



CSCI 5709

Advanced Web Services

Assignment 2

[https://git.cs.dal.ca/nkukadiya/CSCI-5709-Assignments/-
/tree/main/Assignment2?ref_type=heads](https://git.cs.dal.ca/nkukadiya/CSCI-5709-Assignments/-/tree/main/Assignment2?ref_type=heads)

Submitted by: Nikulkumar Kukadiya
B00981020 (nk865270@dal.ca)

Table of Contents

A2.1 Choose 1 feature from the list of “Must Have” features.	3
A2.2 Application details	3
A2.2.1 User personas	3
Therapists (Dr. Jehangir Khan).....	3
User (Kaira Sharma)	4
User (Jacob Martin)	5
A2.2.2 User scenario for each task in the feature	6
Task 1: Live Session creation (User Persona: Dr. Jehangir)	6
Task 2: Live Session management (User Persona: Dr. Jehangir)	9
Task 3: Live session joining and user sessions	11
A2.2.2 High fidelity for tasks	13
A2.2.2 Application architecture	15
A2.2.2 Interaction design	16
Task 1: Create session click stream	16
Task 2: Edit session click stream.....	16
Task 3: Start Live session click stream.....	16
Process / Service Workflow:.....	17
How the feature will be triggers in the process above.	17
Folder Structure:	19
References	21

Figures

Figure 1 Therapist user persona	4
Figure 2 User persona (Kaira).....	5
Figure 3 User persona (Jacob)	6
Figure 4 Task flow diagram 1 (Session creation)	7
Figure 5 Wireframe (Create Session)	8

Figure 6 Task flow diagram 2 (Session Edit).....	9
Figure 7 Edit session wireframe	10
Figure 8 Task flow diagram 3 (Live Session Streaming)	11
Figure 9 Wireframe Live session.....	12
Figure 10 Working site screenshots.	14
Figure 11 Application architecture	15
Figure 12 Create session click stream	16
Figure 13 Edit session click stream.....	16
Figure 14 Live session streaming click stream.....	16
Figure 15 Process/Service workflow	17
Figure 16 React App folder structure.	19
Figure 17 Backend application folder structure.....	20

A2.1 Choose 1 feature from the list of “Must Have” features.

I am choosing the following feature from our project proposal's features.

- Live Session creation and management from expert side.

A2.2 Application details

Enhancing Your Mental Well-being with Our Comprehensive Mindfulness Application

Welcome to the ultimate platform for mental wellness! Our mindfulness application is your companion on the journey to a calmer, more focused, and stress-free life. With features like meditation listings, guided video and audio sessions, breathing exercises, live sessions with experts, chat support, and a journaling tool, we provide everything you need to cultivate mindfulness and improve your mental health.

Goal

- Proven mindfulness techniques for stress and anxiety management.
- Expert-led live sessions to learn new mindfulness practices.
- Supportive community fostering encouragement and shared experiences.
- Intuitive interface for easy navigation and resource access.
- Insightful journaling tool for self-reflection and progress tracking.
- Transformative experience leads to improved mental well-being.

Target User Insights

Calm Essence is designed for individuals seeking to enhance their mental well-being through mindfulness activities. Our targeted users include experts who are therapists and daily practitioners who are performing the mindfulness activities in their day-to-day life to increase their mental well-being or individuals who needs the health providers to increase their mental health.

A2.2.1 User personas

Therapists (Dr. Jehangir Khan)

- Description: Dr. Jehangir khan, an expert of the mindfulness activities id dedicated to support his patients' mental health via different live and in person sessions by teaching them the mindfulness activities. He is professional with years of experience to manage and better their patient's mental wellbeing.
- Assumptions: Jehangir will use calm essence to create the sessions to our platform users and will be actively teaching new methods and techniques to our users.
- Requirements
 - Has a good desktop or laptop with a camera and microphone access.

- Has a better internet connection to perform live streaming.

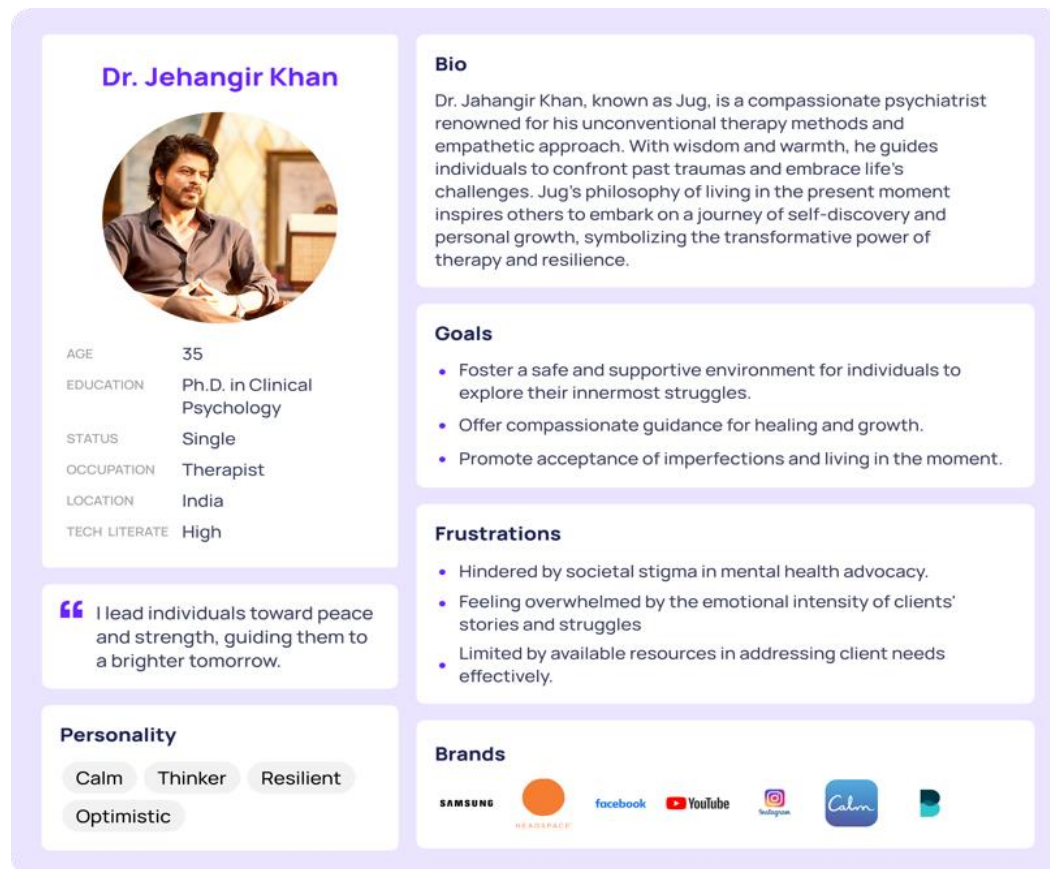


Figure 1 Therapist user persona

User (Kaira Sharma)

- **Description:** Kaira is a cinematographer. She gets easily burned out by life's ups and downs and needs mental support which makes her calm and relaxed in her day to day life.
- **Assumption:** Kaira will use our application to learn meditation and mindfulness activities to feel balanced in her life.
- **Requirements**
 - Kaira has a good internet connection.
 - A mobile phone or a computer in which she will browse our application features.
 - Some domain knowledge of using digital apps.

