

## Opis pracy inżynierskiej

### Temat

Optymalizacja procesów logistycznych z wykorzystaniem aplikacji webowej do zleceń transportowych w celu eliminacji nieefektywnego wykorzystania zasobów

Optimization of Logistic Processes Using a Web Application for Transport Orders to Eliminate Inefficient Use of Resources

Krystian Tomczyk, 264173

### Harmonogram

#### Rozdział I: Planowanie i Projektowanie

##### 1. Analiza wymagań:

- Szczegółowe zebranie i analiza wymagań funkcjonalnych i niefunkcjonalnych aplikacji.
- Określenie grupy docelowej użytkowników.
- Opracowanie przypadków użycia (use cases).

##### 2. Projektowanie architektury:

- Wybór technologii (TypeScript, React.js, Next.js, Tailwind CSS, Node.js, Express.js, PostgreSQL).
- Projektowanie struktury bazy danych (schemat, tabele, relacje).

- Projektowanie interfejsu API REST.
- Stworzenie makiet interfejsu użytkownika (UI).

## **Rozdział II: Implementacja Backend**

### **Konfiguracja środowiska:**

- Utworzenie repozytorium kodu źródłowego.
- Konfiguracja środowiska Node.js i TypeScript.
- Instalacja i konfiguracja PostgreSQL.

#### **1. Implementacja modeli danych:**

- Stworzenie modeli danych reprezentujących użytkowników, ogłoszenia, zlecenia, trasy itp.
- Implementacja logiki walidacji danych.

#### **2. Implementacja kontrolerów:**

- Stworzenie kontrolerów obsługujących żądania HTTP do endpointów API.
- Implementacja logiki biznesowej w usługach (services).

#### **3. Implementacja uwierzytelniania i autoryzacji:**

- Implementacja systemu rejestracji i logowania użytkowników.
- Zabezpieczenie endpointów API za pomocą tokenów JWT.

## **Rozdział III: Implementacja Frontend**

#### **1. Konfiguracja środowiska:**

- Konfiguracja środowiska React.js i Next.js.
- Instalacja i konfiguracja Tailwind CSS.

#### **2. Implementacja komponentów:**

- Stworzenie komponentów reprezentujących poszczególne elementy interfejsu użytkownika.
- Implementacja logiki interakcji z API.

#### **3. Implementacja routingu:**

- Konfiguracja routingu Next.js do nawigacji między widokami.

#### **4. Stylowanie interfejsu:**

- Zastosowanie Tailwind CSS do stylizacji komponentów.

## **Rozdział IV: Testowanie i Integracja**

### **1. Testy jednostkowe:**

- Napisanie testów jednostkowych dla komponentów frontendowych i modułów backendowych.

### **2. Integracja komponentów:**

- Połączenie frontendu i backendu w działającą całość.

## **Rozdział V: Wdrożenie i Utrzymanie**

### **1. Wdrożenie na serwer produkcyjny:**

- Przygotowanie środowiska produkcyjnego.
- Wdrożenie aplikacji na serwer.

### **2. Monitorowanie i utrzymanie:**

- Monitorowanie działania aplikacji.
- Naprawa błędów i wprowadzanie poprawek.

## **Cel projektu**

Aplikacja będzie umożliwiać przeglądanie ogłoszeń zamieszczanych przez użytkowników zlecających transport materiałów, produktów lub osób z punktu A do punktu B. Głównym celem aplikacji jest zminimalizowanie problemu przejazdów pojazdów ciężarowych bez ładunku, na przykład w drodze powrotnej. Aplikacja przeznaczona będzie na rynek międzynarodowy, oferując różne wersje językowe.

# Funkcje aplikacji

## Dla użytkowników zlecających transport

1. Przeglądanie ogłoszeń transportowych:
  - 1.1. Filtrowanie ogłoszeń według kryteriów takich jak: trasa, rodzaj przewożonego towaru, wymagane kwalifikacje, cena.
  - 1.2. Sortowanie ogłoszeń według daty, ceny, odległości.
2. **Dodawanie ogłoszeń:**
  - 2.1. Określenie punktów początkowego i końcowego trasy (A i B).
  - 2.2. Wybór kategorii przewożonego towaru (np. materiały budowlane, meble) lub wskazanie transportu osób.
  - 2.3. Podanie wymiarów i ilości przewożonego towaru lub liczby osób.
  - 2.4. Określenie wymaganych kwalifikacji kierowcy (opcjonalnie).
  - 2.5. Ustalenie stawki za przewóz (opcjonalnie, z możliwością negocjacji).
3. **Kontakt z przewoźnikami:**
  - 3.1. Możliwość bezpośredniego kontaktu z przewoźnikami poprzez wewnętrzny system wiadomości.
  - 3.2. Zawieranie umów na przewóz zasobów bezpośrednio w aplikacji.
  - 3.3. Opcja przedpłaty za transport (opcjonalnie).

## Dla użytkowników świadczących usługi transportowe

1. **Przeglądanie ogłoszeń zleceń transportowych:**
  - 1.1. Dostęp do szczegółowych informacji o zleceniach (trasa, rodzaj towaru, wymagane kwalifikacje, oferowana stawka).
2. **Przyjmowanie zleceń:**
  - 2.1. Możliwość akceptacji zlecenia bezpośrednio w aplikacji.
  - 2.2. Formularz zobowiązania do wykonania transportu.
3. **Dodawanie ogłoszeń o dostępnych trasach:**
  - 3.1. Określenie planowanej trasy przejazdu.
  - 3.2. Podanie daty i godziny wyjazdu.
  - 3.3. Wskazanie dostępnych miejsc w pojeździe.
  - 3.4. Określenie stawki za przewóz (opcjonalnie, z możliwością negocjacji).
4. **Kontakt ze zleceniodawcami:**
  - 4.1. Możliwość bezpośredniego kontaktu ze zleceniodawcami poprzez wewnętrzny system wiadomości.
  - 4.2. Zawieranie umów na przewóz zasobów bezpośrednio w aplikacji.

## Funkcjonalności ogólne

1. **System rejestracji i logowania:**
  - 1.1. Tworzenie kont użytkowników.
  - 1.2. Bezpieczne logowanie do aplikacji.

## **2. Powiązanie zleceniodawcy ze zleceniobiorcą:**

- 2.1. Automatyczne przypisanie zleceniobiorcy po zaakceptowaniu zlecenia.
- 2.2. Możliwość oceny i wystawiania komentarzy po zrealizowanym transporcie.

## **3. Proponowane ogłoszenia dla danej trasy:**

- 3.1. Wyświetlanie listy zleceń transportowych zbliżonych do trasy wpisanej przez użytkownika.

## **4. Wersje językowe:**

- 4.1. Dostępność aplikacji w różnych językach (np. polski, angielski, niemiecki).

# Aspekt inżynierski

## Dokumentacja użytkowa

### **1. Wprowadzenie**

Aplikacja webowa do zleceń transportowych została stworzona w celu optymalizacji procesów logistycznych poprzez eliminację nieefektywnego wykorzystania zasobów transportowych. Umożliwia ona użytkownikom zarówno zlecenie, jak i oferowanie usług transportowych, łącząc ich potrzeby w jednym miejscu.

### **2. Rejestracja i Logowanie**

Aby korzystać z pełnej funkcjonalności aplikacji, należy utworzyć konto użytkownika. Proces rejestracji jest prosty i wymaga podania podstawowych danych. Po zarejestrowaniu można zalogować się do aplikacji przy użyciu swojego adresu e-mail i hasła.

### **3. Dla Użytkowników Zlecających Transport**

#### **3.1. Przeglądanie Ogłoszeń Transportowych**

W sekcji "Ogłoszenia transportowe" można przeglądać oferty przewoźników. Ogłoszenia zawierają informacje o trasie, rodzaju przewożonego towaru, wymaganych kwalifikacjach oraz cenie. Możliwe jest filtrowanie i sortowanie ogłoszeń według różnych kryteriów, aby znaleźć ofertę najlepiej dopasowaną do potrzeb.

