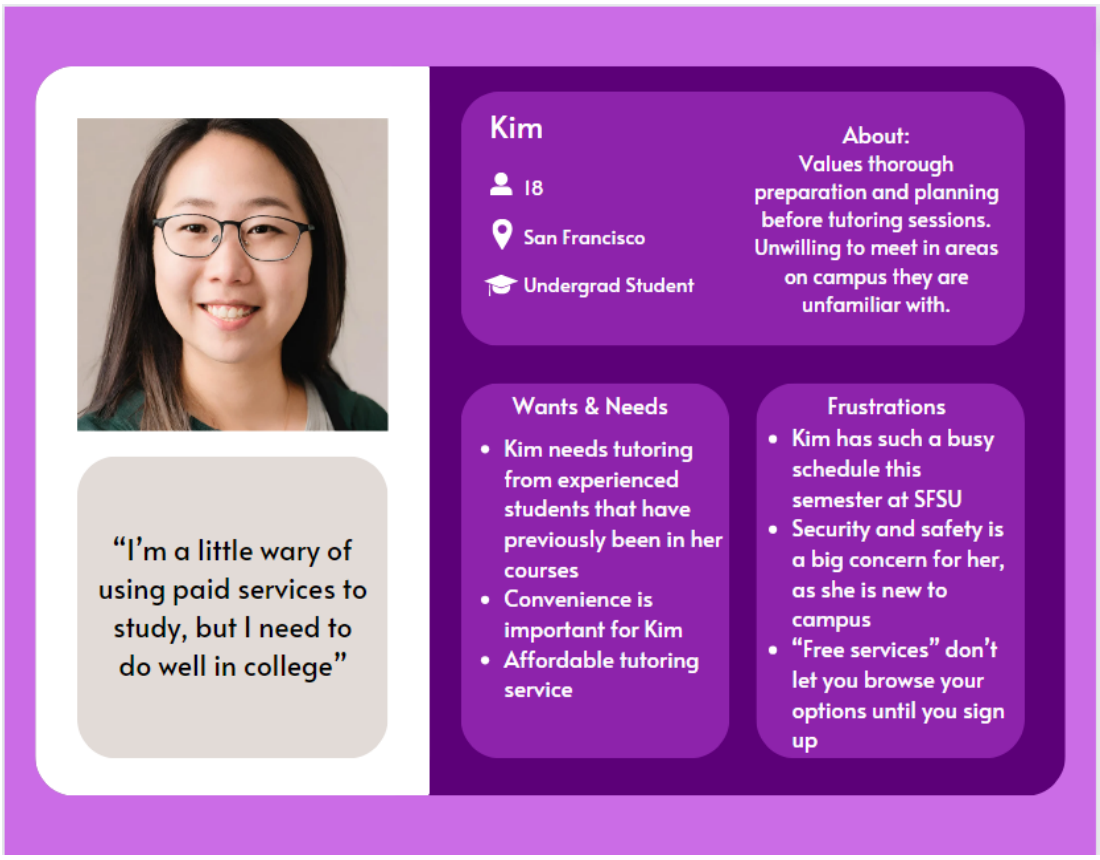
Our application aims to quickly connect students to tutors and students interested in becoming tutors to paid mentorship roles. Students at SFSU are already footing the bill when it comes to tuition prices, and they're on the rise as of September 2023. We can save students time and money by skipping the Chegg membership or paying for Quizlet premium. Students can get 1 on 1, specific help with homework assignments, coursework, study guides, and more at a reasonable, cost-efficient rate. This is also a great opportunity for SFSU students that excel in their courses to make money by becoming a mentor and offering tutoring services on our site to other SFSU students. It's a one stop shop for any SFSU student seeking or offering expert help. Our goal is to streamline the process of finding tutoring, by allowing students to filter by subject and by courses offered at SFSU. This way, students are learning from one another in the same courses, and they do not have to rely on outdated and irrelevant internet forums or paid services from people that have never even been in that class. What makes our application so unique is that it is exclusive to San Francisco State University, whether you want to find a tutor or tutor others in any of the courses offered at SFSU. Searching by courses and subjects will optimize the experience for students seeking tutoring services, and SFSU tutors will be able to cater their service directly to SFSU students.

The team behind wiseGATOR is a group of 6 ambitious computer scientists at San Francisco State University. With a startup funded by Professor Petkovic of the College of Science & Engineering, this team is dedicated to bringing wiseGATOR to personal devices of SFSU students by the end of 2023. Team 6 has experienced team members with a strong background in both Web and Software Development. Our values are aligned with the interest of both students and professors alike to provide a successful experience for all of our users in their educational careers. It is important to

us as students that we can make a meaningful positive impact on our peers and utilize our best talents to extend that to more students and alumni of SFSU.

2. Personae and main Use Cases:

Personas:



"I could use a little extra money, so I ought to put my free time and talent to use"

Victor

23

San Francisco

Graduate Student

About:
Desperately needs a part time job and has great tutoring experience and familiarity with other sites. Enthusiastic about helping fellow students succeed but grows impatient with unintuitively designed applications.

Wants & Needs

- Adamant on helping peers succeed in their academic journey and wants to sign up to be a tutor for SFSU
- Platform to advertise his skills and knowledge to acquire tutoring job

Frustrations

- Just transferred to campus, not aware of what is available yet
- Although willing to help students, not patient with bad UI/UX
- Does not need to learn how to use website

"I'm looking for a position that will allow me to apply my knowledge of databases and take on an administrative role"

Jamie

22

San Francisco

Undergrad Student

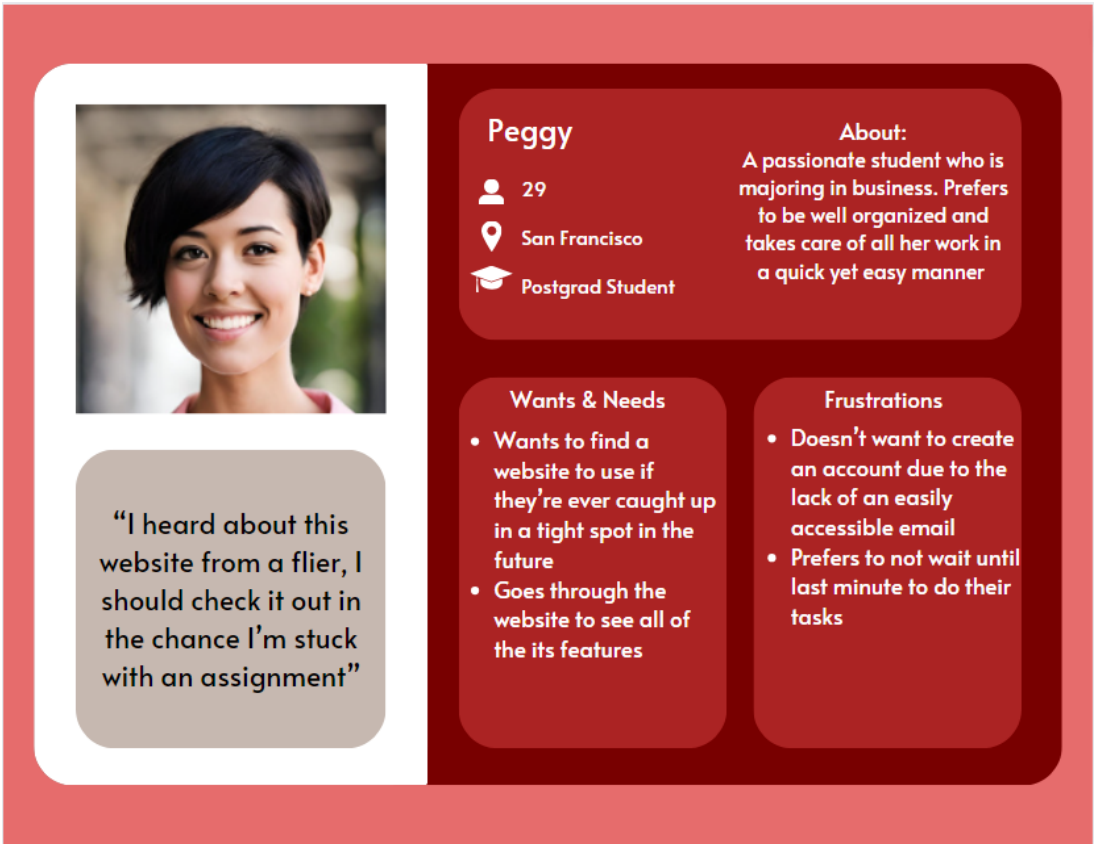
About:
Aspiring database administrator, looking to work with a startup of students from SFSU. Needs to keep his schedule balanced while he studies

Wants & Needs

- Administrative role that complements his experience and area of interest
- Reasonable schedule as a grad student
- Meaningful work that helps him grow as a potential systems manager

Frustrations

- Overly complicated applications make it hard for him to effectively utilize his time
- Does not want to learn any brand new technologies but rather practice with familiar ones



Use Cases:

1. **User-friendly Interface & Messaging:** Kim wants to efficiently set up a safe tutoring appointment. She filters through her choices of tutors with the conveniently accessible interactive calendar. Here she can put in classes where he needs help and is prompted with multiple tutors. Before setting up an appointment via message, she is prompted to log in or register. She can draft a message to send to her desired tutor and upon submitting the website requires that she logs in to an existing account or registers for one using her SFSU email.
2. **Tutor Dashboard:** Victor wants to become a tutor on our site. After accessing our website he selects the easy to select tutor application button and is requested to log in or register, as tutors must be registered users. He is given a list of subjects to choose from and is then presented with SFSU exclusive classes related to the subject(s) selected; he then chooses the classes he wishes to tutor for. After appropriately moving forward he is able to input his available times to tutor and will promptly receive any related session information regarding potential tutees. If Victor wishes to do so, he may browse the tutor listings already on the site and until he chooses to apply to become a tutor, he will not be required to register or sign in. Victor's posts will be reviewed by an admin before going live, which means that anyone who accesses the site can view his post. As

a registered user, Victor navigates to the dashboard and sees any messages he has received from other registered users, inquiring about his tutoring service.

3. **Submission Review:** Jamie wants to be an admin for our site. Upon applying to become an admin he is to acquire a special administrator account, this is because to administer our website you must have permissions that exceed those of our users. As an admin he will have access to our database and tables in relation to our website e.g. students, tutors, times, etc. Admins will also have full access to our site's functionalities and features for admin or tutoring related use. The main responsibility of an admin is to approve posts submitted by tutors before they are published on our website for anyone, registered or not, to view.
4. **Unregistered User Browse:** Peggy is a smart college student who's just browsing the site. In the past she's gotten a lot of spam emails from her time at applying to college, and is now very hesitant on giving her email out. One day, while walking around campus at SFSU, she sees a flier advertising the tutor website and decides to check it out. She's aware of places advertising tutoring (such as Discord servers or users on "Twitter/X"), but they're all different in terms of features, price, and accessibility, so she has no clue what to expect from this site. Upon visiting, she notices the option to sign-up, yet she ignores this, assuming that they'll likely require her to enter her email. She checks out the features of the site such as available tutors and searching by name of course. She is able to freely browse the site and look at any of the available tutors. She can click on them to learn more and even preview their posts and its content. The search bar also contains a drop down menu so that she can identify what subjects the site has tutors for. After seeing the site, she takes note of it for future use.

3.List of main data items and entities:

Users:

Unregistered User
An unregistered user is any person who is browsing the site without being logged into any account. They can browse all available tutors and perform searches. They can apply to be a tutor at any time but will be prompted to register.

Registered User
Unregistered users become a registered user when they sign up for an account or if they log in to an existing one. Registered users have all of the permissions an unregistered user has with the added bonus of being able to submit a message to a tutor and apply to be a tutor.

Admin
Admin is logged in to the web service or registered user, and is a person who is responsible for managing and supporting a tutoring web service. Admin has full access to all features and functionalities of the web service and its database. They will accept and reject tutor posts in accordance to terms of use and ethical guidelines.

Types of actions:

Search
Anyone regardless of registered or unregistered status can perform searches on the site. The feature can be accessed by clicking the search bar on the top. The search function allows users to filter out tutors based on the subject.

Data:

Topic
A subject is a type of class for tutors to identify their services with so that students know what the tutor specializes in. For example, a tutor may be for the subject English (ENG) and tutor students in reading and or writing. This will be a fixed list in our database.

Registration Record
Registered users inherit the permissions of unregistered users but may apply to be a tutor and send messages to tutors. They will have the following metadata in our database.

- ID Number
- Name
- Email
- Password (encrypted)

Tutor Record
Tutors are registered users that have applied to become a tutor and had their post approved by the administrator. They will have a table in our database with the following metadata.

- ID Number
- Name
- Topic
- Description
- CV/Flyer
- Resume

- Picture (Optional)
- Video (Optional)

Message
Messages will be shared to a tutor from a student, inquiring about their post to get more information. This will be form data, an input from a registered user and then made visible to the tutor on their dashboard after storing it to the database.

- Tutor ID (Prefilled)
- Sender ID (Prefilled)
- Date (Prefilled)
- Text (User input)

4.Initial list of functional requirements:

Non Registered Users:
1. Non registered users shall be able to have access to tutor information (pictures, videos, description, reviews)
2. Non registered users shall be able to filter tutors based on topic/subject e.g. MATH, ENG
3. Non registered users shall be able to narrow search results by searching course number
4. Non registered users shall be able to view all available tutors as default
5. Non registered users shall be able to register
6. Non registered users shall be able to apply to become a tutor, and will be prompted to register or log in
7. Non registered users shall be able to log in
8. Non registered users shall be able to benefit from our structured database system containing information on students, tutors, available times, dates, classes, and days.

Registered Users: (Will inherit all previous permissions listed above)
9. Registered users shall be able to draft a message to a tutor, but are prompted to register upon submitting
10. Registered users shall be able to click “Forgot Password”
11. Registered users shall be able to apply to be a tutor, submitting their information
12. Registered users shall be able to view dashboard, containing any messages received from other registered users
13. Registered users shall be able to upload a video
14. Registered users shall be able to upload a photo

15. Registered users shall be able to upload their CV, Flyer, or Resume
16. Registered users shall be able to send a message to a tutor
17. Registered users that apply to be a tutor shall be prompted to choose what topic(s) they tutor.
18. Registered users shall be able to log out of their account

Administrator: (Will inherit all previous permissions listed above)
19. Admin shall be able to view database
20. Admin shall be required to approve registered users to become tutors
21. Admin shall be able to reject user posts before they are published

5.List of non-functional requirements:

1. Application shall be developed, tested and deployed using tools and servers approved by
Class CTO and as agreed in M0
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers
3. All or selected application functions shall render well on mobile devices
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 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
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).

