BRAUDE

College of Engineering, karmiel

Capstone Project Phase B
24-2-D-21

CareHub

Development of an Application for Daily
Management and Support for Parkinson's Patients

Supervisor: Dr. Julia Sheidin
Omer Sommerstein - Omers354@gmail.com
Aviram Fishman - Aviram.Fishman@e.braude.ac.il
GitHub: https://github.com/omersbh/CareHub

Table of Contents:

1. Abstract

Parkinson's disease is a neurodegenerative disorder that harms motor abilities, causing symptoms such as tremors, stiffness, and slowness of movement. Patients experience “ON” and “OFF” states that significantly affect their ability to carry out daily activities. While there is no cure, proper management through physical exercise, nutrition, medication, and symptom tracking can improve the quality of life for patients.

The project was initiated when Michael Jackont, a Parkinson's patient, approached Braude College requesting software tailored to his needs. Hence, this project focuses on developing a user-friendly web application that enables Parkinson's patients to log essential health and lifestyle data, including symptoms, physical training, nutrition, and medication. The system ensures minimal effort for patients while allowing caregivers, such as family members and healthcare professionals, to access this information and help better support and management.

By addressing challenges like accessibility, usability, and training, this solution aims to increase the daily lives of Parkinson's patients and their caregivers, providing an extensive tool for effective disease management.

Keywords: Parkinson's Disease, Parkinson's Disease Management, User-Centered Design, UI for Parkinson's Patients, Healthcare Technology, Symptom Tracking.

2. Introduction

Parkinson's disease (PD), the second most common neurodegenerative disorder, mainly affects individuals over 60, with a higher popularity in men (3:2 ratio) [1]. By 2030, PD cases are expected to reach 9 million globally [2]. This disease significantly harms motor abilities, causing tremors, stiffness, and slowness of movement. Patients experience “ON” and “OFF” states, where the former allows manageable daily activities, and the latter causes significant limitations. While medications can extend the duration of the “ON” state, no cure currently exists. Research has shown that physical exercise [3], proper nutrition [4], social interactions, and complementary treatments positively influence the quality of life for PD patients.

PD patients often document daily events, such as medication schedules, physical activity, and dietary habits, using basic tools like Excel or Google Sheets. These records aimed to identify patterns that extended their “ON” states. However, cognitive challenges frequently interfere with real-time data logging, reducing accuracy and reliability. To address these challenges, we developed CareHub, a web-based application designed to enhance disease management and facilitate easier access.

3. Background and Related Work

PD has been recognized for centuries, with the first clear description given by James Parkinson in his 1817 essay, "An Essay on the Shaking Palsy" [5]. While there is no cure, significant progress has been made in managing the disease through medications and lifestyle adjustments. Studies show that tailoring a patient's lifestyle to their condition can help extend the "ON" state, where symptoms are better controlled [6].

3.1 Impact on Parkinson's Disease Patients

PD significantly impacts patients' lives, affecting both motor functions like trembling, muscle stiffness, and slow movement, as well as non-motor functions, including sleep disturbances, cognitive changes, and mood disorders [7]. These symptoms collectively impair daily activities, such as personal hygiene and driving, leading to social isolation, emotional distress, and financial challenges like early retirement or job loss [8]. While treatments can help manage symptoms, they also present challenges, including medication side effects and the need for frequent medical care [9]. The progressive nature of PD requires patients and their families to continually adapt to new challenges, significantly affecting their quality of life and independence over time [8].

3.2 Parkinson's Disease Management

Consistent adherence to prescribed medications is crucial for symptom control, with studies showing that patients with better medication adherence experience improved motor function and quality of life [10]. Regular exercise is another cornerstone of PD management, with research indicating that it can improve the manner of walking, balance, tumbling, flexibility, grip strength, and motor coordination [11]. Proper nutrition also plays a vital role, with experts emphasizing the importance of a balanced diet rich in fiber, omega-3 fatty acids, and antioxidants, as well as considering protein intake timing to optimize medication effectiveness [12]. Sleep management is critical, as sleep disorders are common in PD and can significantly impact daily functioning. Implementing practices before sleeping, such as stretching exercises, yoga, and mindfulness exercises, can improve the overall quality of life and sleep [13]. Stress management is equally important, as stress can make PD symptoms worse. Mindfulness-based interventions have shown promise in reducing stress and improving the quality of life in PD patients [14]. This multifaceted approach to daily management requires active involvement from patients, caregivers, and healthcare providers. However, by focusing on these key areas, individuals with PD can optimize their symptom control and maintain a higher quality of life.

3.3 Supporting the Need for Software Solutions

Historically, before the technological revolution, managing disease relied heavily on manual methods. This caused challenges, particularly for patients with cognitive impairments, as PD can affect memory and executive function [15]. These difficulties often led to more frequent "OFF" periods, where symptoms are less controlled. The arrival of computers and smartphones has opened a new path for assisting PD patients. With most of the population now owning smartphones, these devices offer promising tools for disease management. [16] highlighted the potential of technology in PD care, noting that smartphones can be used for symptom monitoring, medication reminders, and even to collect data for clinical research. Smartphone applications specifically designed for PD patients have proliferated in recent years.

3.4 Summary

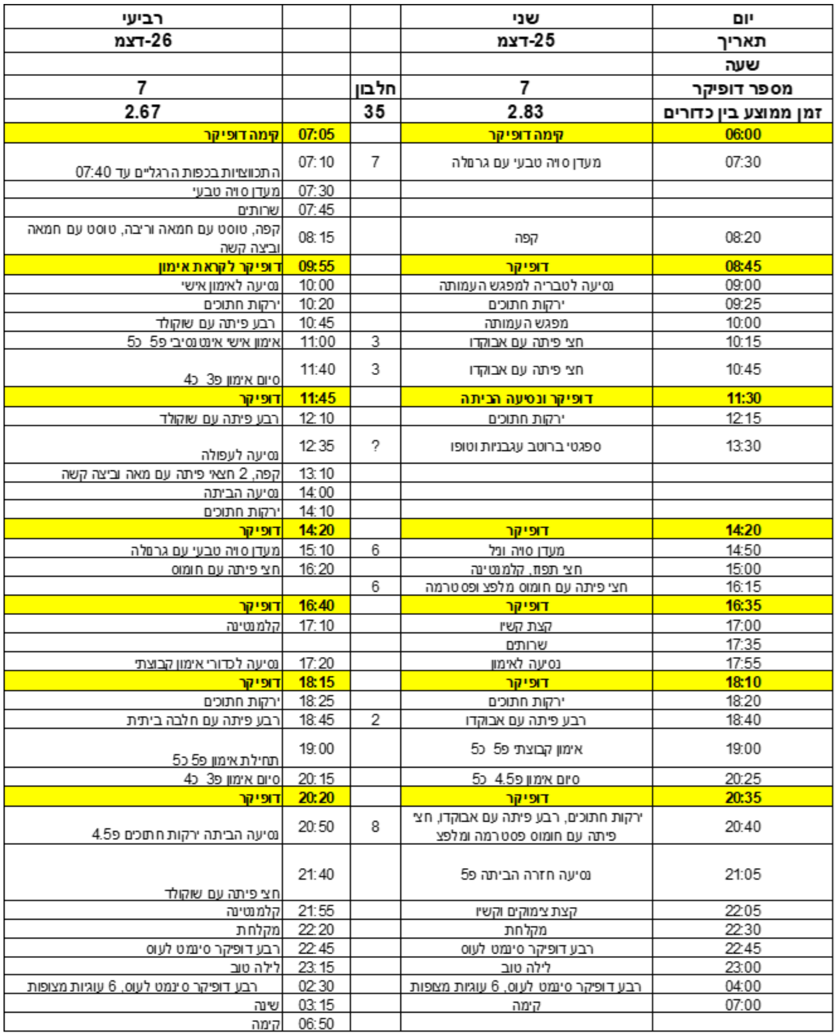
Applications should allow users to customize various settings to their individual needs and preferences, such as adjustments for text size, voice output settings, and visual theme options. Applications should incorporate accessibility features, such as suggested options or drop-downs. They should feature interactive elements, schedule reminders, and notifications to engage with the user. Software should integrate tools for measuring and recording user progress, such as detailed progress logs or achievement markers. Applications should have a straightforward and easily comprehensible interface that PD patients can use easily. Additionally, the software should operate in the background with minimal user input required, benefiting individuals with PD.

4. Solution Description

As part of our research, we reviewed applications that aid in managing PD. This review was carried out in collaboration with our client Michael Jackont, 57 years old, a citizen of Israel - dealing with PD. Michael joined us through Braude's flagship project, which collaborates closely with the engineering departments at Braude College and various external organizations to provide technological solutions to those in need.

Michael Jackont currently manages his condition using daily schedule entries in Google's TASK. His schedule includes records for medication intake, meals, exercise, physical condition before training, and bathroom visits. Michael adheres to predetermined times for these activities, which he believes significantly contributes to maintaining the “ON” state. Research indicates that following a predetermined schedule improves the quality of life for patients [17]. As each patient's schedule and actions are personalized and vary between individuals, it is crucial to manage the day optimally while considering PD impact on cognition and memory capacity.

Michael reports that maintaining schedule records creates numerous challenges and demands considerable time and resources from him (see Figure 1).



רביעי			שני	יום
26-דצמ			25-דצמ	תאריך
				שעה
7		חלבון	7	מספר דופיקר
2.67		35	2.83	זמן ממוצע בין כדורים
קימה דופיקר	07:05		קימה דופיקר	06:00
התכווציות בכפות הרגליים עד 07:40	07:10	7	מעדן סויה טבעי עם גרנולה	07:30
מעדן סויה טבעי	07:30			
שרותים	07:45			
קפה, טוסט עם חמאה וריבה, טוסט עם חמאה וביצה קשה	08:15		קפה	08:20
דופיקר לקראת אימון	09:55		דופיקר	08:45
נסיעה לאימון אישי	10:00		נסיעה לטבריה למפגש העמותה	09:00
ירקות חתוכים	10:20		ירקות חתוכים	09:25
רבע פיתה עם שוקולד	10:45		מפגש העמותה	10:00
אימון אישי אינטנסיבי פ5 כ5	11:00	3	חצי פיתה עם אבוקדו	10:15
סיום אימון פ3 כ4	11:40	3	חצי פיתה עם אבוקדו	10:45
דופיקר	11:45		דופיקר ונסיעה הביתה	11:30
רבע פיתה עם שוקולד	12:10		ירקות חתוכים	12:15
נסיעה לעפולה	12:35	?	ספגטי ברוטב עגבניות וטופו	13:30
קפה, 2 חצאי פיתה עם מאה וביצה קשה	13:10			
נסיעה הביתה	14:00			
ירקות חתוכים	14:10			
דופיקר	14:20		דופיקר	14:20
מעדן סויה טבעי עם גרנולה	15:10	6	מעדן סויה ופל	14:50
חצי פיתה עם חומוס	16:20		חצי תפוח, קלמנטינה	15:00
		6	חצי פיתה עם חומוס מלפפ ופסטרמה	16:15
דופיקר	16:40		דופיקר	16:35
קלמנטינה	17:10		קצת קשיו	17:00
			שרותים	17:35
נסיעה לכדורי אימון קבוצתי	17:20		נסיעה לאימון	17:55
דופיקר	18:15		דופיקר	18:10
ירקות חתוכים	18:25		ירקות חתוכים	18:20
רבע פיתה עם חלבה ביתית	18:45	2	רבע פיתה עם אבוקדו	18:40
תחילת אימון פ5 כ5	19:00		אימון קבוצתי פ5 כ5	19:00
סיום אימון פ3 כ4	20:15		סיום אימון פ4.5 כ5	20:25
דופיקר	20:20		דופיקר	20:35
נסיעה הביתה ירקות חתוכים פ4.5	20:50	8	ירקות חתוכים, רבע פיתה עם אבוקדו, חצי פיתה עם חומוס פסטרמה ומלפצ	20:40
חצי פיתה עם שוקולד	21:40		נסיעה חזרה הביתה פ5	21:05
קלמנטינה	21:55		קצת צימוקים וקשיו	22:05
מקלחת	22:20		מקלחת	22:30
רבע דופיקר סינמט לעוס	22:45		רבע דופיקר סינמט לעוס	22:45
לילה טוב	23:15		לילה טוב	23:00
רבע דופיקר סינמט לעוס, 6 עוגיות מצופות	02:30		רבע דופיקר סינמט לעוס, 6 עוגיות מצפות	04:00
שינה	03:15		קימה	07:00
קימה	06:50			

