PROJECT REPORT

on

Design and Development of Digital Health Applications for Healthcare and Mental Wellness

UNDER SUPERVISION OF

Dr. Sujata Pal

Department of Computer Science and Engineering

Indian Institute of Technology Ropar

SUBMITTED BY:

Cherika Kaushal

B.Tech, Computer Science & Engineering

Punjabi University, Patiala

SUMMER RESEARCH INTERNSHIP 2026

1 Introduction

1.1 Background

Longitudinal health research depends on continuous and structured data collection to monitor changes in participants over time. During this internship, I focused on developing the software infrastructure for two digital health research applications: the Women's Health Longitudinal Research Platform and the Stress Management Research Platform. Both platforms support structured participant data collection along integration with physiological data extracted from a custom wearable device, and a researcher dashboard for reviewing and managing the collected data.

1.2 Objective

The internship is organised around the following objectives:

1. To design and develop mobile and web applications for women's health monitoring and stress management.

2. To integrate a wearable data collection device with the participant application and validate end-to-end physiological data transmission and storage.

The first objective focuses on the development of two research platforms. The Women's Health Platform stores survey responses, symptom reporting, menstrual tracking, and menopause-related monitoring. The Stress Management Platform supports session-based stress monitoring by combining app responses with physiological data. The second objective focuses on integrating the custom wearable device with the software platform. Physiological data is received through the app, transmitted via REST APIs, validated by the backend and stored in MongoDB, analysed via dashboard for further analysis.

1.3 Scope of Work

The internship covers the complete software development of both platforms. The major responsibilities include participant app development, researcher dashboard development, UI/UX design, backend and REST API development, MongoDB schema design, authentication, email verification and wearable data integration, testing, and version control using Git and GitHub.

1.4 Motivation

Longitudinal health studies often rely on separate tools for participant onboarding, survey management, physiological data collection, database storage, and data analysis. Devel-

oping both platforms within a shared software framework provides greater control over the overall research workflow. Common components such as authentication, API validation, database connectivity, email verification, and researcher access are shared across both platforms, while their individual research workflows and datasets remain separate. Another key motivation is to support a custom wearable device for physiological data collection.

1.4.1 Wearable Technology Evaluation

The following wearable devices were evaluated:

- Fitbit Charge 6
- Oura Ring
- Apple Watch

The evaluation considered following technical and practical aspects:

- Heart rate (HR) monitoring
- Heart rate variability (HRV)
- Sleep tracking capabilities
- Skin and body temperature sensing
- Physical activity monitoring
- Data accessibility through APIs
- Ease of software integration
- Cost of deployment

This comparative analysis helped in identifying the physiological parameters required by the software platform and defined the expected data structure for wearable integration. Based on the research objectives and project requirements, we selected custom wearable physiological sensing device instead of relying on a commercial wearable ecosystem. Accordingly, we designed the software architecture to support easy integration with the custom research device through standardised data synchronisation, storage, and visualisation pipelines. The wearable hardware was developed separately.

2 Project Overview

2.1 Women's Health Longitudinal Research Platform

The Women's Health Longitudinal Research Platform is developed to support the structured collection of data for research on menstrual health, menopause, and the menopausal transition. It includes a participant application that allows users to register, complete questionnaires, record symptoms, and monitor their health over time. A web-based researcher dashboard provides access to participant records, session history, physiological data, and cohort-level trends for research and analysis.

2.2 Stress Management Research Platform

The Stress Management Research Platform is developed to support session-based stress monitoring. It combines self-reported stress responses with physiological measurements collected during each research session. The platform includes a participant application for session-based activities and a researcher dashboard for reviewing individual sessions, participant records, and aggregated research data.

2.3 Shared Research Infrastructure

Instead of relying on commercial fitness-tracker APIs, both platforms support physiological data collected through a custom made research wearable. The wearable hardware and firmware are developed separately. This work focuses on the participant applications, researcher dashboards, backend services, and database infrastructure responsible for receiving, storing, and displaying the collected research data.

2.4 Why a Shared Backend

Although the two platforms address different research objectives, both require structured storage of participant data linked to research sessions and timestamps. To support this, a shared Node.js and Express.js backend is used, while separate routes and MongoDB collections are maintained for each platform.

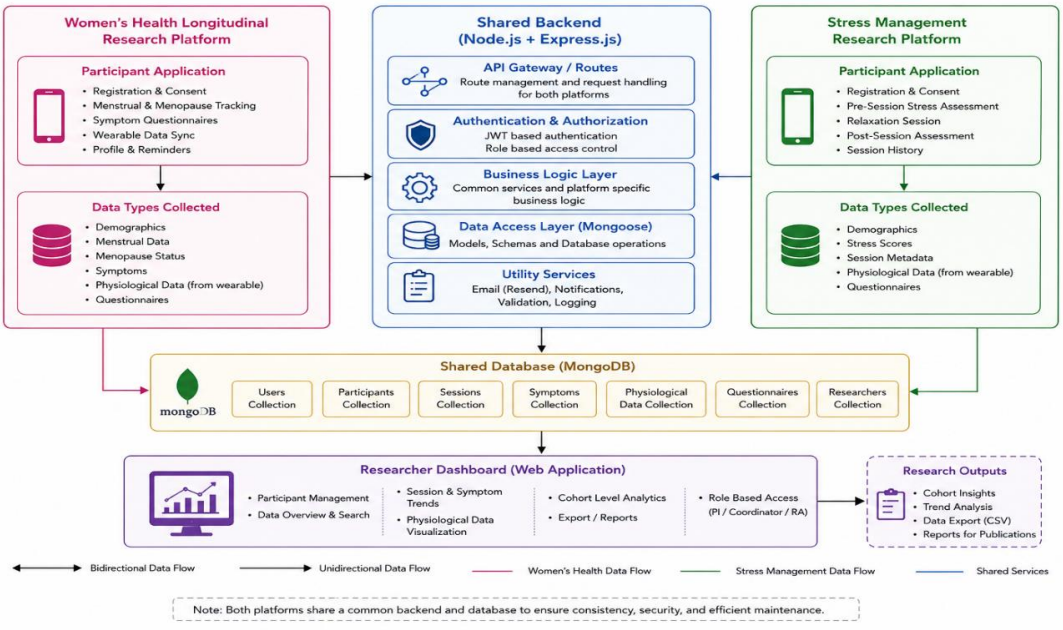

Figure 1. Software architecture of the Digital Health Applications platform

2.5 Technology Stack

Table 1 summarizes the technology stack used across both platforms.

Table 1: Technology Stack

Layer	**Technology**
Frontend	React, TypeScript, Vite
Researcher Dashboard	Next.js, TailwindCSS, ShadCN, Recharts
Backend	Node.js, Express.js
Database	MongoDB (Mongoose ODM)
Authentication	JWT, Brevo Email API
Version Control	Git, GitHub

2.6 System Architecture

The system follows a three-tier architecture consisting of the client application, backend server, and database. The participant application, available on both mobile and web platforms, collects self-reported information along with physiological data associated with a research session. The collected data is transmitted to the backend through REST APIs, where it is validated, timestamped, and stored in MongoDB. The researcher dashboard connects to the same backend through dedicated API endpoints to access participant-level records and cohort-level summaries.

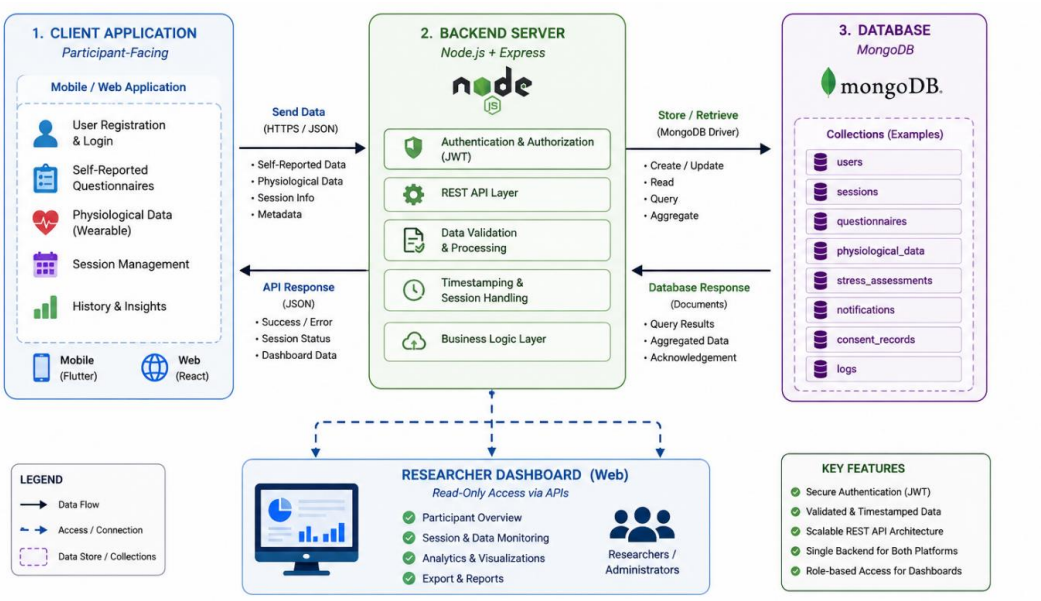

Figure 2. Independent layers

The modular architecture facilitates better maintainability, scalability and code reusability and enables both research platforms to be integrated with common backend services without using the same data or workflows.The platform can therefore offer reliable end-to-end data synchronization, without relying on the wearable hardware implementation.

3 Women's Health Research Platform

3.1 Project Overview

The Women's Health Research Platform is developed to support the systematic collection of data for research on menstrual health, menopause, and related physiological and behavioural changes. It allows participants to complete questionnaires, record symptoms, track menstrual health, and take part in long-term research studies. The platform includes an app for participants and a web-based dashboard for the researcher operating through a shared backend for authentication, data validation, storage, and retrieval.

3.2 Participant Mobile Application

The participant application is developed using React, TypeScript, and Vite, used for Android deployment. It provides a step-by-step workflow for participant registration, login, study participation, questionnaire completion, health tracking, and research session activities and designed the interface for repeated use, along with each screen dedicated to a specific task. The main sections include home screen, tracking page, survey collection and participant profile. The onboarding process includes account registration, email verification, participant identification, consent, and collection of baseline information.

Once logged in, participants can view their pending activities, completed or scheduled questionnaires and their completed sessions

