

Dokumentacja projektu

Temat: Biuro podróży Vacationey

Przedmiot: Techniki internetowe



Opracowali: Wojciech Seweryn, Mateusz Mulka, Anastazja Spólnik

Kierunek: Informatyka i Ekonometria

Rok: 2021/22

Spis treści

[Cel projektu 3](#_Toc106399604)

[Zastosowanie biznesowe 3](#_Toc106399605)

[Baza danych i diagram 3](#_Toc106399606)

[Modele 4](#_Toc106399607)

[Kontrolery 5](#_Toc106399608)

[Widoki 7](#_Toc106399609)

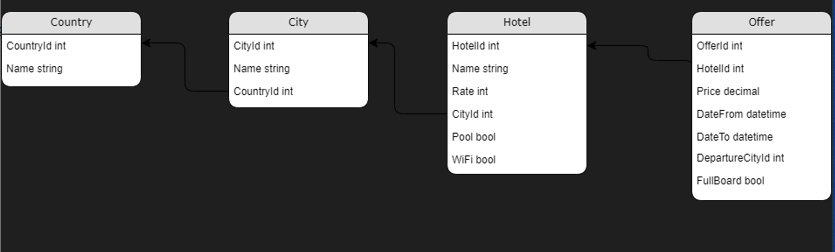
# Cel projektu

Celem projektu było stworzenie funkcjonalnej strony internetowej dla biura podróży połączonej z bazą danych. Chcieliśmy w jak najbardziej przejrzysty sposób zaprezentować oferty wakacyjnych wyjazdów biura, ułatwić wyszukiwanie konkretnych preferencji klienta. Pod względem wizualnym aplikacja jest przejrzysta i prosta w obsłudze. Umożliwia łatwe wyszukiwanie ofert i hoteli oraz dodawanie, usuwanie i modyfikowanie istniejących.

# Zastosowanie biznesowe

Aplikacja mogłaby posłużyć profesjonalnemu biuru podróży do zbierania zamówień od swoich klientów oraz do dodawania nowych ofert do katalogu. Jej prosta budowa oraz klarowność znacząco ułatwiają korzystanie. Strona internetowa zawiera przyjazne i czytelne widoki. Dodatkowo, obsługa błędów uniemożliwia zapis niepoprawnych danych w bazie.

# Baza danych i diagram



Baza danych składa się z 4 tabel: City, Country, Hotel oraz Offer. Tabele połączone są ze sobą relacjami jeden do wielu. Jest to prosty schemat bazy, zapewniający kontrolę nad encjami  
i możliwość łatwego filtrowania i sortowania. Oferta zawiera id hotelu, cenę, datę przyjazdu  
i odjazdu, miasto, z którego wylatujemy oraz informację, czy wybieramy opcję z pełnym wyżywieniem. W tabeli Hotel znajdziemy informacje o nazwie, ocenie, mieście, w którym hotel się znajduje oraz czy posiada basen i dostęp do Wi-Fi. Tabele City oraz Country zawierają nazwy miast i krajów.

# Modele

Podstawowe modele wykorzystywane do tworzenia bazy danych

* City.cs,
* Country.cs
* Offer.cs
* Hotel.cs

Obraz zawierający tekst, sprzęt elektroniczny

Opis wygenerowany automatycznie

W pliku DatabaseContext.cs tworzymy model bazy danych w oparciu o podejście Code First  
z użyciem Entity Framework Core określając 4 DbSets. Klasa dziedziczy po IdentityDbContext<IdentityUser> co generuje dodatkowe tabele umożliwiające autentykację (logowanie, rejestrowanie, wylogowywanie). Klasa SeedModel służy do zapełniania pustej bazy danych poprzez stworzenie 500 ofert, na podstawie losowych wcześniej stworzonych miast, państw oraz hoteli. ViewModels służą do wyświetlania danych, nie są zapisywane w bazie danynch, są przekazywane jako model do Widoków. CreateViewModels wykorzystywane są do metod Create i Edit, natomiast ViewModels do Details, Index i Delete. Dzięki nim możemy w prosty sposób wyświetlić wiele elementów w widoku.

