

<b>AI1</b>	<b>Dokumentacja projektu</b>
<b>Autor</b>	Cyprian Schiffer, 125201
<b>Kierunek, rok</b>	Informatyka, II rok, st. Stacjonarne (3,5-l)
<b>Temat projektu</b>	Obsługa sieci paczkomatów

## Spis treści

### 1. Wstęp

1.1. Opis założeń projektu.....	2
1.2. Fragment rzeczywistości objęty działaniem aplikacji.....	3
1.3. Co użytkownik może zrobić w aplikacji.....	3
1.4. Role i uprawnienia w aplikacji.....	3
1.5. Cele projektu.....	4

### 2. Narzędzia i technologie

2.1. HTML.....	4
2.2. CSS.....	4
2.3. JavaScript.....	5
2.4. PHP.....	5
2.5. Laravel.....	5
2.6. phpMyAdmin.....	5
2.7. Baza Danych.....	6
2.8. Środowisko Programistyczne.....	6
2.9. Licencje i koszty.....	6

### 3. Baza danych

3.1. Schemat ERD.....	7
3.2. Opis zawartości bazy danych i powiązań między tabelami.....	7

### 4. GUI

4.1. Strona główna.....	10
4.2. Nadawanie paczki.....	10
4.3. Panel Admina.....	12
5. Uruchomienie aplikacji	
5.1. Wymagane oprogramowanie.....	13
5.2. Kroki konfiguracyjne.....	13
6. Funkcjonalności aplikacji	
6.1. Rejestracja.....	14
6.2. Logowanie.....	15
6.3. Wysyłanie paczek, widoki paczek.....	16
6.4. CRUD przeprowadzany przez administratora.....	19
6.5. Zarządzanie użytkownikami przez administratora.....	21
6.6. Przeglądanie ogólnodostępnych zasobów .....	22
6.7. Zarządzanie swoimi zasobami przez użytkownika aplikacji.....	23
6.8. Zarządzanie swoimi danymi przez użytkownika aplikacji.....	24
6.9. Logika biznesowa.....	25
6.10. Walidacja danych.....	26
7. Responsywność	
7.1 Użycie kart.....	27

## **1. Wstęp**

### **1.1 Opis założeń projektu**

Projekt „Obsługa Sieci Paczkomatów” to aplikacja internetowa zbudowana w oparciu o framework Laravel. Jej celem jest zapewnienie użytkownikom wygodnego i efektywnego sposobu zarządzania przesyłkami oraz paczkomatami. System obsługuje różnorodne aspekty zarządzania paczkami, począwszy od ich wysyłania i odbierania przez użytkowników, po zarządzanie paczkomatami przez administratorów oraz kurierów.

## **1.2 Fragment rzeczywistości objęty działaniem aplikacji**

Aplikacja obejmuje działanie sieci paczkomatów rozmieszczonych w różnych lokalizacjach, dostępnych publicznie. Umożliwia użytkownikom korzystanie z tych paczkomatów do wysyłania i odbierania paczek, a także przeglądanie dostępnych paczkomatów w ich okolicy. Aplikacja ma na celu zautomatyzowanie i usprawnienie procesu zarządzania przesyłkami, zapewniając użytkownikom szybki i bezproblemowy dostęp do usług paczkomatowych.

## **1.3 Co użytkownik może zrobić w aplikacji**

W ramach aplikacji użytkownicy mogą:

- Przeglądać listę dostępnych paczkomatów, w tym ich lokalizacje.
- Wysyłać paczki do innych użytkowników.
- Śledzić status wysłanych paczek jak i odebranych.
- Śledzić status paczek do niego nadanych.

## **1.4 Role i uprawnienia w aplikacji**

Aplikacja definiuje trzy główne role użytkowników, z różnymi poziomami uprawnień:

- Użytkownik: Może przeglądać dostępne paczkomaty, wysyłać paczki oraz odbierać paczki.
- Kurier: Odpowiedzialny za zarządzanie paczkami klientów, w tym ich odbiór, dostarczenie do paczkomatów oraz aktualizację statusów przesyłek.
- Administrator: Zarządza wszystkimi zasobami systemu, w tym dodawaniem, edytowaniem i usuwaniem paczkomatów, zarządzaniem użytkownikami i kurierami, oraz monitorowaniem działania systemu.

## **1.5 Cele projektu**

Celem projektu jest stworzenie intuicyjnej i funkcjonalnej aplikacji, która umożliwi:

- Użytkownikom wygodne zarządzanie swoimi paczkami.
- Kurierom efektywne zarządzanie procesem dostarczania paczek.
- Administratorom łatwe zarządzanie całą siecią paczkomatów oraz monitorowanie i kontrolę nad działaniem systemu.

## **2. Narzędzia i technologie**

### **2.1 HTML**

Opis: Język znaczników używany do tworzenia i strukturyzacji treści na stronach internetowych.

Dokumentacja: <https://developer.mozilla.org/en-US/docs/Web/HTML>

Licencja: Darmowy

### **2.2 CSS**

Opis: Język stylów używany do opisywania wyglądu dokumentu napisanego w HTML.

Dokumentacja: <https://developer.mozilla.org/en-US/docs/Web/CSS>

Licencja: Darmowy

## **2.3 JavaScript**

Opis: Język programowania używany do tworzenia dynamicznych i interaktywnych elementów na stronach internetowych.

Dokumentacja: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

Licencja: Darmowy

## **2.4 PHP**

Wersja: PHP 8.2.12

Opis: Język skryptowy wykorzystywany do tworzenia dynamicznych stron internetowych i aplikacji webowych.

Dokumentacja: <https://www.php.net/manual/en/>

Licencja: PHP License

## **2.5 Laravel**

Wersja: Laravel 11.x

Opis: Framework PHP do tworzenia aplikacji webowych, który upraszcza procesy związane z routingiem, zarządzaniem bazami danych, migracjami oraz autoryzacją użytkowników.

Dokumentacja: <https://laravel.com/docs/11.x>

Licencja: MIT License

## **2.6 phpMyAdmin**

Opis: Narzędzie służące do zarządzania bazami danych MySQL/MariaDB za pomocą interfejsu webowego.

Dokumentacja: <https://www.phpmyadmin.net/docs/>

Licencja: GNU General Public License

## **2.7 Baza Danych**

### **MySQL/MariaDB**

Opis: Relacyjna baza danych używana do przechowywania danych aplikacji.

Dokumentacja: <https://dev.mysql.com/doc/>

MariaDB Documentation: <https://mariadb.com/kb/en/documentation/>

Licencja:

MySQL: GNU General Public License

MariaDB: MariaDB Public License

## **2.8 Środowisko Programistyczne**

### **Visual Studio Code**

Opis: Lekki, ale potężny edytor kodu źródłowego, który działa na desktopie i jest dostępny dla systemów Windows, macOS i Linux.