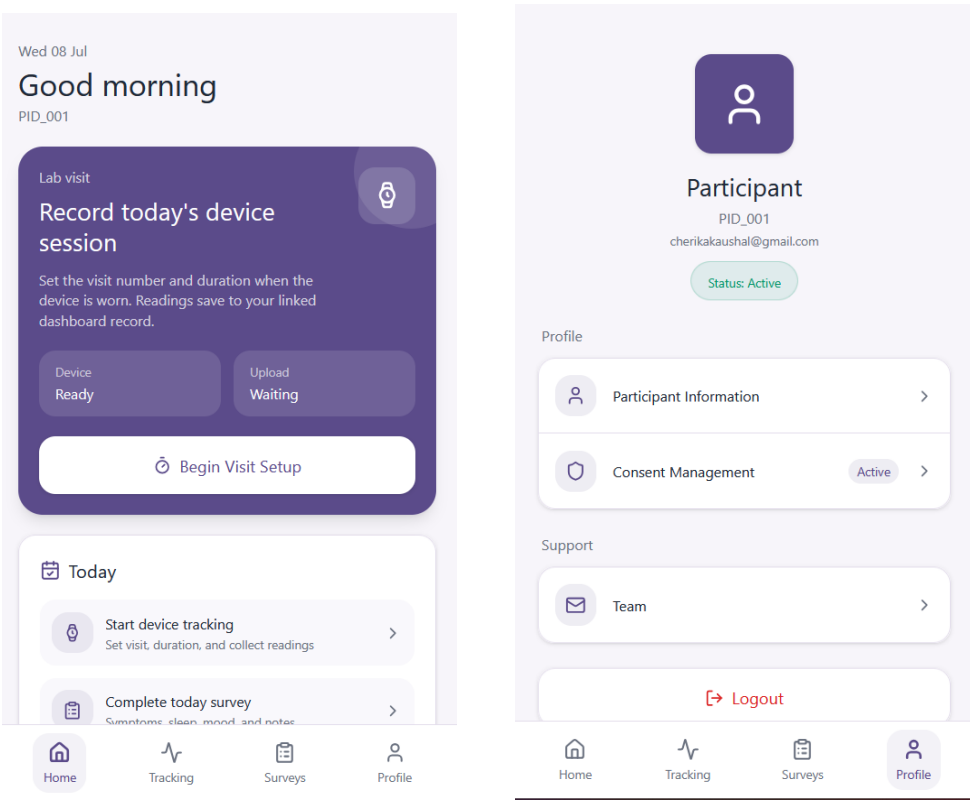

Figure 3. Home page and profile page

3.3 Questionnaire and Symptom-Logging Workflow

The app supports data collection via daily symptom check-ins and weekly, monthly questionnaires. Each response is stored with participant ID (eg. PID_001) and timestamps. This structured approach enables monitoring changes over time, analyse symptom patterns, and compare participant responses throughout.

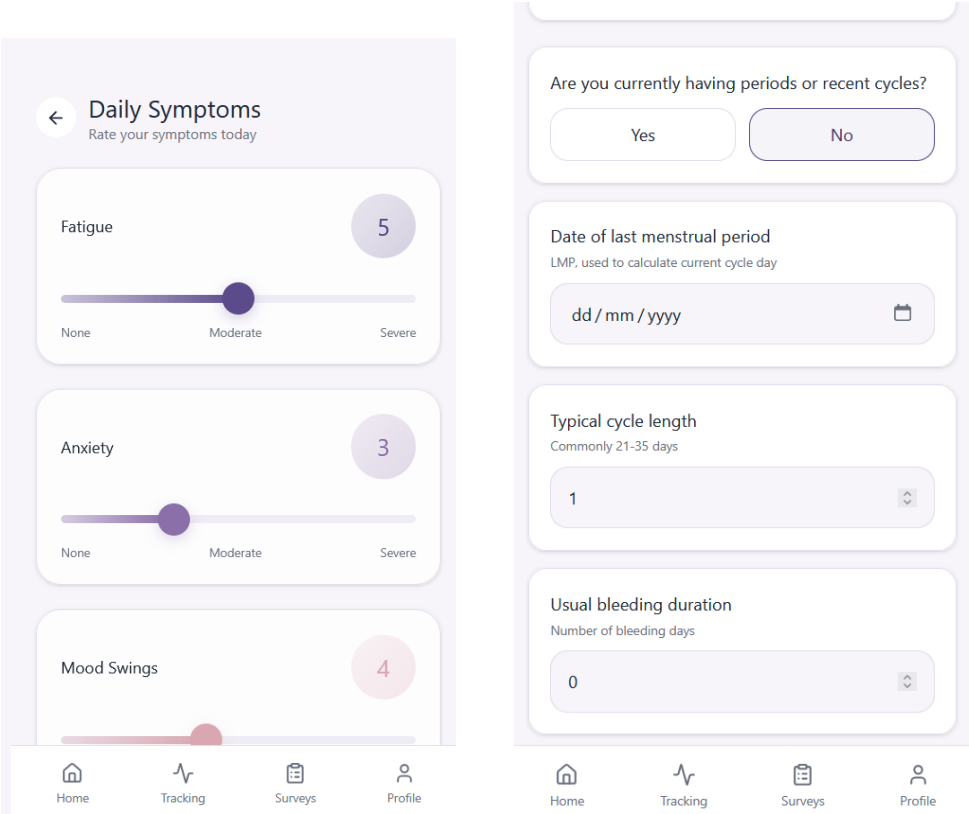

Figure 4. Symptoms survey and questionnaire

3.4 Menstrual and Menopause Tracking

The app includes modules for menstrual tracking and menopause-related questionnaires allowing participants to choose accordingly and record cycle information, symptoms, and health-related observations over time. All records are linked with the participant profiles stored successfully on backend.

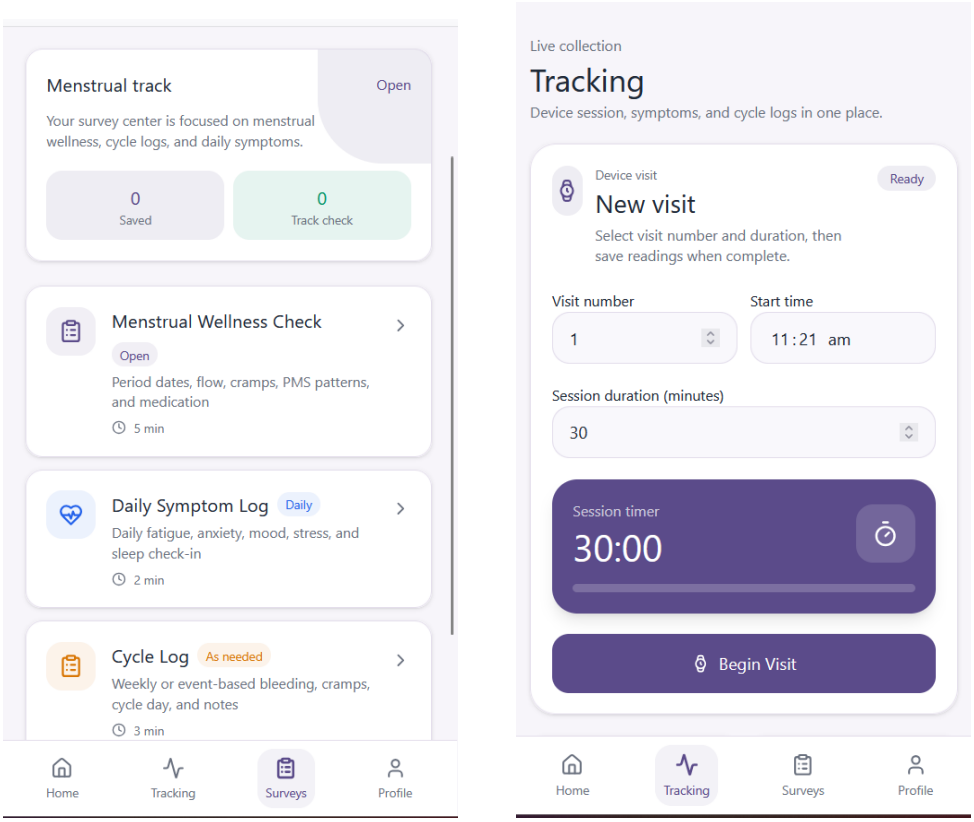

Figure 5. Menstrual track survey and tracking page

3.5 Researcher Dashboard

A web-based researcher dashboard is developed for super administrators and viewers to manage participant records, research sessions, questionnaires, symptoms, physiological data, and consent records. It also includes a dataset export module that enables the super administrator to download structured research data for further analysis.

