**UNIWERSYTET WARMIŃSKO MAZURSKI W OLSZTYNIE**

**WYDZIAŁ MATEMATYKI I INFORMATYKI**

**Dominik Mateusz Lewczyński**

**Kierunek: Informatyka**

**Specjalność: Informatyka ogólna**

**Implementacja Aplikacji internetowej czasopisma naukowego**

Praca inżynierska wykonana

w Katedrze Multimediów i Grafiki Komputerowej

pod kierunkiem

dr hab. Aleksandra Denisiuka

**Olsztyn 2023.**

**UNIVERSITY OF WARMIA AND MAZURY IN OLSZTYN**

**THE FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**Dominik Mateusz Lewczyński**

**Field of study: Computer science**

**Specialty: General computer science**

**Implementation of an internet application of a scientific journal**

Engineering Thesis written in

Department of Multimedia and Computer Graphics

under the supervision of

dr hab. Aleksander Denisiuk

**Olsztyn 2023.**

**Spis treści**

[**1.** **Streszczenie** 4](#_Toc125447174)

[**2.** **Abstract** 5](#_Toc125447175)

[**3.** **Wstęp** 6](#_Toc125447176)

[**4.** **Opis podrozdziałów** 7](#_Toc125447177)

[**5.** **Wymagania ogólne** 8](#_Toc125447178)

[**5.1.** **Cel pracy** 8](#_Toc125447179)

[**5.2.** **Podstawowe wymagania użytkownika** 8](#_Toc125447180)

[**6.** **Dokumentacja projektowa** 9](#_Toc125447181)

[**6.1.** **Wymagania sprzętowe** 9](#_Toc125447182)

[**6.2.** **Słownik pojęć systemowych** 9](#_Toc125447183)

[**6.3.** **Aktorzy** 9](#_Toc125447184)

[**6.4.** **Wymagania funkcjonalne** 10](#_Toc125447185)

[**6.5.** **Diagram przypadków użycia** 11](#_Toc125447186)

[**6.6.** **Diagram hierarchii funkcji** 12](#_Toc125447187)

[**6.7.** **Opis funkcjonalności** 13](#_Toc125447188)

[**7.** **Implementacja** 23](#_Toc125447189)

[**7.1.** **Strategia realizacji projektu** 23](#_Toc125447190)

[**7.2.** **Główne moduły aplikacji** 24](#_Toc125447191)

[**7.3.** **Wzorce wykorzystane w projekcie** 24](#_Toc125447192)

[**7.4.** **Przebieg informacji w systemie** 25](#_Toc125447193)

[**7.5.** **Frontend** 25](#_Toc125447194)

[**7.6.** **Backend** 26](#_Toc125447195)

[**7.7.** **Bazy danych**. 26](#_Toc125447196)

[**7.8.** **Testy** 26](#_Toc125447197)

[**8.** **Instrukcja użytkownika** 27](#_Toc125447198)

[**8.1.** **Interfejs strony głównej** 27](#_Toc125447199)

[**8.2.** **Konto użytkownika** 30](#_Toc125447200)

[**8.3.** **Dodawanie artykułu** 33](#_Toc125447201)

[**8.4.** **Przygotowanie do publikacji** 35](#_Toc125447202)

[**9.** **Licencje** 45](#_Toc125447203)

[**9.1.** **Narzędzia** 45](#_Toc125447204)

[**9.2.** **Grafiki** 45](#_Toc125447205)

[**9.3.** **Biblioteki zewnętrzne** 46](#_Toc125447206)

[**9.4.** **Źródła wiedzy programistycznej** 46](#_Toc125447207)

[**10.** **Podsumowanie** 47](#_Toc125447208)

[**11.** **Bibliografia** 48](#_Toc125447209)

# **Streszczenie**

Praca przedstawia proces tworzenia „**Aplikacji internetowej czasopisma naukowego**”. Ma to być aplikacja umożliwiająca autorowi artykułu na jego przesłanie do Wydawnictwa. Artykuł ma przejść przez proces redagowania, recenzowania oraz publikowania na stronie głównej. Każdy użytkownik, który się zarejestruję jako **Autor** ma mieć możliwość **dodania kilku artykułów,** **ich aktualizowania** oraz **usuwania**. Każdy użytkownik, który się zarejestruję jako **Recenzent** ma mieć możliwość **zrecenzowania otrzymanej pracy** od **Redaktora**. Każdy użytkownik, który się zarejestruje jako **Redaktor** ma mieć możliwość **redagowania, czyli wysłania do recenzji oraz ustalenia czy publikować czy nie**. **Wydawnictwo** jest jedynym użytkownikiem, który nie musi się rejestrować, ponieważ jest już dodany do systemu   
i ma możliwość publikowania tekstów na stronie głównej oraz zarządzania kontami użytkowników. Aplikację stworzyłem przy użyciu biblioteki React.js do frontendu i przy użyciu biblioteki Express.js do backendu oraz przy użyciu MySQL do tworzenia bazy danych. Napisałem oraz przedstawiłem dokumentację projektową powstałą na podstawie wymagań użytkownika oraz wstępnych założeń. Dodałem obszerną instrukcję użytkownika wraz z odpowiednimi zrzutami ekranów. Całość wieńczy podsumowanie, mówiące o zrealizowaniu założeń wstępnych wraz ze wskazaniem możliwości dalszego rozwoju pracy.

