

Dokumentacja Aplikacji Internetowej „Be Better”

SPIS TREŚCI:

- 1.1.** Opis ogólny aplikacji
- 1.2.** Założenia i cele
- 2.1.** Struktura katalogów i plików
- 2.2.** Szczegółowa instrukcja instalacji
- 3.1.** Architektura aplikacji
- 3.2.** Opis klas
- 3.3.** Opis funkcji
- 3.4.** Struktura zarządzania danymi
- 4.1.** Dalsze ambicje i możliwości rozszerzeń

BeBetter

1.1. Opis ogólny aplikacji

"Be Better" to przede wszystkim aplikacja stworzona z myślą o osobach pragnących osiągać samorozwój i podnosić jakość swojego życia. Jej intuicyjny i przyjazny użytkownikowi interfejs został zaprojektowany tak, aby umożliwiać bezproblemowe wdrażanie osobistych celów w życie. Dzięki prostocie obsługi, użytkownicy mogą łatwo definiować i monitorować swoje cele oraz poznawać nowe ścieżki rozwoju.

Interfejs aplikacji dostarcza narzędzi niezbędnych do skutecznego zarządzania własnym postępem. Umożliwia użytkownikom kontrolowanie swojego rozwoju w czasie rzeczywistym, śledzenie zmian w podejściu do wyznaczonych celów oraz dostarcza analizy osiągnięć. Dzięki tym funkcjom, użytkownicy mają możliwość świadomego podejścia do procesu samorozwoju, co przekłada się na skuteczniejsze dążenie do zamierzonych rezultatów.

Nasz produkt nie tylko dostarcza narzędzi do zarządzania celami, ale również inspiruje i motywuje użytkowników poprzez dostęp do treści edukacyjnych, artykułów motywacyjnych czy codziennych przypomnień. To kompleksowe podejście wspiera użytkowników w trakcie ich podróży ku lepszej wersji siebie, tworząc przestrzeń do ciągłego rozwoju i osiągania coraz wyższych standardów życiowych.

BeBetter

1.2. Założenia i cele

BeBetter

Aplikacja dostarczona użytkownikowi została zaprojektowana z wyraźnym naciskiem na prostotę i intuicyjność. Jej interfejs został starannie opracowany, aby użytkownik mógł skupić się jedynie na realizacji z góry postawionych celów i zadań. Dzięki temu podejściu eliminuje się zbędne komplikacje, a korzystanie z aplikacji staje się niezwykle intuicyjne nawet dla osób niezaznajomionych z technologią.

W ramach tej aplikacji, użytkownik nie musi martwić się o monitorowanie i pamiętanie o codziennym wykonywaniu zdrowych nawyków. Cała funkcjonalność związana z dbaniem o nawyki zdrowotne jest automatycznie przejmowana przez aplikację. Od śledzenia ilości wypitej wody po monitorowanie aktywności fizycznej czy czasu snu, aplikacja działa jako wsparcie, które aktywnie angażuje się w utrzymanie zdrowego stylu życia.

Dzięki temu podejściu, użytkownicy mogą skoncentrować się na samorozwoju i osiągnięciu swoich celów, pozostawiając aplikacji zadanie śledzenia i analizowania ich postępów. To sprawia, że proces utrzymania zdrowych nawyków staje się prostszy i bardziej dostępny, a użytkownicy mogą cieszyć się efektywnym wsparciem w drodze do osiągnięcia swoich osobistych celów zdrowotnych.

BeBetter

2.1. Struktura katalogów i plików

Główna strona znajduje się w pliku *index.html*.

Podstrony są w folderze *src/sites* i obejmują strony takie jak logowanie, rejestracja, narzędzia, notatki i progres.

Pliki stylujące są w folderze *src/css*, zawierają pliki *style.css*, *style.css.map* oraz *style.scss*.

Ikonki są w folderze *src/icons*.

Pliki JavaScript, które odpowiadają za funkcjonalność strony, są w folderze *src/js* i obejmują pliki takie jak *main.js*, *tools.js*, *create_acc.js*, *logowanie.js*, *progres.js* oraz *notes.js*.

Ta struktura folderów i plików ilustruje organizację projektu i zależności między poszczególnymi elementami.