6.Competitive analysis:

Feature	Gator Wise	SFSU Navigator	OLA Tutoring	Wyzant
Tutors attend SFSU	++	+	-	-
Search by SFSU Class	++	+	-	-
Ease of registering for tutors	++	-	-	+
Lazy Login	++	-	+	+
Browse all tutors	++	-	+	+
Search tutors by class	++	-	-	-

** – Lacking Feature ** + Has Feature ** ++ Has Superior Feature

We will be implementing a search bar that features a drop down menu and takes input from users. The drop down menu will be a non-specific search by SFSU course topics to narrow down the list of tutors visible to the user,so if users do not know the name of tutors, they can still search by their subject, similar to other tutoring websites. The key feature of our application is that the subjects available to browse will pertain to courses offered at SFSU and tutors on our site will identify themselves with at least one, so that users are aware of their options for tutors. Additionally, users can narrow results with or without the subject dropdown by using the search bar to type in information about our tutors and it will filter results by comparing it to the tutors we have in our database. This information could be the name of the tutor, what classes they've taken, or what topics

are covered in certain courses. For example, a user may need help in MATH which is a subject that tutors can identify themselves with, but if the user wishes to search “Calculus” specifically, the site will find any tutors that have disclosed their experience in Calculus. The courses/subjects available will be specific to SFSU, which streamlines the experience for our users, allowing them to find exactly what they are looking for in a tutor with our advanced search feature.

7.High-level system architecture and technologies used:

Software Stack:

Server Host: AWS
Operating System: Ubuntu 16.2
Database: MySQL v8.0.34
Web Server: NGINX 1.24
Server-Side Language: JavaScript

Additional Technologies:

Web Framework: ExpressJS
IDE: VSCode
Web Analytics: Google Analytics
SSL Cert: Lets Encrypt (Cert Bot)
We plan to use ChatGPT: YES

Browsers:

Google Chrome
Mozilla Firefox

ChatGPT:
Not used for M1

8.Use of ChatGPT:

ChatGPT was not used for this milestone. The AI text to image generator provided by Canva was used in the creation of the Personas’ pictures.

9.Team and roles:

Joaquin Warren - Team Lead
Ronnie Huang - Github Master
Karl Moreno - Front-End Lead
Darien Banuelos - Front-End Team Member
Sean Ryan - Back-End Lead
Philip Karnatsevich - Back-End Team Member

10. Checklist:

So far all team members are engaged and attending team sessions when required

On Track

Team found a time slot to meet outside of the class

Done

Back end, Front end leads and Github master chosen

Done

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

Team reviewed class slides on requirements and use cases before drafting Milestone 1

On Track

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

On Track

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

DONE

Milestone 2:

SW Engineering CSC 648/848 Fall 2023
Wise Guidance And Tutoring Outreach Resource (wiseGATOR)
Team 6

Team Member	Role
Joaquin Warren jwarren3@mail.sfsu.edu	Team Lead
Ronnie Huang	Github Master
Karl Moreno	Front-end Lead
Darien Banuelos	Front-end Developer
Sean Ryan	Back-end Lead
Philip Karnatsevich	Back-end Developer

Milestone 2 Part I
10/14/23

Revision History

Date Submitted	10/14/23
Date Revised	

Milestone 2 Part 2
10/21/23

Revision History

Date Submitted	10/21/23
Date Revised	

1. Executive Summary

If you're a student, you're familiar with the struggle of juggling multiple classes and meeting overlapping deadlines. Time is money when you're in college, and with our new application, we can save our users both! We relate to the challenges students face when looking for convenient and affordable access to tutoring services on campus. That's why we started developing wiseGATOR, a brand new SFSU Guidance and Tutoring Outreach Resource. It's not only important to pass the classes, but to understand and retain the key concepts in each course. wiseGator allows SFSU students to find SFSU tutors relevant to their coursework, to get fast help and mentorship. SFSU students often depend on one another to get assistance with their classes and homework. This application will bridge the gap between students in need of tutoring services and experienced students that can offer specialized help.

Our application aims to quickly connect SFSU students to SFSU tutors and SFSU students interested in becoming tutors to paid mentorship roles. Students at SFSU are already footing the bill when it comes to tuition prices, and they're on the rise as of September 2023. We can save students time and money by skipping the Chegg membership or paying for Quizlet premium. Students can get 1 on 1, specific help with homework assignments, coursework, study guides, and more at a reasonable, cost-efficient rate. This is also a great opportunity for SFSU students that excel in their courses to make money by becoming a mentor and offering tutoring services on our site to other SFSU students. It's a one stop shop for any SFSU student seeking or offering expert help. Our goal is to streamline the process of finding tutoring, by allowing students to filter by subject and by courses offered at SFSU. This way, students are learning from one another in the same courses, and they do not have to rely on outdated and irrelevant internet forums or paid services from people that have never even been in that class. What makes our application so unique is that it is exclusive to San Francisco State University, whether you want to find a tutor or tutor others in any of the courses offered at SFSU. Searching by courses and subjects will optimize the experience for students seeking tutoring services, and SFSU tutors will be able to cater their service directly to SFSU students.

The team behind wiseGATOR is a group of 6 ambitious computer scientists at San Francisco State University. With a startup funded by Professor Petkovic of the College of Science & Engineering, this team is dedicated to bringing wiseGATOR to personal devices of SFSU students by the end of 2023. Team 6 has experienced team members with a strong background in both Web and Software Development. Our values are aligned with the interest of both students and professors alike to provide a successful experience for all of our users in their educational careers. It is important to us as students that we can make a meaningful positive impact on our peers and utilize our best talents to extend that to more students and alumni of SFSU.

2. List of Main Data Items and Entities

Users:

Unregistered User
An unregistered user is any person who is browsing the site without being logged into any account. They can browse all available tutors and perform searches. They can apply to be a tutor at any time but will be prompted to register.

Registered User
Unregistered users become a registered user when they sign up for an account or if they log in to an existing one. Registered users have all of the permissions an unregistered user has with the added bonus of being able to submit a message to a tutor and apply to be a tutor.

Admin
Admin is logged in to the web service or registered user, and is a person who is responsible for managing and supporting a tutoring web service. Admin has full access to all features and functionalities of the web service and its database. They will accept and reject tutor posts in accordance with terms of use and ethical guidelines.

Data:

Topic
A subject is a type of class for tutors to identify their services with so that students know what the tutor specializes in. For example, a tutor may be for the subject English (ENG) and tutor students in reading and or writing. This will be a fixed list in our database.

Registration Record
Registered users inherit the permissions of unregistered users but may apply to be a tutor and send messages to tutors. They will have the following metadata in our database.
- ID Number (Mandatory)
- Name (Mandatory)
- Email (Mandatory)
- Password (Mandatory & Encrypted)

Tutor Record
Tutors are registered users that have applied to become a tutor and had their post approved by the administrator. They will have a table in our database with the following metadata.
- ID Number (Mandatory)
- Name (Mandatory)
- Topic (Mandatory)
- Description (Mandatory)
- CV/Flyer (Optional)
- Resume (Optional)
- Picture (Optional)

● Video (Optional)

Message
Messages will be shared to a tutor from a student, inquiring about their post to get more information. This will be form data, an input from a registered user and then made visible to the tutor on their dashboard after storing it to the database.
● Tutor ID (Prefilled & Mandatory)
● Sender ID (Prefilled & Mandatory)
● Date (Prefilled & Mandatory)
● Text (User input & Mandatory)

3. Functional Requirements - Prioritized

Priority 1

o Unreg user

1. Shall be able to have access to all tutor information
2. Shall be able to view all available tutors as default
3. Shall be able to filter tutors based on topic/subject e.g. MATH, ENG
4. Shall be able to narrow search results by typing in search bar
5. Shall be able to register
6. Shall be able to apply to become a tutor, and will be prompted to register or log in

o Reg user

7. Shall be able to apply to be a tutor, submitting their information
8. Shall be able to upload a video
9. Shall be able to upload a photo
10. Shall be able to upload their CV, Flyer, or Resume
11. Shall be able to log in
12. Shall be able to log out of their account
13. Shall be able to message a tutor of their choice
14. Shall be able to access their messages and tutor posts via dashboard
15. Shall be prompted to choose what topic(s) they tutor when applying as tutor
16. Shall be able to write a bio for themselves inside of the description field

o Admin

17. Shall be able to view database
18. Shall be required to approve registered users to become tutors before they go live
19. Shall be able to reject user posts before they are published

Priority 2

o Unreg user

o Reg user

- Registered users shall be able to draft a message to a tutor

- Registered users shall be able to view dashboard, containing any messages received from other registered users
-

o Admin
- Admins shall be able to view all privately sent message between registered users
- Admins shall be able to delete users from database
- Admins shall respond to issues with our website.

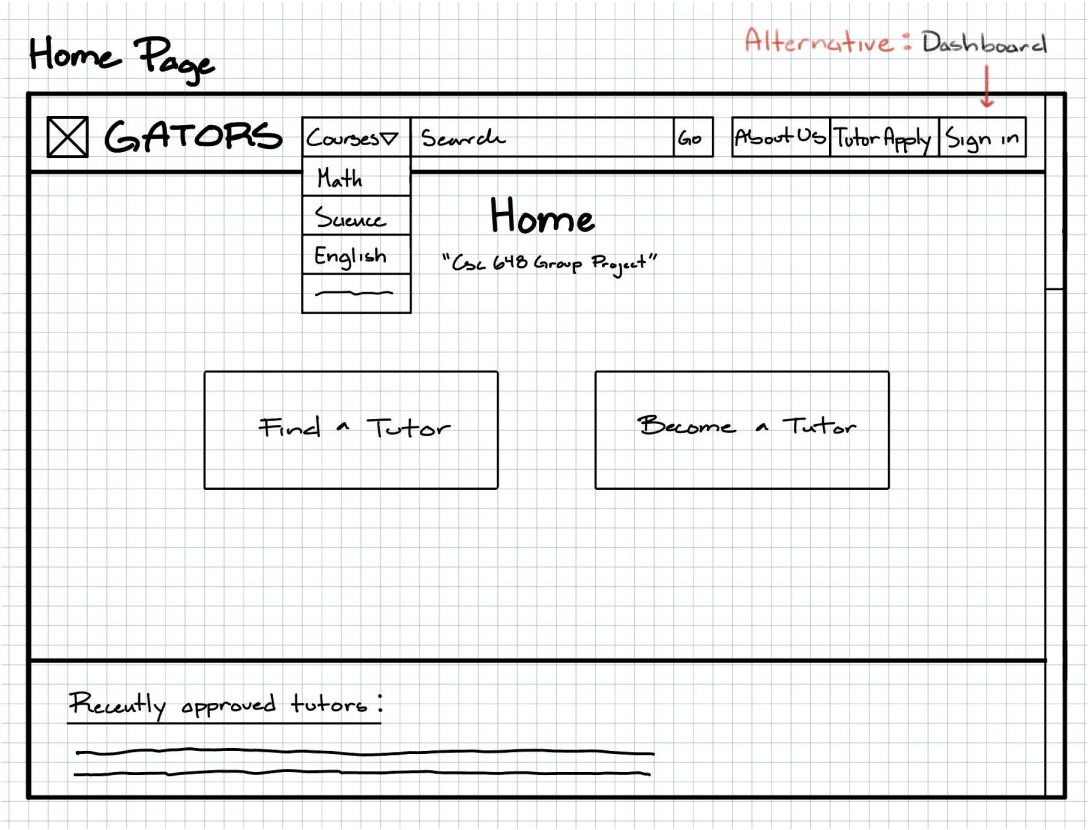
Priority 3

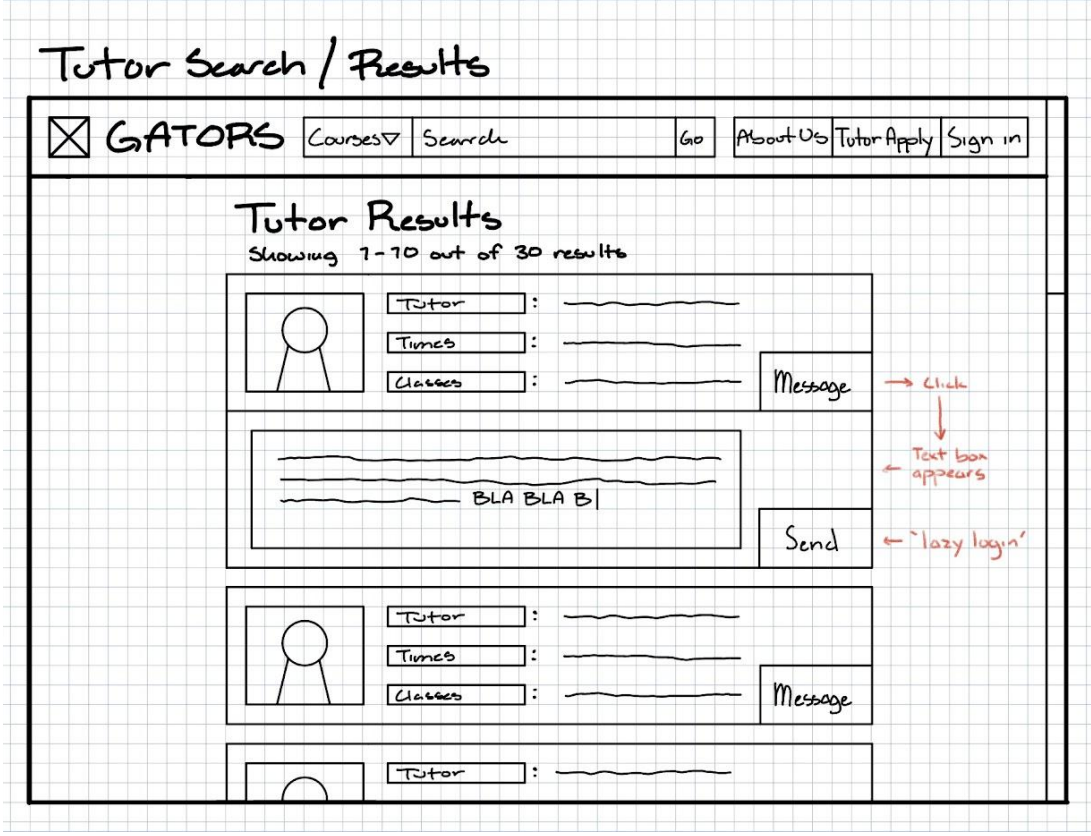
o Unreg user

o Reg user
- Registered users shall be able to submit reviews for the site that will be included in the tutors search result
- Registered users shall be able to apply to be admin
- Registered users shall be able to report inappropriate behavior
- Registered users shall be able to give a rating for a tutor