# **Abstract**

The work presents the process of creating the **"Internet application of a scientific journal"**. It is supposed to be an application enabling the **Author** of the article to send it to the **Publisher**. The article is to go through the process of editing, reviewing and publishing on the home page. Each user who registers as an **Author** is to be able   
to **add several articles, update them** and **delete them**. Each user who registers as   
a **Reviewer** is to be able to **review the work received from the Editor**. Each user who registers as an **Editor** is to be able to **edit, i.e. send for review and decide whether to publish or not**. **The publisher** is the only user who does not need   
to register because he is already added to the system and has the ability to publish texts on the home page and manage user accounts. I created the application using the React.js library for the frontend and using the Express.js library for the backend and using MySQL to create the database. I wrote and presented project documentation based on user requirements and initial assumptions. I have added a comprehensive user manual with relevant screenshots. The whole is crowned with a summary, talking about the implementation of the initial assumptions together with an indication   
of the possibility of further development of the work.

# **Wstęp**

Publikacja pracy naukowej, technicznej, badawczej lub wyników badań jest jednym z kryteriów oceny zgłaszanych projektów jak i oceny autora. Wobec szybko zmieniającej się wiedzy jak i dużej liczby potencjalnych autorów zainteresowanych daną dziedziną, opóźnienia w publikacji nie powinny występować. Czasami wymaga się od autora drukowania materiałów i dostarczenie pracy do wydawnictwa. Naniesienie ewentualnych poprawek wymaga ponownego drukowania pracy   
i ponownego dostarczenia.

Innym problemem, jaki występuje podczas składania publikacji, jest konieczność przesłania plików z artykułem oraz innych materiałów za pośrednictwem poczty elektronicznej lub poczty tradycyjnej. Jest to czasochłonne i niezbyt wygodne, ponieważ często są to duże pliki i trudno je przesłać poprzez pocztę elektroniczną. Wysyłka pracy pocztą tradycyjną może zakończyć się jej zaginięciem bez możliwości odzyskania.

Kolejnym problemem jest proces recenzowania artykułów. Proces ten jest kluczowy dla jakości publikacji, ale często trwa długo. Recenzenci mają swoje zajęcia, inne prace do oceny, muszą przeczytać artykuł, napisać recenzję i przesłać ją z powrotem do redakcji.

Ostatecznie, proces publikowania artykułów w tradycyjny sposób jest czasochłonny i skomplikowany. W celu rozwiązania tych problemów, wiele czasopism naukowych decyduje się na implementację aplikacji internetowej.

W ostatnich latach coraz większą popularność zyskują publikacje naukowe dostępne w formie elektronicznej. Nie tylko ułatwiają one dostęp do najnowszych osiągnięć naukowych, ale także pozwalają na szybkie i łatwe ich przeszukiwanie oraz udostępnianie innym osobom. Czasopisma naukowe są ważnym źródłem wiedzy   
i inspiracji dla pracowników nauki oraz studentów, dlatego też coraz częściej są one dostępne w formie elektronicznej, umożliwiającej szybki dostęp do aktualnych publikacji.

Jednym z narzędzi umożliwiających publikowanie i udostępnianie czasopism naukowych w formie elektronicznej jest aplikacja internetowa. Aplikacja internetowa czasopisma naukowego pozwala na zautomatyzowanie procesu składania   
i publikowania artykułów. Dzięki niej, autorzy mogą łatwiej przesłać swoje prace,   
a recenzenci mogą szybciej i łatwiej przeglądać i recenzować artykuły. Aplikacja ta pozwala także na szybkie opublikowanie na stronie głównej przez co czas publikowania zostaje skrócony.

# **Opis podrozdziałów**

Niniejsza praca opisuje aplikację internetową stworzoną za pomocą technologii React.js, Express.js, jej możliwości, budowę, projekt oraz zakres działania. Wszystko to opisane jest w następujących rozdziałach:

* Wymagania ogólne – podstawowe założenia powstającej aplikacji,
  + Cel pracy – określono cel powstania aplikacji,
  + Podstawowe wymagania użytkownika – przedstawienie podstawowych wymagań użytkownika wobec działania aplikacji,
* Dokumentacja projektowa – część poświęcona dokumentacji projektowej
  + Wymagania sprzętowe – opis wymagań dotyczących użytkowania programu,
  + Słownik pojęć systemowych – opis poszczególnych pojęć systemowych
  + Aktorzy – opis poszczególnych Użytkowników, występujących   
    w systemie,
  + Wymagania funkcjonalne – przedstawienie głównych założeń programu,
  + Diagram przypadków użycia – część poświęcona diagramowi przypadków użycia,
  + Diagram hierarchii funkcji – część poświęcona funkcjonalnościom przedstawionym na diagramie hierarchii funkcji,
  + Opis funkcjonalności – część poświęcona szczegółowemu opisaniu wszystkich funkcjonalności.
* Instrukcja Użytkownika – opis działania programu oraz jego użytkowanie,
  + Konto Użytkownika – czynności związane z kontem Użytkownika,
  + Dodanie artykułu – część poświęcona dodawaniu artykułów   
    przez Autora,
  + Przygotowanie do publikacji – część poświęcona procesowi polegającemu na recenzowaniu, redagowaniu oraz opublikowaniu artykułu na stronie głównej
* Licencje – opis licencji dotyczących technologii oraz elementów wykorzystanych w programie,
  + Narzędzia,
  + Grafika,
  + Biblioteki zewnętrzne,
  + Źródła wiedzy programistycznej.
* Podsumowanie – bilans całej pracy, przedstawienie wyciągniętych wniosków oraz oszacowanie stopnia zrealizowania założeń początkowych.
* Streszczenie – skrócony opis pracy.
* Abstract – skrócony opis pracy (streszczenie) w języku angielskim.

