MentorMe Project Proposal

By Nithisha Sathishkumar, Andrew Lim, Sahong Song, Zain Zaman

Problem Statement

Our team's project, the MentorMe, seeks to close educational inequalities by making tutoring services accessible, free/paid, and personalized to students in need. Many students struggle to find dependable academic support outside of school due to a lack of local tutoring choices, excessive fees, or scheduling difficulties. As a result, students, particularly those in underprivileged areas, do not have continuous access to tutoring, which could boost their academic performance and confidence in a variety of topics. This platform aims to provide a virtual learning environment in which students can easily interact with tutors for tailored guidance in a variety of academic areas, at any time and from any location.

Objective

The MentorMe program will provide students with an easy-to-use online tutoring platform that allows them to choose from a variety of disciplines and interact with different instructors via an online interface. On this platform, students can:

Must Feature:

- The tutoring system allows students to choose from a variety of subjects and meet with their tutors online.
- Tutor profiles include information on their rating, student reviews, subjects they teach, experience, and an optional display of their pricing/free
- The platform includes a simple scheduling interface with potential online notification features.
- A built-in chat system provides a quick way for students and tutors to communicate.
- Tutors are displayed in a list based on subject, price, rating, availability, and languages spoken.
- Payments can be processed through PayPal, with options for Credit Card, Google Pay, orApple Pay.
- Sessions are conducted over Zoom, with access to a whiteboard feature and email or chat for communication.
- Each tutoring session is set at 30 minutes in length.
- Students have the option to attend drop-in meetings and can tip tutors if they wish.
- Group tutoring sessions are also available.

Nice to have features if time permits:

- An optional form allows students to request additional subjects they'd like to study.

- A feature lets students view the tutoring history of each tutor.
- The platform includes a built-in online conference tool with real-time translation and transcription capabilities.

Biography

Listed below are members of our project, along with their skills and prior experience:

Nithisha Sathishkumar, a 4th year Computer Science student in university of Washington Bothell, has experience in several programming languages, such as Java, Python, C++, JavaScript, HTML, CSS, swift, PostgreSQL and more. She has experience with Git, Github, and Visual Studio Code. Academically, she has taken an introductory course in Java and C++ along with a Data Structures and Algorithms course, Database course, UI/UX design, Software design and development and is currently learning about Web Development. Personally, she has created a movie inventory management bank using C++, a personal website, Signtalk app which converts ASL sign to text and audio Real Time, Weather using JavaScript, and more. This project will help her build the foundation for web development and collaboration with the team.

Andrew Lim, a 4th-year Computer Science student at the University of Washington, has experience in several programming languages, such as C++, Java, HTML, CSS, and JavaScript. He has experience with Visual Studio Code, IntelliJ, Git, and GitHub. He has taken courses in Data Structures and Algorithms, Database Systems, Game Development, Introduction to AI, Information Assurance and Cybersecurity, Software Design and Development, and is currently enrolled in a Web Development course. He is taking this course to enhance his skills in building personal portfolio websites and other web-based projects. Additionally, Andrew is passionate about exploring the latest web technologies to create user-friendly and visually appealing interfaces. He enjoys collaborating on projects that have a positive impact on real-world challenges.

Sahong Song, a 4th-year Computer Science student at the University of Washington Bothell, has developed skills in programming languages such as Java, C++, JavaScript, and PostgreSQL, HTML. Experienced in using tools like VSCode, Git, Git, and IntelliJ, she is comfortable working across different development environments. Academically, she has completed courses in Java, Data Structures and Algorithms, and is now focusing on Web Development. She also did a capstone project that involves creating a web generator tailored for small and medium-sized businesses. This project aims to simplify web presence for businesses by offering customizable, easy-to-deploy website solutions, highlighting her passion for practical, impact-driven software development.

Zain Zaman, 4th year Applied Computing and Data Science Student at the University of Washington Bothell; skills primarily consist of Python, SQL, and C++, and is comfortable

using tools such as VSCode, psql, and Git. Academically, he's taken courses about Data Structures and Algorithms, as well as Databases, and has experience designing database systems for real world use.

Project Development Timeline

This section presents our team's project timeline in a table format, outlining weekly milestones and anticipated completion dates. It provides a clear view of key development stages, helping us track progress and stay aligned on our project goals.

Table 1: In-Depth Project Timeline

Week #	Focus	Goals	Individual Tasks
Week 5 (10/28 - 11/03)	Project Proposal & Feature Identifications	Team meeting to brainstorm possible features that we must have and nice to have, wish to have. We also discuss role assignments, technical requirements, and overall, our main objectives as a group. Ensure Project is approved by the Professor.	Andrew: Provided a list of features we wish to implement. Also created an in-depth table showcasing our project timeline, making it as realistic as possible to ensure we can meet the deadline.
			Nithisha: wrote a problem statement, objective and one-use case scenario.
			Song: create use cases, and help on APIs
			Zain: Searched For APIs that would help with our project, as well as wrote out a use case Scenario
Week 6 (11/04 - 11/10)	UI/UX Design, Wireframe and Figma Design	Create a clear, functional design blueprint for the website using wireframe for blueprint and Figma for a prototype for the website.	Andrew: Work with the team to create a front-end design in Figma, determining color scheme, fonts, and layouts.
			Nithisha: create Figma for the prototype
			Song: Help on building functional design blueprint

