

AI1	Dokumentacja projektu
<b>Autor</b>	Adrian Kowalczuk, 124017
<b>Kierunek, rok</b>	Informatyka i Ekonometria, II rok, st. stacjonarne (3,5-I)
<b>Temat projektu</b>	<i>Zarządzanie klubami wędkarskimi</i>

## Spis treści

Narzędzia i technologie.....	3
Baza danych.....	4
GUI .....	5
Uruchamianie aplikacji.....	8
Funkcjonalność aplikacji.....	9
CRUD administratora .....	12
Walidacja.....	13

## Wstęp

Projekt dotyczy aplikacji webowej, w którym możemy zarządzać klubem wędkarskim. Zamysłem całego projektu jest możliwość tworzenia nowych klubów wędkarskich oraz łatwe przypisywanie członków do odpowiednich klubów

### 1. Zarządzanie klubami wędkarskimi:

- a. Dodawanie członków: w aplikacji posiadamy możliwość tworzenia nowych klubów wędkarskich. Tworzenie dotyczy ustalania nazwy danego klubu, adresu oraz jest możliwość opisanie danego klubu
- b. Edytowanie istniejących klubów: w aplikacji posiadamy możliwość edytacji danego klubu w przypadku wprowadzenia błędnych danych przy jego tworzeniu. Mamy możliwość zmieniania nazwy danego klubu, adresu jak i zarówno opisu szczegółowego
- c. Usuwanie klubu : Możliwość usunięcia klubów, które już nie funkcjonują lub są nieaktualne.

### 2. Zarządzanie członkami klubów

- a. Dodawanie nowych członków: w aplikacji posiadamy możliwość dodawania nowych użytkowników. Mamy możliwość wprowadzenia Imienia i Nazwiska danego członka. W momencie dodawania nowego członka posiadamy również możliwość wprowadzenia podstawowych danych takich jak numer telefonu, adres email oraz przynależności do danego klubu.
- b. Edytowanie istniejących członków klubów: Możliwość aktualizowania danych członków, takich jak zmiana adresu, numeru telefonu, czy aktualizacja informacji o członkostwie.
- c. Usuwanie członków klubu : Możliwość usunięcia członków z bazy danych klubu, na przykład w przypadku rezygnacji z członkostwa

## Narzędzia i technologie

Języki programowania:

- PHP 8.2.12 - Licencja: PHP jest licencjonowany na podstawie PHP License v3.01, która jest zgodna z Open Source Initiative (OSI).
- JavaScript - Licencja: JavaScript jako język jest standardem ECMA-262. Większość bibliotek i narzędzi JS jest licencjonowana na zasadach open source, najczęściej pod MIT, Apache, lub GPL.

Framework:

- Laravel 11.4 - Licencja: Laravel jest licencjonowany na podstawie MIT License.
- Bootstrap: 5.3 - Licencja: Bootstrap jest licencjonowany na podstawie MIT License.

Bazy danych:

- MySQL 10.4 - Licencja: MySQL jest licencjonowany na podstawie GNU General Public License (GPL)

oraz oferowany na zasadach komercyjnych przez Oracle.

Środowisko programistyczne:

- Visual Studio Code - Licencja: Visual Studio Code jest licencjonowany na podstawie Microsoft Software License Terms.

Inne narzędzia:

- XAMPP 8.2.12 - Licencja: XAMPP jest licencjonowany na podstawie GNU General Public License (GPL).
- Composer – Licencja: Composer jest licencjonowany na podstawie MIT License.

Linki do pobrania niezbędnych programów:

<https://code.visualstudio.com/Download>

<https://www.php.net/downloads.php>

<https://www.apachefriends.org/pl/download.html>

<https://getcomposer.org/download/>

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

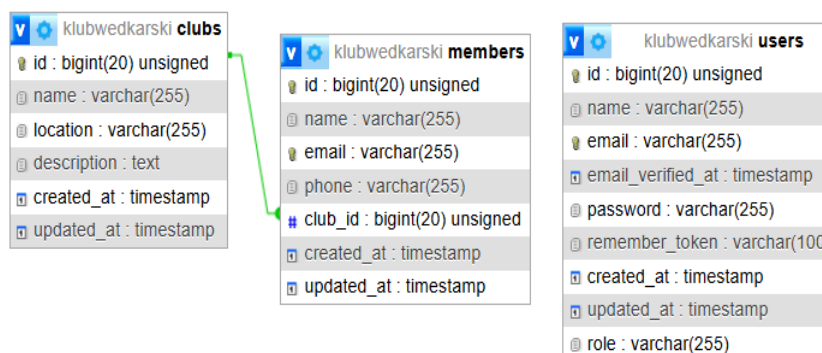
<https://getbootstrap.com/docs/4.0/getting-started/download/>

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

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

## Baza danych

Schemat ERD



Zdjęcie 1

Tabela 'clubs':

- id (INT, Primary Key, Auto Increment): Unikalny identyfikator klubu.
- name (VARCHAR, NOT NULL): Nazwa klubu wędkarskiego.
- location (VARCHAR, NOT NULL): Lokalizacja klubu, może zawierać adres, miasto lub region.

-description (TEXT, Optional): Opis klubu, zawierający dodatkowe informacje, takie jak historia, cele czy zasady działania.

Tabela 'members':

- id (INT, Primary Key, Auto Increment): Unikalny identyfikator członka klubu.
- name (VARCHAR, NOT NULL): Nazwa członka klubu.
- email (VARCHAR, NOT NULL, UNIQUE): Adres email członka klubu, używany do kontaktu
- phone (VARCHAR, Optional): Numer telefonu członka klubu.
- club\_id (INT, Foreign Key): Identyfikator klubu, do którego należy członek, powiązany z tabelą 'clubs'.

Tabela 'users':

- id (INT, Primary Key, Auto Increment): Unikalny identyfikator użytkownika.
- username (VARCHAR, NOT NULL, UNIQUE): Nazwa użytkownika używana do logowania.
- email (VARCHAR, NOT NULL, UNIQUE): Adres email użytkownika.
- password (VARCHAR, NOT NULL): Hasło użytkownika, przechowywane w formie zaszyfrowanej.

Opis powiązań bazy danych

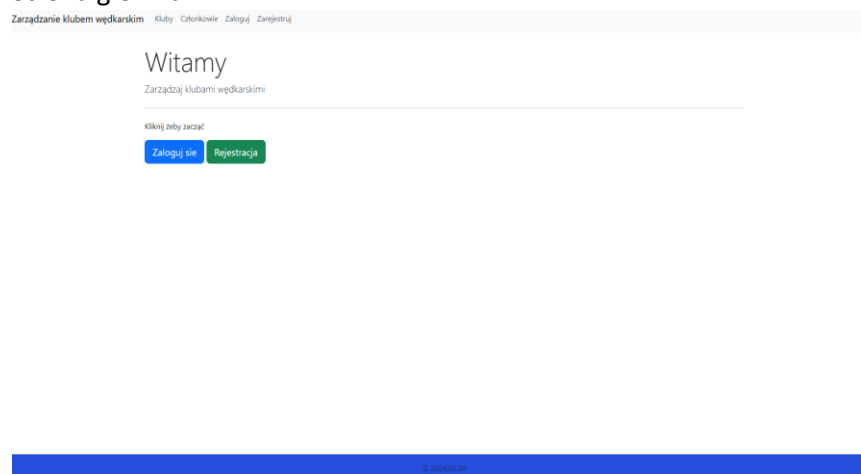
Powiązanie między tabelami 'clubs' i 'members':

Tabela 'members' posiada kolumnę club\_id, która jest kluczem obcym (Foreign Key) odnoszącym się do kolumny id w tabeli 'clubs'.

Dzięki temu powiązaniu możemy przypisać członków do konkretnych klubów wędkarskich. Każdy członek jest związany z jednym klubem, co umożliwia zarządzanie członkostwem w różnych klubach.

