**Paweł Frankowski U-19537**

**Programowanie w .NET.**

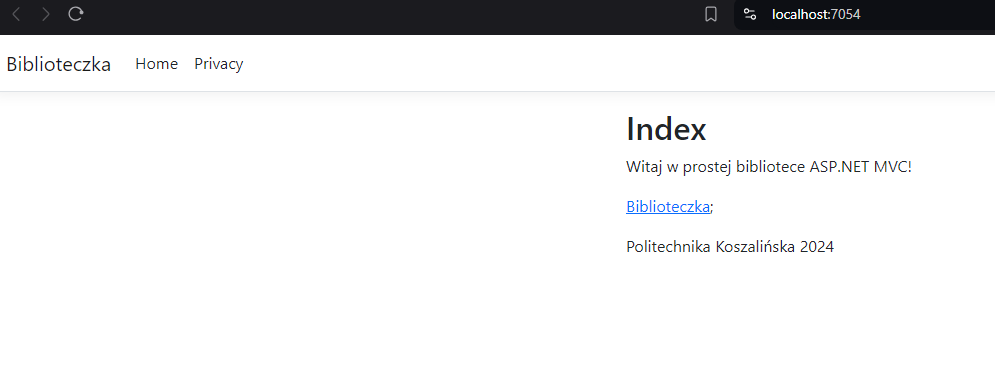
**Sprawozdanie z ćwiczenia praktycznego №4**

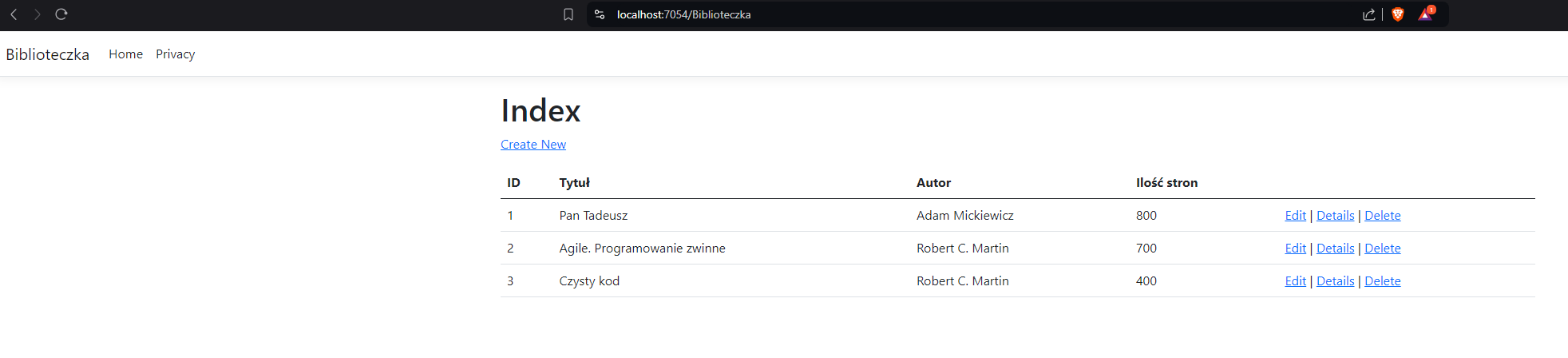
**Cel pracy:** Utworzenie prostej aplikacji opartej o wzorzec architektoniczny MVC. Zapoznanie się z ogólnymi zasadami MVC, przyjętymi konwencjami, a następnie stworzenie prostej aplikacji bazodanowej.

