

# Sprawozdanie z pracy projektowej: Amora

Jakub Trznadel Mateusz Tecza, Roman Rusinek

June 7, 2024

## 1 Opis funkcjonalny systemu

- Amora to platforma internetowa umożliwiająca użytkownikom znalezienie potencjalnych partnerów na podstawie ich preferencji płciowych.
- Użytkownicy mogą zarejestrować się, utworzyć profil, szukać "match'y", zobaczyć swój profil.
- Poprzez podstronę powiadomienia użytkownicy mogą akceptować "match'e" z innymi użytkownikami oraz przeglądać poprzednie "match'e" w zakładce "Previous Matches".
- Użytkownik może odnaleźć numer telefonu swojej pary i poprzez niego nawiązać z nią kontakt. Do pomocy w rozmowie może wykorzystać kawały, które znajdują się w tej samej zakładce.
- Do dyspozycji użytkownika, nawet niezalogowanego, jest sekcja FAQ oraz formularz zgłoszeniowy.

## 2 Opis technologiczny

- Projekt został zbudowany w oparciu o framework ASP.NET Core MVC 8.0.
- Wykorzystujemy technologie HTML, CSS i JavaScript do budowy interfejsu użytkownika.
- Jako bazy danych używamy MsSQL Server.
- Do zarządzania zależnościami używamy menedżera pakietów NuGet.
- System cache jest implementowany przy użyciu wbudowanego w ASP.NET Core mechanizmu pamięci podręcznej.
- Do mapowania obiektowo-relacyjnego (ORM) wykorzystujemy Entity Framework Core.
- Jako framework CSS używamy Bootstrap 5.1.0.

### 3 Zagadnienia kwalifikacyjne

1. Framework MVC: Wykorzystujemy framework ASP.NET Core MVC do budowy backendu aplikacji.  
(Model) Model danych - opis struktur danych i powiązań pomiędzy nimi  
(View) Interfejs, czyli to co widzi użytkownik  
(Controller) Logika działania - powiązania między zdarzeniami zachodzącymi w systemie
2. Framework CSS: Do stylizacji interfejsu użytkownika używamy frameworka Bootstrap.  
Bootstrap to popularny framework front-endowy, który zapewnia zestaw narzędzi, komponentów i stylów CSS.
3. Baza danych: Projekt wykorzystuje Microsoft SQL jako baze danych.
4. Cache: Wdrożono mechanizm pamięci podręcznej do optymalizacji działania aplikacji.  
System Cache jest wykorzystywany w naszym projekcie do zapisywania oraz wyświetlania danych z profilu użytkownika tak aby za każdym wejściem na profil projekt nie musiał na nowo pobierać danych z bazy danych.
5. Dependency manager: Do zarządzania zależnościami aplikacji używamy menedżera pakietów NuGet.  
NuGet jest menedżerem pakietów dla platformy .NET, który umożliwia łatwe dodawanie, usuwanie i aktualizowanie zależności bibliotek i narzędzi do projektów, co ułatwia zarządzanie zależnościami i zapewnia ponowne wykorzystanie kodu.
6. HTML: Szkielet aplikacji internetowej został zbudowany zgodnie z standardami HTML.  
HTML służy do strukturyzowania treści na stronach internetowych poprzez definiowanie różnych elementów, takich jak nagłówki, paragrafy, listy, obrazy, linki etc.

