

Physioplan-Assignment

Project Documentation

1. UI Components

This project uses React to create a dropdown component for assigning exercises.

The primary components and functionality are:

- **Dropdown Component:** The main component that manages the weekly exercise plan, including daily frequency and therapist notes.
- **State Management:** Utilizes React's `useState` and `useEffect` hooks to manage states such as selected exercises, daily frequency, and therapist notes.
- **Week Day Selection:** Allows selection of days (Sunday through Saturday) with buttons representing each day.
- **Body Part Categories:** Displays exercise categories on hover, allowing users to add exercises to the selected day.
- **Exercise Management:** Allows exercises to be added, duplicated, or deleted for each day, with options to save or clear daily data.
- **Therapist Notes:** Textarea for adding therapist notes specific to the selected day.

2. Backend API Structure

The backend is a RESTful API that manages exercise plans and categories. Key endpoints include:

- **GET /exercise:** Fetches all exercise categories and their respective exercises. Used to populate the body part categories in the UI.
- **GET /plans:** The `/plans` endpoint fetches all saved weekly exercise plans, and if none exist, returns a blank weekly plan template for users to view or edit.
- **GET /plan/{day}:** Fetches the exercise plan for a specified day (e.g., Monday). Returns the list of exercises, frequency, and any notes.
- **POST /plan/{day}:** Saves the updated exercise plan for the specified day. Accepts JSON data containing exercises, frequency, and notes.
- **PUT /plan/{day}/clear:** Clears the exercise plan for a specified day, setting frequency and notes to default values.

3. Running the Project

To run the project locally:

1. Install Dependencies:

- In each directory, install dependencies if you haven't done so:

cd backend

npm install

cd physioplan (frontend)

npm install

2. Start Backend and Frontend Concurrently:

- Use the command below to start both backend and frontend simultaneously:

npm run dev

- Ensure both the frontend and backend folders are configured to start with this command, typically through a tool like concurrently.

3. Database:

- MongoDB is used to persist exercise plan data. Ensure MongoDB is running and that your backend is connected to the MongoDB database to store the exercise plans, frequency data, and therapist notes.

4. Accessing the Application:

- Open <http://localhost:3000> in your browser to view and interact with the UI components.

Notes: Both frontend and backend should be running concurrently.

Note:-

- I implemented a system to drag and reorder items, but it impacted the interactivity and smoothness of the application, so I decided to remove it.
- The site is **responsive**, adapting well to different screen sizes for an optimized user experience on various devices.