Dokumentacja: [Visual Studio Code Documentation](#)

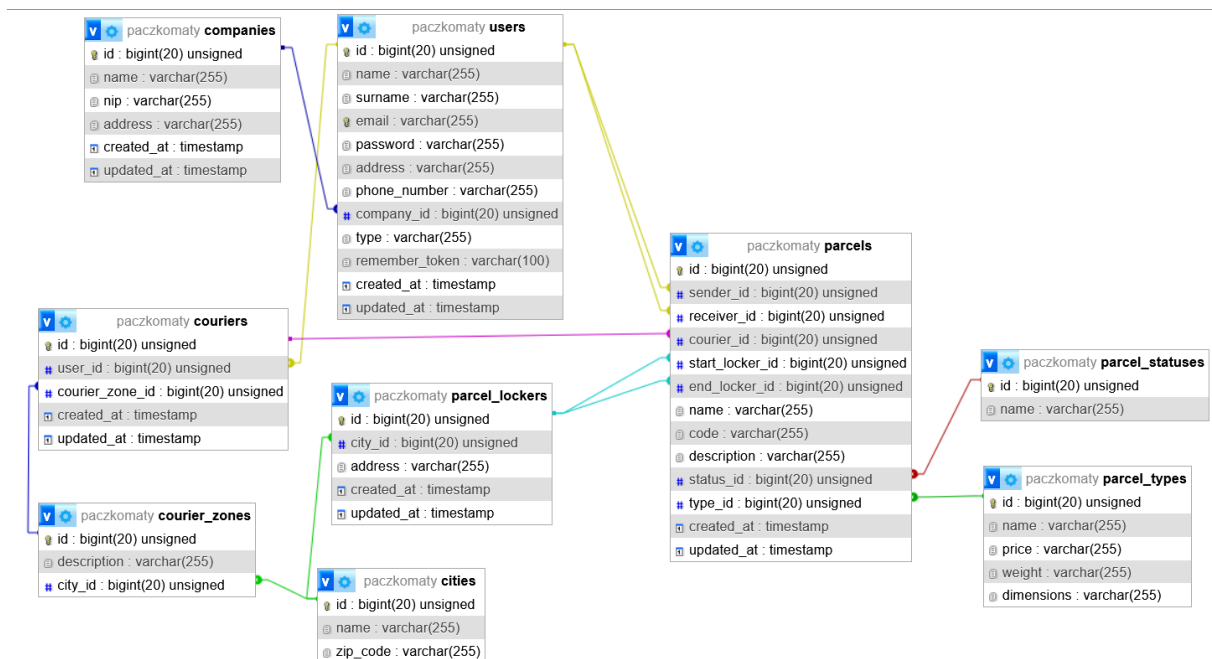
Licencja: Freeware

## **2.9 Licencje i Koszty**

Wszystkie wymienione technologie są darmowe do użytku i posiadają licencje, które pozwalają na ich swobodne wykorzystanie w projektach komercyjnych i niekomercyjnych. PHP, Laravel, HTML, CSS, JavaScript oraz phpMyAdmin są licencjonowane na różnych zasadach, takich jak PHP License, MIT License oraz GNU General Public License, co zapewnia dużą elastyczność i możliwość dostosowania do różnych potrzeb projektowych.

## 3. Baza Danych

### 3.1 Schemat ERD



### 3.2 Opis zawartości i powiązań między tabelami

Baza danych projektu „Obsługa Sieci Paczkomatów” składa się z kilku kluczowych tabel, które przechowują informacje o użytkownikach, firmach, kurierach, strefach kurierów, miastach, paczkomatach oraz paczkach. Poniżej znajduje się krótki opis każdej tabeli:

#### - users

Przechowuje informacje o użytkownikach aplikacji, takie jak ID, imię, nazwisko, email, hasło, itp.

#### **- companies**

Przechowuje informacje o firmach, które mogą być przypisane do użytkowników. Zawiera dane takie jak ID firmy, nazwa, adres, NIP

#### **- couriers**

Przechowuje informacje o kurierach, w tym ID kuriera i przypisane do niego ID użytkownika (user\_id), który jest kontem powiązanym z kurierem.

#### **- courier\_zones**

Przechowuje informacje o strefach kurierów. Każda strefa jest przypisana do konkretnego kuriera za pomocą ID kuriera (courier\_id).

#### **- cities**

Przechowuje informacje o miastach, w tym ID miasta, nazwa, kod pocztowy, itp. Miasta są powiązane zarówno z strefami kurierów (courier\_zones), jak i paczkomatami (parcel\_lockers).

#### **- parcel\_lockers**

Przechowuje informacje o paczkomatach, takie jak ID paczkomatu, lokalizacja (powiązana z ID miasta) oraz adres.

#### **- parcels**

Przechowuje informacje o paczkach, w tym ID paczki, ID nadawcy (sender\_id), ID odbiorcy (receiver\_id), ID kuriera (courier\_id) odpowiedzialnego za przesyłkę, ID paczkomatu początkowego (start\_locker\_id), ID paczkomatu końcowego (end\_locker\_id), ID statusu paczki (status\_id), ID typu paczki (type\_id)

#### **- parcel\_statuses**

Przechowuje informacje o statusach paczek, takich jak ID statusu, nazwa statusu (np. wysłana, w drodze, oczekuje na odbiór).

#### **parcel\_types**

Przechowuje informacje o typach paczek, takich jak ID typu, nazwa typu (np. mała, duża).

### **Opis Powiązań Pomiędzy Tabelami**

Baza danych posiada liczne powiązania między tabelami, które umożliwiają efektywne zarządzanie informacjami o użytkownikach, paczkomatach i przesyłkach. Poniżej znajduje się opis tych powiązań:

**users ↔ companies**



Relacja: Jeden użytkownik może być przypisany do jednej firmy.

Powiązanie: users.company\_id jest kluczem obcym odnoszącym się do company.id.

#### **users ↔ couriers**

Relacja: Jeden użytkownik może być kurierem.

Powiązanie: couriers.user\_id jest kluczem obcym odnoszącym się do users.id.

#### **couriers ↔ courier\_zones**

Relacja: Jeden kurier może być przypisany do jednej strefy.

Powiązanie: couriers.courier\_zone\_id jest kluczem obcym odnoszącym się do courier\_zones.id.

#### **courier\_zones ↔ cities**

Relacja: Jedna strefa kuriera może obejmować jedno miasto.

Powiązanie: courier\_zones.city\_id jest kluczem obcym odnoszącym się do cities.id.

#### **parcel\_lockers ↔ cities**

Relacja: Jeden paczkomat znajduje się w jednym mieście.

Powiązanie: parcel\_lockers.city\_id jest kluczem obcym odnoszącym się do cities.id.

#### **parcels**

Relacja: Paczka posiada wiele powiązań, w tym:

Nadawca paczki (parcels.sender\_id) jest powiązany z tabelą users.

Odbiorca paczki (parcels.receiver\_id) jest powiązany z tabelą users.