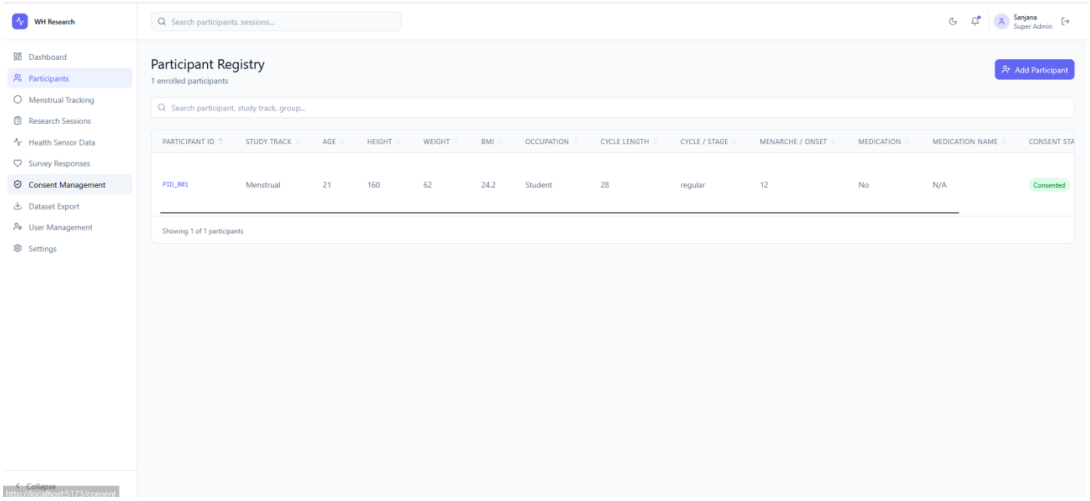

Figure 6. Participant management

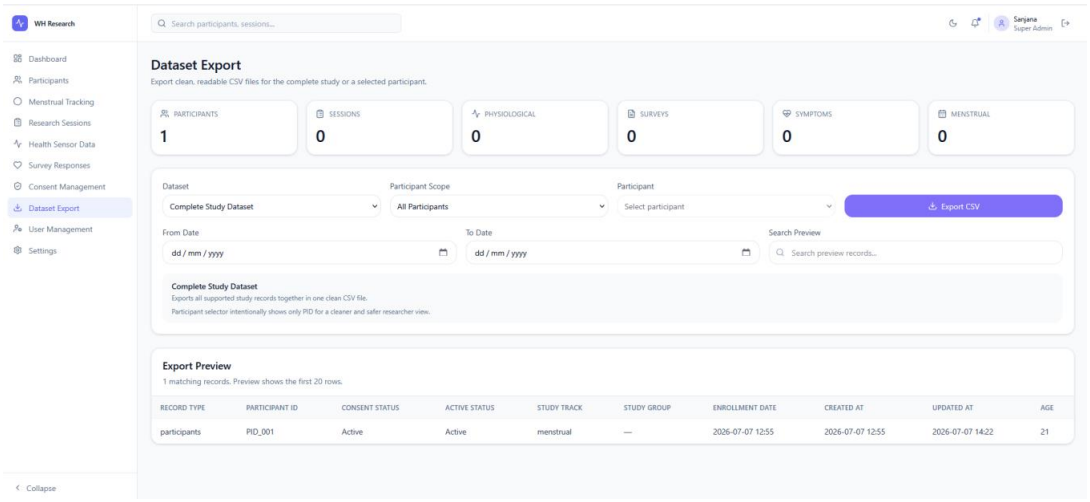

Figure 7. Dataset export

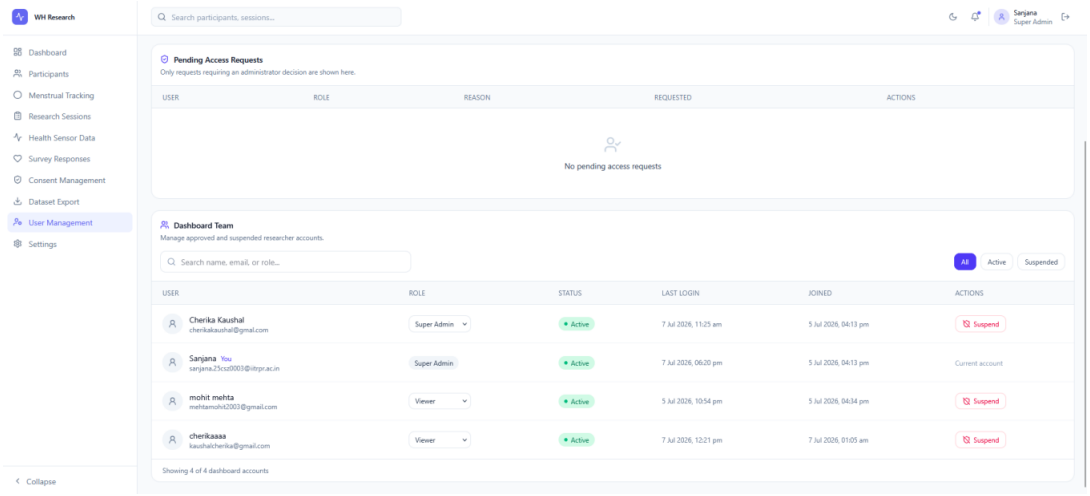

Figure 8. User management

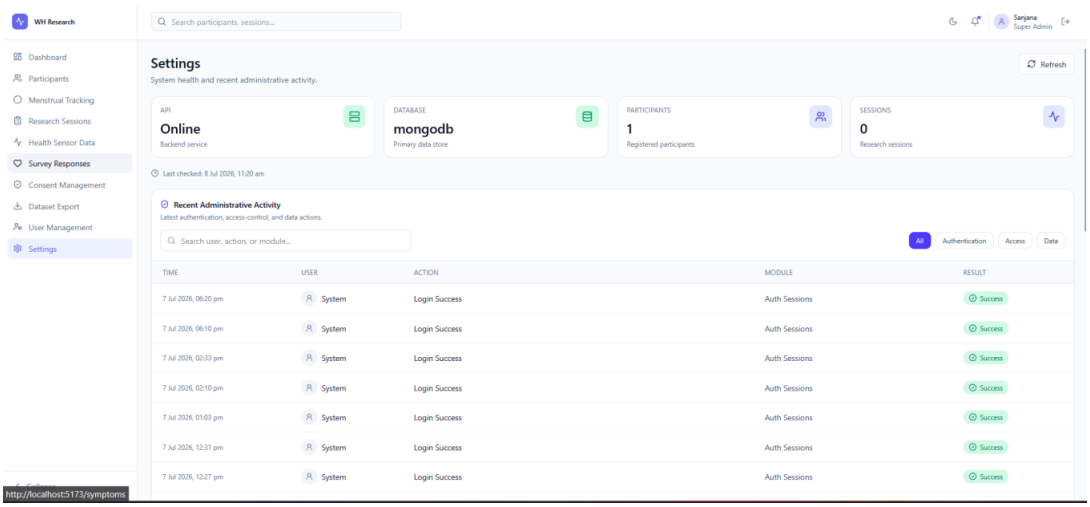

Figure 9. System settings

3.6 Backend, Database, and Authentication

Table 2: Authentication Flow

Step	Action
1	User submits credentials to /auth/login
2	Backend verifies password hash and account status
3	JWT issued with role and participant/researcher ID embedded
4	Client stores token and attaches it to subsequent requests
5	Middleware validates token and role on each protected route

The Women's Health Research Platform is built on the Node.js and Express.js backends, with MongoDB as the main database for storing research and participant data. The REST APIs are structured in different routes to maintain the modularity of the application: authentication, participants, surveys, symptom logs, menstruation, research sessions, physiological data and dashboard operations. Data models are outlined in mongoose schemas, all of which have their own collections: Users, Participants, questionnaire answers, symptom log, menstrual log, research sessions, physiological data, consent log, and researcher notes. Authentication is done with JSON Web Tokens (JWT) with role-based access, along with email verification and email communication regarding the password by using the Brevo API.

