תמונה שמכילה טקסט, גופן, לוגו, גרפיקה

התיאור נוצר באופן אוטומטי

**Capstone Project Phase B**

**Project Code: 25-1-D-6**

**Parkinson's Perspective: Visualizing Impact**

**Supervisor: Dr. Julia Sheidin**

**Supervisor: Dr. Avital Shulner Tal**

**Omer Fatal – omer.omer92@gmail.com**

**Tanya Gendelman - TanyaGendelman@gmail.com**

**[GitHub Repository](https://github.com/OmerFatal/FinalProject_Parkinson-s_Visualization)**

**Table of contents**

[**Abstract**](#_iqrx3clbyd07) **3**

[**2. Introduction**](#_x7inie3wslvk) **4**

[**3. Background and Related Work**](#_r44x4s8ey6e3) **4**

[3.1 Parkinson disease in general](#_fwc0mmv064af) 4

[3.2 Impact on Parkinson's Disease Patients](#_hvf08ekw47pd) 5

[3.3 The Information Needs of People with Parkinson's Disease](#_r9tuhxpqsqju) 5

[3.4 The Importance of Explanations in Parkinson's Disease Management](#_d6zfow5xeu86) 5

[3.5 Summary](#_s7futhsqbdeo) 6

[**4. Solution Description**](#_wbwnp94gya81) **6**

[**5. Development Process**](#_ydsoruyomi9z) **8**

[5.1 Application Features and User Role](#_i5ckfsl5wo9) 8

[5.2 Code Implementation](#_cpkcyaucbil8) 10

Frontend Development………………………………………………………………………...11

[Backend Development](#_8gj61osynv33) 12

[**6. Faced Challenges and Solutions**](#_xyuu7zfw28z) **12**

[6.1 Planning and Functional Challenges](#_3qbyyddt5nk) 12

[6.2 Design and Aesthetics Challenges](#_krnrj9egz7ot) 14

[**7. Results and Conclusions**](#_i8ib8t8vezbs) **15**

[7.1 Goals and Achievements](#_284jutftn57l) 15

[7.2 Conclusions](#_8nhcjvmhexcc) 16

[7.3 Limitations and Future Work](#_cksw8po59k3m) 17

[**8. Evaluation / Verification Plan**](#_8zs9x9fgqqcj) **17**

[8.1 Testing Plan](#_nxtufe7ru77a) 17

[8.2 User evaluation](#_mqhsn47nj24r) 19

[**9. User Documentation**](#_tqoy4l8802f3) **20**

[9.1 User Guide](#_gngqaluvordo) 20

[9.1.1 Operating Instructions – Visualization Module](#_u06s59s2x9p0) 20

[9.1.2 Maintenance Guide](#_oq101zsv2nre) 26

**References 37**

# 

# Abstract

Parkinson’s disease is a progressive neurological disorder that affects movement, balance, and coordination, making daily life increasingly challenging for patients. To support day-to-day management, we developed an intuitive, browser-based visualization module that transforms self-reported data into clear and engaging visual representations.

Designed specifically for Michael, a person living with Parkinson’s, the system is built on the personal health data he records in the CareHub application. It features a monthly heatmap that offers a quick overview of Michael’s physical and emotional condition, and a daily dashboard with interactive graphs related to medication intake, nutrition, physical activity, and cognitive tasks.

All data is presented in a single, user-friendly interface that enables smooth navigation between visualizations. This allows users to easily observe changes over time and better understand their daily routines.

The design is tailored to Michael’s needs, providing him with a meaningful way to connect with his own data and stay actively engaged in managing his health.

# 2. Introduction

Parkinson’s disease (PD) is a progressive neurological condition that significantly impairs motor function, balance, and coordination, leading to symptoms such as tremors, stiffness, and slowness of movement ‎[[6]](#_vd638a50ccx8). For individuals living with PD, managing the condition is a complex and ongoing process that requires careful monitoring of symptoms and daily routines [‎[2]](#_vd638a50ccx8). Even small adjustments in everyday habits—such as the timing of medication, consistency of physical activity, or dietary choices—can have a meaningful impact on symptom progression and overall well-being [‎[1][7]](#_vd638a50ccx8). However, many patients struggle to connect their daily actions with their health outcomes, leaving them with limited clarity and reduced confidence when making care-related decisions ‎[[2][4]](#_vd638a50ccx8).

With this in mind, our vision is to help patients better navigate Parkinson’s disease by turning complex information into clear and accessible visual content. Using intuitive and customized visualizations, we aim to empower patients—particularly Michael, the user at the center of our project—to better understand their condition and become more actively engaged in their treatment journey.

To address the challenge of data interpretation and usability, we developed our project: a web-based extension to an existing application in which patients record personal health data. This extension visualizes the collected information in a unified dashboard, presenting daily activities, nutrition, medication adherence, and overall wellness indicators. The result is a comprehensive and user-friendly tool that enables patients to track their condition, gain a clearer understanding of their health status, and make informed choices that support better quality of life.

# 3. Background and Related Work

## **3.1 Parkinson disease in general**

Parkinson’s disease (PD) is a chronic neurological condition characterized by the gradual degeneration of brain cells, particularly in the substantia nigra region [‎[6]](#_vd638a50ccx8). This degeneration leads to a decrease in dopamine production, a neurotransmitter that plays a key role in regulating movement and other neurological functions [‎[1]](#_vd638a50ccx8). While the precise cause of PD remains uncertain, it is believed to result from a combination of genetic predispositions and environmental exposures [‎[6]](#_vd638a50ccx8). Although the majority of cases are diagnosed after the age of 60, early-onset Parkinson’s can also occur [‎[2]](#_vd638a50ccx8). As the second most common neurodegenerative disorder globally, the prevalence of PD is expected to rise with the aging population [‎[2]](#_vd638a50ccx8). Early detection and advancements in treatment are essential for effective disease management [‎[7]](#_vd638a50ccx8)

## **3.2 Impact on Parkinson's Disease Patients**

PD has a wide-ranging impact on patients' lives, affecting both motor abilities—such as tremors, muscle rigidity, and slowed movement—and non-motor aspects like sleep issues, cognitive decline, and mood disturbances [‎[8].](#_vd638a50ccx8) These combined symptoms make it difficult to perform everyday tasks such as personal care and driving, often resulting in social withdrawal, emotional strain, and financial difficulties, including premature retirement or job loss ‎[[8]](#_vd638a50ccx8).  
 Although available treatments can alleviate some symptoms, they often come with side effects and require ongoing medical supervision [‎[9].](#_vd638a50ccx8) Given the progressive nature of the disease, patients and their families must constantly adapt to new challenges, which gradually undermine their independence and significantly affect their overall quality of life [‎[8].](#_vd638a50ccx8)

## **3.3 The Information Needs of People with Parkinson's Disease**

People living with PD often face difficulties in accessing the information they need to manage their condition effectively. Many patients lack a clear understanding of how their daily routines—such as diet, physical activity, and medication timing—directly influence their symptoms and overall well-being ‎[[1][6]](#_vd638a50ccx8). For instance, they may be unsure which exercises are most beneficial, how certain foods affect their energy or mobility, or how adjusting medication schedules could improve symptom control [‎[1][7]](#_vd638a50ccx8).

Additionally, patients frequently struggle to track how their condition evolves over time or to recognize how factors like stress and sleep quality contribute to fluctuations in symptoms [‎[6]](#_vd638a50ccx8). These gaps in understanding, make it harder to make informed and personalized care decisions. While general advice is available, it is often too broad and fails to address the unique needs of each individual, leaving patients without practical guidance on how to live well with Parkinson’s [‎[2][4].](#_vd638a50ccx8)  
Online communities offer some level of support by allowing patients to share experiences and advice. However, such discussions often lack scientific grounding or personalized direction. This highlights a clear need for reliable, tailored, and evidence-based information that can empower patients to better understand their condition and make more confident, health-related decisions [‎[3].](#_vd638a50ccx8)

## **3.4 The Importance of Explanations in Parkinson's Disease Management**

Textual elements within the system support user orientation by providing brief, contextual descriptions.For example, when hovering over an activity or medication on a graph, the system displays relevant information such as the name of the medication, dosage, or category of the activity. While these are not full textual explanations or health recommendations, they enhance user understanding of the visualized data and make the navigation experience more intuitive.  
This lightweight textual support helps users better interpret the meaning behind each data point without overwhelming them with complex medical language.

