UNIWERSYTET WSB MERITO W GDAŃSKU

WYDZIAŁ INFORMATYKI I NOWYCH TECHNOLOGII

Paweł Pijanowski

nr albumu 58194

**URZYTKOWA STRONA INTERNETOWA**

**KOLOROWEGO PRZEDSZKOLA**

Praca inżynierska

na kierunku Informatyka

Praca napisana pod kierunkiem

Dr. Dariusz Kralewski

Gdańsk 2024

Spis treści

1. Wstęp
   1. Wprowadzenie
   2. Zakres pracy
2. Wykorzystane technologie
   1. Framework
   2. Użyte języki oraz pakiety
   3. Baza danych
   4. Back-end
   5. Programy i aplikacje
3. Struktura bazy danych
   1. Tabele
   2. Transakcje
4. Połączenie z bazą danych
   1. Kontekst bazy danych
   2. Modele danych
   3. Kontrolery
5. Struktura strony
   1. Strony pojedyncze
   2. Sekcja autoryzacji
   3. Sekcja kadry
   4. Sekcja ofert
   5. Sekcja aktualności
   6. Galeria
6. Bibliografia
7. Przypisy

1. Wstęp
   1. Wprowadzenie

Głównym założeniem tego projektu jest stworzenie strony internetowej spełniającej wymagania i wytyczne klienta. Aplikacja ma być stroną przedszkola o nazwie „Kolorowe”. Pierwszym krokiem osiągnięcia tego celu była rozmowa z klientem, musiałem się dowiedzieć czego dokładnie sobie życzy i opracować pomysł na jak najbardziej efektywne rozwiązanie jego problemów, jednocześnie sugerując dodatkowe funkcje, które przy okazji można byłoby dodać, żeby strona była bardziej dostępna, łatwiejsza do obsługi oraz zawierała wszystko co może być potrzebne na stronie przedszkola.

* 1. Zakres pracy

Najważniejszym celem tej pracy osiągnięcie oczekiwań klienta, który po rozmowie wstępnej przedstawił mi swoje wymagania. Strona internetowa ma służyć nie tylko za stronę informacyjną dla przedszkola, na której można byłoby znaleźć informacje takie jak dokładna lokalizacja, aktualna oferta czy kadra. Najważniejszymi trzema funkcjami, które muszą się pojawić będzie galeria zdjęć, na której pracownik będzie mógł wgrać zdjęcia związane z przedszkolem. System tworzenia, edytowania i usuwania postów, które wyświetlałyby się na stronie. Oraz możliwość kontaktowania się bezpośrednio przez stronę internetową, poprzez wypełnienie formularza, która zostanie wysłana na przedszkolną skrzynkę odbiorczą do przeczytania.

Dodatkowo, po dalszej rozmowie z klientem, zdecydował się on się na parę sugestii, które mu zaproponowałem. Takie jak na przykład możliwość dodawania i usuwania ofert znajdujących się na stronie oraz możliwość edytowania listy kadry, gdyby jedno lub drugie okazało się nieaktualne, wraz z czasem działania przedszkola.

Wszystkie te funkcje oczywiście muszą być zablokowane dla zwykłej osoby odwiedzającej tą stronę i powinny być dostępne tylko dla personelu z odpowiednimi kompetencjami. Ze względu na to, na stronie będzie dodatkowo dodana funkcja logowania dla administratora.

W dalszej części pracy szczegółowo opiszę, jak dokładnie wypełniłem każdy z podanych wyżej celów, jakie problemy napotkałem oraz jakie technologie zostały przeze mnie użyte, żeby strona działała zgodnie z założeniami.

1. Wykorzystane technologie
   1. Framework

W przypadku o tworzenie stron internetowych w dzisiejszych czasach, wręcz wymaganym jest wybranie Frameworku na bazie, którego będzie skonstruowana. Aktualnie na rynku dominują trzy, które są uznawana za standard w branży, są to Vue.js, React i Angular. Po rozważeniu swoich opcji, zdecydowałem się użyć Vue.js. Głównymi powodami wybrania tego Frameworku była przede wszystkim jego budowa wykorzystująca podejście MVC (Model-View-Controller), które jest dla mnie najbardziej instynktowne i najwygodniejsze.

Obraz zawierający tekst, zrzut ekranu, Czcionka, design

Opis wygenerowany automatycznie

Rysunek 1 - przykład wzorca Model-View-Controller

„Model-View-Controller” jest wzorcem używanym w programowaniu, który polega na podzieleniu kodu na trzy główne komponenty, zgodnie ze swoją nazwą: Model, Widok i Kontroler. Rolą Modelu jest reprezentacja danych oraz logika związana z ich działaniem. Modele są odpowiedzialne za chociażby komunikację z bazą danych, przechowywanie danych oraz obsługa logiki strony. Za to rolą Widoku jest prezentacja danych pochodzących z Modelu użytkownikowi oraz reagowanie na interakcje od użytkownika, takie jak przyciski czy wypełnianie formularza. Ostatnim komponentem jest Kontroler, odpowiadający za logikę aplikacji oraz jest pośrednikiem między interakcjami Modelu i Widoku. Kontroler zajmuje się chociażby obsługą żądań użytkownika, wywoływanie odpowiedniego Widoku oraz aktualizowanie Modelu.