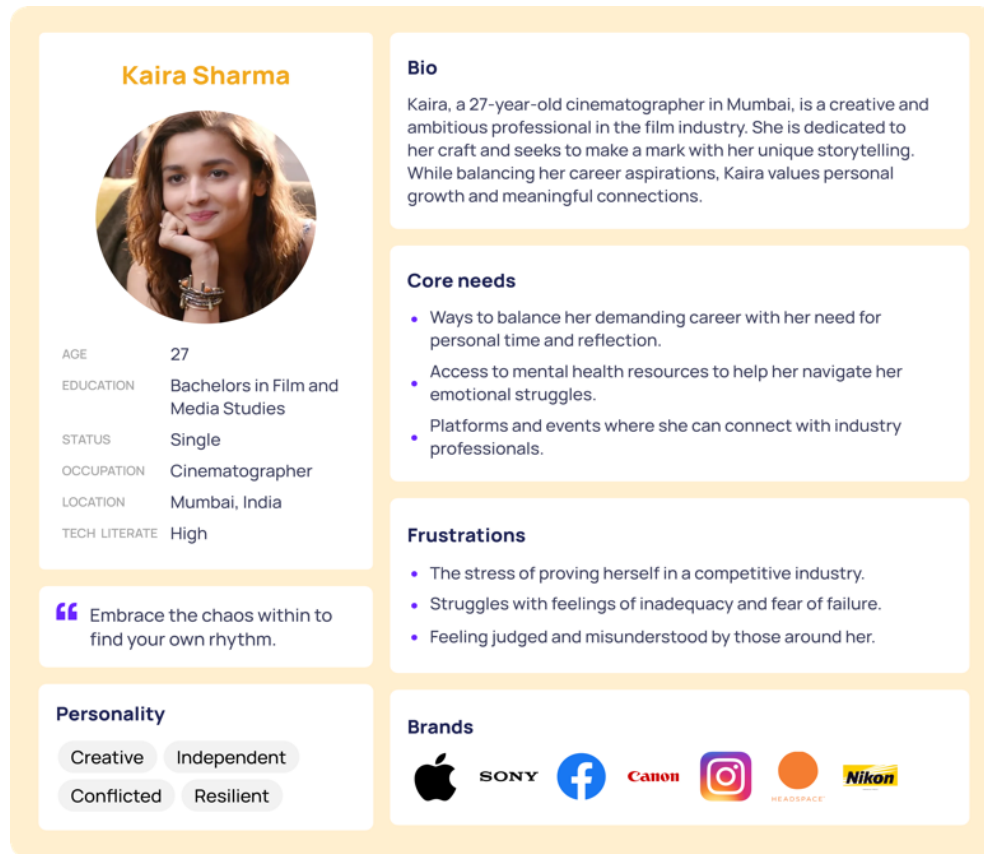


Figure 2 User persona (Kaira)

User (Jacob Martin)

- **Description:** Jacob is a retired school teacher. He needs spirituality and mental peace in his retirement. He is finding ways to manage stress and trying to find meaning in daily life.
- **Assumption:** Jacob searched for a mindfulness application and finds our application as a useful service to get mental clarity.
- **Requirements**
 - Jacob has a good internet connection.
 - Jacob has a device to access our application.

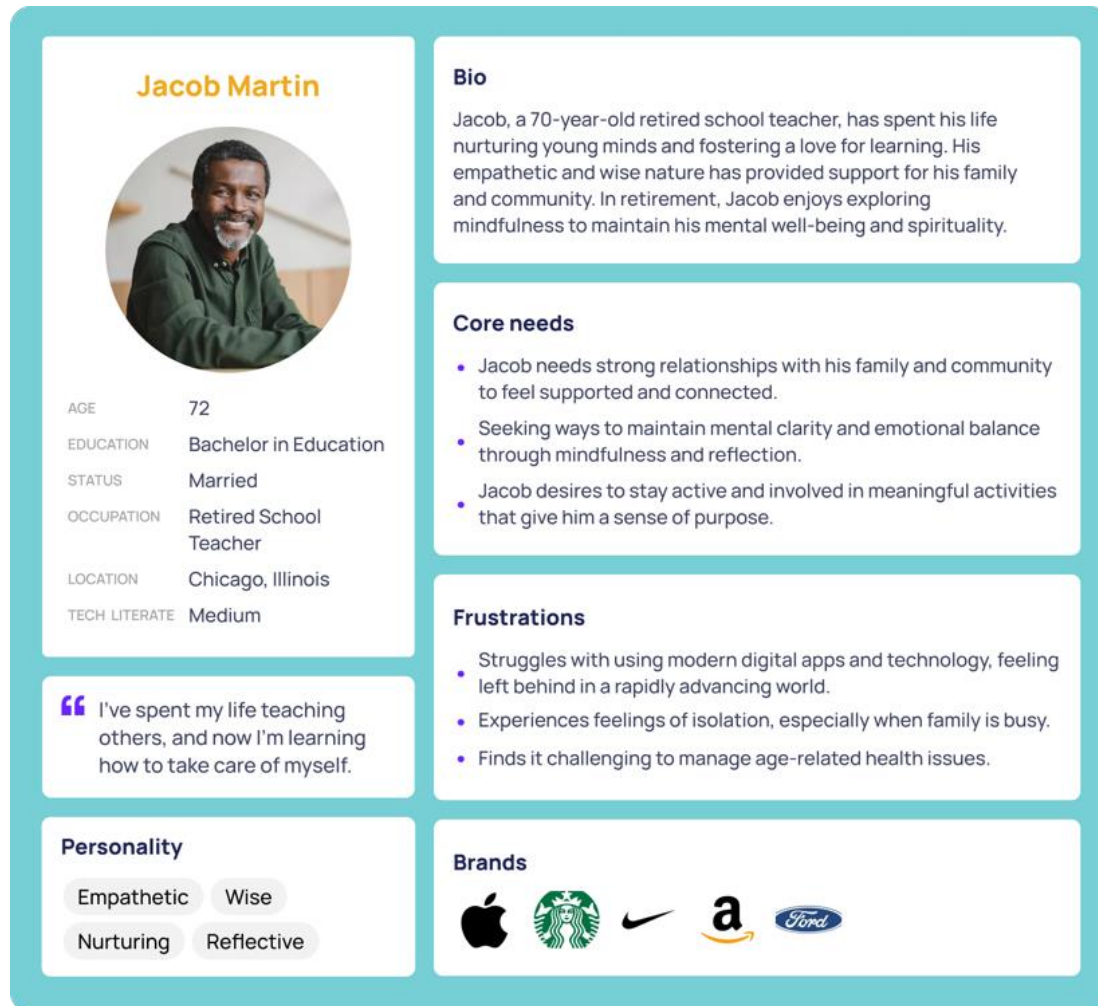


Figure 3 User persona (Jacob)

A2.2.2 User scenario for each task in the feature

Task 1: Live Session creation (User Persona: Dr. Jehangir)

User scenario: Dr. Jehangir Khan wanted to share some insightful meditation techniques with the users. He quickly logged into our application and from the live session module created a session for the next Monday from 3 to 4 PM.

Use Case:

1. Dr. Jehangir Khan opens the web browser and navigates to the application's website.
2. The system displays the login page.
3. Dr. Jehangir Khan enters his username and password, then clicks the login button.
4. The system authenticates Dr. Jehangir Khan's credentials.
5. Dr. Jehangir Khan navigates to the live session module within the application.
6. Dr. Jehangir Khan selects the option to create a new live session.

7. The system prompts Dr. Jehangir Khan to input session details such as title, date, time, and duration.
 - 7.1. If Dr. Jehangir Khan enters invalid session details, such as missing required fields or conflicting time slots, the system displays an error message indicating the issue.
 - 7.2. The system prompts Dr. Jehangir Khan to correct the session details.
 - 7.3. Dr. Jehangir Khan reviews the error message and adjusts the session details accordingly.
 - 7.4. Dr. Jehangir Khan resubmits the session creation request.
 - 7.5. If the session details are still invalid, the system repeats the error message and correction process until valid session details are provided or Dr. Jehangir Khan cancels the session creation.
8. Dr. Jehangir Khan enters the correct session details, specifying Monday as the day, 3 PM as the start time, and 4 PM as the end time.
9. Dr. Jehangir Khan confirms the session creation.
10. The system saves the session details and notifies Dr. Jehangir Khan of successful creation message.

Task Flow diagram:

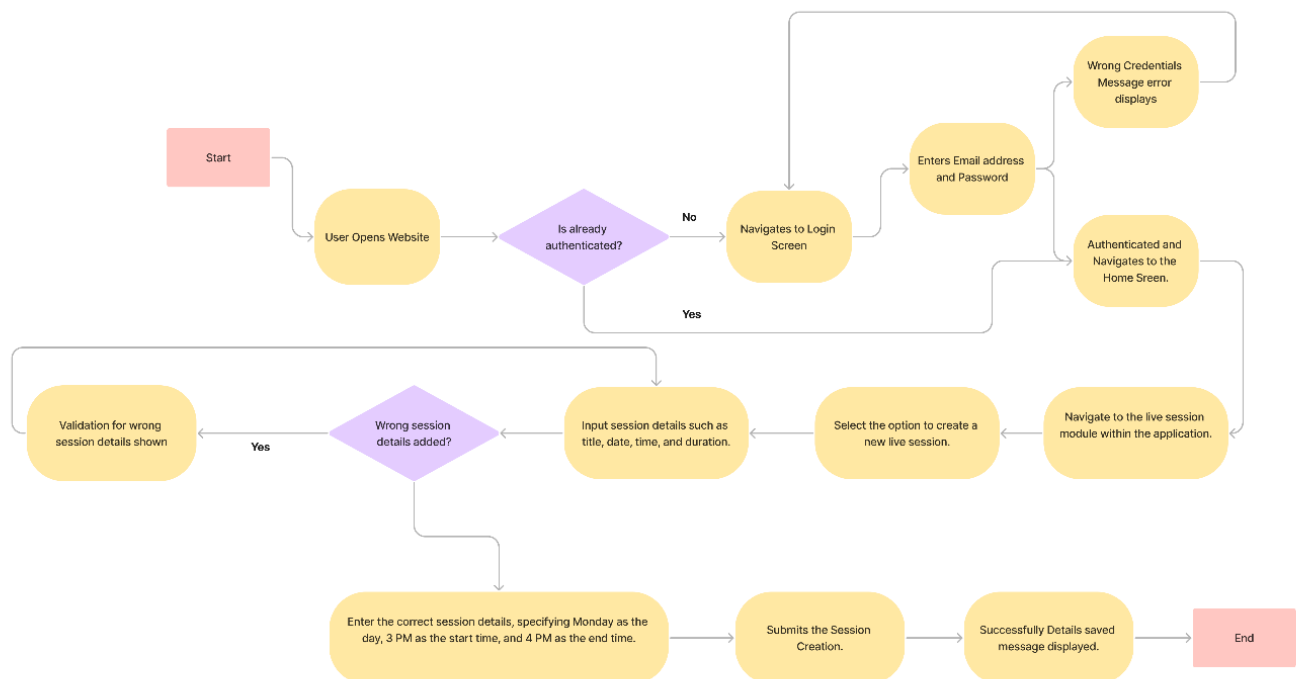


Figure 4 Task flow diagram 1 (Session creation)

Wireframe:

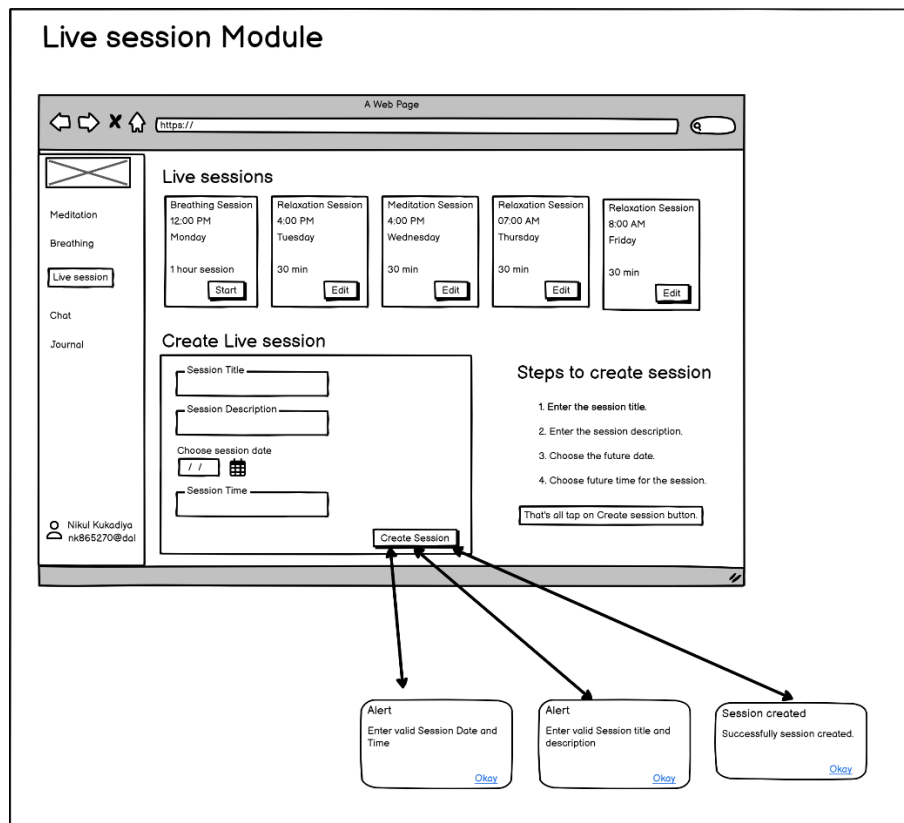


Figure 5 Wireframe (Create Session)

Task 2: Live Session management (User Persona: Dr. Jehangir)

User scenario: Dr. Jehangir Khan realised that he has appointment scheduled on the Monday, so he needs to update the session timing. He logged into application and tap on edit on the live session card and edited the schedule from Monday to Tuesday.