Przykładowo modele dla Hotel:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zapewnia dostęp do wszystkich pól potrzebnych do wyświetlania informacji o Hotelu

Obraz zawierający tekst

Opis wygenerowany automatycznie

Zapewnia dostęp do wszystkich pól potrzebnych do edycji lub tworzenia klasy Hotel

# Kontrolery

Kontrolery odpowiednio reagują na wczytane dane, zarządzają i aktualizują model, odświeżając widoki. W naszym projekcie stworzyliśmy 5 kontrolerów: CityController, CountryController, HomeController (domyślny dla strony początkowej), HotelController oraz OfferController

Obraz zawierający tekst

Opis wygenerowany automatycznie

Każdy z nich dziedziczy po klasie Controller. W kontrolerze CityController możemy znaleźć metody takie jak: Create (zwracająca widok do tworzenia miasta), Edit (możliwość edycji istniejącego miasta), Delete (usuwanie konkretnego miasta) oraz Index (zwraca model z listą miast) Bardzo podobną strukturę ma CountryController. W kontrolerach hotelu i oferty oprócz wymienionych funkcji pojawia się też metoda get Details, pokazująca szczegóły hotelu oraz oferty. HomeController odpowiedzialny jest jedynie za zwrócenie widoku początkowego ze stroną główną.

Metoda Index przyjmuje parametry, podane w adresie url, w kontrolerze Hotel przyjmuje 3 różne parametry: citySearch (filtruje tabele bazy danych, zwraca tylko hotele z określonego miasta), sortOrder (ascending, descending), sortuje hotele według oceny, page – na jednej stronie znajduje się 10 ofert.



Metoda GET Details przyjmuje jako parametr id – klucz główny w tabeli, zwraca widok z modelem klasy CreateViewModel



Metoda GET Create zwraca CreateViewModel



Metoda POST Create tworzy obiekt klasy na podstawie danych wprowadzonych przez użytkownika i zapisuje go w bazie danych



Metoda GET Edit przyjmuje jako parametr id – klucz główny w tabeli, zwraca widok z modelem klasy CreateViewModel



Metoda POST Edit modyfikuje obiekt klasy na podstawie danych wprowadzonych przez użytkownika i zapisuje go w bazie danych



Metoda GET Delete Edit przyjmuje jako parametr id – klucz główny w tabeli, zwraca widok z modelem klasy ViewModel, pytając użytkownika o potwierdzenie



Metoda POST Delete przyjmuje jako parametr id – klucz główny w tabeli, usuwa obiekt z bazy danych



Metody dla każdego kontrolera działają na takiej samej zasadzie

# Widoki

Obraz zawierający tekst, sprzęt elektroniczny

Opis wygenerowany automatycznieObraz zawierający tekst, sprzęt elektroniczny, ekran, zrzut ekranu

Opis wygenerowany automatycznieWidoki są częścią modelu MVC odpowiedzialną za prezentowanie danych użytkownikowi. Widoki dla miast, państw, ofert oraz hoteli zawierają widoki do tworzenia, usuwania, edytowania oraz główny widok index odpowiedzialny za pobranie, sortowanie i wyświetlenie danych.

Widok strony początkowej Home ma tylko plik Index ze stroną startową

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst, sprzęt elektroniczny, zrzut ekranu

Opis wygenerowany automatycznieW folderze Shared znajdują się widoki łączone, wykorzystywane przez główny layout, takie jak n=Navbar, Footer (widoczne dla każdego Widoku) i Error (customowany widok, zwracany dla wszystkich niepoprawnych operacji ze szczegółami).

* Index – Przyjmuje jako model kolekcję obiektów typu ViewModel dla określonej klasy, na samej górze zawiera filtry, generuje widok częściowy dla każdego obiektu z listy. Na dole przyciski do zmiany strony
* Details – Przyjmuje jako model obiekt typu ViewModel dla określonej klasy. Pokazuje szczegóły obiektu na całej stronie.
* Delete - Przyjmuje jako model obiekt typu ViewModel dla określonej klasy. Pokazuje to samo co widok Details, tylko z możliwością usunięcia.
* Edit - Przyjmuje jako model obiekt typu CreateViewModel dla określonej klasy. Pokazuje formularz do edytowania modelu.
* Create - Przyjmuje jako model obiekt typu CreateViewModel dla określonej klasy. Pokazuje pusty formularz do tworzenia modelu.

Dla każdego Kontrolera widoki są skonstruowane w taki sam sposób przy drobnych wizualnych zmianach.

Kod źródłowy i instrukcja obsługi znajdują się na stronie <https://github.com/sewcio543/vacationey>