Struktura folderów i zależności między nimi:

```
1  - project_root
2  - src
3    - css
4      - style.css
5      - style.css.map
6      - style.scss
7    - icons
8    - js
9      - main.js
10     - tools.js
11     - create_acc.js
12     - logowanie.js
13     - progres.js
14     - notes.js
15   - sites
16     - logowanie.html
17     - rejestracja.html
18     - narzedzia.html
19     - notatki.html
20     - progres.html
21   - index.html
```

BeBetter

2.2. Szczegółowa instrukcja instalacji

BeBetter

By pobrać nasz produkt:

- 1. Sklonuj lub pobierz repozytorium. git clone
<https://github.com/NikodemMlynski/BeBetterActual>**
- 2. Otwórz terminal i przejdź do katalogu projektu. cd
BeBetterActual**
- 3. Zainstaluj wymagane zależności, wykonując poniższą
komendę: npm install**
- 4. Po zakończeniu instalacji przenieś się do folderu
backend i uruchom aplikację, wpisując: cd backend
node *index.js* oraz otwierając *index.html*.**

**Aplikacja BeBetter zostanie uruchomiona na lokalnym
serwerze. Otwórz przeglądarkę i przejdź pod adres
<http://localhost:3000>, aby korzystać z aplikacji. Lub wejdź
na naszą domenę hackheroes.mrawsky.fun**

BeBetter

3.1. Architektura aplikacji

Cały projekt bazuje na technologiach powszechnie używanych do tworzenia stron internetowych, takich jak HTML, CSS i JavaScript. Do strukturyzacji i prezentacji treści na stronie wykorzystujemy HTML, natomiast za estetykę i styl projektu odpowiada język CSS. W celu ułatwienia czytelności stylów, zdecydowaliśmy się użyć preprocesora SCSS, który następnie jest kompilowany do standardowego CSS.

Dla zapewnienia interaktywności i dynamiczności całego projektu używamy języka JavaScript. Skrypt ten odpowiada za różnorodne funkcje oraz efekty, które wzbogacają doświadczenie użytkownika na stronie.

W obszarze przesyłania danych na serwer oraz ich dalszego kierowania do bazy danych, wykorzystujemy framework Express.js. Ten narzędziowy framework ułatwia zarządzanie żadaniami HTTP oraz zapewnia efektywną obsługę tras i zapytań.

W celu przechowywania wszystkich danych ze strony korzystamy z bazy danych MongoDB. To rozwiązanie umożliwia nam elastyczne i skalowalne zarządzanie danymi, co jest kluczowe dla efektywnej pracy z aplikacją.

Dodatkowo, za animacje na stronie głównej odpowiada biblioteka AOS, co pozwala na płynne i efektowne animacje, dodając dynamiczność i atrakcyjność wizualną strony głównej.

BeBetter

3.2. Opis klas

Plik main.js:

- **Klasa Slider:** Odpowiada za płynne przesuwanie porad na stronie.
- **Klasa TaskDisplayer:** Zarządza pobieraniem nawyków z serwera i ich wyświetlaniem na stronie. Dodatkowo umożliwia użytkownikowi odhaczanie i usuwanie zadań, dostarczając funkcjonalności sprzyjających skutecznemu zarządzaniu nawykami.

Plik tools.js:

- **Klasa Tools:** Odpowiada za różnorodne funkcje, takie jak:
 - Wybieranie poszczególnych narzędzi i ich centralne wyświetlanie na stronie.
 - Wysyłanie obiektów nawyków oraz obiektów opisów dnia do serwera.
 - Realizacja ćwiczeń oddechowych oraz wybieranie konkretnych ćwiczeń.
 - Walidacja wszystkich formularzy na stronie narzędzi.
 - Wyświetlanie afirmacji i płynne przemieszczanie się między nimi.
 - Dodawanie obiektów wykonanych zadań z inteligentnego dziennika do serwera.

BeBetter

Plik notes.js:

- Klasa Notes: Zajmuje się:
 - Wyświetlaniem celów długoterminowych.
 - Wyświetlaniem szczegółowych opisów wszystkich dni z inteligentnego dziennika.
 - Pobieraniem danych odnośnie celów długoterminowych oraz opisów dni z serwera.
- Klasa Progres: Odpowiada za:
 - Wyświetlanie pełnego wykresu ilustrującego czas poświęcony poszczególnym nawykom w danym dniu.
 - Pobieranie danych dotyczących czasu poświęconego na nawyki, a następnie ich reprezentowanie w wykresie.