Use case:

1. Dr. Jehangir Khan opens the web browser and navigates to the application's website.
2. The system displays the login page.
3. Dr. Jehangir Khan enters his username and password, then clicks the login button.
4. The system authenticates Dr. Jehangir Khan's credentials.
5. Dr. Jehangir Khan navigates to the live session module within the application.
6. Dr. Jehangir Khan locates the scheduled session for Monday within the list of sessions.
7. Dr. Jehangir Khan selects the Monday session that needs to be rescheduled.
 - 7.1. Mistakenly He selected another session.
 - 7.2. He checked the session details and identified this as the wrong session.
 - 7.3. He taps back to go to the previous screen where the session listing is available.
 - 7.4. He chose the right session and clicked on it.
8. The system displays the details of the selected session, including date, time, and duration.
9. Dr. Jehangir Khan chooses the option to modify the session details.
10. Dr. Jehangir Khan adjusts the session date to Tuesday, ensuring it does not conflict with his appointment.
11. Dr. Jehangir Khan confirms the changes to the session.

The system updates the session details accordingly and notifies Dr. Jehangir Khan.

Task Flow diagram:

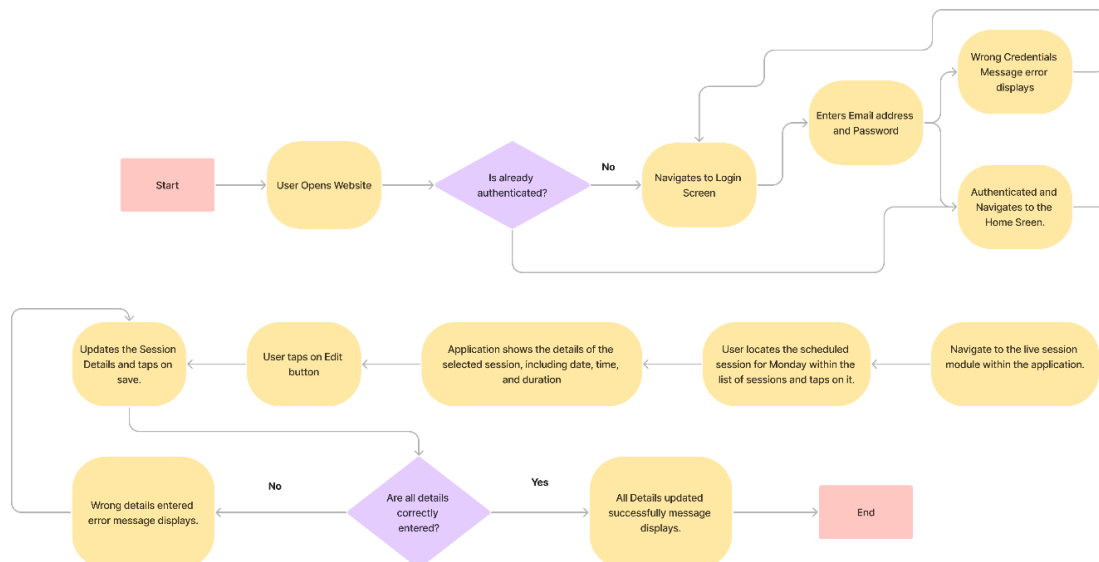


Figure 6 Task flow diagram 2 (Session Edit)

Wireframe:

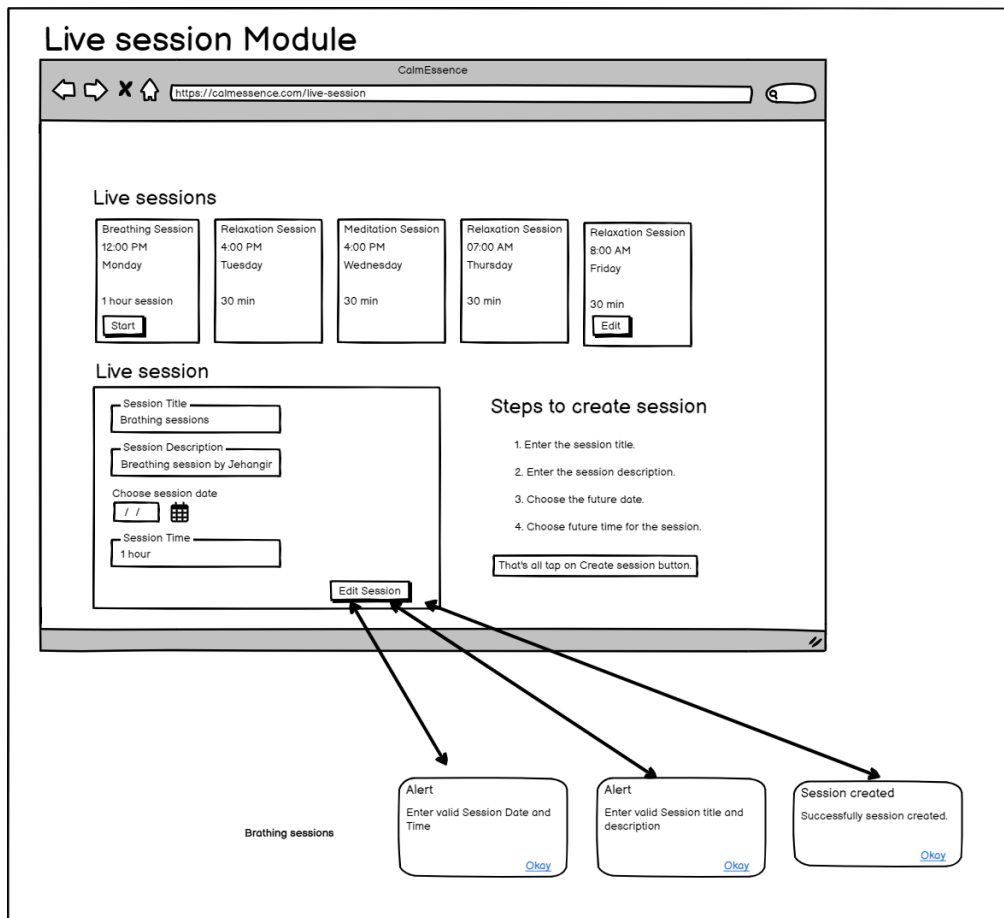


Figure 7 Edit session wireframe

Task 3: Live session joining and user sessions

User scenario: At 3 PM on Tuesday, Dr. Jehangir Khan receives an alert for the live session. He accesses the web application, selects the live session option, and initiates the session, enabling other users to join him for meditation updates.

Use case:

1. Dr. Jehangir Khan opens the web browser and navigates to the application's website.
2. The system displays the login page.
3. Dr. Jehangir Khan enters his username and password, then clicks the login button.
4. The system authenticates Dr. Jehangir Khan's credentials.
5. Dr. Jehangir Khan navigates to the live session module within the application.
6. Dr. Jehangir Khan locates the scheduled session for Tuesday within the list of sessions.
7. Dr. Jehangir Khan selects the Tuesday session that needs to be started.
8. He taps on the start button and redirects to the live session screen.
9. On the Session screen, application asked for the camera and audio permission
10. Dr. Jehangir accepted the permission and live session has been started.
11. Dr. Jehangir Khan started the session and now tapped on the End session button.
11. Session has been ended and Dr. Jehangir khan redirected to session list screen.