Jednak Vue.js jest jedynie bazowany na tym wzorcu, ale interpretuje go w sposób bardziej nowoczesny. Jest tak zwanym „nowoczesnym Frameworkiem” gdzie klasyczne podejście zostało zastąpione przez tak zwane komponenty, które są niezależnymi blokami łączącymi dane (Model), logikę (Kontroler) i szablony (Widoki) w jednej strukturze. Jednocześnie Vue.js oferuje reaktywność danych, co oznacza, że zmiany w Modelu automatycznie wpływają na Widok bez potrzeby informowania Kontrolera, przez co zmiany na stronie są implementowane w czasie rzeczywistym.

Innymi zaletami Vue.js, które nie są związane z jego wewnętrzną strukturą są chociażby: jego lekkość, co przyczynia się do szybkiego działania aplikacji, oraz coś co bardzo przydało się podczas tego projektu, jego ekosystem. Vue.js, albowiem posiada szeroką bazę bibliotek oraz narzędzi, ułatwiających pracę nad całą stroną. Dobrym przykładem takich narzędzi, które zostały użyte w projekcie są: vue-router do zarządzania trasami w aplikacji oraz Vite, który przydał się przy budowaniu strony.

* 1. Użyte języki oraz pakiety

Elementem wręcz niezbędnym przy każdej stronie internetowej, stanowiący jej podstawowy budulec jest HTML 5 (HyperText Markup Language). Pozwala on tworzyć elementy, które przeglądarka interpretuje jako teksty, obrazy, listy, nagłówki i formularze. Działa on na zasadzie struktury hierarchicznej, oznacza to, że dokument tekstowy HTML, ma strukturę drzewiastą, w której jeden element może zawierać kolejne elementy.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 2- przykład kodu HTML 5, wzięty z projektu.

Ważną cechą języka HTML, jest jego integracja z innymi technologiami takimi jak CSS (Cascading Style Sheets) czy JavaScript. Wcześniej wspomniane przeze mnie Vue.js, łączy te trzy języki, pozwalając programistą pisać kod w jednym pliku i bez problemów łączyć jego elementy ze sobą.

Jednym z najważniejszych i powszechnych elementów towarzyszących językowi HTML jest CSS. Służy on do opisywania wyglądu i formatowania dokumentów HTML. Dzięki temu można kontrolować takie aspekty strony jak czcionka, tło, kolory, odstępy i umieszczenie wszystkiego na stronie.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 3- przykład kodu CSS, wzięty z projektu.

Ostatnim niezbędnym komponentem każdej strony internetowej jest JavaScript, jak sama nazwa sugeruje, jest to skryptowy język programowania. W moim projekcie okazał się być niezbędnym elementem, odpowiedzialnym za połączenie strony internetowej z bazą danych, co umożliwiło główną funkcje strony wymagane przez klienta, takie jak później opisana sekcja aktualności.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 4- przykład kodu JavaScript, wzięty z projektu

Jednak JavaScript potrzebuje silnika, na którym będzie działać lokalnie, poza stroną internetową, owym silnikiem jest Node.js, który dokładnie mówiąc jest środowiskiem uruchomieniowym. Pozwoliło mi to na zastosowanie szerokiej biblioteki zawartej ze wcześniej wspomnianym środowiskiem. Dzięki NPM (Node Package Manager) w prosty sposób można ściągnąć dowolny pakiet, zawierający kod gotowy do użycia w projekcie. Z tego względu chciałbym zaprezentować listę pakietów, które są wykorzystane w moim projekcie, oraz wyjaśnienie, dlaczego zostały użyte:

* Leaflet v1.9.4 – Jest to otwarto-źródłowa biblioteka pozwalająca na użycie interaktywnej mapy na stronie. Jest używana, żeby wskazać dokładną lokalizację przedszkola.
* Bootstrap v5.3.3 – Bootstrap jest biblioteka CSS, która ma za zadanie ułatwić i usprawnić wygląd strony. W projekcie nie jest używana zbyt często, ale jednym z zastosowań jest kafelkowy wygląd galerii.
* Font Awsome v6.7.2 – Jest to prosty zestaw ikonek, które można używać w dowolny sposób. W projekcie są wykorzystane w kilku miejscach, na przykład strzałki do zmieniania stron w aktualnościach.
* Vue3-editor v0.1.1 – Ta biblioteka dodaje prosty edytor tekstu, który jest używany w większości funkcji mojej strony, w celu tworzenia oraz edytowania tekstu.
* Date-fns v4.1.0 – Do zadania tej paczki należy reinterpretowanie formatu daty znajdującej się w detalach aktualności.
  1. Baza Danych

Wybór bazy danych obsługującej tą aplikację był prosty, różnych alternatyw jest wiele w internecie, takie jak: MongoDB, PostgreSQL czy SqlLite. Jednak po zastanowieniu się zdecydowałem się użyć MySQL. Jednak, żeby serwer w ogóle funkcjonował misy być na czymś postawiony, w moim wypadku zdecydowałem się użyć pakietu oprogramowania XAMPP. Pomimo tego, że posiada on wiele funkcji, ja potrzebuję tylko dwóch z nich: serwera Apache, którego rolą jest obsługiwanie żądań HTTPS, oraz silnika MySQL, który odpowiada bezpośrednio za działanie bazy danych. Oba z nich są włączone w postaci serwera lokalnego na portach komputera.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Ikona komputerowa