			Zain: Work together with the team to build a good Figma layout, as well probably gauge the scope of our project
Week 7 (11/11 - 11/17)	Database Design and Frontend Set Up	Start setting up and developing a database (what data, relational model, database system – SQL/MongoDB). In addition, our team must start to set up the back-end development.	Andrew: Work together with the team on developing a relational schema and diagram to help map one entity to another.
			Nithisha: create the database scheme and help create database
			Song: Work together with the team on developing a relational schema
			Zain: Help develop the schema, as well as incorporate with Firebase
Week 8 (11/18 - 11/24)	Frontend Development and Backend Set Up	Start translating the Figma wireframes and prototypes into code for the frontend, focusing on layout, styling, and user interactions. We should also start setting up the backend structure to handle data storage (Database)	Andrew: build the html and CSS based on the specific pages
			Nithisha: build the html and CSS based on the specific pages
			Song: build the html and CSS based on the specific pages
			Zain: build the html and CSS based on the specific pages
Week 9 (11/25 - 12/2)	Backend Development and Testing	Develop backend API endpoints to support essential features, like user registration and session booking, and connect them to the database to ensure tutor and student data is stored and accessible. Perform integration testing to verify seamless	Andrew: Work on the login page and integrating APIs with the team.
			Nithisha: work on connecting the database to items

		frontend-backend interactions, optimizing data retrieval for secure and efficient connections.	Song: Work on user registration and the database
			Zain: Develop Scheduling logic, and potentially user registration
Week 10 (12/2 - 12/8)	Testing and Presentation	Finalize all implemented features, run comprehensive testing and website functionality, and prepare for soft launch or presenting to class/professor.	Andrew: Test the login functionality and overall functionality of the website.
			Nithisha: testing the website functionality and preparing for the presentation and soft launch.
			Song: Help to run comprehensive testing
			Zain: Help test the website, as well as fix any issues that arise

Note: Individual tasks may be incomplete in early stages of development and are bound to change as we slowly progress as a team.

API and their use

This section is a listing of api, we are planning to use for this project

Table 2: API List

API	Link to Documentation/Website	Purpose

ZoomAPI	https://developers.zoom.us/docs/api/	We plan to incorporate the Zoom API to: 1.) Allow Zoom meetings to automatically be scheduled when a user selects a time to meet with their tutor. This should also update if the user changes the meeting time or cancels a meeting 2.) Let users join zoom meetings directly from the website
FirebaseAPI	https://firebase.google.com/docs	We plan to use firebase to host our database, which will hold information such as: Account information (both student and tutors), Courses, and ratings
PayPal	https://developer.paypal.com/api/rest/responses/	The PayPal API will be used to process transactions, ensuring a smoother and securer checkout process for users.
Square	https://developer.squareup.com/docs/payments-overview	Integrating payments on a website or mobile platform is seamless with Square's online payment API. It offers a secure, PCI-compliant solution for accepting credit card payments.
Google Sign-In API	https://developers.google.com/identity/sign-in/android/sign-in-identity	We plan to use Google's Sign in API to allow users to a faster sign-in method

Calendarific (Scheduling)	https://calendarific.com/	For our tutoring platform, Calendarific can be a valuable addition to manage and optimize the scheduling feature. It will improve the usability of the platform, especially by aligning session bookings with the users' availability around holidays, making the scheduling feature more user-friendly and adaptable.
LibreTranslate	https://libretranslate.com/docs/	If time permits, we'd like to add translations within our website, LibreTranslate is a free translation API that would allows users to translate elements such as chat messages, and web content, allowing for a broad range of people to use our website

Implementation

This section we discuss how we plan to implement our website to seamlessly work together.