Task Flow diagram:

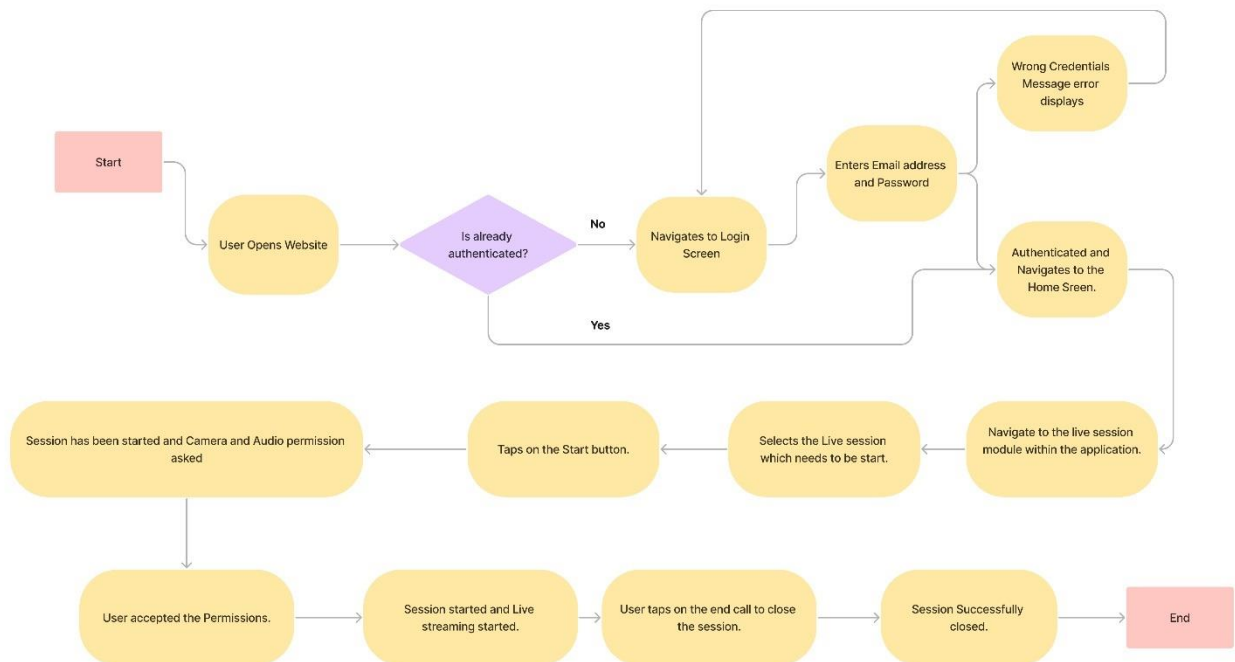


Figure 8 Task flow diagram 3 (Live Session Streaming)

Wireframe:

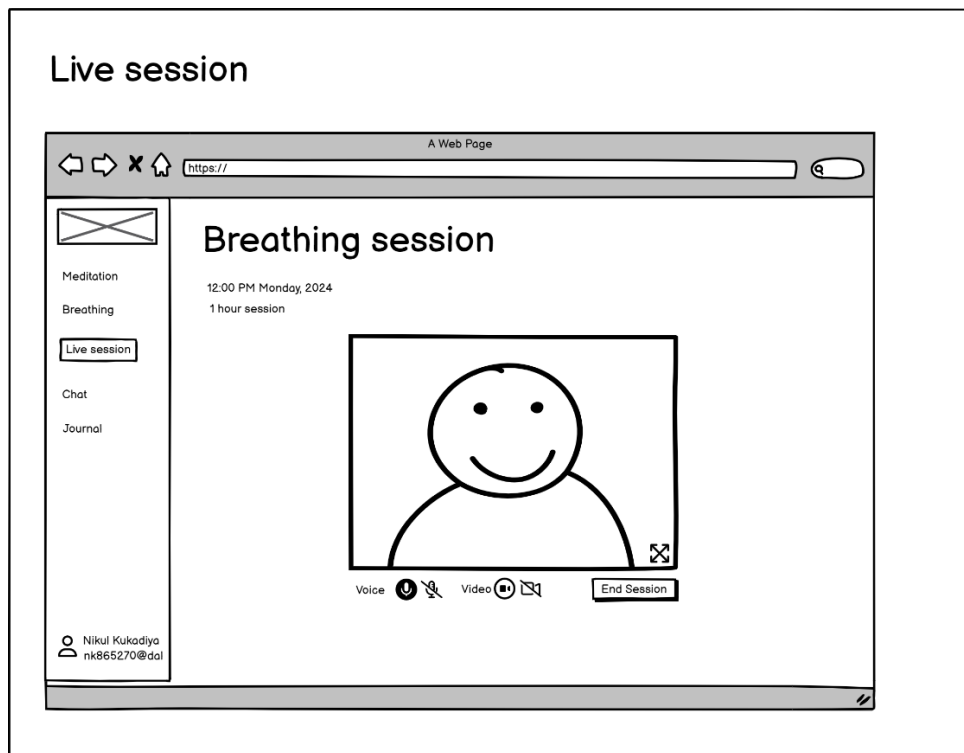
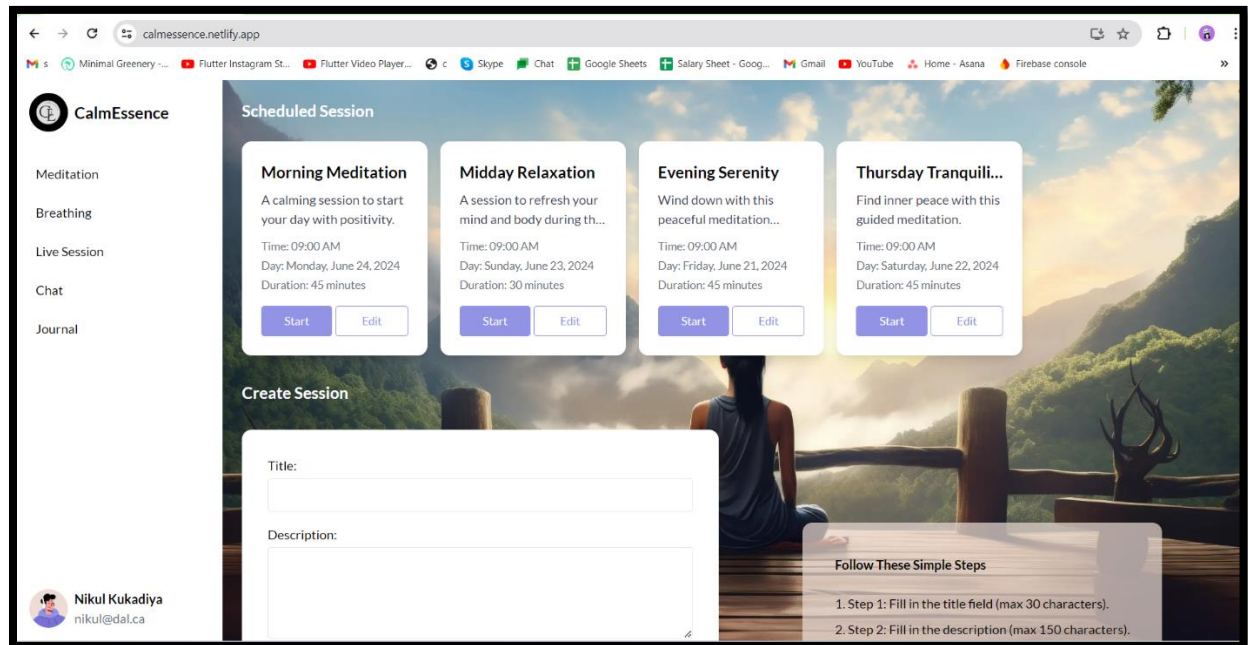
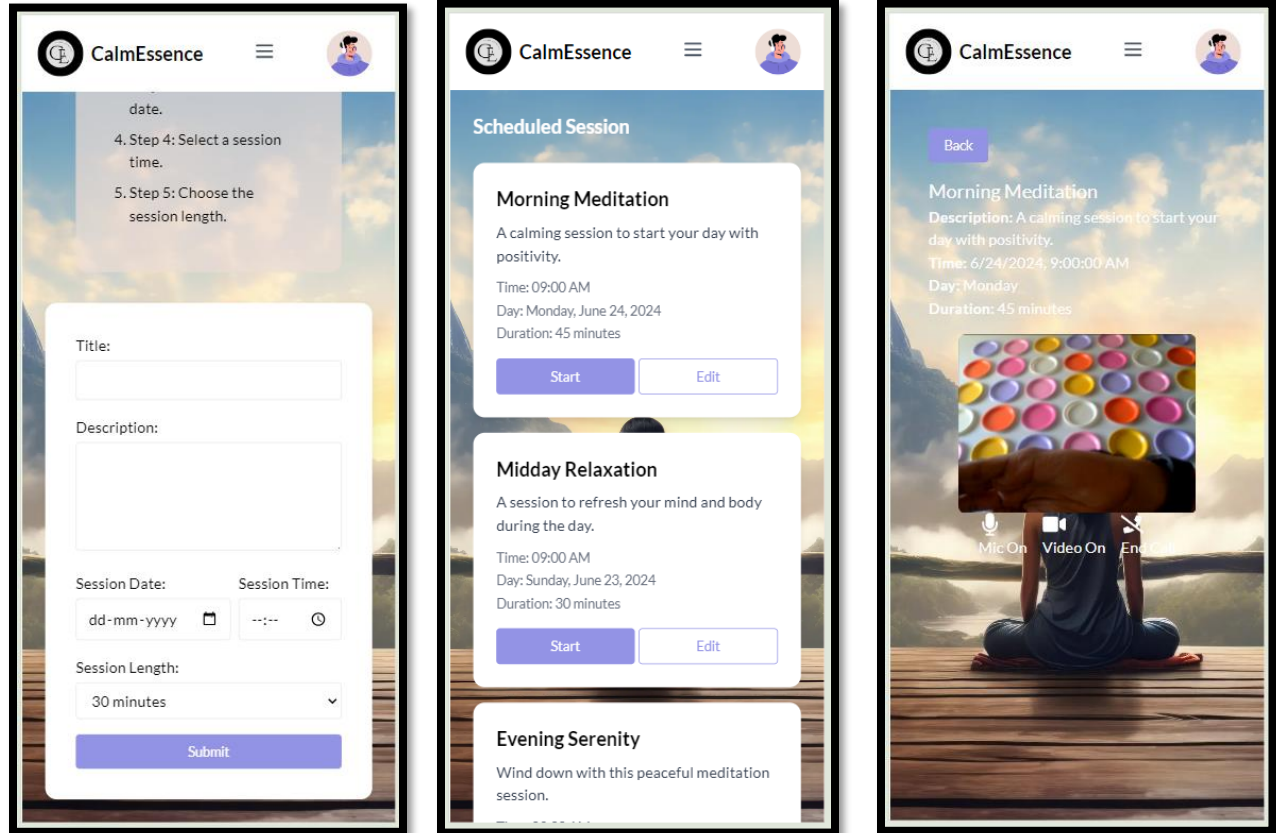