# **Wymagania ogólne**

Celem niniejszego rozdziału jest przedstawienie, jakie były początkowe założenia oraz wymagania użytkownika docelowego wobec tworzonej aplikacji. W podrozdziale 3.1 opisuję, dlaczego tworzona jest aplikacja. W podrozdziale 3.2 wyjaśniam, jakie były wymagania użytkownika wobec tworzonej aplikacji.

## **Cel pracy**

Celem pracy inżynierskiej jest zaprojektowanie i implementacja aplikacji internetowej do publikowania artykułów w czasopismach naukowych, która wyeliminuję potrzeby wizyt w wydawnictwie, ulepszenie procesu edycji i jego maksymalne przyspieszenie. Obecnie wiedza w każdej dziedzinie rozwija się bardzo szybko. Opóźnienia w publikacjach nie są więc pożądane.

W pracy zostaną przedstawione wymagania funkcjonalne oraz techniczne aplikacji, a także zostanie przedstawiona jej architektura oraz opisane poszczególne jej elementy.

## **Podstawowe wymagania użytkownika**

Poniżej przedstawiam wymagania Użytkownika wobec tworzonej aplikacji internetowej:

* Najnowsza wersja przeglądarki internetowej do poprawnego działania aplikacji. Dostępne przeglądarki to np.: Google Chrome, Mozilla Firefox, Microsoft Edge oraz Opera.
* Komputer, na którym powinna działać aplikacja powinien spełniać minimalne wymagania. (procesor Pentium 4, pamięć RAM 4GB, system operacyjny Windows XP, dysk twardy 20GB)
* Połączenie z Internetem jest wymagane, ponieważ serwer musi stać na szybkim i stabilnym łączu.
* Aplikacja ma być intuicyjna oraz estetyczna. Użytkownik nie powinien mieć problemu z korzystaniem z aplikacji.
* Aplikacja musi być odpowiednio zabezpieczona by chronić dane użytkowników.
* Aplikacja powinna działać na dużych jak i mniejszych ekranach.
* Aplikacja musi być kompatybilna z innymi systemami IT.
* Aplikacja powinna być wydajna oraz powinna działać szybko.

# **Dokumentacja projektowa**

## **Wymagania sprzętowe**

Wymagania aplikacji internetowej czasopisma naukowego dotyczące sprzętu wyglądają następująco:

* Komputer lub urządzenie mobilne z dostępem do Internetu,
* Połączenie internetowe: Dowolne (Wi-Fi lub transfer sieci komórkowej),
* Zainstalowana przeglądarka internetowa: Do wyboru jest Google Chrome, Mozilla Firefox, Microsoft Edge, Safari oraz Opera).

Technologie jakich użyto do stworzenia aplikacji zapewnia działanie na każdej przeglądarce na każdym urządzeniu. Aplikacja jest także responsywna przez co jest dostosowana do każdego ekranu o różnej wielkości.

* 1. **Słownik pojęć systemowych:**
* **Autor** = imię + nazwisko + dane użytkownika. Użytkownik systemu, który tworzy nowe artykuły.
* **Recenzent** = imię + nazwisko + dane użytkownika. Użytkownik systemu, który recenzuje przesłany artykuł.
* **Redaktor** = imię + nazwisko + dane użytkownika. Użytkownik systemu, który redaguje przesłany artykuł.
* **Wydawnictwo** = imię + nazwisko + dane użytkownika. Użytkownik systemu, operator systemu. Jego zadaniem jest publikowanie artykułów.
* **Użytkownik** = adres e-mail + login + hasło + rola użytkownika. Osoba podlegająca autoryzacji w systemie.
* **Publikacja** – zamieszczenie artykułu (który przeszedł etap przygotowania do publikacji) na stronie internetowej Wydawnictwa.