Opis wygenerowany automatycznie

Rysunek 5- pakiet oprogramowania XAMPP z włączonym serwerem Apache i MySQL

Jest wiele czynników które przyczyniły się do mojego wyboru MySQL, przede wszystkim jest bardzo prosta w obsłudze, szczególnie biorąc pod uwagę fakt, że XAMPP integruje serwer bazy danych z interfejsem „phpmyadmin”, co stanowczo ułatwia nawigację oraz modyfikację. Dodatkowo, MySQL jest bazą zoptymalizowaną pod działanie ze stronami internetowymi, oferując wysoką wydajność przy dużej ilości użytkowników, co jest możliwością przy tej aplikacji. Jest on również jedną z najpopularniejszych baz danych na świecie, co oznacza, że jakiekolwiek problemy, które ja mogłem mieć podczas tworzenia strony zostały już napotkane i rozwiązane przez innych użytkowników. Bardzo ważnym czynnikiem był również koszt, MySQL bowiem jest aplikację z otwartym kodem źródłowym (ang. open-source) co oznacza, że jest darmowy do użytku, jednocześnie i tak oferując płatną wersję (np.: MySQL Enterprise Edition”), które mają dodatkowe funkcje i oferują wsparcie techniczne.

* 1. Back-end

Podstawą każdej aplikacji internetowej jest back-end, odpowiada on za działanie strony za kulisami obsługuje on logikę biznesową oraz łączy się z serwerem oraz bazą danych. Za API aplikacji zdecydowałem się wybrać framework „.NET core”, jest to uniwersalna platforma programistyczna stworzona przez firmę Microsoft. Sama popularność jednak nie jest główną przyczyną mojego wyboru, był to ekosystem i dodatkowa narzędzia jakie oferuje. Najważniejsze z tych narzędzi jest oferowane dzięki „Visual Studio 2022”, którymi są paczki NuGet. Dzięki tym paczką w prosty sposób można ściągnąć dowolny pakiet, zawierający kod gotowy do użycia w projekcie. Z tego względu chciałbym zaprezentować listę najważniejszych z nich:

* MailKit – jest to prosta paczka, której głównym zadaniem jest ułatwianie wysyłania maili, używając strony internetowej.
* BCrypt.Net – kolejna dość prosta paczka, służy ona do inskrypcji tekstu za pomocą zmodyfikowanego algorytmu „Blowfish”. W projekcie służy ona do szyfrowania hasła w bazie danych.
* Microsoft.EntityFrameworkCore – w moim projekcie jest parę paczek zaczynających się za pomocą tej nazwy, jednak rozszerzają one tylko zastosowanie głównej części. Sam EntityFrameworkCore jest to nowoczesny maper obiektów dla baz danych dla .NET. Obsługuje on zapytania LINQ, śledzenie zmian, aktualizację i migrację schematów.
* Microsoft.AspNetCore.Authentication.JwtBearer – ta paczki odpowiada za proste zadanie, ma przesyłać token uwierzytelniający do front-endu.
  1. Programy i aplikacje

Podczas tworzenia tego projektu, oczywiście musiałem użyć paru programów do jego stworzenia. Podstawą tego projektu były aplikacje „Microsoft Visual Studio Code”, który służył mi jako terminal dla NPM oraz jako kompilator kodu dla HTML, CSS i JavaScript. Drugi to „Microsoft Visual Studio 2022” było to główne narzędzie do stworzenia back-endu do połączenia z bazą danych. Sama baza jest stworzona za pomocą MySQL, jest to możliwe dzięki kolejnemu programowi, który używam, czyli XAMPP (ang. X-platform, Apache, MySql, php, pearl). Jest to pakiet oprogramowani, który zawiera funkcje bardzo przydatne mojemu projektowi, pierwszą z nich jest serwer Apache, którego rolą jest obsługa żądań HTTP oraz silnika MySql, który jest drugą potrzebną mi funkcją. Posiada on też wbudowany interfejs „phpmyadmin”, który jest nadzwyczaj przydatny w tym przypadku, gdyż w dużym stopniu pomaga ze wszystkimi akcjami związanymi w bazie danych.

Ponieważ ten projekt był robiony na dwóch maszynach (laptopie oraz komputerze stacjonarnym) ważną jego częścią jest archiwizacja oraz przenoszenie między maszynami. Pracę w tym ułatwił i umożliwił mi git, a dokładnie aplikacja Sourcetree, która pomogła mi bez problemów aktualizować mój projekt na git-cie.

1. Struktura bazy danych
   1. Tabele

Głównym składnikiem tworzącym bazy danych są tabele, jest to główna forma kategoryzowania danych. Tabele później dzielą się na kolumny, które już bezpośrednio zawierają informację. Ze względu na to, że bazy danych są niezbędnym składnikiem każdej strony internetowej koniecznie musiałem je zaimplementować dla prawidłowego działania aplikacji.

Obraz zawierający tekst, Czcionka, numer, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 6- Diagram bazy danych użytej w projekcie.

