

# **System Zarządzania Stroną Siłowni**

Systemy Zarządzania Treścią

**Damian Ćwikliński 145387**

Politechnika Poznańska

prowadzący dr inż. Michał Apolinarski

# Spis treści

<b>1</b>	<b>Charakterystyka ogólna projektu</b>	<b>2</b>
<b>2</b>	<b>Wymagania</b>	<b>3</b>
2.1	Funkcjonalne . . . . .	3
2.1.1	Redaktor . . . . .	3
2.1.2	Administrator . . . . .	3
2.1.3	Czytelnik . . . . .	3
2.2	Niefunkcjonalne . . . . .	4
<b>3</b>	<b>Architektura systemu</b>	<b>5</b>
3.1	Narzędzia . . . . .	5
3.2	Backend aplikacji . . . . .	5
3.3	Frontend aplikacji . . . . .	5
<b>4</b>	<b>Schemat baz danych</b>	<b>6</b>
4.1	Model relacyjny . . . . .	6
<b>5</b>	<b>Diagram przypadków użycia</b>	<b>7</b>
<b>6</b>	<b>Diagramy klas</b>	<b>8</b>
6.1	Przykładowa klasa backendu . . . . .	8
6.2	Przykładowy element w API . . . . .	9
<b>7</b>	<b>Projekty interfejsu graficznego</b>	<b>9</b>
7.1	Panel administratora . . . . .	9
7.1.1	Lista elementów . . . . .	9
7.1.2	Ekran logowania . . . . .	9
7.1.3	Ekran widoku elementu . . . . .	10
7.2	Strona główna . . . . .	10
7.2.1	Homepage - góra . . . . .	10
7.2.2	Homepage - dół . . . . .	11
7.2.3	Widok ofert . . . . .	11
7.2.4	Widok posta . . . . .	12
7.2.5	Dane kontaktowe . . . . .	12
7.2.6	Formularz kontaktowy . . . . .	12
<b>8</b>	<b>Najważniejsze fragmenty kodu</b>	<b>13</b>
<b>9</b>	<b>Analiza bezpieczeństwa</b>	<b>17</b>
<b>10</b>	<b>Podsumowanie</b>	<b>17</b>
10.1	Cele zrealizowane . . . . .	17
10.2	Cele niezrealizowane . . . . .	18
10.3	Napotkane problemy . . . . .	18

10.4 Perspektywa rozwoju . . . . .	18
------------------------------------	----

# 1 Charakterystyka ogólna projektu

Projekt zakłada stworzenia systemu zarządzania stroną internetową zawierającą informacje o siłowni i jej ofercie.

Aplikacja składająca się z części serwerowej oraz klienckiej ma pozwalać na edytowanie treści strony przez nieznanego się na programowaniu użytkownika. Użytkownik ma mieć możliwość zmiany zdjęć, informacji oraz banerów bez stosowania kodu.

Poza zmianą treści głównej strony użytkownik powinien również mieć możliwość dodawania artykułów dotyczących wydarzeń organizowanych przez siłownię bądź też innych nowości związanych z placówką.

Do stworzenia warstwy wizualnej inspiracja została zaczerpnięta ze strony [TemplateMonster](#).

## **2 Wymagania**

### **2.1 Funkcjonalne**

Wymagania z podziałem na role użytkowników:

#### **2.1.1 Redaktor**

- edycja, dodawanie, usuwanie artykułów i kategorii
- edycja banerów strony (tekst i obraz)
- edycja elementu witającego użytkownika
- edycja, dodawanie, usuwanie informacji o trenerach w tym linki do ich serwisów społecznościowych
- dodawanie i usuwanie opinii klientów
- wylogowywanie się
- zmiana własnego hasła

#### **2.1.2 Administrator**

- posiadanie wszystkie możliwości redaktora
- dodawanie, usuwanie i edycja redaktorów
- edycja informacji o ofercie siłowni
- edycja informacji o kontakcie i placówkach
- zmiana maila na którego przychodzą wiadomości z formularza kontaktowego

#### **2.1.3 Czytelnik**

- przeglądanie zawartości strony
- możliwość skorzystania z formularza kontaktowego
- możliwość zalogowania do konta administratora bądź redaktora
- wyszukiwanie artykułów po nazwie autora lub tytule