Visual representations, such as graphs, charts, and heatmaps, simplify complex information and help identify patterns and trends over time. These tools support effective self-management by showing how lifestyle changes impact health outcomes [‎[8]](#_vd638a50ccx8).

By combining both types of explanations, it becomes possible to bridge the gap between raw data and tailor information to diverse needs, and enable patients to make more informed decisions and communicate more effectively with healthcare providers, leading to smarter and more effective Parkinson’s management.

## **3.5 Summary**

Visualizations play a central role in making health data accessible, understandable, and actionable for individuals with PD. By transforming raw data into intuitive graphical formats—such as line graphs, bar charts, heatmaps, and timelines—our solution enables patients to recognize symptom patterns, track changes over time, and understand how daily routines impact their condition. These visual tools are designed to be clear and interactive, allowing users to easily navigate between different data views, compare trends, and gain insights that support self-management. The use of customized visual dashboards empowers patients like Michael to take an active role in monitoring their health, adapting their routines, and improving their overall quality of life, without requiring advanced technical knowledge or constant reliance on healthcare professionals.

# 4. Solution Description

As part of our research, we reviewed a wide range of applications and technological solutions focused on presenting data for individuals with PD. The goal of this review was to explore how complex medical information can be made accessible, clear, and personally relevant. In addition, we held several meetings with Michael—a 57-year-old Parkinson’s patient from Israel—to gain a deep understanding of his daily challenges, his specific needs, and the types of visualizations that could help him better understand his condition. This process guided us in developing a precise and effective solution tailored specifically to Michael.

Michael currently manages his condition using the CAREHUB application, which was developed by students Omer and Aviram as part of a final-year project at Braude College. The app allows Michael to log daily information, including medication intake, meals, physical activity, and his physical state prior to those activities. Each entry is time-stamped, enabling detailed and accurate tracking of his routine.

Building on this platform, our project functions as an extension to the existing CAREHUB app—it connects to the data Michael records and presents it in a clear, user-friendly visual format. All the data is consolidated into a single central dashboard, which displays trends using graphical visualizations. This creates a simple, efficient, and personalized interface that helps Michael understand how his daily actions influence his symptoms, while allowing him to track his health progress over time with ease.

We used a user-centered design (UCD) approach, placing Michael’s real-life challenges at the core of our development process. Through constant feedback loops and testing with mockups, we ensured that our visualizations would be both easy to interpret and relevant to Michael’s needs. The result is a system that not only supports his ongoing health management but also encourages self-awareness and long-term engagement.

**Functional Requirements:**

1. The system retrieves and displays categorized health data from CAREHUB, such as medications, meals, physical activity, symptoms, and overall condition.
2. The system presents a summarized overview of user-reported data across different days, allowing selection of a specific day, month, or year.
3. Clicking on a specific date within the heat map redirects the user to a detailed daily dashboard for that day.
4. The daily dashboard includes clear visualizations (graphs and charts) for each logged activity.
5. The user can switch between daily, monthly, and yearly views to track trends and changes over time.
6. The system includes filtering options.
7. The system is synchronized in real time with the data entered in CAREHUB.

**Non-Functional Requirements:**

1. The system displays an interactive heatmap in the form of a monthly calendar, where each day is divided into three triangles representing cognitive, physical, and emotional states, using a color scale from 1 (best) to 5 (worst).
2. The transition from the heat map to the daily dashboard should be seamless and fast.
3. The visual layout must avoid information overload—each graph and data point should be presented clearly and concisely.
4. All visual elements must be responsive and optimized for both desktop and mobile use.
5. The system must comply with medical data privacy regulations and ensure secure handling of sensitive information.
6. The system should be easy to maintain and scalable, allowing for the future addition of new features (e.g., additional visualization types or analytical tools).

# 5. Development Process

The development process began with a review of the medical background of Parkinson’s disease and the needs of individuals living with it. We then explored existing visualization tools and identified relevant user requirements based on Michael’s previously logged data.

Next, we reviewed various visualization techniques and selected those best suited for our goals. The system architecture was planned accordingly, focusing on usability and clarity. Throughout the development process, feedback was collected and used to iteratively refine the interface, resulting in a tailored and practical visualization platform.

## **5.1 Application Features and User Role**

Our visualization platform was designed specifically and exclusively for Michael Jackont, a 57-year-old individual living with Parkinson’s disease. It serves as a complementary tool to the CareHub web application, which Michael uses to log his daily routines, including medication intake, physical activity, meals, symptoms, and overall well-being.

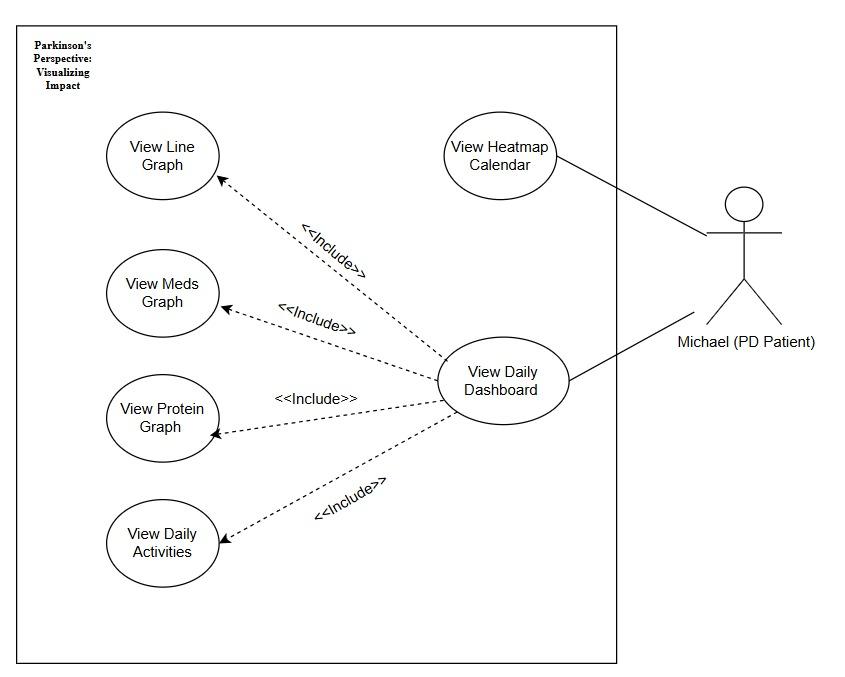
While CareHub focuses on data entry, our solution transforms Michael’s personal data into meaningful visual representations of data. These data are presented in a clear and intuitive interface tailored to his cognitive and physical needs.

The main use cases of the system, as illustrated in the Use Case Diagram (see Figure 1), include:

* **Viewing a heatmap calendar**, which serves as the system’s entry point and provides an overview of each day’s condition through a color-coded scale.
* **Accessing the daily dashboard**, which serves as the central screen for analyzing data collected during the day   
  + Line graphs showing Parkinson’s state, mood, and physical difficulty throughout the day (see figure 4)
  + Stacked bar chart for Medication intake summary by type and time (see figure 5).
  + Bar chart for Protein consumption breakdown with nutritional details (see figure 6).
  + Gantt chart for Daily activities displayed by category (cognitive, sport, household) (see figure 7).