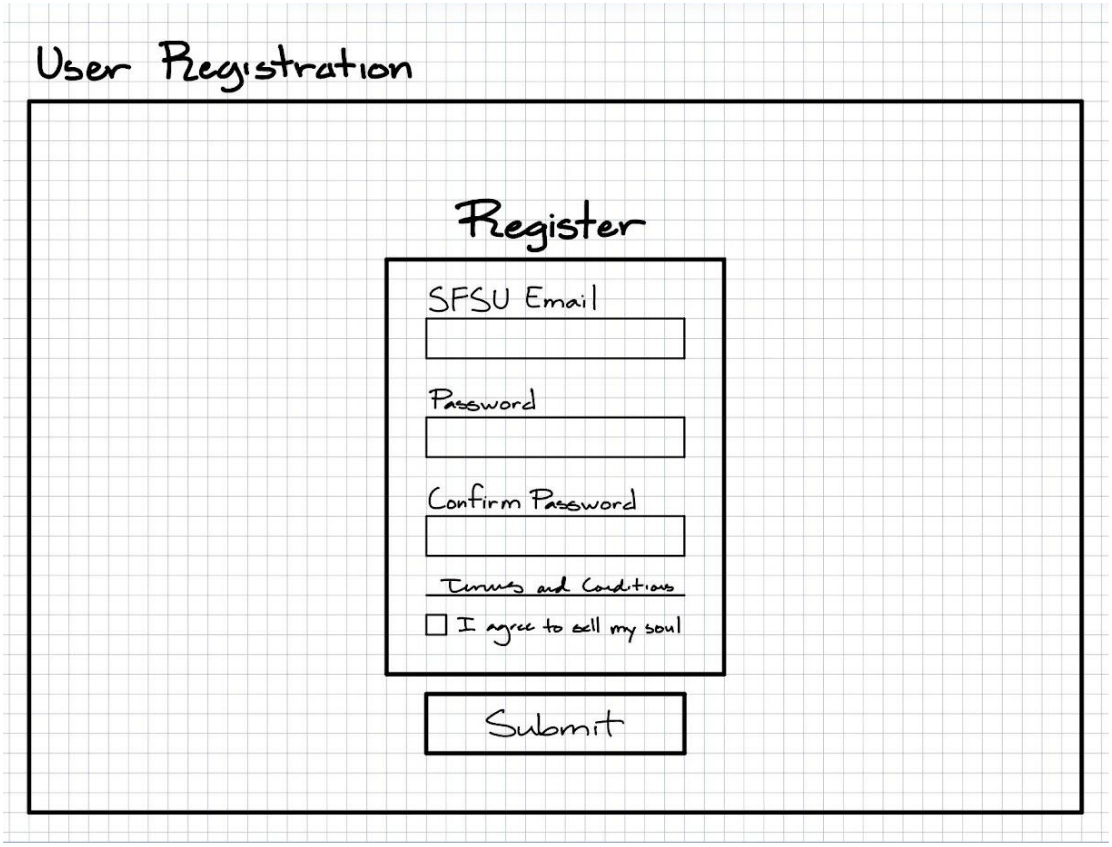
o Admin
- Admin shall be able to choose what posts gets placed on the front page
- Admin shall be able to post their own announcements on the front page
- Admin shall be able to approve requests from registered users to be admin

4. UI Storyboards for Each Main Use Case

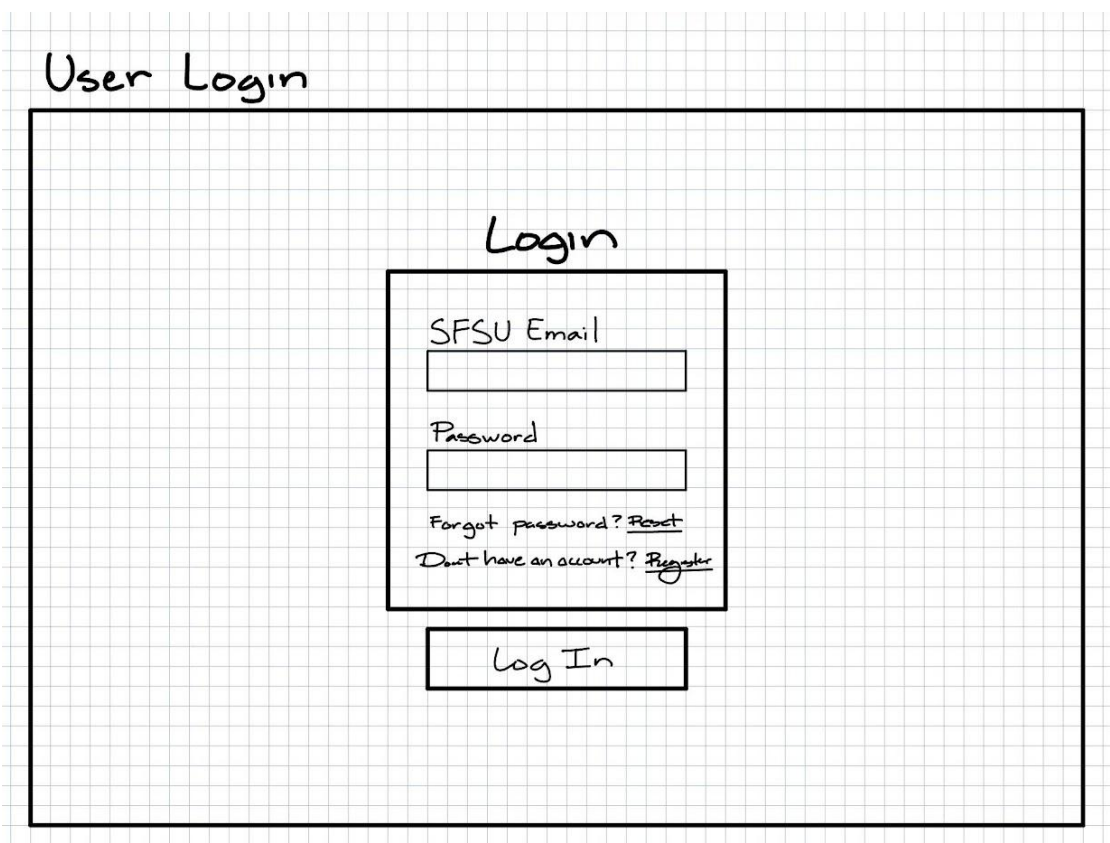




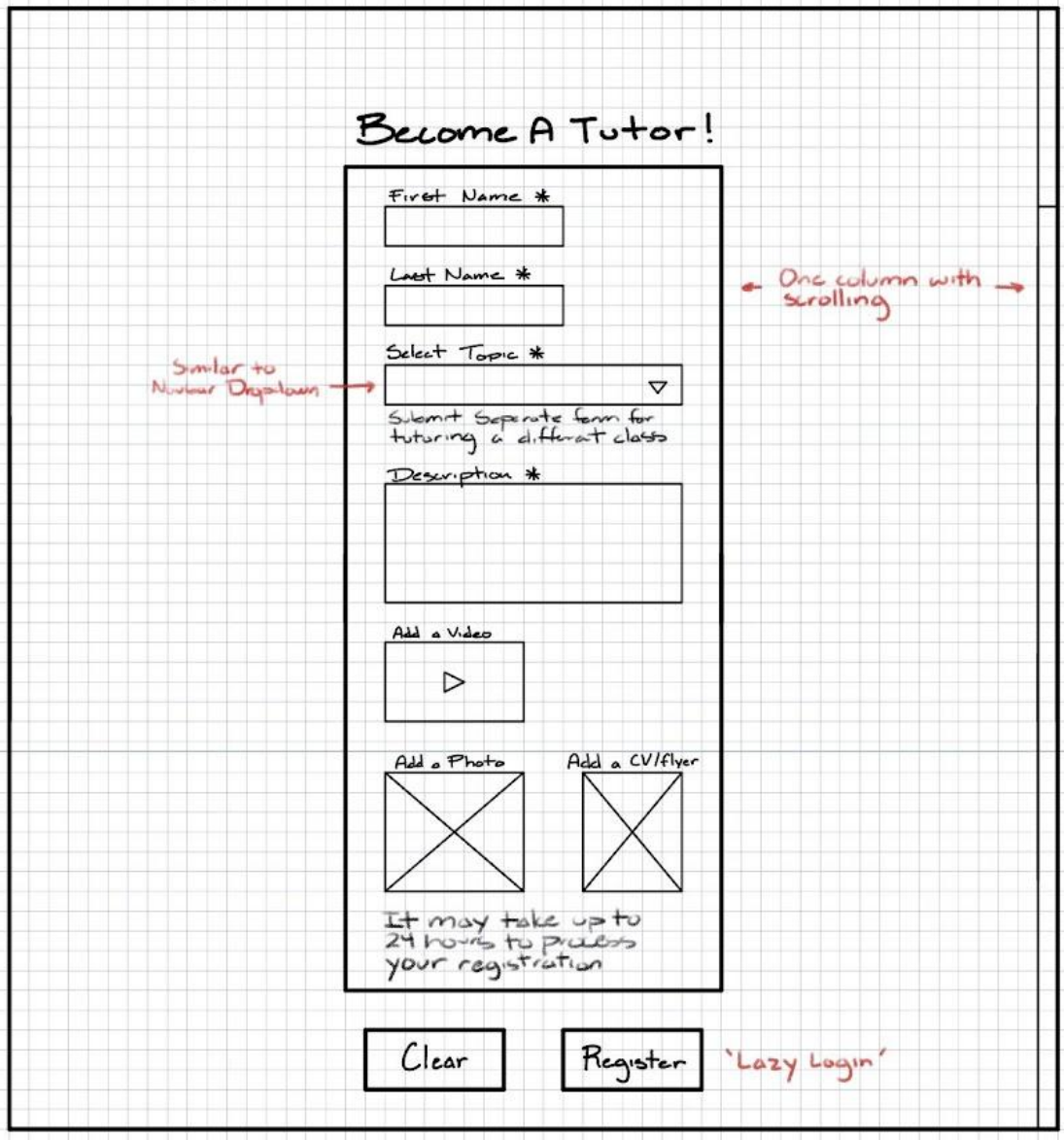
User Registration

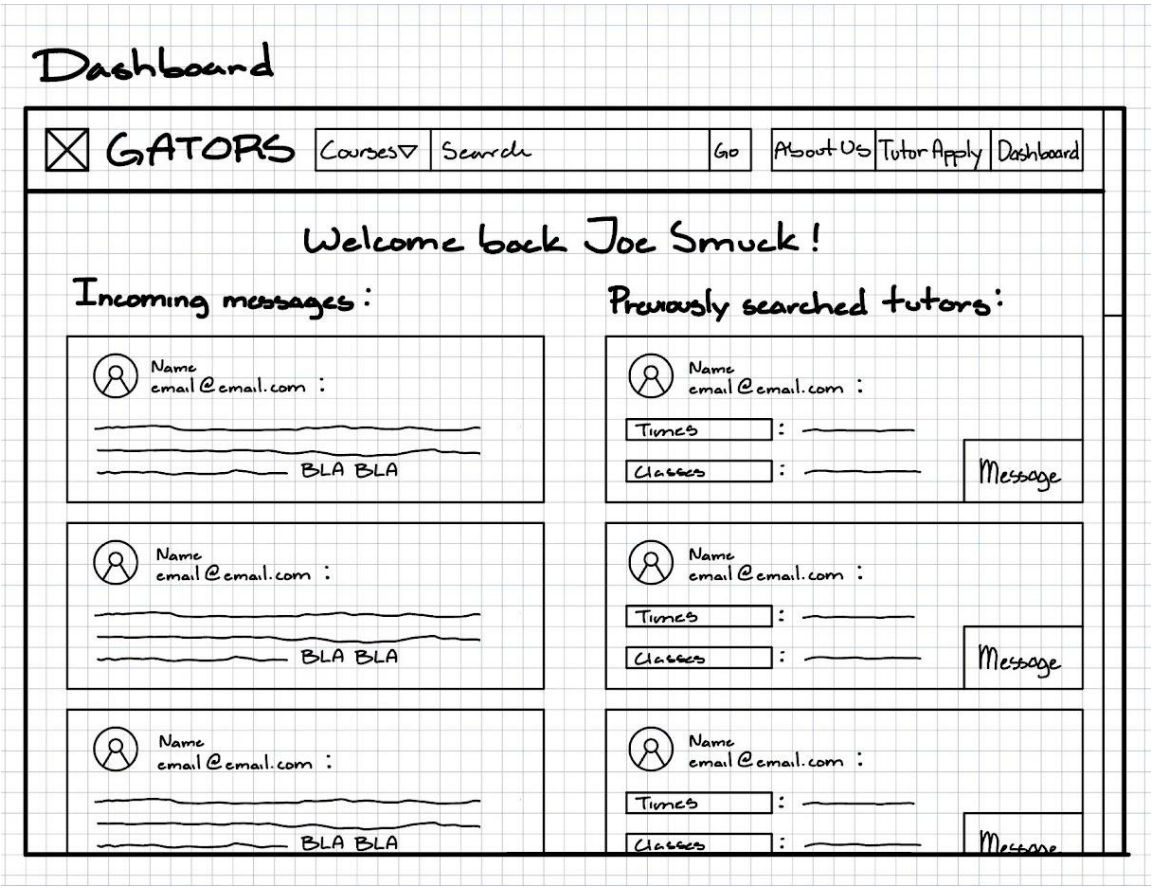


User Login



Tutor Application





5. High Level Architecture, Database Organization summary

*We will be using the utf8mb4 character set.

Tables:

Topic

- Subject (VARCHAR(4))

Registered Users

- ID Number (INT)
- Name (VARCHAR(100))
- Email (VARCHAR(254))
- Password (BINARY(20)) *Stores the hash of the password using SHA-1

Tutors

- ID Number (INT)
- Name (VARCHAR(100))
- Topic (VARCHAR(4)) *A value from Topic Table
- Description (TEXT)
- CV/Resume/Flyer (VARCHAR(50)) *Points to pdf in file

- Picture (VARCHAR(50)) *Points to png in file
- Video (VARCHAR(50)) *Points to mp4 in file

Messages
- Tutor ID (INT)
- Sender ID (INT)
- Tutor Post ID (INT)
- Date(TIMESTAMP)
- Message (TEXT)

We will save pdfs, videos, and images in corresponding files on the server. The file name will be saved as a string and saved to our MySQL database. We will utilize relative pointers to files in our secure database of images and videos.

We are using MySQL and %like for our search functionality.

We are not currently planning on using any other algorithms.

We decided to add Bootstrap to our software stack. This front-end framework will be very helpful for our front-end team and will ensure that our application is responsive on mobile.