Pierwszą tabelą, którą należałoby przedstawić jest tabela „users”. Jest ona bezpośrednio powiązana z systemem autoryzacji, który wprowadziłem do aplikacji. Tabela „users” ma w sobie trzy kolumny: Id które jest głównym kluczem tej tabeli oraz automatycznie się zwiększa z każdym nowym wejściem. Username które jest odpowiedzialne za zapisywanie nazwy użytkownika oraz PasswordHash zapisujący ciąg znaków, który w back-endzie jest kodowany i dekodowany, służy on rolę hasła dla użytkownika.

Następną tabelą jest „offers”, stanowi ona część systemu zamieszczania ofert na stronie internetowej. System polega bezpośrednio na wyświetlaniu dwóch kolumn z tej tabeli Title i Content. Tak jak same nazwy sugerują Title zawiera tytuł oferty a Content jej zawartość. Tak jak poprzednia tabela, ta również zawiera kolumnę Id, która jest głównym kluczem i z każdym nowym wejściem automatycznie się zwiększa.

Tabela „employees” wyróżnia się najbardziej ze wszystkich, zawiera ona rekordową ilość siedmiu kolumn, gdzie sześć z nich jest poświęcona dla każdej z informacji, która musi być wyświetlona na stronie. Przedszkole posiada 4 grupy dla dzieci: czerwoną, niebieską, zieloną i żółtą. Dodatkowo potrzebna jest jeszcze informacja dotycząca dyrekcji oraz specjalistów, którzy są tam zatrudnieni do zajęć dodatkowych. Zawiera ona również kolumnę Id, ale jako jedyna z póki co przedstawionych tabel nie zwiększa ona numeru przy nowym wejściu, ze względu na to, że nowe wejścia nigdy nie powinny się pojawić. Ze względu na to stronie, że na stronie liczba kategorii w których ludzie są zatrudnieni, ograniczają się do jedynie tych sześciu, zdecydowałem się stworzyć bazę danych, gdzie wystarczy aktualizować wpis o Id = 1. Na samej stronie wyświetlane są tylko wartości innych kolumn, których przypisane jest Id =1, dzięki czemu wystarczy aktualizować pierwszy wpis w bazie danych, żeby zmienić informacje zamieszczone na stronie.



Rysunek 7- tabela "employees" z przykładowymi danymi

Przedostatnią tabelą, o której się wypowiem jest „images”, jej rolą jest przechowywanie wszystkich zdjęć w formie tekstu, jako ścieżkę do zdjęcia, które jest zapisane w folderze projektu. Główną rolę wypełnia kolumna FileName, jest to miejsce, gdzie wspomniana ścieżka jest zapisywana. Kolumna Id, tak jak przy tabeli „offers” i „users” jest głównym kluczem i z każdym nowym wejściem automatycznie się zwiększa. Kolumna PostId jest obcym kluczem, o którym napiszę w następnej sekcji. Kiedy wspomniałem, że ta tabela zapisuje wszystkie zdjęcia, miałem też na myśli zdjęcia, które są przypisane do postów.

Tabela „posts” jest miejscem, gdzie zapisywane są składniki postów, z których składa się strona aktualności w aplikacji. Przechowywane dane to: data w kolumnie Date, zawartość postu w kolumnie Content oraz tytułu w kolumnie Title. Tak jak większość poprzednich tabel, zawiera ona również kolumnę Id, której numer zwiększa się z każdym nowym wejściem automatycznie.

Bardzo ważnym elementem tej tabeli jest połączenie kolumny Id za pomocą obcego klucza do kolumny PostId w tabeli „images”. Jest to połączenie typu „wielu do jednego” (ang. „Many to one”), oznacza to, że w na jedno Id w tabeli „posts” może być wiele przypisanych wejść w tabeli „images”, ze względu na to że to samo zdjęcie może być przypisane do wielu postów.

1. Połączenie z bazą danych
   1. Wstęp

Ta sekcja mojej pracy jest poświęcona back-endowi, wyjaśniania jego komponentów oraz dokładne pokrycie jego działania.

* 1. Modele danych

Modele w aplikacjach ASP.NET są wręcz niezbędnym komponentem architektury MVC (Model-View-Controller). Ich główną funkcją jest reprezentowanie i zarządzanie danymi i ich logiką biznesową.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Czcionka

Opis wygenerowany automatycznie

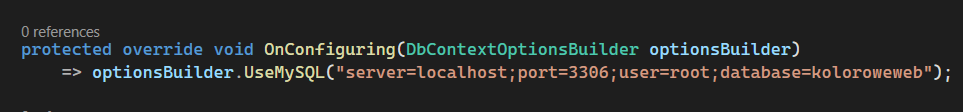
Rysunek 8- model danych tabeli "users"

Żeby logika poprawnie funkcjonowała trzeba stworzyć odpowiedni model, w moim przypadku jest to klasa choć może to być zarówno ich zestaw. Najważniejszą funkcją modelu danych jest ich mapowanie z bazy danych lub innego źródła na obiekty c#. Modele również mogą zawierać logikę weryfikacji danych do sprawdzania ich poprawności. Natomiast dla mnie, najważniejszą ich funkcją jest interakcja z bazą danych. Używając narzędzi ORM (Object Related Mapping) takich jak na przykład „Entity Framework” modele mogą być używane do operacji CRUD na bazie danych.