Figure 9 Wireframe Live session.

A2.2.2 High fidelity for tasks



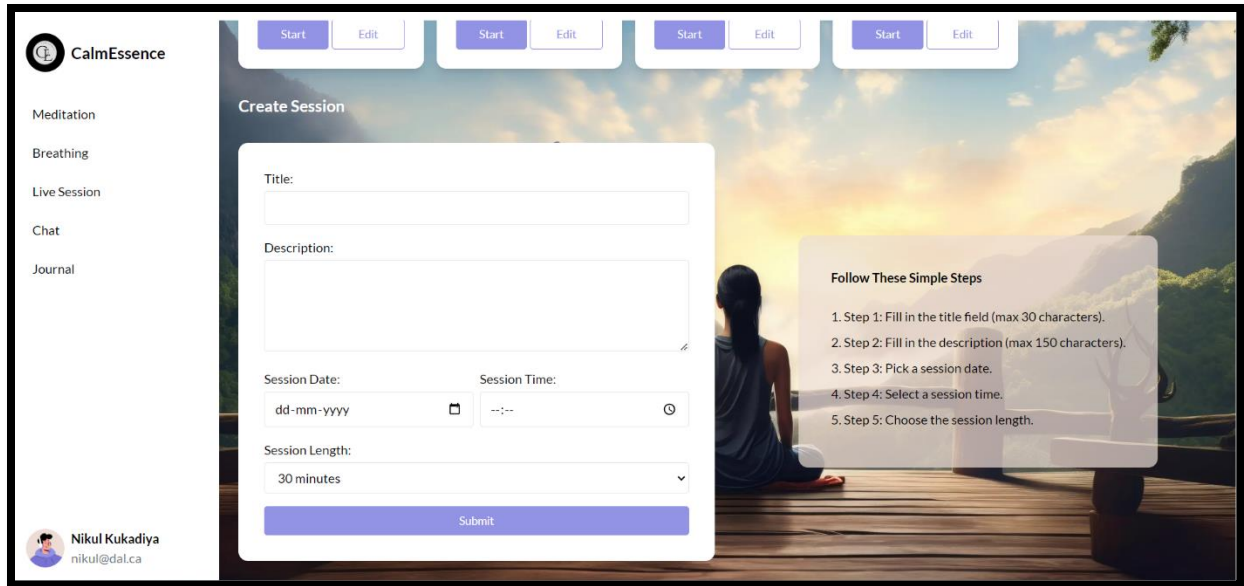


Figure 10 Working site screenshots.

A2.2.2 Application architecture

Below figure shows the architecture of the application in which we are showing how the application will be connected to the backend services and the database. Firstly, the web browser will be loading the react application, which will load all the pages from the deployed application. Once the react app is loaded then APIs will be sent to specific endpoints defined in the Node.js and Express.js backend. The requests are being redirected to the app routes defined in the Node.js application and will be redirected to the controllers of the backend application. Controllers relate to the Database service and will fetch, update and insert the data into the MySQL database. Once the query has been executed it will be redirected the response to the frontend react app and then components will render the result on the frontend side.

Frontend: React framework will be used for rendering the views with the use of Tailwind CSS for the stylings.

Backend: Node.js and Express.js will be used for application APIs and will be configured with the database using the database models.

Database: MySQL is the database layer that will be used to store the data. SQL connect will be used for creating and managing the connection with the node and database.

