

Mikołaj Karwacki

Jakub Kuszczalski

# Dokumentacja Techniczna Sklepu Internetowego Luis De Lagasin

## 1. Wprowadzenie

### Cel dokumentacji

Dokumentacja techniczna ma na celu opisanie struktury oraz funkcjonalności sklepu internetowego "Luis De Lagasin". W sklepie dostępne są różnorodne produkty odzieżowe, a platforma umożliwia użytkownikom przeglądanie, logowanie, rejestrację, zakupy, oraz zarządzanie przez panel administracyjny.

### Architektura systemu

System oparty jest na dwóch warstwach: frontend, napisany w HTML, CSS, JavaScript, oraz backend, napisany w języku C# z użyciem Entity Framework i API, połączony za pomocą JavaScript fetchAPI.

## 2. Funkcjonalności Sklepu

### Logowanie

Użytkownicy mogą zalogować się do sklepu za pomocą swojego konta, podając e-mail oraz hasło.

### Rejestracja

Nowi użytkownicy mogą utworzyć konto, dostarczając niezbędne informacje, takie jak imię, nazwisko, adres e-mail, oraz hasło.

### Kategorie

Produkty są podzielone na kategorie, umożliwiając użytkownikom łatwiejsze przeglądanie. Kategorie obejmują bluzy, koszulki.

### Przeglądanie produktów

Użytkownicy mogą przeglądać produkty w ramach wybranej kategorii, oraz wyświetlać szczegółowe informacje o każdym produkcie.

### **Panel Administracyjny (CRUD)**

- Admin
  - Administrator ma dostęp do panelu administracyjnego, umożliwiającego zarządzanie produktami, użytkownikami.
- Klient
  - Klient ma dostęp do panelu, gdzie może zarządzać swoim kontem, historią zakupów oraz aktualnym koszykiem.

### **Dodawanie do koszyka**

Użytkownicy mogą dodawać produkty do koszyka podczas przeglądania sklepu.

### **Zarządzanie koszykiem**

Klienci mają możliwość dodawania, usuwania oraz modyfikowania ilości produktów w swoim koszyku przed dokonaniem zakupu.

## **3. Technologie**

### **Frontend**

HTML5

CSS3

JavaScript (ES6+)

Biblioteka do interakcji z API (FetchAPI)

### **Backend**

Język programowania: C#

Framework: ASP.NET Core

ORM: Entity Framework Core

API do komunikacji między frontend a backend

## **4. Bezpieczeństwo**

### **Bezpieczeństwo logowania**

Hasła są przechowywane w formie zaszyfrowanej (np. hashowanie) w bazie danych.

## **Bezpieczeństwo transmisji danych**

Wszystkie dane przesyłane między klientem a serwerem są szyfrowane za pomocą protokołów bezpieczeństwa, takich jak SSL/TLS.

## **5. Wdrożenie**

### **Baza Danych**

Za pomocą SQL SERVER MANAGMENT STUDIO importujesz bazę która znajduje się w folderze Data/bazaLagasin.bacpac. ( UWAGA jest to przykładowa baza która posiada admina, jeśli stworzyć swoją bazę za pomocą add-migration i update-database należy przełączyć w pliku lunchSettings.json zawartość lunchURL na swagger i utworzyć konto administratora.)

## **6. Instrukcja uruchomienia projektu**

1. Pobierz repozytorium w formacie ZIP i rozpakuj
2. Pobierz SQL SERVER oraz SQL SERVER MANAGMENT STUDIO.
3. Za pomocą SQL SERVER MANAGMENT STUDIO importujesz bazę która znajduje się w folderze Data/bazaLagasin.bacpac. ( UWAGA jest to przykładowa baza która posiada admina, jeśli stworzyć swoją bazę za pomocą add-migration i update-database należy przełączyć w pliku lunchSettings.json zawartość lunchURL na swagger i utworzyć konto administratora.)
4. Dostosuj konfigurację, taką jak połączenie z bazą danych w pliku Data/ApplicationDbContext.cs.  
Przykład: `connectionString = "Data Source=TU DAJESZ SWÓJ SERVER;Initial Catalog=bazaLagasin;Integrated Security=True";`
5. Należy zobaczyć czy pakiety nuget działają poprawnie.
6. Teraz możesz uruchomić projekt. Jeśli połączenie z bazą nie działa spróbuj dodać migracje i zaktualizować bazę :)
7. Sprawdź dokumentację API w kodzie lub korzystaj z Endpointów opisanych poniżej.