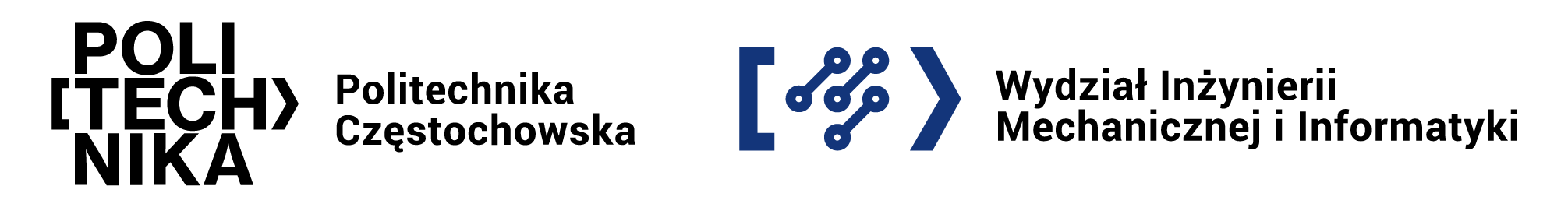
****

PRACA DYPLOMOWA INŻYNIERSKA / MAGISTERSKA

**Projekt i implementacja systemu informatycznego wspomagającego zarządzanie klientami i usługami w sieci klubów fitness**

*Project and implementation of an IT system supporting the management of clients and services in the network of fitness clubs*

Patryk Skwara

Nr albumu: 129766

Kierunek: Informatyka

Forma studiów: stacjonarne

Poziom studiów : I

Promotor pracy: Dr inż. Mariusz Ciesielski

Praca przyjęta dnia:

Podpis promotora:

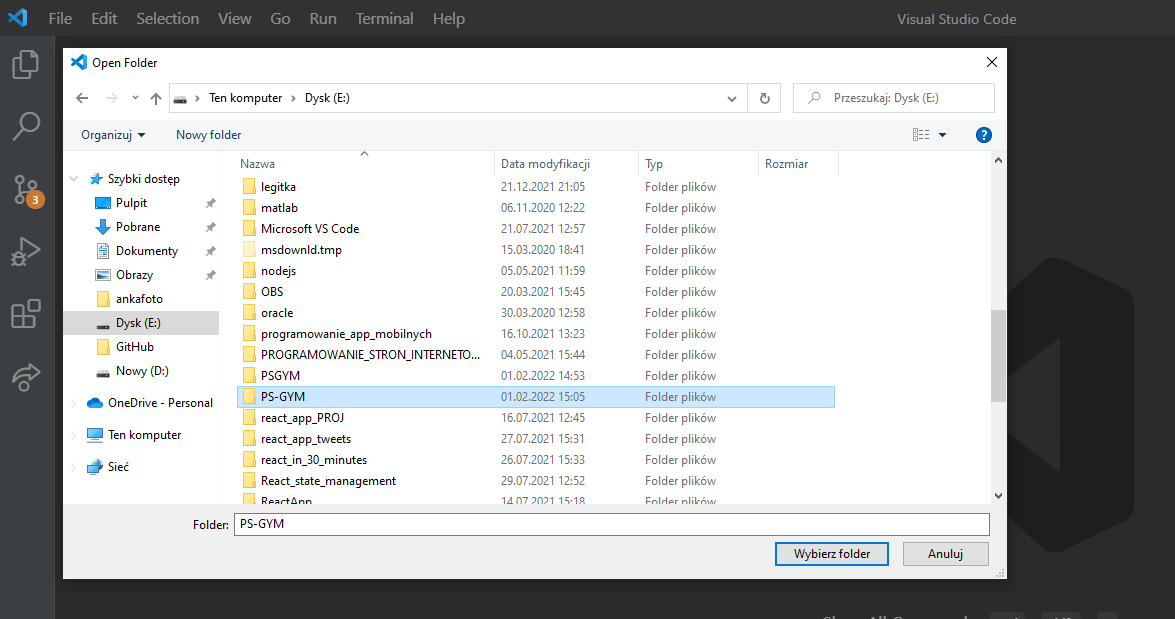
*Częstochowa, …….*

1. **Opis aplikacji**
   1. Uruchomienie aplikacji

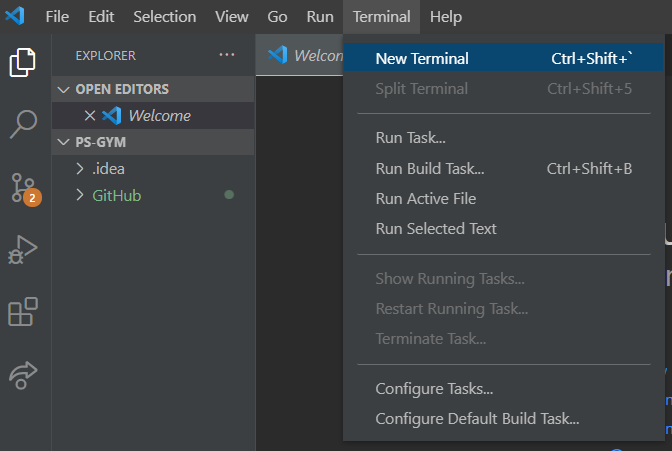
1.1a Aby uruchomić aplikację trzeba uwzględnić podział na aplikację front-end oraz back-end pośredniczącą między serwerem a bazą danych.

1.1b Uruchomienie aplikacji klienckiej

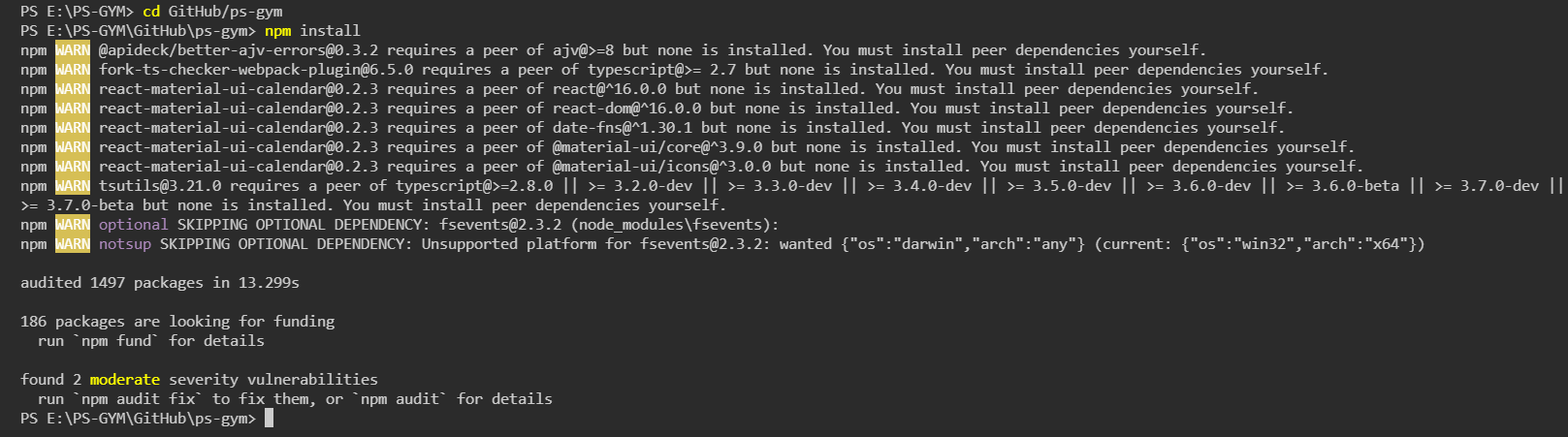
Po pobraniu aplikacji z repozytorium należy uruchomić folder PS-GYM przy pomocy środowiska programistycznego (np. Visual Studio Code).



Następnie należy skorzystać z zakładki „Terminal” i wybrać opcję „New Terminal”

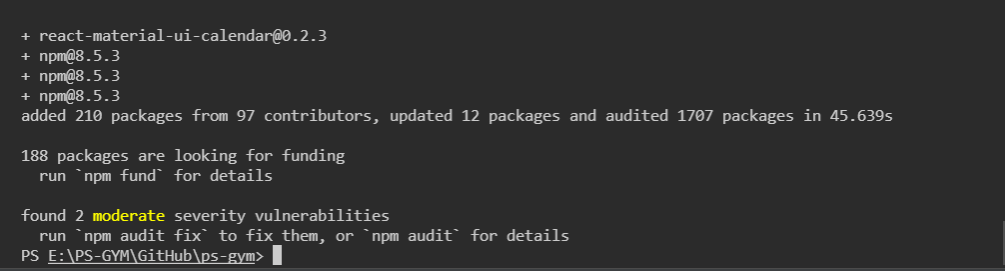


Teraz trzeba dostać się do folderu z kodem aplikacji. Uzyskamy to za pomocą polecenia: cd GitHub/ps-gym. Następnie należy zainstalować szereg pakietów biblioteki React przy pomocy polecenia npm install.