Figure 1: Management of PD - Michael Jackont

Michael has experimented with various applications, both self-discovered and recommended (such as MyMovesMatter application), but found them unhelpful to the extent that he preferred to continue managing his schedule manually. In his view, these apps required him to "work" for them by entering data rather than serving as a user-friendly personal assistant that allowed him to gain insights about the information.

Building on the insights gained in our collaboration with Michael Jackont we tailored the application to address his specific challenges and needs. We began with a detailed meeting where Michael shared his daily routine, demonstrated his use of Google Tasks, and presented a simple prototype of his envisioned app screens. These inputs were instrumental in shaping the application's design and functionality.

To further refine the solution, we engaged with key individuals in Michael's support network. These discussions provided additional perspectives on the challenges faced by both Michael and his caregivers, ensuring that the application effectively addresses his unique circumstances. The summary of the interviews and observations was presented in the Phase A report.

The application, currently developed exclusively for Michael, focuses on features that help him log and manage his routines with minimal effort while enabling his caregivers to access this data for better support and coordination. These include:

- Tool for recording daily symptoms, intake medication, nutrition and sports training to identify patterns and improve treatment plans for the PD patient.
- A secure interface that allows caregivers to view and coordinate the PD patient schedule in real-time. This feature will be implemented in the future.
- Visualizations to help the PD patient and his caregivers monitor progress and adapt routines.

Using a user-centered design (UCD) approach, we prioritized real-world needs and experiences, ensuring the app is practical, effective, and user-friendly. This foundation aims to deliver a solution that significantly enhances Michael's daily life and care coordination.

The main requirements for our system:

Functional Requirements:

1. The system allows PD patients to enter medicine intake data.
2. The system allows PD patients to log physical activities.
3. The system allows PD patients to record food intake.
4. The system allows PD patients to capture and upload food pictures.
5. The system allows PD patients to log their symptoms.
6. The system allows PD patients to rate their overall feeling.
7. The system allows new users to sign up and create an account.
8. The system allows registered users to login.
9. The system allows users to set their roles (PD Patient or Caregiver).
10. The system allows displaying patient data in table format.
11. The system allows users to edit or correct previously entered data.

Non-Functional Requirements:

1. The system should be designed with an intuitive interface suitable for users with motor

disabilities.

2. The system should minimize required scrolling across screens.
3. The system should employ a user-centric approach, integrating seamlessly into the user's daily routine.
4. The system should require minimal effort to operate.
5. The system should present data in a way that helps PD patients better organize their day and increase "ON" states.
6. The system should be available for use most of the time, excluding scheduled maintenance.
7. The system should adjust to relevant healthcare data protection regulations to protect user data.
8. The system should be designed for easy maintenance and cost-effective operations.
9. The system should be designed to allow the future addition of new data entry methods (photo logging).
10. The system should be easy to maintain and cost-effective.

After defining the system's requirements, we revisited our main client, Michael Jackont, to ensure that our design aligned with his specific needs. Building on the analysis of existing market solutions conducted during Phase A, where the prototype was initially designed, we planned the system's functionality and identified the need for two primary modules: Input and Output.

The Input module enables the user to log essential health and lifestyle data, including medications, nutrition, symptoms, training sessions, and the user’s overall feeling (PD state). The Output module presents this logged data in a structured table format, allowing users and caregivers to review and analyze the information effectively.

5. Development Process

5.1 Application Features and User Roles

The CareHub web application offers an intuitive and easy-to-use platform for logging medications, nutrition, physical activity, symptoms, training sessions, and overall feelings, helping our client to identify trends over time. Accessible through any modern web browser on desktops, tablets, or mobile devices, CareHub eliminates the need for specialized hardware or software. Its user-friendly interface ensures that individuals like Michael, even with minimal technical experience, can navigate and use its features effectively.

The main activities supported by the system are illustrated in the Use-Case diagram (see Figure 2).

The system defines two main user roles:

1. PD Patient
2. General User

The PD Patient can log critical health and routine data, including:

- Nutrition entries
- Training sessions
- Medications
- Symptoms
- Overall feelings (rated in each log)

In the system, the PD Patient role inherits from the General User role, meaning PD patients can also include caregivers or family members. Caregivers, acting as General Users, can view the data logged by the PD patient, fostering a collaborative and supportive care environment.

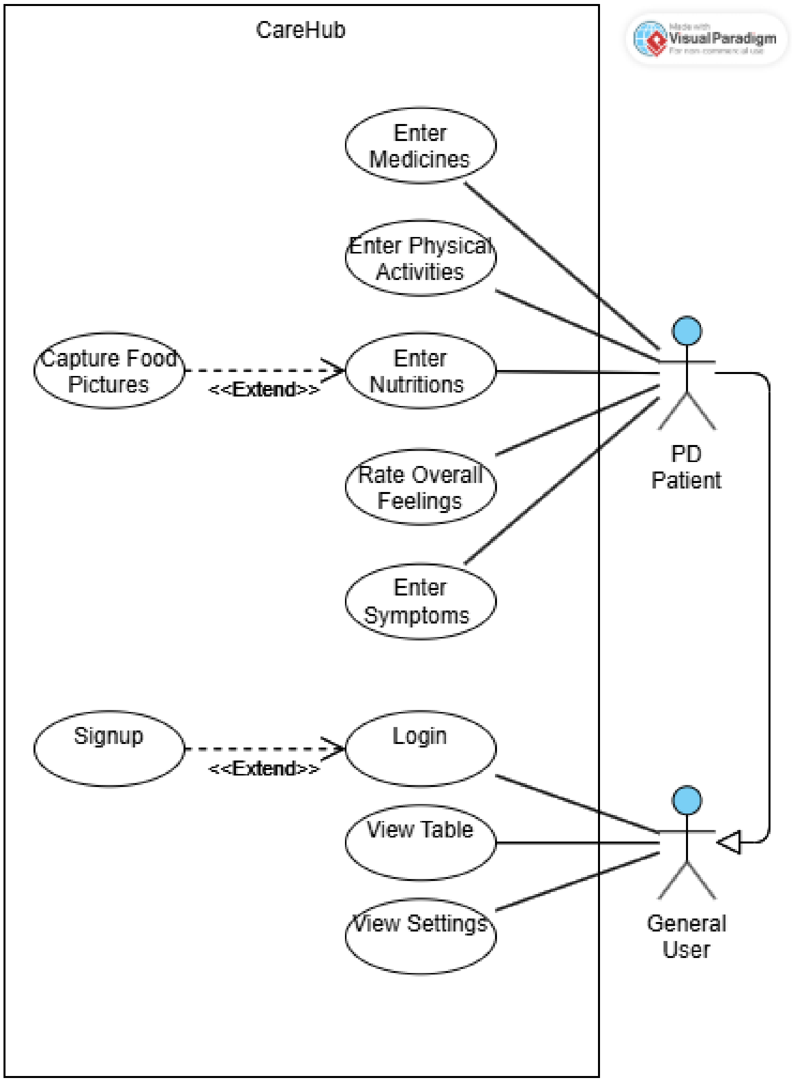


Figure 2: Use-Case Diagram

5.2 Initial Prototype

As part of our development process, we created a prototype of the application's user interface using Figma, a web-based design tool. Figma allowed us to collaborate effectively as a team, enabling real-time editing and feedback to ensure the design met the needs of both PD patients and their caregivers. We focused on creating an intuitive and user-friendly interface that simplifies data entry and retrieval for essential daily activities, such as logging physical training, medications, food intake, and symptoms.

The prototyping phase was instrumental in visualizing the app's functionality before implementation. It helped us refine features, adjust workflows, and ensure accessibility across various devices. By simulating user interactions with the prototype, we identified potential usability challenges early and optimized the design for simplicity and efficiency. This iterative approach ensured that our final product would align with the goals of enhancing the daily management experience for PD patients while providing seamless data sharing with caregivers and healthcare providers.

5.3 Code Implementation

During the code implementation phase, we developed the CareHub web application using the MERN stack, which includes React for the frontend, Node.js and Express.js for the backend, and MongoDB for the database (see Figure 3).

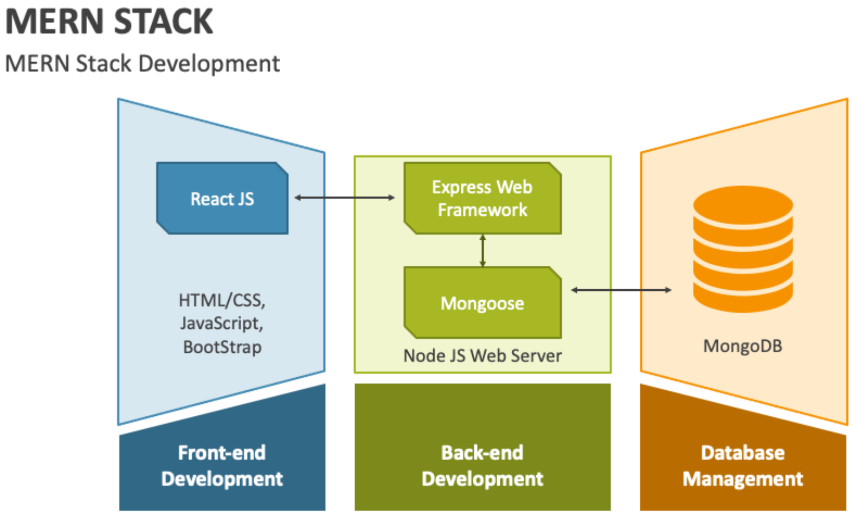


Figure 3: Application Architecture.