## 2.2 Niefunkcjonalne

- system sprawdza poprawność formatu wprowadzanych danych (np. email)
- zmniejszanie rozdzielczości wysłanych zdjęć
- autoryzacja z pomocą tokenów JWT
- system kompatybilny z najpopularniejszymi przeglądarkami (Chrome, Firefox, Safari)
- użytkownik może wykonać wszystkie akcje bez ingerencji w kod źródłowy
- system zostanie przetestowany z użyciem testów jednostkowych
- możliwość pracy wielu użytkowników jednocześnie
- system wczytuje stronę w czasie poniżej 3 sekund

## 3 Architektura systemu

### 3.1 Narzędzia

Narzędzia użyte do tworzenia projektu:

- **GitHub** - repozytorium w którym trzymany jest kod
- **Notion** - platforma do śledzenia postępu prac i dzielenia pracy na zadania
- **Docker** - oprogramowanie do wirtualizacji, użyte do łatwego uruchamiania kodu na różnych środowiskach

### 3.2 Backend aplikacji

Technologie użyte do stworzenia części serwerowej aplikacji:

- **Java** - język użyty do stworzenia API
- **Spring** - framework języka Java w którym stworzono API
- **PostgreSQL** - Baza danych użyta w aplikacji

### 3.3 Frontend aplikacji

Technologie użyte do stworzenia części klienckiej projektu:

- **TypeScript** - rozszerzenie języka JavaScript
- **HTML** - język znaczników użyty do budowy układu strony
- **CSS** - język służący do opisu formy prezentacji strony
- **React.js** - biblioteka języka JavaScript obsługująca także TypeScript
- **Bootstrap 5** - biblioteka ułatwiająca dostosowywanie stylu strony.

