**BookStore – Dokumentacja**

**Spis treści**

1. **Opis projektu**
2. **Wymagania i instalacja**  
   2.1. Pobranie kodu z GitHuba  
   2.2. Uruchomienie aplikacji
3. **Konfiguracja**  
   3.1. Łańcuch połączenia z bazą  
   3.2. Użytkownicy testowi
4. **Opis działania aplikacji**  
   4.1. Strona główna – lista książek  
   4.2. Logowanie i rejestracja  
   4.3. Koszyk i zamówienie  
   4.4. Panel administratora (dodawanie/edycja książek)
5. **Formularze i walidacja**
6. **Technologie użyte w projekcie**

**1. Opis projektu**

**BookStore** to aplikacja webowa napisana w **ASP.NET Core MVC** z wykorzystaniem **LiteDB** jako plikowej bazy danych. Umożliwia:

* Przeglądanie listy książek,
* Rejestrację i logowanie użytkowników,
* Dodawanie książek do koszyka i składanie zamówień,
* Zarządzanie książkami (tylko dla administratora).

**2. Wymagania i instalacja**

**2.1. Pobranie kodu z GitHuba**

1. Otwórz repozytorium **BookStore** w serwisie GitHub.
2. Wybierz **Code** → **Download ZIP** (lub sklonuj repozytorium przez git clone).
3. Rozpakuj pobrane archiwum w wybranym miejscu na dysku.

**2.2. Uruchomienie aplikacji**

1. Otwórz **Visual Studio 2022** (lub nowsze), wybierz **Open a project or solution** i wskaż plik .sln lub .csproj.
2. Poczekaj, aż środowisko pobierze wszystkie pakiety NuGet.
3. Ustaw **BookStore** (projekt główny) jako StartUp Project, jeśli korzystasz z Visual Studio.
4. Kliknij **Start** lub naciśnij **F5**, aby uruchomić aplikację lokalnie (np. https://localhost:5001).

**3. Konfiguracja**

**3.1. Łańcuch połączenia z bazą**

* Aplikacja korzysta z **LiteDB** – plik BookStore.db tworzony jest automatycznie w folderze roboczym.
* Nie ma potrzeby konfigurowania zewnętrznego serwera bazy danych.

**3.2. Użytkownicy testowi**

* Domyślnie w klasie LiteDBContext tworzy się **administrator**:
  + **Login**: admin1
  + **Hasło**: admin1
* Zwykłych użytkowników można dodawać przez formularz rejestracji.

**4. Opis działania aplikacji**

**4.1. Strona główna – lista książek**

* Po uruchomieniu aplikacji domyślnie wyświetla się Book/Index.
* Użytkownik może przeglądać książki, a zalogowany **nie-admin** ma przycisk **Buy** (dodanie do koszyka).
* Admin widzi przyciski **Edit** i **Delete** przy każdej książce.

**4.2. Logowanie i rejestracja**

* **Register**: tworzy nowego użytkownika (pola login i hasło).
* **Login**: pozwala się zalogować – jeśli dane są poprawne, wyświetla się w nawigacji „Hello, [login]!”.

**4.3. Koszyk i zamówienie**

* Kliknięcie **Buy** (dla zalogowanego, nie-admina) dodaje książkę do koszyka.
* Przycisk **Your cart** w nawigacji wyświetla modal z zawartością koszyka.
* **Go to shipping** przenosi na stronę z formularzem zamówienia (dane do wysyłki).
* Po wypełnieniu formularza (przy poprawnych danych) pokazuje się komunikat **„Order has been created.”**.

**4.4. Panel administratora (dodawanie/edycja książek)**

* Administrator (admin1/admin1) może edytować i usuwać książki oraz **Add New Book**.

**5. Formularze i walidacja**

W projekcie zaimplementowano co najmniej cztery formularze, każdy z walidacją:

1. **Formularz rejestracji** (Register) – sprawdza, czy nazwa użytkownika istnieje, wymaga podania hasła.
2. **Formularz logowania** (Login) – wymaga prawidłowego hasła i loginu.
3. **Formularz edycji/dodawania książek** (Create/Edit) – tytuł, autor, cena.
4. **Formularz zamówienia** (Shipping) – zawiera [Required] przy głównych polach (imię, nazwisko, telefon, adres).

**6. Technologie użyte w projekcie**

* **ASP.NET Core MVC (.NET 6/7)**
* **LiteDB** (plikowa baza danych)
* **Bootstrap 5** (stylowanie, modal)
* **jQuery, jquery.validate** (AJAX, walidacja)
* **C#** jako główny język backendu

**Podsumowanie:**  
Projekt można uruchomić z poziomu kodu źródłowego po pobraniu z GitHuba – wystarczy otworzyć w Visual Studio, pobrać zależności i uruchomić. Domyślny admin: admin1 (hasło: admin1). Zwykłych użytkowników rejestruje się w aplikacji.