The frontend was designed with React, utilizing .jsx and .css files to create dynamic and user-friendly interfaces. This allowed us to implement responsive components tailored to the specific needs of PD patients, ensuring the app is accessible on multiple devices, including desktops, tablets, and smartphones,

On the backend, we employed Node.js and Express.js to build a strong API that manages data flow between the application and the database. This architecture enabled secure and efficient handling of user data, including medication, nutrition, training logs, and symptoms.

For the database, we chose MongoDB, a NoSQL database, to store and manage the application's data. Its flexibility allowed us to structure the data dynamically to accommodate the unique requirements of each user entry.

The MERN stack's seamless integration allowed us to deliver a responsive, reliable, and accessible web application designed specifically for Michael's needs.

Frontend Development

The frontend was built using React We implemented:

- **Responsive Design:** Ensured the app is accessible on multiple devices, including desktops, tablets, and smartphones, using CSS-in-JS libraries and media queries.
- **Material-UI Integration:** Incorporated Material-UI components for consistent design, accessibility, and usability across the application.
- **State Management:** Utilized React's Context API and hooks to manage global states, ensuring smooth inter-component communication and user interaction.

Backend Development

The backend was developed using Node.js and Express.js, focusing on creating a robust API layer for seamless communication between the frontend and the database:

- **RESTful API Endpoints:** Built endpoints for CRUD (Create, Read, Update, Delete) operations on data like medication schedules, training logs, and symptom tracking.
- **Authentication and Security:** Implemented JWT (JSON Web Tokens) for secure user authentication and role-based access control, protecting sensitive data.
- **Middleware:** Used Express.js middleware for logging, error handling, and request validation to enhance the application's reliability and maintainability.

Database Architecture

The application's data is stored in **MongoDB**, a flexible and scalable NoSQL database:

- **Schema Design:** Created schemas using Mongoose to model user data, intake medication, nutrition, training, symptoms and caregiver access.
- **Dynamic Data Handling:** The database structure accommodates diverse user entries, allowing PD patients and caregivers to log and retrieve data effortlessly.
- **Data Security:** Ensured encryption of sensitive data fields, such as passwords, in the database to maintain user privacy.

5.4 Client Feedback and Final Refinement

To ensure the application met its intended goals, we scheduled a client meeting with Michael Jackont, our primary user, to review the developed features and gather his feedback. This meeting provided an opportunity to validate the app's functionality and usability, aligning it closely with Michael's needs. Additionally, we shared the app with key caregivers and healthcare professionals in Michael's support network to collect further insights.

Following the client meeting and feedback sessions, we entered the refinement stage. We made necessary adjustments to improve the app’s usability and functionality using the input gathered. This iterative process addressed usability concerns, enhanced key features, and ensured the system delivered a seamless and user-friendly experience.

To formally evaluate the usability of the application, Michael was provided with a System Usability Scale (SUS) form. The results (we got score of 80/100) were overwhelmingly positive, indicating satisfaction with the app’s design, ease of use, and its potential to simplify routine management. Written feedback from caregivers and healthcare providers also highlighted the app's effectiveness in improving daily symptom tracking and facilitating better communication between patients and caregivers.

These feedback-driven refinements and usability evaluations ensured that the final version of CareHub is a reliable, accessible, and impactful tool tailored to the needs of Parkinson’s patients and their support teams.

6. Faced Challenges and Solutions

6.1 Planning and Functional Challenges

1. Choosing the Platform
Challenge:
Initially, we considered developing a native app for iOS and Android. However, we realized that maintaining long-term support, updates, and compatibility for two separate platforms would be resource-intensive and impractical for our team. Additionally, the risk of performance issues or limitations on certain devices leads to further concerns.

Solution:
We chose a web app using the MERN stack, which allows us to offer cross-platform compatibility with minimal maintenance. This decision ensured accessibility across devices (PCs, tablets, and smartphones) without the need to develop separate applications.

2. Data Management and Security

Challenge:
Ensuring the secure storage and retrieval of sensitive patient data was a major concern. We had to comply with data privacy standards while avoiding overly complex solutions that could slow development.

Solution:
We implemented encrypted data information. MongoDB was used for scalable and secure data storage, with encrypted data transmission between the front-end and back-end using HTTPS.

3. Accessibility

Challenge:
We aimed to create an app tailored to Michael's specific needs while ensuring it could function seamlessly without requiring advanced hardware or additional software installations. Since the app is currently designed exclusively for Michael, accessibility and usability were key priorities.

Solution:
To meet Michael's needs, we chose web technologies that allow the app to run on any modern browser without requiring specialized hardware. By relying on free and open-source tools, including React for the front end, Node.js and Express.js for the back end, and MongoDB for the database, we ensured that the app could be developed without incurring any costs.

4. Team Skills and Learning Curve

Challenge:
We had limited experience with some technologies in the MERN stack, which initially slowed progress.

Solution:
We conducted focused learning sessions and utilized online resources to build our skills. This collaborative effort enabled us to overcome the learning curve and effectively implement the required features.

5. Testing and Iteration

Challenge:

Ensuring the app functioned well across various devices and scenarios required thorough testing, which was time-consuming and revealed unexpected bugs.

Solution:

We adopted an iterative development approach, continuously testing the app during each sprint. Feedback from Michael before finishing the development allowed us to identify and fix issues early enough.

6. Workflow Planning

Challenge:

Scheduling meetings was difficult due to conflicting availability within our team, which led to relying on individual work and asynchronous collaboration. This occasionally caused inconsistency in code implementation and challenges during integration.

Solution:

We addressed this by improving communication through regular check-ins, using asynchronous tools, and maintaining clear documentation to ensure smooth collaboration and efficient workflow.

6.2 Design and Aesthetics Challenges

1. Balancing Simplicity and Attractiveness

Challenge:

Ensuring the app looks modern and visually appealing without compromising on simplicity is crucial for users with cognitive challenges.

Solution:

We adopted a minimalist design approach focusing on functionality, avoiding unnecessary visual elements while maintaining a clean and modern aesthetic.

2. Color Choices and Visual Clarity

Challenge:

Choosing colors that were both aesthetically pleasing and accessible for users with visual impairments or difficulty distinguishing colors presented a unique design challenge. We needed a palette that adhered to accessibility guidelines while maintaining a clean and professional appearance.

Solution:
To ensure optimal readability and accessibility, we selected WCAG-compliant, high contrast colors. For this, we utilized ColorBrewer [19], a tool specifically designed to create visually distinct and accessible color schemes. We chose the "Paired" scheme with five qualitative colors, ensuring the design was not only visually appealing but also suitable for users with color vision deficiencies. This careful selection process enhanced the app's usability for all users, particularly those with Parkinson's-related visual challenges.
The chosen colors:
#a6cee3 - 🟦 #1f78b4 - 🟦 #b2df8a - 🟩 #33a02c - 🟩 #fb9a99 - 🟥

3. User-Centric Design for PD Patients
Challenge:
Designing an interface for PD patients required addressing motor challenges, such as tremors, and cognitive difficulties. Complex navigation or small interface elements could make usability way harder.

Solution:
We employed user-centered design principles and conducted feedback sessions with caregivers and PD patients. As a result, we implemented large buttons, simple layouts, and minimalistic navigation to ensure ease of use.

7. Results and Conclusions

7.1 Goals and Achievements

The primary goal of our project was to create a web-based application that would assist PD patients in managing various aspects of their daily routine. This included logging key information such as medicines, nutrition, symptoms, and physical activities or training sessions. We also aimed to provide caregivers with an easy way to access this data, giving them a clearer view of the patient's health and progress.

One of the main achievements was the successful development of a fully functional web app that allows PD patients to track their daily activities. By focusing on ease of use and accessibility, the app enables users to enter and monitor crucial data without unnecessary complexity. This was essential since many patients struggle with cognitive and motor challenges, which is why we prioritized a simple, intuitive interface.

Additionally, the app displays the entered data in an organized table format, making it easier for both patients and caregivers to review and understand the patient's daily habits. This feature received positive feedback, as it helped users track correlations between medication, nutrition, and physical activity.

Another significant achievement was the implementation of secure data storage and authentication measures. As the app handles sensitive health data, ensuring that patient information is protected is a top priority. We integrated secure login mechanisms and encrypted data storage, paying attention to privacy standards to ensure that all personal information was kept safe.

7.2 Conclusions

Looking back at the development process, we can conclude that the decision to build a web app, rather than a mobile app, was the right one given our project's constraints. Developing for both iOS and Android would have required more resources, including time for maintenance, updates, and testing across different devices. Going for a web app ensured cross-platform compatibility without the additional challenges of supporting multiple operating systems.

Using the MERN stack (MongoDB, Express.js, React, Node.js) provided a solid foundation for the app's development. The MERN stack allowed for the seamless integration of the front-end and back-end, offering a smooth user experience and scalable architecture. Despite the challenges we encountered, such as coordinating schedules for team collaboration and addressing design considerations for users with varying abilities, the app met our core goals and proved to be functional and user-friendly.

The feedback from Michael, our client, confirmed the app's success. He found the design intuitive and the features useful in managing daily tasks. This positive response emphasizes the importance of user-centered design in ensuring the app’s usability. Through ongoing testing, we were able to continuously refine the app’s interface and features to better suit Michael's needs.

While the core features of the app are functional and serve Michael's needs, there are areas where it could still be improved. One potential enhancement is the addition of more personalization options, such as tailored recommendations based on the user's data, which could help patients manage their conditions more effectively. Furthermore, expanding the app’s features to integrate with wearable health devices or adding real-time symptom tracking could provide more comprehensive support for patients, offering a more holistic view of their health. Additionally, the app currently lacks an option to add new symptoms, which could limit its ability to track the full range of experiences that PD patients might face. This is an area for future development to ensure that the app can adapt to the dynamic nature of Parkinson's disease.

7.3 Future Work

Looking ahead, there are several ways the app can be improved and expanded, which future developers or students might continue working on. Although we won’t continue developing the app, we hope that future students or developers will take these ideas and keep improving it. By adding new features and listening to user feedback, the app can continue to help people with PD and their caregivers for a long time.

One of the first areas to improve is personalization. The app now allows users to log their daily activities and see trends over time, but it could go further by offering personalized feedback. For example, based on the data entered—such as medications, diet, or physical activity—the app could suggest changes to help manage symptoms. This would make the app even more useful, helping patients make better choices for their health.

Another important area is integrating with wearable devices. Many patients already use devices like fitness trackers or smartwatches to monitor their activity, sleep, or heart rate. Adding this data to the app would give a fuller picture of the patient’s health. The data from these devices could automatically update in the app, saving users from entering it manually. Additionally, wearables could provide real-time alerts if something unusual is detected, like a drop-in activity or irregular heart rate, allowing for quicker action.

Another area that can be improved is the app accessibility. The current version has features to help patients with motor and cognitive challenges, but there’s room to do more. For example, adding voice commands and the ability to adjust font sizes could make it easier for all users to navigate. Also, adding more customized voices—like choosing between male or female voices—would make the app feel more personal to different users.

