



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Specjalność: PROGRAMOWANIE

Filip Liszcz
Nr albumu studenta w69172

Aplikacja ecommerce dla szkółki roślin

Promotor: dr hab. inż. Teresa Mroczek, prof. WSiZ

PRACA DYPLOMOWA INŻYNIERSKA

Rzeszów 2025

Spis treści

Wstęp. Cel i zakres pracy	4
1 Analiza istniejących rozwiązań w dziedzinie pracy	5
1.0.1 Wady i zalety poszczególnych rozwiązań	5
1.0.2 Podsumowanie analizy	6
2 Uzasadnienie wyboru technologii	7
3 Koncepcja systemu	10
3.0.1 Funkcjonalność i wymagania systemu	10
4 Implementacja najważniejszych rozwiązań	12
5 Testy	13
6 Prezentacja aplikacji	14
7 Podsumowanie	15
Streszczenie	16
Bibliografia	17

Wstęp

W ostatnich latach zaobserwowano dynamiczny wzrost popularności sklepów e-commerce, co wynika z postępującej cyfryzacji społeczeństwa oraz zmieniających się nawyków zakupowych konsumentów. Trend ten został dodatkowo wzmacniony przez pandemię COVID-19, która rozpoczęła się w 2020 roku. W Polsce, w 2020 roku, odnotowano około 40% wzrost w rynku e-commerce, a wartość rynku przekroczyła 92 miliardy złotych [1].

Pandemia zmusiła wiele firm do przeniesienia działalności do internetu, czyniąc sprzedaż online nie tylko wygodną opcją, ale i strategiczną koniecznością. W tym kontekście platformy e-commerce zyskały znaczenie jako kluczowe narzędzie dla przedsiębiorstw reprezentujących różnorodne branże, w tym te bardziej niszowe, takie jak sektor ogrodniczy.

Szkółki roślinne również odnajdują swoje miejsce w sektorze ecommerce. Dzięki platformom sprzedażowym online, szkoły mogą dotrzeć do szerszego grona klientów, oferując im łatwy dostęp do szerokiej gamy roślin oraz szczegółowe informacje o nich. Jednakże, sprzedaż roślin online wiąże się z pewnymi wyzwaniami, takimi jak zapewnienie jakości roślin podczas transportu oraz efektywne zarządzanie stanem magazynowym.

Celem pracy jest opracowanie funkcjonalnej i intuicyjnej platformy, która zapewni klientom łatwy dostęp do szerokiej oferty roślin, a jednocześnie umożliwia szkoły efektywne zarządzanie procedami sprzedaży i kontaktami z odbiorcami. W ramach pracy przeanalizowane zostaną istotne kwestie techniczne, takie jak dobór odpowiednich technologii, projektowanie interfejsu użytkownika oraz integracja z systemami płatności. Szczególna uwaga poświęcona zostanie specyficznym potrzebom branży roślinnej, obejmującym m.in. prezentację produktów oraz kontrolę dostępności roślin na szkółce.

Rozdział 1

Analiza istniejących rozwiązań w dziedzinie pracy

Analiza istniejących platform e-commerce dla sklepów z roślinami pozwala lepiej zrozumieć ich podejście do projektowania kluczowych funkcjonalności oraz tworzenia intuicyjnych doświadczeń użytkownika. Na potrzeby porównania wybrano trzy popularne sklepy internetowe działające w Polsce:

- Rośliny doniczkowe z całego świata | Sklep internetowy | Uroslo.pl [2],
- Internetowy sklep ogrodniczy | PodkarpackieSady.pl [3],
- Internetowy sklep ogrodniczy | Florini.pl [4].

Uroslo.pl

Wyróżnia się bardzo nowoczesnym designem oraz dynamicznymi animacjami, takimi jak animacje ładowania strony. Przyciąga to uwagę klientów i tworzy atrakcyjne wrażenia wizualne. Strona jest również dostosowana do urządzeń mobilnych, oferując logiczny układ komponentów. Jednak nowi użytkownicy mogą czuć się przytłoczeni dużą liczbą elementów interfejsu, animacji oraz dostępnych opcji.

