

Dokumentacja Techniczna - System Zarządzania Wydatkami "Twoje-Wydatki"

Data utworzenia: 2024-06-26

Autor: Adam Ostajewski

1. Opis systemu

"Twoje-Wydatki" to aplikacja webowa umożliwiająca użytkownikom zarządzanie swoimi wydatkami. Użytkownicy mogą rejestrować i logować się do systemu, a następnie dodawać, edytować oraz usuwać swoje wydatki. System oferuje również podsumowanie wydatków oraz wyświetlanie polityki prywatności.

2. Architektura systemu

- a. **Framework:** Entity Framework Core 8
- b. **Frontend:** HTML, CSS, Bootstrap
- c. **Backend:** .NET 8
- d. **Baza danych:** Microsoft SQL Server

3. Moduły systemu

a. Rejestracja i logowanie

- i. Rejestracja nowych użytkowników
- ii. Logowanie i wylogowywanie użytkowników
- iii. Zarządzanie kontem użytkownika (po kliknięciu w Profil - zmiana hasła, zmiana profilu)

b. Wydatki

- i. Dodawanie nowych wydatków
- ii. Edytowanie istniejących wydatków
- iii. Usuwanie wydatków
- iv. Wyświetlanie listy wszystkich wydatków użytkownika
- v. Podsumowanie wydatków (suma wydatków)

c. Profil

- i. Zmiana profilu użytkownika – imię, nazwisko, email
- ii. Zmiana hasła

4. Interfejs użytkownika

- a. Intuicyjny i przyjazny dla użytkownika interfejs
- b. Walidacja formularzy po stronie klienta i serwera

5. Bezpieczeństwo

- a. Zastosowanie HTTPS dla bezpiecznej komunikacji
- b. Walidacja formularzy

6. Wymagania systemowe

- a. .NET 8.0 SDK i Runtime
- b. Serwer SQL zgodny z Entity Framework Core
- c. Przeglądarka internetowa wspierająca HTML5 i JavaScript

7. Opis struktury projektu

- a. Controllers: Zawiera kontrolery obsługujące logikę aplikacji
- b. Models: Zawiera modele danych używane w aplikacji
- c. Views: Zawiera widoki (pliki .cshtml) renderowane przez kontrolery
- d. Database: Zawiera plik konfiguracyjny do obsługi bazy danych
- e. Interfaces: Zawiera interfejsy dla serwisów
- f. Services: Zawiera serwisy,
- g. Extensions: Zawiera rozszerzenia do klas,
- h. wwwroot: Zawiera statyczne pliki takie jak CSS, JavaScript

8. Modele

a. Expense

Pola:

- i. Id (int): unikalny identyfikator
- ii. Value (decimal): Wartość wydatku
- iii. Description (nvarchar(512)): Opis wydatku
- iv. UserId(int): Id użytkownika dodającego wydatek

b. User

Pola:

- i. Id (int): unikalny identyfikator
- ii. FirstName (nvarchar(128)): imię użytkownika
- iii. LastName (nvarchar(128)): nazwisko użytkownika
- iv. Email (nvarchar(max)): email użytkownika
- v. Password (nvarchar(max)): hasło użytkownika

c. Common – zawiera model, który został używany do dziedziczenia w logowaniu i rejestracji

d. ViewModels – zawiera modele reprezentujące np. rejestrację, logowanie, wyświetlanie wydatków czy też zmiany hasła użytkownika.

9. Kontrolery

a. HomeController

Metody:

Index

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla stronę główną.
- Zwracane dane: Widok Index.

Error

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla stronę błędu.
- Zwracane dane: Widok Error.

b. ProfileController

Metody:

Index

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla stronę główną profilu.
- Zwracane dane: Widok Index profilu.

ChangeProfile

- HTTP: GET
- Parametry: Brak
- Opis: Pobiera dane użytkownika i wypełnia formularz edycji profilu.
- Zwracane dane: Widok edycji profilu.

ChangePassword

- HTTP: GET
- Parametry: Brak

- Opis: Wyświetla formularz zmiany hasła
- Zwracane dane: Widok edycji hasła.

ChangeProfile

- HTTP: POST
- Parametry: ChangeProfileViewModel – imie, nazwisko, email
- Opis: Zmienia dane w profilu użytkownika
- Zwracane dane: Przekierowywane do profilu.