* 1. Kontekst bazy danych

Plik kontekstowy w back-endzie to plik, który służy jako centrum zarządzania danymi kontekstu. W tym przypadku kontekst dotyczy konfiguracji i usług, które muszą być dostępne dla wielu innych części aplikacji.

Pierwszą i tą mniej ważną funkcją jest stworzenie bezpośredniego połączenia z bazą danych za pomocą funkcji „OnConfiguring”, pozwalającej utworzyć dokładną ścieżkę do serwera MySQL.



Rysunek 9- funkcja "OnConfiguring" z projektu

Najważniejszym celem pliku kontekstowego w mojej aplikacji, jest zbudowanie modeli według wzoru z bazy danych. Za pomocą metody „OnModelCreating” jestem w stanie osiągnąć dokładnie to, zbudować model dla każdej wcześniej opisanej tabeli w bazie danych, jednocześnie definiując klucze główne, mapując kolumny i określa dokładne relacje między tabelami.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 10- funkcja „modelBuilder” budująca model dla tabeli "posts"

* 1. Kontrolery

Zdecydowanie najobszerniejszą częścią back-endu są kontrolery, niezbędne do modyfikowania każdej z tablic znajdującej się w bazie danych, do tego potrzeba jest stworzyć osobny kontroler do każdej z nich. Każdy z plików zawiera dokładne operacje CRUD (Create, Read, Update, Delete), których zadaniem jak sama nazwa mówi, odpowiadają za odpowiednie operacje na bazie danych: tworzenie nowych zapisów, czytanie już istniejących zapisów, aktualizowanie zapisów oraz ich usuwanie. Jednak wymienione operacje to nie jest jedyny powód, żeby

Pierwszym kontrolerem do omówienia jest „AuthController.cs”. Zajmuje się on obsługą sekcji związanej z autoryzacją użytkownika, czyli w tym wypadku rejestrowanie nowych i logowanie już istniejących już użytkowników.

Obraz zawierający tekst, zrzut ekranu, Czcionka, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 11- punkt końcowy API "register" z pliku "AuthController.cs"

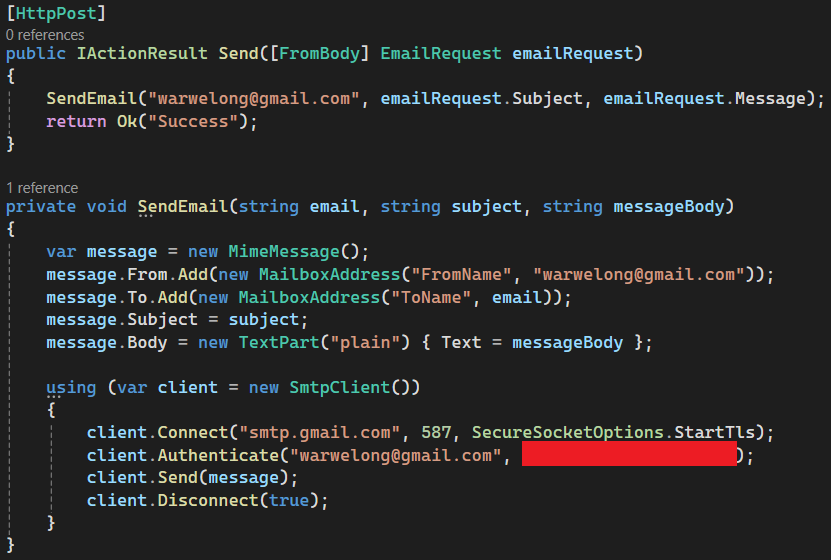
Najważniejszą częścią tego programu jest algorytm szyfrujący stworzony za pomocą paczki NuGet „BCrypt.Net-Next”. Zamienia ona normalne hasło wpisane przez użytkownika w postaci kodu mieszanego i dopiero zaszyfrowane hasło jest zapisywane w bazie danych. Powodem kodowania hasła w bazie, jest zabezpieczenie, jest to jedyna przechowywana informacja, która nie zostanie i tak wyświetlona na stronie. Więc na wypadek dostania się do bazy danych osoba, która zyskała dostęp dalej nie będzie znała hasła do konta administratora. Dzięki temu jedynymi operacjami wymaganymi w tym kontrolerze są „post”.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

Rysunek 12 - funkcja "GenerateJwtToken" z pliku "AuthController.cs"

Dodatkową funkcją jaką spełnia ten kontroler jest tworzenie tokenu uwierzytelniającego do czego służy funkcja „GenerateJwtToken”. Jest ona wywołana na końcu operacji logowania i dzięki niej działa cały system autoryzacji, tylko użytkownik z tokenem zapisanym w lokalnym schowku, jest w stanie wykonywać akcje zarezerwowane dla administratorów.



Rysunek 13- punkt końcowy API z pliku "CommunicationController.cs"

Następnym plikiem do analizy jest „CommunicationController.cs”, jest to jedyny kontroler w moim kodzie, który nie jest odpowiedzialny za operację powiązaną z bazą danych. Zadaniem tego kontrolera jest pośrednictwo w wysyłaniu maila przy wypełnieniu formularza na stronie internetowej.