PodkarpackieSady.pl

Animacje zostały ograniczone do minimum – jedynie na stronie głównej znajduje się slider ze zdjęciami. Interfejs cechuje się minimalistycznym, a zarazem funkcjonalnym designem. Umożliwia logowanie się za pośrednictwem Facebooka oraz konta Google, co staje się coraz powszechniejszym rozwiązaniem w nowoczesnych stronach internetowych.

Florini.pl

Sklep posiada przejrzysty design oraz rozbudowany zestaw filtrów dotyczących wyszukiwania roślin. Oferuje również sekcję z odpowiedziami na najczęściej zadawane pytania klientów (FAQ). W wersji mobilnej brakuje jednak rozwijanego menu, które obecnie stanowi standard w projektowaniu responsywnych interfejsów.

1.0.1 Wady i zalety poszczególnych rozwiązań

Wybrane do analizy sklepy e-commerce prezentują różne strategie projektowania nowoczesnych sklepów internetowych z roślinami. Zostały one dobrane ze względu na popularność oraz zróżnicowane podejście do designu i funkcjonalności, które wpływają na ogólne doświadczenia użytkownika. W tabeli na następnej stronie przedstawiono najważniejsze aspekty, mające największy wpływ na komfort korzystania z platformy.

Funkcjonalność	Uroslo	Podkarp.Sady	Florini
Zakupy bez rejestracji	Tak	Tak	Tak
Filtrowanie produktów	Rozbudowane	Proste	Zaawansowane
Logowanie przez inne platformy	Nie	Tak	Tak
Sekcja FAQ	Nie	Nie	Tak
Responsywność	Bardzo dobra	Bardzo dobra	Wymaga dostosowania
Design	Nowoczesny	Prosty	Przejrzysty
Intuicyjność interfejsu	Średnia	Bardzo dobra	Dobra

Tabela 1.1: Porównanie funkcjonalności sklepów e-commerce

Każda z przedstawionych funkcjonalności pozytywnie wpływa na komfort korzystania z platformy. Im więcej dobrze wdrożonych rozwiązań oferuje sklep, tym lepsze jest ogólne doświadczenie użytkownika. Z kolei brak kluczowych funkcji lub ich niedostateczna jakość może utrudniać nawigację i obniżać satysfakcję klientów. Pokazuje to, jak istotne jest zrównoważone podejście do projektowania nowoczesnych platform e-commerce.

1.0.2 Podsumowanie analizy

Każdy z analizowanych sklepów ma swoje mocne strony. PodkarpackieSady wyróżnia się przyjaznym interfejsem oraz możliwością logowania przez zewnętrzne aplikacje, co zwiększa wygodę użytkowania. Florini oferuje zaawansowane filtrowanie produktów oraz sekcję FAQ, co znacznie ułatwia wyszukiwanie i uzyskiwanie potrzebnych informacji. Z kolei Uroslo przyciąga uwagę nowoczesnym designem i atrakcyjnym wyglądem, jednak jego interfejs mógłby zostać uproszczony i zoptymalizowany pod kątem płynności działania.

Wszystkie analizowane sklepy umożliwiają zakupy bez rejestracji, co obecnie stanowi standard w branży e-commerce. Florini natomiast wymaga dopracowania w zakresie responsywności mobilnej.

Wybór najlepszego sklepu zależy od indywidualnych priorytetów użytkownika. Jeśli najważniejsze są prostota i wygodna obsługa, najlepszym wyborem będzie PodkarpackieSady. Jeśli użytkownik ceni sobie szeroki dostęp do informacji oraz precyzyjne filtrowanie, bardziej odpowiedni okaże się Florini. Dla osób przywiązujących dużą wagę do nowoczesnego wyglądu i wizualnych doświadczeń Uroslo może być najbardziej atrakcyjną opcją – mimo że wymaga jeszcze kilku usprawnień.