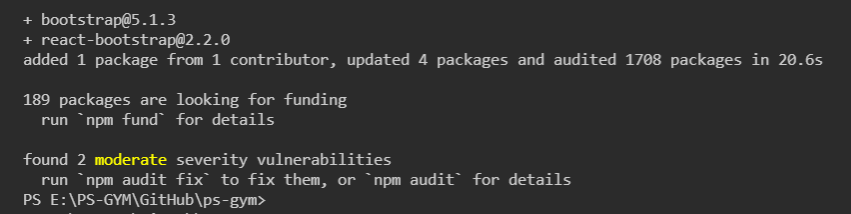


W następnym kroku trzeba zainstalować bibliotekę stylów React Material przy pomocy poleceń:

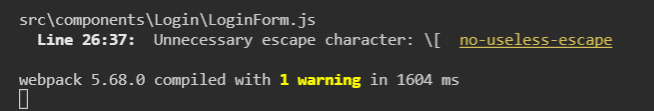
* npm install @mui/material,
* npm install @emotion/react,
* npm install @emotion/styled,
* npm install react-material-ui-calendar



Brakuje już tylko biblioteki stylów bootstrap. W celu instalacji należy wydać polecenie: npm install react-boostrap [bootstrap@5.1.3](mailto:bootstrap@5.1.3)

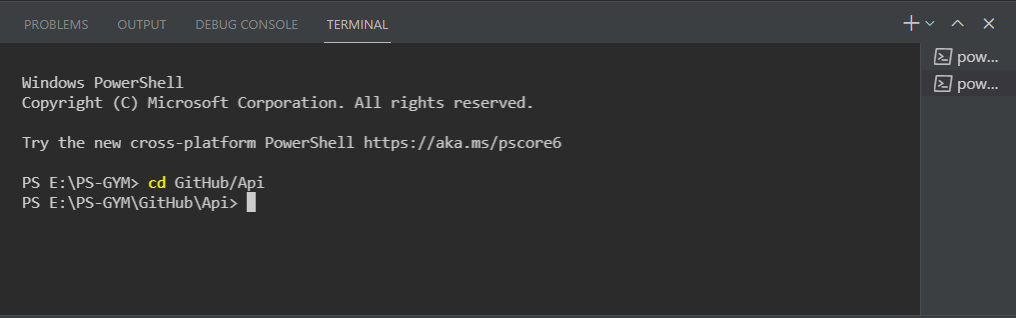


Teraz już wystarczy tylko wydać polecenie npm start aby uruchomić aplikację.

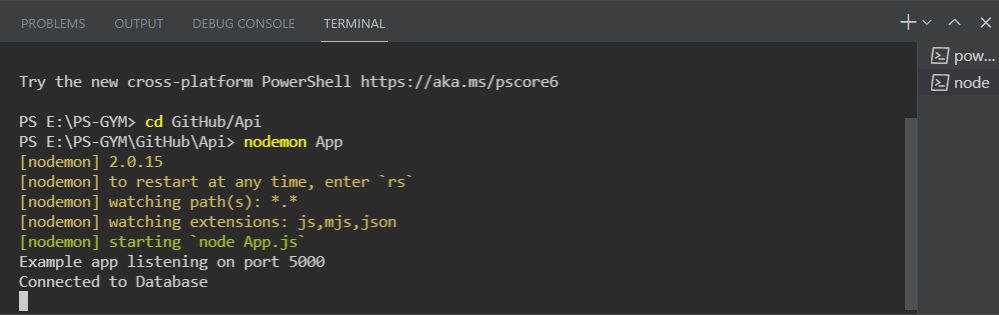


1.1c Uruchomienie aplikacji back-end’owej pośredniczącej między serwerem a bazą danych.

Pierwszym krokiem jest pobranie środowiska uruchomieniowego Node.js ze strony <https://nodejs.org/en/>. Następnie należy otworzyć nowy terminal w środowisku programistycznym, po czym przejść do katalogu z serwerem. Aby to zrobić należy wykonać polecenie cd GitHub/Api.



Teraz należy globalnie zainstalować narzędzie uruchomieniowe dzięki poleceniu npm install -g nodemon, a następnie wpisać polecenie nodemon App. Terminal powinien wyświetlić taki rezultat:



* 1. Projekt i implementacja aplikacji wspomagającej zarządzanie klientami i usługami w sieci klubów fitness.