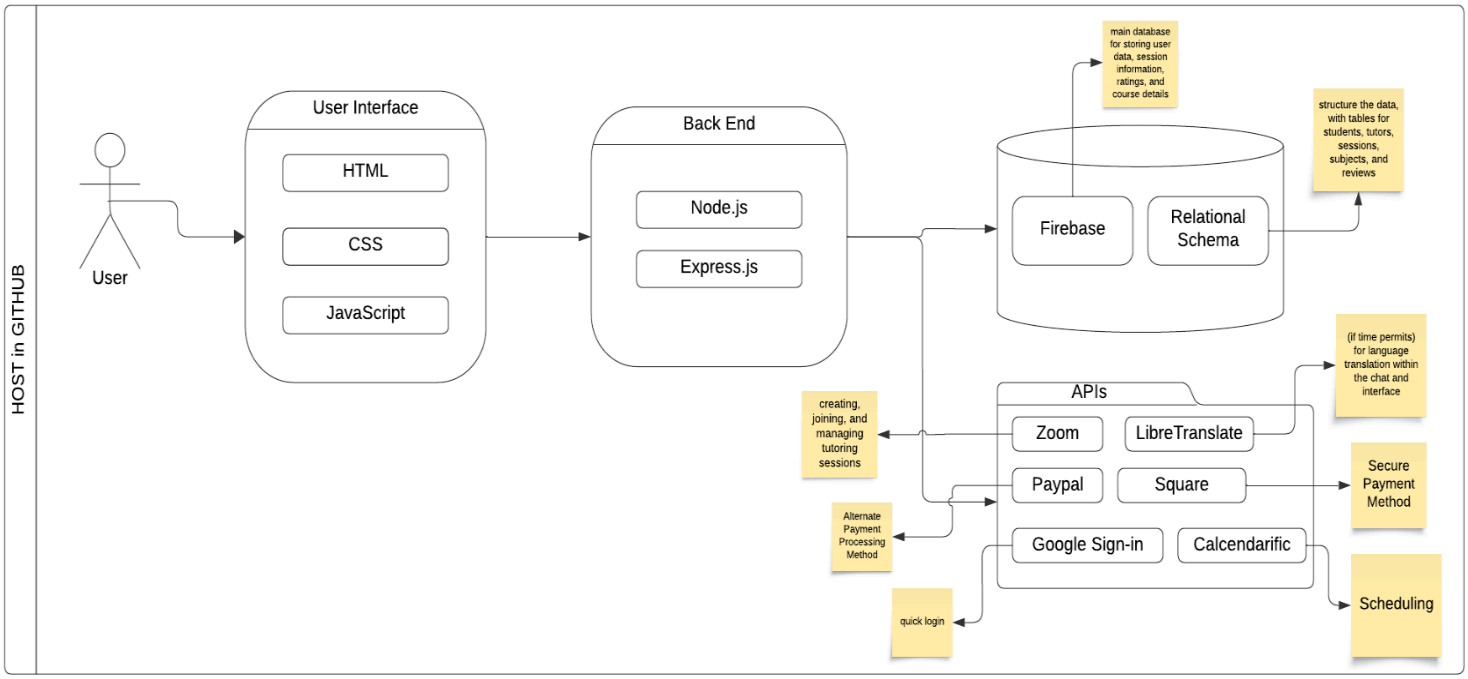
The primary implementation phases for MentorMe are to ensure a smooth interaction between our databases that store user and session information, APIs for scheduling, video conferencing, and payment processing, and a responsive frontend to provide an easy user experience. Each user will be required to register an account, complete with basic security features such as a username and password. Once enrolled, students will have access to many areas of the platform, including the option to create profiles with academic needs and interests in order to obtain individualized tutoring recommendations. Tutor profiles will also be saved, including information on subjects taught, ratings, availability, and pricing.

The user information, including student and tutor profiles, will be stored and managed by Firebase. Students can then search for subjects, filter tutor profiles based on availability, rating, pricing, and other parameters, and schedule sessions. The scheduling process will leverage the Zoom API to automatically establish Zoom meetings once a session has been confirmed. Users will be able to access their planned sessions through Zoom links directly

from the platform. A simple scheduling tool will also allow students to select available hours, and any changes to bookings or cancellations will be updated accordingly.

For paid sessions, the platform will securely process transactions via the PayPal or Square APIs, with additional payment choices such as Google Pay and Apple Pay. MentorMe's backend will validate payments before confirming sessions to ensure teachers are reimbursed for their work. In the case of free sessions, the payment phase will be skipped, with the session confirmed on both the student and instructor dashboards.

The backend will handle user identification, booking logic, and payment processing, allowing for secure and quick data transmission between Firebase and frontend components. Once developed, the frontend will provide this information in an easy-to-use manner, allowing students to explore available tutors, schedule sessions, and attend meetings all from a single interface. Optional features such as drop-in sessions and language support via LibreTranslate may also be added if time allows, to improve the user experience even more.



Risk Assessment

This section we listed below are most of the major risks, along with methods we will implement to fix those issues.

- Due to conflicting class schedules, our team will struggle to find time to collaborate on the project. Furthermore, as the holiday season approaches, there will be a spike in the number of final and team members taking time off.

    - Emphasize remote work with a strong emphasis on consistent communication and bi-weekly group meetings through discord
- Integrating several APIs such as Zoom, Firebase, and PayPal may cause issues, potentially delaying the project.
    - Before full integration, test each API individually and check the documentation to anticipate common issues.
- Balancing MentorMe development with other classes may limit the time available for research and project work.
    - Set clear, attainable goals with a deadline to ensure consistent growth and focus on important features.
- Each team member is focused on certain aspects of the project, which may lead to a reliance on individual roles.
    - Document work thoroughly and arrange knowledge-sharing sessions so that team members can step in as needed.

Technical Decisions

This section table comparison between different considerations and their pros and cons.

Table 3: Pro and cons

Considerations	**Pros**	**Cons**
HTML, CSS and JS for user interface	● Easy to use ● Supported on majority of browsers ● Familiarity with the language	● Limited interactivity ● Can get overly complicated
Firebase (Database Server)	● Easy to set-up and deploy ● Scales well with more users	● Can become pricey ● Limited Customization over database
Zoom(Video Call)	● Easy to Integrate ● High video call quality without having to building from the ground up	● Limited Free Plan (40-minute meetings per session)
Github Pages (Website hosting)	● Free Hosting and Easy to Set up ● Git Control	● Not very Scalable ● Limited Storage on Sites
Node.js	● Lightweight and simple ● Allows for more control on website	● Lack of experience among group ● Learning Curve

	without getting to complicated	
Express.js	• Integrates well with Node.js • Easier time with routing websites, leading to faster development	• Lots of third-party modules required for functions

Architecture Diagram:

To visually represent the architecture of MentorMe, we'll focus on several primary components and their interactions: **Frontend Interface**, **Backend Server**, **Database Management**, and **External APIs**.