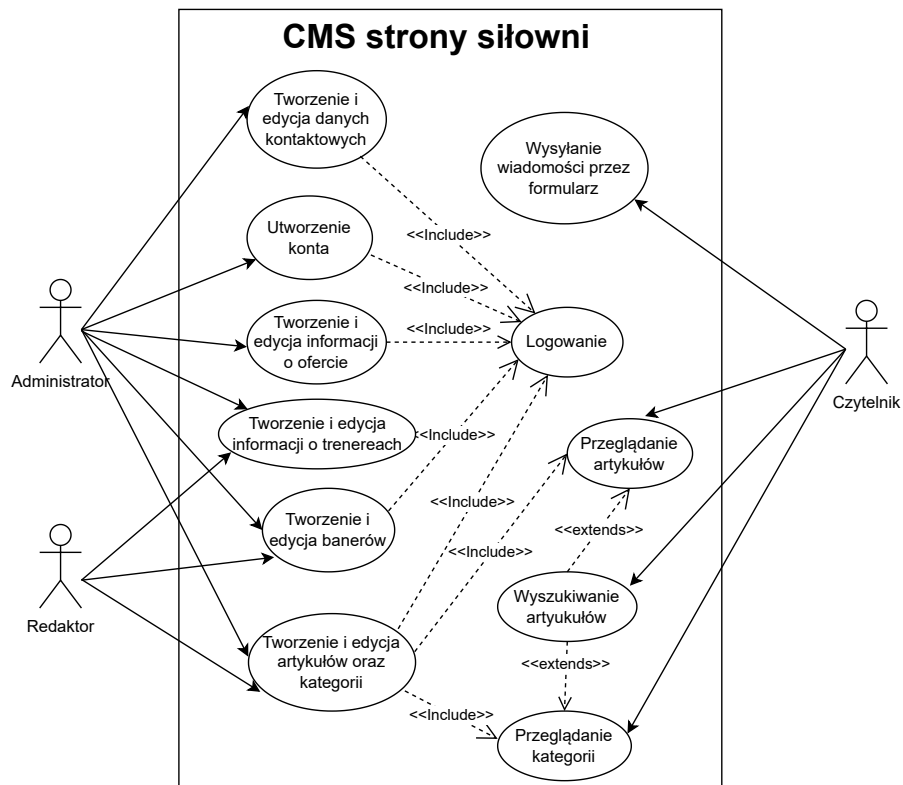
## 4 Schemat baz danych

### 4.1 Model relacyjny



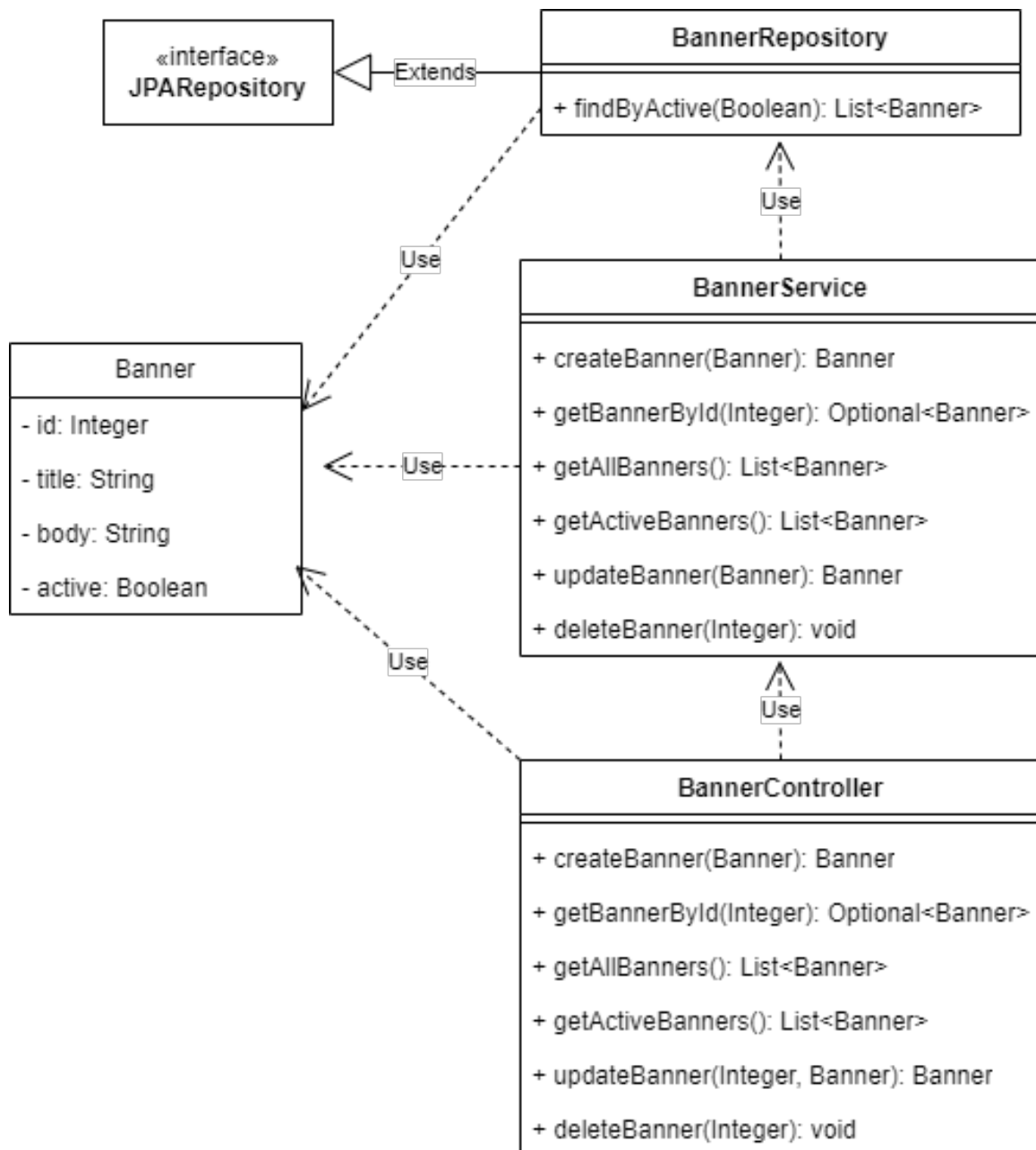


## 5 Diagram przypadków użycia



## 6 Diagramy klas

### 6.1 Przykładowy diagram backendu



## 6.2 Przykładowy element w API

# 7 Projekty interfejsu graficznego

## 7.1 Panel administratora

### 7.1.1 Lista elementów

Użytkownicy

Lokacje

Oferty

Posty

Banery

Opinie

Trenerzy

Kategorie

Utwórz

Nazwa	Cena	Aktywna	Akcje
Oferta 1	99.99	Tak	<div>Zobacz</div> <div>Edytuj</div> <div>Usuń</div>
Oferta 2	50	Tak	<div>Zobacz</div> <div>Edytuj</div> <div>Usuń</div>
Oferta 3	64.99	Tak	<div>Zobacz</div> <div>Edytuj</div> <div>Usuń</div>