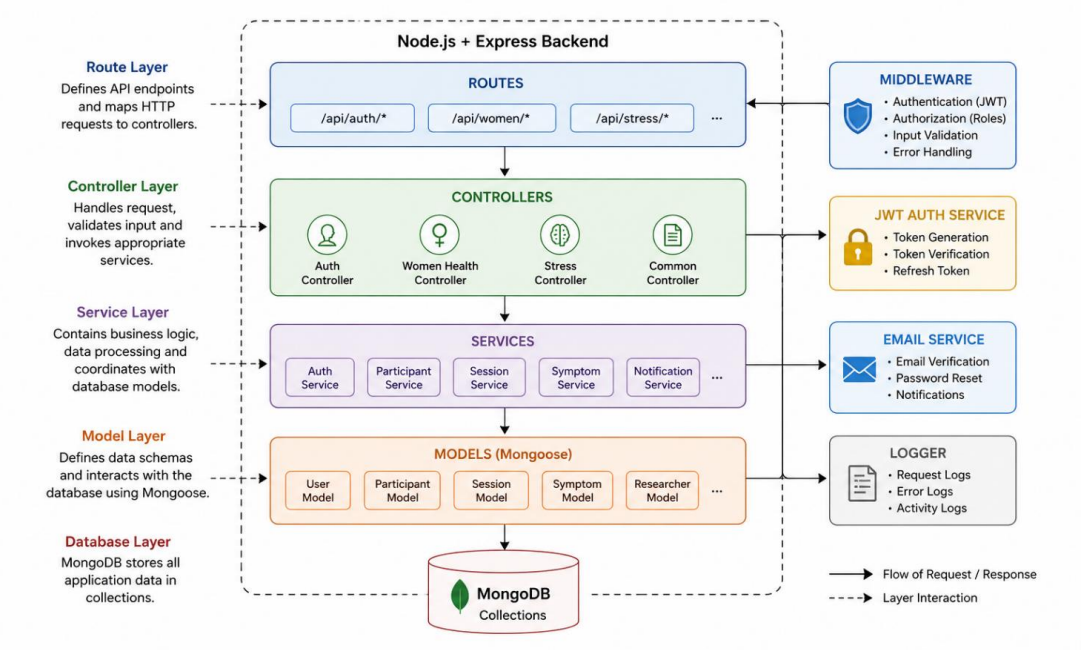

Figure 10. Backend architecture of the Digital Health Applications showing REST APIs, MongoDB, JWT authentication, and email services.

4 Stress Management Research Platform

4.1 Project Overview

The Stress Management Research Platform is developed to support the session-based collection of stress-related data for research purposes. It combines participant-reported responses with physiological measurements collected during every session, allowing researchers to analyse stress symptoms in a structured and organised manner. The platform includes a participant application and a dashboard that operate through a shared backend. To maintain data consistency and traceability, every record is linked to the corresponding participant, research session, and timestamp.

4.2 Participant Application

The participant application is built using React, TypeScript, and Vite. It provides a simple workflow that allows participants to log in, join a research session, complete the required questionnaires, and submit stress-related information.Participants can start a session, complete the assigned activities, record their self-reported responses, and submit the collected data through the application.

