

Celem projektu jest stworzenie webowej aplikacji typu marketplace, umożliwiającej użytkownikom przeglądanie, wyszukiwanie oraz zakup produktów wystawionych przez innych użytkowników. Aplikacja pozwala na zarządzanie kontem, ofertami, koszykiem oraz historią zamówień. Projekt kładzie nacisk na czytelny i intuicyjny interfejs użytkownika oraz poprawną integrację z relacyjną bazą danych. Na samym dole tego pliku znajduje się wstępny zarys interfejsu – głównej strony aplikacji.

Główne funkcjonalności:

Jakub Mierzejewski (337264)

1. Moduł użytkownika (rejestracja, logowanie, profil)

Odpowiada za obsługę tworzenia konta, logowania oraz sesji użytkownika. Implementuje walidacje formularzy oraz komunikację z backendem i bazą danych. Zapewnia możliwość edycji podstawowych danych profilu.

2. Koszyk i proces zakupowy

Tworzy funkcjonalność dodawania produktów do koszyka oraz ich usuwania, wraz z bieżącą walidacją zawartości. Implementuje mechanizm „Kup teraz”, generujący zamówienie na podstawie koszyka. Odpowiada za zapis zamówienia w bazie (Orders, OrderItems, CartItems).

3. Historia zamówień + testy modułów użytkownika i zakupów

Realizuje widok historii zakupów pobieranej z bazy danych. Utrzymuje kompletny przepływ zakupowy po stronie użytkownika. Przygotowuje testy jednostkowe związane z logowaniem, koszykiem i generowaniem zamówień.

Mikołaj Kielak (337243)

1. System ofert (CRUD ofert + zdjęcia produktów)

Projektuje i implementuje pełny moduł ofert, obejmujący dodawanie, edytowanie i usuwanie produktów. Odpowiada za obsługę zdjęć dla ofert oraz integrację z backendem i bazą danych. Zapewnia czytelny interfejs tworzenia i przeglądania ofert.

2. Kategorie produktów (CRUD + filtrowanie)

Tworzy hierarchiczny system kategorii oraz ich obsługę administracyjną. Implementuje relację produktów do kategorii oraz filtrowanie ofert po wybranych kategoriach. Dbą o poprawne wyświetlanie kategorii w interfejsie użytkownika.

3. Ulubione + wyszukiwanie i sortowanie ofert + testy modułu ofert

Odpowiada za mechanizm ulubionych produktów - dodawanie, usuwanie i prezentację listy ulubionych użytkownika. Implementuje wyszukiwarkę ofert oraz sortowanie po cenie, nazwie i innych parametrach. Wykonuje testy endpointów związanych z ofertami, kategoriami i ulubionymi.

Sebastian Stański (337288)

1. Panel administracyjny (użytkownicy, oferty, uprawnienia)

Projektuje i implementuje panel administratora, umożliwiający przeglądanie użytkowników i ofert. Dodaje funkcje dezaktywacji kont i ofert oraz zarządzania uprawnieniami. Zapewnia spójny podział ról oraz weryfikację dostępu.

2. Projekt i implementacja bazy danych + migracje

Tworzy kompletny model relacyjnej bazy danych wraz z głównymi tabelami (Users, Products, Categories, CartItems, Orders, OrderItems, Favorites, ProductImages). Odpowiada za konfigurację kluczy obcych, relacji oraz migracje. Przygotowuje wstępne dane i seedery.

3. CI/CD oraz testy integracyjne

Konfiguruje środowisko ciągłej integracji z wykorzystaniem GitHub Actions, obejmujące testy, budowanie backendu i frontendu. Odpowiada za testy integracyjne kluczowych ścieżek użytkownika: rejestracja, publikacja oferty, złożenie zamówienia. Dbą o automatyczne sprawdzanie jakości projektu.

Cele projektu:

Zaprojektowanie i stworzenie funkcjonalnego interfejsu zgodnie z dobrymi praktykami UI/UX.

Wykorzystanie relacyjnej bazy danych do operacji CRUD na ofertach, użytkownikach i zamówieniach.

Wdrożenie mechanizmów zarządzania stanem aplikacji, koszykiem i sesją użytkownika.

Stworzenie testów jednostkowych i integracyjnych dla kluczowych modułów.

Wprowadzenie CI/CD (np. GitHub Actions) do automatyzacji testów i budowania projektu.

Repozytorium z przejrzystą historią zmian oraz podziałem pracy zespołu.

Planowane technologie do wykorzystania przy implementacji:

Frontend: React / Next.js, Backend: Node.js (NestJS / Express) lub Java Spring Boot / Django – do ustalenia

Baza danych: PostgreSQL / MySQL, Kontrola wersji: Git + GitHub, CI: GitHub Actions

Testy: Jest / JUnit / PyTest (zależnie od backendu)

Wstępna makieta, zarys interfejsu:

