

# Dokumentacja projektu ASP.NET MVC

**Temat:** Sklep internetowy

**Przedmiot:** BDwAI

**Studenci:** Dawid Tarnawa (s168680), Filip Woźniak (s168757)

## 1. Cel i zakres projektu:

Celem projektu było zaprojektowanie i implementacja kompletnej aplikacji sklepu internetowego w technologii ASP.NET Core MVC. System umożliwia obsługę procesu sprzedaży od strony klienta (przeglądanie oferty, koszyk, zamówienie, kontakt) oraz zarządzanie treścią i sprzedażą od strony administratora.

Aplikacja realizuje kluczowe założenia nowoczesnych systemów webowych:

- Architektura MVC (Model-View-Controller).
- Komunikacja z bazą danych poprzez ORM (Entity Framework Core).
- Uwierzytelnianie i autoryzacja użytkowników (Identity).
- Obsługa sesji (Koszyk zakupowy).

## 2. Wymagania systemowe Do poprawnego uruchomienia aplikacji w środowisku deweloperskim wymagane są:

System operacyjny: Windows 10 lub nowszy.

Środowisko: Microsoft Visual Studio 2022 (z workloadem ASP.NET and web development).

Platforma: .NET 8.0 (lub nowszy, kompatybilny z wersją projektu).

Baza danych: Microsoft SQL Server LocalDB (składnik instalowany domyślnie z Visual Studio).

## 3. Konfiguracja i Instalacja

3.1. Uruchomienie rozwiązania Należy otworzyć plik rozwiązania

SklepInternetowy.sln w środowisku Visual Studio.

Po otwarciu należy poczekać na automatyczne pobranie brakujących pakietów NuGet (Restore Packages).

3.2. Konfiguracja połączenia z bazą danych Aplikacja skonfigurowana jest do pracy z lokalną instancją bazy danych (LocalDB). Łańcuch połączenia (Connection String) znajduje się w pliku appsettings.json:

```
"ConnectionStrings": {  
  "DefaultConnection": "Server(localdb)\\mssqllocaldb;Database=SklepInternetowy;Trusted_Connection=True;MultipleActiveResultSets=true"  
}
```

3.3. Inicjalizacja bazy danych (Migracje) Projekt wykorzystuje podejście Code First. Przed pierwszym uruchomieniem konieczne jest utworzenie struktury bazy

danych na podstawie modeli.

#### **Instrukcja wykonania migracji:**

1. Package Manager Console wpisać polecenie: Update-Database
2. Zatwierdzić klawiszem Enter. Operacja utworzy lokalną bazę danych oraz tabele: AspNetUsers, Products, Categories, Orders, OrderItems, ContactMessages.

#### **4. Konta testowe i Bezpieczeństwo**

W systemie zaimplementowano podział na role: Administrator oraz Użytkownik. Poniżej znajdują się dane logowania do kont przygotowanych na potrzeby weryfikacji projektu.

- 4.1. Konto Administratora Posiada pełny dostęp do ukrytego panelu zarządzania.

**Login (Email):** [admin@sklep.pl](mailto:admin@sklep.pl)

**Hasło:** Hasło123!

- 4.2. Konto Użytkownika (Klient) Konto służące do testowania procesu składania zamówienia.

**Login (Email):** [klient@test.pl](mailto:klient@test.pl)

**Hasło:** Hasło123!

(Uwaga: System umożliwia również samodzielną rejestrację nowych użytkowników poprzez formularz dostępny pod przyciskiem "Rejestracja").

#### **5. Opis funkcjonalności aplikacji**

##### **5.1. Widok Klienta (Frontend)**

Interfejs użytkownika został zaprojektowany w nowoczesnym stylu z wykorzystaniem biblioteki Bootstrap. Nawigacja opiera się na intuicyjnych przyciskach typu "pill" (zaokrąglone).

##### **1. Strona Główna (Home):**

- Wyświetla listę wszystkich dostępnych produktów.
- Każdy produkt prezentowany jest na karcie ("Card") zawierającej: nazwę, kategorię, opis, cenę oraz przycisk "Dodaj do koszyka".
- Nazwy kategorii są klikalne i wyróżnione stylem (badge).

##### **2. Koszyk Zakupowy (Sesja):**

- Dostępny pod przyciskiem "Koszyk".
- Działa w oparciu o Session State – użytkownik może dodawać produkty jako gość (niezalogowany).
- Umożliwia podgląd wybranych produktów, zmianę ich ilości, podsumowanie ceny oraz usuwanie pozycji.
- Finalizacja zakupu ("Przejdź do kasy") wymaga zalogowania.

##### **3. Proces Zamówienia:**

- Po zalogowaniu użytkownik wypełnia formularz adresu dostawy.
- Zamówienie jest zapisywane w bazie danych, a koszyk jest czyszczony.

##### **4. Moduł Kontakt:**

- Dostępny pod przyciskiem Kontakt.
- Formularz umożliwia wystanie wiadomości do obsługi sklepu (Email, Telefon, Temat, Treść).
- Wiadomości są zapisywane w bazie danych.

## **5.2. Panel Administratora (Backend)**

Po zalogowaniu na konto z rolą Admin, w menu głównym pojawia się wyróżniony (czerwony) przycisk Panel Admina. Udostępnia on następujące funkcje:

### **1. Zarządzaj Kategoriami:**

- Pełny CRUD (Dodawanie, Edycja, Usuwanie) kategorii produktów.

### **2. Zarządzaj Produktami:**

- Dodawanie nowych produktów ze wskazaniem kategorii i ceny.
- Edycja opisów i cen istniejących produktów.

### **3. Zarządzaj Zamówieniami:**

- Lista wszystkich złożonych zamówień w sklepie.
- Podgląd szczegółów: data zamówienia, adres wysyłki, łączna kwota.

### **4. Wiadomości od klientów:**

- Skrzynka odbiorcza formularza kontaktowego.
- Możliwość odczytu i usuwania wiadomości przesłanych przez użytkowników.

## **6. Struktura techniczna Aplikacja wykorzystuje następujące rozwiązania:**

- Entity Framework Core (Code First): Do mapowania obiektowo-relacyjnego.
- ASP.NET Core Identity: Do obsługi autoryzacji i uwierzytelniania.
- LINQ: Do filtrowania i pobierania danych.
- Session: Do przechowywania stanu koszyka tymczasowego.
- Dependency Injection: Wstrzykiwanie kontekstu bazy danych do kontrolerów.