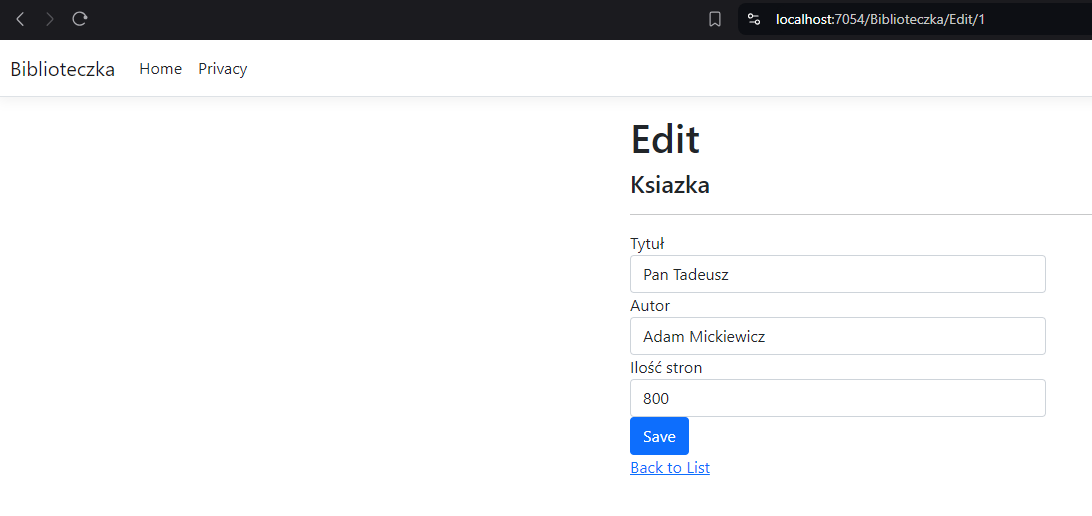
1. **Zapoznać się z następnymi podstawami:**

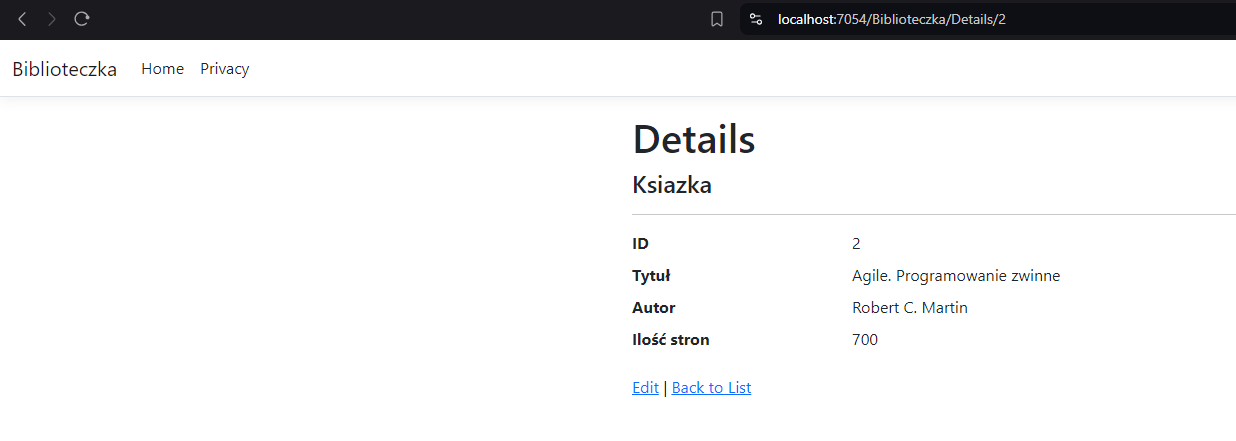
* Ogólne założenia MVC
  + MVC (Model-View-Controller) to wzorzec architektoniczny, który dzieli aplikację na trzy główne komponenty: Model (logika danych i reguły), Widok (interfejs użytkownika) oraz Kontroler (logika aplikacji i interakcje użytkownika). Taki podział ułatwia zarządzanie kodem, jego testowanie oraz rozwój aplikacji.
* Różnice między Web Forms a MVC
  + Web Forms to model oparty na zdarzeniach, który ukrywa szczegóły architektury, co prowadzi do silnego sprzężenia między kodem a interfejsem. MVC, z drugiej strony, promuje separację zadań, umożliwiając lepszą kontrolę nad strukturą aplikacji oraz łatwiejsze testowanie jednostkowe.
* Sposoby budowania modelu danych (Code First, Data First)
  + Code First: Deweloperzy tworzą modele danych w kodzie, a następnie generują bazę danych na ich podstawie. Umożliwia to łatwe wprowadzanie zmian w modelach i ich szybką aktualizację.
  + Data First: Proces zaczyna się od istniejącej bazy danych, z której generowane są klasy modeli. Jest to podejście stosowane w projektach, gdzie baza danych jest już zdefiniowana.
* Inne wzorce architektoniczne wykorzystywane do tworzenia aplikacji webowych oraz okienkowych
  + Oprócz MVC, inne popularne wzorce to MVVM (Model-View-ViewModel), MVP (Model-View-Presenter) oraz architektura mikroserwisów. Każdy z tych wzorców ma swoje zastosowania w budowie aplikacji webowych i desktopowych, pomagając w zarządzaniu złożonością kodu.
* Podstawy HTML
  + HTML (HyperText Markup Language) to język znaczników używany do tworzenia stron internetowych. Służy do strukturyzacji treści oraz definiowania elementów, takich jak nagłówki, akapity, obrazy i formularze. Znajomość HTML jest kluczowa dla web developerów, ponieważ stanowi fundament dla wszystkich technologii front-endowych.