Frontend Interface:

- User Authentication: Allows students and tutors to log in or sign up, using email or Google Sign-In API for convenience.
- Profile Management: Enables students and tutors to create and manage profiles, detailing academic needs (for students) or subjects and qualifications (for tutors).
- Tutor Search & Filter: Provides search capabilities for students to find tutors based on subject, availability, price, and rating.
- Session Booking Interface: Interface to schedule sessions, displaying tutors' available time slots.
- Payment Checkout: Offers payment processing options through PayPal, Square, and Google Pay.
- Chat System: Built-in chat for instant communication between students and tutors.
- Interaction: Frontend interacts with the backend to retrieve data for profile displays, search filters, chat messages, and payment confirmations.

Backend server:

- Authentication Service: Manages login sessions and permissions, verifying users and ensuring data security, particularly for payment and booking operations.
- Profile Service: Manages the creation and updates of profiles, retrieving data to personalize user experiences.
- Tutor Availability Service: Coordinates availability slots and updates tutors' calendars when students book sessions.
- Booking & Scheduling Service: Processes session bookings and cancellations, leveraging the Zoom API to schedule meeting links.

- Payment Processing Service: Communicates with PayPal and Square APIs to handle secure transactions for paid sessions.
- Notification Service: Sends booking confirmations and reminders through email or platform notifications to students and tutors.
- Chat Service: Manages real-time chat sessions, storing chat histories as needed for records.
- Interaction: The backend is the intermediary between the frontend and other components, managing data processing, API calls, and ensuring security protocols for user interactions.

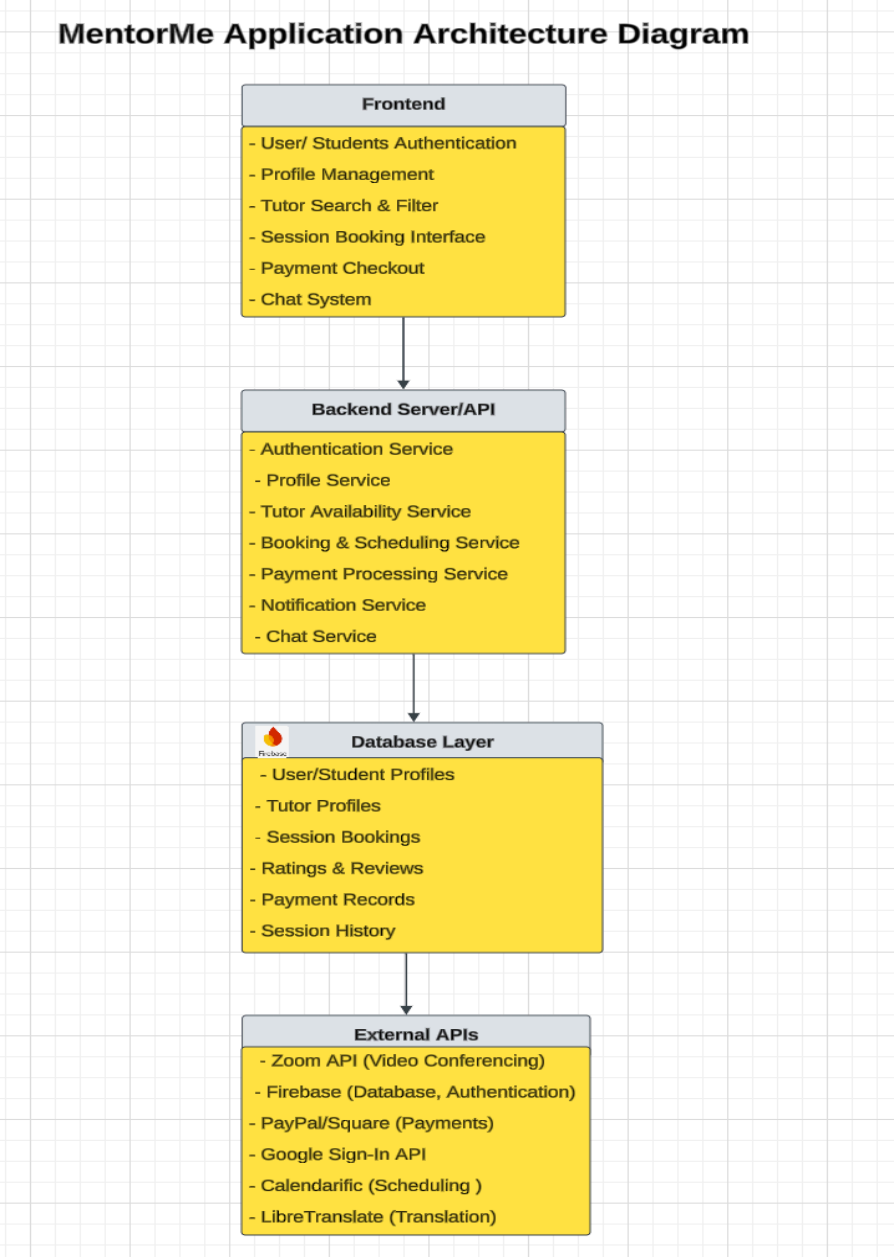
Database layer:

- User Profiles: Stores essential information for both students and tutors, including login credentials, preferences, subjects, and availability.
- Tutor Profiles: Contains additional fields like qualifications, ratings, pricing, and experience for each tutor.
- Session Bookings: Holds records of scheduled sessions, including timestamps, subjects, and associated user IDs.
- Ratings & Reviews: Allows students to leave feedback, which is stored and displayed in tutor profiles.
- Payment Records: Logs successful payments for session tracking, allowing financial audits and payment resolutions.
- Session History: Maintains logs of past sessions for student reference and tutor analysis.
- Interaction: Data from the database is requested by the backend and supplied to the frontend as necessary, supporting seamless data flow throughout the application.