6. Key Risks for the Project

Schedule Risk:
As we make progress towards the delivery of the product, we are also on a timeline filled with other deadlines and obligations as it grows later into the calendar year. There are many holidays this semester and possible obligations besides this project arise in October, November, and December. We all have other classes with projects, homework, assignments, midterms, and more. Then we also have to make sure we still have time for our family and friends during Thanksgiving and the holidays.
Realistically, and thanks to the professor's guidance, we as a team of 6 will definitely be able to manage a project of this scale. We will keep the scope to a feasible degree but include all requirements. The way we can successfully deliver on time is by delegating manageable parts to each team member and playing to our strengths. This would mean team members use their best skills and talents and apply them to those areas of the project, ensuring quality and efficiency.

7. Project Management

At this time, the team delegates tasks to each team member based on their role and proficiency in certain areas. The project is very feasible in the timeframe given that we have very talented

team members and many resources available to us. Our team utilizes discord as our main message channel but that does not stop us from also following up 1 on 1 to track progress towards each milestone. The team lead is very comfortable allowing each team member space to work outside of our meetings on their parts and everyone is accountable. However, with advice from the Professor, we will be trying out Trello to improve our workflow. Trello will have three categories of tasks; To Do, Doing, and Done. This will be a good way to visually understand what tasks are available, who is primarily assigned to them, and when they are due. We use soft, smaller deadlines to get us to the overall goal of delivering what is expected of us. Our in class meetings after lecture are also very productive, and we meet almost every week outside of class to make sure everyone is caught up.

8. Use of ChatGPT

We have not used ChatGPT for M2.

9. Milestone 2 Part 2

Item	Credentials
Website URL	http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/
Website URL to Search Page	http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/search-tutors?subject=&search=
SSH URL	ssh ubuntu@ec2-3-139-70-137.us-east-2.compute.amazonaws.com
SSH Username	ubuntu
SSH Password/Key	GatorsTeam6!
Database URL	gatordb.ckz0o6p7rp5j.us-east-2.rds.amazonaws.com port 3306
Database Username	GatorTeam6
Database Password	GatorBytes6!
Link to Github page that performs the search	https://github.com/CSC-648-SFSU/csc648-03-fa23-team06/blob/HTML-SEARCH-REVAMP/application/webpage/search-tutors/search-tutors.html

Additional Information
Please explain what is working for your prototype and what can be tested. Also, what search terms to Use. Our prototype is near completion, however my team ran into an issue with our front-end. Our data is accessible however we are unable to perform the actions with the onclick event listeners although they are set up for use. Incorporating the buttons to be the trigger so that our data is not dynamic is the main issue we are facing. My team has been working very diligently on our vertical prototype and they still have questions that we would like to consult our CTO for. The search bar is present and there is no CSS at the moment because the team focused more on functionality.

Milestone 3:

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

Team number: 6

Meeting date: 11/15 (6:30 PM - 7:00 PM)

- Summary of feedback on UI (record all pages that need revision)
    - Home: Clickable Logo
        - Move the main home logo up, and some tutors right under to indicate scrolling is possible
        - Become a tutor can be the same button in nav
    - Search Results: Smaller cards, a bit more text, move message closer;
        - Display quantity of tutors;
        - Clickable tutors that open a new tab with more info
        - Popup window to send message, prefilled fields, in gray text: leave contact info + sendmessage
    - Register Tutor: Star mandatory fields
        - Make everything one column scrollable page, apply not register
    - Register: Move star to top right
    - Dashboard: Display quantity of messages, pre sort by date
        - Instead of prev searched tutors show users' sent messages
- Summary of feedback on code and architecture
    - Add header comments to each file
    - Include more inline comments that do not state the obvious or just indicate what each block of code is/does
- Summary of feedback on github usage
    - When adding fixes, be more specific in the commit message ***what*** was fixed
    - Instead of "Persistent Search" write "Made Search Persistent"
- Summary of feedback on DB
    - N/A
- Summary of feedback on teamwork and risk management
    - N/A
- 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
    - More documentation to be included moving forward and also to document what has been completed so far.
- List below agreed upon P1 list of features for final delivery which constitute product plan. NOTE: after this meeting the team focuses solely on this P1 list of features, e.g. the

development is in “feature freeze mode”. All listed P1 features (no more no less) MUST be delivered in usable way, free of bugs
    - Sending Messages to Tutors
    - Be able to Register
    - Be able to Login to an existing account
    - Show messages in dashboard for signed in users
- Any other comments and issues
    - Make the logo clickable and have it refresh the page
    - Change Sign in to Log in
    - Add a disclaimer that the website is for a class project and not a real tutoring website
    - Specify what type of website it is using text under the main page’s WiseGator brand
    - WiseGator should continuously play its extra text, last longer, or be switched to static
    - 40 character limit in search fields and other places with input
- Check Point (CP) if given, DUE:
    - N/A

IMPORTANT for selection of P1 features:

- analyze what needs to be done, prioritize based on two factors
    - importance for the product/user
    - cost/ability to deliver it in a given schedule
- Based on this come up with the plan (list of P1 features) then execute it
- After this the team is in “feature freeze” mode, focus is on P1 features only

Milestone 4:

SW Engineering CSC 648/848 Fall 2023
WiseGATOR
Team 6

Role	Team Members
Team Lead	Joaquin Warren jwarren3@mail.sfsu.edu
Github Master	Ronnie Huang
Front-end Lead	Karl Moreno
Front-end Developer	Darien Banuelos
Back-end Lead	Sean Ryan
Back-end Developer	Philip Karnatsevich

Milestone 4

12/13/23

History Table

Date of Submitted	12/13/23
Date Revised	

1) Product Summary

WiseGATOR (Guidance And Tutoring Outreach Resource)

Our application aims to quickly connect SFSU students to SFSU tutors and SFSU students interested in becoming tutors to paid mentorship roles. Students can get 1 on 1, specific help with homework assignments, coursework, study guides, and more at a reasonable, cost-efficient rate. This is also a great opportunity for SFSU students that excel in their courses to make money by becoming a mentor and offering tutoring services on our site to other SFSU students. It's a one stop shop for any SFSU student seeking or offering expert help. Our goal is to streamline the process of finding tutoring, by allowing students to filter by subject and by courses offered at SFSU. This way, students are learning from one another in the same courses, and they do not have to rely on outdated and irrelevant internet forums or paid services from people that have never even been in that class. What makes our application so unique is that it is exclusive to San Francisco State University, whether you want to find a tutor or tutor others in any of the courses offered at SFSU. Searching by courses and subjects will optimize the experience for students seeking tutoring services, and SFSU tutors will be able to cater their service directly to SFSU students.

1. Unregistered users shall be able to have access to view all tutors
2. Unregistered users shall be able to view all available tutors as default
3. Unregistered users shall be able to filter tutors based on topic/subject e.g. MATH, ENG
4. Unregistered users shall be able to narrow search results by typing in search bar
5. Unregistered users shall be able to register
6. Unregistered users shall be able to apply to become a tutor, and will be prompted to register or log in
7. Registered users shall be able to apply to be a tutor, submitting their information
8. Registered users shall be able to upload a video
9. Registered users shall be able to upload a photo
10. Registered users shall be able to upload their CV, Flyer, or Resume
11. Registered users shall be able to log in
12. Registered users shall be able to log out of their account
13. Registered users shall be able to message a tutor of their choice
14. Registered users shall be able to access their messages and tutor posts via dashboard
15. Registered users shall be prompted to choose what topic(s) they tutor when applying as tutor
16. Registered users shall be able to write information about themselves inside of the description field
17. Admin shall be able to view database
18. Admin shall be required to approve registered users to become tutors before they go live
19. Admin shall be able to reject user posts before they are published

Link to Website: http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/

2) Usability test plan for selected function:

Test Objectives: We will be writing a usability test plan for the searching functionality of our application. We will test the ability to find relevant tutors because it will be one of two main services offered through our application. Finding tutors is a feature that both students and students that want to offer tutoring may both need, since tutors can also be tutored by other students in subjects or courses they need help with.

Test Background and Setup: To set up our test, each tester will need an end system with an interface and the ability to input. Some examples are a smartphone and its screen or a laptop with a screen and keyboard. Each tester will begin at our website's landing page, what we refer to as the homepage. This is where users are to perform the intended test of searching for a tutor.

Our intended users are students at San Francisco State University, not specifically in any area of study. We want to offer our services to as many students as we can, because it increases the chances of students applying for tutor jobs that have good experience to help other students. This means that tutors get more business, making more money from using our website.

Our users will start with this link, and open it on their preferred internet browser: http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/

Users will conduct this usability test from any location they'd like that has a reliable connection to the internet. They can do it from home or on campus at San Francisco State University. There will not be any cameras or other ways to monitor students as they test our website's searching feature other than a team member to proctor. Their purpose is not to help but just to count clicks and keep track of time. There will not be any required training for the students since they are our primary intended users and are quite familiar with simple search bar interfaces. Our search bar is intentionally not intricate so that it is very friendly for all future users, whether tech savvy or not.

Plan for Evaluation of Effectiveness: In order to measure effectiveness, we would need some figures to help us see if our usability is effective. The main way to do this is to see if a user was able to search for a tutor and find one relevant to their search. If the user was able to complete the task, or in other words, find a tutor, we take the quantity of successful searches and divide it by unsuccessful searches. This will give us a percent of users that found the search functionality effective. A good way to validate search effectiveness cases is to give them the name of a tutor to find, so we don't count cases where someone searched for a non-existent tutor, an impossible task. Given the tutor exists, we can conduct that test with multiple tutors to get a more accurate percentage of successful tests.

Plan for Evaluation of Efficiency: To measure efficiency, we can measure two different quantitative factors being number of clicks, and time. To be more specific, the number of clicks can be recorded for each user up until they find the target tutor. The time taken will be exactly

the minutes and seconds taken to reach the result, which is to see the tutor that we assigned them to search for. The time taken and the number of clicks can both be averaged to get a sense of how efficient our search feature is. This test is best conducted on users that are new to the website and haven't used it or seen it before. They can be the same test users as the evaluation for effectiveness, but a member of our team will count clicks and time the test.

Plan for Evaluation of User Satisfaction:

We will instruct users to do the following: Search for and identify a tutor on our website. After this we will have them fill out the following Likert scales to assess user's subjective thoughts on our website's effectiveness, efficiency, and overall satisfaction.

The search feature on this website is easy to use (Circle One)

Strongly Agree Agree Neutral Disagree Strongly Disagree

The search bar has an intuitive design (Circle One)

Strongly Agree Agree Neutral Disagree Strongly Disagree

I am satisfied with my experience using WiseGATOR (Circle One)

Strongly Agree Agree Neutral Disagree Strongly Disagree

3) QA test plan and QA testing:

a) Our usability test will be that our search functionality can find tutors from our database that are related to the text entered in our search bar. Testers will need the appropriate hardware and software in order to test searching. It will require an end-system that has a way to input text such as a laptop and keyboard. It will also require the following software; latest version of OS such as Windows or Mac, the most recent version of a browser of their choice (examples include Google Chrome, Safari, Firefox). Lastly, they will need the link to access our website:
http://ec2-3-14-250-152.us-east-2.compute.amazonaws.com:3000/
QA Tester will navigate to the link above on their preferred browser with their personal device. Each QA Tester will then conduct the following tests on our website by using the search bar we implemented.

Test # browser used	Test Title	Test Description	Test Input	Expected Output	Result

1	Search Tutor	The tester only inputs a tutor topic, while leaving the search bar to be nothing.	On the home page, other than the "About Me" pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying "WiseGATOR"). Make sure the gray box says "Select Topic". If not, click on said box, and a white box should show up below it. In the white box, press "Select Topic". To the right hand side of the gray box, there should be a longer white box. Press it, and then type “Sean”. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display Sean Ryan, with the subject ID of “CSC”.	
2	Search Topic	The tester only inputs a name on the search bar, while leaving the tutor topic to be nothing.	On the home page, other than the "About Me" pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Once you find the gray box, click on said box, and a white box should show up. In the white box, press "ARTH". To the right hand side of the gray box, there should be a longer white box. Make sure the white box is empty. If not, press it, and then delete all the text inside it. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display Darien Banuelo, with the subject ID of “ARTH”.	
3	Search Tutor with Topic	The tester inputs both a name on the search bar and a tutor topic.	On the home page, other than the “About Me” pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Once you find the gray box, click on said box, and a white box should show up. In the white box, press "DM". To the right hand side of the gray box, there should be a longer white box. Press it, and then type “Ka”. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display only Karl Moreno, with the subject ID of “DM”.	
4	Empty Search	The tester leaves both the	On the home page, other than the “About Me” pages, go to the top of the page, and locate the first gray box to	The search result	

		search bar and tutor topic to be nothing.	the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Make sure the gray box says "Select Topic". If not, click on said box, and a white box should show up below it. In the white box, press "Select Topic". To the right hand side of the gray box, there should be a longer white box. Make sure the white box is empty. If not, press it, and then delete all the text inside it. After which, press “Search”, which is on the right hand side of the search bar.	should display every tutor on the site.	

b) Tested on the following browsers: Chrome, Firefox, Microsoft Edge

Test # With Google Chrome	Test Title	Test Description	Test Input	Expected Output	Result
1	Search Tutor	The tester only inputs a tutor topic, while leaving the search bar to be nothing.	On the home page, other than the "About Me" pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying "WiseGATOR"). Make sure the gray box says "Select Topic". If not, click on said box, and a white box should show up below it. In the white box, press "Select Topic". To the right hand side of the gray box, there should be a longer white box. Press it, and then type “Sean”. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display Sean Ryan, with the subject ID of “CSC”.	Pass
2	Search Topic	The tester only inputs a name on the search bar, while leaving the tutor topic to be nothing.	On the home page, other than the "About Me" pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Once you find the gray box, click on said box, and a white box should show up. In the white box, press "ARTH". To the right hand side of the gray box, there should be a longer white box. Make sure the white box is empty. If not, press it, and then	The search result should display Darien Banuelo, with the subject ID of “ARTH”.	Pass

			delete all the text inside it. After which, press “Search”, which is on the right hand side of the search bar.		
3	Search Tutor with Topic	The tester inputs both a name on the search bar and a tutor topic.	On the home page, other than the “About Me” pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Once you find the gray box, click on said box, and a white box should show up. In the white box, press "DM". To the right hand side of the gray box, there should be a longer white box. Press it, and then type “Ka”. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display only Karl Moreno, with the subject ID of “DM”.	Pass
4	Empty Search	The tester leaves both the search bar and tutor topic to be nothing.	On the home page, other than the “About Me” pages, go to the top of the page, and locate the first gray box to the left in the top bar (on the right hand side of the text saying “WiseGATOR”). Make sure the gray box says "Select Topic". If not, click on said box, and a white box should show up below it. In the white box, press "Select Topic". To the right hand side of the gray box, there should be a longer white box. Make sure the white box is empty. If not, press it, and then delete all the text inside it. After which, press “Search”, which is on the right hand side of the search bar.	The search result should display every tutor on the site.	Pass

4) Peer Code Review:

Code Review Request for Search-Tutors

You replied on Sun 12/10/2023 7:02 PM

SR Sean Michael Ryan
To: Joaquin Nguyen Warren

Sun 12/10/2023 5:32 PM

Good Evening Team Leader!