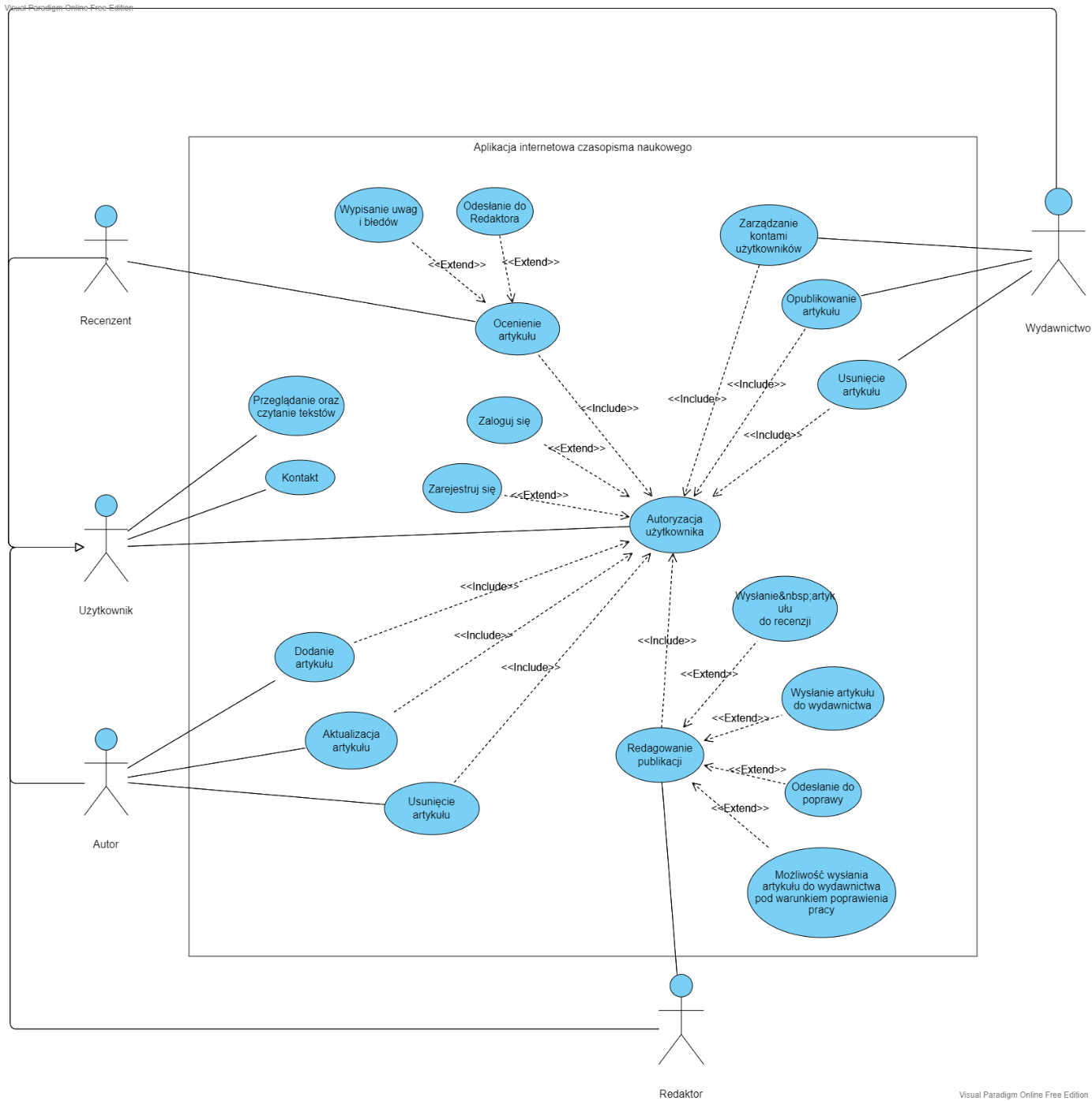
## **Aktorzy**

* **Autor** – Użytkownik systemu która ma za zadanie dodawać, aktualizować oraz usuwać artykuły.
* **Recenzent** – Użytkownik systemu, który ma za zadanie zrecenzować artykuł np.: wypisać błędy bądź zamieścić inne uwagi.
* **Redaktor** – Użytkownik systemu, który ma zredagować otrzymany tekst, przesłać go do Recenzenta w celu jego oceny. Ustalenie czy publikacja będzie realizowana bądź nie oraz wysłanie artykułu do wydawnictwa, jeśli ustalono, że praca zostanie opublikowana.
* **Wydawnictwo** – Użytkownik systemu, który ma za zadanie opublikować na swojej stronie głównej przesłany artykuł. Może także zarządzać kontami aktorów poprzez aktualizowanie danych: imię, nazwisko, adres E-mail, login, hasło oraz role użytkownika.
* **Użytkownik** – jest to osoba, która podlega autoryzacji w systemie.

