



Trustee

Dokumentacja programistyczna

Programowanie zespołowe 2020/2021

Zespół #6

Spis treści:

Dokumentacja techniczna	1
Programowanie zespołowe 2020/2021	1
Zespół #6	1
Spis treści:	2
1. O projekcie:	3
2. Aplikacja frontend	4
3.Backend	4
4. Baza danych	5
5. Licencje oraz wykorzystane narzędzia.	5

1. O projekcie:

1.1. Cele biznesowe i funkcjonalności

Trustee jest aplikacją do zarządzania zadaniami w małych oraz średnich firmach. Projekt rozwiązuje problem dostępu do najważniejszych informacji zadania - zasobów takich jak pliki, linki, osoby powiązane z problemem, czy materiały pomocnicze. W ten sposób pomaga pracownikom: doświadczonym oraz tym, którzy dopiero się wdrażają. Zmniejsza także obciążenie pracą kierowników m.in. w zakresie przydzielania zadań.

Głównym elementem projektu jest algorytm rekomendujący m.in. odpowiednie osoby do wykonania danego zadania, podczas jego tworzenia. Opiera się on na etykietach, które opisują właściwości zadań oraz pracowników.

Główne funkcjonalności:

- Dodawanie oraz edycja podstawowych informacji dot. zadań, dodawanie komentarzy i plików do zadań.
- Rekomendacja odpowiednich pracowników pasujących najlepiej do danego zadania
- Możliwość zarządzania projektami oraz pracownikami.
- Automatyczne dodawanie przydatnych plików oraz listy osób do danego zagadnienia.
- Etykietowanie zadań oraz pracowników.

1.2. Opis aplikacji

Projekt to aplikacja webowa, Single Page Application (SPA). Został napisany przy użyciu frameworka Blazor Server (ASP.NET Core 3.1) i EntityFrameworkCore do obsługi bazy danych MySQL Server 8.0. Narzędziami użytymi do jego napisania są Visual Studio 2019 z dodatkami wspierającymi Git/GitHub oraz MySQL Workbench 8.0.

Na chwilę obecną aplikacja wraz z serwerem bazy danych znajduje się na maszynie wirtualnej WMil pod adresem: <http://192.168.134.61/>. Aby ją uruchomić należy posiadać łącze VPN z WMil oraz znać dane logowania do aplikacji.

2. Frontend

2.1. Struktura

Folder **Pages** zawiera składniki/strony z routowalnymi elementami *.razor* (), które składają się na aplikację. Ścieżka dla każdej strony jest określana przy użyciu dyrektywy *@page*.

- **Employees** (*Employees.razor*) - strona pracowników
- **Empty** (*Empty.razor*) - strona pomocnicza
- **Log, Login** (*Log.cshtml*, *Login.razor*) - strona logowania

Trustee - Dokumentacja techniczna

- **Projects** (*Projects.razor*) - strona projektów
- **SingUp** (*SingUp.razor*) - nieaktywna strona rejestracji
- **TasksKanban** (*TasksKanban.razor*) - strona główna, z zadaniami na tablicy
- **TaskList** (*TasksList.razor*) - strona zadań

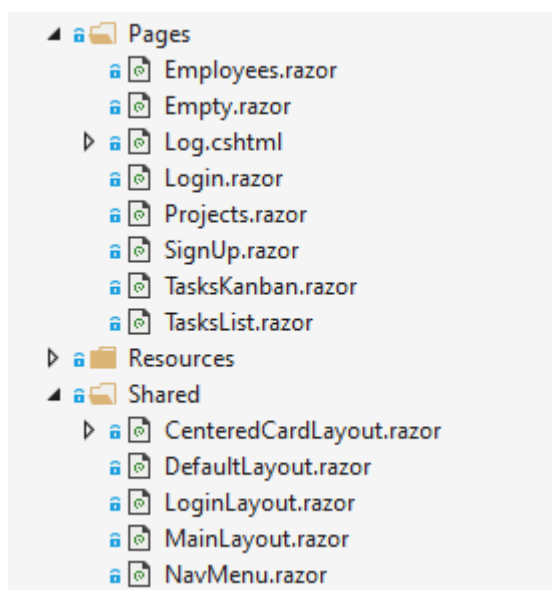
Folder **Shared** zawiera następujące składniki udostępnione i arkusze stylów:

MainLayout component (*MainLayout.razor*): składnik układu aplikacji.