These visual components extend the daily dashboard and allow Michael to understand patterns in his daily life better. The selection of visualization types was intentional—each one was carefully chosen to suit the nature of the data and to make data interpretation easier. Line graphs were selected for their strength in illustrating trends over time, which is essential for tracking daily fluctuations in Michael’s physical, emotional, and neurological state. Stacked bar charts intuitively display the timing and distribution of different types of medication intake. The protein bar chart provides a straightforward breakdown for comparing various nutritional sources, based on the dietary information Michael tracks. The Gantt chart was chosen because it gives a clear overview of his daily schedule by category, helping him understand how different types of activities influence his symptoms throughout the day.

The dashboard is interactive and dynamically designed to reflect data updates. While the initial goal was to connect it directly to the CareHub database, due to several challenges (outlined below), the current implementation loads data from a CSV file. Since the solution was developed specifically for Michael, he is the only user role defined in the system. All features were designed in close collaboration with him, tailored to his daily routine and based on his feedback throughout the development process. This user-centered approach ensures that the platform is not only technically effective, but also relevant, accessible, and supportive in managing his Parkinson’s symptoms.



**Figure 1: Use Case diagram**

## **5.2 Code Implementation**

During the code implementation phase, we developed our visualization platform as a complementary tool to the CareHub application. Our system connects to CareHub's existing MongoDB database and presents the data in a visual and intuitive manner. To establish this connection, we worked closely with Omer and Aviram throughout the entire semester and were granted access to their database, ensuring seamless integration between our system and CareHub’s existing data infrastructure.

We chose to develop a web-based application primarily for its accessibility and ease of use. A browser-based platform allows Michael to access the system from any device with an internet connection, without the need for installation or platform-specific support. This also enables us to deploy updates quickly and maintain compatibility with CareHub's cloud-based architecture.

The frontend of our platform was built using React, JavaScript, and HTML, with CSS for styling. Initially, we explored using Tableau to create the visualizations. However, we quickly realized that in order to fully tailor the design and data presentation to Michael’s specific needs—and to ensure tight synchronization with the underlying data from CareHub—it would be more effective to build the visual components ourselves through code.

We chose React as our main framework because it offers a highly flexible and component-based architecture, which allowed us to build reusable, responsive, and dynamic interfaces. Combined with JavaScript, HTML, and CSS, this stack enabled us to design an intuitive and accessible user experience, with full control over the layout, interactivity, and integration with the database. This approach ensured that the final product was not only functionally robust, but also visually adapted to Michael’s daily use and feedback.

A diagram of a software development process

AI-generated content may be incorrect.

**Figure 2: system's architecture**

**Frontend Development**

The frontend was built using React, implementing the following principles:

* **Responsive Design**: We ensured the system is accessible and displayed optimally across various devices, including desktops, tablets, and smartphones, by utilizing **CSS media queries** and other responsive CSS properties.
* **Visual Data Presentation**: We developed dedicated UI components to display data from the CareHub database, including:
  + **Heatmap**: A visual representation of Michael's daily status using a color-coded system, allowing for a quick overview of patterns.
  + **Interactive Daily Dashboard**: A central screen presenting detailed graphs and visualizations of data collected on a given day, such as Parkinson's status, mood, medication intake, nutrition, and activities.
  + **Charts and Graphs**: We implemented line graphs, and tabular displays to present various data points clearly and understandably.
* **State Management**: We leveraged React Hooks and built-in React mechanisms for application state management, which allowed for seamless communication between components and dynamic updates of visualizations based on new data.

### **Backend Development**

The backend of the platform was developed using **Node.js** with serverless API routes, enabling direct and secure communication with CareHub’s existing **MongoDB** database.

This approach allowed us to retrieve existing data without the need to build a new data entry system, while maintaining a clear separation between frontend and backend logic. We implemented several dedicated API routes, each responsible for retrieving a specific type of data (such as mood, medication, nutrition, sleep, and activities). Each route returns the data in a structured JSON format, ready to be rendered by the corresponding visualizations on the frontend.

The connection to the database is handled through a reusable helper module that manages access to MongoDB using environment variables. Additionally, a dedicated **CORS** configuration ensures that only our frontend interface can access the API, maintaining data security and controlled access.

Choosing a serverless architecture simplified the development process, offered flexibility for future expansion, and eliminated the need to maintain backend servers. This solution was particularly suitable for our project, which focused on a single user and required seamless integration with an existing system—without introducing technical overhead for the user.

# 6. Faced Challenges and Solutions

Developing our visualization platform involved navigating several challenges, each met with strategic solutions to ensure a robust and user-centric product.

## **6.1 Planning and Functional Challenges**

**1. Integration with Existing Data**

**Challenge:** Our primary challenge was seamlessly integrating with CareHub's existing application data. Initially, we wanted to avoid disrupting their application's operation by connecting directly to their database.

**Solution:** To address this, we requested the Excel file containing the data Omer and Aviram, the creators of the CareHub application, input into their MongoDB database. We first used this data as raw, static information for developing and testing our visualizations. Only after ensuring our solution was stable, we connect the platform directly to the MongoDB database. This approach allowed us to build our visualization platform as a complementary client-side tool, leveraging Michael's already logged information without needing to create a separate data entry process or duplicate data, and, crucially, without disrupting CareHub's operations.

**2. Data Availability and Visualization Completeness**

**Challenge:** One of the challenges we encountered was that Michael did not consistently log data for every day throughout the month. As a result, when we initially connected to the MongoDB database, the retrieved dataset was sparse and incomplete, leading to empty or underpopulated visualizations that did not reflect the intended functionality of the platform.

**Solution:** To address this, we supplemented the available data by using a CSV file containing the actual entries Michael had submitted and enriched it with fabricated data. This allowed us to simulate a more complete dataset, enabling the visualizations to render as intended and properly demonstrate their design, flow, and user experience during development and evaluation.

**3. Data Representation and Clarity**

**Challenge:** Transforming raw, numerical data into meaningful and easily digestible visual representations was crucial. The complexity of Parkinson's symptoms, medication schedules, and activity logs meant that poorly designed visualizations could be confusing or misleading. We needed to present a clear narrative of Michael's daily patterns.

**Solution:** We focused on developing a diverse set of intuitive and focused visualizations, including a heatmap for a general daily overview and detailed daily dashboards with specific graphs for Parkinson's status, mood, medication, nutrition, and activities. We prioritized clear labeling, appropriate scaling, and concise summaries to ensure even complex data points were easily interpretable by Michael.

**4. Asynchronous Collaboration and Documentation**

**Challenge:** Our team occasionally faced challenges with conflicting schedules, which made synchronous meetings difficult. This led to reliance on individual work and asynchronous collaboration, which sometimes resulted in inconsistencies in code implementation or integration issues.

**Solution:** We significantly improved our communication protocols through regular, brief check-ins and extensive use of asynchronous collaboration tools. We also maintained clear and comprehensive documentation for all code, design decisions, and database interactions. This ensured that despite varied working hours, the team remained aligned, and the development workflow remained smooth and efficient.

**5. Team Skills and Learning Curve**

**Challenge:** As a team, we had varying levels of experience with some of the chosen frontend technologies, particularly in creating advanced data visualizations with React, JavaScript, HTML, and CSS. This presented an initial learning curve that could potentially slow down development.

**Solution:** We addressed this by fostering a collaborative learning environment. We conducted focused learning sessions, utilized extensive online resources, and engaged in peer programming to rapidly build our collective skills. This approach enabled us to overcome technical hurdles efficiently and implement sophisticated visualization features.

## **6.2 Design and Aesthetics Challenges**

Developing our visualization platform involved navigating several design and aesthetic challenges, each met with targeted solutions to ensure an aesthetically pleasing, accessible, and user-friendly product.

**1. Navigating Between Simplicity and Visual Engagement**

**Challenge:** While simplicity was a clear design requirement due to Michael’s cognitive needs, we found it challenging to strike the right balance between creating a clean, distraction-free interface and ensuring the platform remained visually engaging and modern. At times, we were tempted to include animations, hover effects, or decorative elements that enhanced aesthetics—but quickly realized that they risked creating cognitive overload or detracting from usability.