## **Wymagania funkcjonalne**

* **Autoryzuj użytkownika** - zbiór usług pozwalających na utworzenie konta oraz dostęp do niego.
  + **Zarejestruj się** – pozwala Użytkownikowi utworzyć nowe konto.
  + **Zaloguj się** – po podaniu odpowiednich danych, pozwala uzyskać dostęp do konta danego Użytkownika.
* **Zarządzanie kontami Użytkowników** – usługa pozwalająca Wydawnictwu na zarządzanie kontami Autorów, Recenzentów oraz Redaktorów.
* **Przeglądanie oraz czytanie tekstów** – usługa pozwalająca przejrzeć dostępne artykuły.
* **Kontakt** – usługa umożliwiająca skontaktowanie się z czasopismem naukowym poprzez formularz kontaktowy.
* **Dodanie artykułu** – usługa pozwalająca na dodanie tekstu przez Autora poprzez dodanie pliku PDF.
* **Aktualizacja artykułu** – usługa pozwalająca Autorowi na aktualizację przesłanej pracy poprzez zmianę tytułu, jej opisu oraz przesłanie nowego pliku PDF.
* **Usuniecie artykułu** – usługa pozwalająca Autorowi oraz Wydawnictwu na usunięcie przesłanej pracy
* **Ocenienie artykułu** – ocena tekstu przez Recenzenta.
  + **Wypisanie uwag i błędów** – usługa pozwalająca na wypisanie uwag oraz błędów.
  + **Odesłanie do redaktora** – Recenzent ma możliwość odesłania pracy do Redaktora po znalezieniu błędów, gdy są jakieś uwagi albo kiedy tych błędów nie znaleziono, a uwag nie ma.
* **Redagowanie artykułu** – redagowanie pracy przez Redaktora
  + **Wysłanie artykułu do recenzji** – pozwala na wysłanie tekstu do Recenzenta w celu jego oceny.
  + **Możliwość wysłania artykułu do wydawnictwa pod warunkiem poprawienia pracy** – Redaktor ma możliwość odesłania tekstu po znalezieniu błędów lub dołączeniu uwag przez Recenzenta. Informuje Autora o możliwości dokonania publikacji artykułu po poprawieniu błędów i uwzględnieniu uwag Recenzenta.
  + **Brak zgody na opublikowanie** – Redaktor ma możliwość poinformowania Autora o tym, że artykuł nie zostanie opublikowany.  
    Dołącza informację o powodach braku publikacji.
  + **Wysłanie artykułu do wydawnictwa** – usługa pozwalająca na wysłanie pracy do Wydawnictwa w celu opublikowania jej na stronie głównej.
* **Opublikowanie artykułu** – usługa pozwalająca Wydawnictwu na opublikowanie artykułu na swojej stronie głównej.

## **Diagram przypadków użycia**



Rysunek Diagram przypadków użycia dla aplikacji internetowej czasopisma naukowego

Na powyższym diagramie przypadków użycia została przedstawiona funkcjonalność aplikacji, z której mogą korzystać Użytkownicy. Autor, Recenzent, Redaktor oraz Wydawnictwo mają przedzielone konkretne funkcjonalności. Są oni połączeni z aktorem o nazwie Użytkownik, ponieważ funkcjonalności „Przeglądanie oraz czytanie tekstów” oraz „Autoryzuj użytkownika” to są ich wspólne funkcjonalności, którym podlegają.

## **Diagram hierarchii funkcji**

Rysunek Diagram FHD

. Na powyższym diagramie została umiejscowiona hierarchia poszczególnych funkcjonalności i dodana do odpowiednich Aktorów. Jak można zauważyć Autor, Recenzent, Redaktor oraz Wydawnictwo są połączone z aktorem Użytkownik ze względu na wspólne funkcjonalności.

## **Opis funkcjonalności**

W tym podrozdziale opisze szczegółowo wszystkie funkcjonalności w postaci scenariuszy.

Tabela 1 Dokumentacja przypadku użycia „Autoryzacja Użytkwonika”

|  |  |
| --- | --- |
| **Nazwa** | **Autoryzacja Użytkownika** |
| Numer | 0 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Użytkownik |
| Krótki opis | Przeprowadzenie autoryzacji Użytkownika |
| Warunki wstępne | Dostęp do Systemu informatycznego |
| Warunki końcowe | Logowanie lub rejestracja do systemu |
| Główny przepływ zdarzeń | 1) Użytkownik loguje się do systemu. Podaje login oraz hasło. Jeżeli dane są poprawne to Użytkownik zostaje zalogowany.  2) Użytkownik rejestruje się do systemu. Podaje imię, nazwisko, e-mail, login, hasło oraz rolę Użytkownika. Jeżeli dane zostały podane prawidłowo to Użytkownik zostaje zarejestrowany. |
| Alternatywne przepływy zdarzeń | 1a) Błędne dane – system wyświetla błąd. Użytkownik nie jest zalogowany do systemu. System ponawia czynność logowania  2a) Błędne dane – system wyświetla błąd. Konto Użytkownik nie zostaje utworzone. System ponawia czynność rejestracji |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela 2 Dokumentacja przypadku użycia „Zarządzanie kontami użytkowników”