Rozdział 2

Uzasadnienie wyboru technologii

Do stworzenia aplikacji e-commerce dla szkółki roślin dostępnych jest wiele technologii, zarówno dla warstwy frontendowej, jak i backendowej. Wśród popularnych frameworków frontendowych można wymienić React, Angular czy Vue.js, natomiast dla backendu często stosuje się Spring Boot, Django, Express.js lub NestJS. Jeśli chodzi o bazy danych, istnieją rozwiązania relacyjne, takie jak PostgreSQL czy MySQL, oraz nierelacyjne, np. MongoDB. Kontrola wersji jest nieodłącznym elementem projektów programistycznych, a GitHub pozostaje jednym z najczęściej wybieranych narzędzi w tym zakresie. Wybór konkretnych technologii zależy od takich czynników, jak skalowalność, popularność, wsparcie społeczności, dostępność bibliotek oraz zgodność z najlepszymi praktykami programistycznymi.

Podjęcie ostatecznej decyzji zależało od szeregu czynników, w tym dostępności bibliotek i narzędzi. W kolejnych podrozdziałach przedstawiono wybrane technologie oraz argumenty uzasadniające ich zastosowanie w ramach realizacji tego projektu.

Język programowania: TypeScript

TypeScript został wybrany jako główny język programowania ze względu na jego zaawansowane możliwości w porównaniu do standardowego JavaScriptu. Wprowadza statyczne typowanie, co pozwala na wczesne wychwytywanie błędów w kodzie, zanim aplikacja zostanie uruchomiona. Stacyczne typy ułatwiają utrzymanie przejrzystości i jakości kodu, co staje się kluczowe w miarę rozrastańcia się projektu. Ponadto TypeScript wykorzystywany jest zarówno po stronie frontendu, jak i backendu, co zwiększa prostotę pisania projektu i eliminuje konieczność znajomości dodatkowego języka dla backendu. TypeScript cieszy się także dużą popularnością w środowisku programistycznym, co przekłada się na szeroką społeczność oraz dostępność licznych bibliotek i narzędzi.

Frontend

React z Vite

Frontend aplikacji oparto na bibliotece React w połączeniu z narzędziem budującym Vite. React został wybrany ze względu na swoją architekturę opartą na komponentach, która ułatwia tworzenie reusable'nych i łatwych w utrzymaniu interfejsów użytkownika. Mechanizm wirtualnego DOM-u zapewnia wysoką wydajność, co jest istotne dla dynamicznych elementów sklepu, takich jak listy produktów czy koszyk zakupowy. Popularność Reacta przekłada się na dostępność gotowych komponentów i szeroką społeczność, co przyspiesza rozwój aplikacji. Z kolei Vite wyróżnia się szybkością działania dzięki bły-skawicznemu uruchamianiu.

Zustand

Zustand to lekka biblioteka do zarządzania stanem w React, oparta na prostym API hooków. Umożliwia tworzenie globalnego „store” bez zbędnego boilerplate’u i bez konieczności stosowania Context API czy reduktorów. Dzięki reaktywnemu modelowi każda zmiana wartości w store jest automatycznie propagowana do powiązanych komponentów, co eliminuje potrzebę ręcznego odświeżania strony i zapewnia płynną aktualizację interfejsu.

Tailwind CSS

Tailwind CSS to framework utility-first oparty na predefiniowanych klasach, który przyspiesza tworzenie niestandardowych interfejsów. Główną zaletą jest możliwość szybkiego nakładania stylów za pomocą gotowych klas, co zapewnia spójność wizualną w całej aplikacji. Stylizacja odbywa się bezpośrednio w plikach komponentów, bez konieczności utrzymywania oddzielnych arkuszy CSS, co upraszcza strukturę projektu.



Rysunek 2.1: Frontendowe technologie

Backend

NestJS

zostało wybrane jako główne narzędzie backendowe ze względu na czytelną, modułową strukturę, która ułatwia podział kodu na osobne, samodzielne elementy. Pełna zgodność z TypeScriptem gwarantuje spójność typów po stronie serwera i klienta, zmniejszając ryzyko błędów i przyspieszając integrację z frontendem. Wbudowany mechanizm wstrzykiwania zależności upraszcza zarówno pisanie testów, jak i zarządzanie poszczególnymi komponentami aplikacji.

Ponadto NestJS dostarcza zestaw gotowych rozwiązań (routing, walidacja, łączenie z bazą danych), co pozwala skupić się na specyfice projektu zamiast na implementowaniu podstawowych funkcji. Dokumentacja oraz aktywna społeczność wspierają szybkie odnajdywanie przykładów i najlepszych praktyk, co przekłada się na płynniejszy i bardziej przewidywalny przebieg prac. Dzięki tym cechom NestJS stanowi stabilne i skalowalne środowisko dla rozwoju rozbudowanych aplikacji.

Baza danych: PostgreSQL i TypeORM

Do realizacji warstwy przechowywania danych w aplikacji e-commerce wybrano PostgreSQL w połączeniu z TypeORM, by w pełni wykorzystać zalety obu rozwiązań z perspektywy programisty. PostgreSQL, jako dojrzała relacyjna baza danych, umożliwia definiowanie złożonych zapytań i gwarantuje zgodność z zasadami ACID. Z kolei TypeORM dostarcza gotowe, typowane metody do operacji CRUD,

dzięki czemu nie trzeba ręcznie pisać zapytań SQL — wystarczy wywołanie funkcji, np. repository.find() czy repository.save(). Efektem jest czytelniejszy i bezpieczniejszy kod — już na etapie komplikacji TypeScript wychwyci niezgodności z modelem danych, a gotowe migracje schematu bazy generowane przez TypeORM ułatwiają wprowadzanie zmian i ich wersjonowanie. Dzięki temu tworzenie i utrzymanie logiki biznesowej staje się szybsze, bardziej intuicyjne i mniej podatne na błędy.



Rysunek 2.2: Backendowe technologie

Bezpieczeństwo: JWT

JSON Web Token służy do bezpiecznego przekazywania informacji między klientem a serwerem w postaci podpisanych tokenów. W projekcie e-commerce JWT pełni rolę mechanizmu uwierzytelniania i autoryzacji, umożliwiając bezstanowe zarządzanie sesjami użytkowników bez konieczności przechowywania ich po stronie serwera. Dzięki kryptograficznemu podpisowi oraz możliwości precyzyjnego definiowania czasu życia tokenu, JWT zapewnia integralność danych, ochronę przed nieautoryzowaną modyfikacją oraz elastyczność w kontroli dostępu.

Kontrola wersji: GitHub

Do zarządzania wersjami kodu wykorzystano GitHub, najpopularniejszą platformę hostingową dla repozytoriów Git. Umożliwia ona pracę na branchach, co pozwala na eksperymentowanie z nowymi funkcjonalnościami bez wpływu na główną linię kodu, a także zapewnia backup i możliwość śledzenia zmian. Funkcje takie jak pull requesty czy śledzenie błędów wspierają dobre praktyki programistyczne, nawet w projekcie indywidualnym. W porównaniu do GitLaba czy Bitbucketa, GitHub wyróżnia się powszechnością użycia i integracją z innymi narzędziami, co czyni go naturalnym wyborem.