Kurier odpowiedzialny za paczkę (parcels.courier\_id) jest powiązany z tabelą couriers.

Początkowy paczkomat (parcels.start\_locker\_id) jest powiązany z tabelą parcel\_lockers.

Końcowy paczkomat (parcels.end\_locker\_id) jest powiązany z tabelą parcel\_lockers.

#### **parcel\_statuses ↔ parcels**

Relacja: Jeden status jest przypisany do jednej paczki.

Powiązanie: parcels.status\_id jest kluczem obcym odnoszącym się do parcel\_statuses.id.

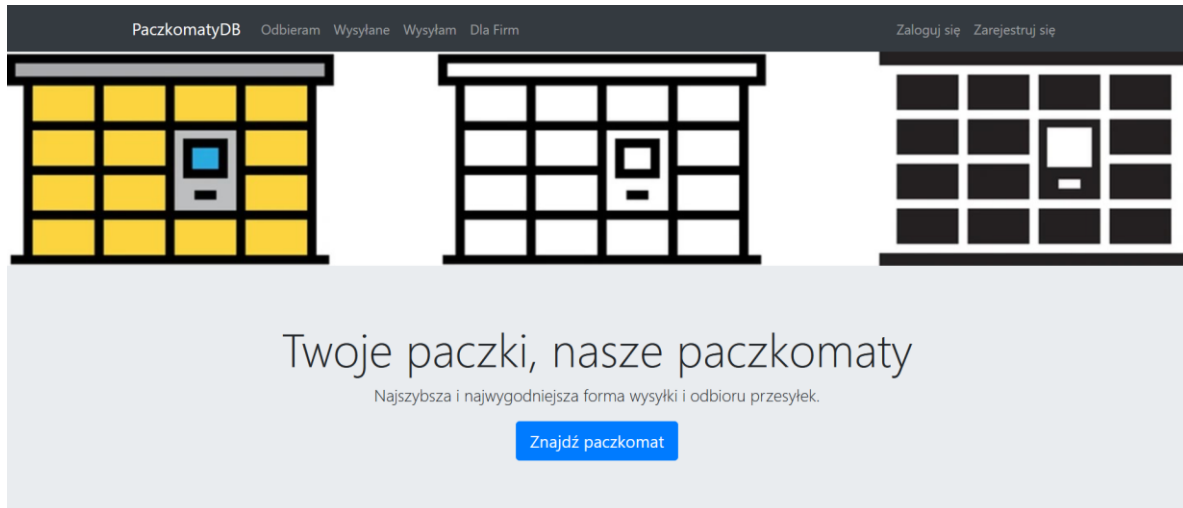
#### **parcel\_types ↔ parcels**

Relacja: Jeden typ paczki jest przypisany do jednej paczki.

Powiązanie: parcels.type\_id jest kluczem obcym odnoszącym się do parcel\_types.id.

## 4. GUI

### 4.1 Strona główna



**Opis:** Strona główna zapewnia przegląd funkcji aplikacji. Użytkownicy mogą nawigować do różnych sekcji, takich jak paczki odebrane, paczki wysłane, nadawanie paczki, usługi dla firm, logowanie oraz rejestracja poprzez pasek nawigacyjny u góry strony. Na środku strony znajduje się przycisk, który przenosi do listy paczkomatów.

### 4.2 Nadawanie paczki

### Nadaj paczkę

Numer odbiorcy:

Paczkomat początkowy:

Paczkomat końcowy:

Nazwa paczki:

Opis paczki:

Typ paczki:

W formularzu nadawania paczki znajdują się następujące pola:

- Numer Odbiorcy: Pole tekstowe, w którym użytkownik wprowadza numer telefonu odbiorcy paczki. Po tym numerze szukany jest odbiorca i przypisywany jako receiver.
- Paczkomat Początkowy: Lista rozwijana umożliwiające wybór paczkomatu, z którego paczka będzie wysłana.
- Paczkomat Końcowy: Lista rozwijana lub umożliwiające wybór paczkomatu, do którego paczka będzie dostarczona.
- Nazwa Paczki: Pole tekstowe, w którym użytkownik wprowadza nazwę paczki.
- Opis Paczki: Pole tekstowe, w którym użytkownik może wprowadzić dodatkowe informacje lub opis paczki.
- Typ Paczki: Lista rozwijana umożliwiające wybór typu paczki, na przykład mała, średnia, duża itp.

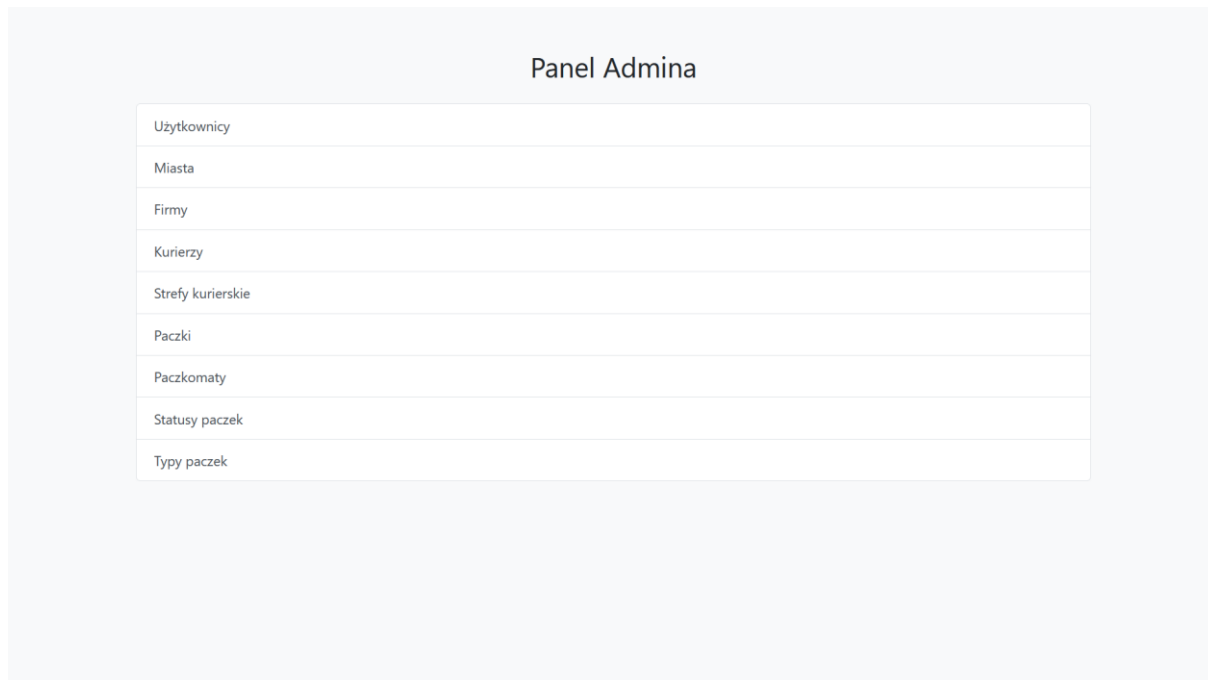
Formularz ten pozwala użytkownikowi wprowadzić niezbędne informacje dotyczące paczki, takie jak adresat, lokalizacja paczkomatu początkowego i końcowego, nazwa i opis paczki oraz jej typ. Dzięki temu użytkownik może w prosty sposób zlecić wystanie paczki z jednego paczkomatu do drugiego.