**Solution:** We embraced a minimalist design approach focused on clarity, functionality, and accessibility. All visual elements were deliberately chosen to support understanding rather than aesthetics alone. We iteratively tested color palettes, layout density, and interactive elements to find a balance that felt modern yet cognitively lightweight, resulting in a clean dashboard and heatmap design that serve their function without overwhelming the user.

**2. Reconciling Color Accessibility with Multi-Variable Visualization**

**Challenge:** While choosing accessible and visually distinct color palettes is a standard design requirement, the real challenge emerged when we needed to represent multiple variables—such as several medications—within the same graph. We had to ensure each medication had a clearly distinguishable color, while still adhering to accessibility standards and maintaining a visually cohesive and professional design. This became particularly difficult as the number of medications grew and the visual space became limited.

**Solution:** To overcome this, we collaborated closely with Michael to select a custom palette of strong, high-contrast, non-pastel colors. For example, in the medication intake chart, each medication was assigned a unique and memorable color (e.g., Levopar in red, Requip in green, etc.). For the heatmap, we designed a warm gradient from red (worst) to green (best), with gray for missing data, ensuring immediate comprehension of daily status. In the line chart, we selected distinct colors for each metric (e.g., mood, Parkinson state, physical difficulty) to enhance readability across the timeline. This process required careful design judgment, testing, and iterations based on Michael’s feedback to reach the right balance between clarity, meaning, and aesthetics.

# 7. Results and Conclusions

## **7.1 Goals and Achievements**

The primary goal of our project was to develop a web-based visualization platform that would serve as a complementary tool to the CareHub application, specifically tailored to the needs of Michael, who lives with Parkinson's disease. Our objective was to transform Michael's personal data, recorded in CareHub (including medication intake, physical activity, meals, symptoms, and general well-being), into meaningful visual representations, presented in a clear and intuitive interface.

One of the main achievements was the successful development of a highly interactive and accessible visualization platform. We focused on creating an optimal user experience for Michael, taking into account his cognitive and physical limitations. The platform presents the data in various easy-to-understand visual formats, including:

* **Heatmap Calendar**: The entry point to the system, offering a visual overview of Michael's daily status using a color-coded system (from red to green), enabling quick identification of better and less favorable days. Days with no data are clearly marked in gray.
* **Daily Dashboard**: A central screen allowing for detailed data analysis for a selected day. The dashboard displays:
  + **Line Graph**: Presenting Parkinson's status, mood, and physical difficulty levels throughout the day, which helps in identifying patterns and correlations between these metrics.
  + **Stacked bar chart for Medication Intake Summary Chart**: Detailing medication types and intake times, with a unique color palette for each medication.
  + **Bar chart for Protein Consumption Breakdown**: Including accompanying nutritional values.
  + **Gantt chart for Daily Activities View**: Categorized by type (cognitive, sport, domestic).

Another significant achievement was the ability to securely connect to CareHub's existing MongoDB database, leveraging Michael's sensitive data. By working closely with the CareHub team, we successfully accessed and presented the data without disrupting their operations or compromising data security. The emphasis on a personalized and interactive interface, built in direct collaboration with Michael and through continuous feedback, ensures that the platform is not only technically effective but also customized meaningful and supportive in Michael's daily management of Parkinson's disease.

## **7.2 Conclusions**

Looking back at the development process, we can conclude that the decision to develop a web-based visualization platform, complementing the CareHub application, was highly appropriate given our project's specific constraints and goals. This approach allowed us to leverage existing data while focusing our resources on creating meaningful visual representations rather than building a new data entry system from scratch.

Our choice of React, JavaScript, HTML, and CSS for the frontend, alongside integration with the MongoDB database (via the CareHub team's provided data and later direct connection), provided a robust and flexible foundation for the platform's development. This technology stack allowed for the creation of a dynamic and intuitive user interface capable of presenting complex health data in an easily digestible format. Despite the challenges we encountered, such as ensuring data entry consistency from a single user (Michael) and meticulously customizing design elements for his specific needs, the platform met our core goals and proved to be highly functional and user-friendly.

The continuous and direct feedback from Michael, our primary user, was instrumental in confirming the platform's success. His input ensured that the design was intuitive, the visualizations were clear, and the features were genuinely useful in helping him understand patterns in his daily routine related to Parkinson's disease. This highly user-centered design approach was critical in ensuring the platform's usability and personal relevance.

While the current version of the visualization platform is functional and effectively serves Michael's needs, there are several areas for future improvement. One potential enhancement includes adding the ability to compare graphs from different periods. This means that not only can Michael select a single day to view his data, but he can also simultaneously select additional days or periods and view the presentation of the graphs overlaid on top of each other to observe changes and differences more clearly. Additionally, perhaps in the future, our interface could serve not only Michael but also other Parkinson's patients, thereby expanding its impact. These future developments would aim to further enhance the platform's value in supporting the ongoing management of Parkinson's disease.

## **7.3 Limitations and Future Work**

Looking ahead, there are several ways our visualization platform can be improved and expanded, which future developers or students might continue working on. Although we will not continue developing the platform directly, 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 platform can continue to help Michael, and potentially, other Parkinson's patients in the future.

One of the first areas for improvement is data comparison and parallel visualization capabilities. Currently, the platform allows Michael to view data for a specific day. A significant enhancement would be to add the option to compare graphs from different days or periods, so Michael could view the presentation of graphs (for example, Parkinson's state or mood) overlaid on top of each other. This would allow him and his caregivers to identify changes and patterns over time more clearly and immediately, enabling a deeper analysis of the data and a better understanding of the impact of various factors.

Another important area is expanding the target audience. While the platform was designed and specifically tailored for Michael, it has the potential to be useful for other Parkinson's patients as well. In the future, developers could adapt the interface and visualizations to more general requirements, thereby making the platform accessible to a wider audience of patients. Furthermore, it might be beneficial to consider adding more extensive personalization options so that each user can adapt the platform to their individual needs.

Additionally, continued user testing will be important in the future. By testing the platform with a broader group of patients and/or caregivers, valuable feedback can be obtained to help improve the platform. This ongoing testing and feedback loop will help ensure that the platform meets the needs of all users.

# 8. Evaluation / Verification Plan

## **8.1 Testing Plan**

As part of the system testing process, we conducted a series of tests to ensure its functionality, usability, and the accurate presentation of data. These tests focused both on the behavior of individual components and on how the visualizations responded to different data scenarios.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Test Subject** | **Test Headline** | **Expected Results** |
| 1 | Heatmap Calendar | Load calendar with recorded data | Heatmap displays all relevant days with correct triangle colors for mood and Parkinson’s state. |
| 2 | Date Selection | Click on a specific date | System redirects to the correct daily dashboard screen |
| 3 | Daily Line Graph | Display mood, Parkinson state, and physical difficulty | Line chart shows all selected parameters accurately, aligned to their respective timestamps |
| 4 | Medication Intake Graph | Hover over medication bar | Tooltip shows correct medication name, dosage, and time of intake |
| 5 | Protein Intake Graph | Hover over protein bar | Tooltip displays full nutritional details – protein, fats, carbohydrates, fiber, and meal name |
| 6 | Activities Timeline | Hover over activity block | Tooltip shows activity name, start time, duration, and intensity level |
| 7 | Average Score Summary | Show averages at the bottom of the line graph | Average scores match reported values for mood, Parkinson state, and physical difficulty |
| 8 | Graph Visibility Toggle | Click toggle buttons for mood / Parkinson / difficulty | Only selected lines are visible on the chart; update is immediate and responsive |
| 9 | Screen Responsiveness | Resize window or test on mobile device | Layout adapts fluidly to screen size without content loss or overlapping |
| 10 | Data Sync | Load data reported in CareHub | Data from CareHub appears correctly in the visualization module and is up to date |
| 11 | Page Navigation | Navigate between screens (calendar ↔ daily dashboard) | Navigation between pages is fast, smooth, and does not interrupt user flow or context |
| 12 | Graph Switching | Switch between graphs within the daily dashboard | Clicking graph sections (line graph, medication, protein, activities) smoothly scrolls or switches to the relevant section with no lag or UI glitches. |

## **8.2 User evaluation**

To ensure our visualization platform met its intended goals, we held multiple meetings with Michael, our primary user. In each session, we jointly reviewed the selected graphs and the way data was presented, ensuring they were personally suitable for Michael. We carefully confirmed that textual explanations were clear and sufficient, that colors weren't too prominent or distracting, that headings were an appropriate size, and that every visual element was clear and specifically tailored for him. These meetings provided a crucial opportunity to validate the platform's functionality and usability, aligning it closely with Michael's unique needs and cognitive capabilities.