Expanding the target audience is another goal. While the app is designed for PD patients, it could also be useful for caregivers or healthcare professionals. In the future, developers could add different versions of the app for these groups, so it would be more useful for everyone involved in the patient’s care. This could include special features for caregivers, like tracking multiple patients, or for healthcare professionals, like providing treatment suggestions.

The app also currently only supports English, but expanding to other languages could make it accessible to more people around the world. This would help even more patients and caregivers benefit from the app’s features.

Finally, more user testing will be important in the future. By testing the app with a wider group of patients and caregivers, we can get valuable feedback that will help improve the app. This ongoing testing and feedback loop will help make sure the app meets the needs of all users.

8. User Documentation

8.1 User Guide

8.1.1 General Description

The CareHub web application assists PD patients and their caregivers in managing daily routines and tracking important health-related data. The app provides a simple and user-friendly platform for logging and monitoring information such as medications, nutrition, physical activity, symptoms, and training sessions. Additionally, users can record their overall feelings, allowing them to observe trends and identify patterns over time.

CareHub is accessible through any modern web browser on a desktop, tablet, or mobile device, eliminating the need for specialized hardware or software installation. The app's intuitive design ensures that users with minimal technical experience can navigate and use its features effectively.

The platform was designed for two types of users:

1. Patients – Patients can log their daily routines, monitor trends using visual graphs, and manage their health data in one centralized location.

2. Caregivers – Caregivers like family members, trainers, or therapists can view the patient’s data to understand their condition better and provide informed support.

The platform is currently available for patients, with caregiver functionality planned for future.

CareHub emphasizes accessibility and simplicity, ensuring users can easily log data with minimal effort. The interface is streamlined to reduce the time and energy needed for input, making it suitable for PD who may have motor or cognitive challenges. The app also includes secure login functionality to protect sensitive health information.

In summary, CareHub is an extensive tool for PD and their caregivers, enabling better communication, improved daily management, and insights into the patient’s well-being over time. This user guide explains how to use the app's features effectively, ensuring that users can maximize its benefits.

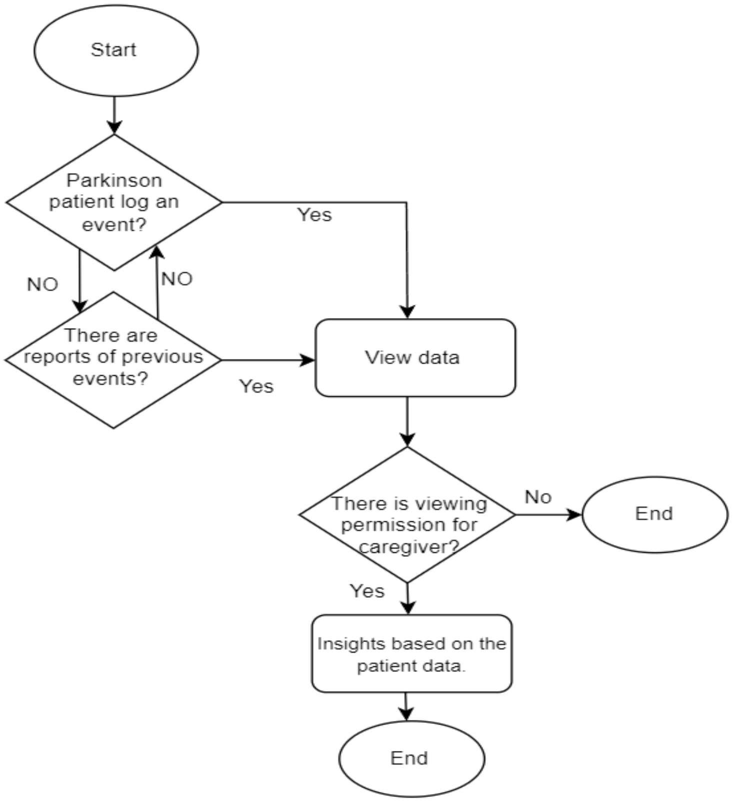

Start
Parkinson patient log an event?
Yes
NO
NO
There are reports of previous events?
Yes
View data
There is viewing permission for caregiver?
No
End
Yes
Insights based on the patient data.
End

Figure 4: Flow Chart of actions in CareHub system.

The flowchart illustrates the process for managing and analyzing data logged by PD in the app. It begins with the patient logging an event, such as symptoms or daily activities. The system allows the patient or authorized caregiver to view the data if an event is logged. If no new event is logged, the system checks if there are previous reports available. If past reports exist, these can also be viewed. The flow then checks if the caregiver has viewing permissions; if granted, the caregiver can access insights derived from the patient's data. The process then concludes, ensuring that patients and caregivers have timely and secure access to vital information supporting ongoing care and monitoring (see Figure 6).

8.1.2 Operating Instructions

CareHub is a web application accessible from any device with an internet connection and a modern web browser. Follow the steps below to start using the app and explore its features:

Accessing the Application:

Open a web browser on your device and navigate to the CareHub website by entering the app's URL in the address bar. Ensure you have a stable internet connection for smooth operation.

Important Notes:

- Ensure your browser is up to date to avoid compatibility issues.
- For security, never share your login details with anyone.

By following these instructions, users can easily access and operate CareHub to manage daily health routines and gain valuable insights into their well-being.

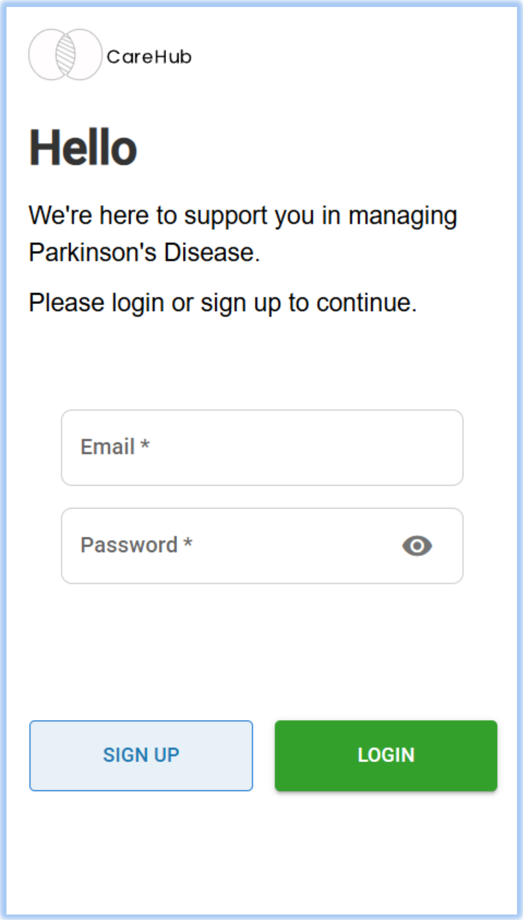
Let's examine each screen of the CareHub application and understand their functionality starting from the next page.

1. Login/Signup Screen:

- **New Users:**
    1. Click the "SIGN UP" button
    2. Enter your email address in the "Email" field
    3. Create a secure password in the "Password" field
    4. Click "LOGIN" to create your account
- **Existing Users:**
    1. Enter your registered email address
    2. Input your password (click the eye icon to view/hide password)
    3. Click "LOGIN" to access your account

Important Notes:

- Fields marked with * are mandatory
- Ensure your password meets security requirements
- Use a unique password not used on other websites



Hello

We're here to support you in managing Parkinson's Disease.

Please login or sign up to continue.

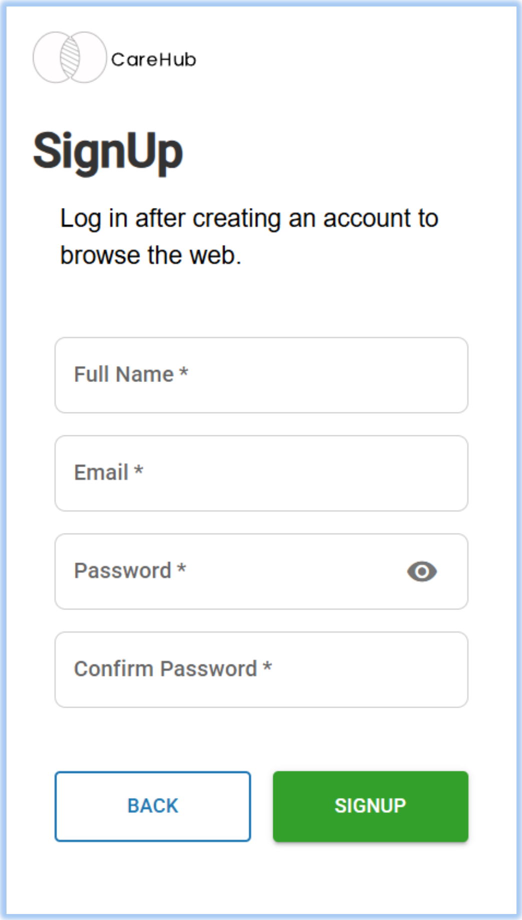
2. Account Creation Screen:

1. Enter your full name
2. Provide a valid email address
3. Create a password (4-12 characters)
4. Re-enter password to confirm
5. Click "SIGNUP" to create account

- Clicking "BACK" cancels account creation
- After signup, log in with your new credentials on the login screen

Requirements:

- Valid email format required (example@domain.com)
- Password: 4-12 characters
- All fields marked * are mandatory



CareHub

SignUp

Log in after creating an account to browse the web.

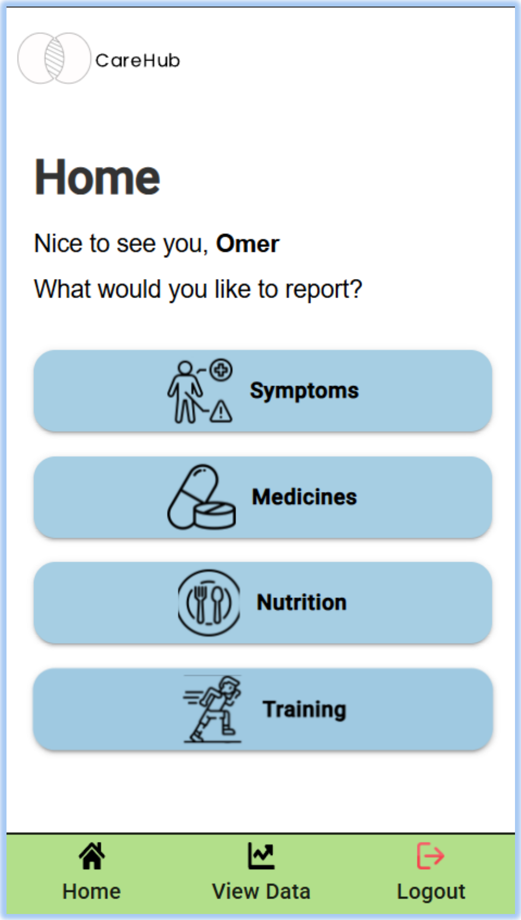
3. Home Screen:

Navigation Options:

1. Symptoms - Report and track symptoms
2. Medicines - Manage medication intake
3. Nutrition - Log food and drink consumption
4. Training - Access exercise routines

Bottom Navigation Bar:

- Home - Return to main menu
- View Data - Access recorded information
- Logout - End session securely



Nice to see you, **Omer**

What would you like to report?