I am sending you links to my code for the search-tutors page and functionality for review. I used embedded java script to display the search results to the page and connected it to the database with a java script file. All advice and criticism is greatly appreciated!

https://github.com/CSC-648-SFSU/csc648-03-fa23-team06/blob/development/application/views/search-tutors.ejs

GitHub

Build software better, together

GitHub is where people build software. More than 100 million people use GitHub to discover, fork, and contribute to over 420 million projects.

github.com

https://github.com/CSC-648-SFSU/csc648-03-fa23-team06/blob/development/application/routes/index.js

GitHub

Build software better, together

GitHub is where people build software. More than 100 million people use GitHub to discover, fork, and contribute to over 420 million projects.

github.com

On the last link the implementation starts on line 241.

Thank you,
Sean Ryan
-Back End Team Lead

Reply Forward

Re: Code Review Request for Search-Tutors

JW

Joaquin Nguyen Warren

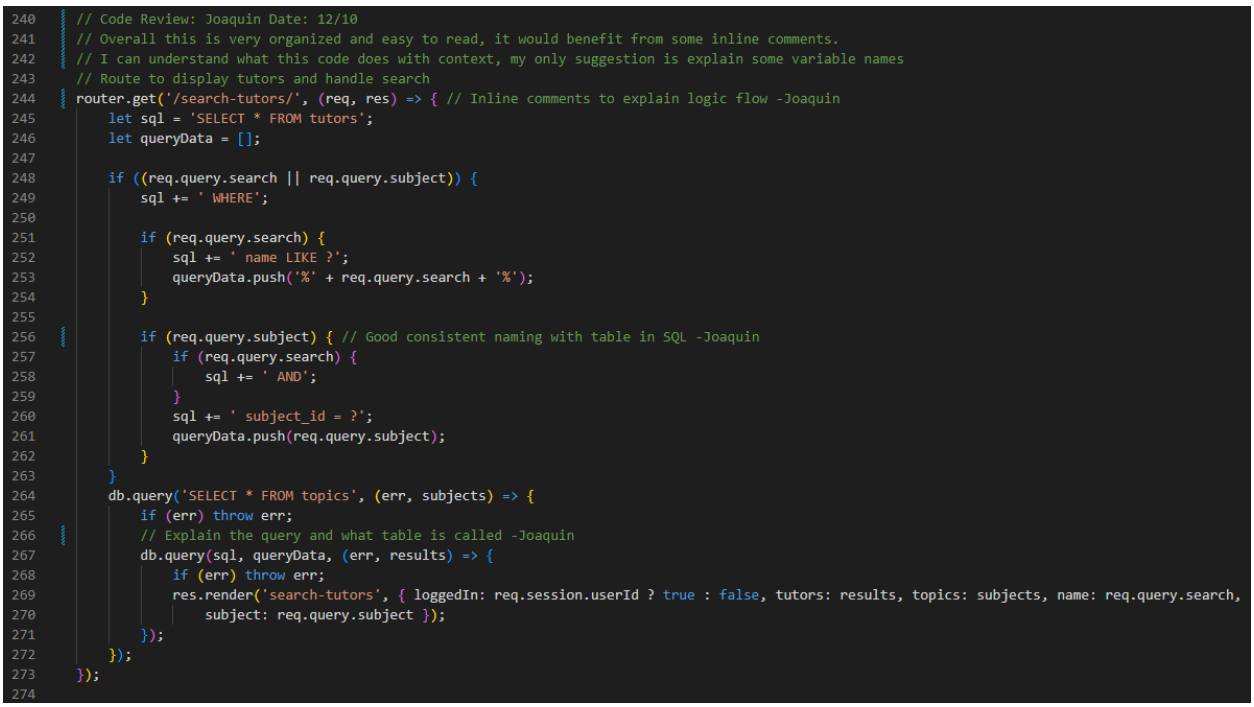
To: Sean Michael Ryan

Sun 12/10/2023 7:02 PM

Hey Sean!

Thanks for reaching out to me, I spent some time reviewing your code to give you something to work with in terms of feedback.

Overall, great work on the search function for our application. Other than it being very functional, I can tell right away that this code is very refined and organized. I have no trouble at all understand the context of each file. My only feedback is to make sure to include in-line comments in a couple parts of the code to further explain variable names and elaborate on the way the code between the files interacts with one another. I've included some screenshots of these opportunities below:



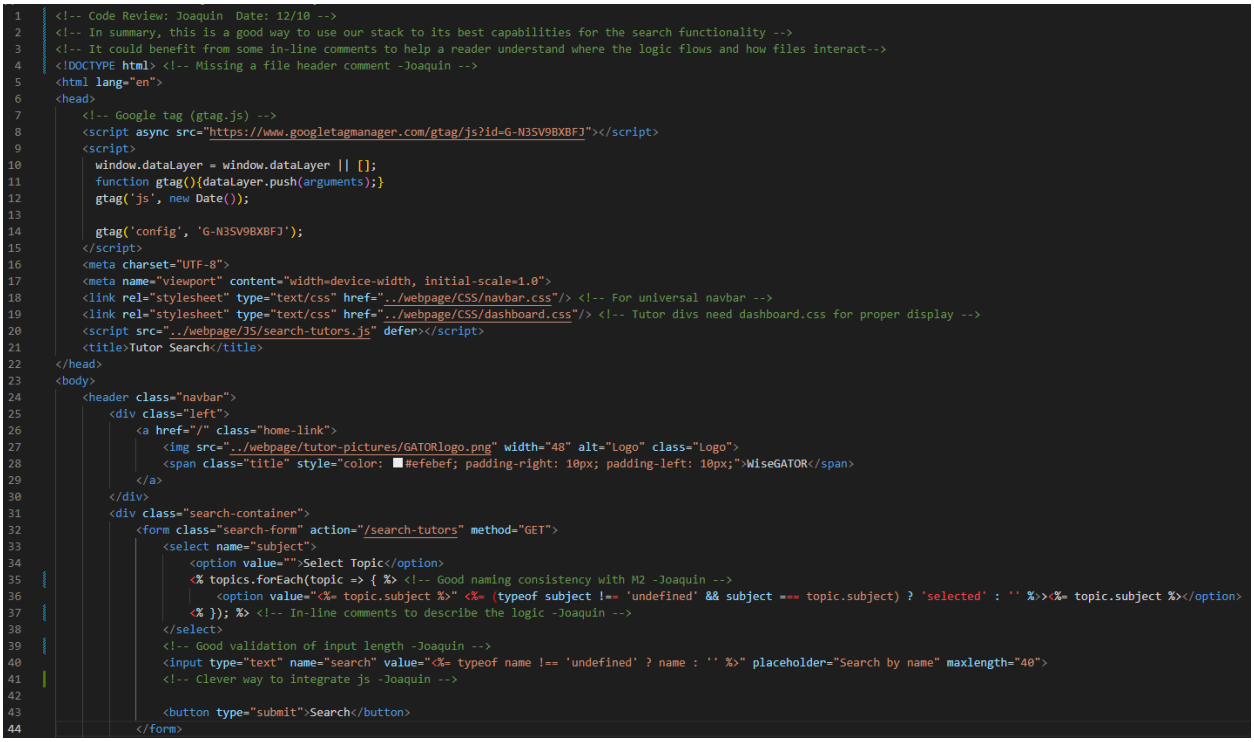
```
// Code Review: Joaquin Date: 12/10
// Overall this is very organized and easy to read, it would benefit from some inline comments.
// I can understand what this code does with context, my only suggestion is explain some variable names
// Route to display tutors and handle search
router.get('/search-tutors/', (req, res) => { // Inline comments to explain logic flow -Joaquin
    let sql = 'SELECT * FROM tutors';
    let queryData = [];

    if ((req.query.search || req.query.subject)) {
        sql += ' WHERE';

        if (req.query.search) {
            sql += ' name LIKE ?';
            queryData.push('%' + req.query.search + '%');
        }

        if (req.query.subject) { // Good consistent naming with table in SQL -Joaquin
            if (req.query.search) {
                sql += ' AND';
            }
            sql += ' subject_id = ?';
            queryData.push(req.query.subject);
        }
    }
    db.query('SELECT * FROM topics', (err, subjects) => {
        if (err) throw err;
        // Explain the query and what table is called -Joaquin
        db.query(sql, queryData, (err, results) => {
            if (err) throw err;
            res.render('search-tutors', { loggedIn: req.session.userId ? true : false, tutors: results, topics: subjects, name: req.query.search,
                subject: req.query.subject });
        });
    });
});

```



```
<!-- Code Review: Joaquin  Date: 12/10 -->
<!-- In summary, this is a good way to use our stack to its best capabilities for the search functionality -->
<!-- It could benefit from some in-line comments to help a reader understand where the logic flows and how files interact-->
<!DOCTYPE html> <!-- Missing a file header comment -Joaquin -->
<html lang="en">
<head>
    <!-- Google tag (gtag.js) -->
    <script async src="https://www.googletagmanager.com/gtag/js?id=G-N3SV9BXBFJ"></script>
    <script>
      window.dataLayer = window.dataLayer || [];
      function gtag(){dataLayer.push(arguments);}
      gtag('js', new Date());

      gtag('config', 'G-N3SV9BXBFJ');
    </script>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" type="text/css" href="../webpage/CSS/navbar.css"/> <!-- For universal navbar -->
    <link rel="stylesheet" type="text/css" href="../webpage/CSS/dashboard.css"/> <!-- Tutor divs need dashboard.css for proper display -->
    <script src="../webpage/JS/search-tutors.js" defer></script>
    <title>Tutor Search</title>
</head>
<body>
    <header class="navbar">
        <div class="left">
            <a href="/" class="home-link">
                <img src="../webpage/tutor-pictures/GATORlogo.png" width="48" alt="Logo" class="Logo">
                <span class="title" style="color: #efebef; padding-right: 10px; padding-left: 10px;">WiseGATOR</span>
            </a>
        </div>
        <div class="search-container">
            <form class="search-form" action="/search-tutors" method="GET">
                <select name="subject">
                    <option value="">Select Topic</option>
                    <% topics.forEach(topic => { %> <!-- Good naming consistency with M2 -Joaquin -->
                        <option value="<%= topic.subject %>" <%= (typeof subject !== 'undefined' && subject === topic.subject) ? 'selected' : '' %>><%= topic.subject %></option>
                    <% }); %> <!-- In-line comments to describe the logic -Joaquin -->
                </select>
                <!-- Good validation of input length -Joaquin -->
                <input type="text" name="search" value="<%= typeof name !== 'undefined' ? name : '' %>" placeholder="Search by name" maxlength="40">
                <!-- Clever way to integrate js -Joaquin -->

                <button type="submit">Search</button>
            </form>
```

Thank you for taking the time to ensure the code's quality. Please keep up the hard work and feel free to message me if you have any questions.

Best,
Joaquin Warren
-Team Lead

---

5) Self-check on best practices for security:

Asset to Protect	**Type of Possible Attack**	**Strategy to Mitigate or Protect Asset**
User Passwords	Cyber Attack/Hacking	Password encryption
Database (containing student emails, passwords,	SQL Injection	Search bar authentication prevent form submission if non-alphanumeric characters are present, only allows up to 40 characters
AWS Console: Credit Card info, Billing info (address)	Account Hacking	Root User/Admin Login Credentials are not shared with anyone, including team members
Source Code	Plagiarizing Code, Using our source code without giving	Keep all ideas within our company, don't share with

	credit to earn profit	competitors. Get a patent for ideas we originally create
Trademarked/Copyrighted Content: Logo, Name of Application	Stolen content	Add our own copyright or trademark to all materials we originally created
Customer Safety and Security	Spam bots, malicious tutor bots that are not real people and want to fish on our website with links and obtain payment information	Registration authentication that checks for SFSU emails, and tutors must be approved by admin before published

6) Self-check of the adherence to original Non-functional specs:

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
DONE

4. Data shall be stored in the database on the team's deployment server.
DONE

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

6. Privacy of users shall be protected
DONE

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

8. Application shall be very easy to use and intuitive
DONE

9. Application shall follow established architecture patterns
DONE

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

11. Google analytics shall be used
DONE

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
DONE

13. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
DONE

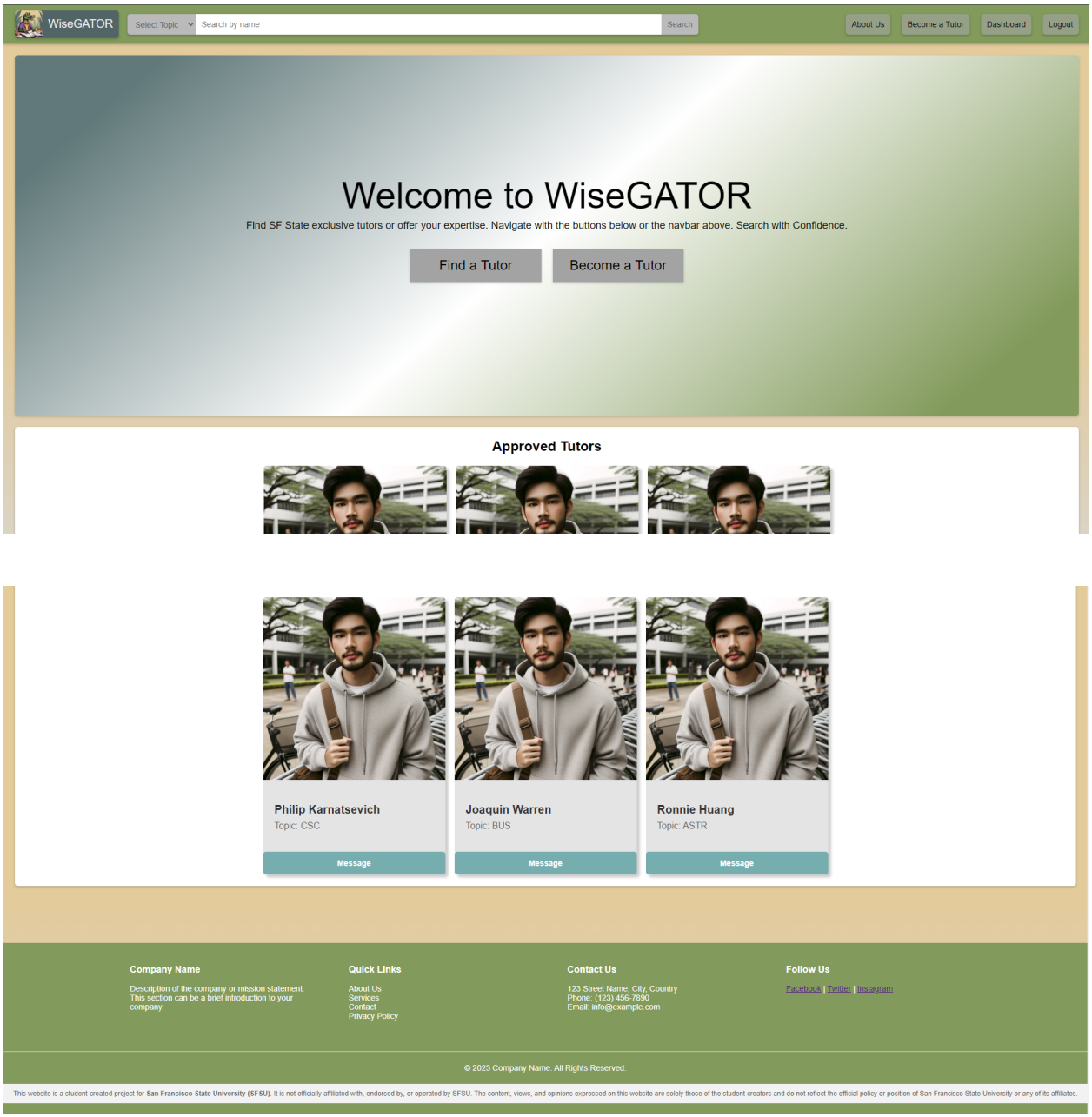
14. Site security: basic best practices shall be applied (as covered in the class) for main data items
DONE