#### **3.2. Dodawanie Ogłoszenia**

Aby zlecić transport, należy dodać ogłoszenie. W formularzu ogłoszenia należy podać:

- 1. Punkty początkowy i końcowy trasy (A i B)
- 2. Rodzaj przewożonego towaru (kategoria, wymiary, ilość) lub liczbę osób
- 3. Wymagane kwalifikacje kierowcy (opcjonalnie)
- 4. Stawkę za przewóz (opcjonalnie, z możliwością negocjacji)

### 3.3. Kontakt z Przewoźnikami i Zawieranie Umów

Po dodaniu ogłoszenia, przewoźnicy mogą się z Tobą skontaktować. Możesz również sam nawiązać kontakt z wybranym przewoźnikiem poprzez wewnętrzny system wiadomości. Po uzgodnieniu szczegółów, istnieje możliwość zawarcia umowy na przewóz zasobów bezpośrednio w aplikacji. Opcjonalnie, można dokonać przedpłaty za transport.

## 4. Dla Użytkowników Świadczących Usługi Transportowe

### 4.1. Przeglądanie Ogłoszeń Zleceń Transportowych

W sekcji "Zlecenia transportowe" można przeglądać ogłoszenia dodane przez użytkowników zlecających transport. Każde ogłoszenie zawiera szczegółowe informacje o zleceniu.

### 4.2. Przyjmowanie Zleceń

Jeśli jesteś zainteresowany zleceniem, możesz je zaakceptować bezpośrednio w aplikacji. Po akceptacji, należy wypełnić formularz zobowiązania do wykonania transportu.

### 4.3. Dodawanie Ogłoszeń o Dostępnych Trasach

Jeśli masz zaplanowaną trasę przejazdu, możesz dodać ogłoszenie informujące o tym. W ogłoszeniu należy podać:

- Planowaną trasę przejazdu
- Datę i godzinę wyjazdu
- Dostępne miejsca w pojeździe
- Stawkę za przewóz (opcjonalnie, z możliwością negocjacji)

### 4.4. Kontakt ze Zleceniodawcami i Zawieranie Umów

Po dodaniu ogłoszenia, zleceniodawcy mogą się z Tobą skontaktować. Możesz również sam nawiązać kontakt z wybranym zleceniodawcą poprzez wewnętrzny system wiadomości. Po uzgodnieniu szczegółów, istnieje możliwość zawarcia umowy na przewóz zasobów bezpośrednio w aplikacji.

## 5. Dodatkowe Funkcje

**Proponowane ogłoszenia:** Po wpisaniu trasy, aplikacja wyświetli listę zleceń transportowych zbliżonych do Twojej trasy.

**System ocen i komentarzy:** Po zrealizowanym transporcie, zarówno zleceniodawca, jak i zleceniobiorca mogą wystawić ocenę i komentarz drugiej stronie.

**Wersje językowe:** Aplikacja dostępna jest w kilku wersjach językowych.

## 6. Pomoc i Wsparcie

W razie pytań lub problemów, prosimy o kontakt z naszym działem obsługi klienta.

### Dokumentacja wprowadzenia

Aplikacja zostanie zbudowana w oparciu o architekturę klient-serwer, gdzie frontend komunikuje się z backendem za pomocą interfejsu API.

**Frontend:** Odpowiada za prezentację danych użytkownikowi oraz interakcję z nim. Został zaimplementowany przy użyciu następujących technologii:

- **TypeScript:** Język programowania będący nadzbiorem *JavaScript*, wprowadzający statyczne typowanie i inne udogodnienia ułatwiające rozwój dużych aplikacji.
- **React.js:** Biblioteka *JavaScript* do budowania interfejsów użytkownika.
- **Next.js:** Framework *React.js* ułatwiający tworzenie aplikacji opartych o renderowanie po stronie serwera (SSR) oraz generowanie statycznych stron (SSG).
- **Tailwind CSS:** Framework CSS do szybkiego stylowania elementów interfejsu.