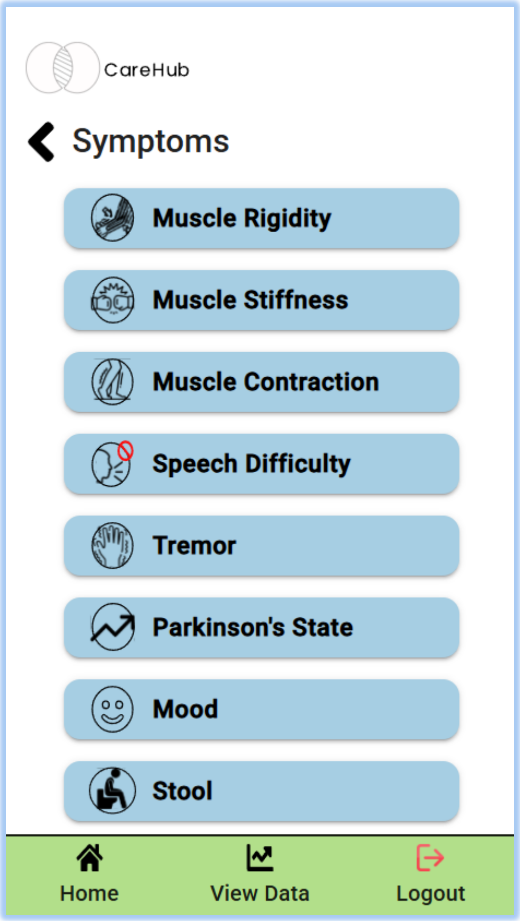
4. Symptoms Screen:

Select from available symptoms to report:

- Muscle Rigidity
- Muscle Stiffness
- Muscle Contraction
- Speech Difficulty
- Tremor
- Parkinson's State
- Mood
- Stool
- General Feeling

Navigation:

- Back arrow (⟵) returns to Home
- Bottom bar: Home, View Data, Logout

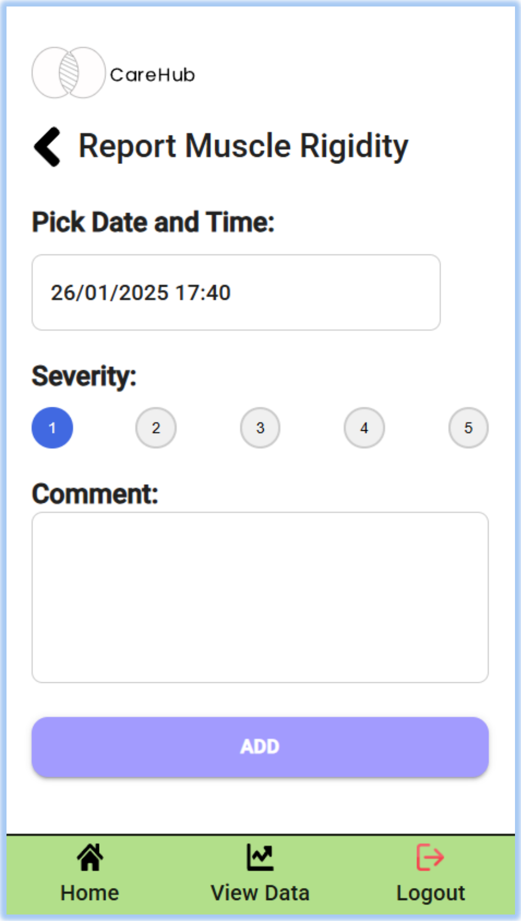


5. Report General Symptom Screen:

1. Select date/time or use current timestamp
2. Rate severity from 1 (mild) to 5 (severe)
3. Add optional comments about symptoms
4. Click "ADD" to save report

Navigation:

- Back arrow (⟵) returns to Symptoms screen
- Bottom bar: Home, View Data, Logout



Report Muscle Rigidity

Pick Date and Time: 26/01/2025 17:40

Severity: ● 1 ○ 2 ○ 3 ○ 4 ○ 5

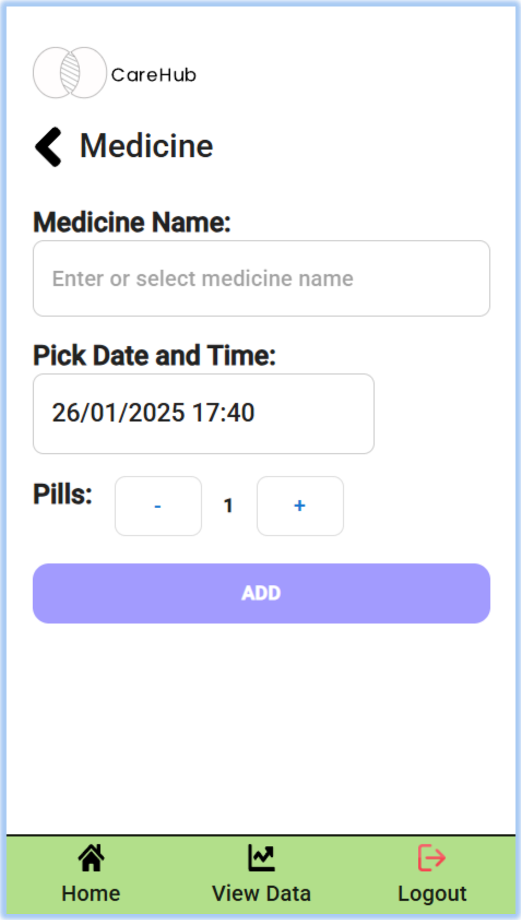
Comment:

6. Medicine Screen:

1. Enter or select medicine name
2. Set date/time or use current timestamp
3. Adjust pill quantity using + and - buttons
4. Click "ADD" to log medication

Navigation:

- Back arrow (←) returns to Home
- Bottom bar: Home, View Data, Logout



Medicine Name: Enter or select medicine name

Pick Date and Time: 26/01/2025 17:40

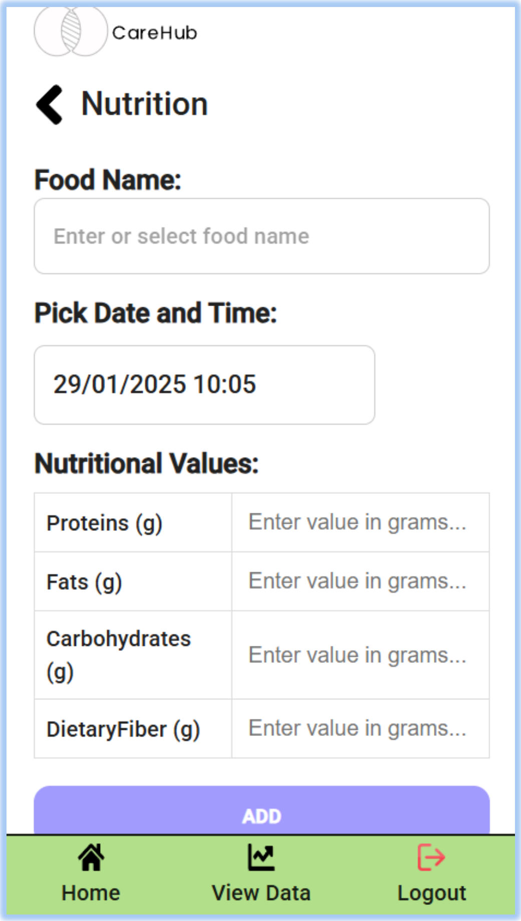
Pills: - 1 +

7. Nutrition Screen:

1. Enter/select food name
2. Set date/time or use current timestamp
3. Input nutritional values:
   - Proteins
   - Fats
   - Carbohydrates
   - Dietary Fiber
4. Click "ADD" to log meal

Navigation:

- Back arrow (⟵) returns to Home
- Bottom bar: Home, View Data, Logout



Food Name:

Enter or select food name

Pick Date and Time:

29/01/2025 10:05

Nutritional Values:

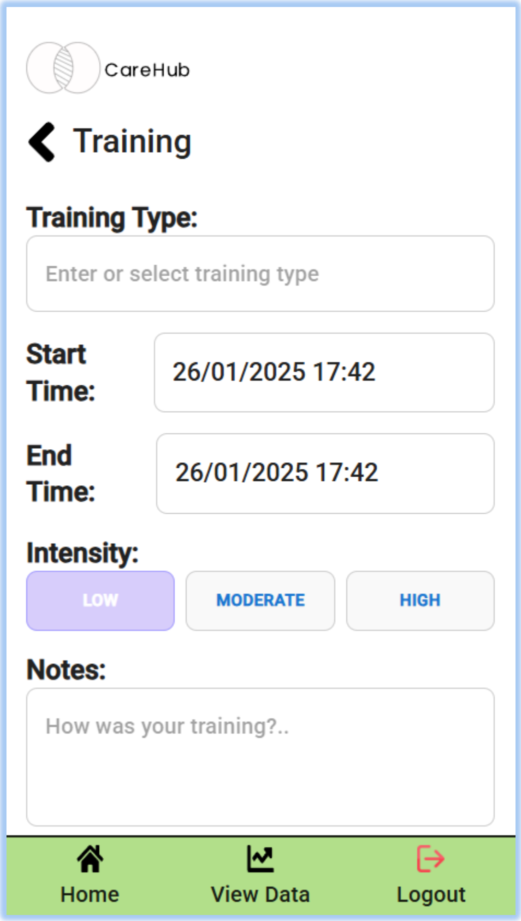
Proteins (g)	Enter value in grams...
Fats (g)	Enter value in grams...
Carbohydrates (g)	Enter value in grams...
DietaryFiber (g)	Enter value in grams...

8. Training Screen:

1. Enter/select training type
2. Set start and end times
3. Select intensity level:
   - LOW (purple)
   - MODERATE (white)
   - HIGH (white)
4. Add optional notes
5. Click "ADD" to log exercise

Navigation:

- Back arrow (⟵) returns to Home
- Bottom bar: Home, View Data, Logout



CareHub

Training

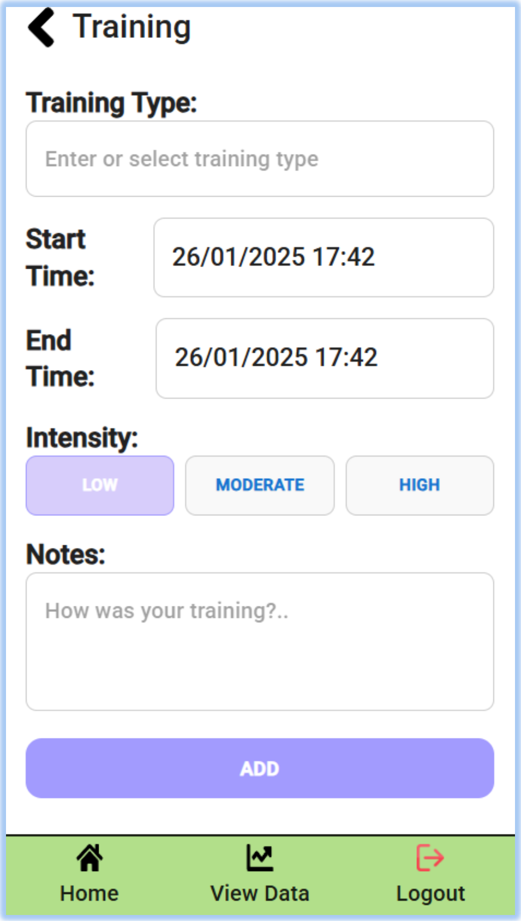
Training Type: Enter or select training type

Start Time: 26/01/2025 17:42

End Time: 26/01/2025 17:42

Intensity:

Notes: How was your training?..