Plik create_acc.js:

- Klasa User: Zajmuje się:
 - Walidacją danych wprowadzanych w formularzu.
 - Sprawdzaniem, czy dany email już istnieje w bazie danych.
 - Dodawaniem użytkowników do bazy danych.

Plik logowanie.js:

- Klasa Login: Odpowiada za:
 - Proces logowania użytkowników do strony.

BeBetter

3.3. Opis funkcji

Plik main.js

- **Klasa Slider:**

- **init()** : jest funkcją domyślną klasy Slider: odpowiada za uruchamianie wszystkich innych funkcji w tej klasie
- **changeSlide()**: odpowiada za zmienianie porady
- **increaseActualSlided()**: odpowiada za przesuwanie porady w prawo
- **decreaseActualSlide()**: odpowiada za przesuwanie porady w lewo
- **checkActualSlide()**: sprawdza czy indeks aktualnej porady nie jest większy od długości tablicy
- **checkActualSlideForDecrease()**: sprawdza czy indeks aktualnej porady nie jest mniejszy niż 0

- **Klasa TaskDisplayer:**

- **init()**: jest funkcją domyślną klasy TaskDisplayer: odpowiada za uruchamianie wszystkich innych funkcji w tej klasie
- **getTaskFromDatabase()**: pobiera nawyki z bazy danych, wywołuje funkcję **displayTaskAtSite()**
- **displayTaskAtSite()**: wyświetla zadania na stronie **index.html**
- **checkIfTaskCompleted()**: sprawdza czy nawyk jest odhaczony, jeśli tak wyświetla zadanie w odpowiednim stylu, jeśli nie wyświetla je domyślnie
- **deleteTask()**: wysła zapytanie do serwera żeby usunął nawyk z bazy danych
- **checkOffTask()**: wysła zapytanie do serwera żeby zmienić wartość pola odhaczone na true

BeBetter

Plik progres.js

- Klasa Progres:
 - **init():** jest funkcją domyślną klasy Progres: odpowiada za uruchamianie wszystkich innych funkcji w tej klasie.
 - **displayTaskToProgresAnalisticByTitle():** odpowiada za wyświetlenie wszystkich nawyków wywołuje funkcję **FillTaskProgresAnalistic()**
 - **fillTaskToProgresAnalistic():** wyświetla nawyki na wykresie w postaci słupków
 - **getTaskToProgresFromDatabase():** pobiera nawyki do wyświetlenia na wykresie z bazy danych
 - **renderTaskToProgresTitleButtonsInHTML:** wyświetla tytuły wszystkich zadań aby można było wybrać które zadanie chcemy aktualnie wyświetlić
 - **increaseTaskToProgresDate():** zwiększa miesiąc wykresu danego zadania
 - **decreaseTaskToProgresDate():** zmniejsza miesiąc wykresu danego zadania

Plik notes.js

- Klasa Notes:
 - **init():** jest funkcją domyślną klasy Notes odpowiada za uruchamianie wszystkich innych funkcji w tej klasie
 - **getLonTermGoalsFromDatabase():** pobiera cele długoterminowe z bazy danych, wywołuje funkcję **renderLongTermGoalToHTML**
 - **renderLongTermGoalToHTML():** wyświetla cele długoterminowe w pliku notes.html
 - **getDiaryNotesFromDatabase():** pobiera wszystkie opisy dni z bazy danych, wywołuje funkcję **renderDiaryNotesInHTML**
 - **renderDiaryNotesInHTML():** wyświetla wszystkie opisy dni w pliku notes.html
 - **getDiaryDate():** konwertuje obiekt date z wpisu z inteligentnego dziennika, na przystępną dla użytkownika date

BeBetter

Plik logowanie.js

- Klasa Login:
 - **init():** jest funkcją domyślną klasy Login odpowiada za uruchamianie wszystkich innych funkcji w tej klasie
 - **getUsersFromDatabase():** pobiera dane użytkowników z bazy danych
 - **checkIfEmailExists():** sprawdza czy podany email istnieje w bazie danych
 - **checkIfPasswordCorrect():** sprawdza czy podane hasło zgadza się z hasłem dla podanego emaila