Wyloguj

### 7.1.2 Ekran logowania

Nazwa użytkownika:

Nazwa użytkownika

Hasło:

Hasło

Zaloguj

[Zapomniałem hasła](#)

### 7.1.3 Ekran widoku elementu

Użytkownicy

Lokacje

Oferty

Posty

Banery

Opinie

Trenerzy

Kategorie

Wyloguj

Nazwa oferty:

Oferta

Treść oferty

**B** / *I* U ~~S~~ {} x<sup>2</sup> x<sub>2</sub> Normal 16 Font

Cena:

0

Zniżka w %:

0

☐ Wyświetlaj wśród ofert

Utwórz

## 7.2 Strona główna

### 7.2.1 Homepage - góra

GymApp

ArtykułyOfertaKontakt

Testowy baner

Jest to bardzo fajny testowy tekst pod banerem

Nasi trenerzy:

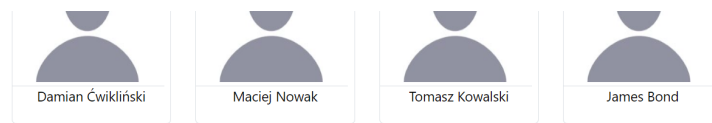
Damian Ćwikliński

Maciej Nowak

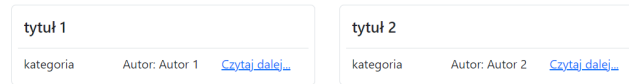
Tomasz Kowalski

James Bond

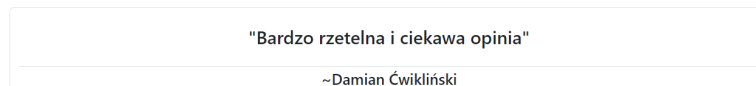
## 7.2.2 Homepage - dół



### Popularne posty:



### Opinie naszych klientów:



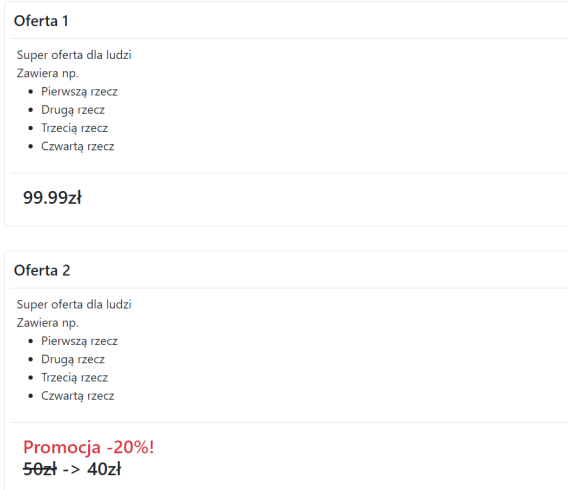
Przejdź do naszej [oferty](#).

Damian Ćwikliński

## 7.2.3 Widok ofert

GymApp

[Artykuły](#) [Oferta](#) [Kontakt](#)



## 7.2.4 Widok posta

GymApp

ArtykułyOfertaKontakt

Super post

Kategoria 1

Autor: Damian

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec aliquam, nunc et convallis condimentum, nisi sem dictum lectus, sed semper leo arcu sed erat. Pellentesque eros dui, sodales non imperdiet nec, facilisis ac nunc. Aenean dignissim dolor ut eros varius cursus. Fusce gravida blandit lorem vitae varius. Aenean sit amet tortor sed ex laoreet molestie pretium vel diam. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam hendrerit lacinia enim. Vestibulum tristique dolor sit amet metus varius, eu scelerisque diam congue. Nullam ut condimentum mauris, quis elementum eros.