Po wypełnieniu formularza, aplikacja obsługuje następujące funkcjonalności:

- Obsługa błędu braku odbiorcy: Jeśli podany numer odbiorcy nie zostanie znaleziony w systemie, aplikacja wyświetla odpowiedni komunikat o błędzie, informujący użytkownika o konieczności podania poprawnego numeru odbiorcy.

- Komunikat potwierdzający wystanie paczki: Po udanym wystaniu paczki, aplikacja wyświetla komunikat potwierdzający, informujący użytkownika o poprawnym zleceniu wystania paczki.

### 4.3 Panel Admina



Panel Admina jest częścią aplikacji, przeznaczoną dla administratora systemu. Pozwala on na zarządzanie różnymi aspektami systemu, takimi jak użytkownicy, firmy, kurierzy, miasta, strefy kurierskie, paczki, paczkomaty, statusy paczek i typy paczek. Poniżej znajduje się opis dostępnych funkcjonalności w panelu admina:

- Użytkownicy: W tej sekcji administrator ma możliwość przeglądania, dodawania, edycji i usuwania użytkowników. Każdy użytkownik może mieć przypisaną rolę, uprawnienia i dane kontaktowe.
- Firmy: Administrator może zarządzać firmami, które są powiązane z użytkownikami. Może przeglądać, dodawać, edytować i usuwać informacje o firmach.
- Kurierzy: Sekcja ta umożliwia administratorowi zarządzanie danymi dotyczącymi kurierów. Może przeglądać, dodawać, edytować i usuwać informacje o kurierach, w tym strefy, którymi są odpowiedzialni.
- Miasta: Administrator może zarządzać miastami, w których działają paczkomaty i kurierzy. Może przeglądać, dodawać, edytować i usuwać informacje o miastach.
- Strefy Kurierskie: Sekcja ta umożliwia administratorowi zarządzanie strefami, które są przypisane do poszczególnych kurierów. Może przeglądać, dodawać, edytować i usuwać informacje o strefach kurierskich.

- Paczki: Administrator ma możliwość przeglądania wszystkich paczek w systemie. Może również dodawać, edytować i usuwać informacje o paczkach, w tym ich statusy i typy.
- Paczkomaty: Sekcja ta umożliwia administratorowi zarządzanie danymi dotyczącymi paczkomatów. Może przeglądać, dodawać, edytować i usuwać informacje o paczkomatach.
- Statusy Paczek: Administrator ma możliwość zarządzania statusami paczek. Może przeglądać, dodawać, edytować i usuwać różne statusy paczek.
- Typy Paczek: Sekcja ta umożliwia administratorowi zarządzanie typami paczek. Może przeglądać, dodawać, edytować i usuwać różne typy paczek.

## **5. Uruchomienie aplikacji**

### **5.1 Wymagane oprogramowanie**

- PHP w wersji 8.2.12
- Composer
- XAMPP Control Panel
- Visual Studio Code
- Node.js

### **5.2 Kroki konfiguracyjne**

- Rozpakowanie archiwum z projektem
- Uruchomienie XAMPP Control Panel i włączenie modułów „Apache” oraz „MySQL”
- Uruchomienie skryptu „start.bat” stawiającego aplikacje
- Otworzenie głównego katalogu projektu w Visual Studio Code
- Zaseedowanie bazy (opcjonalne). Kolejność seedowania:
  - UserSeeder
  - CitySeeder
  - ParcelStatusSeeder
  - ParcelTypeSeeder

-ParcelLockerSeeder

-CourierZoneSeeder

-CourierSeeder

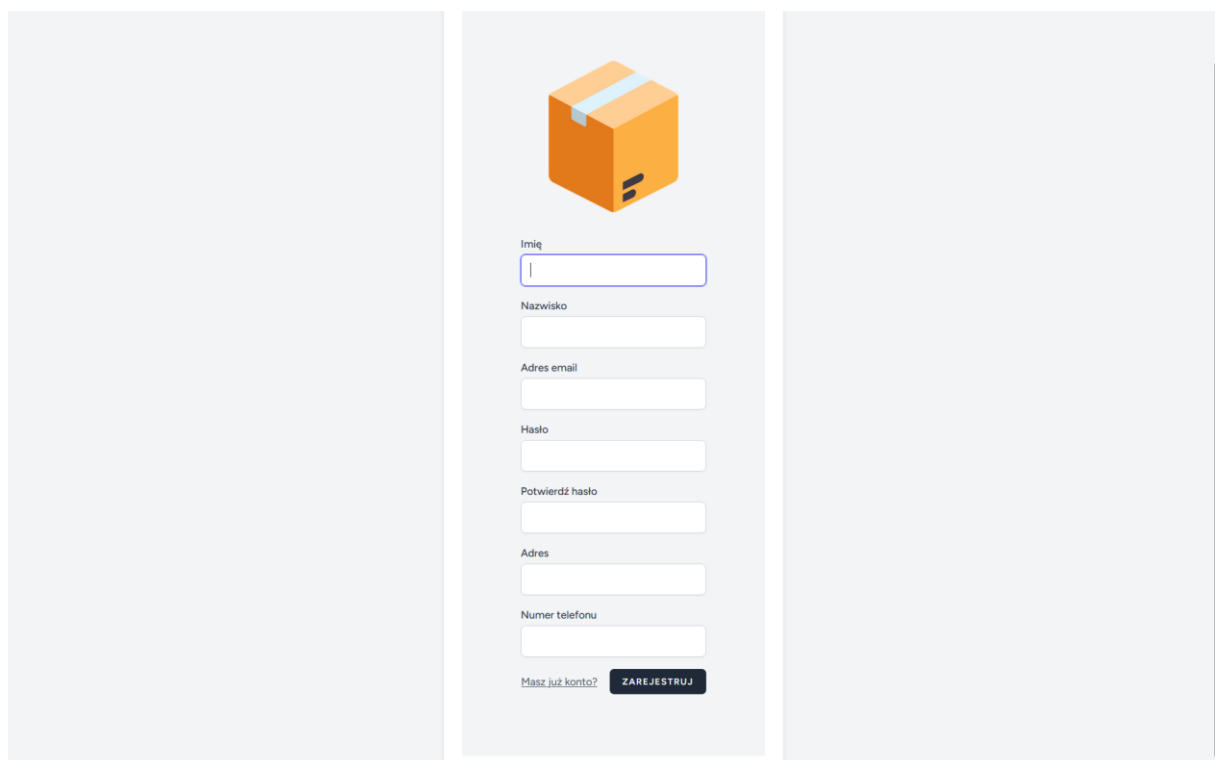
-ParcelsSeeder

-Uruchomienie serwera deweloperskiego z poziomu terminalu cmd przy pomocy komendy 'php artisan serve'