CenteredCardLayout component, **LoginLayout** component (*CenteredCardLayout.razor*, *LoginLayout.razor*): składnik układu strony logowania aplikacji.

NavMenu component (*NavMenu.razor*): implementuje nawigację na pasku bocznym. Zawiera on składnik NavLink, który renderuje linki nawigacji do innych składników Razor.

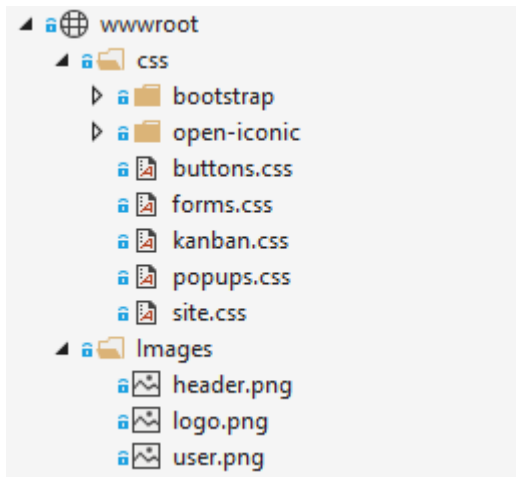
DefaultLayout component (*DefaultLayout.razor*): składnik domyślnego układu aplikacji.



wwwroot: jest to folder główny sieci Web dla aplikacji zawierającej publiczne zasoby statyczne aplikacji, w tym pliki ustawień aplikacji środowiska.

Folder **css**: zawiera katalogi z zasobami Open Iconic oraz Bootstrap. Znajdują się tu również pliki .css z arkuszami stylów dla elementów aplikacji.

Folder **Images**: zawiera pliki obrazów wykorzystywanych w aplikacji.