2. **Aplikacja**

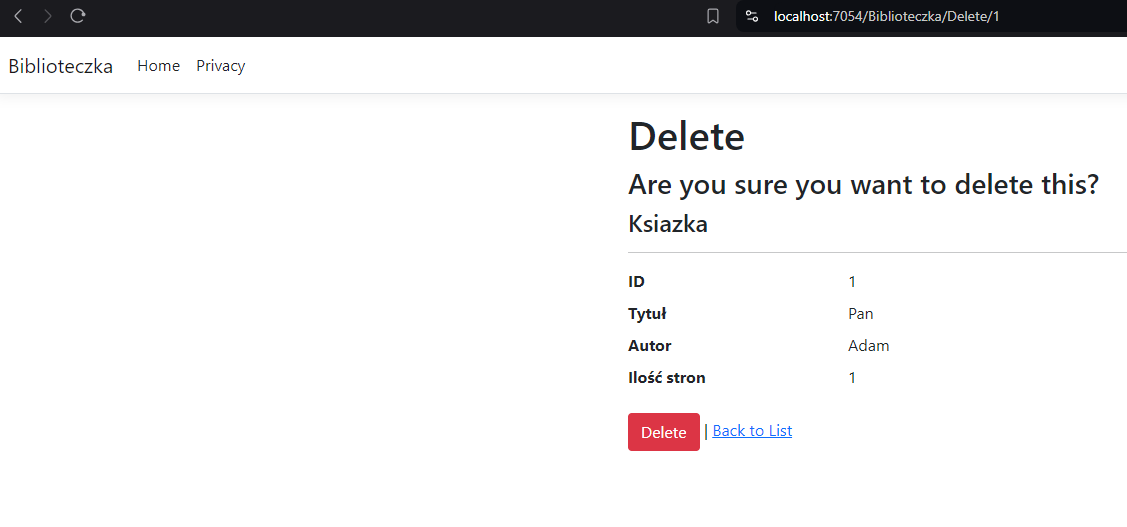
Rys. 1. Widok główny aplikacji

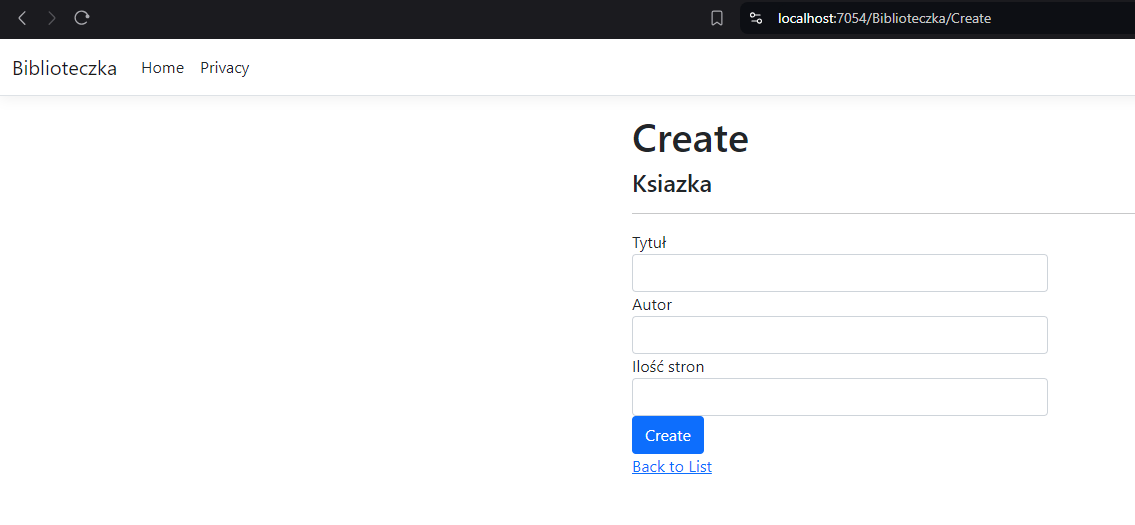
Rys. 2. Widok Biblioteczki pod adresem /Biblioteczka

