Dokumentacja techniczna projektu "Menedżer zadań"

autorzy: Grzegorz Sosna i Bartosz Przebieracz

Spis treści

1. Wprowadzenie

1.1. Nazwa projektu

„Menedżer zadań”

1.2. Cel projektu

Celem projektu jest stworzenie aplikacji webowej typu menedżer zadań, która umożliwi użytkownikom efektywne zarządzanie swoimi obowiązkami. Projekt zakłada stworzenie intuicyjnego interfejsu użytkownika, który będzie prosty w obsłudze i dostępny dla każdego. System zapewni bezpieczne mechanizmy logowania i rejestracji, co pozwoli na ochronę danych użytkowników i personalizację ich doświadczeń. Kluczową funkcjonalnością systemu będzie możliwość dodawania, edytowania, usuwania zadań oraz grupowania, co pozwoli użytkownikom na skuteczne śledzenie swoich postępów.

2. Autorzy

Grzegorz Sosna

Bartosz Przebieracz

3. Technologie

3.1. Język programowania

- C#

3.2. Framework

- ASP.NET Core 8.0

3.3. Baza danych

- MS SQL Server

3.4. IDE

- Visual Studio

4. Instrukcje pierwszego uruchomienia

4.1. Wymagania systemowe

- Visual Studio 2022 (lub nowsze)

4.2. Instalacja

1. **Pobierz kod źródłowy projektu z GitHub**:
    - Otwórz przeglądarkę internetową i przejdź do strony repozytorium GitHub:

      https://github.com/GrzegorzSosna/Task-Manager-Project/tree/gs

    - Skopiuj adres URL repozytorium.
    - W terminalu lub wierszu poleceń wykonaj polecenie git clone <URL>, zastępując <URL> adresem repozytorium.
    - UWAGA: Jeżeli powyższa metoda nie zadziała, możesz również pobrać plik .zip z projektem z GitHub.
2. **Otwórz projekt** w Visual Studio:
    - Uruchom Visual Studio.
    - Wybierz opcję "Otwórz projekt" i przejdź do folderu, do którego sklonowałeś repozytorium.
3. **Skonfiguruj bazę danych** MS SQL Server zgodnie z plikiem konfiguracyjnym appsettings.json.
4. **Przeprowadź migracje bazy danych**:

- Otwórz **Package Manager Console** w Visual Studio.
- Wykonaj polecenie: Update-Database
- Sprawdź, czy migracje zakończyły się sukcesem, przeglądając logi w Visual Studio.

5. **Uruchom projekt** w Visual Studio.

5. Struktura projektu

5.1. Foldery i pliki

- **Controllers**: Zawiera kontrolery MVC.
  - HomeController.cs
  - TasksController.cs
  - StatisticsController.cs
- **Models**: Zawiera klasy modeli.
  - W pliku ErrorViewModel.cs znajduje klasa: ErrorViewModel
  - W pliku Task.cs znajduje się klasa: Task
  - W pliku Statistic.cs znajdują się klasy: Statistic, CategoryCount oraz MonthlyCategoryStatistic
- **Views**: Zawiera widoki.
  - **Home**: Widoki dla HomeController.
  - **Tasks**: Widoki dla TasksController.
  - **Statistics**: Widoki dla StatisticsController
- **wwwroot**: Zasoby statyczne (CSS, JavaScript, obrazy).

6. Modele

6.1. ErrorViewModel.cs

- **Opis**: Używany do przekazywania informacji o błędach do widoków.
- **Właściwości**:
  - RequestId (string) - Identyfikator żądania.

- ShowRequestId (bool) - Określa, czy identyfikator żądania powinien być wyświetlany.

6.2. Task.cs

- **Opis**: Reprezentuje zadanie w aplikacji.
- **Właściwości**:
  - Id (int) - Unikalny identyfikator zadania.
  - Descryption(string) - opis zadania (maks. długość: 100 znaków).
  - DateTime (DateTime) – Data i godzina zakończenia zadania (wymagane).
  - TaskCategory (string) - Grupa, do której zadanie należy (maks. długość: 30 znaków).
  - IsCompleted (bool) – Określa czy zadanie zostało już wykonane.
  - UserId (int) - Identyfikator użytkownika, który utworzył zadanie.

6.3. Statistic.cs

1. Klasa Statistic

- **Opis**: Reprezentuje statystyki dotyczące zadań w aplikacji.
- **Właściwości**:

  - CompletedTasksLastWeek (int): Liczba zadań ukończonych w ostatnim tygodniu.
  - UncompletedTasksLastWeek (int): Liczba zadań nieukończonych w ostatnim tygodniu.
  - CompletedTasksLastMonth (int): Liczba zadań ukończonych w ostatnim miesiącu.
  - UncompletedTasksLastMonth (int): Liczba zadań nieukończonych w ostatnim miesiącu.
  - CompletedTasksLastYear (int): Liczba zadań ukończonych w ostatnim roku.
  - UncompletedTasksLastYear (int): Liczba zadań nieukończonych w ostatnim roku.

