

DOKUMENTACJA PROJEKTU

***E-COMMERCE W TECHNOLOGII C# I REACT***

Z PRZEDMIOTU

PROJEKTOWANIE WIELOWARSTWOWYCH APLIKACJI BIZNESOWYCH

Wykonali:

GOŁACKA Anna, 51412

KOZIEŁ Adrian, 55242

NIŚCIOR Inga, 51418

SZYMAŃSKI Emil, 51424

WIERZBICKI Piotr, 55264

*WARSZAWA 2024*

Spis treści

[Wstęp 3](#_Toc163288453)

[Wykorzystane oprogramowanie 3](#_Toc163288454)

[Wymagania funkcjonalne 3](#_Toc163288455)

[Wymagania niefunkcjonalne 4](#_Toc163288456)

[Sposoby i metody testowania 4](#_Toc163288457)

[Testowanie ręczne 4](#_Toc163288458)

[Testowanie automatyczne 4](#_Toc163288459)

[Testowanie ciągłe 5](#_Toc163288460)

[Warstwa prezentacji 5](#_Toc163288461)

[Baza danych 6](#_Toc163288462)

[Diagram klas UML 7](#_Toc163288463)

[Dokumentacja techniczna kodu 7](#_Toc163288464)

# Wstęp

Przegląd projektu Platforma handlu elektronicznego to nowoczesne rozwiązanie do zakupów online zaprojektowane w celu zapewnienia klientom intuicyjnych i płynnych zakupów. Projekt ten ma na celu opracowanie bogatej w funkcje, skalowalnej i bezpiecznej witryny e-commerce, w której użytkownicy mogą przeglądać produkty, dodawać je do swoich koszyków, dokonywać płatności i śledzić zamówienia.

# Wykorzystane oprogramowanie

Visual studio

GitHub

Microsoft SQL Server

React.js

# Wymagania funkcjonalne

1. **Rejestracja i uwierzytelnianie użytkowników**

* Użytkownicy muszą mieć możliwość utworzenia konta
* Użytkownicy muszą mieć możliwość logowania i wylogowywania się ze swoich kont.
* Użytkownicy muszą mieć możliwość resetowania swoich haseł.

1. **Katalog produktów**

* Witryna powinna wyświetlać obszerną listę produktów ze szczegółowymi opisami, zdjęciami, cenami i dostępnością.
* Klienci powinni mieć możliwość wyszukiwania produktów przy użyciu różnych filtrów (np. kategorii, marki, zakresu cen).
* Administratorzy powinni mieć możliwość dodawania, edytowania i usuwania produktów z katalogu.

1. **Koszyk**

* Użytkownicy powinni mieć możliwość dodawania produktów do swoich koszyków.
* Użytkownicy powinni mieć możliwość przeglądania zawartości swoich koszyków i aktualizowania ich ilości.
* Użytkownicy powinni mieć możliwość usuwania produktów ze swoich koszyków.
* Witryna powinna automatycznie obliczać łączną cenę produktów w koszyku.

1. **Zarządzanie zamówieniami**

* Klienci powinni mieć możliwość składania zamówień i śledzenia ich statusu.
* Administratorzy powinni mieć możliwość przeglądania, przetwarzania i zarządzania zamówieniami klientów.
* Zamówienia powinny być wysyłane lub odbierane w fizycznej lokalizacji.

# Wymagania niefunkcjonalne

1. **Wydajność**: Platforma powinna być w stanie obsłużyć co najmniej 1000 użytkowników jednocześnie bez degradacji wydajności.
2. **Czas odpowiedzi**: Czas odpowiedzi na żądania użytkowników nie powinien przekraczać 2 sekund. To zapewnia płynną i satysfakcjonującą interakcję użytkownika z platformą.
3. **Dostępność**: Platforma powinna być dostępna w 99.9xx%. To jest kluczowe dla e-commerce, ponieważ klienci mogą chcieć dokonywać zakupów o dowolnej porze dnia czy nocy.
4. **Bezpieczeństwo**: Platforma powinna zapewniać bezpieczne i niezawodne przetwarzanie płatności. Powinna również chronić dane osobowe użytkowników zgodnie z obowiązującymi przepisami o ochronie danych.
5. **Skalowalność**: Platforma powinna być łatwo skalowalna, aby móc obsługiwać rosnącą liczbę użytkowników i transakcji.
6. **Użyteczność**: Interfejs użytkownika powinien być intuicyjny i łatwy w użyciu, zarówno dla klientów, jak i sprzedawców. (baza na danej normie wiekowej, wziąć pod uwagę osoby niedowidzące itp)
7. **Kompatybilność**: Platforma powinna być kompatybilna z różnymi przeglądarkami i urządzeniami, aby umożliwić jak największej liczbie użytkowników korzystanie z niej.
8. **Responsywność**: Interfejs użytkownika powinien być responsywny, co oznacza, że powinien dobrze wyglądać i działać na różnych rozdzielczościach ekranu, od smartfonów po komputery stacjonarne.