Rys. 3. Widok /Edit



Rys. 4. Widok /Details

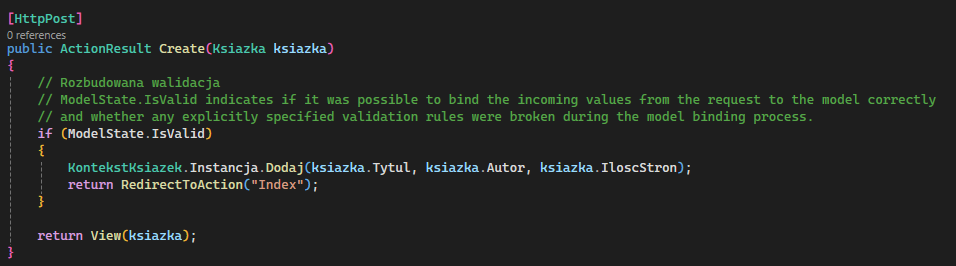
Rys. 5. Widok /Delete z potwierdzeniem

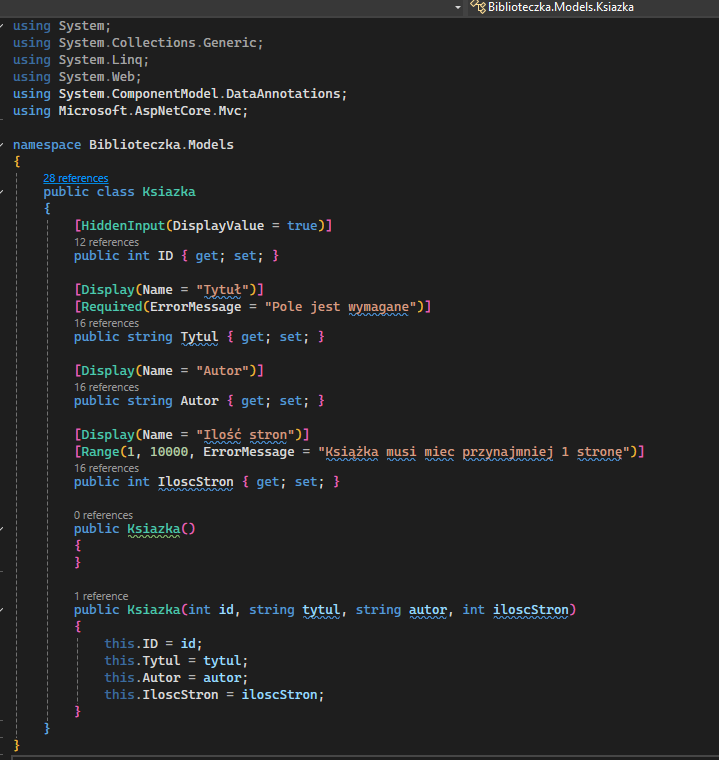


Rys. 6. Widok /Create

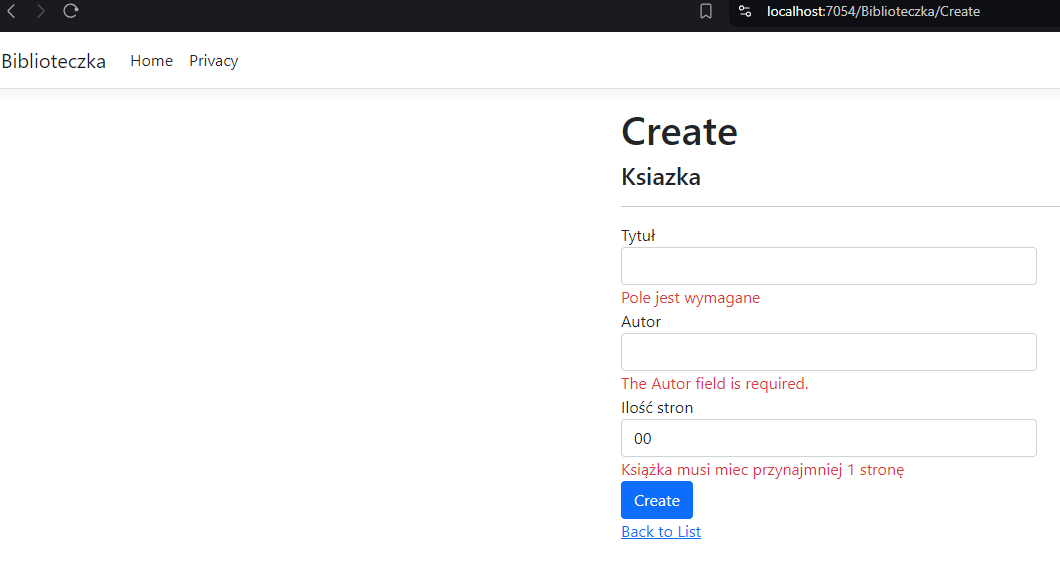
3.  **Walidacja danych**

Aplikacja została rozbudowana o dodatkową walidację przy użyciu ModelState.

Rys. 7. Dodatkowa walidacja zaimplementowana używając ModelState.IsValid



Rys. 8. Implementacja logiki walidacji dla modelu Ksiazka

Rys. 9. Walidacja dla tworzenia nowej książki pod adresem /Create

4. **Podsumowanie**

W ramach ćwiczenia praktycznego nr 4 stworzyłem prostą aplikację bazodanową opartą na wzorcu architektonicznym MVC. Celem pracy było zapoznanie się z podstawami MVC, w tym z ogólnymi zasadami, różnicami między Web Forms a MVC, sposobami budowania modeli danych (Code First i Data First), innymi wzorcami architektonicznymi oraz podstawami HTML. Aplikacja została rozbudowana o walidację danych przy użyciu ModelState, co umożliwiło lepsze zarządzanie błędami podczas wprowadzania danych przez użytkowników. Zrealizowane widoki aplikacji obejmują m.in. edycję, szczegóły, usuwanie oraz dodawania książek.

**Link do repozytorium:** https://github.com/pablolambo/LibraryAppMVC