**Backend:** Odpowiada za obsługę logiki biznesowej, przetwarzanie danych oraz komunikację z bazą danych. Został zaimplementowany przy użyciu następujących technologii:

- **TypeScript:** Język programowania używany do tworzenia logiki serwera.
- **Node.js:** Środowisko uruchomieniowe JavaScript umożliwiające wykonywanie kodu JavaScript (i TypeScript) po stronie serwera.
- **Express.js:** Framework Node.js do tworzenia aplikacji webowych i API.

**Baza Danych:** Aplikacja będzie wykorzystywała bazę danych do przechowywania informacji o użytkownikach, ogłoszeniach, zleceniach, trasach, itp.

- **PostgreSQL:** to zaawansowany system zarządzania relacyjnymi bazami danych, który oferuje bogaty zestaw funkcji, wysoką wydajność, skalowalność oraz niezawodność.

**Komunikacja:** Komunikacja między frontendem a backendem odbywa się za pomocą interfejsu API REST. Frontend wysyła żądania HTTP do backendu, który przetwarza te żądania i zwraca odpowiedź w formacie JSON.

## Wybor technologii

Frontend: **TypeScript** z wykorzystaniem frameworka *react.js* w połączeniu z *next.js* oraz *tailwind CSS*.

Backend: **TypeScript**, ze środowiskiem *Node.js* z wykorzystaniem biblioteki *express*.

Baza danych: **PostgreSQL**

## Implementacja

### Frontend (React.js, Next.js, TypeScript, Tailwind CSS)

- **Struktura projektu:** Projekt frontendowy będzie podzielony na komponenty, które reprezentują poszczególne elementy interfejsu użytkownika (np. formularze, listy ogłoszeń, widoki szczegółów).
- **Zarządzanie stanem:** Do zarządzania stanem aplikacji zostanie wykorzystana biblioteka React Context API lub Redux, w zależności od złożoności projektu.
- **Routing:** Next.js zapewni obsługę routingu, umożliwiając nawigację między różnymi widokami aplikacji.
- **Stylowanie:** Do stylowania komponentów zostanie wykorzystany Tailwind CSS, co pozwoli na szybkie tworzenie responsywnych i atrakcyjnych wizualnie interfejsów.
- **Interakcja z API:** Komponenty frontendowe będą komunikować się z backendem za pomocą biblioteki fetch lub axios, wysyłając żądania HTTP do odpowiednich endpointów API.

### Backend (Node.js, Express.js, TypeScript, PostgreSQL)

- **Struktura projektu:** Projekt backendowy będzie podzielony na moduły odpowiadające za różne funkcjonalności (np. uwierzytelnianie, zarządzanie ogłoszeniami, obsługa zleceń).
- **Modele danych:** Modele danych reprezentujące encje takie jak użytkownicy, ogłoszenia, zlecenia itp. zostaną zdefiniowane przy użyciu biblioteki ORM (np. Sequelize) lub bezpośrednio za pomocą zapytań SQL.
- **Kontrolery:** Kontrolery będą odpowiedzialne za obsługę żądań HTTP przychodzących od frontendu, wywoływanie odpowiedniej logiki biznesowej oraz zwracanie odpowiedzi.



- **Logika biznesowa:** Logika biznesowa będzie implementowana w oddzielnych usługach (services), które będą wykorzystywane przez kontrolery.
- **Walidacja danych:** Dane wejściowe będą walidowane zarówno po stronie frontendu, jak i backendu, aby zapewnić ich poprawność i bezpieczeństwo.

## Testy

### Testy jednostkowe (Unit Tests)

Testy jednostkowe będą służyć do weryfikacji poprawności działania poszczególnych funkcji, metod i klas w izolacji od reszty systemu. Do ich tworzenia zostaną wykorzystane biblioteki takie jak *Jest* i być może dodatkowe biblioteki do mockowania zależności.

## Bibliografia

1. Dokumentacja React.js - <https://react.dev/reference/react>
2. Dokumentacja Next.js - <https://nextjs.org/docs>
3. Dokumentacja Express - <https://expressjs.com/en/4x/api.html>
4. Dokumentacja PostgreSQL - <https://www.postgresql.org/docs/current/index.html>
5. Nowoczesna logistyka - Paul R. Murphy Jr., Donald F. Wood
6. Logistyka międzynarodowa - Marta Wincewicz-Bosy