# Sposoby i metody testowania

Nasza strategia testowania obejmuje zarówno ręczne, jak i automatyczne metody testowania:

### Testowanie ręczne

**Testowanie funkcjonalne**: Obejmuje testowanie konkretnych przypadków użycia i historii użytkownika w celu zapewnienia funkcjonalności aplikacji.

**Testy użyteczności**: Przeprowadzimy testy użyteczności z prawdziwymi użytkownikami, aby zidentyfikować wszelkie problemy w zakresie dostępności, łatwości użytkowania i ogólnego doświadczenia użytkownika.

### Testowanie automatyczne

**Testy jednostkowe**: Obejmuje testowanie poszczególnych komponentów lub funkcji bazy kodu w celu zapewnienia ich prawidłowego działania.

**Testy integracyjne**: Testujemy interakcje między różnymi komponentami lub systemami w naszej aplikacji.

**Testy regresji**: Będziemy okresowo ponownie uruchamiać testy automatyczne, aby zidentyfikować i naprawić wszelkie nowe problemy, które mogły zostać wprowadzone podczas rozwoju.

### Testowanie ciągłe

Będziemy przeprowadzać regularne przeglądy kodu, upewniając się, że wszystkie zmiany są dokładnie testowane przed scaleniem z główną gałęzią.

Nasz serwer ciągłej integracji będzie uruchamiał testy automatyczne za każdym razem, gdy nowe zmiany zostaną przesłane do repozytorium.

Stosując się do tej kompleksowej strategii testowania, możemy zapewnić, że nasza platforma e-commerce dostarcza wysokiej jakości doświadczenia zakupowe dla naszych użytkowników, przy jednoczesnym Warstwa logiki biznesowej

Warstwa logiki biznesowej dla platformy e-commerce obejmuje następujące elementy:

**Zarządzanie produktami**: Powinny być metody umożliwiające dodawanie nowych produktów, edycję istniejących produktów i usuwanie produktów. Każdy produkt może mieć różne atrybuty, takie jak nazwa, opis, cena, zdjęcie i sprzedawca.

**Zarządzanie zamówieniami**: Powinny być metody umożliwiające tworzenie nowych zamówień, aktualizację statusu istniejących zamówień i anulowanie zamówień. Każde zamówienie może mieć różne atrybuty, takie jak lista produktów, łączna kwota, adres dostawy i status.

**Zarządzanie użytkownikami**: Powinny być metody umożliwiające rejestrację nowych użytkowników, logowanie i wylogowywanie, a także aktualizację profilu użytkownika. Każdy użytkownik może mieć różne atrybuty, takie jak nazwa użytkownika, hasło, adres e-mail i adres dostawy.

**Obsługa płatności**: Powinny być metody umożliwiające przetwarzanie płatności za zamówienia. To może obejmować integrację z zewnętrznym dostawcą usług płatniczych.

**Wysyłka**: Powinny być metody umożliwiające obsługę wysyłki zamówień. To może obejmować integrację z zewnętrznym dostawcą usług logistycznych.

# Warstwa prezentacji

Warstwa prezentacji dla platformy e-commerce zawiera następujące elementy:

1. **Strony produktów**: Strony te powinny wyświetlać szczegółowe informacje o każdym produkcie, takie jak nazwa, opis, cena i zdjęcie. Powinny również umożliwiać klientom dodawanie produktu do koszyka.
2. **Strona koszyka**: Strona ta powinna wyświetlać wszystkie produkty dodane do koszyka przez klienta. Powinna umożliwiać klientom zmianę ilości każdego produktu w koszyku, usuwanie produktów z koszyka i składanie zamówienia.
3. **Strony zamówień**: Strony te powinny wyświetlać szczegółowe informacje o każdym zamówieniu, takie jak lista produktów, łączna kwota, status zamówienia i szczegóły dostawy. Powinny również umożliwiać klientom śledzenie statusu swojego zamówienia.
4. **Strony użytkowników**: Strony te powinny umożliwiać użytkownikom rejestrację, logowanie i wylogowywanie, a także aktualizację swojego profilu.

# Baza danych

Baza danych została zaimplementowana automatycznie w programie Visual Studio jako lokalna baza danych. Zawiera dane o klientach, produktach, kategorie i zawartości koszyka.

# Diagram klas UML

(w osobnym pliku, opisac co i jak)

Przykłady:

<https://creately.com/diagram/example/h8fya1lm1/ecommerce-uml-class-diagram>

<https://online.visual-paradigm.com/pl/community/share/ecommerce-app-l6ibu78xn>

# Dokumentacja techniczna kodu

Projekt jest dostępny pod poniższym linkiem:

<https://github.com/Anndrida/e-commerce-project_51412_55242_51418_51424_55264/tree/main/ECommerce>