Following Michael's feedback, we moved into the refinement stage. We made necessary adjustments to improve the platform's usability and the clarity of its visualizations based on the input gathered. For example, Michael emphasized the importance of assigning a unique color to each medication for easier identification, having all visualizations accessible from a single dashboard view, and ensuring that when navigating back after selecting a past date, the system would automatically return to the current date. This iterative process allowed us to address usability concerns, enhance key visual components, and ensure the system delivered a seamless and user-friendly experience tailored to his needs.

To formally evaluate the application's usability, we provided a comprehensive questionnaire to a group of 10 diverse participants. The group included 6 males and 4 females, with an average age of 29.78. While some participants had minimal exposure to digital systems, others—particularly those studying software engineering—had relevant experience in interpreting data visualizations.

The evaluation included the standard System Usability Scale (SUS) to assess overall usability, which resulted in a strong average score of 78.5. In addition, the questionnaire featured a dedicated section for evaluating each individual visualization within the platform, providing deeper insights into the user experience of specific components.

Participants were asked to rate eight specific statements per visualization on a 1–7 scale, addressing factors such as enjoyment, complexity, ease of use, mental effort, clarity, frustration, effectiveness, and ease of learning. Additionally, they were invited to share open-ended feedback on strengths and weaknesses they observed. This two-part evaluation allowed us to gain both quantitative and qualitative insights into user satisfaction and the clarity of our visual design.

These feedback-driven refinements and usability evaluations ensured that the final version of our visualization platform is a reliable, accessible, and impactful tool tailored to Michael's needs, empowering him with a deeper understanding of his Parkinson's journey.

# 9. User Documentation

## **9.1 User Guide**

### **9.1.1 Operating Instructions – Visualization Module**

The visualization module is a browser-based extension of the CareHub application. It is accessible from any modern web browser on a desktop or table and requires no installation. The module displays data previously logged in CareHub, transforming it into interactive graphs, timelines, and personalized visuals that help Michael identify daily patterns.

**1.View Data Button**

The **“View Data”** button serves as the entry point to the visualization module of the CareHub system. Clicking this button redirects the user to an interactive heatmap screen that provides a monthly overview of previously reported physical and emotional data.

The module is accessible at <https://park-vis.vercel.app/>

**How to Use:**

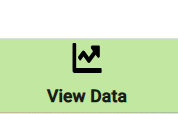
* Log in to the CareHub system using your registered email and password.
* On the main screen, locate the green “View Data” button with the chart icon.
* Click “View Data” to access the Heatmap Calendar screen.

**What Happens Next:**

* You will be directed to the Heatmap Calendar, which displays your Parkinson’s state and mood for each day of the selected month.
* Clicking on a specific date will open the Daily Dashboard, where you can explore detailed graphs for that day.

**Important Notes:**

* Make sure your CareHub data is up to date before accessing the heatmap.
* This module is view-only – it does not support data input or editing.
* If no data is available, the date squares on the calendar will appear in gray.



**2. Heatmap Calendar Screen**

The Heatmap Calendar is the first screen the user sees after clicking the “View Data” button. It presents a full-month calendar where each day is color-coded based on the user’s reported Parkinson’s condition, mood, and physical difficulty.  
 Each date square is divided into **three triangular segments**:

* The top triangle represents the Parkinson state (neurological symptoms).
* The bottom-left triangle represents the user’s mood (emotional state).
* The bottom-right triangle represents the user’s physical difficulty (fatigue, mobility effort).

The color scale ranges from **green (1 – best)** to **red (5 – worst)**. If no data was logged for a given aspect on a specific day, the corresponding triangle appears **gray**.

**How to Use:**

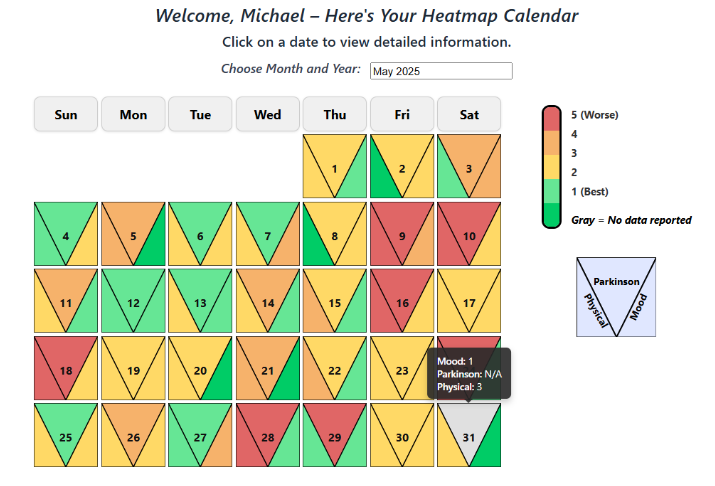
* View your monthly overview at a glance through color-coded indicators.
* Click on any day to access a detailed daily dashboard with graphs and summaries related to that date.

**Additional Features:**

* Use the “Choose Month and Year” field to navigate between months.
* The color legend on the right helps interpret the severity level of each triangle.
* A small key at the bottom-right corner of the screen clarifies which triangle corresponds to which metric (Parkinson, Mood, Physical).

**Purpose:**

To help users easily identify trends, fluctuations, or days that require closer attention—based on simple visual cues—without needing to manually review each individual entry.



**Figure 3 -Heatmap**

**3. Daily Analysis Graph Screen**

The Daily Analysis screen provides a visual timeline of the user’s condition throughout a selected day. This line chart displays fluctuations in:

* Mood (blue)
* Parkinson’s state (red)
* Physical difficulty (green)

The Y-axis represents a score scale from 1 (best) to 5 (worst), allowing users to assess the severity of each parameter over time.

Above the graph, the user can choose which variable to display using toggle buttons. All three values can be shown together or individually.

Event icons (e.g., medication, nutrition, waking up, sleeping, activity) are displayed along the timeline. Hovering over an icon reveals a tooltip describing the event and its exact time (e.g., "Ping Pong" as shown in the example).

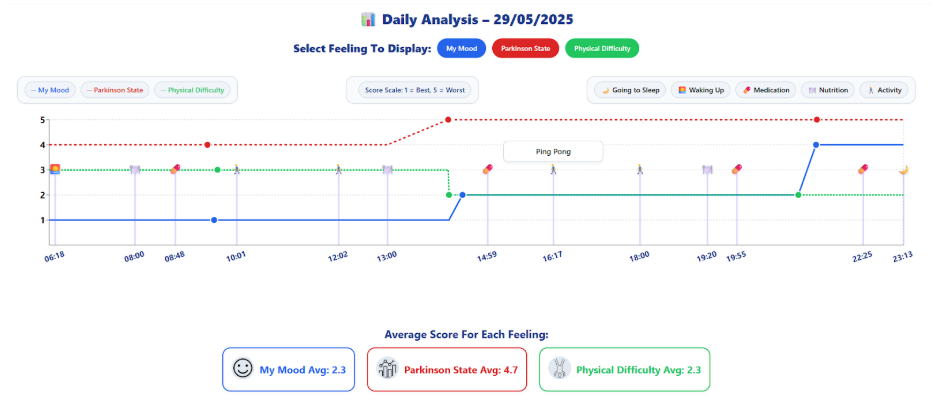
At the bottom of the screen, the system displays the average score for each variable recorded that day.