15. Media formats shall be standard as used in the market today
DONE

16. Modern SE processes and tools shall be used as specified in the class, including collaborative and continuous SW development
DONE

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).
DONE

3) Product Screenshots:



WiseGATOR | Select Topic | Search by name | Search | About Us | Become a Tutor | Dashboard | Logout

9-8:20 Search Results Returned

Philip Karnatsevich
Subject ID: CSC

WiseGATOR | Select Topic | Search by name | Search | About Us | Become a Tutor | Dashboard | Logout

Dashboard

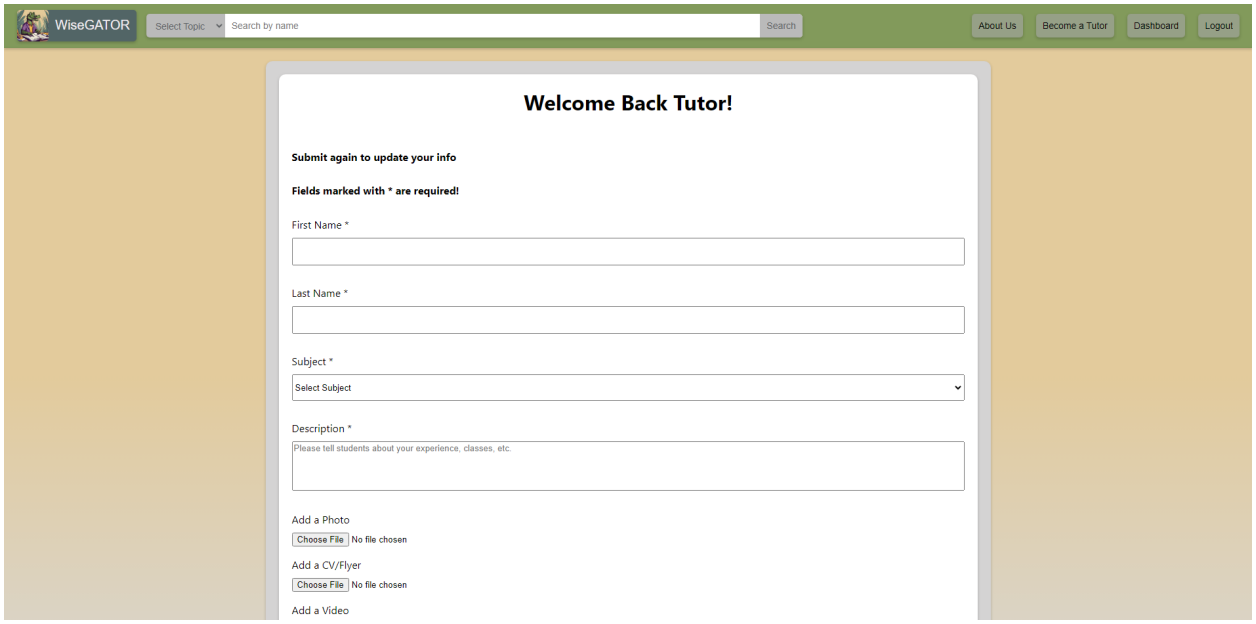
Welcome Back, Joaquin

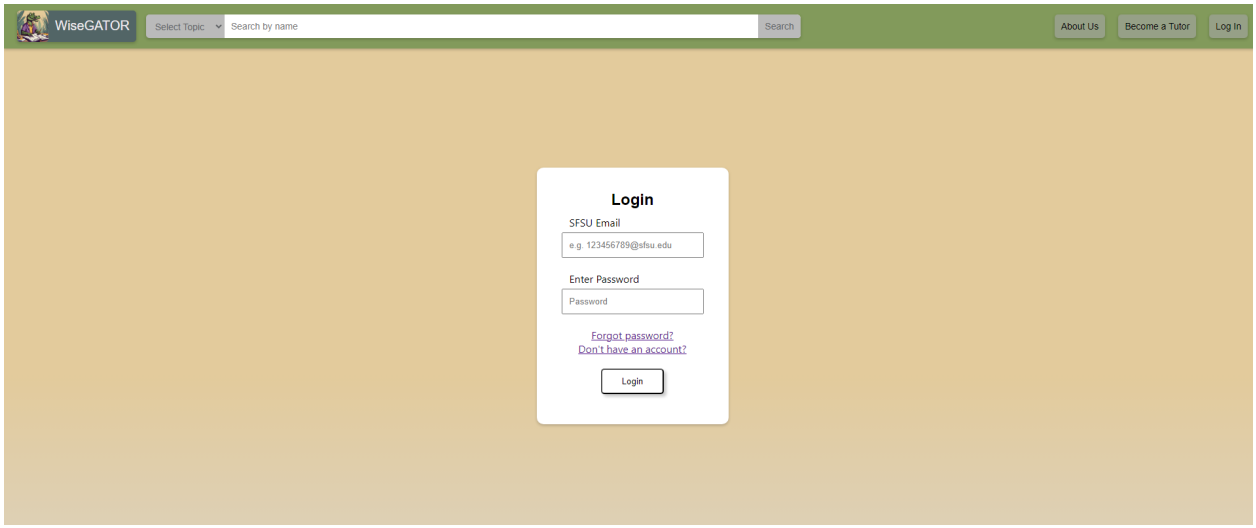
You have 1 Messages:

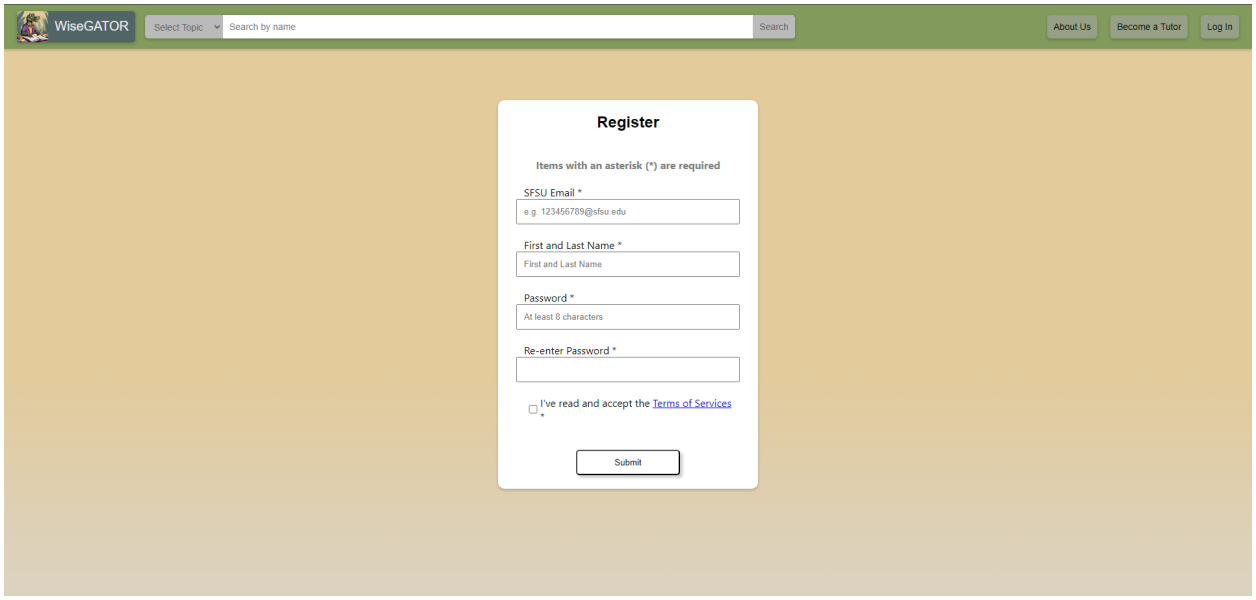
Mon Dec 11 2023 20:05:16 GMT+0000 (Coordinated Universal Time)
From: Sean Ryan
Email: sryan@sfsu.edu
Hey I would like some tutoring please!

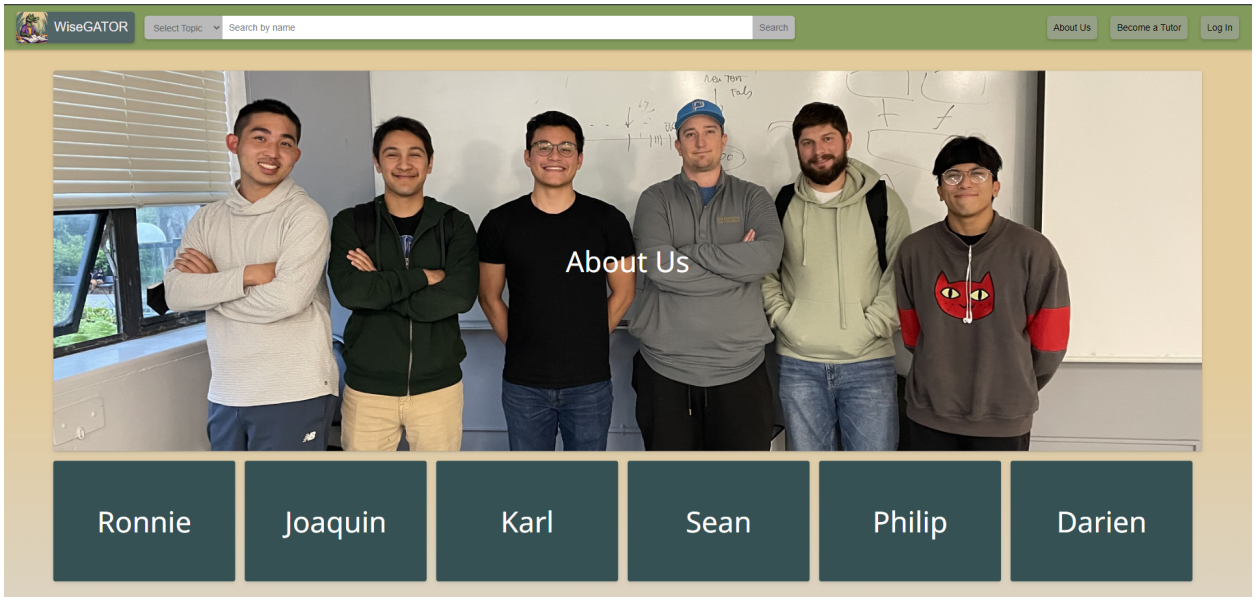
Sent Messages:

Tue Dec 12 2023 16:37:01 GMT+0000 (Coordinated Universal Time)
To: Sean Ryan
Topic: CSC
Hey! Can you tutor me?
Click here for additional tutor related Video, Image, or CV/Flyer


WiseGATOR
Select Topic
Search by name
Search
About Us
Become a Tutor
Dashboard
Logout
Welcome Back Tutor!
Submit again to update your info
Fields marked with * are required!
First Name *
Last Name *
Subject *
Select Subject
Description *
Please tell students about your experience, classes, etc.
Add a Photo
Choose File
No file chosen
Add a CV/Flyer
Choose File
No file chosen
Add a Video


WiseGATOR
Select Topic
Search by name
Search
About Us
Become a Tutor
Log In
Login
SFSU Email
e.g. 123456789@sfsu.edu
Enter Password
Password
Forgot password?
Don't have an account?
Login


WiseGATOR
Select Topic
Search by name
Search
About Us
Become a Tutor
Log In
Register
Items with an asterisk (*) are required
SFSU Email *
e.g. 123456789@sfsu.edu
First and Last Name *
First and Last Name
Password *
At least 8 characters
Re-enter Password *
I've read and accept the Terms of Services *
Submit


WiseGATOR
Select Topic
Search by name
Search
About Us
Become a Tutor
Log In
About Us
Ronnie
Joaquin
Karl
Sean
Philip
Darien

4) Database Organization:

Screenshots for Key Database Tables

Tutors

id	name	subject_id	description	flyer	picture	video	user_id
1	Philip Karnatsevich	CSC	NULL	NULL	tutor_1	NULL	1
2	Joaquin Warren	BUS	NULL	NULL	tutor_2	NULL	2
3	Ronnie Huang	ASTR	NULL	NULL	tutor_3	NULL	3
4	Karl Moreno	DM	NULL	NULL	tutor_4	NULL	4
5	Darien Banuelos	ARTH	NULL	NULL	tutor_5	NULL	5
6	Sean Ryan	CSC	NULL	NULL	tutor_6	NULL	6
7	Joe Schmuck	MATH	NULL	NULL	tutor_7	NULL	7
8	John Doe	CSC	NULL	NULL	tutor_8	NULL	8
9	yoyo	CSC	NULL	NULL	NULL	NULL	11
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Users

id	name	email	password
1	Philip Karnatsevich	philip karnatsevich@example.com	BLOB
2	Joaquin Warren	joaquin warren@example.com	BLOB
3	Ronnie Huang	ronnie huang@example.com	BLOB
4	Karl Moreno	karl moreno@example.com	BLOB
5	Darien Banuelos	darien banuelos@example.com	BLOB
6	Sean Ryan	sean ryan@example.com	BLOB
7	Joe Schmuck	joe schmuck@example.com	BLOB
8	John Doe	john doe@example.com	BLOB
10	Sean Ryan	sryan@sfsu.edu	BLOB
11	yoyo	masterseanm@gmail.com	BLOB
NULL	NULL	NULL	NULL