- Udanie się pod adres <http://localhost:8000> w przeglądarce

## 6. Funkcjonalności aplikacji

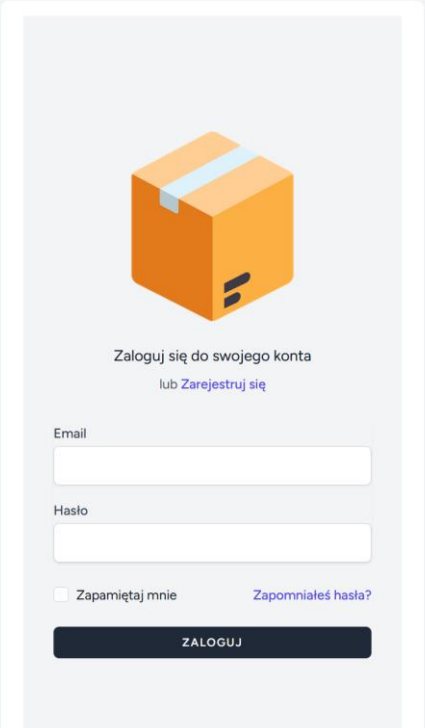
### 6.1 Rejestracja

The image shows a registration form for an application. At the top, there is an orange 3D box icon with a white label and a black barcode. Below the icon, the form consists of several input fields: 'Imię' (Name), 'Nazwisko' (Surname), 'Adres email' (Email address), 'Hasło' (Password), 'Potwierdź hasło' (Confirm password), 'Adres' (Address), and 'Numer telefonu' (Phone number). Each field has a corresponding label above it. At the bottom of the form, there is a link 'Masz już konto?' (Do you already have an account?) and a dark blue button labeled 'ZAREJESTRUJ' (REGISTER).

**Opis:** Użytkownik ma możliwość rejestracji do aplikacji przy pomocy danych takich jak Imię, Nazwisko, E-mail, Hasło, Numer Telefonu.

Aplikacja sprawdza po stronie frontendu czy wprowadzone dane są w poprawnym formacie.

## 6.2 Logowanie



Zaloguj się do swojego konta  
lub [Zarejestruj się](#)

Email

Hasło

☐ Zapamiętaj mnie [Zapomniałeś hasła?](#)

ZALOGUJ

**Opis:** Użytkownik ma możliwość logowania do aplikacji przy pomocy danych takich jak adres e-mail i hasło po wcześniejszym zarejestrowaniu się do aplikacji.

Przykładowe loginy:

E-mail	Hasło	Typ
admin@example.com	password	admin
user1@example.com	password	user
user2@example.com	password	user
courier@example.com	password	courier

## 6.3 Wysyłanie paczek, widoki paczek

PaczkomatyDB

Odbieram

Wysyłane

Wysyłam

Dla Firm

User1 ▾

Nadaj paczkę

Numer odbiorcy:

987654321

Paczkomat początkowy:

Paczkomat 1, ul. Kwiatowa 43, Białystok ▾

Paczkomat końcowy:

Paczkomat 2, ul. Kościelna 35, Białystok ▾

Nazwa paczki:

Dokumentacja- test

Opis paczki:

Prezentacja widoku do dokumentacji

Typ paczki:

Średnia paczka 20x40x80cm ▾

Nadaj paczkę

(nadawanie paczki)



## Moje paczki

Paczka 2

**Nadawca:** User2

**Paczkomat:** Paczkomat 2, ul. Kościelna 35, Białystok

**Status:** W przygotowaniu

**Czas wysłania:** 2024-05-31 10:23:23

**Kod odbioru:** 6659a51bea1b9

(paczki idące do użytkownika)

## Wysyłane paczki

Paczka 1

**Kod nadania:** 6659a51bea1b7

**Odbiorca:** User2

**Paczkomat początkowy:** Paczkomat 2, ul. Kościelna 35, Białystok

**Paczkomat docelowy:** Paczkomat 2, ul. Grunwaldzka 15, Gorzów Wielkopolski

**Opis:** Opis paczki 1

**Status:** W przygotowaniu

**Czas wysłania:** 2024-05-31 10:23:23

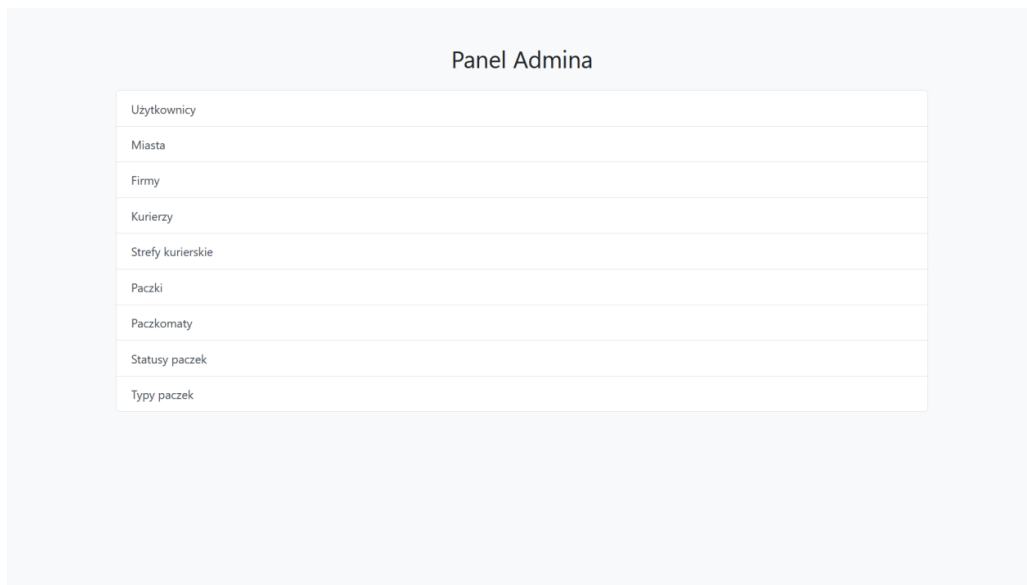
(paczki wysłane przez użytkownika)

Użytkownik po przejściu do zakładki „Wysyłam” ma możliwość wysłania paczki do innego użytkownika jeśli zna jego numer telefonu.