|  |  |
| --- | --- |
| **Nazwa** | **Zarządzanie kontami Użytkowników** |
| Numer | 1 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Wydawnictwo |
| Krótki opis | Funkcja pozwalająca Wydawnictwu na zarządzanie kontem Użytkowników (zmianę hasła, danych, itp.) |
| Warunki wstępne | Wydawnictwo jest zalogowane |
| Warunki końcowe | Wydawnictwo zarządziło kontem Użytkownika |
| Główny przepływ zdarzeń | 1. Wydawnictwo przechodzi do strony „Użytkownicy” 2. Wydawnictwo widzi listę Użytkowników z opcjami „Edytuj” oraz „Usuń” 3. Chcąc edytować dane Użytkownika Wydawnictwo wybiera opcję „Edytuj” przy wybranym Użytkowniku 4. Po kliknięciu przycisku „Edytuj” Wydawnictwo ma możliwość zmiany imienia, nazwiska, adresu e-mail, loginu, hasła oraz roli Użytkownika. 5. Wydawnictwo po wprowadzeniu zmian zatwierdza je przyciskiem „Zapisz zmiany” 6. Chcąc usunąć Użytkownika wybiera opcję „Usuń” przy wybranym Użytkowniku. Następuję automatyczne usunięcie go z bazy danych. |
| Alternatywne przepływy zdarzeń | 4a) Wydawnictwo nie musi zmieniać danych. Może  w każdej chwili opuścić stronę zarządzania kontem Użytkownika co automatycznie anuluje wprowadzone zmiany. |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela 3 Dokumentacja przypadku użycia „Przeglądanie oraz czytanie publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Przeglądanie oraz czytanie tekstów** |
| Numer | 2 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Autor, Recenzent, Redaktor, Wydawnictwo, Użytkownik |
| Krótki opis | Przegląd i wybór tekstu do przeczytania |
| Warunki wstępne | Użytkownik przegląda listę artykułów |
| Warunki końcowe | Użytkownik po wybraniu opracowania może je przeczytać |
| Główny przepływ zdarzeń | 1. Użytkownik na stronie głównej ma wyświetloną listę artykułów. 2. Po wybraniu artykułu Użytkownik jest przekierowany do strony, z tekstem który może przeczytać. |
| Alternatywne przepływy zdarzeń | Brak |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Użytkownik może w każdej chwili opuścić stronę |

Tabela Dokumentacja przypadku użycia „Kontakt"

|  |  |
| --- | --- |
| **Nazwa** | **Kontakt** |
| Numer | 3 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Użytkownik |
| Krótki opis | Skontaktowanie się z czasopismem naukowym poprzez formularz kontakty |
| Warunki wstępne | Użytkownik wybiera opcje „Kontakt” |
| Warunki końcowe | Użytkownikowi udało się wysłać wiadomość |
| Główny przepływ zdarzeń | 1. Użytkownik na stronie głównej na górze wybiera opcje „Kontakt”. 2. Po wybraniu tej opcji Użytkownik jest przekierowany do strony, gdzie znajduje się formularz kontaktowy. 3. Użytkownik wypełnia formularz wpisując dane w wymagane pola, czyli „Imię i nazwisko”, „E-mail”, „Temat” oraz treść wiadomości. 4. Po wypełnieniu formularza użytkownik wysyła wiadomość poprzez przycisk „Wyślij wiadomość” 5. Po naciśnięciu przycisku pokaże się komunikat, że wiadomość została wysłana |
| Alternatywne przepływy zdarzeń | 1a) Użytkownik nie musi wybierać tej opcji i może w każdej chwili opuścić stronę „Kontakt”  2a) Użytkownik nie musi wypełniać formularza i może  w każdej chwili opuścić stronę „Kontakt”.  4a) błąd przy wysyłaniu – Blednę dane lub brak wypełnionego pola oznacza |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Użytkownik może w każdej chwili opuścić stronę |

