Politechnika Śląska

Dokumentacja projektu

2022/2023

Baza lokali

Projekt zaliczeniowy Programowanie obiektowe i graficzne

Kierunek: Informatyka

Członkowie zespołu:
Bartosz Machniak
Bartłomiej Repeć
Łukasz Szary

Gliwice, 2022/2023

Spis treści

1 Wprowadzenie

1.1 Cel projektu

Celem projektu jest stworzenie aplikacji sieciowej, która umożliwia użytkownikowi przeszukiwanie bazy danych lokali. Aplikacja będzie posiadać filtr umożliwiający wybieranie lokali po: cenach dań, promocjach, miastach i kuchni. Użytkownik będzie mieć też możliwość oglądania opinii i dań w danej restauracji.

1.2 Role w projekcie

Bartosz Machniak - Front-end
Bartłomiej Repeć - Back-end
Łukasz Szary - API

2 Założenia projektowe

2.1 Opis wymagań

1. Baza danych:
    - Baza danych będzie przechowywać informacje o restauracjach, takich jak nazwa, adres, menu, oceny, recenzje oraz promocje.
    - Baza danych będzie zaprojektowana w sposób umożliwiający efektywne wyszukiwanie restauracji na podstawie różnych kryteriów, takich jak lokalizacja, kuchnia, oceny, średnia cena dostępnych produktów i istnienie promocji
    - Baza danych będzie miała relacje między różnymi encjami, takimi jak restauracje, menu, opinie oraz promocje.
2. Back-end:
    - Back-end zostanie zaimplementowany w języku C# przy użyciu odpowiedniego frameworka, takiego jak ASP.NET.
    - Back-end będzie odpowiedzialny za przeglądanie danych z bazy i wyszukiwanie z odpowiednimi kryteriami.
    - Back-end będzie udostępniał interfejs programistyczny (API) dla front-endu, który umożliwi komunikację między front-endem a bazą danych.

3. Front-end:

    - Front-end zostanie zaimplementowany w języku JavaScript przy użyciu frameworka JQuery.
    - Front-end będzie odpowiedzialny za interakcję z użytkownikiem i prezentację danych z bazy danych.
    - Front-end będzie umożliwiał użytkownikom wyszukiwanie restauracji na podstawie różnych kryteriów oraz przeglądanie informacji o restauracjach

2.2 Instrukcja obsługi aplikacji

1. Strona główna:

    (a) Po wejściu na stronę, zostajesz przekierowany do strony głównej aplikacji.

    (b) W sekcji po prawej stronie wybierz odpowiednie filtry do wyszukiwania takie jak to, czy są promocje, miasta do przeszukania i kuchnia, która cię interesuje

    (c) W pasku u góry opcjonalnie wpisz słowa, po jakich dodatkowo chciałbyś wyszukać restauracje.

    (d) Po wybraniu filtrów kliknij szukaj, aby wyświetlić wybrane restauracje.

2. Wyszukane restauracje:

    (a) W sekcji na środku strony pojawiają się odpowiednie restauracje zgodne z wyszukiwaniami.

    (b) W elemencie każdej restauracji możesz zobaczyć takie informacje jak nazwa, miasto, adres średnia ocena oraz średnia cena dań w restauracji.

    (c) Kliknij w wybraną restaurację, aby przejść do strony szczegółowej.

3. Widok szczegółów restauracji:

    (a) Na stronie pojawiają się trzy przyciski: 'Menu', 'Opinie' oraz 'Promocje'.

    (b) Po kliknięciu w przycisk 'Opinie' wyświetlą się nam bloki z opiniami różnych użytkowników. Zobaczymy tam treść opinii, ocenę, autora, oraz datę wystawienia opinii.

(c) Po kliknięciu w przycisk 'Menu' wyświetli się oferta danej restauracji. Zobaczymy tam informacje o daniach takie jak cena, nazwa, oraz szczegółowy opis.

(d) Po kliknięciu przycisku 'Promocje' wyświetlą nam się promocje, które są aktualnie dostępne w wybranej restauracji

3 Wnioski

3.1 Podsumowanie projektu

- Podział obowiązków został zrealizowany bardzo dobrze.
- Konwencja nazywania metod w kontrolerach w API mogła zostać ustalona lepsza.
- Część operacji, które wykonuje API można było wykonać w SQL za pomocą widoków i tylko je pobierać.
- Część commit'ów niestety ma nazwy, które nic nie mówią.

