

# „Zadaniator”

## DOKUMENTACJA TECHNICZNA

**Tematyka:** Aplikacja do tworzenia i przeglądania listy zadań.

**Przedmiot:** Programowanie zaawansowane

**Grupa:** K14

**Członkowie grupy:** Miklas Mikołaj  
Kowalewska Paulina

## 1. Wstęp

Aplikacja „Zadaniator” ma służyć do sprawnego zarządzania zadaniami użytkownika. Będzie ona umożliwiała dodawanie, edycję oraz usuwanie zadań. Projekt jest napisany w C# z wykorzystaniem frameworka ASP.NET i technologii webowych (HTML, CSS). Baza danych oparta będzie na platformie MySQL.

## 2. Do uruchomienia naszej aplikacji wymagane są:

- Urządzenie z dostępem do internetu

## 3. Instalacja

- pobierz pliki projektu z repozytorium git <https://github.com/paulinak080/Zadaniator>
- otwórz projekt w programie Visual Studio
- uruchom aplikację w środowisku VS, by wyświetlić kody źródłowe.  
Wejdź w link: <https://zadaniator.azurewebsites.net/> by wyświetlić opublikowaną stronę.

## 4. Funkcjonalności

- rejestracja – odbywa się za pomocą ustawienia maila oraz hasła,
- logowanie,
- możliwość dodawania, usuwania oraz edycji zadań
- wyświetlanie szczegółów zadania,
- wyświetlanie listy zadań użytkownika,
- możliwość odhaczenia wykonanych zadań,
- przeszukiwanie listy zadań użytkownika za pomocą nazwy zadania
- przeglądanie motywacyjnych cytatów
- wylogowanie,

## 5. Opis funkcjonalności stron:

### a) Strona główna – publiczna

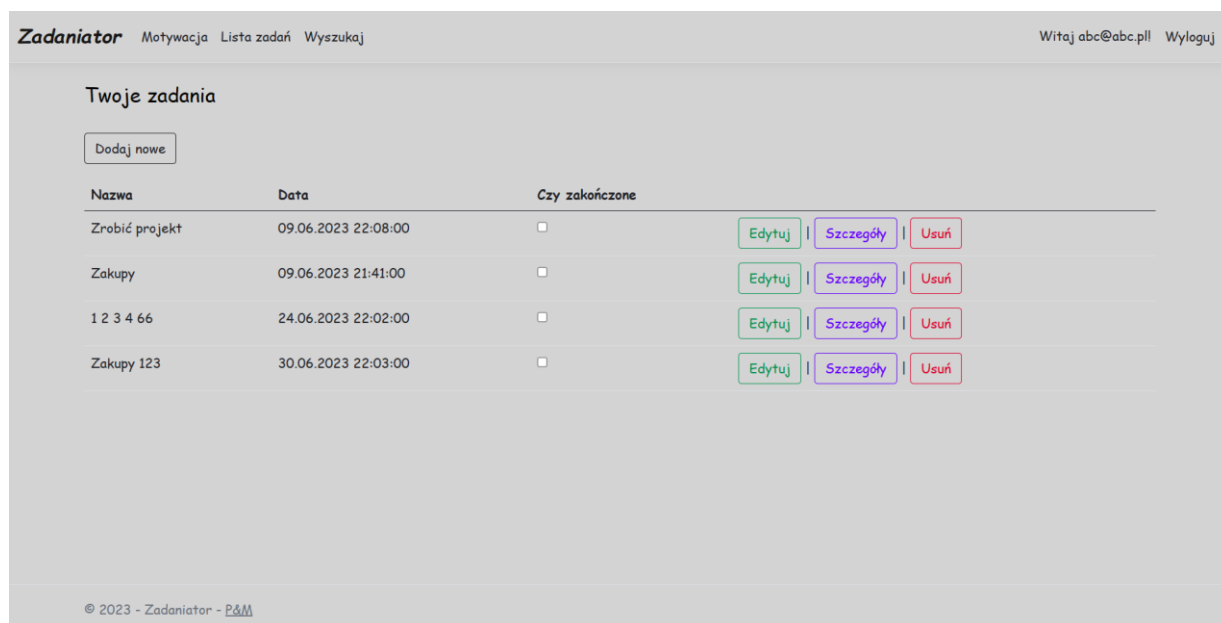
W pasku nawigacyjnym – mamy możliwość przejścia na stronę główną, na stronę z cytatami, założenia konta oraz zalogowania się na utworzone wcześniej konto.