Rozdział 3

Koncepcja systemu

Celem projektu było stworzenie dobrze zoptymalizowanej oraz intuicyjnej strony E-commerce dedykowanej sprzedaży roślin ozdobnych. Docelową grupą odbiorców jest szerokie grono osób – zarówno młodszych, jak i starszych. System ma zapewnić użytkownikom intuicyjne i funkcjonalne narzędzie do przeglądania produktów, umożliwić szybkie oraz bezproblemowe zakupy, a także pozwolić na aktywną interakcję ze społecznością poprzez możliwość wystawiania recenzji zakupionych roślin. Został opracowany z myślą o zróżnicowanej grupie użytkowników, tak aby był łatwy w obsłudze zarówno dla osób mniej doświadczonych, jak i tych obeznanych z technologią. System został podzielony na role dedykowane dla użytkowników. Każda z nich zapewnia wspólne oraz unikalne funkcjonalności.

Zalogowany użytkownik To użytkownicy, którzy wcześniej się zarejestrowali i są zalogowani na stronie, albo skorzystali z funkcji logowania poprzez serwis Google. Mogą edytować swoje dane – zmieniać informacje o sobie oraz adres do wysyłki. Mają możliwość dodawania produktów do koszyka, dokonywania zakupów oraz wystawiania recenzji produktom po pomyślnym zakupie.

Niezalogowany użytkownik Osoby odwiedzające stronę bez konieczności zakładania konta czy logowania. Ich dostęp do funkcjonalności jest nieco ograniczony w porównaniu do zalogowanych użytkowników. Mogą przeglądać produkty, dodawać je do koszyka oraz dokonywać zakupów – przy czym podczas zakupu muszą podać dane do wysyłki.

Admin Użytkownicy odpowiedzialni za prowadzenie sklepu internetowego. Mogą zarządzać produktami – dodawać, edytować oraz usuwać je. Obsługują zamówienia, zmieniając status wysyłanych paczek, a także zarządzają użytkownikami poprzez przeglądanie listy zarejestrowanych klientów oraz monitorowanie historii zamówień.

3.0.1 Funkcjonalność i wymagania systemu

Celem tego podrozdziału jest przedstawienie kluczowych funkcjonalności systemu oraz wymagań nie-funkcjonalnych. Poniższe mechanizmy dotyczą zarówno klientów, jak i administratorów sklepu.

Wymagania funkcjonalne

Rejestracja / Logowanie

Użytkownicy mają możliwość zalogowania się do sklepu internetowego. W standardowej rejestracji użytkownik podaje swoje dane – imię, nazwisko, adres zamieszkania oraz numer telefonu. Po poprawnej rejestracji może się zalogować. Użytkownik logujący się przez Google nie musi się dodatkowo rejestrować. Po poprawnej rejestracji biblioteka **Bcrypt** [?] haszuje hasło w bazie danych. Poprawne logowanie generuje **JWT** [?] oraz **Refresh Token** [?], które zwiększą bezpieczeństwo systemu.

Przegląd produktów Użytkownik ma możliwość przeglądania sklepu. Jeśli dany produkt go zainteresuje, może kliknąć na produkt – zostanie przekierowany na stronę szczegółową, gdzie znajdują się zdjęcia, tytuł, opis oraz recenzje od innych użytkowników.

Dodawanie produktów do koszyka Zarówno użytkownicy zarejestrowani, jak i niezarejestrowani mogą dodawać produkty do koszyka. Dla niezarejestrowanych użytkowników system zapisuje stan koszyka w pamięci cache, natomiast dla zalogowanych użytkowników zapisuje się w bazie danych.

Proces zakupowy Realizacja zamówienia przebiega w kilku krokach: przejście do podsumowania, wprowadzenie danych do wysyłki (w zależności od typu użytkownika) oraz wybór metody płatności. Po pomyślnym opłaceniu przesyłki zamówienie zostaje dodane do historii użytkownika, a administrator otrzymuje je w panelu zarządzania.

Śledzenie zamówienia Dla zalogowanych użytkowników, po dokonaniu zakupu, w sekcji "Moje zamówienia" wyświetla się szczegółowy opis zamówienia wraz z jego aktualnym statusem.