3. Backend

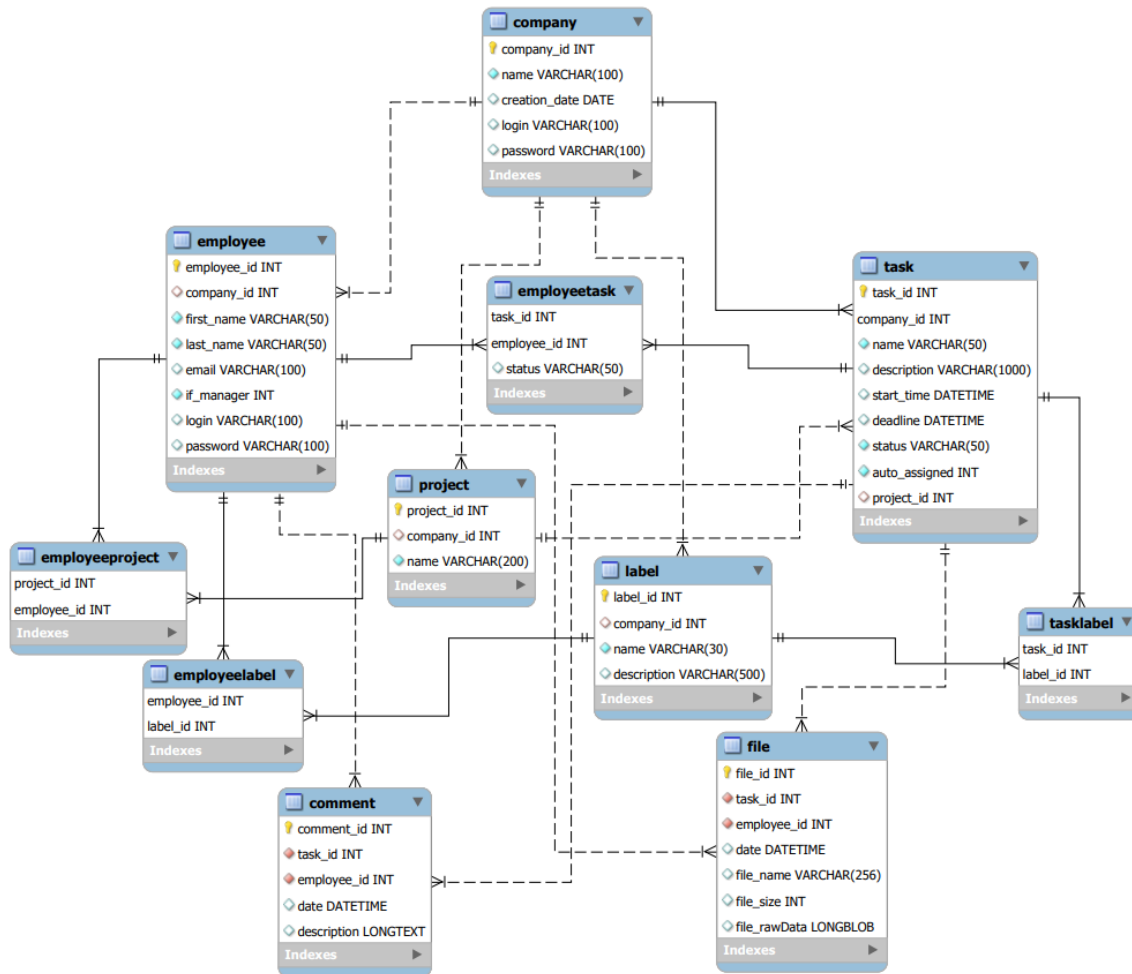
Dokumentacja programistyczna backendu znajduje się w zakładce “**Dokumentacja Doxygen**”.

4. Baza danych

4.1. Opis struktury

Tabele *company*, *employee*, *task*, *project*, *label*, *comment*, *file* przechowują informacje odpowiednio o firmie, pracownikach, zadaniach, projektach, etykietach (zadań i pracowników), komentarzach do zadań, plikach do zadań.

Pozostałe tabele pełnią funkcję kojarzenia rekordów jednej tabeli z rekordem innej w relacjach wiele do wielu, wiele do jednego i jeden do wielu, np. tabela *employeetask* kojarzy pracowników z zadaniami w relacji wiele do wielu - każdy pracownik może być przypisany do wielu zadań i każde zadanie może mieć przypisane wielu pracowników. Rodzaje relacji między tabelami są zaznaczone na schemacie.



5. Licencje oraz wykorzystane narzędzia.

5.1. Narzędzia projektowe

1. Visual Studio 2019 z dodatkami wspierającymi Git/GitHub - praca programistyczna, licencja [Visual Studio Community](#)
2. [MySQL Workbench 8.0](#) - praca z bazą danych
3. [Doxygen, Doxywizard](#) - generowanie dokumentacji technicznej
4. Open Iconic: [fonts](#) and [icons](#) - praca programistyczna
5. [GitHub](#) - repozytorium projektu
6. Syncfusion Community - komponenty w projekcie, na podstawie biblioteki
7. [Google Fonts](#) - czcionki użyte w projekcie
8. [Font Awesome](#) - ikony użyte w projekcie
9. [Unsplash](#) - grafiki użyte w projekcie

Trustee - Dokumentacja techniczna

10. [Videvo](#) - grafika użyta na stronie www projektu, materiał filmowy udostępniony przez Videvo, pobrany z www.videvo.net
11. [Bootstrap](#) - biblioteka CSS
12. [Audacity](#), [Open Shot Video Editor](#), [GIMP](#), Biblioteka Audio Studio Youtube - prace związane ze stworzeniem filmu

5.2. Pozostałe narzędzia:

1. Microsoft Teams - komunikacja zespołu
2. Trello - organizacja zadań
3. Narzędzia Google - organizacja, tworzenie notatek