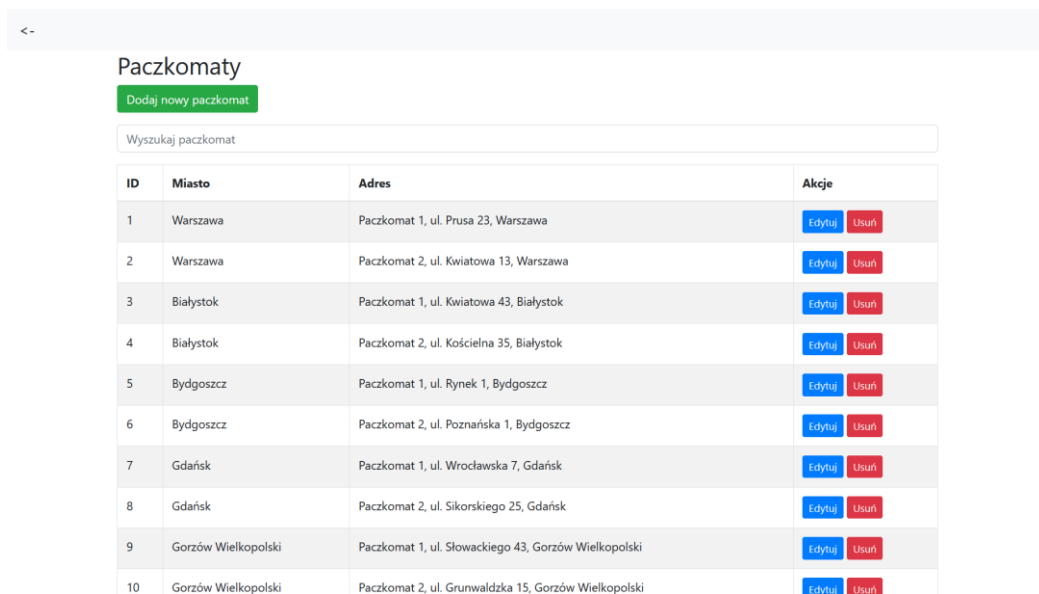
Po przejściu do zakładki „Odbieram” użytkownik ma możliwość zobaczenia paczek, które zostały do niego wysłane i ich dane takie jak nazwa paczki, imię nadawcy, paczkomat końcowy w którym będzie można odebrać paczkę, status paczki, czas wysłania oraz kod odbioru do paczkomatu.

Po przejściu do zakładki „Wysyłam” użytkownik ma możliwość zobaczenia paczke, które zostały przez niego wysłane i ich dane takie jak nazwa paczki, kod nadania, imię odbiorcy, paczkomat początkowy, paczkomat docelowy, opis paczki, status paczki oraz czas wysłania.

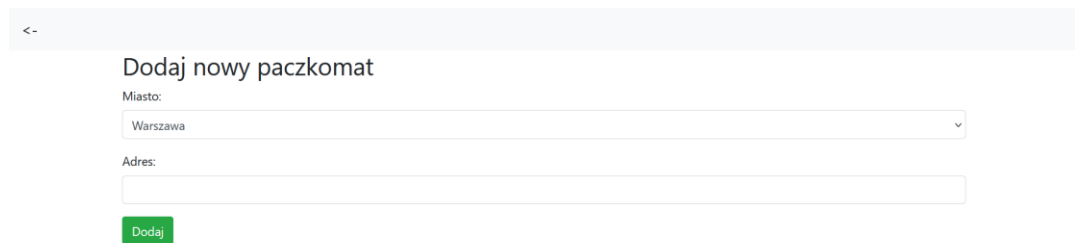
## 6.4 CRUD przeprowadzany przez administratora



(panel admina)



(widok paczkomatów)



<-

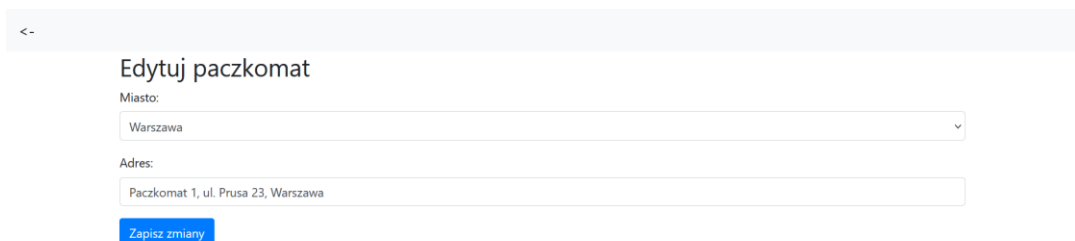
### Dodaj nowy paczkomat

Miasto:

Warszawa

Adres:

(dodawanie paczkomatów)



<-

### Edytuj paczkomat

Miasto:

Warszawa

Adres:

Paczkomat 1, ul. Prusa 23, Warszawa

(edycja wybranego paczkomatu)

Użytkownik o type „admin” ma możliwość przejścia pod adres <http://localhost:8000/admin> gdzie znajduje się panel admina. Z jego poziomu ma możliwość wyświetlenia, dodawania, edycji oraz usuwania danych z każdych tabel w bazie danych.

Jako przykładowe widoki wybrałem widok paczkomatów. Jak widać na zdjęciach administrator ma możliwość zarządzania wszystkimi danymi dotyczącymi paczkomatów. W przypadku innych widoków działa to na tej samej zasadzie.

Administrator ma dostęp do danych z tabel:

-user

- cities
- companies
- couriers
- courier\_zones
- parcels
- parcel\_lockers
- parcel\_statuses
- parcel\_types

## 6.5. Zarządzanie użytkownikami przez administratora

< -

Użytkownicy

Wyszukaj użytkownika

ID	Name	Surname	Email	Address	Phone Number	Company	Type	Actions
1	Admin	Administrator	admin@example.com	Admin Street 1	123456789	N/A	admin	<a href="#">Edit</a> <a href="#">Delete</a>
2	User1	Regular1	user1@example.com	User Street 1	987654321	N/A	user	<a href="#">Edit</a> <a href="#">Delete</a>
3	User2	Regular2	user2@example.com	User Street 2	987654322	N/A	user	<a href="#">Edit</a> <a href="#">Delete</a>
4	Courier	Fast	courier@example.com	Courier Street 1	123123123	N/A	courier	<a href="#">Edit</a> <a href="#">Delete</a>

(widok użytkowników)

< -

Edytuj użytkownika

Imię:

User2

Nazwisko:

Regular2

Email:

user2@example.com

Adres:

User Street 2

Numer telefonu:

987654322

Typ użytkownika:

Użytkownik

ID Firmy:

Zapisz zmiany

(widok edycji użytkownika)

Administrator po wejściu do widoku użytkowników ma możliwość ich przeglądania, edycji oraz usuwania.

W widoku edycji może zmieniać dane użytkownika takie jak imię, nazwisko, e-mail, adres, numer telefonu, typ użytkownika oraz id firmy jeśli takową posiada.

Gdy kurier zaczyna pracować dla tej firmy to administrator mu przyzna typ „courier”. Po tym musi zostać utworzona dla niego strefa kurierska i „konto” kurier w widoku Kurierzy które zostanie przypisane do tego użytkownika.