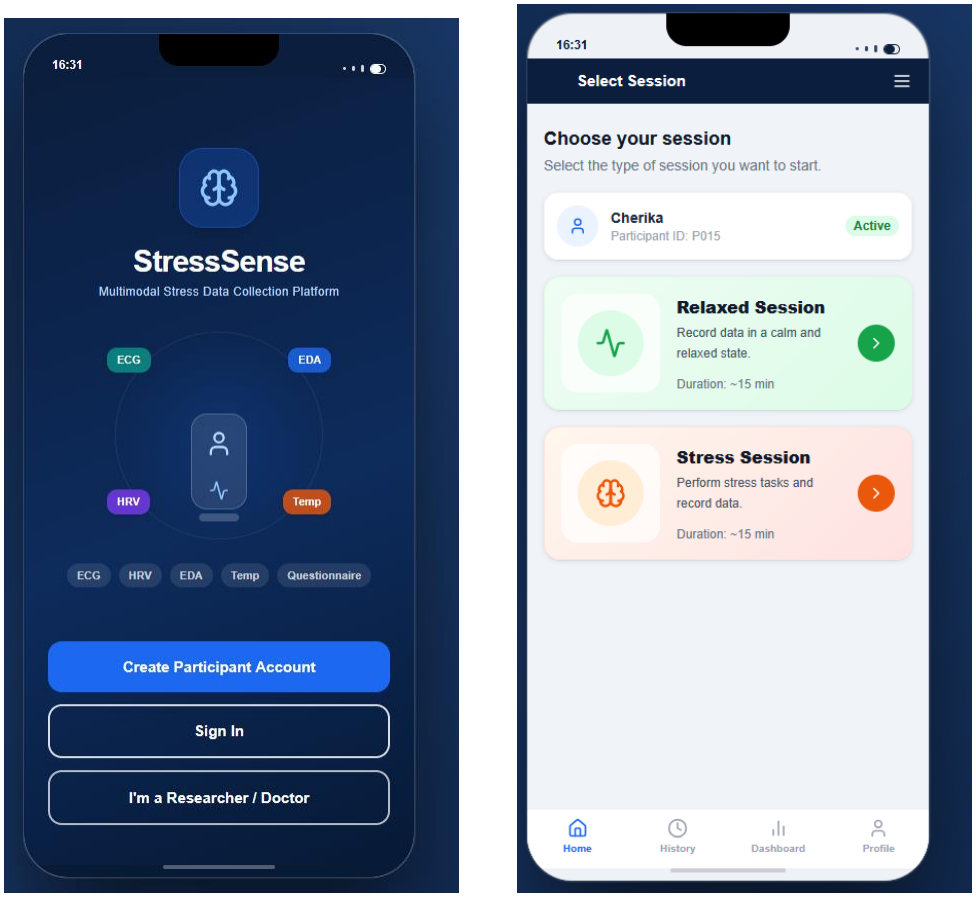

Figure 11. Participant login and stress session selection screen

4.3 Stress Session Workflow

The platform uses a session-based workflow in which each research activity is recorded as an individual session. Each session stores details such as the participant ID (PID_001), start time, duration, status, and questionnaire responses. Physiological measurements collected during the session are linked to the same record, allowing researchers to review questionnaire responses alongside wearable data within a single session.

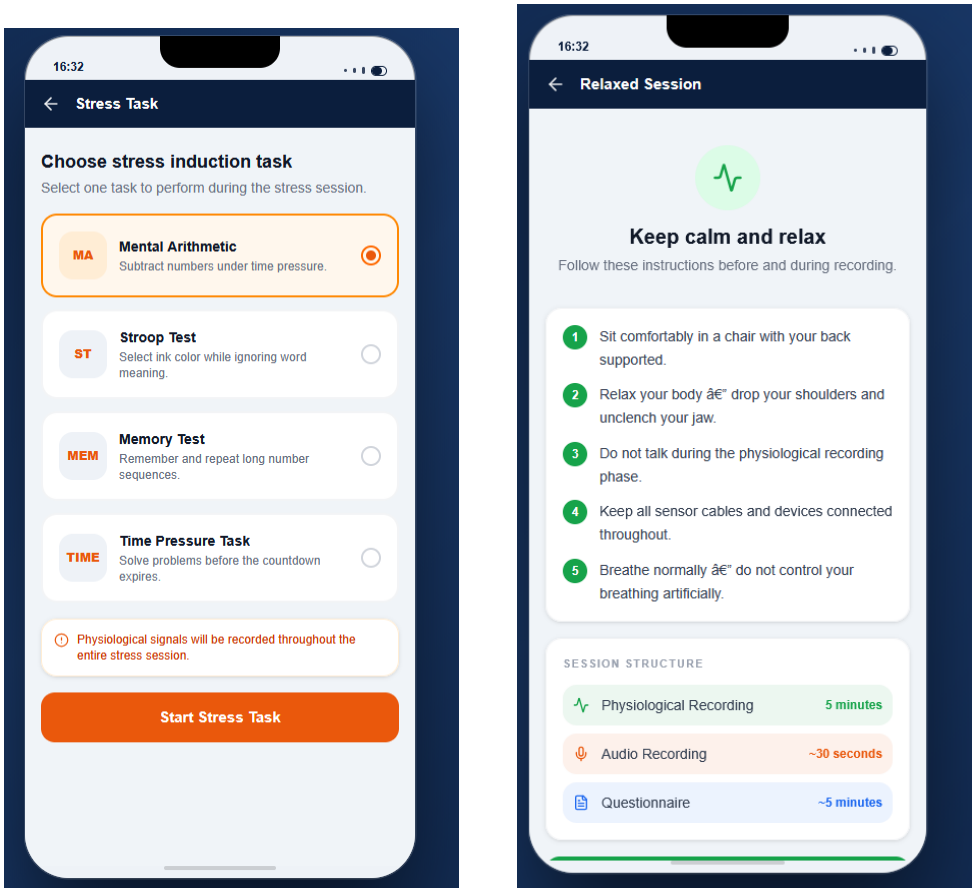

Figure 12. Screens for stress task selection and relaxation sessions.

4.4 Questionnaires and Physiological Data

Structured questionnaires to record participant's observations and levels of stress are included on the platform. The questionnaire data is stored with data from physiological measurements via a custom device that can be worn. The data from the participant application is safely relayed by authenticated REST APIs, checked by the backend and kept in MongoDB for additional analysis.