Tabela 5 Dokumentacja przypadku użycia „Dodanie publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Dodanie artykułu** |
| Numer | 4 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Autor |
| Krótki opis | Dodawanie publikacji przez Autora |
| Warunki wstępne | Autor jest zalogowany |
| Warunki końcowe | Artykuł został dodany przez Autora i przekierowany do Redaktora |
| Główny przepływ zdarzeń | 1. Autor po zalogowaniu na stronie głównej wybiera opcje „Dodaj artykuł”. 2. Po wybraniu tej opcji Autor jest przekierowany do strony, gdzie ma możliwość dodania tekstu w postaci pliku PDF. Jest także możliwość dodania tytułu oraz opisu artykułu 3. Autor dodaje plik poprzez przycisk „Dodaj plik”. 4. Po dodaniu pliku Autor klika przycisk „Wyślij artykuł, który chcesz opublikować”. |
| Alternatywne przepływy zdarzeń | 4a) Kiedy zostanie dodany plik, który nie jest plikiem PDF zostanie wyświetlony komunikat o niezgodności pliku.  4b) Autor zanim zatwierdzi „Wyślij artykuł, który chcesz opublikować” może dodać nowy plik co spowoduje zastąpienie starego pliku nowym. |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela Dokumentacja przypadku użycia „Aktualizacja publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Aktualizacja artykułu** |
| Numer | 5 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Autor |
| Krótki opis | Aktualizacja artykułu przez Autora |
| Warunki wstępne | Autor jest zalogowany |
| Warunki końcowe | Artykuł został zaktualizowany przez Autora  i przekierowany do Redaktora |
| Główny przepływ zdarzeń | 1. Autor po zalogowaniu na stronie głównej wybiera opcje „Artykuły”. 2. Po wybraniu tej opcji Autor jest przekierowany do strony, gdzie jest wyświetlana lista artykułów, które Autor dodał. 3. Chcąc aktualizować tekst wybiera opcję „Edytuj” przy wybranym artykule. 4. Autor zmienia tytuł artykułu, jego opis oraz status na „Wysłany do Redaktora” oraz odpowiada na uwagi dotyczące tego tekstu oraz dodaje plik poprzez przycisk „Dodaj artykuł”. 5. Po dokonaniu zmian Autor zatwierdza je przyciskiem „Zaktualizuj artykuł, który chcesz publikować” i tekst zostaje przekierowana do Redaktora. |
| Alternatywne przepływy zdarzeń | 4a) Kiedy zostanie dodany plik, który nie jest plikiem PDF zostanie wyświetlony komunikat o niezgodności pliku.  4b) Autor zanim zatwierdzi przyciskiem „Zaktualizuj artykuł, który chcesz opublikować” może dodać nowy plik co spowoduje zastąpienie starego pliku nowym. |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela Dokumentacja przypadku użycia „Usunięcie publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Usunięcie artykułu** |
| Numer | 6 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Autor, Wydawnictwo |
| Krótki opis | Usunięcie publikacji przez Autora lub Wydawnictwo |
| Warunki wstępne | Autor lub Wydawnictwo są zalogowani |
| Warunki końcowe | Artykuł został usunięty przez Autora lub Wydawnictwo |
| Główny przepływ zdarzeń | 1. Autor lub Wydawnictwo po zalogowaniu na stronie głównej wybiera opcje „Artykuły”. 2. Po wybraniu tej opcji Autor lub Wydawnictwo jest przekierowane do strony, gdzie ma wgląd w listę artykułów. 3. Chcąc usunąć tekst Autor lub Wydawnictwo zatwierdza przyciskiem „Usuń” przy wybranym artykule. 4. Po naciśnięciu tego przycisku artykuł zostaje usunięty  z bazy danych. |
| Alternatywne przepływy zdarzeń | Brak |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela 8 Dokumentacja przypadku użycia „Ocenienie publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Ocenienie artykułu** |
| Numer | 7 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Recenzent |
| Krótki opis | Ocena artykułu przesłanego przez Autora oraz przekierowanie jego do Redaktora. |
| Warunki wstępne | Recenzent jest zalogowany |
| Warunki końcowe | Praca została oceniona oraz wysłana do Redaktora |
| Główny przepływ zdarzeń | 1. Recenzent po zalogowaniu się na stronie głównej wybiera opcje „Artykuły” 2. Po wciśnięciu tego przycisku Recenzent jest przenoszony do strony, gdzie znajdują się wszystkie artykuły ze statusem „Wysłane do recenzji”. Istnieje możliwość przejrzenia tekstów i pobrania ich na komputer. 3. Przystępuje do czytania w celu oceny artykułu pod kątem znalezienia błędów. 4. Chcąc ocenić tekst Recenzent wybiera opcje „Recenzja artykułu” przy wybranym tekście. Recenzent ma przed sobą dane w postaci tytułu, opisu oraz aktualnego statusu. Ma możliwość wypisania uwag  do pracy. 5. Kiedy błędy zostały znalezione Recenzent wypisuje uwagi w formularzu, zmienia status na „Wysłany  do Redaktora” i przekierowuje artykuł do Redaktora poprzez przycisk „Zaktualizuj status. |
| Alternatywne przepływy zdarzeń | 5a) Kiedy błędy nie zostały wykryte Recenzent pisze informacje w formularzu, że błędy nie zostały znalezione, zmienia status na „Wysłany do Redaktora” i przekierowuje artykuł do Redaktora poprzez naciśniecie przycisku „Zaktualizuj status”. |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela 9 Dokumentacja przypadku użycia „Wysłanie publikacji do wydawnictwa”

|  |  |
| --- | --- |
| **Nazwa** | **Redagowanie artykułu** |
| 1 | 2 |
| Numer | 8 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Redaktor |
| Krótki opis | Usługa pozwalająca na redagowaniu artykułu przez Redaktora oraz wysłanie go do Wydawnictwa w celu publikacji. |
| Warunki wstępne | Redaktor jest zalogowany |
| Warunki końcowe | Redaktor zredagował tekst |
| Główny przepływ zdarzeń | 1. Redaktor po zalogowaniu na stronie głównej wybiera opcję „Artykuły”. 2. Po kliknięciu w opcję jest przenoszony do strony, gdzie znajdują się wszystkie artykuły przesłane do tej pory przez Autora. 3. Po kliknięciu w wybrany tekst poprzez link „PLIK” Redaktor ma przed sobą wyświetloną zawartość pliku. Ma też możliwość pobrania pliku na komputer. 4. Po wybraniu opcji „Zredaguj” ma przed sobą dane  w postaci tytułu, opisu oraz aktualnego statusu. Ma możliwość wypisania uwag do artykułu. 5. Po wybraniu opcji „Zredaguj” Redaktor zostaje przekierowany do strony, gdzie ma przed sobą dane  w postaci tytułu, opisu oraz aktualnego statusu  i wypisanych uwag od Recenzenta lub informacji o braku uwag. 6. Chcąc przekierować tekst w celu zrecenzowania zmienia status na „Wysłany do recenzji” i pisze informację, aby Recenzent ocenił artykuł. 7. Po otrzymaniu odpowiedzi od Recenzenta może  ją przeczytać wtedy, kiedy naciśnie przycisk „Zredaguj” albo „Zobacz uwagi i informacje”. 8. Po przeczytaniu uwag od Recenzenta i po przejrzeniu przesłanego artykułu Redaktor może zdecydować, aby:   a) opublikować artykuł, b) zgodzić się na publikację pod warunkiem poprawy uwag, c) nie wyrazić zgody na publikację.   1. Po podjęciu decyzji, Redaktor wybiera opcję „Zredaguj” a następnie zmienia status na jeden z wybranych: „Do publikacji”, „Wysłany do recenzji”, „Możliwość publikacji pod warunkiem”, „Brak zgody na opublikowanie". Następnie pisze informacje do Autora o podjętej decyzji oraz o jej powodach. Po wprowadzeniu informacji Redaktor przekierowuje artykuł do Autora poprzez przycisk „Zaktualizuj Status” |
| 1 | 2 |
| Alternatywne przepływy zdarzeń | Brak |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