Training

Training Type: Enter or select training type

Start Time: 26/01/2025 17:42

End Time: 26/01/2025 17:42

Intensity:

Notes: How was your training?..

9. View Data Screen:

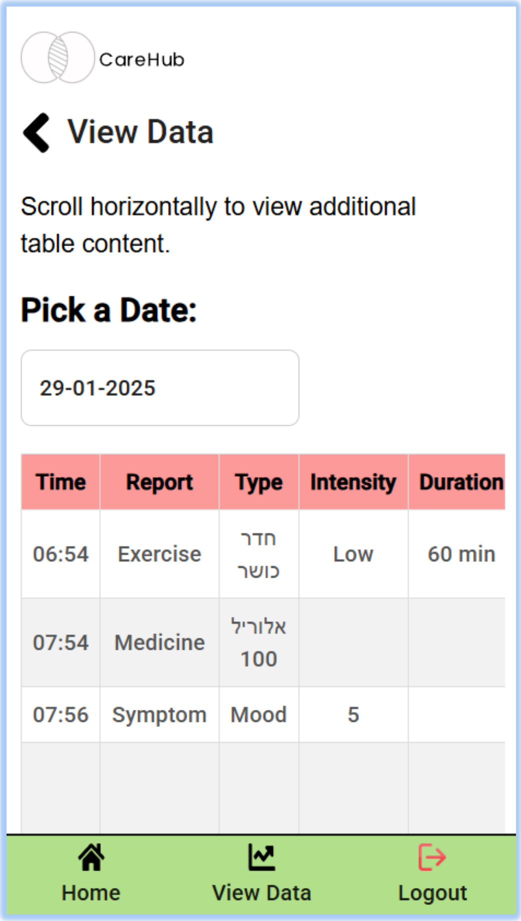
1. Select date from the calendar
2. View daily logs in chronological order:
   - Time
   - Report type (Exercise, Food, Medicine, Symptom)
   - Type/Name
   - Intensity/Duration
   - Notes

Features:

- Scroll horizontally to see all columns
- Data organized by time for easy tracking

Navigation:

- Back arrow (⟵) returns to the previous page we clicked "View Data" from.
- Bottom bar: Home, View Data, Logout



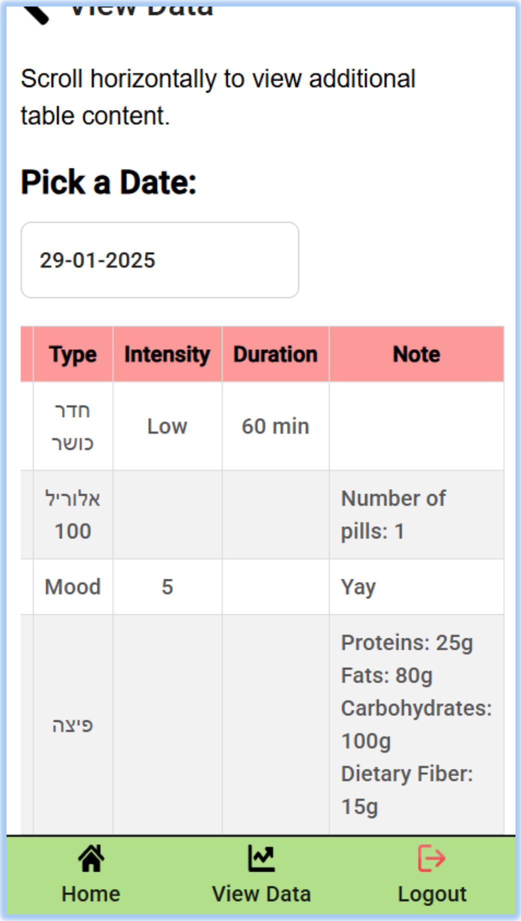
View Data

Scroll horizontally to view additional table content.

Pick a Date:

29-01-2025

Time	Report	Type	Intensity	Duration
06:54	Exercise	חדר כושר	Low	60 min
07:54	Medicine	אלוריל 100		
07:56	Symptom	Mood	5	



View Data

Scroll horizontally to view additional table content.

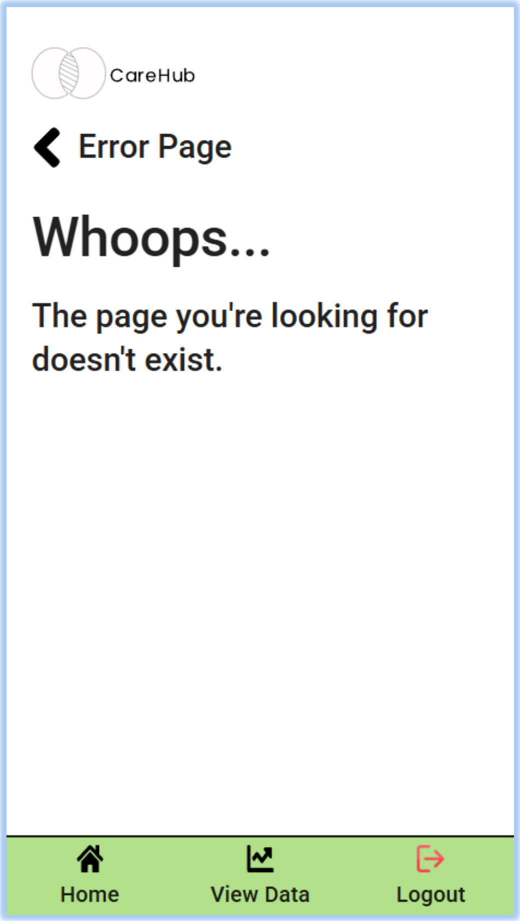
Pick a Date:

29-01-2025

Type	Intensity	Duration	Note
חדר כושר	Low	60 min	
אלוריל 100			Number of pills: 1
Mood	5		Yay
פיצה			Proteins: 25g Fats: 80g Carbohydrates: 100g Dietary Fiber: 15g

10. Error Screen:

This screen serves as the default error page (404) when users attempt to access unavailable or non-existent resources in CareHub.



Error Page

Whoops...

The page you're looking for doesn't exist.

8.2 Maintenance Guide

8.2.1 Web Application Maintenance Guide

Development Tools for MERN Stack:

1. Code Editor or Integrated Development Environment (IDE)

- Tool: Visual Studio Code (VS Code)
- Link: https://code.visualstudio.com/
- Purpose: A powerful and popular code editor with features like syntax highlighting, debugging, and extensions for JavaScript, React, Node.js, and MongoDB.

2. Node.js

- Tool: Node.js
- Link: https://nodejs.org/en
- Purpose: Enables server-side JavaScript execution and includes npm (Node Package Manager) for managing libraries and dependencies.
- Installation Tip: Download and install the latest LTS (Long Term Support) version for stability.

3. MongoDB

- Tool: MongoDB Community Server or MongoDB Atlas (cloud-hosted).
- Link (Community): https://www.mongodb.com/try/download/community
- Link (Atlas): https://www.mongodb.com/atlas
- Purpose: A NoSQL database for storing application data. MongoDB Compass (GUI) can also be installed to manage and visualize the database locally.

4. Git and GitHub

- Tool: Git and GitHub or other version control platforms.
- Link (Git): https://git-scm.com/
- Link (GitHub): https://github.com/omersbh/CareHub
- Purpose: For version control, collaboration, and source code management.
- Installation Tip: Install Git on your local machine and configure a GitHub repository for your project.

5. Browser

- Tool: Google Chrome (preferred).
- Purpose: For testing the web application and debugging using Chrome Developer Tools.
- Optional Extension: Install React Developer Tools for easier debugging of React components.

6. Postman

- Tool: Postman
- Link: https://www.postman.com/
- Purpose: For testing and debugging API endpoints in the Node.js/Express backend.

7. Package Managers

- npm (Node Package Manager): Installed with Node.js.

8. Build Tools

- Tool: Webpack or Vite (configured through the React project setup).
- Purpose: To bundle the application’s JavaScript, CSS, and other assets for production.

Other Tools for Maintenance:

9. Environment Variable Management

- Tool: dotenv npm package.
- Purpose: To manage environment variables securely (e.g., database URLs, API keys).

10. Optional Testing Tools

- Tool: Jest or Mocha for unit testing.
- Purpose: To test individual components and backend endpoints.
- Optional: Use tools like Cypress for end-to-end testing.

11. Deployment Platforms

- Tool: Render.
- Link: https://render.com/
- Purpose: For hosting the web application. MongoDB Atlas can be used for database hosting.

12. Task Automation

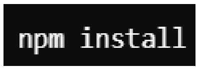
- Tool: nodemon (for live-reloading during development).
- Installation: npm install -g nodemon.
- Purpose: Automatically restarts the server when changes are made.

13. Containerization (Optional)

- Tool: Docker.
- Link: https://www.docker.com/
- Purpose: To containerize the MERN stack app for easier deployment and maintenance.

Checklist for Installation:

1. Install Node.js and ensure npm is working: node -v and npm -v.
2. Install MongoDB locally or set up a MongoDB Atlas cluster.
3. Set up Git and clone your repository.
4. Install required npm packages for the project:



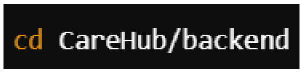
```
npm install
```

5. Set up .env file with the necessary environment variables (e.g., MongoDB connection string).
6. Install any additional tools, such as Postman and Docker, based on your workflow.

Steps to Run the Application Locally:

1. Open the Terminal in VSCode

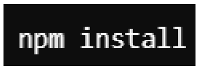
- Launch Visual Studio Code.
- Open the CareHub project directory in VSCode.
- Access the terminal by clicking on Terminal > New Terminal in the top menu or by pressing *Ctrl* + ~ (Windows/Linux) or *Cmd* + ~ (Mac).
- Navigate to the backend directory by typing the following command and pressing Enter:



```
cd CareHub/backend
```

2. Install Dependencies

- In the terminal, ensure you're inside the backend directory.
- Run the following command to install all required dependencies for the backend:

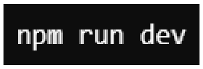


```
npm install
```

- This command will read the package.json file and automatically install all necessary libraries and packages required for the backend.