## GUI

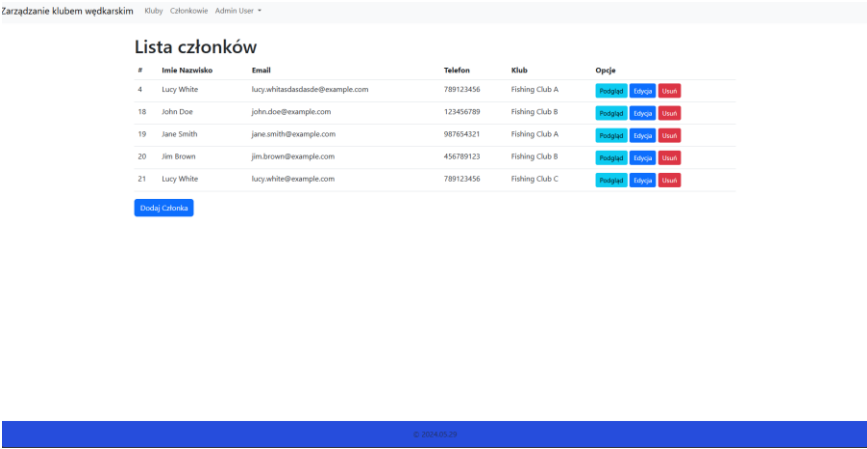
### a. Strona główna



Zdjęcie 2

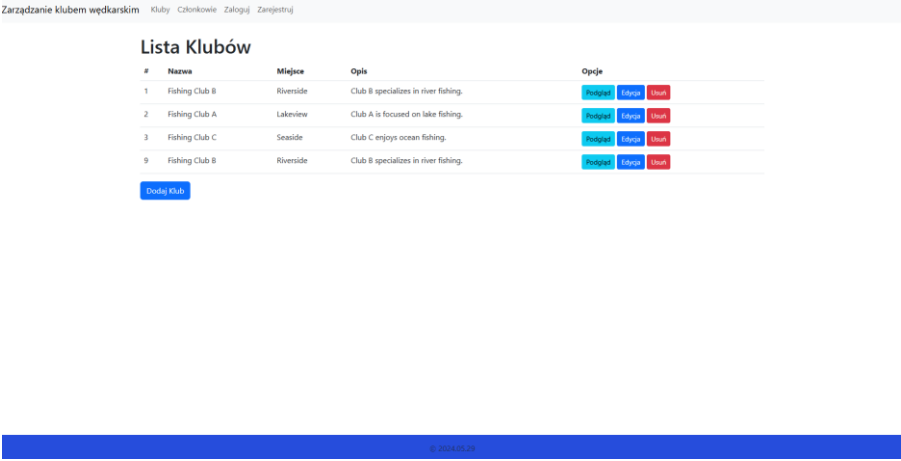
Na stronie głównej wyświetlają nam się w header zakładki do których możemy przejść oraz wyświetla nam się wiadomość powitalna, pod którą znajdują się przyciski do zalogowania lub rejestracji

b. Widok Klubów oraz Listy członków



Zdjęcie 3

Po kliknięciu w zakładkę w opcje Członkowie wyświetla nam się lista członków jacy należą do danych klubów oraz podstawowe informacje o nich

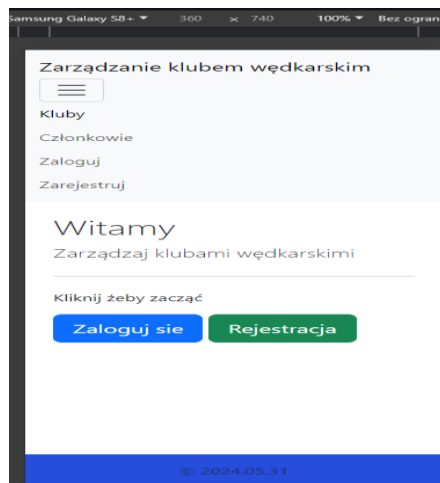


Zdjęcie 4

Po przejściu do zakładki Klubów ukaże nam się Lista Klubów jakie są dostępne oraz podstawowe informacje o nich takie jak nazwa miejsce oraz opis danego klubu

