

Dokumentacja Techniczna

Tytuł projektu

System Zarządzania Zadaniem (Task Management System)

Opis projektu

Projekt jest aplikacją internetową umożliwiającą zarządzanie zadaniami w organizacji. System wspiera zarządzanie rolami użytkowników, dynamiczne przypisywanie zadań, generowanie raportów oraz automatyczne powiadomienia o terminach. Użytkownicy mogą przeglądać, tworzyć, edytować i usuwać zadania, a administratorzy mają dodatkowe funkcje zarządzania użytkownikami i monitorowania logów aktywności.

Autorzy

- Michał Dyczkowski
 - Kamil Grzybek
-

Specyfikacja wykorzystanych technologii

- **Backend:** C# .NET 8
 - **Frontend:** HTML, CSS (Bootstrap), JavaScript (jQuery)
 - **Baza danych:** SQLite (domyślna konfiguracja), z opcjonalną możliwością zmiany na MS SQL Server
 - **Wersja frameworka:** ASP.NET Core 8.0
 - **Inne biblioteki i narzędzia:**
 - Bootstrap (do stylowania interfejsu użytkownika)
 - jQuery Validation (do walidacji formularzy)
 - Entity Framework Core (do zarządzania bazą danych)
 - **Hosting lokalny:** Aplikacja działa na serwerze deweloperskim uruchamianym lokalnie.
-

Instrukcje pierwszego uruchomienia projektu

1. **Pobranie kodu:**
 - Skopiuj projekt z repozytorium lub pobierz odpowiedni plik ZIP.
2. **Wymagania systemowe:**
 - Zainstaluj .NET SDK w wersji 8.0 lub nowszej.
 - Zainstaluj SQLite lub MS SQL Server, jeśli chcesz korzystać z alternatywnej bazy danych.
3. **Konfiguracja bazy danych:**

- Domyślna konfiguracja SQLite znajduje się w pliku appsettings.json. Jeśli używasz MS SQL Server, zmodyfikuj połączenie w tym pliku.

4. Przygotowanie bazy danych:

- Uruchom następujące polecenia w terminalu projektu:

`dotnet ef database update`

To polecenie utworzy bazę danych i zaimportuje wymagane tabele.

5. Uruchomienie projektu:

- W Visual Studio kliknij zielony przycisk uruchamiania projektu (Start) znajdujący się na górnym pasku narzędzi.

6. Logowanie:

- Użyj domyślnego konta administratora:
 - Login: admin@example.com
 - Hasło: Admin123!
-

Struktura projektu

1. Controllers (Kontrolery):

- AccountController.cs: Obsługa logowania, rejestracji i zarządzania kontami użytkowników.
 - Metody:
 - **Login** (GET, POST): Obsługuje logowanie użytkownika. Przy poprawnych danych przekierowuje do strony głównej.
 - **Register** (GET, POST): Umożliwia rejestrację nowego użytkownika w systemie.
 - **Logout** (POST): Wylogowuje aktualnie zalogowanego użytkownika.
- AdminController.cs: Zarządzanie użytkownikami, generowanie raportów i przeglądanie logów aktywności.
 - Metody:
 - **Index** (GET): Wyświetla panel administracyjny.
 - **ManageUsers** (GET): Lista użytkowników w systemie z opcjami zarządzania (np. przypisywanie ról, usuwanie).
 - **AssignTask** (POST): Pozwala administratorowi przypisać zadanie do innego użytkownika, podając identyfikator zadania i nowego właściciela.
 - **ViewLogs** (GET): Wyświetla logi aktywności użytkowników.

- HomeController.cs: Obsługa strony głównej i widoków ogólnodostępnych.
 - Metody:
 - **Index** (GET): Wyświetla stronę główną aplikacji.
 - **Privacy** (GET): Informacje o polityce prywatności.
- TasksController.cs: Zarządzanie zadaniami, w tym tworzenie, edycja, usuwanie i przeglądanie szczegółów.
 - Metody:
 - **Index** (GET): Wyświetla listę wszystkich zadań przypisanych do użytkownika.
 - **Create** (GET, POST): Umożliwia tworzenie nowego zadania.
 - **Edit** (GET, POST): Umożliwia edycję istniejącego zadania.
 - **Delete** (POST): Usuwa zadanie na podstawie podanego identyfikatora.
 - **Details** (GET): Wyświetla szczegółowe informacje o zadaniu.

2. Models (Modele):

- ApplicationUser.cs: Model reprezentujący użytkownika aplikacji, zawierający informacje o rolach i danych osobowych.
- TaskItem.cs: Model zadania, zawierający pola takie jak tytuł, opis, termin i status.
- HistoryLog.cs: Model logu aktywności użytkowników, zapisujący zmiany w zadaniach i innych działaniach w systemie.
- ErrorViewModel.cs: Model do obsługi błędów aplikacji.

System użytkowników

1. Role w systemie:

- **Administrator**: Może zarządzać użytkownikami, przeglądać logi aktywności, generować raporty, zarządzać wszystkimi zadaniami oraz przypisywać zadania innym użytkownikom.
- **Użytkownik**: Może przeglądać i zarządzać swoimi zadaniami.

2. Przypisywanie ról:

- Role można przypisać za pomocą panelu administratora w sekcji zarządzania użytkownikami.

3. Różnice między zalogowanymi a gośćmi:

- Niezalogowani użytkownicy nie mają dostępu do żadnych informacji w systemie.

- Zalogowani użytkownicy mogą przeglądać zadania publiczne, których termin kończy się w przeciągu najbliższych 7 dni.
 - Każdy zalogowany użytkownik ma dostęp do swoich zadań, niezależnie od ich statusu i terminu realizacji.
-

Charakterystyka najciekawszych funkcjonalności

1. Automatyczne powiadomienia:

- System wyświetla automatyczne powiadomienia o zbliżających się terminach realizacji zadań. Najbliższe terminy są zaznaczane na czerwono wśród wszystkich zadań. System oferuje wyszukiwarkę zadań oraz funkcję ich sortowania.

2. Historia zmian:

- Każda rejestracja użytkownika, logowanie oraz usunięcie użytkownika i zadania jest rejestrowane w logach, co pozwala na monitorowanie kluczowych działań w systemie. System umożliwia przeglądanie tych logów wraz z czasem i autorem danej akcji.

3. Dynamiczne raporty:

- Administrator może generować raporty na podstawie wybranych kryteriów, takich jak status zadań, terminy czy aktywność użytkowników.
-