7. CSS: Wykorzystujemy arkusze stylów CSS do ostylewania aplikacji.  
W naszej aplikacji korzystaliśmy z CSS do m.in. dodawania zdjęcia w tle strony, oraz do dokładniejszej konfiguracji wyglądu strony.
8. JavaScript: W aplikacji użyto JavaScript do uinteraktywienia interfejsu użytkownika.  
W naszym projekcie JavaScript używany jest do dodania wyskakującego okna z potwierdzeniem chęci wylogowania się oraz przy dodawaniu zdjęcia podczas rejestracji.
9. Routing: Wykorzystujemy routing i tzw. pretty URLs dla estetycznych adresów URL.
10. ORM: Do mapowania obiektowo-relacyjnego używamy Entity Framework Core.  
Entity Framework umożliwia pracę z danymi w bazie danych za pomocą obiektów i zapytań LINQ, co eliminuje potrzebę pisania bezpośrednich zapytań SQL.
11. Uwierzytelnianie: Uwierzytelnianie akcji w systemie polega na identyfikacji zalogowanego użytkownika przez system, co pozwala mu dostosować działania i udostępnić odpowiednie funkcjonalności zgodnie z jego uprawnieniami.
12. Lokalizacja: W banerze strony jest opcja wyboru języka, między polskim a angielskim. Do tłumaczenia strony korzystaliśmy z plików o rozszerzeniu .resx
13. Mailing: Mailing to proces wysyłania wiadomości e-mail z aplikacji internetowej do określonych użytkowników lub grupy odbiorców. Używamy SMTP do wysyłania powiadomienia o utworzonym koncie. Mamy również możliwość wysłania maila ze zgłoszeniem.
14. Formularze: Formularze to interaktywne elementy interfejsu użytkownika, które umożliwiają użytkownikom przysyłanie danych do aplikacji poprzez wprowadzanie informacji w pola tekstowe, wybieranie opcji, czy przysyłanie plików.  
Korzystamy z formularze poprzez rejestrację, logowanie, wybór płci matchu. wysyłanie zgłoszenia oraz edycji hobby.

15. Asynchroniczne interakcje: Są procesami, w których zadania i odpowiedzi między klientem a serwerem odbywają się niezależnie, umożliwiając wykonywanie innych operacji w trakcie oczekiwania na odpowiedź. Używamy ich podczas wrzucania zdjęcia podczas rejestracji oraz podczas wysyłania e-maila.
16. konsumpcja API: Konsumpcja API polega na pobieraniu danych lub wykonywaniu operacji z zewnętrznego interfejsu programistycznego. Korzystamy z Jokes API które generuje losowe żarty.
17. RWD: Responsywny front-end to taki, który automatycznie dostosowuje się do różnych rozmiarów i typów urządzeń, zapewniając optymalne doświadczenie użytkownika na każdym ekranie.

## 4 Instrukcja uruchomienia projektu lokalnie

1. Pobranie kodu źródłowego: Sklonuj repozytorium z GitHuba na swój lokalny komputer. Możesz to zrobić za pomocą polecenia `git clone [adres_repozytorium]`. Repozytorium projektu
2. Otwarcie projektu: Otwórz pobrany projekt w wybranym edytorze kodu, na przykład Visual Studio lub innym.
3. Instalacja zależności: Upewnij się, że zainstalowane są wszystkie zależności projektu. Możesz to zrobić za pomocą menadżera pakietów NuGet. W konsoli NuGet należy wpisać komendę `update-database`.
4. Uruchomienie aplikacji: Uruchom projekt poprzez `ctrl+F5` (Visual Studio) lub poprzez odpowiadającą opcję w innym edytorze.
5. Otwarcie w przeglądarce: Po zakończeniu procesu uruchamiania projekt będzie dostępny pod adresem `http://localhost:port`.
6. Testowanie funkcjonalności: Przetestuj różne funkcje aplikacji, takie jak logowanie, rejestracja, przeglądanie profili, itp., aby upewnić się, że aplikacja działa poprawnie.

## 5 Wnioski projektowe

Korzystanie z .NET było fascynującym doświadczeniem, choć nie obyło się bez wyzwań, szczególnie podczas pierwszych kroków z Frameworkiem MVC. Metoda prób i błędów była naszą codziennością, a każde napotkane trudności stawały się szansą na naukę. Odkrywanie Frameworka CSS - Bootstrap stanowiło niezwykle interesująca część, oferując narzędzia, których wcześniej nie mieliśmy okazji używać, co znacząco ułatwiło rozwój naszego projektu. Podsumowując, choć okres pracy był wymagający, zdobyte doświadczenie będzie z nami na długo, pozostawiając nam cenne lekcje na przyszłość.