Kontroler ten działa za pomocą użycia innej paczki NuGet o nazwie „MailKit” umożliwia ona w prosty sposób stworzenie systemu dostarczania wiadomości email przez stronę internetową. W tym wypadku zdecydowałem się użyć maila na platformie Gmail, ze względu na wbudowaną kompatybilność z SMTP (Simple Mail Transfer Protocol), pozwala to na szybką konfigurację całej aplikacji.

Kolejny kontroler jest już bardziej standardowy, jest to „EmployeesController.cs”. Jego zadaniem jest pośredniczenie w edytowaniu tabeli „employees” w bazie danych.

Jedyny punkt końcowy API jaki można znaleźć w tym kontrolerze odpowiada za edytowanie, jest to świadomy wybór wywiązujący się z planów aplikacji. Ze względu na to, że na stronie wyświetlane są i tak tylko kolumny, do których jest przypisane Id = 1. Dzięki temu, aby zmienić skład kadry przez stronę internetową, wystarczy zmienić pierwszy zapis każdej z poszczególnych grup, przez co jedyną operacją zapisaną w tym

kontrolerze jest „put” czyli edytowanie.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 14- punkt końcowy API z pliku "EmployeesController.cs"

Następną częścią aplikacji jest kontroler odpowiedzialny za sekcję galerii na stronie, czyli „GalleryController.cs”. Zawiera on cztery punkty końcowe odpowiadające kolejno za operacje fetch, post i delete. Po szybkim sprawdzeniu można zauważyć, że są dwie punkty końcowe, odpowiadające za operację fetch, jest to zrobione ze względu na dwa warunki, w których zdjęcia z galerii będą pokazywane. W pierwszym wypadku, wszystkie zdjęcia muszą zostać wyświetlone na raz w galerii, w drugim, specyficzne zdjęcia musi się wyświetlić w modalu. Z tego powodu, pierwszy punkt końcowy, ma za zadanie wydobyć wszystkie zdjęcia z bazy danych, kiedy drugi wydobywa zdjęcia za pomocą indexu, którym w tym przypadku jest Id, dzięki temu tylko konkretne zdjęcie będzie pokazane. Dodatkowym, ale zarówno ważnym elementem kodu w tej funkcji jest przypisanie stron do zdjęć. Ze względu na to, że ilość zdjęć w bazie danych będzie rosnąć w miarę czasu użytkowania, nie wszystkie zdjęcia mogą być wyświetlone na raz, muszą zostać podzielone na strony. Każda ze stron ma określoną ilość przedmiotów, które zostaną wyświetlone, a ilość stron zależy bezpośrednio od ilości zdjęć zapisanych w bazie danych.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 15- fragment kodu z "GalleryController.cs" odpowiadający za podzielenie na strony

Przedostatnim kontrolerem do opisania jest „OfferController.cs”. Ze wszystkich kontrolerów jest on najprostszy, nie posiada on żadnych dodatkowych funkcji i służy tylko do wykonywania operacji wydobycia, postowania oraz usuwania. Jest tak ze względu na proste działanie sekcji ofert na stronie, mają one być jednorazowe i niezmienne, w przypadku jakiejś zmiany, mają zostać usunięte a nowe stworzone na ich miejsce.

Ostatnim kontrolerem, który jest zarazem najbardziej złożonym jest „PostController.cs” odpowiada on za zarządzanie postami w sekcji aktualności na stronie internetowej. Posiada on rekordową ilość pięciu operacji: dwa odpowiadające za wydobycie, jeden do postowania, jeden do usuwania i jeden do edytowania. Operacje potrzebne do wydobycia są dwie z prostego względu, potrzebna jest opcja wyświetlania wszystkich postów na raz w formie listy w sekcji aktualności, dodatkowo potrzebna jest opcje wyświetlenia pojedynczego postu na całą stronę, żeby poznać jego detale. Tak samo jak w kontrolerze „GalleryController.cs”, ze względu na to, że postów będzie za dużo, żeby zmieściły się, potrzebne jest podzielenie ich na strony. Dużą częścią kodu operacji postowania, zajmuje sekcja poświęcona wytyczeniu danych podawanych jako obraz (Image). Powodem tego jest forma zapisu obrazów na jaką się zdecydowałem, polega ona na zapisaniu ścieżki, gdzie w plikach strony obraz jest zapisywany, do odczytania tego obrazu, wystarczy ścieżka.