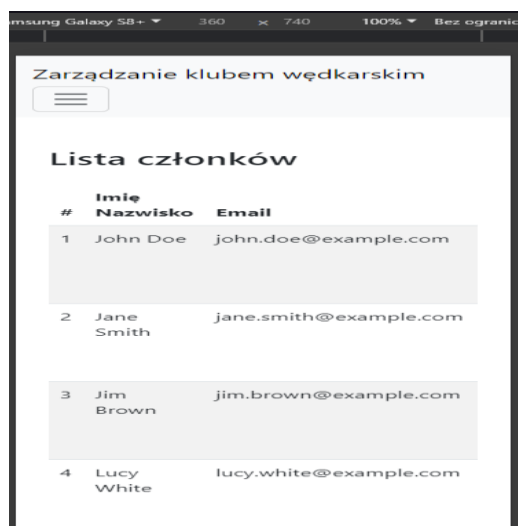
GUI RESPANSYWNE

Responsywność w aplikacji webowej odnosi się do zdolności strony do dostosowywania się do różnych rozmiarów ekranów i urządzeń. Bootstrap to jeden z najpopularniejszych frameworków CSS, który ułatwia tworzenie responsywnych stron internetowych. Bootstrap oferuje zestaw klas, które pomagają w tworzeniu responsywnych układów. Responsywność została zaimplementowana do wszystkich możliwych opcji dodawania/edytowania/usuwania/podglądu.



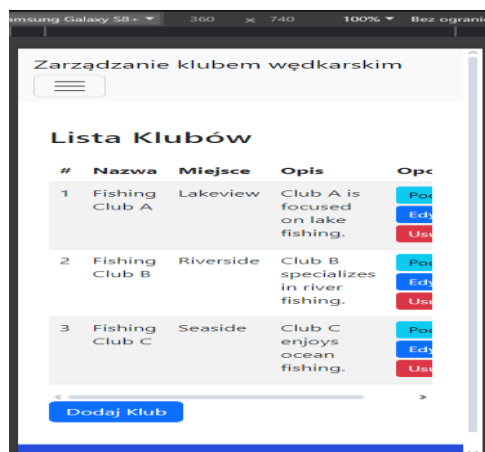
Zdjęcie 5

Na podanym zdjęciu widzimy stronę startową która wyświetli nam się po uruchomieniu aplikacji na ekranie telefonu Samsung Galaxy S8+.



Zdjęcie 6

Na powyższym zdjęciu po przejściu w opcję 'Członkowie' wyświetli nam się lista członków na ekranie telefonu Samsung Galaxy S8+.



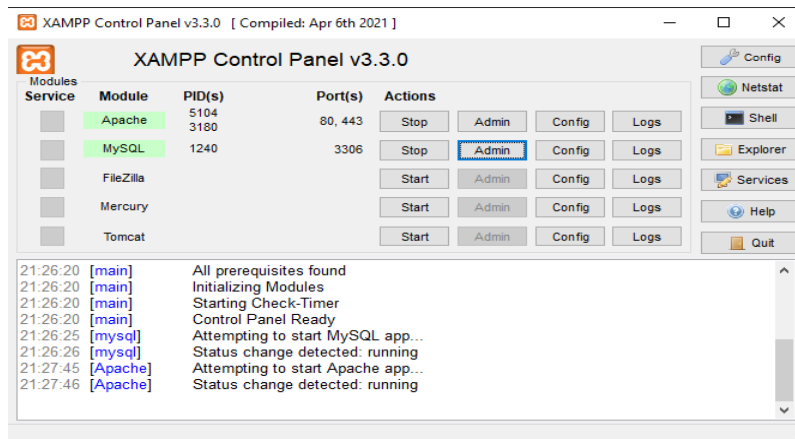
Zdjęcie 7

Na powyższym zdjęciu po przejściu w opcję 'Kluby ' wyświetli nam się lista klubów na ekranie telefonu Samsung Galaxy S8+

