

Projekt_RIP - Web

Szymon Bacański

Styczeń 2023

1 Opis funkcjonalny systemu

Celem "Projektu_RIP - Web" było utworzenie webowej aplikacji łączącej się z wcześniej utworzonym api. Pobiera ona potrzebne dane z wystawionych endpointów

Założeniem posiadania aplikacji webowej jest ich popularność. Prawie każda osoba posiada urządzenie umożliwiające korzystanie z przeglądarki web. Od telefonów, przez telewizory nawet po inne urządzenia codziennego użytku.

2 Streszczenie opisu technologicznego

Node.js jest środowiskiem pozwalającym na łatwiejsze tworzenie aplikacji webowych korzystających z javascript. Wspiera to najpopularniejsze frameworki jak React, Vue czy Angular. Wspiera ono wiele platform, w tym te najbardziej popularne jak windows czy linux.

React jest biblioteką języka javascript, wykorzystywana do tworzenia frontendowych aplikacji. Składa się ona z małych, oddzielonych od siebie elementów. Pozwala ona na utworzenie dynamicznych aplikacji web, które skalują się w zależności od urządzenia.

FullCalendar jest biblioteką javascript, która bez problemu działa z aplikacjami opartymi na najbardziej popularnych bibliotekach jak Ract, Vue czy Angular. Wspomaga ona tworzenie aplikacji, które wykorzystują wyświetlanie danych w stylu kalendarzowym.

3 Instrukcję lokalnego i zdalnego uruchomienia systemu

3.1 Postawienie systemu lokalnie

Wymagane oprogramowanie:

Visual Studio Code lub dowolny inny IDE

npm

Terminal Windows (nie jest potrzebny osobno jeśli jest wbudowany w IDE)

Github Desktop – aby móc wprowadzać zmiany, bądź pobierać aktualizacje jeśli są potrzebne. (nie jest wymagany jeśli jest wbudowany w IDE, bądź jeśli ktoś posiada zainstalowany pakiet GIT do użytku poprzez terminal)

Jak postawić środowisko testowe ?

Jeżeli wszystko zainstalowałeś, przejdź do instrukcji poniżej:

1. Pobranie projektu z repozytorium oraz przejście do jego folderu:

```
git clone https://github.com/sbacanski0730/RIP-Rewak-and-PUM-Web.git
cd RIP-Rewak-and-PUM-Web
```

2. Uruchomienie środowiska deweloperskiego

```
npm start
```

4 Dokumentacja

Link do dokumentacji na naszych repozytoriach:

api: <https://github.com/sbacanski0730/RIP-Rewak-and-PUM-API/tree/main/documentation>

web: <https://github.com/sbacanski0730/RIP-Rewak-and-PUM-Web/tree/main/documentation>

mobile: <https://github.com/sbacanski0730/RIP-Rewak-and-PUM-Mobile/tree/main/documentation>

5 Wnioski projektowe

Tworzenie projektu opierającego się na api z podziałem zadań na dwa podzespoły jest wymagające.

Do pewnego stopnia można obsłużyć dane korzystając z wygenerowanych przykładowych plików json jednak mogą wystąpić różnice między danymi, które końcowo są wyprowadzane z api, a tymi, które znajdowały się w wygenerowanych przykładowych json'ach.

Różnice zdań oraz wymagania technologiczne mogą doprowadzić do małych zderzeń między podzespołami, jednak kluczem do napisania działającego projektu jest porządzenie sobie z tym pomimo istniejących problemów.

Wiele stresu mogą też powodować zbliżające się terminy, które nieubłaganie wydają się być coraz bliżej. Jednak dobra współpraca i porządzenie sobie z różnicami zdań pozwala wyjść ponad takie problemy i dokończyć projekt.