External APIs:

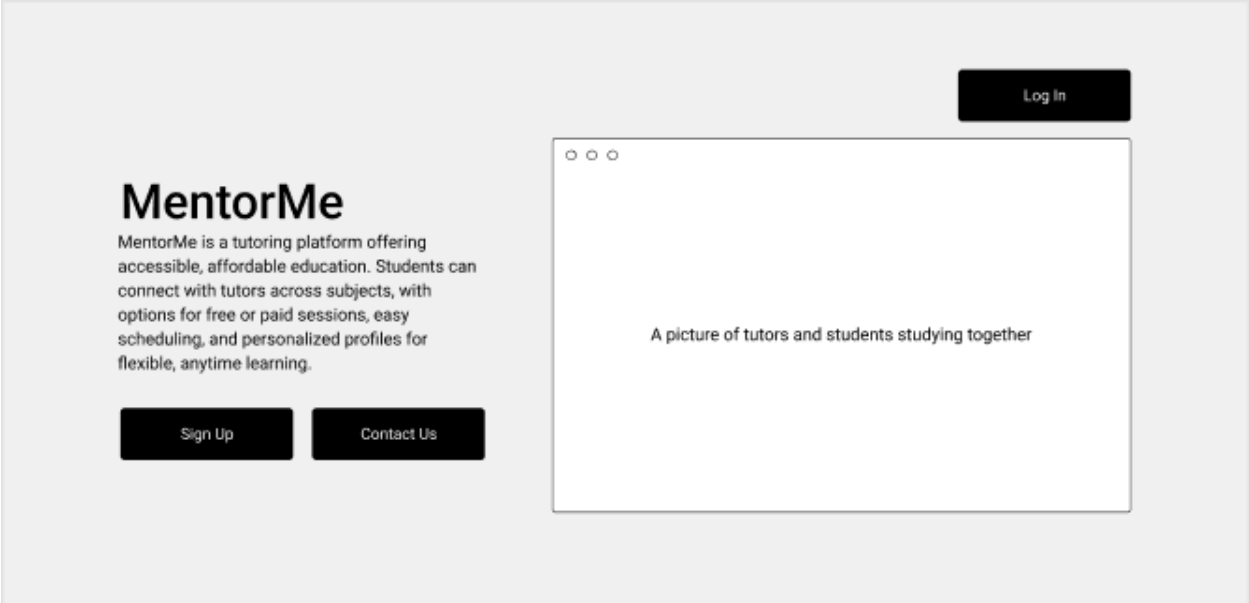
- Zoom API: Used for generating unique meeting links for each session, with scheduling updates when rescheduling or cancellations occur.
- Firebase: Serves as the primary database and authentication provider, offering user management and secure data storage.
- PayPal/Square APIs: Handle secure payment processing, supporting multiple payment methods in the MentorMe checkout system.
- Google Sign-In API: Enables users to log in through Google, simplifying user access and profile setup.
- Calendarific (Scheduling): Enhance scheduling by integrating holiday awareness and cultural date considerations. This API provides a calendar of public holidays and even observances across different countries.

- LibreTranslate API (optional): Provides real-time translation for chat messages or other platform text, enhancing accessibility for non-English users.
- Interaction: The backend server makes calls to these APIs as needed—fetching Zoom links, processing payments, authenticating users, and handling translations. The frontend then displays these responses to users.