## Uruchamianie aplikacji

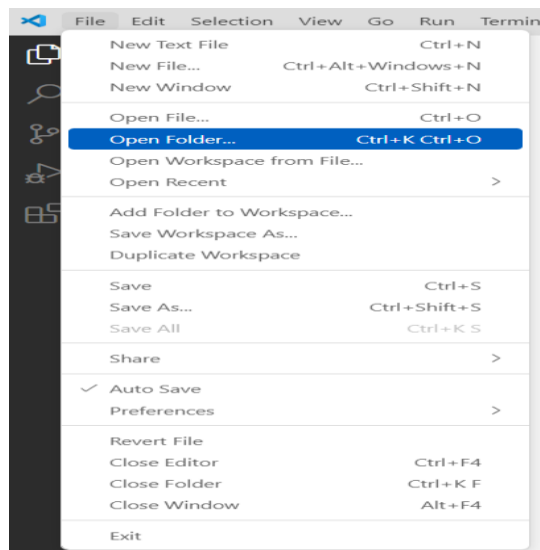
- c. Możemy uruchomić aplikację na dwa sposoby: możemy to zrobić postępując z krokami opisanymi poniżej, lub uruchomić XAMPP'a (włączyć Apache i MySQL) i uruchomić plik start.bat znajdujący się w folderze projektu.

- i. Uruchamiamy XAMPP, włączamy usługi Apache oraz Mysql



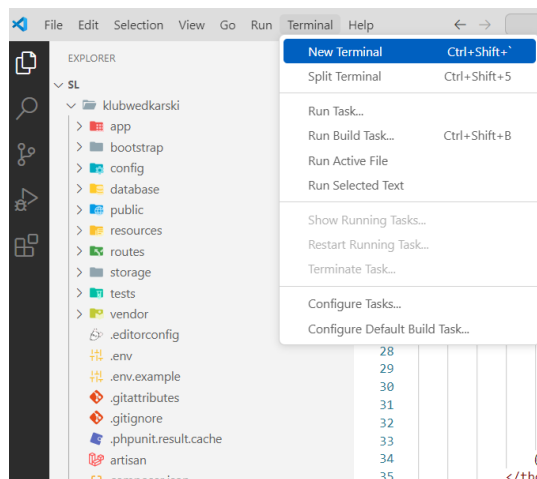
Zdjęcie 8

- ii. Uruchamiamy program Visual Studio Code, w lewym górnym rogu przechodzimy do zakładki File, wybieramy Open Folder i zaznaczamy nasz projekt.



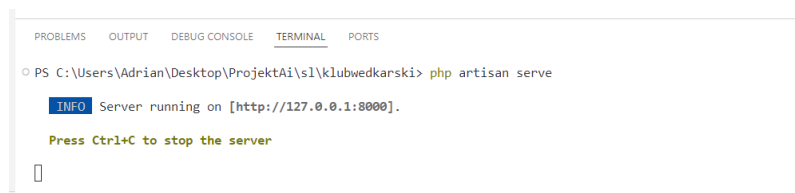
Zdjęcie 9





Zdjęcie 10

- iii. Po uruchomieniu terminala wpisujemy komendę `php artisan serve`



Zdjęcie 11

- iv. Uruchamiamy nowy terminal i wpisujemy komendę `php artisan migrate | php artisan db:seed`

- v. Przechodzimy pod adres wyświetlony w pierwszym terminalu

PS C:\Users\Adrian\Desktop\ProjektAi\s1\klubwedkarski> `php artisan serve`

**INFO** Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

Zdjęcie 12

## Funkcjonalność aplikacji

- d. Logowanie

W aplikacji przechodząc w zakładkę `zaloguj` możemy się zalogować do strony co będzie umożliwiało nam edytowanie, dodawanie oraz usuwanie członków oraz klubów

# Witamy

Zarządzaj klubami wędkarskimi

Kliknij żeby zacząć

[Zaloguj się](#)[Rejestracja](#)

Zdjęcie 13

Zaloguj się

E-Mail: admin@example.com

Hasło: Admin123

☐ Zapamiętaj mnie

[Zaloguj](#)

Zdjęcie 14

Po zalogowaniu dostajemy możliwość operacji na stronie. Np. klikając w zakładkę Klubu -> („przycisk”) Dodaj Klub pojawi nam się następujące okno

Zarządzanie klubem wędkarskim Kluby Członkowie Admin User

### Dodaj nowy klub

Nazwa:

Miejsce:

Opis:

[Utwórz Klub](#)

Zdjęcie 15

## e. Rejestracja

Mamy możliwość również rejestracji na stronie pod zakładką Zarejestruj. Umożliwia nam to zarejestrowanie nowych użytkowników w aplikacji internetowej

Zarządzanie klubem wędkarskim Kluby Członkowie Zaloguj Zarejestruj

### Rejestracja

Nazwa:

E-Mail:

Hasło:

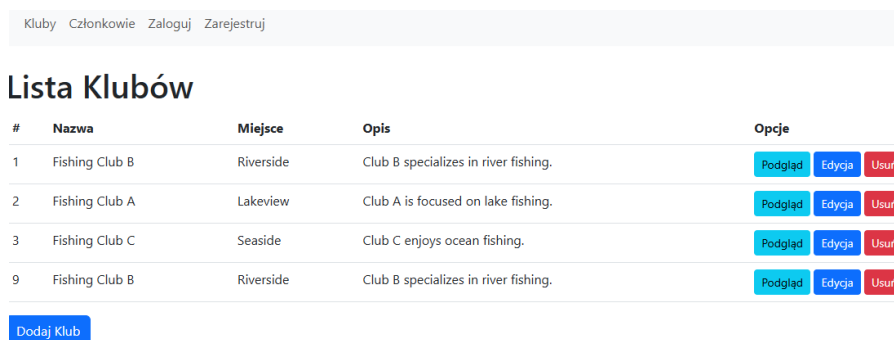
Potwierdź hasło:

[Zarejestruj](#)

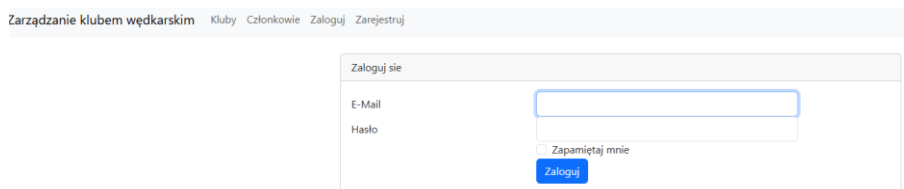
Zdjęcie 16

f. Użytkownicy niezalogowani

Użytkownicy niezalogowani nie będą mieli dostępu do możliwości edytowania usuwania oraz dodawania nowych członków/klubów. Automatycznie będą przekierowywani do okienka logowania. Mogą jedynie przeglądać zasoby strony jak na poniższych zdjęciach



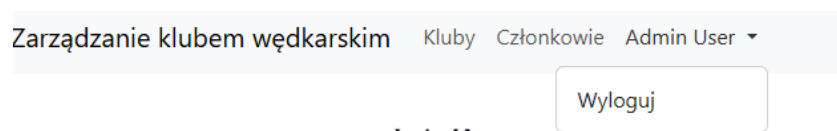
Zdjęcie 17



Zdjęcie 18

g. Wylogowywanie

Mamy również możliwość wylogowywania się ze strony po rozwinięciu w zakładce „Nazwa\_Użytkownika”. Następnie wciskając przycisk „Wyloguj” zostaniemy wylogowani.



Zdjęcie 19

# CRUD administratora

## h. Dodawanie

### Dodaj nowy klub

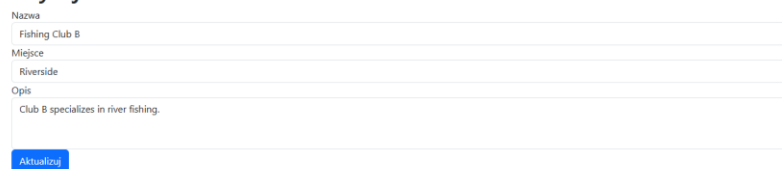


Zdjęcie 20

Po wciśnięciu przycisku "Dodaj Klub" wyświetli się formularz umożliwiający dodanie nowego klubu wędkarskiego. Formularz ten pozwala użytkownikowi wprowadzić niezbędne informacje dotyczące nowego klubu, takie jak nazwa klubu, miejsce oraz opis klubu.