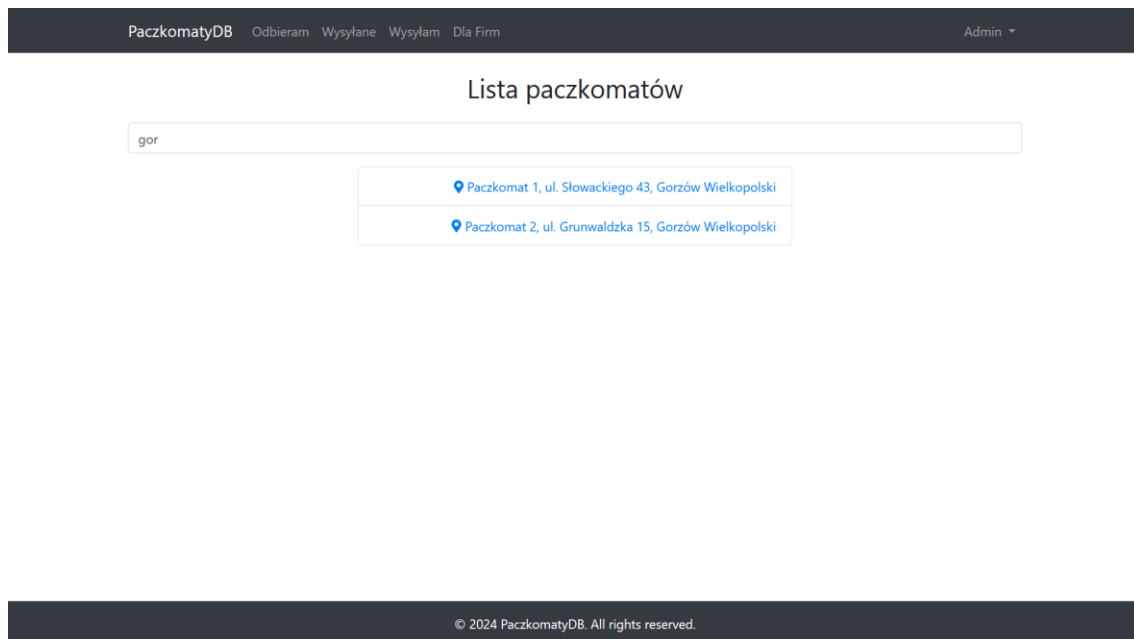
## 6.6. Przeglądanie ogólnodostępnych zasobów

Ogólnodostępnym zasobem w tej aplikacji są paczkomaty. Aby je przeglądać należy ze strony głównej przejść do widoku paczkomatów używając przycisku „Znajdź paczkomat”, który znajduje się na środku strony. Po tym zobaczymy poniższy widok:

PaczkomatyDB		Odbieram	Wysłane	Wysylam	Dla Firm	Admin ▾
Lista paczkomatów						
Wyszukaj paczkomat						
📍 Paczkomat 1, ul. Prusa 23, Warszawa						
📍 Paczkomat 2, ul. Kwiatowa 13, Warszawa						
📍 Paczkomat 1, ul. Kwiatowa 43, Białystok						
📍 Paczkomat 2, ul. Kościelna 35, Białystok						
📍 Paczkomat 1, Rynek 1, Bydgoszcz						
📍 Paczkomat 2, ul. Poznańska 1, Bydgoszcz						
📍 Paczkomat 1, ul. Wrocławska 7, Gdańsk						
📍 Paczkomat 2, ul. Sikorskiego 25, Gdańsk						
📍 Paczkomat 1, ul. Słowackiego 43, Gorzów Wielkopolski						
📍 Paczkomat 2, ul. Grunwaldzka 15, Gorzów Wielkopolski						
📍 Paczkomat 1, ul. Wrocławska 8, Katowice						
📍 Paczkomat 2, ul. Długa 47, Katowice						

Jak widać powyżej mamy listę wszystkich dostępnych paczkomatów. Zasoby te można filtrować poprzez użycie wyszukiwarki nad tabelą. Po wpisaniu dowolnego ciągu znaków lista zostanie odświeżona i wypisane zostaną tylko te paczkomaty, których adres zawiera wpisany ciąg znaków.

Przykład działania wyszukiwarki:



### Kod wyszukiwarki:

```
<script>
document.addEventListener('DOMContentLoaded', function() {
  var searchInput = document.getElementById('searchInput');
  var parcelLockerList = document.getElementById('parcelLockerList');
  var originalList = parcelLockerList.innerHTML.trim();

  searchInput.addEventListener('input', function() {
    var searchText = this.value.trim().toLowerCase();
    var filteredList = '';

    @foreach($parcel_lockers as $parcel_locker)
    if ("{{ $parcel_locker->address }}" .toLowerCase().includes(searchText)) {
      filteredList += '<li class="list-group-item d-flex justify-content-between align-items-center">
        <span><a href="{{ route('send') }}"></span>
        <span class="locker-address"><i class="fas fa-map-marker-alt"></i> {{ $parcel_locker->address }}</span>
      </li>';
    }
    @endforeach

    if (filteredList === '') {
      parcelLockerList.innerHTML = '<li class="list-group-item text-muted">Brak paczkomatów pasujących do wyszukiwania.</li>';
    } else {
      parcelLockerList.innerHTML = filteredList;
    }
  });

  searchInput.addEventListener('keyup', function(event) {
    if (event.key === 'Escape') {
      this.value = '';
      parcelLockerList.innerHTML = originalList;
    }
  });
});
</script>
```

Działa on na zasadzie opisanej powyżej. Po każdym inpcie lista jest odświeżana i wpisywane są tylko te paczkomaty, których adres pasuje do wprowadzonego tekstu.

## 6.7. Zarządzanie swoimi zasobami przez użytkownika aplikacji

Użytkownik ma możliwość nadawania paczek. Ta funkcjonalność została opisana wyżej w dokumentacji.

## 6.8. Zarządzanie swoimi danymi przez użytkownika aplikacji

Użytkownik po przejściu do swojego profilu ma możliwość:

- Zmiany swoich danych takich jak imię, nazwisko oraz e-mail
- Zaktualizowania hasła
- Dodania firmy jeśli jeszcze tego nie zrobił
- Usunięcia konta

### Informacje o profilu

Zaktualizuj informacje o profilu i adres e-mail.

Imię

Admin

Nazwisko

Administrator

E-mail

admin@example.com

ZAPISZ



### Zaktualizuj hasło

Upewnij się, że twoje hasło jest długie i składa się z co najmniej 8 znaków.

Obecne hasło

Nowe hasło

Potwierdź nowe hasło

ZAPISZ

---

### Informacje o firmie

PaczkomatyDB

10-102 Przykładow 12

NIP: 123443287

### Usuń konto

Po usunięciu konta wszystkie dane z nim związane zostaną usunięte i nie będzie możliwości ich przywrócenia.

