

Dokumentacja Techniczna

Tytuł projektu

Wypożyczalnia Sprzętu Sportowego

Opis projektu

Wypożyczalnia Sprzętu Sportowego to aplikacja internetowa umożliwiająca zarządzanie procesem wypożyczania sprzętu sportowego. Użytkownicy mogą przeglądać dostępny sprzęt, rezerwować go na określony czas oraz zarządzać swoimi rezerwacjami.

Autor

- Kacper Frąckowiak

Specyfikacja Technologii

- Backend: ASP.NET 8
- Baza danych: PostgreSQL

Instrukcje Pierwszego Uruchomienia

- Sklonuj repozytorium projektu z GitHub:
- Wypakuj folder
- Uruchom plik SportEquipmentRent.sln
- Uruchom kompilowanie

Struktura Projektu

- Controllers: Zawiera kontrolery odpowiedzialne za obsługę żądań HTTP.
 - Models: Zawiera modele danych używane w aplikacji.
 - Services: Logika biznesowa aplikacji.
 - Views: Pliki HTML generowane na serwerze.
 - wwwroot: Statyczne pliki takie jak CSS, JavaScript i obrazy.
-

Modele

Model: **Renting**

Model **Renting** reprezentuje rezerwację sprzętu sportowego dokonaną przez użytkownika. Jest to kluczowa część aplikacji, ponieważ przechowuje wszystkie szczegóły dotyczące okresu wypożyczenia sprzętu.

- Id: Unikalny identyfikator rezerwacji.
- Start: Data rozpoczęcia rezerwacji.
- End: Data zakończenia rezerwacji.
- SportEquipmentId: Identyfikator sprzętu sportowego, który jest przedmiotem rezerwacji.
- SportEquipment: Związany obiekt sprzętu sportowego, który jest rezerwowany.
- UserId: Identyfikator użytkownika, który dokonał rezerwacji.
- User: Związany obiekt użytkownika, który jest odpowiedzialny za rezerwację.

Model: **SportEquipment**

Model **SportEquipment** reprezentuje sprzęt sportowy dostępny do wypożyczenia w systemie. Przechowuje informacje o dostępnych jednostkach sprzętu oraz umożliwia obliczenie liczby dostępnych sztuk.

- Id: Unikalny identyfikator sprzętu.
- Name: Nazwa sprzętu. Ma ograniczenie do 200 znaków, aby zachować spójność i zapewnić czytelność.
- Units: Całkowita liczba sztuk sprzętu dostępnych w systemie (od 1 do 1000).
- Rentings: Kolekcja obiektów **Renting**, które reprezentują wszystkie rezerwacje sprzętu. Obiekt ten jest używany do śledzenia, jakie jednostki sprzętu zostały wypożyczone i na jak długo.
- AvailableUnits(): Funkcja, która oblicza liczbę dostępnych sztuk sprzętu na podstawie liczby aktywnych rezerwacji. Zwraca liczbę dostępnych jednostek, biorąc pod uwagę tylko te rezerwacje, które są aktualnie aktywne (gdzie data zakończenia rezerwacji jest większa lub równa bieżącej dacie).

Kontrolery

Kontroler: **HomeController**

Metoda: **Index**

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

Metoda: **Privacy**

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla stronę z polityką prywatności.
- Zwracane dane: Widok HTML.

Metoda: **Error**

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla stronę błędu z informacjami diagnostycznymi.
- Zwracane dane: Widok HTML zawierający informacje o błędzie.

Kontroler: **RentingController**

Metoda: **Index**

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla listę wszystkich rezerwacji dostępnych dla bieżącego użytkownika lub administratora.
- Zwracane dane: Widok HTML z listą rezerwacji.

Metoda: **Details**

- HTTP: GET
- Parametry:
- id (int): Identyfikator rezerwacji.
- Opis: Wyświetla szczegóły konkretnej rezerwacji.
- Zwracane dane: Widok HTML z detalami rezerwacji.

Metoda: **Create**

- HTTP: GET, POST
- Parametry:
- GET: eqId (int?, opcjonalne): Identyfikator sprzętu.
- POST: Dane rezerwacji do utworzenia.
- Opis: Wyświetla formularz tworzenia nowej rezerwacji (GET) lub tworzy nową rezerwację (POST).
- Zwracane dane: Widok formularza (GET) lub przekierowanie do listy rezerwacji (POST).

Metoda: **Edit**

- HTTP: GET, POST
- Parametry:
- GET: id (int): Identyfikator rezerwacji.
- POST: Dane rezerwacji do edycji.
- Opis: Wyświetla formularz edycji rezerwacji (GET) lub aktualizuje istniejącą rezerwację (POST).
- Zwracane dane: Widok formularza (GET) lub przekierowanie do listy rezerwacji (POST).

Metoda: **Delete**

- HTTP: GET, POST
- Parametry:
- GET: id (int): Identyfikator rezerwacji.
- POST: Potwierdzenie usunięcia.
- Opis: Wyświetla stronę potwierdzenia usunięcia (GET) lub usuwa rezerwację (POST).
- Zwracane dane: Widok HTML (GET) lub przekierowanie do listy rezerwacji (POST).

Kontroler: **SportEquipmentController**

Metoda: **Index**

- HTTP: GET
- Parametry: Brak
- Opis: Wyświetla listę całego sprzętu sportowego.
- Zwracane dane: Widok HTML z listą sprzętu.

Metoda: **Details**

- HTTP: GET
- Parametry:
- id (int): Identyfikator sprzętu.
- Opis: Wyświetla szczegóły sprzętu sportowego.
- Zwracane dane: Widok HTML z detalami sprzętu.

Metoda: **Create**

- HTTP: GET, POST
- Parametry:
- POST: Dane sprzętu do utworzenia.
- Opis: Wyświetla formularz dodawania nowego sprzętu (GET) lub dodaje nowy sprzęt (POST).
- Zwracane dane: Widok HTML (GET) lub przekierowanie do listy sprzętu (POST).

Metoda: **Edit**

- HTTP: GET, POST
- Parametry:
- GET: id (int): Identyfikator sprzętu.
- POST: Dane sprzętu do edycji.

- Opis: Wyświetla formularz edycji sprzętu (GET) lub aktualizuje istniejący sprzęt (POST).
- Zwracane dane: Widok HTML (GET) lub przekierowanie do listy sprzętu (POST).

Metoda: **Delete**

- HTTP: GET, POST
- Parametry:
- GET: id (int): Identyfikator sprzętu.
- POST: Potwierdzenie usunięcia.
- Opis: Wyświetla stronę potwierdzenia usunięcia sprzętu (GET) lub usuwa sprzęt (POST).
- Zwracane dane: Widok HTML (GET) lub przekierowanie do listy sprzętu (POST).

System użytkowników

Role:

- Administrator: Pełny dostęp do systemu.
- Użytkownik: Możliwość rezerwacji sprzętu.

Mechanizm nadawania ról:

- Role przypisywane podczas rejestracji użytkownika w systemie.

Różnice w uprawnieniach:

- Zalogowani użytkownicy mogą dokonywać rezerwacji sprzętu.
- Goście mogą przeglądać listę dostępnych sprzętów.

Najciekawsze Funkcjonalności

1. Dynamiczna dostępność sprzętu: Sprzęt jest automatycznie oznaczany jako niedostępny w czasie trwania aktywnej rezerwacji.
 2. Historia rezerwacji: Każdy użytkownik ma dostęp do pełnej historii swoich rezerwacji.
 3. Połączenie z zewnętrzną bazą danych.
-