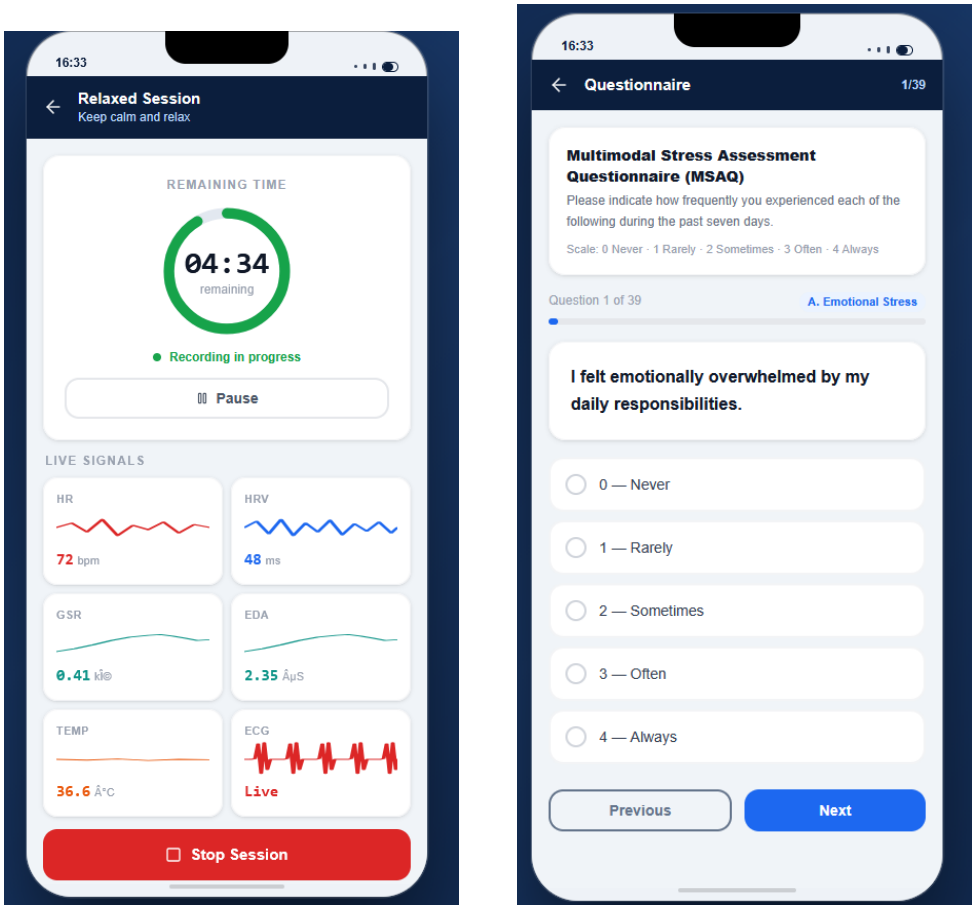

Figure 13. Physiological data monitoring and questionnaire.

4.5 Researcher Dashboard

Dashboard is designed to help authorised person to review participant records, session history, questionnaire responses, and physiological measurements from a single interface.For easy analysis both the individual and overall participant data are presented in tables and summary cards.Protected backend APIs ensures that the information only be available to the authorised person only.

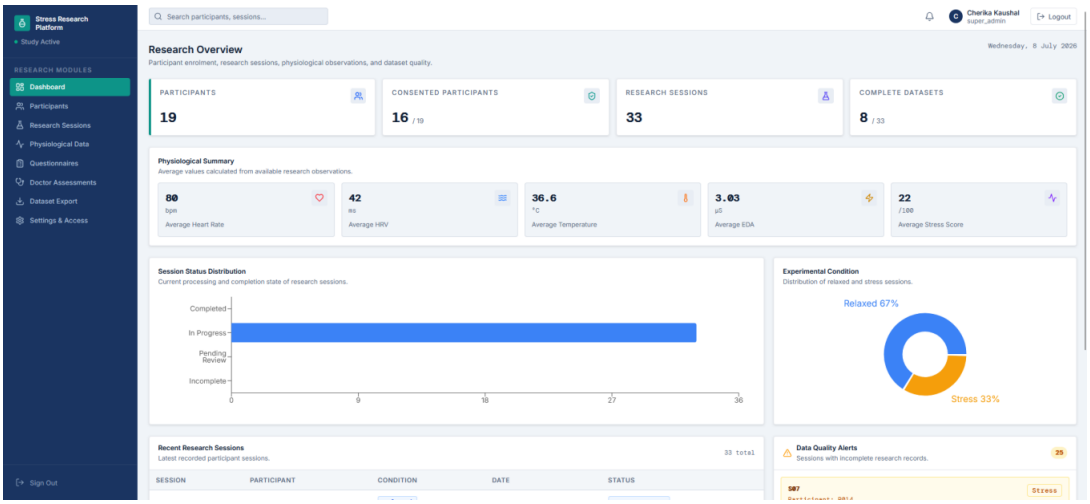

Figure 14. Dashboard showing participants data, physiological measurement and session insights

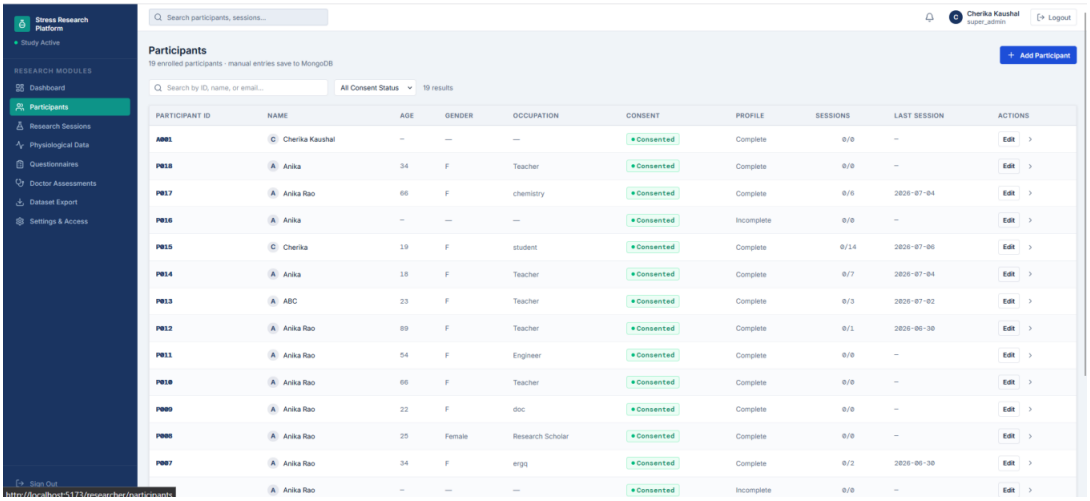

Figure 15. Participant management dashboard for viewing participant profiles, consent record and sessions.