Praesent commodo pretium est a blandit. Nullam consequat ultrices velit eget tincidunt. Proin eget tellus id lorem aliquam sagittis pretium nec ligula. Maecenas a tristique odio. Phasellus sapien ligula, elementum non ante at, dapibus tincidunt nulla. Phasellus aliquam ultrices tellus at malesuada. Donec malesuada metus libero, ac rutrum neque pretium vel. Etiam rhoncus suscipit ipsum. Vestibulum finibus et ante vitae rhoncus. Nunc consequat dui et sollicitudin luctus. Donec in maximus libero. Donec in elit at eros pellentesque finibus a ut erat. Donec eget lectus dignissim odio volutpat semper at at orci. Vivamus sed neque nibh. Pellentesque lobortis leo eu tortor ornare, et auctor felis blandit. Vestibulum pulvinar mauris elementum est iaculis, at aliquam

## 7.2.5 Dane kontaktowe

GymApp

ArtykułyOfertaKontakt

Wybierz placówkę:

Gniezno, ul. Rynek 1/1

Miasto:

Gniezno

Ulica:

ul. Rynek 1/1

Numer telefonu:

123123123

E-mail:

gniezno@gmail.com

## 7.2.6 Formularz kontaktowy

Formularz kontaktowy:

(Wiadomość zostanie wysłana do wybranej placówki)

Imię i nazwisko:

Twoje imię i nazwisko

Email:

Twój email

Tytuł wiadomości:

Tytuł

Wiadomość:

Twoja wiadomość...

Wyślij wiadomość

## 8 Najważniejsze fragmenty kodu

Wysyłanie maili

```
@Service
@RequiredArgsConstructor
public class EmailService {
    private final JavaMailSender emailSender;

    public void sendFromContactForm(ContactFormDto message)
    throws UnsupportedEncodingException, MessagingException{
        sendEmail(message.getSendTo(), message.getSubject(),
            message.getMessage(), message.getSenderEmail(),
            message.getSenderName());
    }

    private void sendEmail(String email, String subject, String content,
        MimeMessage message = emailSender.createMimeMessage();
        MimeMessageHelper helper = new MimeMessageHelper(message);

        helper.setFrom(clientEmail, clientName);
        helper.setTo(email);
        helper.setSubject("[Formularz]" + subject);
        helper.setText("[ " + clientEmail + "]\n" + content);

        emailSender.send(message);
    }
}
```

## Konfiguracja zabezpieczeń

@Bean

```
public SecurityFilterChain filterChain(HttpSecurity httpSecurity)
    throws Exception{
    httpSecurity.csrf(AbstractHttpConfigurer::disable)
        .cors(Customizer.withDefaults())
        .authorizeHttpRequests(auth -> {
            auth.requestMatchers("/api/**").permitAll();
            auth.anyRequest().authenticated();
        })
        .sessionManagement(sessionManagementCustomizer -> {
            sessionManagementCustomizer
        })
        .sessionCreationPolicy(SessionCreationPolicy.STATELESS);
    })
    .authenticationProvider(authenticationProvider)
    .addFilterBefore(jwtAuthenticationFilter ,
        UsernamePasswordAuthenticationFilter.class)
    .httpBasic(Customizer.withDefaults());
return httpSecurity.build();
}
```

## Tworzenie tokena JWT

Tworzenie tokena JWT

```
private String buildToken(
    Map<String , Object> extraClaims ,
    UserDetails userDetails ,
    long expiration
) {
    return Jwts
        .builder()
        .setClaims(extraClaims)
        .setSubject(userDetails.getUsername())
        .setIssuedAt(new Date(System.currentTimeMillis()))
        .setExpiration(new Date(System.currentTimeMillis() + expiration))
        .signWith(getSignInKey(), SignatureAlgorithm.HS256)
        .compact();
}
```



Pobieranie elementu z API

```
export const getOneObject = async (id: string | undefined,
endpoint: string, setData: (data: SetStateAction<any>) => void) => {
  try {
    await axios.get('http://localhost:8080/api/${endpoint}/${id}',
    config).then(response => {
      console.log(response)
      if (response.status === 200) {
        setData(response.data)
      }
      else {
        console.log("Could not get data");
      }
    })
  } catch (error) {
    console.error('Error during fetching:', error);
  }
}
```