Topics

subject
ARTH
ASTR
BUS
CSC
DM
MATH
NULL

Messages

TutorID	SenderID	TutorPostID	Date	Message
1	11	1	2023-12-07 16:17:08	Tutoring at SFSU enhances learning experiences.
2	11	2	2023-12-07 21:17:48	SFSU tutoring provides personalized academic s...
3	11	3	2023-12-07 21:17:57	Effective tutoring strategies are key at SFSU.
1	11	4	2023-12-07 21:34:39	SFSU tutors are well-versed in their subjects.
6	11	5	2023-12-07 21:35:02	Tutoring sessions at SFSU are highly interactive.
1	11	6	2023-12-09 15:05:27	SFSU tutoring services cater to diverse learning...
3	11	7	2023-12-09 15:26:58	Peer tutoring at SFSU fosters collaborative lear...
5	11	8	2023-12-09 15:31:52	SFSU tutors help in understanding complex topics.
5	11	9	2023-12-09 15:32:08	Regular tutoring at SFSU improves academic pe...
5	11	10	2023-12-09 15:32:24	SFSU tutoring sessions are flexible and accomm...
1	11	11	2023-12-09 15:37:43	Tutoring at SFSU contributes to overall student ...
6	11	12	2023-12-09 15:50:43	SFSU tutors are trained to assist in various subj...
1	11	13	2023-12-09 15:53:49	Tutoring at SFSU is a resource for exam prepar...
1	11	14	2023-12-09 15:55:18	SFSU's tutoring program is dedicated to student...
6	11	15	2023-12-09 15:55:29	Tutoring at SFSU helps in building academic conf...
6	11	16	2023-12-09 15:57:59	SFSU tutors encourage critical thinking and prob...
9	11	17	2023-12-09 16:48:47	Tutoring at SFSU is tailored to individual learnin...
4	10	18	2023-12-09 19:06:24	SFSU tutoring is a great way to clarify doubts.
9	10	19	2023-12-09 21:20:07	The tutoring center at SFSU offers a supportive...
NULL	NULL	NULL	NULL	NULL

5) Github organization:

a) <u>The List of Branches:</u>

Team Lead & Github Master

- Main

Team Lead, Front End Lead, & Front End Developer

- CSS

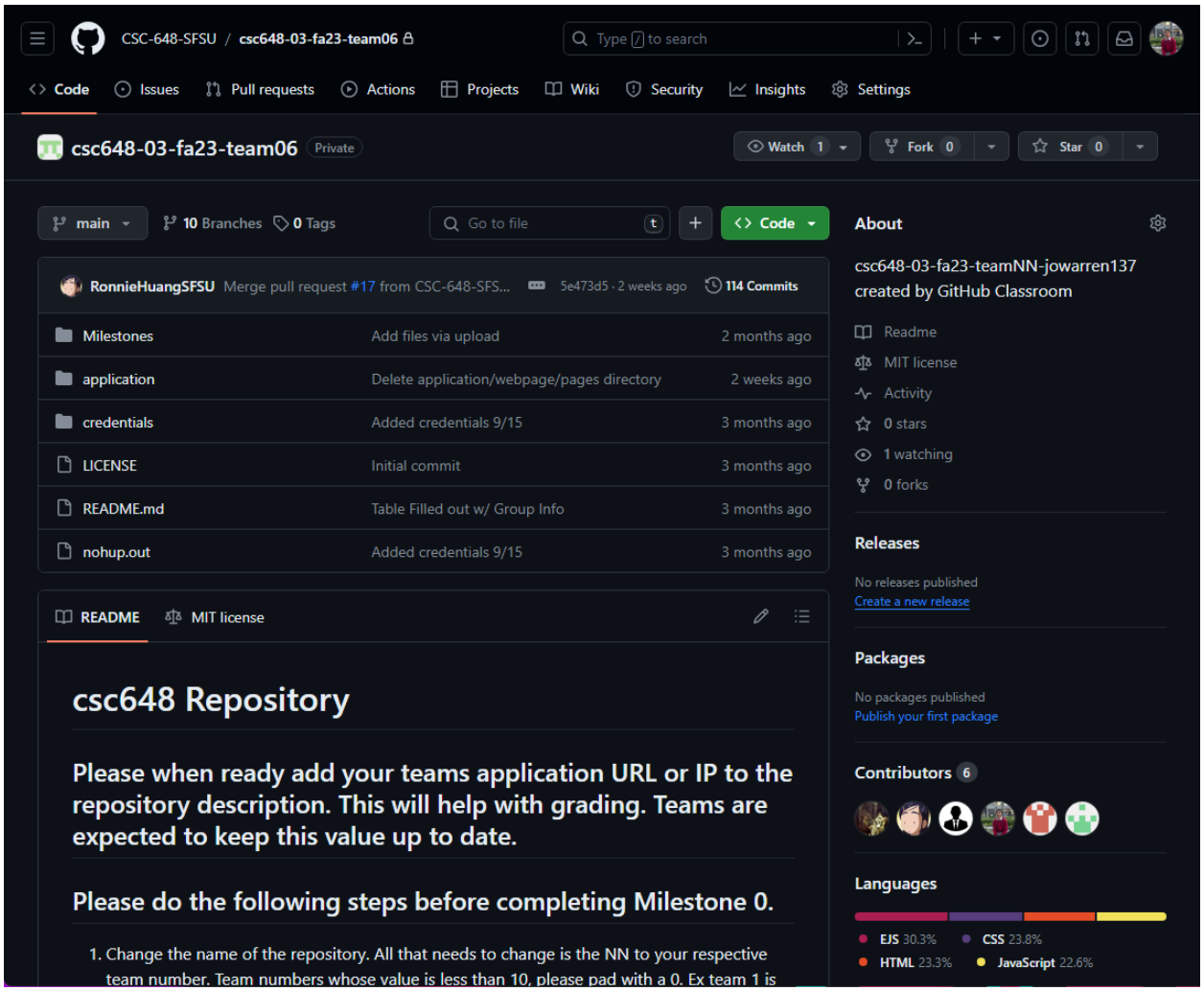
Entire Team

- HTML-SEARCH-REVAMP
- M3draft
- New-Pages
- Search-and-Navigate
- development
- Jowarren137-patch-1

Team Lead, Back End Lead, & Back End Developer

- Login-and-register
- messaging

b) Screenshots of our Github homepage



README MIT license

csc648 Repository

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

Please do the following steps before completing Milestone 0.

1. Change the name of the repository. All that needs to change is the NN to your respective team number. Team numbers whose value is less than 10, please pad with a 0. Ex team 1 is Team01 team 11 is Team11. Please make sure to also remove the username from the repository as well. Teams with incorrectly name repository will have points deducted from their milestone 0 grades.
    - Please follow the naming convention assigned by your instructor.
2. PLEASE REMOVE THE USERNAME FROM THE REPOSITORY NAME!!!
3. Add ALL members of your team to this repository. For it to count, they must ACCEPT the invite.
4. Fill out the table below

Student Name	Student Email	GitHub Username
Joaquin Warren	jwarren3@mail.sfsu.edu	jowarren137
Ronnie Huang	rhuang11@mail.sfsu.edu	RonnieHuangSFSU
Darien Banuelos	dbanuelos1@mail.sfsu.edu	Darienban
Philip Karnatsevich	pkarnatsevich@sfsu.edu	kapitoshcka
Karl Moreno	kmoreno6@sfsu.edu	karlMoreno
Sean Michael Ryan	sryan@sfsu.edu	SeanMichaelRyan

No releases published
Create a new release

Packages

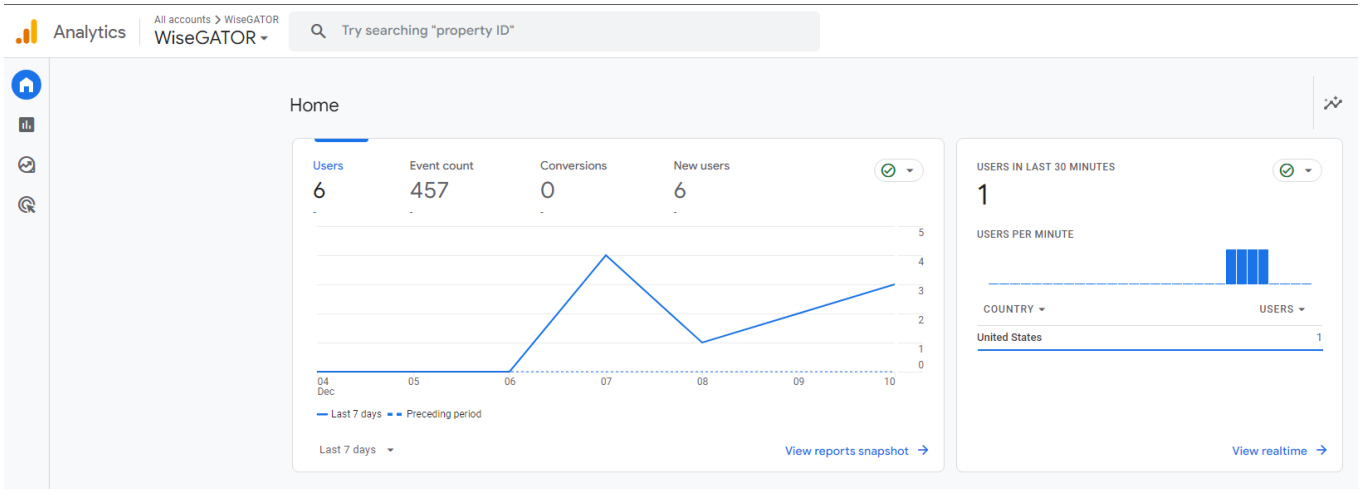
No packages published
Publish your first package

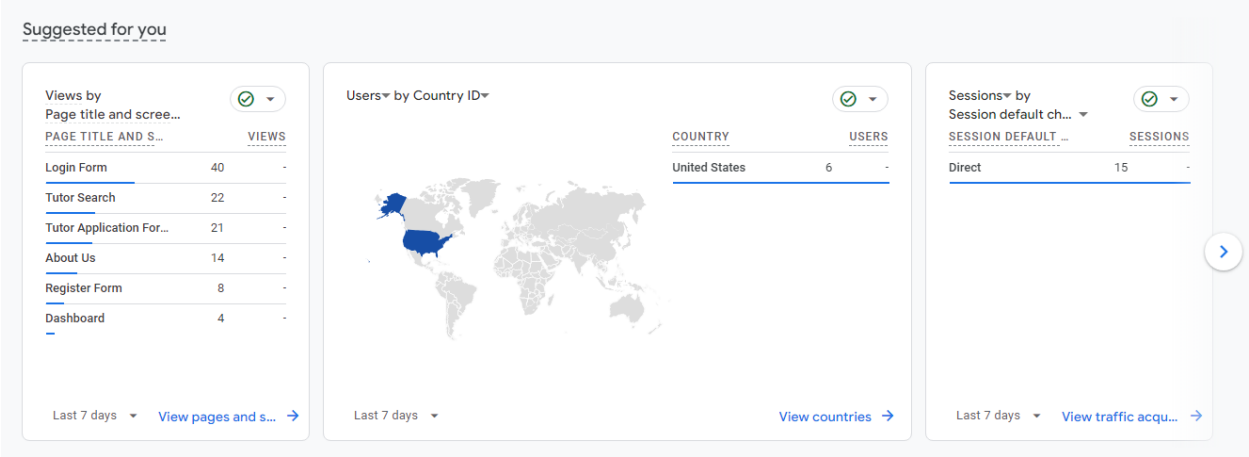
Contributors 6

Languages

EJS 30.3% CSS 23.8% HTML 23.3% JavaScript 22.6%

6) Google analytics stats plot for your WWW site:

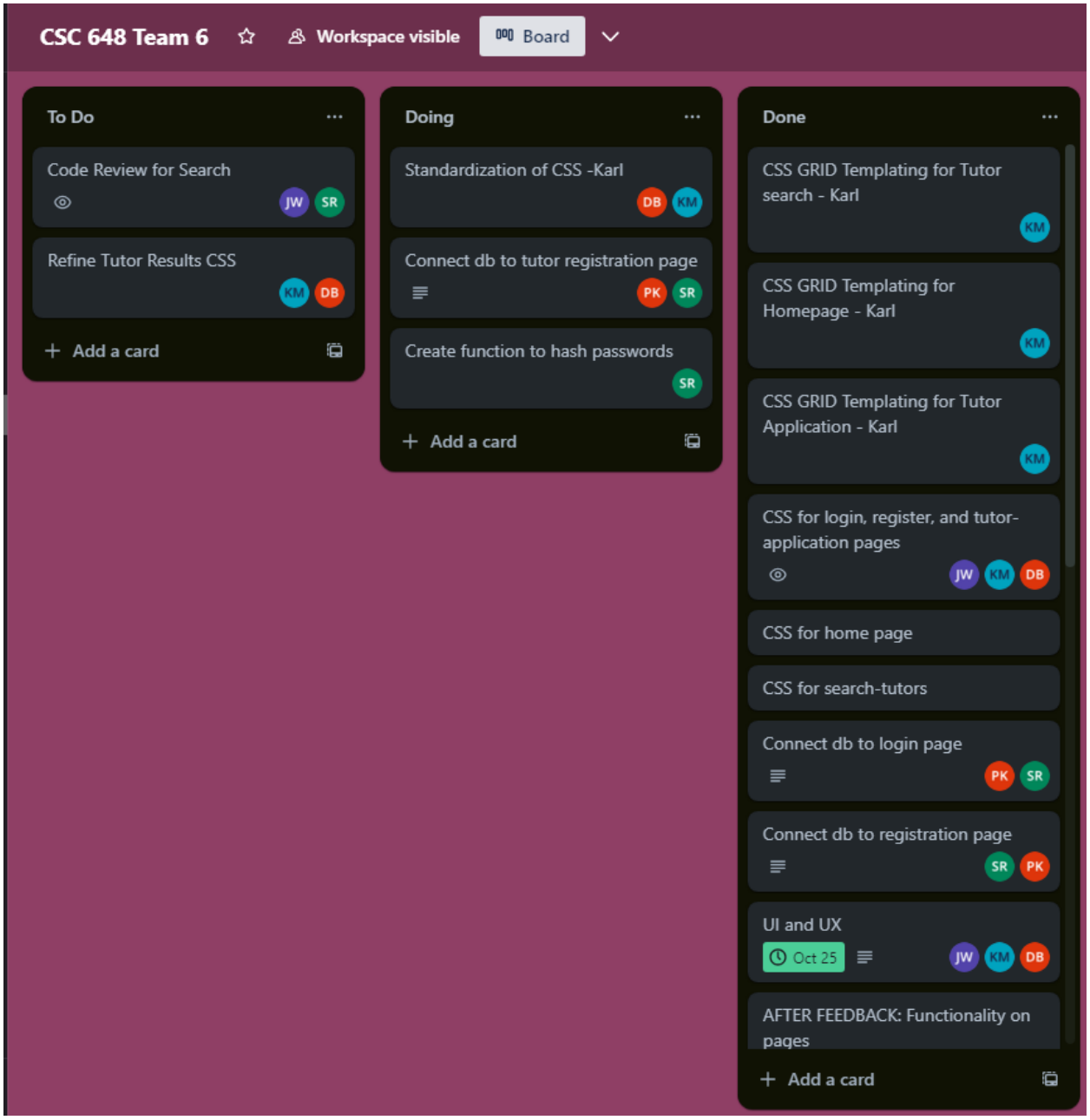




7) Project management:

The tools we used for project management and communication were Trello and Discord. We used Trello to delegate tasks to the team, explicitly decide when these short term goals were due, and for individual accountability to see when we start, when we are in the process of, and when we finish our tasks.

