

# Dokumentacja projektu: MyFitnessPal

## 1. Wprowadzenie

MyFitnessPal to aplikacja do śledzenia diety i aktywności fizycznej, która umożliwia użytkownikom monitorowanie spożywanych kalorii oraz ćwiczeń fizycznych. Aplikacja została zbudowana z wykorzystaniem nowoczesnych frameworków JavaScript, co zapewnia jej wysoką wydajność, łatwość w rozwijaniu i konserwacji oraz przyjazny interfejs użytkownika.

## 2. Wykorzystane Frameworki JavaScript i Narzędzia

### Frontend

#### React

React to biblioteka JavaScript służąca do budowania interfejsów użytkownika. Umożliwia tworzenie komponentów, które mogą być ponownie wykorzystywane, co znacznie upraszcza proces tworzenia skomplikowanych interfejsów.

- **Architektura:** Komponentowa. UI jest podzielone na małe, niezależne części zwane komponentami, które mogą być wielokrotnie używane.
- **Generatory komponentów:** Użyto narzędzia Create React App do generowania podstawowej struktury aplikacji oraz do tworzenia nowych komponentów.
- **Router:** React Router został użyty do zarządzania nawigacją i kierowania ruchem HTTP w aplikacji.
- **Szablony HTML:** JSX (JavaScript XML) używany jest do definiowania komponentów.
- **Zarządzanie stanem:** Redux służy do centralnego zarządzania stanem aplikacji.
- **Stylizacja:** Styled-components pozwala na pisanie CSS w JavaScript i bezpośrednie stylowanie komponentów React.

### Dodatkowe narzędzia frontendu

- **Axios:** Biblioteka do wykonywania zapytań HTTP z frontendu do backendu.
- **Formik i Yup:** Biblioteki do zarządzania formularzami oraz walidacji danych wejściowych.

### Backend

#### Node.js z Express.js

Node.js to środowisko wykonawcze JavaScript, które umożliwia uruchamianie kodu JavaScript na serwerze. Express.js to framework webowy dla Node.js, który upraszcza tworzenie aplikacji serwerowych.

- **Architektura:** MVC (Model-View-Controller). Umożliwia to rozdzielenie logiki aplikacji na warstwy modelu, widoku i kontrolera.

- **Generatory komponentów:** Express Generator został użyty do stworzenia podstawowej struktury aplikacji serwerowej.
- **Router:** Express Router służy do definiowania ścieżek i zarządzania ruchem HTTP.
- **Konektory do baz danych:** Mongoose jako ORM (Object-Relational Mapping) dla MongoDB.

### Dodatkowe narzędzia backendu

- **jsonwebtoken:** Biblioteka do generowania i weryfikowania JSON Web Tokens (JWT) używanych do autoryzacji użytkowników.
- **bcrypt:** Biblioteka do hashowania haseł użytkowników przed zapisaniem ich w bazie danych.
- **nodemailer:** Biblioteka do wysyłania e-maili, np. w celu resetowania hasła użytkownika.
- **dotenv:** Biblioteka do zarządzania zmiennymi środowiskowymi, pozwalająca na bezpieczne przechowywanie wrażliwych informacji.

### Narzędzia dodatkowe

- **Babel:** Używany do transpilacji kodu JavaScript z ES6+ do wersji kompatybilnych z większością przeglądarek.
- **Webpack:** Do bundlingu zasobów frontendu.
- **ESLint i Prettier:** Narzędzia do analizy statycznej kodu i formatowania kodu, które pomagają utrzymać spójność stylu kodowania.

## 3. Współpraca z REST API

Aplikacja MyFitnessPal komunikuje się z serwerem za pomocą REST API. Axios jest biblioteką używaną do wykonywania zapytań HTTP z frontendu do backendu.

### Przykład użycia Axios:

```
import axios from 'axios';
```

```
const fetchUserData = async (userId) => {
  try {
    const response = await axios.get(`/api/users/${userId}`);
    return response.data;
  } catch (error) {
    console.error("Error fetching user data:", error);
  }
};
```

## 4. Problemy i ich rozwiązania

### **Problem: Zarządzanie stanem w dużych aplikacjach**

**Rozwiązanie:** Użycie Redux do centralizacji zarządzania stanem. Redux pozwala na przechowywanie stanu aplikacji w jednym globalnym store, co ułatwia debugowanie i testowanie.

### **Problem: Asynchroniczne zapytania do serwera**

**Rozwiązanie:** Użycie middleware Redux Thunk do obsługi asynchronicznych akcji. Pozwala to na wykonywanie zapytań do serwera bez blokowania interfejsu użytkownika.

### **Problem: Walidacja danych wejściowych**

**Rozwiązanie:** Użycie bibliotek takich jak Yup (w Formik) w frontendzie oraz Joi w backendzie do walidacji danych wejściowych, co zapewnia, że dane przesyłane przez użytkowników są prawidłowe i bezpieczne.

### **Problem: Zarządzanie autoryzacją i sesjami użytkowników**

**Rozwiązanie:** Użycie JSON Web Tokens (JWT) do zarządzania sesjami użytkowników. JWT umożliwia bezpieczne przechowywanie informacji o sesji w tokenie, który jest przesyłany z każdą prośbą HTTP.

### **Problem: Bezpieczne przechowywanie haseł**

**Rozwiązanie:** Użycie bcrypt do hashowania haseł przed ich zapisaniem w bazie danych, co zwiększa bezpieczeństwo danych użytkowników.

### **Problem: Wysyłanie wiadomości e-mail**

**Rozwiązanie:** Użycie Nodemailer do wysyłania wiadomości e-mail, np. w celu resetowania hasła użytkownika.

### **Problem: Zarządzanie zmiennymi środowiskowymi**

**Rozwiązanie:** Użycie dotenv do zarządzania zmiennymi środowiskowymi, co pozwala na bezpieczne przechowywanie wrażliwych informacji, takich jak klucze API czy hasła do bazy danych.

## 5. Podsumowanie

Projekt MyFitnessPal jest przykładem nowoczesnej aplikacji webowej, wykorzystującej zaawansowane narzędzia i frameworki JavaScript. Dzięki zastosowaniu takich technologii jak React, Node.js, Express.js oraz Redux, aplikacja jest skalowalna, łatwa w utrzymaniu i oferuje przyjazny interfejs użytkownika. Rozwiązania zastosowane w projekcie pozwalają na efektywne zarządzanie stanem aplikacji, obsługę asynchronicznych zapytań oraz zapewnienie bezpieczeństwa danych użytkowników.