ChangePassword

- HTTP: POST
- Parametry: ChangePasswordViewModel – obecne hasło, nowe hasło, potwierdź nowe hasło
- Opis: Zmienia hasło
- Zwracane dane: Przekierowywane do profilu.

c. AuthController

Metody:

Login

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla formularz logowania użytkownika
- Zwracane dane: Widok formularza logowania użytkownika

Register

- HTTP: GET

- Parametry: Brak
- Opis: Wyświetla formularz rejestracji użytkownika
- Zwracane dane: Widok formularza rejestracji użytkownika

Logout

- HTTP: GET
- Parametry: Brak
- Opis: Wylogowywane z systemu
- Zwracane dane: Przekierowywane do strony głównej.

Login

- HTTP: POST
- Parametry: LoginViewModel– email, hasło
- Opis: Loguje do systemu
- Zwracane dane: Przekierowywane do strony głównej.

Register

- HTTP: POST
- Parametry: RegisterViewModel– Imię, nazwisko, email, hasło, potwierdź hasło
- Opis: Rejestracja nowego użytkownika
- Zwracane dane: Przekierowywane do strony głównej.

d. ExpenseController

Metody:

Index

- HTTP: GET
- Parametry: Brak
- Opis: Lista wraz z podsumowaniem wydatków pracownika
- Zwracane dane: Zwraca widok listy i podsumowania wydatków użytkownika.

Create

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla formularz tworzenia wydatku
- Zwracane dane: Zwraca widok tworzenia wydatku

Create

- HTTP: POST
- Parametry: Brak
- Opis: Tworzy nowy wpis wydatku do bazy danych
- Zwracane dane: Przekierowanie do listy wydatków

Remove

- HTTP: GET
- Parametry: ID wpisu wydatku
- Opis: Po naciśnięciu przycisku na liście wydatków – Usuń, wpis zostaje usunięty
- Zwracane dane: Zwraca widok listy i podsumowania wydatków użytkownika.

Edit

- HTTP: GET

- Parametry: ID wpisu wydatku
- Opis: Uzupełniony formularz wydatku, gotowy do edycji
- Zwracane dane: Zwraca widok listy i podsumowania wydatków użytkownika.

Edit

- HTTP: POST
- Parametry: ExpenseViewModel – ID, kwotę i opis
- Opis: Wysyła formularz uzupełniony do systemu i zmienia dane
- Zwracane dane: Zwraca widok listy i podsumowania wydatków użytkownika.

e. BaseApiController – służy do dziedziczenia, zawiera logikę niezbędną do pobierania obecnego użytkownika.

10. System użytkowników

a. Możliwości użytkowników zalogowanych

- Dodawanie, edytowanie i usuwanie własnych wydatków
- Przeglądanie sumy swoich wydatków

b. Możliwości użytkowników niezalogowanych

- Strona główna
- Logowanie
- Rejestracja

c. Informacje powiązane z użytkownikiem

- Użytkownik może mieć powiązane wydatki w bazie danych
- Dane użytkowników są przechowywane w bazie danych SQL

11. Charakterystyka najciekawszych funkcjonalności

- Walidacja danych wejściowych
- Responsive Design
- Bezpieczeństwo

12. Instalacja i konfiguracja

- Zainstaluj .NET 8.0 SCK i Runtime,
- Zainstaluj odpowiednie oprogramowanie do zarządzania bazą danych SQL, np. Microsoft SQL Server
- Skonfiguruj połączenie z bazą danych
- Wykonaj polecenie Update-Database, aby zaktualizować bazę danych,
- Uruchom aplikację za pomocą dotnet run

13. Wdrożenie aplikacji na serwerze

- a. Skonfiguruj serwer docelowy: zainstaluj niezbędne zależności, takie jak .NET Runtime,
- b. Skopiuj pliki aplikacji na serwer,
- c. Upewnij się że serwer ma odpowiednie połączenie z bazą danych SQL i jest skonfigurowany do pracy z aplikacją.

Po pomyślnej instalacji i konfiguracji, System Zarządzania Wydatkami będzie gotowy do użycia, oferując funkcjonalność zarządzania wydatkami użytkowników w sposób łatwy i bezpieczny.