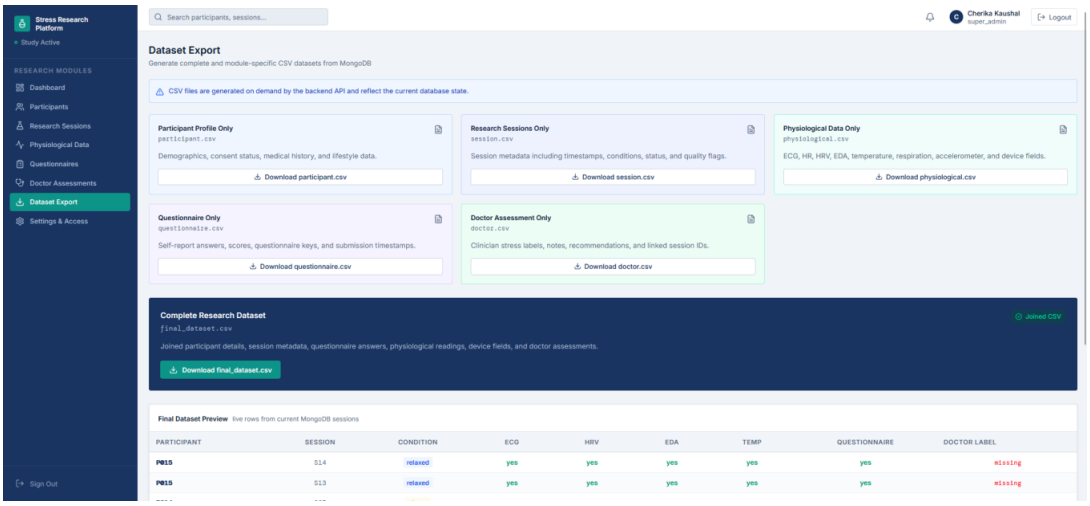

Figure 16. Dataset Export through the Dashboard.

4.6 Backend, Database, and Authentication

The backend of the Women's Health Research Platform was created using the combination of Node.js and Express.js programming technologies and utilized MongoDB as the chief database system for research data storage and participant data. To maintain modularity in the application, the API was separated into various routes for executing several functions, including authentication, participant management, surveys and symptom logging, menstrual tracking, research session management, physiological information management, and dashboard management. In this case, Mongoose schemas were applied

for data modeling; also, separate collections for users, participants, questionnaire responses, symptom logs, menstrual records, research sessions, physiological data, consent documents, and researchers' notes were created. JWT was utilized in the community to ensure authentication and role-based access to the application together with the Brevo API for emailing and communication on passwords. The backend system was transferred from FastAPI and PostgreSQL technologies to Node.JS and MongoDB.

5 Testing and Validation

5.1 Testing Summary

Table 3: System Testing Results

Test Area	**Method**	**Result**
Authentication	Manual and scripted login/signup with valid and invalid credentials	Pass
Role-based access	Attempted dashboard access with participant token	Correctly rejected
Dashboard rendering	Cross-checked dashboard values against raw database records	Values matched
API contract after migration	Compared FastAPI vs. Node.js responses for identical requests	No breaking changes
Form validation	Submitted incomplete/invalid session and symptom-log entries	Correctly rejected with structured errors

5.2 End-to-End System Testing

End-to-end testing verifies the end-to-end data flow. The wearable-data workflow is also tested to ensure that the physiological readings are correct and properly connected to the app, sent by the authenticated REST APIs without losing any data.

6 Results

Along the way, both platforms managed to complete a functional end-to-end workflow. The Women's Health Platform is used to support onboarding of participants, periodic check in and follow up. The Stress Management Platform facilitates session-based monitoring, and self- reported data for each session.The backend of the Women's Health app is successfully transitioned from FastAPI/PostgreSQL to Node.js/MongoDB, with a few

adjustments on the front end. The data shown on the researcher dashboards is also checked against the data in the database.

7 Conclusion

This project focused on the development of two digital health research applications: the Women's Health Longitudinal Research Platform and the Stress Management Research Platform. As part of the project, I contributed to building participant-facing applications, researcher dashboards, backend APIs, MongoDB databases, authentication modules, and the overall research workflow.This enables the end-to-end data flow required for the platform. Overall, this project helped me with practical experience in full-stack application development, REST API design, database management, wearable device integration, and software testing while allowing me to contribute to real-world healthcare research applications.

References

[1] Lin, G., Li, J. Y., Kalani, K., Truong, K. N., & Mariakakis, A. (2026). A longitudinal dataset of physiological, hormonal, metabolic, and self-reported menstrual health data. Scientific Data, 13(1), 411.

[2] Kilungeja, G., Graham, K., Liu, X., & Nasseri, M. (2025). Machine learning-based menstrual phase identification using wearable device data. npj Women's Health, 3 (1), 29.

[3] Keeler Bruce, L., González, D., Dasgupta, S., & Smarr, B. L. (2024). Biometrics of complete human pregnancy recorded by wearable devices. NPJ Digital Medicine, 7(1), 207.

[4] Zhou, C., Xie, Z., Wang, Q., Wang, Z., Xie, B., Yang, Y., ... & Zhu, D. (2025). Development and validation of questionnaire-based machine-learning models to predict early natural menopause: a national cross-sectional study. npj Women's Health, 3(1), 49.

[5] Kuschel, M., Vieluf, S., Reinsberger, C., Loddenkemper, T., & Hasija, T. (2026). Explainable AI Using Inherently Interpretable Components for Wearable-based Health Monitoring. arXiv preprint arXiv:2603.12880.

[6] Selvan, M. P., Kalaivani, S. D., Parida, K., Sahana, S., Lekhana, Y. S., & Mary, S. P. (2026, March). XAI Based Early Menopause Detection using Super Learner and

Discriminant Analysis. In 2026 International Conference on Data Science, Agents and Artificial Intelligence (ICDSAAI) (Vol. 1, pp. 1-6). IEEE.