**How to Use:**

1. After selecting a day from the Heatmap Calendar, the Daily Analysis screen is loaded automatically.
2. Use the filter buttons to toggle between mood, Parkinson’s condition, and physical effort.
3. Hover over any icon for contextual information about logged events.
4. Review the average values shown below for a summary of the day.

**Purpose:**

This screen helps users visually understand how their condition changed throughout the day and how specific events (such as meals, activities, or medications) may have influenced their well-being.



**Figure 4 -Daily Analysis chart**

**4. Medication Intake Summary Screen**

The Medication Intake Summary screen presents a clear bar chart of all medications taken during the selected day. Each bar represents a specific time of intake, and its segments are color-coded by medication type (e.g., Sinemet, Azilect, Requip).

Hovering over a bar reveals a tooltip showing detailed information, including:

* Medication name
* Exact dosage (in pills)
* Time of intake

The chart allows multiple medications to appear within the same bar if they were taken at the same time.

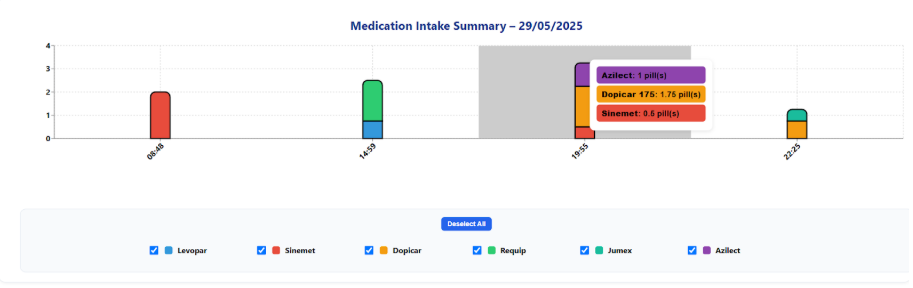
At the bottom of the screen, a legend with checkboxes lets users toggle the visibility of specific medications in the chart.

**How to Use:**

1. Scroll to the Medication Intake section in the Daily Dashboard.
2. Hover over any bar to see detailed dosage and timing.
3. Use the checkbox filters below to show/hide specific medications.

**Purpose:**

This visualization helps users and caregivers monitor medication adherence, recognize patterns in timing and dosage, and identify any irregularities in the treatment schedule.



**Figure 5 -Medication chart**

**5. Protein Intake Summary Screen**

The Protein Intake Summary screen displays a bar chart representing the user's protein consumption throughout the selected day. Each bar corresponds to a reported meal and shows the amount of protein in grams.

When hovering over a bar, a tooltip appears with detailed nutritional information, including:

* Protein (g)
* Fats (g)
* Carbohydrates (g)
* Dietary fiber (g)
* Meal description (e.g., “Omelette with hard-boiled egg”)

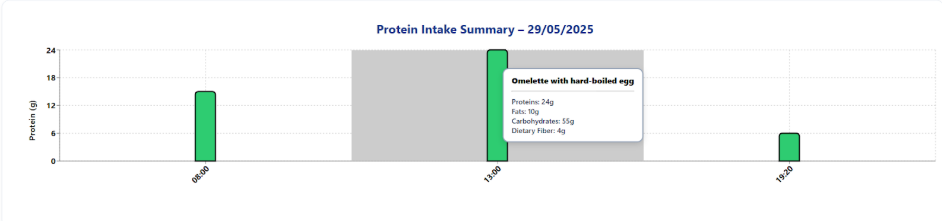
Meals are visually distributed along the timeline based on their recorded time of consumption.

**How to Use:**

1. Scroll down to the Protein Intake section within the Daily Dashboard.
2. Hover over any green bar to view full meal details.
3. Use this information to assess meal content and timing.

**Purpose:**

This screen helps users understand their protein intake across the day, evaluate the nutritional content of their meals, and observe how timing and quantity may relate to symptom control or medication effectiveness.



**Figure 6 -Protein intake chart**

**7. Daily Activities Timeline Screen**

The Daily Activities Timeline displays all recorded activities for the selected day, categorized by type:

* Sport (blue)
* Cognitive (green
* Household (yellow)

Each bar on the timeline represents an activity's duration and time of day. Activities are arranged along the horizontal time axis, with color-coded rows for each category.

Hovering over a bar reveals a tooltip with key information:

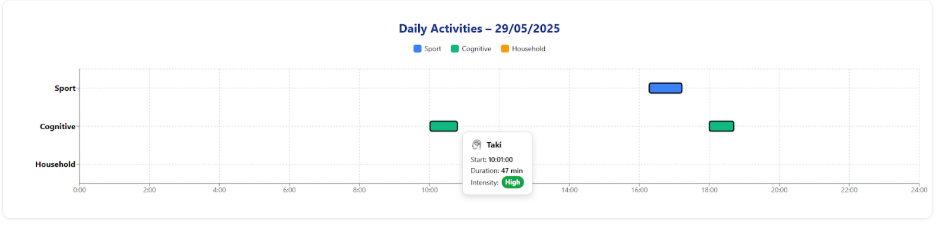
* Activity name
* Start time
* Duration
* Intensity level (e.g., "High")

**How to Use:**

1. Scroll to the Daily Activities section within the Daily Dashboard.
2. Hover over any activity bar to view full details.
3. Compare activities by time, intensity, and category.

**Purpose:**

This screen allows users to reflect on their daily routine and activity patterns, recognize how different activities may affect their condition, and evaluate their overall energy balance throughout the day.



**Figure 7 -Daily Activity chart**

### **9.1.2 Maintenance Guide**

**Installation:**

Make sure the following tools are installed:

* Node.js – (version 16 or later).
* npm (comes with Node.js).
* Vercel CLI (used for local development).

To install Node.js, visit:

* <https://nodejs.org> and download the LST version.
* To install Vercel globally, run: npm install -g vercel
* To install Vercel globally, run in the terminal: npm install -g vercel

To confirm successful installation, run the following commands in the terminal:

* node -v: It should return a version like v16.x or higher.
* npm -v: It should return a version like 8.x or higher.
* vercel -v: It should return the installed Vercel CLI version.

Important Notes:

* Please use an up-to-date version of Chrome, Firefox, or Edge for best compatibility.
* A stable internet connection is required to ensure smooth loading of graphs.
* This module is read-only and does not support data entry – all information is synced directly from CareHub.
* The tool is customized specifically for Michael and is not designed for multi-user or public access.

Development Tools:

* Visual Studio Code – used as the main code editor for writing and managing the project files.
* React– the primary frontend library used to build the user interface and interactive components.
* Tailwind CSS – a utility-first CSS framework used to design responsive and clean UI layouts.
* JavaScript– the main programming language used to implement the logic and components.
* Vercel CLI – used to run and test the app locally, and to optionally deploy it online.
* CSV File – used as the format for importing and analyzing patient data.
* Node.js + npm – used to manage dependencies and scripts for running the app locally.

Running Instructions:

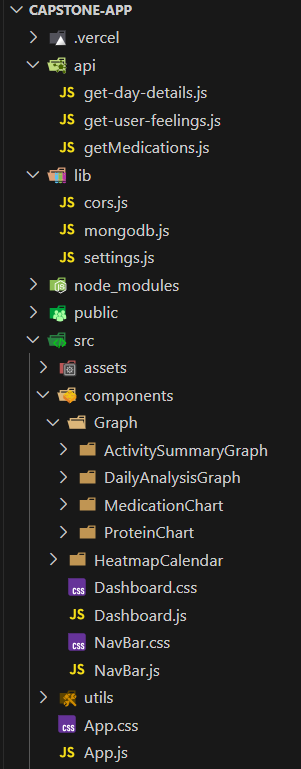
Steps to Run the Application Locally:

* Open the Terminal in Visual Studio Code:
* Launch Visual Studio Code.
* Open the project folder "FinalProject\_Parkinson-s\_Visualizaton" in VS Code.
* Access the terminal by clicking on Terminal > New Terminal in the top menu or by pressing Ctrl + ~ (Windows/Linux) or Cmd + ~ (Mac).
* Make sure you're in the root directory of the project:
* 

Install Dependencies:

* In the terminal, run the following command to install all necessary packages: 
* Once all dependencies are installed, run the following command to launch the app locally using Vercel: 
* After running the above command, the terminal will display a local URL (usually something like http://localhost:3000).
* Open this URL in your browser to view the application

Project Folder Structure:



The project is organized into four main directories: public, src, api, and lib.