1.2a Aplikacja stworzona na potrzeby tej pracy powinna być oprogramowaniem umożliwiającym użytkownikowi czyli w domyśle recepcjoniście klubu fitness pełne zarządzanie obiektem. Użytkownik powinien mieć możliwość:

* Zalogowania się do aplikacji przy użyciu swojego indywidualnego konta stworzonego przez administratora,
* Rejestracji klienta,
* Szybkiego wyszukania klienta,
* Edycji danych osobowych klienta,
* Zmiany statusu klienta,
* Wglądu do terminu ważności karnetu,
* Aktualizacji karnetu uprawniającego do korzystania z klubu,
* Dodawania wpisów konkretnemu klientowi odnośnie jego wizyt w klubie tzn. symulacja działania karty magnetycznej polegająca na odczytaniu danych klienta przy użyciu czytnika, który znajduje się przy drzwiach wejściowych do klubu w celu zarejestrowania momentu wejścia do klubu oraz wyjścia z klubu.
* Przeglądania historii wizyt klienta,
* Przeglądania sprzętu dostępnego dla klientów,
* Dodawania nowej pozycji do danej kategorii sprzętu,
* Edycji konkretnej pozycji w danej kategorii sprzętu,
* Przeglądania pracowników zatrudnionych na etat w klubie,
* Przeglądania wydarzeń przydzielonych do konkretnego pracownika,
* Przeglądania kalendarza wszystkich wydarzeń,
* Dodawania wydarzenia,
* Edycji wydarzenia,
* Podglądu szczegółowych informacji na temat konkretnego wydarzenia,
* Usuwania, archiwizowania wydarzenia,
* Przeglądania asortymentu produktów przeznaczonych do sprzedaży klientom,
* Dodawania nowej pozycji produktu,
* Aktualizowania stanu produktu,
* Przeglądania swoich danych osobowych,
* Zmiany zdjęcia profilowego,
* Zmiany hasła,
* Zmiany adresu e-mail

Oprócz konta użytkownika istnieje również konto administratora aplikacji, który posiada wszystkie uprawnienia użytkownika oraz jego funkcjonalność jest rozszerzona o:

* Przeglądanie użytkowników,
* Rejestracje użytkownika,
* Usuwanie konta użytkownika

1.2b Wymagania funkcjonalne i niefunkcjonalne

1.2b.1 Wymagania funkcjonalne:

Aplikacja jest podzielona na poszczególne zakładki, kolejno: klienci, sprzęt, pracownicy, wydarzenia, asortyment, użytkownicy (w przypadku konta administratora) oraz ustawienia.

Na poniższym diagramie przypadków użycia widnieje dwóch aktorów – Administrator i użytkownik czyli konta aplikacji przydzielone do następujących funkcjonalności:

1. Kolumna - Klienci:

* Rejestracja klienta – Administrator, użytkownik,
* Edycja danych osobowych – Administrator, użytkownik,
* Przeglądanie klientów – Administrator, użytkownik,
* Wyszukiwanie klientów – Administrator, użytkownik,
* Aktualizacja karnetu – Administrator, użytkownik,
* Rejestracja wizyt - Administrator, użytkownik,
* Przegląd wizyt - Administrator, użytkownik,
* Zmiana statusu - Administrator, użytkownik,

1. Kolumna – Sprzęt:

* Przeglądanie sprzętu - Administrator, użytkownik,
* Dodanie nowej pozycji - Administrator, użytkownik,
* Edycja pozycji - Administrator, użytkownik,

1. Kolumna – Pracownicy:

* Przeglądanie pracowników - Administrator, użytkownik,
* Podgląd wydarzeń - Administrator, użytkownik,

1. Kolumna – Wydarzenia:

* Przeglądanie wydarzeń - Administrator, użytkownik,
* Dodanie wydarzenia - Administrator, użytkownik,
* Edycja wydarzenia - Administrator, użytkownik,
* Podgląd szczegółowych informacji - Administrator, użytkownik,
* Usuwanie wydarzenia - Administrator, użytkownik,