3. Start the Server

- After successfully installing dependencies, run the following command to start the server in development mode:



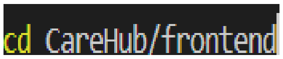
```
npm run dev
```

- This will:

    - Start the backend server using a tool like nodemon, which automatically reloads the server when you make code changes.
    - Output logs in the terminal, such as the server running and listening on the specified port

4. Open new Terminal in VSCode

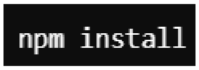
- Navigate to the frontend directory by typing the following command and pressing Enter:



```
cd CareHub/frontend
```

5. Install Dependencies

- Run the following command to install required dependencies for the frontend

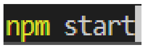


```
npm install
```

- This command will read the package.json file and automatically install all necessary libraries and packages required for the frontend.

6. Start the Frontend

- Run the following command to start the Frontend



```
npm start
```

- This will:
    - Start the frontend on the local host URL: http://localhost:5173.
    - Automatically reload the frontend if you make changes to the code.

8.2.2 Database Structure

The CareHub database is designed using a NoSQL MongoDB structure. It organizes data into collections, each tailored to store specific types of information related to users and their activities. Below is a detailed explanation of the database collections and their respective fields:

1. *user* Collection
This collection stores the core information for all users, including patients and caregivers.
Fields:

- fullName (String): The full name of the user.
- email (String): The user's unique email address, used for login and identification.
- password (String): A hashed password for user authentication.
- role (String): The role of the user, such as "patient" or "caregiver".

2. *medicine* Collection
This collection tracks the medications taken by users.

Fields:

- userId (ObjectId): A reference to the corresponding user in the user collection.
- name (String): The name of the medicine.
- type (String): The category or type of the medicine (e.g., tablet, syrup).
- quantity (Number): The dosage or quantity of the medicine taken.
- dateTaken (Date): The date and time when the medicine was taken.

3. *nutrition* Collection

This collection logs the dietary intake of users.

Fields:

- userId (ObjectId): A reference to the corresponding user in the user collection.
- foodName (String): The name of the food item consumed.
- dateTaken (Date): The date and time when the food was consumed.
- image (String): A URL or file path to an image of the food item (optional).
- nutritionalValues (Object): A nested object containing the nutritional breakdown:
    - proteins (Number): Amount of protein in grams.
    - fats (Number): Amount of fat in grams.
    - carbohydrates (Number): Amount of carbohydrates in grams.
    - dietaryFiber (Number): Amount of dietary fiber in grams.

4. *symptom* Collection

This collection records symptoms experienced by users, providing detailed insights into their health status.

Fields:

- userId (ObjectId): A reference to the corresponding user in the user collection.
- type (String): The type of symptom experienced. Possible values include:
    - 'Muscle Rigidity'
    - 'Muscle Stiffness'
    - 'Muscle Contraction'
    - 'Speech Difficulty'
    - 'Tremor'
    - 'Parkinson's State'
    - 'Mood'
    - 'Stool'
- severity (Number): A severity rating on a scale from 1 (mild) to 5 (severe).
- notes (String): Additional notes or comments about the symptom (optional).
- date (Date): The date and time when the symptom was recorded.

5. *training* Collection

This collection tracks the physical activities and exercises performed by users.

Fields:

- userId (ObjectId): A reference to the corresponding user in the user collection.
- exerciseName (String): The name of the exercise or activity performed.
- duration (Number): The duration of the activity in minutes.
- intensity (String): The intensity level of the exercise (e.g., "low," "moderate," "high").
- date (Date): The date and time when the training occurred.
- notes (String): Additional comments or observations about the training session (optional).

Key Points:

1. Relationships: The userId field acts as a foreign key, linking records in the medicine, nutrition, symptom, and training collections to a specific user in the user collection.
2. Scalability: The NoSQL design ensures scalability by allowing flexible, schema-less storage while maintaining the ability to easily add new collections or fields in the future.
3. Data Security: Sensitive fields such as password are stored in hashed format to ensure security. Authentication and authorization mechanisms are implemented to control access to user data.

8.2.3 Backend Folder Structure

The backend of CareHub is organized into a structured folder hierarchy that ensures scalability, readability, and maintainability. Below is a detailed explanation of the backend folders and their roles:

1. *config* Folder

The config folder contains configuration files required to set up and manage core application settings.

Files:

- allowedOrigins.js: Specifies the origins allowed to interact with the backend, primarily for enabling secure CORS (Cross-Origin Resource Sharing).
- corsOptions.js: Configures CORS settings, such as methods and headers.
- dbConn.js: Handles the database connection logic, including setting up the connection with MongoDB using environment variables for security.

2. *controllers* Folder

This folder contains individual controllers for each data collection. Each controller handles the core business logic for creating, reading, updating, and deleting data (CRUD operations).

Files:

- medicinesController.js.
- nutritionsController.js.
- ymptomsController.js.

- trainingsController.js.
- usersController.js.

Each controller:

- Handles incoming requests for specific routes.
- Interacts with the database to perform CRUD operations.
- Returns appropriate responses to the client.

3. *logs* Folder

The logs folder is dedicated to application logging, helping track errors and requests for debugging and monitoring purposes.

Files:

- errLog.log: Stores error logs, capturing details of application failures or exceptions.
- reqLog.log: Logs incoming requests, including request details like endpoint, timestamp, and method.

4. *middleware* Folder

Middleware functions act as brokers between incoming requests and route handlers, performing tasks such as logging, error handling, and security checks.

Files:

- errorHandler.js: Manages error handling by capturing application-level errors and returning standardized responses.
- logger.js: Implements request logging by capturing details of each request and storing them in reqLog.log.

5. *routes* Folder

This folder defines the routing logic for different pages and features. Routes are grouped based on functionality for better organization.

Files:

- root.js.
- medicineRoutes.js.
- nutritionRoutes.js.
- symptomRoutes.js.
- trainingRoutes.js.
- userRoutes.js.

Each file defines endpoints (e.g., /api/medicine, /api/nutrition) and maps them to the corresponding controller functions.

6. *views* Folder

The views folder contains static HTML files used for specific scenarios, such as error pages or landing pages.

Files:

- 404.html: A custom 404 error page displayed when a user accesses an undefined route.
- index.html: The main landing page for the application.

7. *server.js* File

The server.js file is the entry point of the backend application. It initializes and configures the server, middleware, routes, and database connection.

Key Responsibilities:

- Sets up the Express server.
- Configures middleware such as CORS, logging, and error handling.
- Imports and uses routes for various collections.
- Establishes a connection to the MongoDB database.
- Listens for incoming requests on the specified port.

8. .env File

The .env file is a critical component in the backend of CareHub. It securely stores sensitive configuration data and environment-specific variables that should not be hardcoded into the source code. Using the .env file helps enhance security and flexibility when deploying the app in different environments (development, testing, production).

9. .gitignore File

The .gitignore file is an essential component for maintaining a clean and secure repository. It specifies files and directories that should not be tracked by version control systems like Git. This helps reduce repository size, avoid committing unnecessary files, and protect sensitive information.

Summary of Backend Design:

This structured backend design ensures modularity, ease of debugging, and scalability. Each folder and file has a clear purpose, making it easier for developers (current or future) to understand, maintain, and enhance the backend functionality of CareHub.

9. Evaluation / Verification Plan

9.1 Testing Plan

To ensure the reliability of our final product, we have developed a testing plan, which is detailed in the table below. This plan is based on the Use Case descriptions we created and the GUI screens, allowing us to identify potential problematic scenarios, weaknesses, and processes requiring accuracy verification.

#	Test Subject	Test Headline	Expected Results	Actual Results
1	User Registration	New user signup with valid information	User account is created successfully, and user is redirected to login page	✓
		Attempt to sign up with an existing email	System displays an error message indicating the email is already in use	✓
2	User Authentication	Existing user login with correct information	User is logged in and directed to their dashboard	✓
		Login with incorrect password	System displays an error message and prevents login	✓
3	Nutrition Intake	PD Patient enters daily nutrition data	Nutrition data is saved and reflected in the patient's health record	✓
4	Physical Activity	PD Patient logs a new physical activity	Activity is recorded and added to the patient's exercise log	✓
5	Food Intake	PD Patient records food intake for a meal	Food intake is saved and updated in the patient's log	✓
6	Symptom Logging	PD Patient logs current symptoms	Symptoms are recorded and added to the patient's health timeline	✓
7	Overall Feeling Rating	PD Patient rates their overall feeling for the day	Rating is saved and reflected in the patient's daily health summary	✓

8	Patient Data View	User accesses patient data	Patient's health data is displayed in an extensive and organized way	✓
9	Accessibility	Navigate through app using only mouse or finger (touch screen)	All features are accessible and operable without using keyboard	✓
10	Performance	Load patient dashboard	Dashboard loads within 3 seconds on a standard internet connection	This depends on the deploy platform and payments
11	Data Protection (Security)	Attempt to access patient data without authorization	Access is denied and security measures are triggered	✓
12	Error Handling	Enter invalid data in a form field	Clear error message is displayed, and form is not submitted	✓
13	Responsiveness	Access application on mobile device	Interface adapts to screen size without loss of functionality	✓
14	Data Consistency	Enter data and view it across different pages	Entered data is consistently displayed across all relevant sections of the app	✓
15	User Interface	Evaluate scrolling requirement on main pages	Minimal scrolling required on desktop and mobile interfaces	✓

9.2 Evaluation by User

The system faced extensive user evaluation through testing sessions with Michael Jackont, a PD patient who served as our primary user tester. To assess the system's usability in a structured manner, we employed the System Usability Scale (SUS) questionnaire, a widely recognized tool for measuring user experience, and got a score of 80. Additionally, detailed feedback was collected and documented throughout the testing process, highlighting specific areas for improvement. Based on this valuable input, we implemented several refinements to enhance the system's accessibility and user experience. The complete SUS results and detailed user feedback documentation are in the Appendix chapter, along with a comprehensive list of the implemented improvements that emerged from this evaluation phase.