Plik create_acc.js

- Klasa User:
 - **init():** jest funkcją domyślną klasy User odpowiada za uruchamianie wszystkich innych funkcji w tej klasie
 - **validateUserData():** sprawdza czy podane przez użytkownika dane spełniają podstawowe wymgi tworzenia konta
 - **setInputsEmpty():** ustawia wszystkie pola formularza na puste po przesłaniu formularza
 - **addUserToDatabase():** dodaje użytkownika do bazy danych
 - **getUsersFromDatabase():** pobiera dane użytkowników z bazy danych

BeBetter

3.4. Struktura zarządzania danymi

Struktura bazy danych:

- Baza danych nazywa się BeBetter.
- Jest to baza MongoDB.
- W tej bazie danych znajdują się kolekcje których nazwa zależy od podanego emailu użytkownika:
 - `tasks`email``: zawiera obiekty zadań: tytuł, czas zadania(15:00), opis
 - `diary_notes`email``: zawiera obiekty opisów dnia: ogólny opis dnia, dobre decyzje, złe decyzje, ilość szklanek wody, ilość kcalorii, ilość snu, jakość snu, data
 - `long_term_goals`email``: zawiera cele długoterminowe
 - `task_to_progres`email``: zawiera ilość czasu, datę i tytuł nawyku, który potem jest wyświetlany na stronie wykresu
 - `users`: przechowuje obiekty zawierające dane o użytkownikach, nie jest zależne od emaila

BeBetter

4.1. Dalsze ambicje i możliwości rozszerzeń

BeBetter

Ścieżki rozwoju naszego produktu obejmują szeroki zakres możliwości, jednak zdecydowanie najważniejszym kierunkiem jest nawiązanie współpracy i dodanie jeszcze bardziej wartościowych treści. Naszym celem jest nie tylko utrzymanie roli inteligentnego dziennika, lecz także przekształcenie aplikacji w inspirującego przewodnika, zachęcającego użytkownika do podjęcia nowych wyzwań i odkrywania nowych ścieżek rozwoju. Centralnym elementem tych kierunków są plany dotyczące:

- **Współpracy i Partnerstw:**
 - Rozwijanie relacji z partnerami, trenerami, ekspertami w dziedzinie rozwoju osobistego oraz organizacjami zajmującymi się zdrowiem psychicznym. Dążyć do współpracy, która umożliwi dostarczanie jeszcze bardziej spersonalizowanych i skutecznych treści.
- **Wartościowych Treści:**
 - Rozszerzanie bazy treści o materiały edukacyjne, inspirujące artykuły i praktyczne poradniki. Zapewnienie użytkownikowi dostępu do treści, które nie tylko wspierają codzienną praktykę, ale także poszerzają horyzonty i rozwijają umiejętności w obszarze samorozwoju.

BeBetter

- **Motywujących Wyzwań:**
 - Implementacja interaktywnych funkcji, takich jak wyzwania i cele, które motywują użytkowników do podejmowania nowych, zdrowych nawyków. Tworzenie systemów nagród i uznania dla osiągnięć, co dodatkowo zachęca do zaangażowania.
- **Edukacyjnych Artykułów:**
 - Udostępnianie ciekawych i przystępnych edukacyjnych artykułów na temat rozwoju osobistego, zdrowego stylu życia oraz skutecznych strategii osiągania celów. Zapewnienie użytkownikom dostępu do wiedzy, która inspirowie do ciągłego doskonalenia siebie.
- **Przyjaznej Społeczności:**
 - Kreowanie społeczności użytkowników, którzy dzielą się swoimi doświadczeniami, celebrowanie wspólnych sukcesów i wspieranie się nawzajem w procesie rozwoju. Wprowadzanie funkcji społecznościowych, które umożliwiają wymianę pomysłów i wspólne podejmowanie wyzwań.

W ten sposób pragniemy uczynić naszą aplikację nie tylko praktycznym narzędziem do monitorowania postępów, ale także inspirującym źródłem motywacji i wiedzy, które pomagają użytkownikom osiągać nowe poziomy rozwoju osobistego.

BeBetter