

Imię i Nazwisko	Kierunek	Numer albumu
Konrad Lenart	Inżynieria Obliczeniowa	299631

1. Temat projektu:

- a. Tematem projektu jest menadżer haseł, program, który uzyskuje dostęp do dwóch baz danych:
 - i. Baza użytkowników,
 - ii. Baza haseł
- b. Każdy użytkownik posiada unikalne ID. Na tej podstawie określone jest (podczas dodawania hasła) do jakiego użytkownika dane hasło należy.
- c. Program wyposażony jest w logowanie, które wykonane pomyślnie daje nam dostęp do JWT (Jason Web Token). Na podstawie tokenu, użytkownik uzyskuje autoryzację do pewnych metod. Dostęp do metod jest różny dla roli admina oraz użytkownika.

2. Opis implementacji:

- a. Program napisany został w całości w C#, z wykorzystaniem ASP.NET, .NET3.1, AutoMapper'a oraz EntityFramework.
- b. Program składa się z dwóch kontrolerów (w tej wersji, docelowo program będzie rozbudowany o kolejne): 1. UserController, 2. PasswdController. UserController jest odpowiedzialny za tworzenie nowych użytkowników, uzyskiwanie informacji na temat istniejących, za logowanie użytkowników oraz ich uwierzytelnianie. PasswdController odpowiada za tworzenie nowych haseł dla autoryzowanych użytkowników oraz zwracanie już istniejących na podstawie odpowiednich danych przesłanych w zapytaniu.
- c. Architektura:
 - i. Controller - do danego kontrolera trafiają odpowiednie zapytania w protokole HTTP, kontroler także udziela odpowiedzi na nie.
 - ii. Repository - odpowiada za implementację dostępu do bazy danych (np. w przypadku testów jednostkowych możemy użyć innej implementacji niż w przypadku normalnego działania, nie używając silnika bazy danych), posiada kontekst, wobec którego padają zapytania.
 - iii. Data Access (Database Context) - odpowiada za komunikację z bazą danych, najpierw dane są zapisywane na serwerze i nie muszą od razu trafić do bazy danych, jak wiemy bardzo częste otwieranie i zamykanie połączeń z bazą danych nie jest zbyt wydajne. Dzięki takiemu rozwiązaniu mamy wiedzę na temat tego, np. że stworzono nowego użytkownika zanim jego dane trafią do bazy danych.
 - iv. Model - odpowiada za odpowiednią reprezentację zmiennych np. User, Passwd.
 - v. DTO - Data Transfer Object, są to obiekty, które przypominają modele, posiadają tylko takie pola, które w danym kontekście chcemy wykorzystać. np. Jeżeli użytkownik się zalogował to nie zwracamy mu w odpowiedzi hasła, które podał ponieważ jest to niebezpieczne. Zwracamy okrojoną część modelu. W zależności od zapytania używamy różnych DTO.

3. Przykładowe endpointy:

- a. <http://localhost:5001/api/users> - dla Admina, zwraca wszystkich użytkowników
- b. <http://localhost:5001/api/users/login> - dla użytkownika, pozwala na zalogowanie się
- c. <http://localhost:5001/api/passwd/mypasswds> - dla użytkownika, pozwala na zwrócenie posiadanych haseł