

Performance Tracker

- Opis projektu:

Performance Tracker to aplikacja, która wspomaga zarządzanie danymi dotyczącymi przepracowanych godzin i opisów działań pracowników. Aplikacja umożliwia przeglądanie najlepszych wyników indywidualnej aktywności. Autoryzowani pracownicy mogą wprowadzać swoje dane i działania, a także przeglądać ranking swoich prac. Aplikacja zawiera formularze rejestracji i logowania. Ponadto każdy użytkownik może przeczytać informacje o aplikacji oraz o osobach pracujących w firmie.

- Technologie wykorzystane w projekcie:

- Język programowania: C#
- Framework: .NET 8
- IDE: Visual Studio 2022
- Baza danych: SQL Server
- Entity Framework Core: ORM do zarządzania bazą danych
- Inne: ASP.NET Core MVC

- Instrukcja pierwszego uruchomienia projektu:

- Otwórz projekt w Visual Studio 2022.
- W konsoli menedżera pakietów uruchom polecenie:

```
Update-Database
```

- Uruchom projekt.

- Struktura projektu:

Controllers (kluczowe kontrolery, które obsługują żądania HTTP, realizując logikę biznesową):

- TasksController
- TaskTypesController
- HomeController
- WorkSessionsController
- WorkStatsController

Models (klasy definiujące strukturę danych i połączenia między nimi):

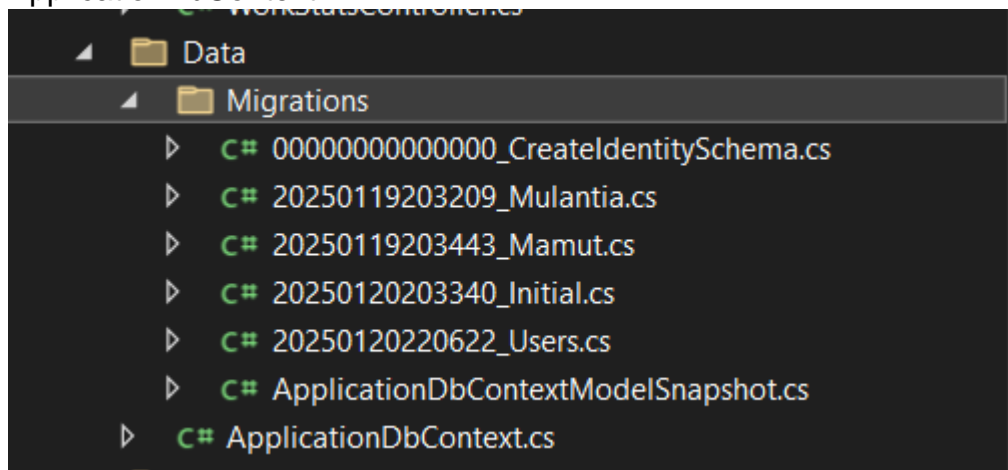
- Task
- TaskType
- WorkSession
- ErrorViewModel
- WorkStat

Views (strony, które prezentują dane użytkownikom):

- Home
- Shared
- TaskTypes
- Tasks
- WorkSessions
- WorkStats

Data (kontekst bazy danych oraz pliki migracji do zarządzania schematem bazy):

- ApplicationDbContext



- Szczegółowy opis modeli

Model Task:

```
public int Id { get; set; }

public int Hours { get; set; } [ Pole może mieć maksymalną wartość od 1 do 24 ]

public int Days { get; set; } [ Pole może mieć maksymalną wartość od 1 do 365 ]

public int Money { get; set; }

public int TaskTypeId { get; set; }

public virtual TaskType? TaskType { get; set; }
```

```
public int WorkSessionId { get; set; }  
public virtual WorkSession? WorkSession { get; set; }  
public string? UserId { get; set; }  
public virtual IdentityUser? User { get; set; }
```