USUŃ KONTO

## 6.9 Logika biznesowa

Poniżej przedstawiam ogólny przykład opisu funkcjonalności "Tworzenie nowej paczki" w aplikacji obsługującej sieć paczkomatów:

Wybór opcji wysłania paczki: Użytkownik wchodzi na stronę aplikacji i wybiera opcję "Wyślij paczkę" z panelu nawigacyjnego.

- Formularz nadawania paczki: Użytkownik jest przenoszony do formularza, gdzie może wprowadzić szczegóły dotyczące paczki, takie jak numer telefonu odbiorcy, nazwa, opis, oraz wybór paczkomatu początkowego i końcowego.
- Walidacja danych: Wprowadzone przez użytkownika dane są walidowane po stronie klienta za pomocą html oraz po stronie serwera w kontrolerze aplikacji.
- Przetwarzanie danych: Po poprawnej walidacji, dane są przesyłane do kontrolera aplikacji w celu utworzenia nowego rekordu paczki w bazie danych.
- Odbiorca: Odbiorca paczki jest automatycznie wyszukiwany na podstawie numeru telefonu. W razie gdyby nie został on znaleziony, użytkownik otrzyma stosowny komunikat.
- Generowanie unikalnego kodu: System generuje unikalny kod dla paczki, który będzie używany do identyfikacji i śledzenia paczki w systemie.
- Przypisanie kuriera: System automatycznie przypisuje kuriera do paczki.
- Aktualizacja stanu paczkomatów: Po zatwierdzeniu paczki, system aktualizuje dostępność paczkomatów początkowego i końcowego, zaznaczając, że paczka została umieszczona w paczkomacie początkowym i oczekuje na odbiór w paczkomacie końcowym.
- Powiadomienie użytkownika: Użytkownik otrzymuje powiadomienie o pomyślnym nadaniu paczki, zawierające unikalny kod śledzenia oraz informacje o kurierze i paczkomatach.

## 6.10. Walidacja danych

W aplikacji walidacja danych odbywa się zarówno po stronie frontendu jak i backendu

Przykład walidacji frontendowej:

```
<div class="form-group">
  <label for="recipient_number">Numer odbiorcy:</label>
  <input type="text" name="recipient_number" id="recipient_number" class="form-control" required required pattern="[0-9]{9}" title="Numer" />
</div>
```

Jest to walidacja numeru telefonu odbiorcy w formularzu nadania paczki.

Przykład walidacji backendowej:

```

public function store(Request $request)
{
    $request->validate([
        'sender_id' => 'required|exists:users,id',
        'receiver_id' => 'required|exists:users,id',
        'courier_id' => 'required|exists:couriers,id',
        'start_locker_id' => 'required|exists:parcel_lockers,id',
        'end_locker_id' => 'required|exists:parcel_lockers,id',
        'name' => 'required|string|max:255',
        'code' => 'required|string|max:255',
        'description' => 'required|string|max:255',
        'status_id' => 'required|exists:parcel_statuses,id',
        'type_id' => 'required|exists:parcel_types,id',
    ]);

    Parcels::create($request->all());
    return redirect()->route('admin.packages')->with('success', 'Paczka została dodana.');
```

Jest to walidacja w kontrolerze paczki. Sprawdza ona typ danych przy tworzeniu nowej paczki.

W pozostałych przypadkach walidacja wygląda podobnie.

## 7. Responsywność

Aplikacja została przystosowana do wszelkiego rodzaju urządzeń, od desktopów poprzez telefony aż do tabletów. Responsywny wygląd sprawia, że nie dochodzi do sytuacji w której elementy na ekranie byłyby źle rozmieszczone przez źle dobraną rozdzielczość.

### 7.1 Użycie kart

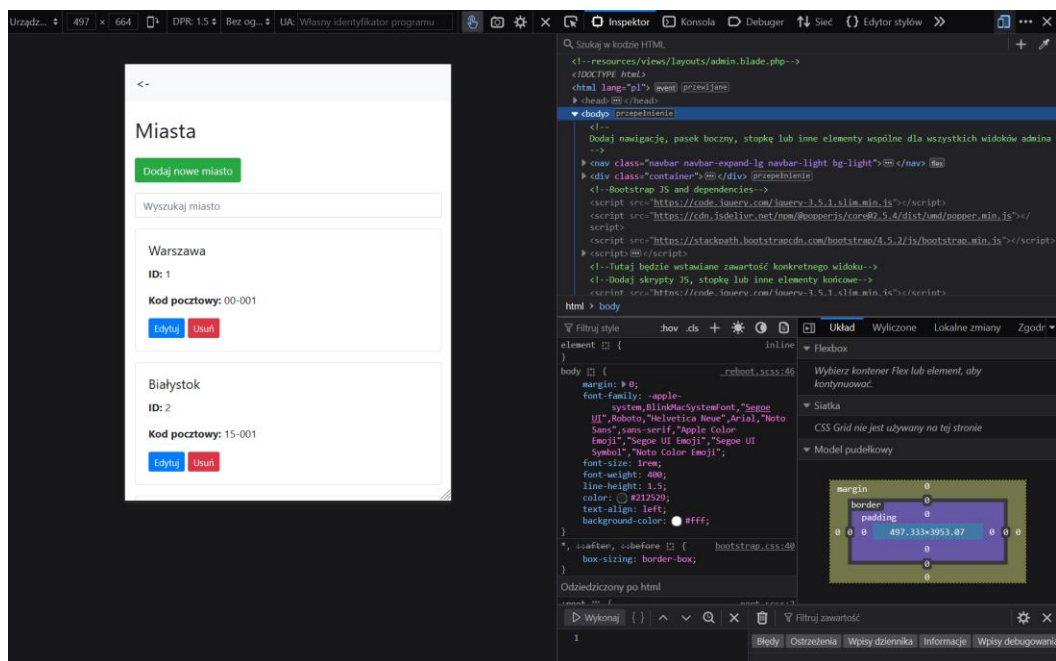
<-

Miasta

Dodaj nowe miasto

Wyszukaj miasto

ID	Nazwa	Kod pocztowy	Akcje
1	Warszawa	00-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
2	Białystok	15-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
3	Bydgoszcz	85-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
4	Gdańsk	80-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
5	Gorzów Wielkopolski	65-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
6	Katowice	40-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
7	Kielce	25-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
8	Kraków	30-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
9	Lublin	20-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
10	Łódź	90-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
11	Olsztyn	10-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
12	Opole	45-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>
13	Poznań	60-001	<a href="#">Edytuj</a> <a href="#">Usuń</a>



Na powyższych zrzutach ekranu przedstawiłem jak został rozwiązany problem responsywności tabel. Na mniejszych ekranach tabele nie są wyświetlane, zamiast nich użyto kart z frameworku bootstrap. Zapewniają one schludny interfejs co przekłada się na wygodę użytkownika.