The public folder contains static files and data resources. Inside public/data, there is a single file: daily\_data.csv, which contains the patient's daily records.

The src folder contains the core source code of the application. Under src/components, you'll find the reusable UI components.

The api folder contains three serverless API routes that communicate with the MongoDB database. These routes are used to retrieve data for specific use cases in the application.

The lib folder contains helper modules for backend functionality.

The src/components/HeatmapCalendar folder contains the logic and rendering for the interactive calendar heatmap, which allows users to select and view data by date.

The files src/components/NavBar.js and NavBar.css define the top navigation bar.

The files src/components/Dashboard.js and Dashboard.css handle the main dashboard screen shown after selecting a day.

The src/assets folder contains all the icons and images used throughout the dashboard.

The src/components/Graph folder contains four subfolders, each dedicated to a specific chart:

* DailyAnalysisGraph.
* MedicationChart.
* ProteinChart.
* ActivitySummaryGraph.

The src/utils folder contains the loadCSV.js file, which is responsible for loading the CSV data used by the dashboard and heatmap calendar components.

App.js:

This is the main entry point of the application. It handles the routing between the two main views: the Heatmap Calendar and the Dashboard. Upon mounting, it loads user data from a CSV file (daily\_data.csv) using the loadCSVData utility function. The data is then preprocessed to normalize date formats and extract relevant state values (mood, Parkinson’s state, physical difficulty) based on the Type field. The cleaned data is passed down as props to the respective components via React Router.

**HeatmapCalendar Folder:**

HeatmapCalendar.js:

This component displays a monthly heatmap calendar where each day is divided into three triangles representing My Mood, Parkinson’s State, and Physical Difficulty. For each day, the average score is calculated separately for each of the three types based on the user's entries. The component manages the selected month and year, handles missing or future data, and controls the display of messages and modals.

CalendarGrid.js:

This component generates the monthly grid of days in the heatmap calendar. Each day is displayed as a clickable square containing a triangle-based visualization of the daily averages. If a day has no data or is in the future, a modal message is triggered instead of navigation. Clicking a valid day redirects the user to the detailed dashboard view for that date.

[CalendarHeader.js](http://calendarheader.js):

This component renders the weekday headers (Sunday to Saturday) at the top of the heatmap calendar grid. It helps structure the calendar layout for visual clarity.

FutureDateModal.js:

This component displays a modal when the user clicks on a future date in the heatmap calendar. It informs the user that the selected date hasn't occurred yet and prompts them to choose a past or current date instead.

NoDataMessage.js:

This component displays a short message when there is no data available for the selected month. It appears above the calendar and prompts the user to choose a different month.

NoDataModal.js:

This component shows a modal when the user clicks on a day that has no recorded data. It displays the selected date and asks the user to choose a different day with available data.

HeatmapCalendarHeader.js:

This component displays the header section of the heatmap calendar. It shows a personalized welcome message, a short instruction, and a picker for selecting month and year.

HeatmapCalendarFooter.js:

This component renders a simple footer at the bottom of the heatmap calendar page, displaying the current year and a copyright.

MonthYearPicker.js:

This component provides a date picker for selecting the month and year of the heatmap calendar. It uses a styled version of the react-datepicker library, limited to past and current dates only.

ModalsContainer.js:

This component manages the conditional display of modal dialogs. It renders a message when the user clicks on either a future date or a day without data, based on the relevant state flags.

NavBarHeatMap.js:

This component displays a simple navigation bar at the top of the heatmap calendar page. It contains a title and helps provide visual structure to the layout.

HeatmapLegend.js:

This component displays a vertical color scale that explains the meaning of the heatmap colors, from 1 (best) to 5 (worst). It also includes a note indicating that gray represents missing data.

LegendSampleBox.js:

This component displays a small sample square that visually explains how each triangle in a heatmap cell represents a different category: Parkinson, Physical, and Mood. It uses an SVG graphic with labels to illustrate the layout clearly.

TriangularHeatmapCell.js:

This component renders a single day in the heatmap calendar as a square divided into three colored triangles, each representing a different state: Parkinson State, Physical Difficulty, and My Mood. On hover, it shows a tooltip with the exact values. The triangle colors are dynamically set based on the provided data.

useHeatmapCalendarState.js:

This custom React hook manages the state and logic for the heatmap calendar. It tracks the selected month and year, handles modal visibility for future dates and missing data, and calculates daily average scores for mood, Parkinson’s state, and physical difficulty. The data is automatically grouped and averaged by date, and the selected month is persisted using localStorage.

**Dashboard.js:**

This component displays the main dashboard view for a selected date. It renders four graphs: DailyAnalysisGraph, MedicationChart, ProteinChart, and ActivitySummaryGraph. It receives data entries and the selected date from the URL, and it updates localStorage to preserve the last viewed month.

NavBar.js:

This component renders the top navigation bar for the dashboard. It includes a logo, a button to return to the heatmap (while preserving the selected month), and four scrollable section buttons.

**Graph Folder:**

**DailyAnalysisGraph Folder:**

DailyAnalysisGraph.js

This component displays a detailed daily timeline graph for a selected date. It shows the user's My Mood, Parkinson's State, and Physical Difficulty over time. The user can toggle each line on or off and view a legend and summary of daily averages. The graph data is filtered based on visibility settings and rendered accordingly.

VerticalLinesWithIcons.js:

This component renders vertical lines and action icons on the timeline graph to mark specific user activities. When hovering over an icon, a tooltip appears with detailed information. It helps visually correlate actions with changes in My Mood, Parkinson's State, and Physical Difficulty.

useFilteredEntries.js:

This custom hook processes and filters user entries for a selected date. It extracts values related to My Mood, Parkinson's State, and Physical Difficulty, maps actions to icons, and prepares the data structure for the timeline graph. It returns the main graph data, a timeline of user actions, and the timestamp of the last action.

ToggleButtons.js:

This component displays toggle buttons for controlling the visibility of the three graph lines: My Mood, Parkinson's State, and Physical Difficulty. If no data is available for a certain type, it shows a warning message instead of a button.

mapReportToIcon.js:

This utility function maps each report type and activity to a specific icon identifier. It distinguishes between activities, medication, nutrition, and sleep-related entries based on their text content and type.

LegendSection.js:

This component displays the legend for the daily analysis graph. It shows color-coded labels for My Mood, Parkinson's State, and Physical Difficulty, along with a score scale explanation and icons for common activities like sleep, medication, and nutrition.

GraphContainer.js:

This component renders the main visual timeline chart using Recharts. It plots the selected states: My Mood, Parkinson's State, and Physical Difficulty as colored lines. It also includes vertical markers for user actions, custom tick labels, and dot markers for mood entries.

FeelingDots.js:

This component renders interactive dots on the graph to highlight individual entries for My Mood, Parkinson's State, and Physical Difficulty. When hovering over the dots, a tooltip displays the exact value, time, and type. It supports overlapping values by spacing the dots for better readability.

CustomXAxisWithTimes.js:

This component customizes the X-axis labels on the timeline graph. It displays the times of day at an angle for better readability, using a formatted time string for each tick.

AveragesDisplay.js:

This component displays the daily average scores for My Mood, Parkinson's State, and Physical Difficulty. Each average is shown in a styled box with a matching icon and color, helping users quickly interpret their overall condition for the day.

utils.js:

This utility file provides helper functions for time conversion, date formatting, and timeline construction. It includes logic to generate the full timeline, map actions to icons, and build structured data for the graph, including values for My Mood, Parkinson's State, and Physical Difficulty.