Discord was used for messaging, communication, and meetings for the team. We conducted each of our remote meetings weekly on discord in a voice call with screen sharing to view the same content in real time.


CSC 648 Team 6
Workspace visible
Board
To Do
Code Review for Search
JW SR
Refine Tutor Results CSS
KM DB
Add a card
Doing
Standardization of CSS -Karl
DB KM
Connect db to tutor registration page
PK SR
Create function to hash passwords
SR
Add a card
Done
CSS GRID Templating for Tutor search - Karl
KM
CSS GRID Templating for Homepage - Karl
KM
CSS GRID Templating for Tutor Application - Karl
KM
CSS for login, register, and tutor-application pages
JW KM DB
CSS for home page
CSS for search-tutors
Connect db to login page
PK SR
Connect db to registration page
SR PK
UI and UX
Oct 25
JW KM DB
AFTER FEEDBACK: Functionality on pages
Add a card

8) Team member self assessment and contributions:

JW Joaquin Nguyen Warren

To: Sean Michael Ryan; Philip Karnatsevich; Karl Christopher Moreno; Darien Thuan Banuelos; Ronnie Y. L. Huang Sun 12/10/2023 6:48 PM

a) Contributions to team project and teamwork:

- Organized and hosted meeting times and content to review for entire team
- Appointed team members to roles respective of their experience and interest
- Delegated tasks to other members of the team
- Created and maintained the AWS EC2 server with a process manager
- Created project name and logo (brand)
- Lead the Discord Server with announcements, resources, and communication for team
- Project milestone submissions and directly communicated with Project Manager
- Consulted CTO for software stack approval and recommendations
- Drafted files for main html pages of website
- Provided Code review and feedback for iterative revision development
- Followed up with documentation in all areas of project

b) Number of submissions to GitHub team Dev. Branch:

29

c) Main Challenges Encountered in Team Project:

One of the main challenges I encountered in the group project was handling unexpected errors whenever I tried to do something for the first time. Setting up the AWS server even with the tutorial provided by the professor was a little bit challenging. It was something that I had not ever done before, and I was very conscious of making mistakes because it could result in charges being made to my own credit card. Additionally, I had to put a lot of my own personal information in that server and be cautious with my credentials. The end result turned out great after using the internet a lot to resolve most of the issues, but at first, I could not tell if I was done or not with the tutorial alone. The more specific the instructions were with the team project later in the semester, the more comfortable I felt tackling them with my team.

On the note of understanding what was expected from the team. As the team leader, I had pressure to make sure anything we submitted was our best effort. This meant asking for deadlines when appropriate or even in anticipation of unexpected emergencies. It was my duty to have each of my members on the same page and when communication got a little fuzzy that is where complications could arise. If there were any questions left unanswered or even unasked, it could slow down development. I always followed up after explaining tasks or assigned work so that we all had the same vision in mind when working and more importantly, collaborating.

d) What to do better next time based on what was learned in the class about SE management and processes:

Next time, I would want to be more involved in each part of the project so that way I can be there when the code is written. I thought at first this could be overbearing and a lot of my team members, myself included, work better with some privacy to just code and run into issues and fix them or talk about them with others. The reason I would want to do this is because it can help avoid small mistakes when it comes to specifications, and I've personally been able to understand code better when someone code and explains it live. This would save time because no time would be used to develop anything that we would end up removing or rewriting. I feel that it is possible also to improve the efficiency of development by bouncing ideas off one another in real time like a peer programming setting. As a SE lead, the role comes with a lot of responsibilities so I would make sure to be extra firm on deadlines and have a weekly due date for some contributions in the project.

I would also find a new approach to delegating tasks for my team. I can make a deadline for team members but it's important to make the task feasible in that amount of time. Some tasks required more than one team member, and this delayed due dates because it could be a day or two until they could even begin work together at the same time. I offered a lot of flexibility to my team, and I don't expect my future leads to be that way with me, but at the end of the day we are all students and all on one team. The extra understanding goes a long way especially when its reciprocated. My group was very humble too and I'll find more ways to recognize them next time.

e) Anything else you deem important for instructors to know:

I had an extremely well-rounded team, and although there is variance when it comes to GitHub commits, I did not require every team member individually make commits to development. Sometimes, they would work offline or send code to one another instead, which worked better for us. One thing I tried my best to do, is let each of my team members play to their strengths when it came to the team project. I quickly noted their abilities, experience, and preferences and found a role in the team that best suited each team member. I also encouraged team members to volunteer to help in any area of development they wanted to so long as their owned work ends up completed.

SR

Sean Michael Ryan

To: Philip Karnatsevich; Karl Christopher Moreno; Darien Thuan Banuelos; Ronnie Y. L. Huang; Joaquin Nguyen Warren Sun 12/10/2023 5:58 PM

a) Their contributions to team project and teamwork:

- Set up server jointly with team
- Implemented Database architecture
- Reorganized webapp directories
- Implemented ability to search for tutors
- Implemented login checking credentials against server and creating a session
- Implemented registration saving account to server
- Implemented messaging and posting messages to dashboard
- Implemented tutor registration
- Helped teammates integrate frontend with existing architecture

b) Number of submissions to github team Dev. Branch:
42

c) Main challenges encountered in team project:
My biggest challenge was displaying search results before we had implemented ejs in the project. I tried using css only which was difficult because I was unfamiliar with the fetch api and the code had to be handled in the static directory which imposes some limitations. Another challenge was implementing messages. Several functions created race conditions and I had to become more familiar with async functions and await expressions in order to solve them.
d)What to do better next time based on what was learned in the class about SE management and processes:
The most helpful change I would make would be to analyze the data that needed to be displayed on the storyboards and organize the database to most easily represent that. I ended up having to make extra calls to the database because some tables return rows that don't have all the data I needed to display.
e) Anything else you deem important for instructors to know:
I think that even though other members may have less contributions on github, their contributions to teamwork, advice given on specific issues, and participation in planning was extremely helpful to implementing the features that make up our product.

R

Ronnie Y. L. Huang

To: Joaquin Nguyen Warren; Karl Christopher Moreno; Sean Michael Ryan; Darien Thuan Banuelos; **+1 other** Sun 12/10/2023 10:04 PM

a. I primarily helped with managing GitHub and checking if the code being pushed functioned, helped do each Milestone's write-ups, and occasionally helped with the coding process.
b. The number of submissions I made in my team's Dev. Branch were 13 commits due to not being one of the main programmers (and in the rare times I did, the code was usually given and pushed by another group member)
c. My biggest challenge in the team project was getting the site to run on my own PC. While half of the group were able to run it just fine, the other half weren't able to. When I eventually asked for help (as I wanted to test if sites are functioning properly or not), and even after hours and hours of downloading programs and a lot of testing, I still wasn't able to run the site. The only way I was able to run the site was to download an extension to do so, which wasn't the most effective as it lacked the ability to use features like the website's search.
d. What we'll do better next time is to create a place that specifically used to bring up minor issues with the code that couldn't be fixed with a small edit. Beforehand, the place where we brought up issues we had with the code was scattered throughout our group chat, and in doing so, it felt unorganized, and it wasn't clear as to what issues were done, and what we were still working on. We were eventually introduced to Tello, however, despite how much it helped us keep track of what we were doing, due the amount of minor issues we had, it was infeasible to put each and every one of them into Trello, and so one of us suggested making a separate place in a group chat for only minor code issues, which helped us a lot.
e. My team's group leader is very skilled at what he does, he understands every group member's strength, weakness, and experiences, and uses it to bring out the very best from his group. Even the rest of them were also very adequate too at coding, always finishing the milestones earlier than their due date and was always willing to help other group members in need. My love for my team extends even outside of coding, as they felt less like a group meant to do a project with, but friends that I could talk to, and it created a sense of family and togetherness that I just rarely ever feel in a group project, and it certain made participation and group meetings more fun.

DB

Darien Thuan Banuelos

To: Joaquin Nguyen Warren; Karl Christopher Moreno; Sean Michael Ryan; Philip Karnatsevich; Ronnie Y. L. Huang

Mon 12/11/2023 2:30 PM

a) Their contributions to team project and teamwork:

- Designed individual about us pages for servers initial test launch
- Drafted Initial wireframe diagram for website based on use cases
- Assisted in documentation for development of project and class milestones
- Implemented universal navbar with clickable logo throughout website
- Implemented login and register page frontend
- Implemented dashboard frontend
- Applied website theme and color scheme to all pages
- Practiced iterative revision development by providing code review
- Helped ensure functional backend aligned with existing frontend design

b) Number of submissions to github team Dev. Branch:

16, my tasks were usually delegated to design and front end

c) Main challenges encountered in team project:

The main challenge I encountered started to arise later in the development of our website. The issue was how our front-end file architecture was created at the beginning of development. The way our pages were initially connecting to the database forced us to use inline styling at one point. Ejs pages were linked to different CSS pages that didn't correspond to them and some assets that needed to be universal were being treated on a per page basis. We were warned about spaghetti code at the beginning of class and the challenges that come with spaghetti code started to become apparent when changing one minor thing in CSS would sometimes break our pages. We never ended up restructuring to the extent that was initially planned, however, better planning and preparation allowed us to have a more reliable and easier-to-work-with structure.

d) What to do better based on what was learned about SE management and processes:

Based on what we learned about software engineering management I think better planning in the initial stages of design would have benefited and streamlined our work. Also, I think more effort in communicating how different parts of our project work together for a better overall understanding would also benefit how we would have designed the structure of different aspects of our project. The team's overall efforts, however, allowed us to overcome this oversight and we were able to get a working model.

e) Anything else you deem important for instructors to know:

This team is composed of very skilled and well rounded members. When we all put our efforts into a task you can expect us to get it submitted the day before. Comes from good management from our Lead, good direction from our Dev Leads, and a willingness to learn and adapt by our supporting developers.

PK

Philip Karnatsevich

To: Sean Michael Ryan; Joaquin Nguyen Warren; Darien Thuan Banuelos; Karl Christopher Moreno; Ronnie Y. L. Huang

Tue 12/12/2023 2:40 AM

a) Their contributions to team project and teamwork:
I worked closely with my team to create comprehensive documentation, focusing on supporting the Backend lead in accomplishing key objectives such as designing the database architecture. Maintained consistent communication through Discord channels and actively participated in all scheduled team meetings. Contributed to the definition of main use cases and diligently reviewed each one. Played a role in reviewing and finalizing both functional and non-functional requirements to ensure their feasibility. In particular, I took responsibility for the implementation of the login and registration functions and worked closely with the Backend lead to develop a reliable login system for credential verification.

b) Number of submissions to GitHub team Dev. Branch:
As I collaborated closely with the Backend lead to brainstorm ideas for implementing the project's backend, my submission count was lower than that of other team members.

c) Main challenges encountered in team project:
Backend development in our project is quite challenging, especially considering my limited experience. I have to deal with designing and managing a solid database architecture, which is no easy task. Developing server-side logic adds another layer of complexity that I'm working hard to navigate.

d) What to do better next time based on what was learned in the class about SE management and processes:
As I reflect on my recent collaboration with the team, I realize that while I played a role in supporting the Backend and implementing specific functionalities such as login and registration, my submission count was lower than that of my peers. To enhance my contribution to future projects, I aim to manage my time more efficiently. This means balancing collaborative tasks with individual responsibilities. Additionally, given the challenging nature of backend development, I plan to dedicate more time to building my skills in designing and managing database architecture and navigating the complexities of server-side logic. This experience has highlighted the importance of a well-rounded skill set, and I am committed to continuous learning and improvement for future collaborations.

e) Anything else you deem important for instructors to know:
I had an incredibly positive experience working with our team. The collaboration among team members was truly impressive. From the initial stages of designing our project to implementing various functionalities, everyone contributed their skills and expertise. Communication through platforms like Discord was efficient, and attending scheduled team meetings fostered a sense of unity and shared purpose. The team's dedication and mutual support not only made the development process smooth but also created a positive and motivating work environment. It was a valuable learning experience, and I look forward to future collaborations with such a talented and cohesive team.

KM

Karl Christopher Moreno

To: Joaquin Nguyen Warren

Tue 12/12/2023 7:47 AM

Karl Moreno

Contributions in the following areas:

- Conf
  - Database.js
- Routes
  - index.js
- Views
  - aboutMe-landing.ejs
  - dashboard.ejs
  - login-form.ejs
  - register-form.ejs
  - search-tutors.ejs
  - tutor-application.ejs
- About Pages
  - About landing
- CSS in general
- Design of page
- Searching functionality
- Messaging functionality
- APIs to and from database

As of writing I have made about 28 commits on the page but after all changes are pulled in that will undoubtedly be much higher

Some of the main challenges I encountered in working on this project was time to work. Though I have a very high contribution to the website on the technical side. I feel that if it weren't for outside factors like my new job I could've delivered far more Priority 2 features within the site.

One thing we could do better as a team is realize quickly the strengths and shortcomings of the team as a whole and work to create a better system of deliveries and time needed to accomplish those. I feel if we implemented SCRUM a lot earlier and had a very clear plan on strong deadlines on key technical aspects a lot more progress could've been made. I don't say that as a team critique, I say that as a critique to myself as I could've held up that standard but didn't just because I wasn't thinking about it. But now I know next time how much better of project I could've had with a lot more future planning not in the technical aspects of website. But the personal management of the team.