## i. Edytowanie

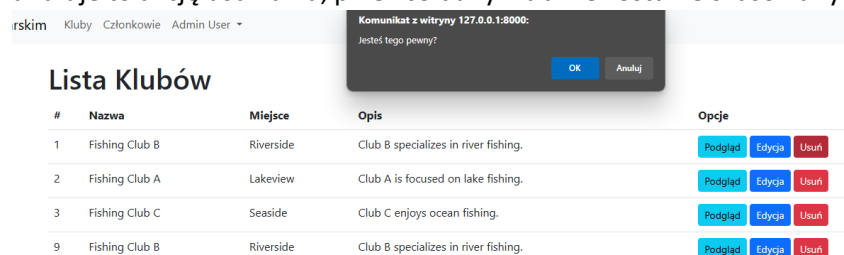
### Edytuj Klub



Zdjęcie 21

Po wciśnięciu przycisku „Edytuj” użytkownikowi wyświetli się formularz edytowania informacji o wybranym klubie wędkarskim. Formularz ten pozwala na aktualizację istniejących danych klubu, takich jak nazwa, miejsce oraz opis

- j. Usuwanie- po kliknięciu przycisku usuń wyświetli nam się komunikat który musimy potwierdzić aby usunąć dany klub/członka. W kodzie zawarty został do przycisku atrybut który zawiera kod JavaScript. Wywołuje on wbudowaną funkcję JavaScript ‘confirm’ która wyświetla okno dialogowe z wiadomością „Jesteś tego pewny” oraz z dwoma przyciskami „OK” , „Anuluj”. W przypadku kliknięcia w przycisk „Anuluj” anuluje to akcję usuwania, przez co dany klub nie zostanie skasowany



#	Nazwa	Miejsce	Opis	Opcje
1	Fishing Club B	Riverside	Club B specializes in river fishing.	<a href="#">Podgląd</a> <a href="#">Edycja</a> <a href="#">Usuń</a>
2	Fishing Club A	Lakeview	Club A is focused on lake fishing.	<a href="#">Podgląd</a> <a href="#">Edycja</a> <a href="#">Usuń</a>
3	Fishing Club C	Seaside	Club C enjoys ocean fishing.	<a href="#">Podgląd</a> <a href="#">Edycja</a> <a href="#">Usuń</a>
9	Fishing Club B	Riverside	Club B specializes in river fishing.	<a href="#">Podgląd</a> <a href="#">Edycja</a> <a href="#">Usuń</a>

Zdjęcie 22

Wszystkie powyższe akcje zawarte w podpukach h, i, j działają analogicznie do możliwości dodawania/ edytacji/ usuwania członków dostępnych pod zakładką członkowie.

# Walidacja

## i. Rejestracja

```
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Models\User;
6  use App\Http\Controllers\Controller;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Hash;
9  use Illuminate\Support\Facades\Auth;
10
11 class RegisterController extends Controller
12 {
13     public function showRegistrationForm()
14     {
15         return view('auth.register');
16     }
17
18     public function register(Request $request)
19     {
20         $request->validate([
21             'name' => 'required|string|max:255',
22             'email' => 'required|string|email|max:255|unique:users',
23             'password' => 'required|string|min:8|confirmed',
24         ]);
25
26         $user = User::create([
27             'name' => $request->name,
28             'email' => $request->email,
29             'password' => Hash::make($request->password),
30         ]);
31
32         Auth::login($user);
33
34         return redirect()->route('dashboard');
35     }
36 }
```

Zdjęcie 23

name: Pole name jest wymagane (required), musi być ciągiem znaków (string) oraz mieć maksymalnie 255 znaków (max:255).

email: Pole email jest wymagane, musi być ciągiem znaków, być poprawnym adresem email, mieć maksymalnie 255 znaków i być unikalne w tabeli users.

password: Pole password jest wymagane, musi być ciągiem znaków, mieć co najmniej 8 znaków (min:8) i musi być potwierdzone przez pole password\_confirmation, które musi być obecne w formularzu.