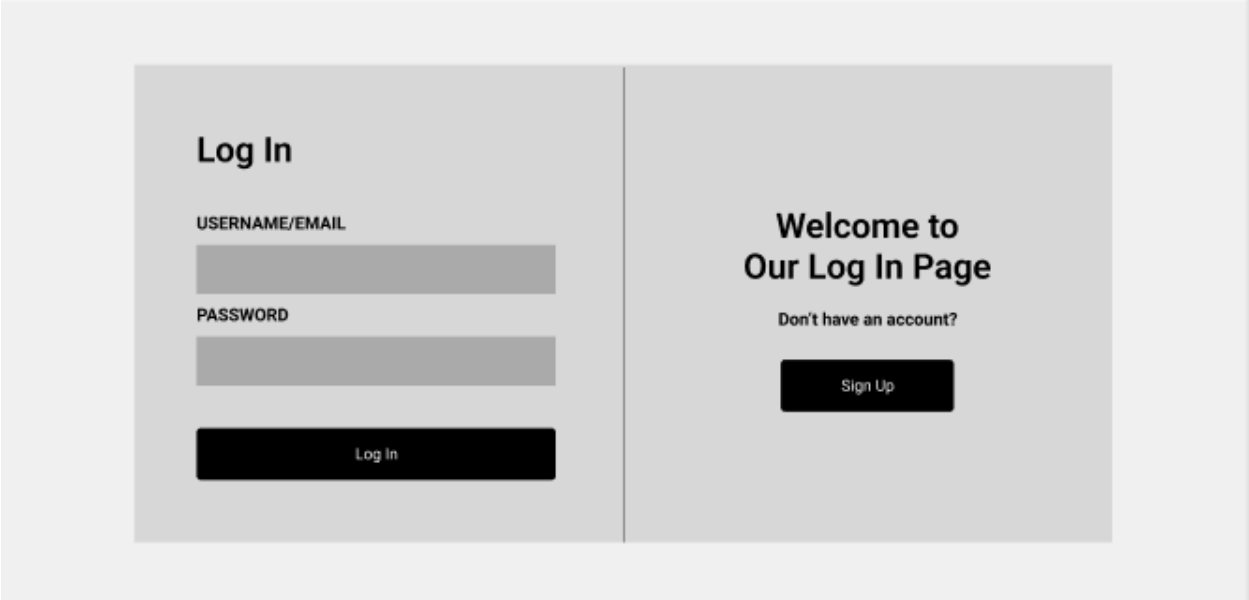


Sample Wireframe of MentorMe (First Draft)