Po zalogowaniu dostępne są również zakładki „Lista zadań” oraz „Wyszukaj”

Dalsza część strony głównej przedstawia temat przewodni aplikacji oraz zachęca do założenia konta.



- b) Podstrona „Motywacja” – publiczna.  
Jest to strona dostępna dla niezalogowanych użytkowników, wyświetlająca motywujące cytaty.
- c) Podstrona „Lista zadań” – prywatna  
Zalogowani użytkownicy mogą tu wyświetlać stworzoną przez siebie listę zadań.



Strona oferuje opcje:

- „Dodaj nowe” - tworzenie nowego zadania , uzupełniając rubryki nazwa, opis oraz planowana data zakończenia zadania.

## Dodaj nowe zadanie

Nazwa

Opis

Data

UserId

[Dodaj](#) | [Powrót do listy](#)

- „Edytuj” – edytować można nazwę, opis oraz datę. Umieszczona jest tu również dodatkowa funkcja oznaczenia zadania jako zakończone.

## Edytuj

Nazwa

Opis

Data

☐ Czy zakończone

[Zapisz](#) | [Powrót do listy](#)

- „Szczegóły” – wyświetla szczegóły zadania. Z tego poziomu można bezpośrednio przejść do edycji.

## Szczegóły

Nazwa	Zrobić projekt
Opis	Na programowanie zaawansowane
Data	09.06.2023 22:08:00
Czy zakończone	<input type="checkbox"/>
User	77035866-761c-4d2b-9f0d-735f5f7db177

[Edytuj](#) | [Powrót do listy](#)

- „Usuń” – usuwanie zadania.

## Usuń

Czy na pewno chcesz usunąć to zadanie?

Nazwa	Zrobić projekt
Opis	Na programowanie zaawansowane
Data	09.06.2023 22:08:00
Czy zakończone	<input type="checkbox"/>
User	77035866-761c-4d2b-9f0d-735f5f7db177

Usuń | Powrót do listy

Z każdej z wymienionych powyżej opcji można przejść bezpośrednio z powrotem do listy zadań.

- d) Podstrona „Wyszukaj” – prywatna  
Umożliwia użytkownikowi przeszukanie swojej listy zadań za pomocą nazwy zadania. Jest to funkcja przydatna w przypadku posiadania długiej listy zadań.

## 6. Opis struktury bazy danych

Baza danych aplikacji będzie składała się z dwóch tabel: „users” oraz „tasks”.

- tabela „users” zawiera informacje o użytkowniku (id, nickname, password, e-mail)
- tabela „tasks” zawiera informacje o zadaniach (id, name, description. Status, data dodania, termin wykonania)

## 7. Struktura projektu

Folder Controllers: Zawiera kontrolery aplikacji, które obsługują żądania HTTP i zarządzają logiką aplikacji.

Folder Models: Zawiera modele danych, które reprezentują struktury danych w aplikacji. Zawiera klasy odpowiadające tabelom w bazie danych, klasy DTO (Data Transfer Object) itp. oraz w naszym przypadku model „Task1Model.cs”

Folder Views: Zawiera pliki widoków, które definiują wygląd strony internetowej. Np. folder Views/Shared: Zawiera widoki wspólne. Przykładami są pliki layoutu (\_Layout.cshtml) oraz loginu (\_LoginPartial.cshtml).

## 8. Opis procesu rejestracji i logowania użytkowników

Aby zarejestrować się w aplikacji „Zadaniator”, użytkownik musi przejść do strony rejestracji i podać swoje dane, takie jak adres e-mail oraz podwójna weryfikacja hasła.

Użytkownik loguje się do aplikacji za pomocą swojego e-maila oraz hasła. Po podaniu danych do istniejącego konta, użytkownik zostaje automatycznie przekierowany do strony głównej.

### *9. Opis procesu zarządzania aplikacją „Zadaniator”*

Dodawania, edycja oraz usuwanie zadań jest możliwe na stronie dla zalogowanych użytkowników. Każde z zadań będzie można zaznaczyć jako “Wykonane”. Niezalogowani użytkownicy mają możliwość jedynie przeglądać stronę główną oraz zakładkę „Motywacja”

### *10. Możliwe usprawnienia w przyszłości*

- alerty,
  - sortowanie zadań,
  - filtrowanie zadań,
- oraz dokąd wyobraźnia poniesie : )

### *11. Podsumowanie*

Nasza aplikacja, jest prostym narzędziem do tworzenia listy zadań. Projekt został napisany w języku C# z użyciem frameworka ASP.NET i technologii webowych. Baza danych oparta jest na platformie MySQL.