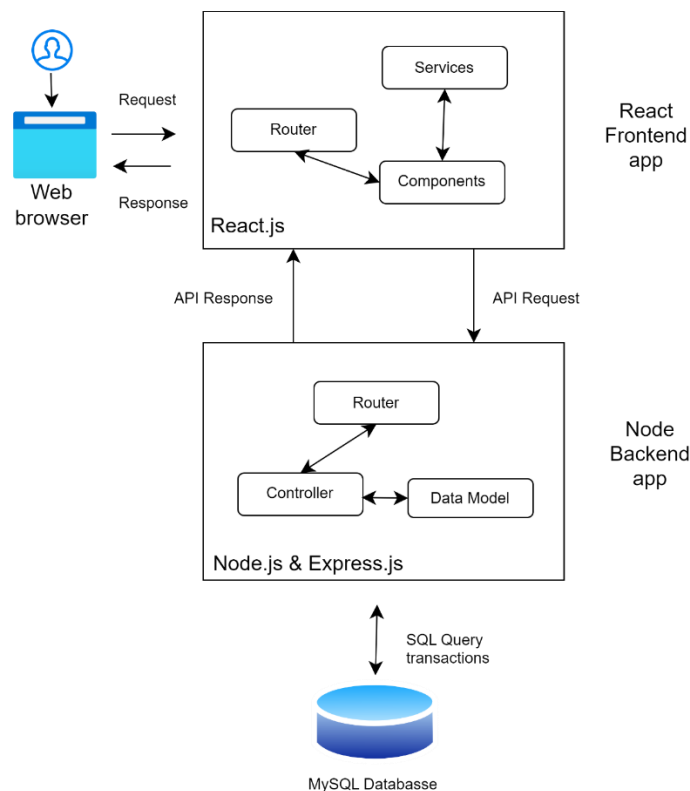


Figure 11 Application architecture

A2.2.2 Interaction design

Task 1: Create session click stream

Figure illustrate the click stream diagram of the create session from the calm essence website and one by one he/she will create the session.

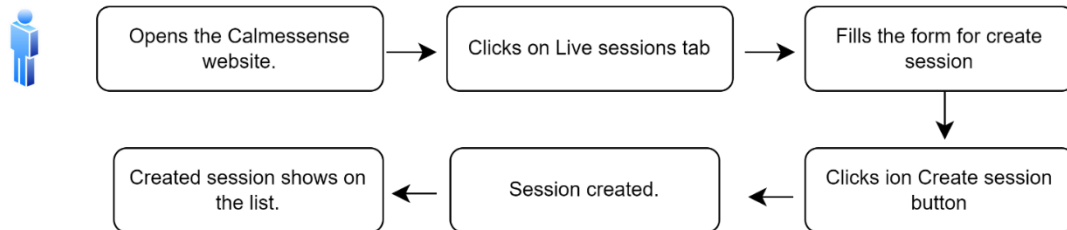


Figure 12 Create session click stream

Task 2: Edit session click stream

Figure illustrates the click stream diagram of the edit session from the calm essence website and one by one he/she will edit the session.

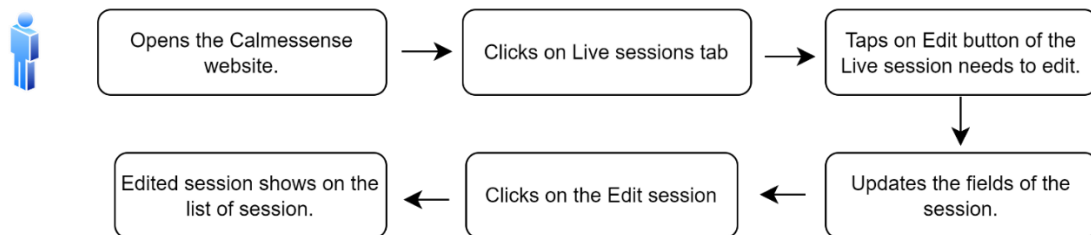


Figure 13 Edit session click stream

Task 3: Start Live session click stream

Figure illustrates the click stream diagram of the live session joining from the calm essence website and one by one he/she will start and stream the session.

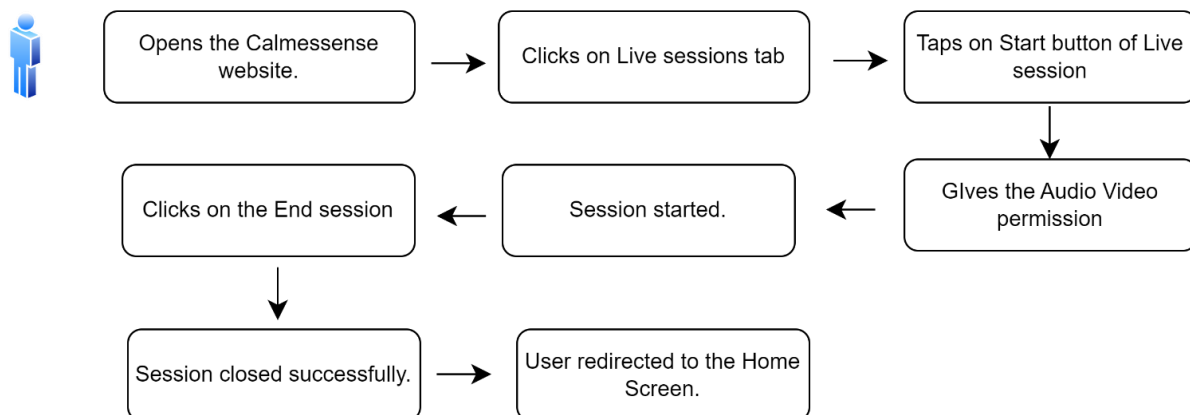


Figure 14 Live session streaming click stream

Process / Service Workflow:

The process and service workflow of the Calm essence will be as below. The workflow will start from the user when they want to use our application. Then, the react routes will be renders the components and those components will be shown in the frontend web browser. Once a user starts accessing the application, the react app sends the API requests to the Node.js application which will fetch, insert or update the data from the MySQL database. Once the query execution completes the response has been returned to the frontend react screen. React components will render the results to the user.