1. Kolumna – Asortyment:

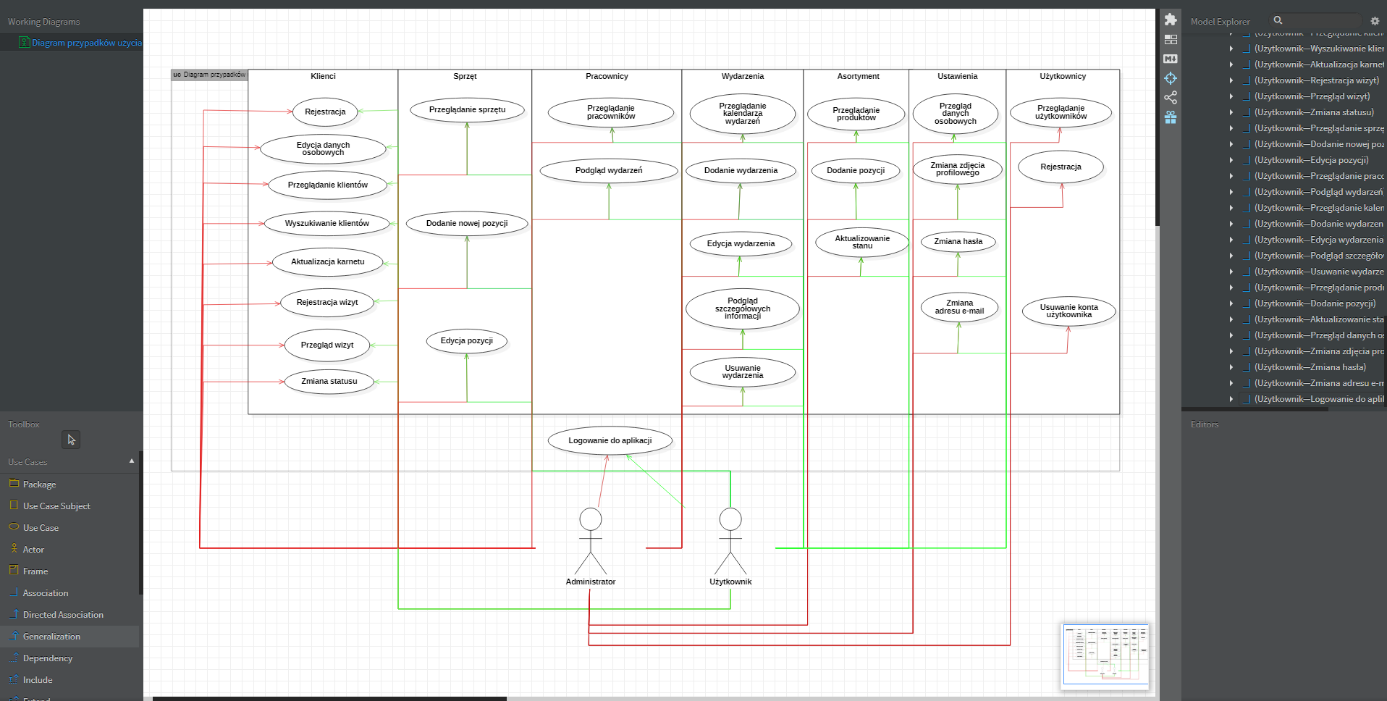
* Przeglądanie produktów - Administrator, użytkownik,
* Dodanie pozycji - Administrator, użytkownik,
* Aktualizowanie stanu - Administrator, użytkownik,

1. Kolumna – Ustawienia:

* Przegląd danych osobowych - Administrator, użytkownik,
* Zmiana zdjęcia profilowego - Administrator, użytkownik,
* Zmiana hasła - Administrator, użytkownik,
* Zmiana adresu e-mail - Administrator, użytkownik,

1. Kolumna – Użytkownicy:

* Przeglądanie użytkowników – Administrator,
* Rejestracja - Administrator,
* Usuwanie konta użytkownika - Administrator



\*diagram przypadków użycia

1.2b.2 Wymagania niefunkcjonalne:

* Użytkownik aplikacji nie powinien musieć przeładowywać strony po dokonaniu jakichkolwiek zmian. W związku z tym w aplikacji została wykorzystana technologia Single Page Application przy użyciu biblioteki React js,
* W momencie uruchomienia aplikacji wszelkie potrzebne dane zostają pobrane za pośrednictwem aplikacji serwerowej z bazy danych,
* Użytkownik podczas logowania powinien mieć automatycznie generowany token dostępowy,
* Użytkownik w momencie logowania do aplikacji jest zweryfikowany przez serwer w taki sposób, aby mógł zalogować się tylko za pośrednictwem swojego konta widniejącego w bazie danych,
* Hasła wszystkich użytkowników powinny zostać zaszyfrowane,
* Aplikacja jest elastyczna, przejrzysta i prosta w obsłudze. Trzeba pamiętać, że z aplikacji będą korzystać osoby niezwiązane z dziedziną informatyki.
* Wszelkie zmiany w aplikacji są odnotowane i zapisane w bazie danych tak, aby po ponownym zalogowaniu się do aplikacji nie utracić wprowadzonych zmian,
* Szczególne pola formularzy są objęte walidacją aby uchronić aplikację przed niechcianymi danymi,
* Podczas wylogowania się z aplikacji token uwierzytelniający jest usuwany z pamięci podręcznej, dzięki czemu użytkownik nie może powrócić do aplikacji wykorzystując przycisk „wstecz”,
* Tylko jeden użytkownik może być zalogowany w aplikacji w czasie rzeczywistym