Obraz zawierający tekst, zrzut ekranu, oprogramowanie

Opis wygenerowany automatycznie

Rysunek 16 - operacja zapisania postu z pliku "PostController.cs"

1. Struktura aplikacji
   1. Wstęp

W tej części mam zamiar przedstawić wszystkie aspekty front-endu mojej aplikacji, jakie napotkałem problemy, jak je rozwiązałem oraz jak wszystkie elementy działają i połączenia projektu w jedną całość

* 1. Strony pojedyncze

Pierwszym elementem, na który składa się moja aplikacja są tak zwane strony pojedyncze. Tym mianem określam strony znajdujące się pod ścieżką „src/components” w folderze zawierającym front-end. Ze wdględu na to, że nie są rozbudowanymi elementami, opiszę je dość szybko:

* AboutPage.vue - Jest to plik zawierający sekcję „O nas” na stronie internetowej. Jak sama nazwa sugeruje, strona zawiera podstawowe informacje o przedszkolu oraz mapę z dokładną jego lokacją.
* ContactPage.vue – Na tej stronie poza mailem i telefonem przedszkole, umieszczony jest formularz, który po wypełnieniu można wysłać bezpośrednio na skrzynkę odbiorczą przedszkola.
* MainPage.vue – Strona główna, jedynym jej zadaniem jest przywitanie użytkownika i odesłanie go w stronę bardziej pożądanych przez użytkownika informacji.
  1. Sekcja autoryzacji

Sekcja autoryzacji ma bardzo prostą rolę, ma weryfikować czy użytkownik, który chce się zalogować, ma do tego uprawnienia. Plik „Regiter.vue” jest pierwszym elementem tego systemu, jego rolą jest stworzenie nowego profilu użytkownika. Po wpisaniu nowych danych w formularzu na stronie oraz kliknięciu przycisku, kod na stronie aktywuję metodę w „AuthService.js”, której zadaniem jest połączenie się z back-endem i stworzenie nowego wejścia w bazie danych, w tabeli „users”. Powoduje to stworzenie nowego profilu użytkownika, z hasłem zabezpieczonym w postaci kodu mieszanego (ang. „hash code”).

{{zdjęcie z logowaniem na stronie}}

Nie jest to rozwiązanie, które zostanie w ostateczniej wersji aplikacji, która zostanie oddana klientowi ze względów bezpieczeństwa. Zamiast tego profile administratorów zostaną wygenerowane na przyszłość.

Kolejnym elementem sekcji autoryzacji jest strona „Login.vue”. Jej zadaniem jest umożliwienie pracownikowi przedszkola zalogować się na tą stronę, daje im to dostęp do większości funkcji w innych sekcjach, takie jak: tworzenie i usuwanie postów, modyfikacja listy kadry, wstawianie nowych zdjęć do galerii czy usuwanie ofert. Na stronie trzeba wypełnić mały formularz z loginem i hasłem, po wpisaniu prawidłowych informacji, dojdzie do zalogowania. Na górnym pasku pojawi się informacja, że użytkownik jest zalogowany i zostanie przerzucony do następnej strony w sekcji, czyli „AdminPage.vue”.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Rysunek 17- funkcja "login" z pliku "AuthService.js"

Ostatnią częścią sekcji autoryzacji, jest „AdminPage.vue”, jest to główne centrum zarządzania całą stroną internetową. Jest to miejsce, gdzie można dodać nowe posty, zdjęcia czy oferty, znajdują się tam również odnośniki przenoszące do stron jak „PostListPage.vue” czy „EmployeesPage.vue” gdzie potrzeba bezpośredniego dostępu, żeby je edytować. Do każdego elementu jest też przypisana krótka instrukcja, żeby uniknąć nieporozumienia, nie ważne jaki pracownik przedszkola by zajmował się stroną.

{{opis przycisków}}

Wszystko to jest spięte ze sobą dzięki „AuthRouter.js” dodający ścieżki URL do aplikacji i pozwalający na nawigację między nimi.

* 1. Sekcja kadry

Rolą sekcji kadry jest przedstawienie klientom informacji o dokładnym składzie pracowników przedszkola. Strona jest podzielona na sześć małych sekcji, każda przeznaczona innej grupie kadry. Jedna z nich jest przeznaczona dyrekcji przedszkola, jedna specjalistom (np.: logopeda, nauczyciel tańca czy angielskiego), oraz cztery dla grup przedszkolnych: czerwonej, niebieskiej, żółtej i zielonej.

Każda z tych małych sekcji może być modyfikowana przez administratora poprzez kliknięcie przycisku „Edytuj”, użytkownik dostanie możliwość zmiany zawartości wybranej sekcji. Po wprowadzeniu pożądanych zmian, trzeba wcisnąć „Zapisz” co spowoduje uruchomienie w back-endzie operacji edycji bazy danych „employees”, powodując zapisanie zmian na stałe.

Pomimo małych rozmiarów, ta sekcja zawiera swój router i service, oddzielające funkcje tej strony.

{{zdjęcie skończonej sekcji kardy}}

* 1. Sekcja ofert

Kolejną sekcją projekty jest strona dotycząca ofert. Ma ona prezentować aktualną ofertę przedszkola, która z czasem się zmienia, przez co jest możliwość w każdej chwili dodać lub usunąć ofertę. Każda z nich składa się z dwóch widocznych komponentów, tytułu oraz kontent, które są wyświetlane na stronie. Po zalogowaniu administratorowi pokazuje się przycisk służący do usuwania postu, przy którym się znajduje. Na dole strony znajduje się również przycisk do dodania nowej oferty, jest to taki sam przycisk jak na wcześniej omówionej stronie administratora, jego działanie polega na przekierowaniu użytkownika do „OfferCreationPage.vue”, strony służącej do tworzenia nowych postów.