Model TaskType:

```
public enum TaskTypeName  
{  
    ITSpecialist = 1,  
    Accountant = 2,  
    Cleaner = 3  
}  
public class TaskType  
{  
    public int Id { get; set; }  
    public TaskTypeName Name { get; set; }  
    public virtual ICollection<Task> Tasks { get; set; } = new List<Task>();  
  
}
```

Model WorkSession:

```
public int Id { get; set; }  
public DateTime LogIn { get; set; }  
public DateTime LogOut { get; set; }  
public virtual ICollection<Task> Tasks { get; set; } = new List<Task>();  
public string? UserId { get; set; }  
public IdentityUser? User { get; set; }
```

Model ErrorViewModel:

```
public string? RequestId { get; set; }
```

```
public bool ShowRequestId => !string.IsNullOrEmpty(RequestId);
```

Model WorkStat

```
public int Id { get; set; }  
public int TaskTypeId { get; set; }  
public virtual TaskType? TaskType { get; set; }  
public int NumberOfHoursWorked { get; set; }  
public int WorkMoney { get; set; }  
public string? UserId { get; set; }  
public virtual IdentityUser? User { get; set; }
```

- Kontrolery i ich funkcje

TasksController

- GET: Tasks
- GET: Tasks/Details/5
- GET: Tasks/Create
- GET: Tasks/Edit/5
- GET: Tasks/Delete/5
- POST: Tasks/Delete/5
- POST: Tasks/Edit/5
- POST: Tasks/Create

TaskTypesController

- GET: TaskTypes
- GET: TaskTypes/Details/5
- GET: TaskTypes/Create
- POST: TaskTypes/Create
- GET: TaskTypes/Edit/5
- POST: TaskTypes/Edit/5

- GET: TaskTypes/Delete/5
- POST: TaskTypes/Delete/5

WorkSessionsController

- GET: WorkSessions
- GET: WorkSessions/Details/5
- GET: WorkSessions/Create
- POST: WorkSessions/Create
- GET: WorkSessions/Edit/5
- POST: WorkSessions/Edit/5
- GET: WorkSessions/Delete/5
- POST: WorkSessions/Delete/5

WorkStatsController

- GET: WorkStats

HomeController

Nie ma funkcji

- System użytkowników

- Role użytkowników:

- User: Może wprowadzać dane pracy oraz dowolnie przeglądać.
 - Guest: Może jedynie przeglądać informacje o aplikacji oraz o stanowiskach pracujących w firmie.

- Autoryzacja:

- Użytkownicy rejestrują się podając e-mail i hasło.
 - Rejestracja nowych użytkowników wymaga wypełnienia formularza rejestracji.

- Podczas rejestracji generowany jest link weryfikacyjny wysyłany na podany adres e-mail.

- Pełen dostęp do funkcji wymaga potwierdzenia adresu e-mail.

- Najciekawsze funkcjonalności

- Ranking wyników (funkcja pozwalająca na przegląd przepracowanych godzin i wynikających z nich zarobków) (dla autoryzowanych).

- Rejestracja i logowanie (system bezpieczeństwa oparty na hasłach i autoryzacji, zapewniający ochronę danych).

- Osobiste informacje przypisane do konkretnego użytkownika.

- Podzielone jest na 3 strefy:

1 każdy może zobaczyć i przeczytać (Home)

2 każdy z użytkowników może korzystać (Types of work)

3 przypisane do pojedynczych użytkowników (Working session, Task in session, Work Statics)

- W " Types of work „ istnieć może jeden ten sam rodzaj zawodu

- Użytkownik jest w stanie ujrzeć nazwy kont innych użytkowników przy dodawaniu i edytowaniu danych w " Task in session „. Pomimo tego, gdy użytkownik wybierze jakiegokolwiek innego użytkownika to zawrze informacje zostaną przypisane do użytkownika który robi zmiany (na którym koncie ktoś jest obecnie zalogowanym)