Wystawianie recenzji Funkcjonalność dostępna jest wyłącznie dla zalogowanych użytkowników, którzy mają dany produkt w historii zakupów. Użytkownik może wystawić ocenę w postaci gwiazdek oraz opcjonalnie dodać komentarz i zdjęcie.

Zarządzanie produktami Administratorzy mają dostęp do panelu administracyjnego, który umożliwia tworzenie, edycję oraz usuwanie produktów. Podczas edycji mogą zmieniać tytuł, opis oraz dostępność produktu (ilość).

Zmiana statusu zamówień Administrator ma możliwość zmieniania statusu zamówień. Domyslnym statusem zamówienia jest "Przyjęto do sklepu". Następnie administrator może zmieniać status, na przykład na "Wysłano" lub "Zakończono", informując w ten sposób klientów o postępie realizacji.

Monitorowanie historii zamówień Administratorzy mają dostęp do listy zarejestrowanych użytkowników oraz ich zamówień. Dzięki temu mogą łatwo zarządzać statusem zamówień i monitorować aktywność klientów.

Wymagania niefunkcjonalne

Dostępność - System powinien być dostępny 24/7 z maksymalnym czasem przestoju nie przekraczającym 0,1% rocznie.

Wydajność - System powinien obsłużyć co najmniej 1000 jednoczesnych użytkowników bezauważalnego spadku wydajności, z czasem odpowiedzi na zapytania nie przekraczającym 2 sekund.

Zabezpieczenia - Wszystkie dane użytkowników (np. hasła, dane do wysyłki) powinny być szyfrowane za pomocą protokołu SSL/TLS, a dostęp do panelu administratora powinien być chroniony dodatkowym uwierzytelnianiem dwuskładnikowym.

Skalowalność - System powinien być zaprojektowany w sposób umożliwiający łatwe zwiększenie zasobów serwerowych w miarę wzrostu liczby użytkowników.

Użyteczność - Interfejs użytkownika powinien być intuicyjny i dostępny na urządzeniach mobilnych oraz desktopowych, z responsywnym designem dostosowanym do różnych rozdzielczości ekranów.

Kompatybilność - System powinien działać poprawnie na popularnych przeglądarkach (Chrome, Firefox, Safari, Edge) w ich najnowszych wersjach.

Przechowywanie danych - Dane zamówień i użytkowników powinny być przechowywane przez co najmniej 5 lat w celu zapewnienia historii transakcji i zgodności z przepisami.

Niezawodność - System powinien automatycznie tworzyć kopie zapasowe danych co 24 godziny, aby zminimalizować ryzyko utraty danych.

Rozdział 4

Implementacja najważniejszych rozwiązań

Rozdział 5

Testy

Rozdział 6

Prezentacja aplikacji

Rozdział 7

Podsumowanie

Streszczenie

Bibliografia

- [1] Mazars Polska, Noerr, SpotData, „Rynek e-commerce w Polsce: chwilowe przyspieszenie czy trwała zmiana? Perspektywy i strategie rozwoju firm”, Warszawa, 25 marca 2020.
- [2] Uroslo.pl, „Rośliny doniczkowe z całego świata – Sklep internetowy Uroslo.pl”, dostęp: 15.06.2025, <https://www.uroslo.pl/>
- [3] Podkarpackie Sady, „Internetowy sklep ogrodniczy – PodkarpackieSady.pl”, dostęp: 15.06.2025, <https://www.podkarpackiesady.pl/>
- [4] Florini, „Internetowy sklep ogrodniczy – Florini.pl”, dostęp: 15.06.2025, <https://www.florini.pl/>
- [5] Bcrypt, „bcrypt – biblioteka do haszowania haseł (npm)”, dostęp: 15.06.2025, <https://www.npmjs.com/package/bcrypt>
- [6] JWT.io, „JSON Web Token (JWT)”, dostęp: 15.06.2025, <https://jwt.io/>
- [7] „Refresh token – mechanizm odświeżania tokenów w autoryzacji”, dostęp: 15.06.2025, <https://shorturl.at/joRa2>