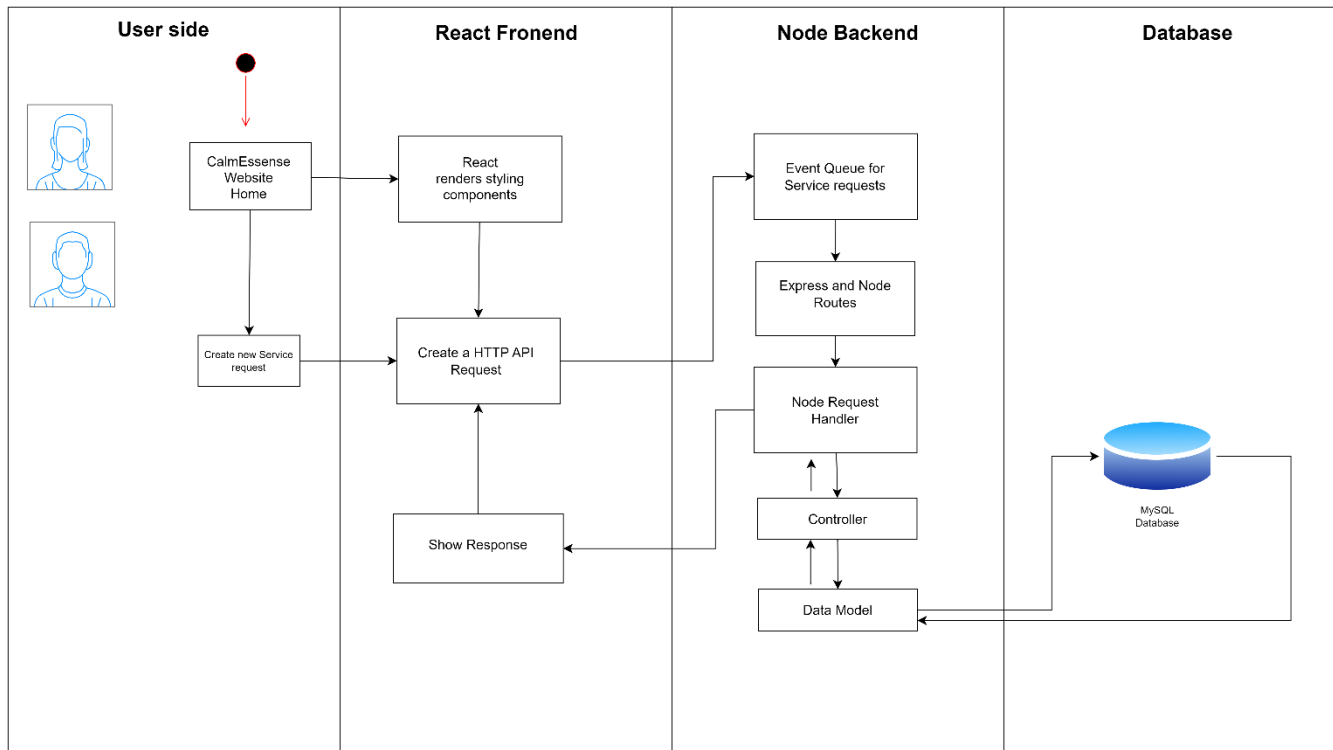


Figure 15 Process/Service workflow

How the feature will be triggers in the process above.

Task 1: User will be opens the application, react components will render the view, User clicks on the Live session module which generates the HTTP API request to fetch all live sessions from the Database. Once the API request is fulfilled the response will be returned to the frontend rendering the components. User enters the details in the form and taps on the create session button. API will be called from frontend and will generate a new entry in the database and returns response to frontend regarding successfully created session.

Task 2: User will open the application, react components will render the view, User clicks on the Live session module which generates the HTTP API request to fetch all live sessions from the Database. User selects the session which will be edited and taps on the Edit session button. A new API request will be generated and will be update performs to the database via NodeJS.

Task 3: User will open the application, react components will render the view, User clicks on the Live session module which generates the HTTP API request to fetch all live sessions from the Database. User selects the session which will be start and taps on the Start session button. User redirects to the Live session screen and all the session details fetched. A new Token for streaming generated via agora SDK and will be enables the live Streaming service. User will tap on end session and session will be terminated.

Folder Structure:

We have 2 folder structures one for the Frontend (React.js [4]) and Backend (Node.js) both folders have different files and directory structures as per the framework.

Frontend (React.js [4]) files are stored in the calmessence folder in the root directory of the GitLab repo. In which the entry point of the application is App.js which contains all the routes of the application. Inside the components folder all the working pages will be stored. All the pages have their own react components which will be shown as per the page requests and requirements as well as the API requests are store in these files. In the root of the repo the Tailwind CSS[5] configuration file has been stored in which the theme of the application will be placed.

Backend (Node.js) files are store in the backend folder in the root directory. In which the routes folder contains file which will be routed all the requests to respective controllers. Controllers are placed inside the controller directory and database connection configuration are stored in databaseService.js file.

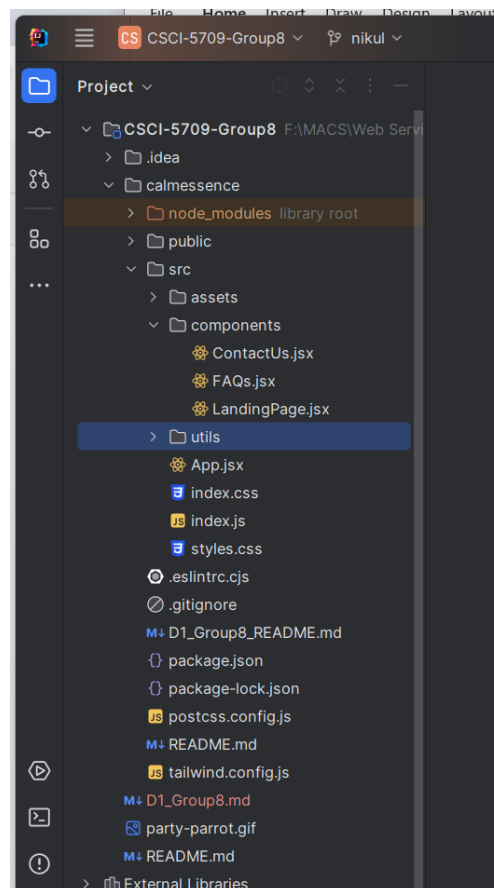


Figure 16 React App folder structure.

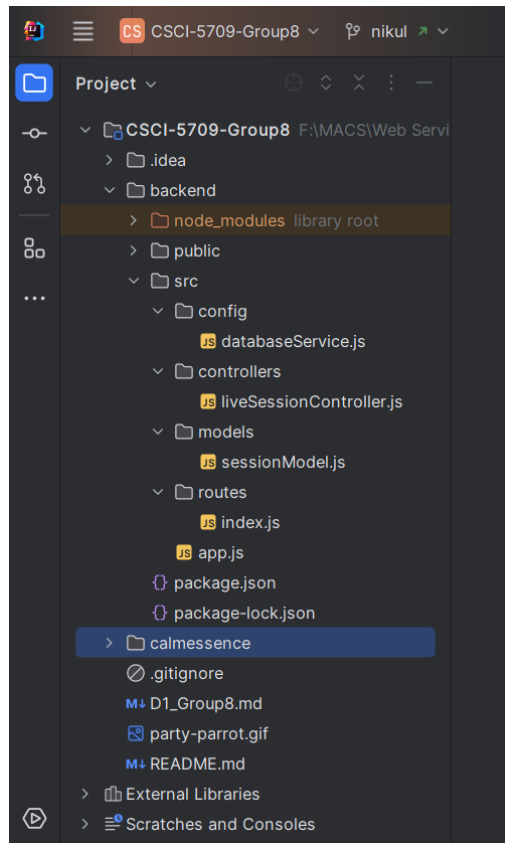


Figure 17 Backend application folder structure.

References

- [1] "Figma", *Figma Inc.* [Online]. Available at: <https://www.figma.com/> [Accessed: 08 July 2024]
- [2] "FigJam", *Figma Inc.* [Online]. Available at: <https://www.figma.com/figjam/> [Accessed: 08 July 2024]
- [3] "Balsamiq Cloud", Balsamiq Studios LLC. [Online]. Available at: <https://balsamiq.cloud/> [Accessed: 08 July 2024]
- [4] "React", React. [Online]. Available at: <https://react.dev/> [Accessed: 05 June 2024]
- [5] "Tailwind CSS", Tailwind Labs Inc. [Online]. Available at: <https://tailwindcss.com/> [Accessed: 05 June 2024]