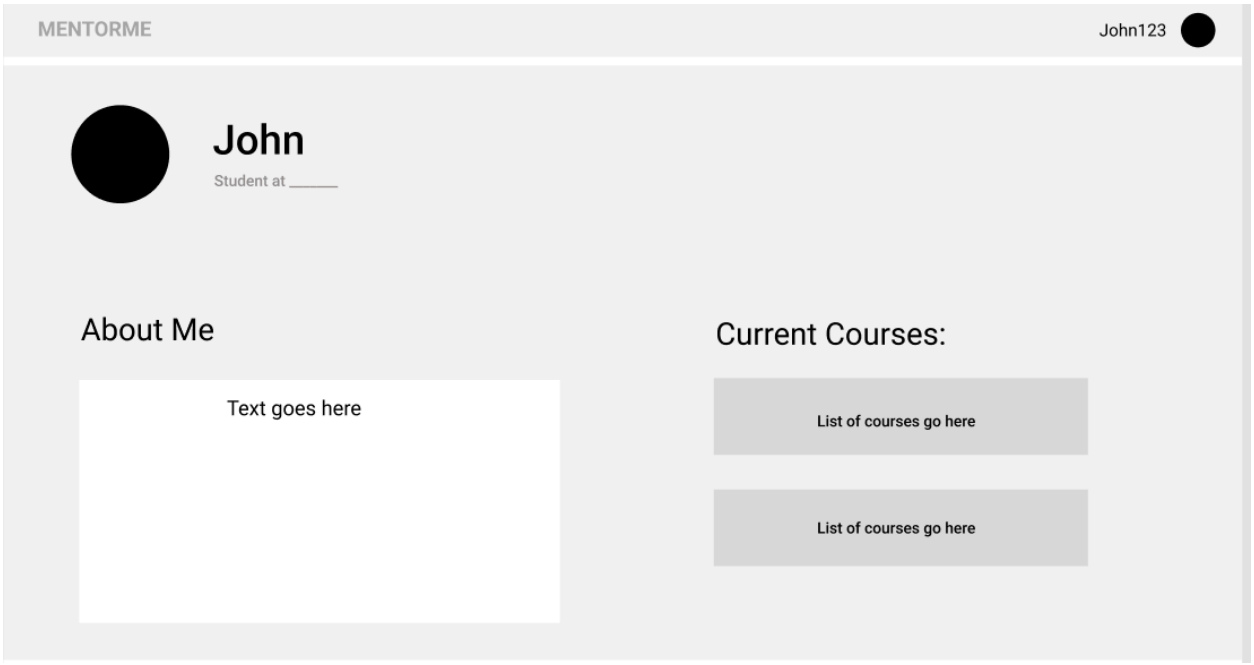
Wireframe 1: Website Landing Page



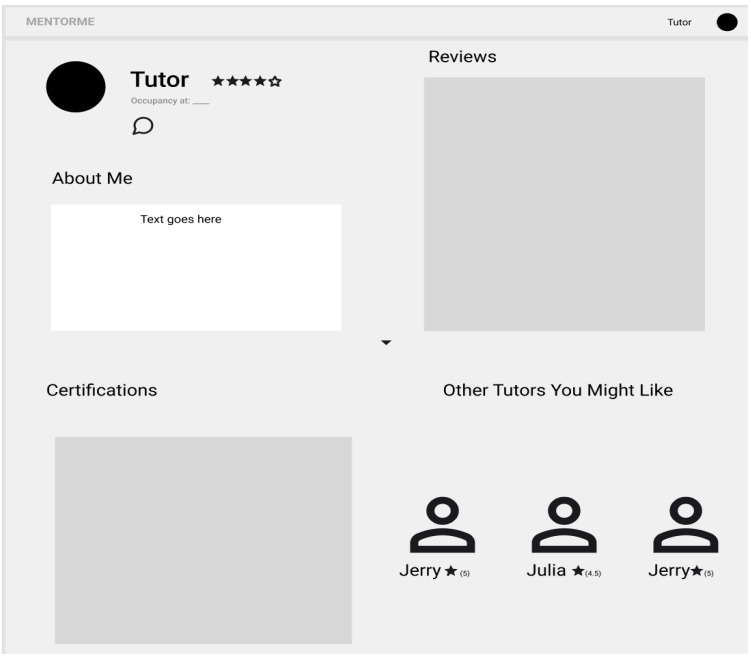
Wireframe 2: Website Login Page



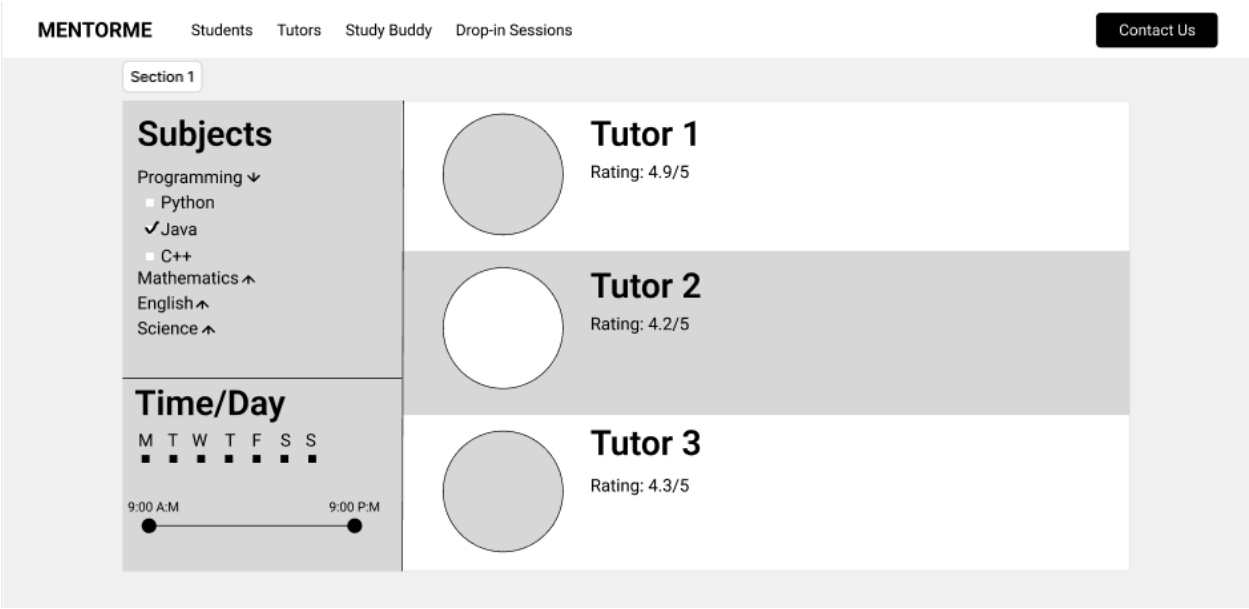
Wireframe 3: Student Profile Page



Wireframe 4: Tutor Profile Page



Wireframe 5: Student's View of the Available Tutors for a Subject



Use Case

This section talks about use case scenarios of the mentorMe which include who the users are, preconditions, postconditions, scenarios, alternate and exceptional.

1. **Jamie(student) scheduling a tutoring session on MentorMe**
    a. Stakeholder and Interests:
        i. Jamie: Needs academic help in algebra and wants a tutor with relevant expertise
        ii. Alex(tutor): Provides tutoring services in algebra and is available for new sessions
    b. Preconditions:
        i. Jamie has registered and logged into the MentorMe platform.
        ii. Tutors, including Alex, are available for the subject Jamie needs help with
    c. Postconditions:
        i. Jamie successfully books a tutoring session with Alex.
        ii. A confirmation notification is sent to both Jamie and Alex.
    d. Scenario:
        i. Jamie logs into MentorMe and selects “Algebra” from the list of available subjects.
        ii. MentorMe displays a list of tutors, including Alex, with profiles that include ratings, reviews, experience, subjects taught, and pricing options.

iii. Jamie filters the list by availability, language, and price, then selects Alex as the tutor.
iv. MentorMe displays Alex's available time slots.
v. Jamie chooses a suitable time and books a 30-minute session with Alex.
vi. Payment options are displayed; Jamie completes the payment using Google Pay.
vii. MentorMe sends a confirmation notification to Jamie via email, including the session details and Zoom link.
viii. Alex also receives a notification with the scheduled session information.

e. Alternate:
    i. If Alex is unavailable, Jamie can choose a different tutor from the list, such as Taylor or Chris.
    ii. If Jamie chooses a free tutor, the payment step is skipped.
    iii. If Jamie prefers a group session, they can select that option instead.
f. Exceptional:
    i. If Jamie encounters a scheduling conflict, they can cancel the session up to 24 hours in advance.
    ii. If there is a technical issue with the session link, Jamie can reschedule or contact support.