- CategoryCounts (List<CategoryCount>): Lista zadań pogrupowanych według kategorii.
- EfficiencyLastWeek (double): Efektywność zadań w ostatnim tygodniu.
- EfficiencyLastMonth (double): Efektywność zadań w ostatnim miesiącu.
- EfficiencyLastYear (double): Efektywność zadań w ostatnim roku.

2. Klasa CategoryCount

- **Opis**: Reprezentuje licznik zadań według kategorii.
- **Właściwości**:
  - Category (string): Nazwa kategorii.
  - Count (int): Liczba zadań w danej kategorii.
  - CompletedCount (int): Liczba ukończonych zadań w danej kategorii.
  - Efficiency (double): Efektywność zadań w danej kategorii.

3. Klasa MonthlyCategoryStatistic

- **Opis**: Reprezentuje miesięczne statystyki zadań według kategorii.
- **Właściwości**:
  - Month (int): Miesiąc statystyk.
  - Year (int): Rok statystyk.
  - CompletedTasks (int): Liczba ukończonych zadań w danym miesiącu.
  - UncompletedTasks (int): Liczba nieukończonych zadań w danym miesiącu.
  - Efficiency (double): Efektywność zadań w danym miesiącu.

7. Kontrolery

7.1. HomeController.cs

- **Metody**:
  - Index() (GET): Wyświetla stronę główną.
  - Register() (GET): Wyświetla formularz rejestracji nowego użytkownika.
  - Register(User model) (POST): Przetwarza dane rejestracyjne nowego użytkownika.
  - Login() (GET): Wyświetla formularz logowania.
  - Login(User model) (POST): Przetwarza dane logowania użytkownika.
  - Logout() (POST): Wylogowuje użytkownika.

7.2. TasksController.cs

- **Metody**:
  - Index() (GET): Wyświetla listę zadań zalogowanego użytkownika.
  - Create() (GET): Wyświetla formularz tworzenia nowego zadania.
  - Create(Task model) (POST): Przetwarza dane nowego zadania.
  - Edit(int id) (GET): Wyświetla formularz edycji zadania.
  - Edit(int id, Task model) (POST): Przetwarza dane edytowanego zadania.
  - Delete(int id) (POST): Usuwa zadanie.

7.3. StatisticsController.cs

- **Metody**:
  - Index(): Wyświetla listę zadań zalogowanego użytkownika za ostatni tydzień, miesiąc i rok
- **Kluczowe punkty**:
  - Pobieranie danych: Kontroler pobiera zadania użytkownika z bazy danych na podstawie dat wykonania (ostatni tydzień, miesiąc, rok) oraz identyfikatora użytkownika.
  - Obliczenia: Oblicza liczbę ukończonych i nieukończonych zadań w różnych okresach oraz efektywność wykonania zadań.
  - Grupowanie: Grupuje zadania według kategorii i oblicza statystyki dla każdej kategorii.
  - Model statystyk: Tworzy model Statistic, który przechowuje obliczone wartości i przekazuje go do widoku.

8. System użytkowników

8.1. Role

- Aktualnie system nie implementuje różnych ról użytkowników. Każdy zarejestrowany użytkownik ma takie same uprawnienia.

8.2. Użytkownicy

- **Zalogowani użytkownicy**:
  - Mogą tworzyć, edytować i usuwać swoje zadania.
  - Mają dostęp do analizy swoich wykonanych zadań.
  - Każdy użytkownik widzi tylko swoje zadania.
- **Goście**:
  - Mogą przeglądać stronę główną z krótkim opisem aplikacji.
  - Mogą się zarejestrować lub zalogować.
- **Informacje związane z użytkownikami**:
  - Użytkownicy mają przypisane unikalne identyfikatory (UserId).
  - Zadania są powiązane z konkretnymi użytkownikami za pomocą UserId.

9. Najciekawsze funkcjonalności

9.1. Zarządzanie zadaniami

- **Opis**: Użytkownicy mogą tworzyć, edytować i usuwać zadania, przypisywać im daty zakończenia oraz przypisywać do grup.
- **Cel**: Ułatwia użytkownikom organizowanie i śledzenie swoich zadań.

9.2. Analiza wykonanych zadań

- **Opis**: Podstrona umożliwiająca użytkownikom przeglądanie statystyk dotyczących wykonanych zadań.
- **Cel**: Pomaga użytkownikom analizować swoją produktywność i zarządzanie czasem.

9.3. Bezpieczne logowanie

- **Opis**: Hasła są ukrywane podczas wpisywania, aby zapewnić prywatność użytkownika.
- **Cel**: Zwiększa bezpieczeństwo i prywatność użytkowników podczas logowania.