**MedicationChart Folder:**

MedicationChart.js:

This component displays a bar chart summarizing the user's medication intake for a selected date. It groups medications by time, extracts dosage information, and allows toggling pill categories on and off. A legend helps users filter visible pill types.

ChartCore.js:

This component filters and groups medications by time and type, applies custom colors and rounded corners to highlight key pills, and dynamically adjusts the Y-axis range based on dosage totals. It supports pill visibility filtering and uses a custom tooltip for detailed representations .

ChartDataUtils.js:

This utility file provides a helper function to sort medication entries by time. It ensures that pills taken earlier in the day appear first in the chart.

PillTypes.js:

This constants file defines the classification of pill types and their color mappings. It groups individual pill names under broader categories and assigns a specific color to each category for consistent use across the medication chart.

LegendPills.js:

This component renders a checkbox-based legend that lets users toggle the visibility of each pill type in the medication chart. Each item is displayed with its corresponding color and label.

MedicationTooltip.js:

This component displays a customized tooltip for the medication chart. When hovering over a bar, it shows all pills taken at that time, including their names, dosages, and category colors.

useMedicationData.js:

This custom React hook processes medication entries for a given date. It filters and groups the data by time and pill type, extracts dosage amounts from notes, and identifies which pill types were used. The processed data is returned for use in the medication chart.

**ProteinChart Folder:**

ProteinChart.js:

This component displays a bar chart showing the user's protein intake for the selected date. Each bar represents a meal with its corresponding protein amount. Bars include tooltips and a black border to highlight individual values.

ProteinTooltip.js:

This component renders a custom tooltip for the protein chart. It shows the food name and additional notes when hovering over a bar, with a styled card-like design.

useProteinData.js:

This custom hook filters and transforms nutrition entries for a selected date. It extracts protein values from the notes field and structures the data for use in the chart.

**ActivitySummaryGraph Folder:**

ActivitySummaryGraph.js:

This component displays a visual summary of the user's daily activities. It uses a custom scatter chart to show when different types of activities occurred throughout the day, divided into three categories: Sport, Cognitive, and Household. The graph includes tooltips and a custom legend.

ActivityTooltip.js:

This component displays a custom tooltip for activity data. It shows the activity's name, start time, duration, category icon, and intensity level with a color indicator.

CustomBars.js:

This component renders custom bars on the activity graph, representing each activity’s start time, duration, and category. Bars are color-coded based on activity type (Sport, Cognitive, Household) and support mouse hover interactions to trigger tooltips.

LegendBox.js:

This component renders a simple color-coded legend for the activity graph. Each color box represents a different activity category: Sport, Cognitive, and Household. It helps users understand the color coding used in the chart.

useActivityData.js:

This custom hook filters and processes activity data for a specific date. It converts time to minutes, categorizes activities into Sport, Cognitive, or Household, and prepares structured objects used for rendering the graph.

**Utils Folder:**

loadCSV.js:

This utility function loads and parses a CSV file using the PapaParse library. It returns the CSV data as a JavaScript object array, enabling other components to easily access and use structured data.

**data Folder:**

The data folder contains the daily\_data.csv file, which serves as the primary data source for the entire dashboard. This CSV file includes user-reported entries such as mood, physical condition, Parkinson’s state, nutrition, sleep, medication, and activities. It is loaded at runtime using the loadCSVData utility.

**Lib Folder:**

mongodb.js:

This file handles the connection to the MongoDB database. It reads the connection string and database name from environment variables and exports a connectToDatabase function.

cors.js:

This file configures Cross-Origin Resource Sharing (CORS) for the API routes. It allows requests from a specific frontend domain (defined in settings.js) and supports multiple HTTP methods such as GET, POST, DELETE, etc. The cors object is imported into each API route to enforce secure communication between frontend and backend.

settings.js:

This configuration file defines the allowed frontend URL used by the CORS middleware. It exports a single constant URL, which points to the deployed version of the application (on Vercel).

**api Folder:**

get-day-details.js:

This API route retrieves all user data for a specific date provided in the query string). It fetches documents from five MongoDB collections: feelings, activities, nutritions, medicines, and sleepwakes, filtering them by the full day’s time range.

get-user-feelings.js:

This API route fetches all user feeling records from the feelings collection in the MongoDB database. It supports only GET requests and returns all entries along with the total count.

getMedications.js:

This API route retrieves all medication records from the medicines collection. It returns a JSON response with the full list of medications and the number of entries.

# References

[1] Langeskov-Christensen, M., Franzén, E., Grøndahl Hvid, L., & Dalgas, U. (2024). *Exercise as medicine in Parkinson's disease*. *Journal of Neurology, Neurosurgery & Psychiatry*, 95(11), 1077–1088.

[2] Smolensky, L., Amondikar, N., Crawford, K., Neu, S., Kopil, C., Daeschler, M., ... & Tanner, C. M. (2020). *The Fox Insight study: Collecting online, longitudinal patient-reported outcomes from a large cohort of individuals with Parkinson’s disease*.

[3] So, J., Gupta, A. S., Yang, F. X., & Gajos, K. Z. (2019). *"It's better to be grounded in reality": A speculative exploration of patient-centered digital phenotyping for neurological conditions*. In *CHI Conference on Human Factors in Computing Systems* (pp. 1–13). ACM.

[4] Ferreira-Brito, F., Branco, D., Móteiro, M., Ferreira, J. J., & Guerreiro, T. (2024). *Clinical dashboards for Parkinson's disease monitoring: A systematic review*. *Journal of Scientific and Clinical Medicine*, 168(1), 33–47.

[5] Galmarini, M., Schneider, F., & Smith, T. (2024). *The effectiveness of visual-based interventions on health literacy in clinical populations: A systematic review and meta-analysis*. *Journal of Health Communication*, 29(1), 45–62.

[6] Bloem, B. R., Okun, M. S., & Klein, C. (2021). *Parkinson's disease*. *The Lancet*, 397(10291), 2284–2303.

[7] 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's disease*. *Nature Reviews Neurology*, 13(11), 689–703.

[8] Navarta-Sánchez, M. V., Palmar-Santos, A., Pedraz-Marcos, A., Reidy, C., Soilemezi, D., Haahr, A., ... & Portillo, M. C. (2023). *Perspectives of people with Parkinson's disease and family carers about disease management in community settings: A cross-country qualitative study*. *Journal of Clinical Nursing*, 32(15–16), 5201–5218.

[9] Hellqvist, C., Dizdar Segrell, N., Hagell, P., Berterö, C., & Sund-Levander, M. (2018). *Improving self-management for persons with Parkinson's disease through education focusing on management of daily life: Patients’ and relatives’ experience of the Swedish National Parkinson School*. *Journal of Clinical Nursing*, 27(19–20), 3719–3728.

[18] Serrano, J. A., Thoms, A., Weber, P., & SENSE-PARK Consortium. (2014). *Patients initiated timeline marking of events in Parkinson’s disease*. In *Lecture Notes in Computer Science* (Vol. 8534, pp. 325–334). Springer.

[19] Kandel, J., Duppen, C., Zhang, Q., Jiang, H., Angelopoulos, A., Neall, A., ... & Szafir, D. A. (2024). *PD-Insighter: A visual analytics system to monitor daily actions for Parkinson’s disease treatment*.

[20] Dias, S. B., Rodrigues, P. M., Figueiredo, M., Ferreira, J. J., & Guerreiro, T. (2019). *DataPark: Visualizing physical activity patterns in Parkinson’s patients*. *Proceedings of CHI 2019*, 1–10.

[21] Scripps Research Translational Institute. (2019). *Clinician dashboard for Parkinson's disease: Visualizing medication intake, ON/OFF diary, and symptom severity ratings*. *npj Digital Medicine*, 2(95), 1–5.