2. Darren (tutor) setting up a tutor profile

a. Stakeholder and Interests
    i. Darren (Tutor) wants to showcase their qualifications and experience to attract students
    ii. Students look for teachers that have ample experience in their topic
    iii. Administrators – want to have reputable tutors on their site
b. Preconditions:
    i. Darren (Tutor) has created an account with MentorMe as a tutor
c. Postconditions:
    i. Darren's tutor page is successfully setup, showcasing his qualifications, cost, availability, and user reviews if applicable
    ii. Darren's profile is visible to the public, whether found naturally, or through a specific search
    iii. Important qualifications of Darren (such as his degree, or certificates) are validated
    iv. Darren has a payment method setup to receive compensation
d. Scenario

i. Darren creates a new MentorMe Account, selecting “tutor” as his role.
ii. Darren is then prompted to fill out required details about his profile (Availability, course offering, About Me, price), with an option to completely fill out his profile as well, he can complete optional fields at any time.
iii. After Darren’s profile is complete, it becomes visible to users browsing the website
iv. Jerry, a user looking to get help with Calculus, sees Darren’s profile offering, and decides to book a meeting with Darren.

e. Alternate
i. If Darren creates an account but doesn’t fill out any fields/all the required fields, he will be notified through email, reminding him to set up his account. His account will remain private until he does so
ii. If Darren add qualifications to his profile, they will be pending until approved

f. Exceptional
i. If an error arises during setup, the system will save the processed fields, while removing the ones that haven’t

3. Jimmy (student) setting up a student profile

a. Stakeholder and interests:
i. Student: Jimmy wants to create a complete student profile so that he can access and manage his tutoring sessions.
ii. Tutor: Tutor can know the student’s academic needs and preferences to provide personalized support.
iii. Platform Administrators: Ensure that profiles are accurate and complete for streamlined user experience.

b. Preconditions:
a. Jimmy has created an account and logged into MentorMe

c. Postconditions:
a. Jimmy’s profile is set up successfully, with details about his academic needs, preferred subjects, and any additional information filled out.
b. His profile is visible to tutors, who can review his needs and interests to better tailor their sessions.

d. Scenario:
a. Jimmy logs into MentorMe and accesses the “Profile Setup” section.

b. MentorMe prompts Jimmy to enter details like his grade level, preferred subjects, academic goals, and learning preferences.
c. Jimmy completes his profile by specifying his availability for sessions, preferred communication method (e.g., chat or video), and any special requirements (e.g., subject focus).
d. Jimmy reviews and saves his profile, which becomes visible to tutors on the platform.
e. MentorMe sends a confirmation email, notifying Jimmy that his profile setup is complete and ready to use for scheduling sessions.

e. Alternate:
    a. If Jimmy skips optional fields, he is notified that they can be completed later for a more tailored experience.

f. Exceptional:
    a. If Jimmy encounters an error (e.g., lost connection), MentorMe saves his progress, allowing him to resume setup from where he left off once he reconnects.

4. Jamie (student) wants to attend a drop in one-on–one tutoring session

a. Stakeholder and Interests:
    i. Student: Jamie Seeks immediate, flexible academic help without prior scheduling.
    ii. Available Tutor: Provides support on-demand for various topics.
    iii. Platform Administrators: Ensure seamless access to drop-in tutoring sessions.

b. Preconditions:
    i. Jamie has registered and logged into his MentorMe account.
    ii. There is an available tutor for the subject Lily needs help with

c. Postconditions:
    i. Jamie successfully joins a one-on-one session with an available tutor.
    ii. The session is recorded in Lily's history for future reference.

d. Scenario:
    i. Jamie logs into MentorMe, selects her desired subject (e.g., Algebra), and clicks the “Drop-In Tutoring” option.
    ii. MentorMe searches for available tutors in real time and displays any tutors ready for a drop-in session.

iii. Jamie chooses a tutor and is immediately connected through a Zoom link or built-in video chat for a one-on-one session.
iv. He receives personalized assistance on specific questions or concepts he was struggling with.
v. After the 30-minute session, MentorMe logs the session in his history and asks Lily to provide optional feedback.

e. Alternate:
    i. If no tutor is currently available, Jamie can set an alert to notify her as soon as a tutor becomes available for a drop-in session.

f. Exceptional:
    i. If Jamie encounters a tech issue like internet instability or disruption, he can rejoin the session if it is within the 30 mins time frame.

5. Student making a payment for a tutoring session

a. Stakeholder and interests:
    i. Student: wants a secure, convenient way to pay for a tutoring session using a preferred payment method.
    ii. Tutor: Receives payment upon completion of the session.
    iii. Platform Administrators**:** Ensure payment processing is secure, reliable, and supports various methods for user convenience.

b. Precondition:
    i. The student has registered and logged into the MentorMe platform.
    ii. The student has selected a tutor and scheduled a session with confirmed pricing details.

c. Postcondition:
    i. The payment is processed successfully, and the session is confirmed.
    ii. Both student and tutor receive a confirmation notification, and payment details are stored securely.

d. Scenario:
    i. The student logs into MentorMe, selects a tutor and session time, and proceeds to the checkout page.
    ii. MentorMe displays available payment options: PayPal, Credit Card, Google Pay, or Apple Pay.

iii. The student chooses their preferred payment method, such as Google Pay, and enters the necessary information.
iv. MentorMe securely processes the payment through an integrated payment gateway.
v. Upon successful payment, MentorMe displays a confirmation message and sends a confirmation email with session details and payment receipt to both the student and tutor.
vi. The session is now marked as paid and confirmed on the student's dashboard.

e. Alternate:
    i. If the student selects a free tutor, the payment step is skipped, and the session is directly confirmed.

f. Exceptional:
    i. If the payment fails (e.g., due to insufficient funds or technical issues), MentorMe displays an error message with options to try a different payment method or troubleshoot the issue.
    ii. If the student accidentally cancels the transaction, they are prompted to restart the payment process to complete their booking.