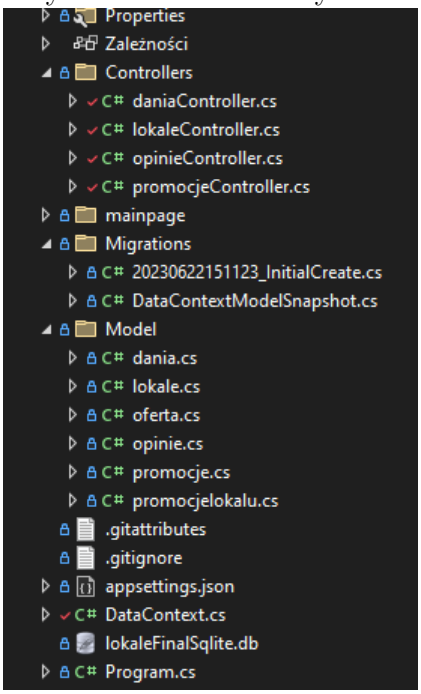
3.2 Potencjał rozwoju

- Powstałą aplikację można rozszerzyć o funkcjonalność polegającą na proponowaniu użytkownikowi lokali podobnych to tych, które wyszukuje. Przykładowo jeśli ktoś szuka lokali z kuchnią włoską w Katowicach w przedziale cenowym można mu polecić lokale w Katowicach z kuchnią indyjską w podobnym zakresie cenowym.
- Bazę danych można uzupełnić o dodatkowe lokale, dania, opinie.
- Można dodać profile użytkowników, za ich pomocą można przechowywać historię wyszukiwań, dodawać własne opinie, spis wystawionych opinii.

4 Dokumentacja kodu

4.1 Podział na pliki

Rysunek 1: Kontrolery API



- W folderze ***Controllers*** znajdują się pliki kontrolerów API kolejno dla obiektów klas ***dania***, ***lokale***, ***opinie***, ***promocje***.
- W folderze ***Migrations*** znajdują się pliki wygenerowane automatycznie przez Entity Framework, przy migracji bazy danych.
- W folderze ***Model*** znajdują się poszczególne klasy, które odwzorowywują encje poszczególnych tabel w bazie.
- ***DataContext.cs*** zawiera kontekst bazy danych - spaja klasy z folderu ***Model*** z danymi bazy. ***lokaleFinalSqlite.db*** to baza danych.

4.2 Opis kodu

API linki URL

- api/dania - Zwraca wszystkie dania z bazy danych.

- api/dania/{id} - Zwraca danie o danym id(int).
- api/dania/GetAvdCenaForLokal/{id} - Zwraca średnią cenę(jako string) dań dla danego id lokalu(int).
- api/dania/GetDaniaByLokaleId/{id} - Zwraca listę dań dla danego id lokalu(int).
- api/lokale - Zwraca wszystkie lokale z bazy danych.
- api/lokale/{id} - Zwraca lokal o danym id(int).
- api/lokale/GetlokaleByKuchniaMiastoPromocjaCenaScopePhrase/{kuchnia},{miasto}, {whetherPromocja},{beginScopeCena},{endScopeCena},{phrase}

  Zwraca lokale oraz średnią cene ich potraw i średnią opinię. Spełniające warunki, które określone są parametrami(kuchnia, miasto, whetherPromocja, beginScopeCena, endScopeCena), i których połączona nazwa, kuchnia, adres, miasto zawiera przynajmniej jedno ze słów ze zdania podanego jako argument(Phrase) . Parametry kuchna(string), miasto(string) określają zbiór kuchni/miast, nazwt pisane ciągiem, przykładowe ich formy to: WłoskaIndyjskaPolska, PolskaIndyjska, Włoska , KatowiceGliwice, Zabrze. Wartość parametrów "Any"lub null reprezentuje wszystkie miasta/kuchnie w bazie.

  Paramet whetherPromocja(bool) dla wartości true wypisze lokale z przynajmiej jedną promocją, dla false pominie ten warunek.

  Parametry beginScopeCena i endScopeCena(double) określają przedział cenowy, który musi spełnić przynajmniej jedno danie w lokalu, wartość null jest traktowana dla beginScopeCena jako 0 a dla endScopeCena jako maksymalna wartość dla doubla.

  Przykład:
  api/lokale/GetlokaleByKuchniaMiastoPromocjaCenaScope/
  PolskaIndyjska,Katowice,true,0,50
  URL wypisze lokale w Katowicach serwujące dania włoskie lub indyjskie, z przynajmniej jedną promocją, z przynajmniej jednym daniem w zakresie cenowym od 0 do 50zł.
- api/lokale/GetMiasta - Zwraca listę różnych miast, w których są lokale.
- api/lokale/GetKuchnie - Zwraca listę różnych kuchni w lokalach.

- api/lokale/GetlokaleByPhrase/phrase - Zwraca lokale, których połączona nazwa, kuchnia, adres, miasto zawiera przynajmniej jedno ze słów ze zdania podanego jako argument(string).
- api/promocje - Zwraca wszystkie promocje.
- api/promocje/{id} - Zwraca promocje o danym id(int).
- api/promocje/GetPromocjeByLokaleId/{id} - Zwraca promocje dla lokalu o danym id(int).
- api/opinie - Zwraca wszystkie opinie.
- api/opinie/{id} - Zwraca opinie o danym id(int).
- api/opinie/GetOpiniebyLokaleId/{lokaleId} - Zwraca liste opini dla danego id lokalu(int).
- api/opinie/GetAvgOceneByLokaleId/{lokaleId} - Zwraca średnią wartość(jako string) ocen dla lokalu o danym id(int).