Funkcja logowania do aplikacji

```
try {
  const response = await axios.post('http://localhost:8080/api/auth/login',
  {username, password})

  if (response.status === 200) {
    sessionStorage.setItem("token", response.data.token);
    sessionStorage.setItem("role", response.data.roleName);
    navigate("/manage");
  } else {
    console.error('Login failed ');
  }
} catch (error) {
  console.error('Error during login:', error);
}
```

Tworzenie listy z elementami

```
const renderCrudButtons = (id: number): ReactNode => {
  return (
    <td className='btns-crud'>
      <Link className='mx-1' to={id.toString()}>
        <button className='btn btn-primary'>Zobacz</button>
      </Link>
      <Link className='mx-1' to={id + "/edit"}>
        <button className='btn btn-primary'>Edytuj</button>
      </Link>
      <Link className='mx-1' to={""}>
        <button className='btn btn-primary' onClick={() => handle
      </Link>
    </td>)}

const renderHeaders = (headers: Array<string>): ReactNode => {
  return headers.map(header => <th key={header}>{header}</th>)}

function renderListItems(items: Array<ListItem>) {
  if (items.length !== 0){
    return items.map(item => <tr key={`_${item.id}`}>{item.content.map
  }
  return <tr key='1'>Brak danych do wyswietlenia</tr>
}

return (
  <div className='container p-3'>
    <Link to="create">
      <button className='btn btn-primary'>Utworz</button>
    </Link>
    <table className='table table-bordered border mt-5'>
      <thead className='table-dark'>
        <tr key="0">
          {renderHeaders ([...headers, " Akcje"])}
        </tr>
      </thead>
      <tbody>
        {renderListItems(items)}
      </tbody>
    </table>
  </div>)
```

## 9 Analiza bezpieczeństwa

Strona nie przechowuje żadnych danych wrażliwych, co oznacza, że bezpieczeństwo choć ważne, nie jest priorytetem. Użytkownikami systemu z założenia są osoby zaufane.

Do zabezpieczenia elementów po stronie backendu wykorzystano bibliotekę spring security oraz tokeny JWT. Tokeny nie pozwalają na dostęp do danych nieupoważnionym osobom nawet gdyby odnalazły endpointy, bądź dostały się nieupoważnienie do chronionych części strony.

Po stronie frontendu zablokowano pewne możliwości osobom niezalogowanym lub z niskimi uprawnieniami. Zakładki dostępne dla admina są chowane, gdy zalogowany jest zwykły użytkownik, nieautoryzowana próba wejścia do chronionych części aplikacji kończy się wyrzuceniem osoby na stronę logowania.

Elementy tworzone edytorem tekstu tak jak treść posta, konwertowana jest na format JSON co powinno zabezpieczyć przed niechcianymi egzekucjami kodu.

## 10 Podsumowanie

### 10.1 Cele zrealizowane

- Główna strona z banerami, ofertami, postami, informacjami kontaktowymi
- Formularz kontaktowy wysyłający maile na adresy placówek
- Panel admina pozwalający na edycję głównej strony
- Logowanie do panelu admina oraz zabezpieczenie aplikacji przy pomocy tokenów JWT
- Podział na konta użytkowników i administratorów
- Edycja, dodawanie i usuwanie
  - banerów
  - użytkowników
  - postów
  - kategorii
  - oferty
  - informacji kontaktowych
  - trenerów
  - opinii klientów

## **10.2 Cele niezrealizowane**

- Dodawanie obrazów przez użytkownika
- Resetowanie hasła
- Brak testów jednostkowych

## **10.3 Napotkane problemy**

- Słaba znajomość wybranych technologii
- Problemy z konfiguracją usługi SMTP
- Decyzja o zrobieniu wszystkiego od zera z perspektywy czasu była złą decyzją

## **10.4 Perspektywa rozwoju**

- Zmiana zdjęć trenerów i różne obrazy dla banerów
- Dodanie dodatkowych sekcji na stronie głównej
- Dodanie kolejnych zakładek w górnym pasku
- Opcje wyszukiwania postów
- Formularz kontaktowy dla trenerów
- możliwość zmieniania kolejności elementów strony