References

1. Kalia, L. V., & Lang, A. E. (2015). Parkinson's disease. The Lancet, 386(9996), 896-912.
2. Dorsey, E. A., Constantinescu, R., Thompson, J. P., Biglan, K. M., Holloway, R. G., Kieburtz, K., ... & Tanner, C. M. (2007). Projected number of people with Parkinson disease in the most populous nations, 2005 through 2030. Neurology, 68(5), 384-386.
3. Mak, M. K., Wong-Yu, I. S., Shen, X., & Chung, C. L. (2017). Long-term effects of exercise and physical therapy in people with Parkinson disease. Nature Reviews Neurology, 13(11), 689-703.
4. Ongun, N. (2018). Does nutritional status affect Parkinson's Disease features and quality of life?. PLoS One, 13(10), e0205100.
5. Goetz, C. G. (2011). The history of Parkinson's disease: Early clinical descriptions and neurological therapies. Cold Spring Harbor Perspectives in Medicine, 1(1), a008862.
6. van der Kolk, N. M., & King, L. A. (2013). Effects of exercise on mobility in people with Parkinson's disease. Movement Disorders, 28(11), 1386-1396.
7. Chaudhuri, K. R., & Schapira, A. H. (2009). Non-motor symptoms of Parkinson’s disease: Dopaminergic pathophysiology and treatment. Lancet Neurology, 8(5), 464-474.
8. Schrag, A., Jahanshahi, M., & Quinn, N. (2000). What contributes to quality of life in patients with Parkinson's disease? Journal of Neurology, Neurosurgery & Psychiatry, 69(3), 308-312.
9. Poewe, W., & Mahlknecht, P. (2009). The clinical progression of Parkinson’s disease. Parkinsonism & Related Disorders, 15(S3), S28-S32.
10. Grosset, K. A., Bone, I., & Grosset, D. G. (2009). Switching medication in Parkinson’s disease: Practical issues and management. Postgraduate Medical Journal, 85(1006), 622-628.
11. Goodwin, V. A., Richards, S. H., Taylor, R. S., Taylor, A. H., & Campbell, J. L. (2008). The effectiveness of exercise interventions for people with Parkinson's disease: A systematic review and meta-analysis. Movement Disorders, 23(5), 631-640.
12. Barichella, M., Cereda, E., Pezzoli, G., & Faierman, S. A. (2017). Diet and nutritional considerations in Parkinson's disease. Frontiers in Aging Neuroscience, 9, Article 80.
13. Pandey, S., & Bajaj, B. K. (2018). Sleep disorders in Parkinson's disease. Journal of

Neurology, 265(4), 850-860.

14. Pickut, B. A., Vanneste, S., Hirsch, M. A., van Hecke, W., Kerckhofs, E., Marien, P., & Parizel, P. M. (2015). Mindfulness training among individuals with Parkinson's disease: Neurobehavioral effects. Parkinson's Disease, 2015, Article 816404.
15. Svenningsson, P., Westman, E., Ballard, C., & Aarsland, D. (2012). Cognitive impairment in patients with Parkinson's disease: diagnosis, biomarkers, and treatment. The Lancet Neurology, 11(8), 697-707.
16. Espay, A. J., Bonato, P., Nahab, F. B., Maetzler, W., Dean, J. M., Klucken, J., ... & Papapetropoulos, S. (2016). Technology in Parkinson's disease: Challenges and opportunities. Movement Disorders, 31(9), 1272-1282.
17. Armstrong, M. J., & Okun, M. S. (2020). Diagnosis and Treatment of Parkinson Disease: A Review. *JAMA*, *323*(6), 548–560.
18. https://medium.com/@ashish-k-mishra/getting-started-with-mern-stack-building-your-first-web-application-0bbb6799d251
19. https://colorbrewer2.org/#type=qualitative&scheme=Paired&n=5

Appendix

System Usability Scale (SUS):

Questions and Raw Scores that Michael Jackont, the PD patient, filled:

1. I think that I would like to use this system frequently (5)
2. I found the system unnecessarily complex (4)
3. I thought the system was easy to use (5)
4. I think that I would need the support of a technical person to use this system (1)
5. I found the various functions in this system were well integrated (5)
6. I thought there was too much inconsistency in this system (4)
7. I would imagine that most people would learn to use this system very quickly (4)
8. I found the system very cumbersome to use (1)
9. I felt very confident using the system (4)
10. I needed to learn a lot of things before I could get going with this system (1)

Calculations: Odd-numbered questions (subtract 1 from score):

- Q1: 5 - 1 = 4
- Q3: 5 - 1 = 4
- Q5: 5 - 1 = 4
- Q7: 4 - 1 = 3
- Q9: 4 - 1 = 3 Odd-numbered subtotal: 18

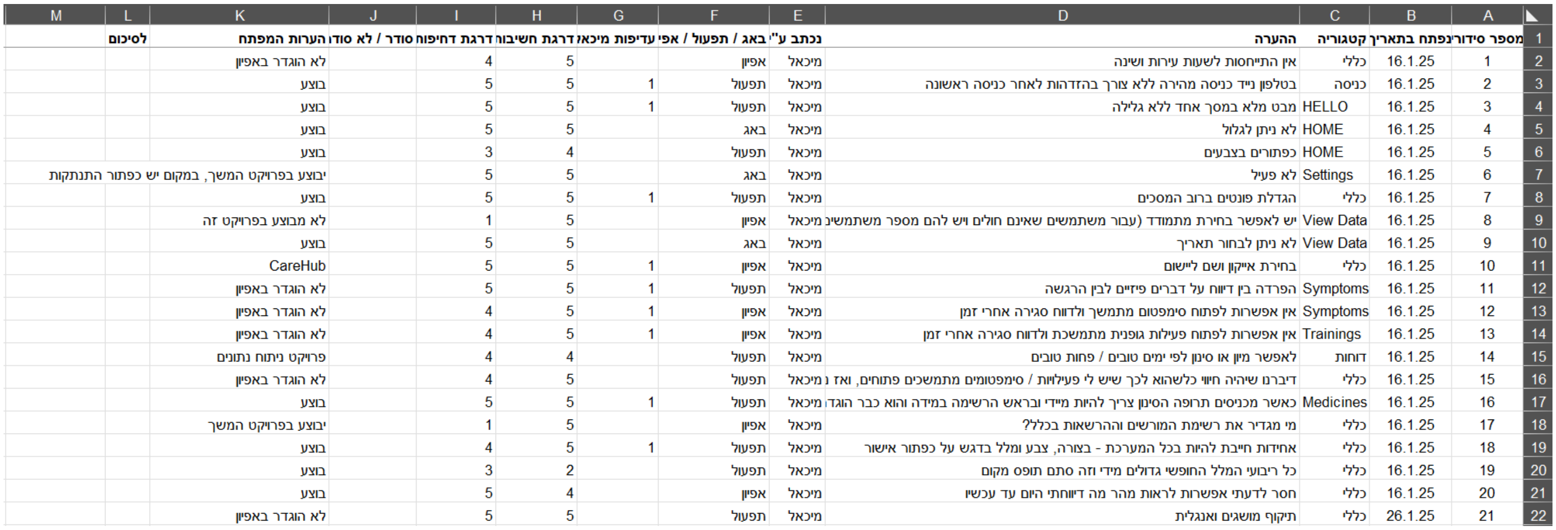
Even-numbered questions (subtract score from 5):

- Q2: 5 - 4 = 1
- Q4: 5 - 1 = 4
- Q6: 5 - 4 = 1
- Q8: 5 - 1 = 4
- Q10: 5 - 1 = 4 Even-numbered subtotal: 14

Total calculation:

- Sum of adjusted scores: 18 + 14 = 32
- Final SUS Score: 32 × 2.5 = 80/100

This score of 80 indicates excellent system usability, as it's well above the average score of 68.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	מספר סידור	נפתח בתאריך	קטגוריה	ההערה	נכתב ע"'	באג / תפעול / אפי	עדיפות מיכא	דרגת חשיבו	דרגת דחיפו	סודר / לא סוד	הערות המפתח	לסיכום	
2	1	16.1.25	כללי	אין התייחסות לשעות עירות ושינה	מיכאל	אפיון		5	4		לא הוגדר באפיון		
3	2	16.1.25	כניסה	בטלפון נייד כניסה מהירה ללא צורך בהזדהות לאחר כניסה ראשונה	מיכאל	תפעול	1	5	5		בוצע		
4	3	16.1.25	HELLO	מבט מלא במסך אחד ללא גלילה	מיכאל	תפעול	1	5	5		בוצע		
5	4	16.1.25	HOME	לא ניתן לגלול	מיכאל	באג		5	5		בוצע		
6	5	16.1.25	HOME	כפתורים בצבעים	מיכאל	תפעול		4	3		בוצע		
7	6	16.1.25	Settings	לא פעיל	מיכאל	באג		5	5		יבוצע בפרויקט המשך, במקום יש כפתור התנתקות		
8	7	16.1.25	כללי	הגדלת פונטים ברוב המסכים	מיכאל	תפעול	1	5	5		בוצע		
9	8	16.1.25	View Data	יש לאפשר בחירת מתמודד (עבור משתמשים שאינם חולים ויש להם מספר משתמשים	מיכאל	אפיון		5	1		לא מבוצע בפרויקט זה		
10	9	16.1.25	View Data	לא ניתן לבחור תאריך	מיכאל	באג		5	5		בוצע		
11	10	16.1.25	כללי	בחירת אייקון ושם ליישום	מיכאל	אפיון	1	5	5		CareHub		
12	11	16.1.25	Symptoms	הפרדה בין דיווח על דברים פיזיים לבין הרגשה	מיכאל	תפעול	1	5	5		לא הוגדר באפיון		
13	12	16.1.25	Symptoms	אין אפשרות לפתוח סימפטום מתמשך ולדווח סגירה אחרי זמן	מיכאל	אפיון	1	5	4		לא הוגדר באפיון		
14	13	16.1.25	Trainings	אין אפשרות לפתוח פעילות גופנית מתמשכת ולדווח סגירה אחרי זמן	מיכאל	אפיון	1	5	4		לא הוגדר באפיון		
15	14	16.1.25	דוחות	לאפשר מיון או סינון לפי ימים טובים / פחות טובים	מיכאל	תפעול		4	4		פרויקט ניתוח נתונים		
16	15	16.1.25	כללי	דיברנו שיהיה חיווי כלשהוא לכך שיש לי פעילויות / סימפטומים מתמשכים פתוחים, ואז נ	מיכאל	תפעול		5	4		לא הוגדר באפיון		
17	16	16.1.25	Medicines	כאשר מכניסים תרופה הסינון צריך להיות מיידי ובראש הרשימה במידה והוא כבר הוגד	מיכאל	תפעול	1	5	5		בוצע		
18	17	16.1.25	כללי	מי מגדיר את רשימת המורשים וההרשאות בכלל?	מיכאל	אפיון		5	1		יבוצע בפרויקט המשך		
19	18	16.1.25	כללי	אחידות חייבת להיות בכל המערכת - בצורה, צבע ומלל בדגש על כפתור אישור	מיכאל	תפעול	1	5	4		בוצע		
20	19	16.1.25	כללי	כל ריבועי המלל החופשי גדולים מידי וזה סתם תופס מקום	מיכאל	תפעול		2	3		בוצע		
21	20	16.1.25	כללי	חסר לדעתי אפשרות לראות מהר מה דיווחתי היום עד עכשיו	מיכאל	אפיון		4	5		בוצע		
22	21	26.1.25	כללי	תיקוף מושגים ואנגלית	מיכאל	תפעול		5	5		לא הוגדר באפיון		

The PD patient, Michael Jackont, provided a detailed Excel file containing notes and suggestions for improvements in the application. This feedback document included various observations and recommendations for different features and functionalities of the system. Some of these issues have been successfully addressed and implemented, while others remain to be handled in future updates. This kind of user feedback is valuable for the continuous improvement and development of the application to better serve PD patients' needs.