{{zdjęcie strony z listą ofert}}

Strona do tworzenia ofert działa na podstawie paczki NPM o nazwie „Vue3-editor”, tworzy ona edytor tekstu, który zapisuje takie rzeczy jak formatowanie, stylizacje i wielkość czcionki. Jest to część formularza do wypełnienia, zapisująca kontent. Druga część formularza, znajdująca się nad edytorem tekstu, jest to zwykłe pole do wpisania tekstu, które prosi o tytuł. Ze względu na to, że tytuł jest o wiele mniej skomplikowany uznałem, że z tego względu nie potrzebny jest do tego cały edytor tekstu.

{{ zdjęcie tworzenia ofert}}

Całość spina Router i tak jak w poprzednich sekcjach, mimo tego, że cała sekcja nie jest zbyt obszerna, i tak są potrzebny jest plik serwisowy, zawierający funkcję do wydobycia, postowania i usuwania ofert.

* 1. Galeria

Galeria jest bardzo prostym i znanym konceptem obecnych na wielu stronach internetowych, dodatkowo jest bardzo dobry sposobem, żeby pokazać przedszkole nowym klientom, z tych powodów nie mogło jej zabraknąć na tej stronie. Głównym elementem tej sekcji jest sama galeria, znajdująca się na środku ekranu, gdzie każde ze zdjęć jest pokazane w formie kafelka.

{{zdjęcie kafelków w galerii}}

Jeśli odwiedzający użytkownik kliknie na którekolwiek ze zdjęć, zostanie one wyświetlone na cały ekran, za pomocą wcześniej napisanego modalu. Modal ten zawiera cztery elementy dla użytkownika, dwie strzałki wskazujące w prawo i lewo, służące do nawigacji między zdjęciami bez wyjścia z modalu, przycisk „Zamknij” służący do zamknięcia modalu oraz samo zdjęcie, w pełnej rozdzielczości na środku ekranu. Istnieje też jedna opcja dostępna tylko dla administratora, dostępna jedynie po zalogowaniu, służy ona do usunięcia danego zdjęcia. Wracając do głównej strony galerii, na dole znajdują się dwie dodatkowe sekcje. Pierwsza z nich jest poświęcona zmienianiu stron galerii, jeśli galeria ma mniej niż piętnaście zdjęć te przyciski się nie pojawią, ponieważ następna strona nie jest potrzebna. Ostatnią rzeczą widoczną na stronie, jest przycisk widoczny tylko dla zalogowanych użytkowników służący do dodania nowego zdjęcia, taki sam przycisk jest też dostępny na wcześniej opisanej stronie administracji.

{{zdjęcie modalu ze zdjęciem}}

Ze względu na swoją budowę, plik serwisowy jest jednym z bardziej rozbudowanych w tym projekcie, zawiera on funkcje postowania nowego zdjęcia, usunięcia go oraz wyświetlenia zarówno wszystkich zdjęć (używane w galerii) oraz pojedynczych zdjęć (używany w modalu).

* 1. Sekcja aktualności

Ostatnią, ale zarazem największą sekcją tej aplikacji jest sekcja aktualności, jest to najważniejsza wymagana przez klienta funkcjonalność, pozwalająca mu zamieszczać na stronie posty dla klientów do przeczytania. Pierwszym elementem tej sekcji jest „PostListPage.vue”, stanowi ona listę postów podzieloną na strony tak jak galeria opisana wcześniej. Na każdej stronie tej listy znajduje się po pięć postów, na których jest wyświetlony tytuł danego posta, początek jego kontent, oraz zdjęcie, jeśli post owe zawiera. Przy każdym poście znajduje się też przycisk „Czytaj dalej”, co otworzy stronę, która jest następną częścią tej sekcji „PostDetails.vue”.

Ta strona jest przeznaczona do wyświetlenia przypisanego do tego postu zdjęcia, pod którym jest wyświetlona cała zawartość do czytania. Dla normalnych użytkowników, jedyną dostępną opcją będzie opuszczenie strony po przeczytaniu jednak, jeśli administrator jest zalogowany zobaczy on dwie dodatkowe opcje: jedna to edytowanie posta, a druga to jego usunięcie.

Ostatnimi dwoma elementami są strony, które można by nazwać bliźniaczymi. Jest to „PostCreationPage.vue” i „PostEditingPage.vue” obydwie z nich, mają wręcz taką samą budowę, lecz spełniają zupełnie inny cel. Pierwsza z nich jest dostępna z listy wszystkich postów i otwiera stronę z pustym oknem tekstowym stworzonym za pomocą „Vue3-editor”. Po wpisaniu tytułu w górnym polu tekstowym i treści postu w tym poniżej i naciśnięciu „Zapisz” post zostanie za pomocą serwisu wysłany do bazy danych i tam zapisany w całości. Jeśli użytkownik by chciał jest też możliwość dodania zdjęcia, które będzie się wyświetlać na liście oraz jako główny element postu po otworzeniu jego detali. Druga strona za to, jest otwierana właśnie po wejściu w wspominane wcześniej detale i kliknięciu przycisku do edytowania tego postu. Otworzy nam się bardzo podobna strona, z tym samym edytorem tekstu ale tym razem nie będzie pusty, będzie miał w sobie treść wybranego posta, udostępniona bez problemu do edycji.

1. Bibliografia