## ii. Club controller

```
public function store(Request $request)
{
    $request->validate([
        'name' => 'required',
        'location' => 'required',
    ]);

    Club::create($request->all());

    return redirect()->route('clubs.index')->with('success', 'Club created successfully.');
```

```
public function show(Club $club)
{
    return view('clubs.show', compact('club'));
}
```

```
public function edit(Club $club)
{
    return view('clubs.edit', compact('club'));
}
```

```
public function update(Request $request, Club $club)
{
    $request->validate([
        'name' => 'required',
        'location' => 'required',
    ]);

    $club->update($request->all());

    return redirect()->route('clubs.index')->with('success', 'Club updated successfully.');
```

Zdjęcie 24

### Metoda store():

Przed zapisaniem nowego klubu do bazy danych, wywoływana jest metoda `validate()` na obiekcie `$request`. W tej metodzie definiowane są reguły walidacji dla pól `name` i `location`. Jeśli walidacja nie powiedzie się, użytkownik zostanie przekierowany z powrotem do formularza tworzenia klubu (`clubs.create`) z odpowiednimi komunikatami błędów. Jeśli dane zostaną pomyślnie zwalidowane, klub zostanie utworzony i użytkownik zostanie przekierowany na stronę wyświetlającą wszystkie kluby (`clubs.index`) z komunikatem o sukcesie.

### Metoda update():

Podobnie jak w metodzie `store()`, najpierw następuje walidacja danych wejściowych za pomocą metody `validate()` na obiekcie `$request`. Jeśli walidacja nie powiedzie się, użytkownik zostanie przekierowany z powrotem do formularza edycji klubu (`clubs.edit`) z odpowiednimi komunikatami błędów. Jeśli dane zostaną pomyślnie zwalidowane, klub zostanie zaktualizowany, a użytkownik zostanie przekierowany na stronę wyświetlającą wszystkie kluby (`clubs.index`) z komunikatem o sukcesie.

### iii. Member controller

```
public function update(Request $request, Member $member)
{
    logger("Update");
    $request->validate([
        'name' => 'required',
        'email' => 'required|email',
        'phone' => 'nullable',
        'club_id' => 'required|exists:clubs,id',
    ]);

    $member->update($request->all());

    return redirect()->route('members.index')->with('success', 'Member updated successfully.');
```

Zdjęcie 24

```
public function store(Request $request)
{
    logger("store");
    $request->validate([
        'name' => 'required',
        'email' => 'required|email',
        'phone' => 'nullable',
        'club_id' => 'required|exists:clubs,id',
    ]);

    Member::create($request->all());

    return redirect()->route('members.index')->with('success', 'Member created successfully.');
```

Zdjęcie 25

Metoda store(Request \$request):

Ta metoda zajmuje się tworzeniem nowego rekordu członka (Member). Po pierwsze, jest zapisywany do dziennika komunikat store, co może być przydatne do śledzenia działania kodu. Następnie dane przesyłane w zapytaniu są walidowane za pomocą metody validate() na obiekcie \$request. Określa się, że pole name jest wymagane, email musi być poprawnym adresem email, phone jest opcjonalne (może być null), a club\_id musi istnieć w tabeli klubów (tj. musi istnieć odpowiedni rekord w tabeli clubs o podanym id). Jeśli walidacja przebiegnie pomyślnie, tworzony jest nowy obiekt klasy Member za pomocą metody create() z danymi z żądania (\$request->all()), a następnie użytkownik zostaje przekierowany na stronę indeksu członków z wiadomością o sukcesie.

Metoda update(Request \$request, Member \$member):

Jest to metoda odpowiedzialna za aktualizowanie istniejącego rekordu członka. Podobnie jak w metodzie store(), jest zapisywany komunikat Update do dziennika. Następnie dane przesyłane w zapytaniu są walidowane za pomocą metody validate(). Walidacja przebiega tak samo jak w przypadku metody store(). Jeśli walidacja się powiedzie, wywoływana jest metoda update() na obiekcie \$member, który został przekazany jako argument metody. Dane aktualizacji pochodzą z \$request->all(), a następnie użytkownik zostaje przekierowany na stronę indeksu członków z wiadomością o sukcesie.