Tabela 10 Dokumentacja przypadku użycia „Opublikowanie publikacji”

|  |  |
| --- | --- |
| **Nazwa** | **Opublikowanie artykułu** |
| Numer | 9 |
| Twórca | Dominik Lewczyński |
| Poziom ważności | Wysoki |
| Typ przypadku użycia | Ogólny, niezbędny |
| Aktorzy | Wydawnictwo |
| Krótki opis | Opublikowanie artykułu |
| Warunki wstępne | Otrzymanie artykułu od Redaktora |
| Warunki końcowe | Opublikowanie artykułu |
| Główny przepływ zdarzeń | 1. Wydawnictwo przechodzi do strony „Artykuły” gdzie ma widoczne teksty ze statusem „Do publikacji” 2. Wydawnictwo wybiera artykuły z listy. Ma możliwość ich przejrzenia. 3. Poprzez zmianę statusu na „Opublikowany” oraz potwierdzeniu praca zostaje opublikowana na stronie głównej |
| Alternatywne przepływy zdarzeń | Brak |
| Wyjątki wymagania | Brak |
| Specjalne wymagania | Brak |
| Notatki i kwestie | Brak |

# **Implementacja**

W niniejszym rozdziale opisałem szczegóły dotyczące implementacji aplikacji internetowej czasopisma naukowego. W kolejnych podrozdziałach przedstawiłem jaka była strategia realizacji projektu, z jakich głównych modułów składa się aplikacja, jakie wzorce wykorzystano w projekcie, jaki jest przebieg informacji a także szczegółowo opisałem technologie zastosowane na frontendzie i backendzie oraz jaką bazę danych wykorzystałem i w jaki sposób przeprowadziłem testy. Całą aplikację napisałem za pomocą języka programowania JavaScript oraz frameworków opartych o ten język.

## **Strategia realizacji projektu**

Implementowanie aplikacji internetowej czasopisma naukowego realizowałem zgodnie z następującą strategią:

* **Projektowanie bazy danych**: W pierwszej kolejności została zaprojektowana baza danych MySQL, która będzie przechowywać dane dotyczące artykułów, Autorów, Recenzentów oraz inne dane potrzebne do działania aplikacji. Szczegóły odnośnie do tego co wykorzystałem do stworzenia bazy danych znajduje się w folderze [**7.7 Bazy danych**](#_Bazy_danych.),
* **Tworzenie backendu**: Stworzyłem backend za pomocą frameworka Express.js który odpowiada ze komunikację z bazą danych, przetwarzanie danych oraz udostępnianie danych do części frontendowej. Więcej szczegółów o tym co użyłem do stworzenia frontendu znajduje się w podrozdziale   
  [**7.6 Backend**](#_Backend),
* **Testowanie backendu**: Po stworzeniu backendu zacząłem testowanie poprzez dodawanie nowych użytkowników oraz dodawanie przykładowych artykułów, aby upewnić się, że wszystko działa prawidłowo oraz że integracja z bazą danych istnieje. Więcej szczegółów o tym czego użyłem do testowania znajduje się w podrozdziale [**7.8 Testy**](#_Testy).
* **Tworzenie frontendu**: Opracowałem frontend aplikacji przy użyciu React.js ze zdefiniowanym szablonem Redux. Do wyglądu aplikacji zastosowałem framework CSS o nazwie Bulma, natomiast do wysyłania wiadomości e-mail wykorzystałem EmailJS. Więcej szczegółów o tym co użyłem do stworzenia frontendu znajduje się w podrozdziale [**7.5 Frontend**](#_Frontend),
* **Integracja**. Następnie frontend i backend połączyłem ze sobą, aby umożliwić komunikacje między nimi,
* **Testowanie aplikacji**: Na koniec testowałem aplikację korzystając z niej, aby upewnić się, że całość działa prawidłowo i nie występują problemy   
  w czasie użytkowania.

## **Główne moduły aplikacji**

Aplikacja składa się z kilku głównych modułów:

* **Panel administracyjny**: Umożliwia Wydawnictwu na zarzadzanie artykułami,   
  a także zapewnia zarządzenie kontami Autorów, Recenzentów oraz Redaktorów.
* **System składania prac**: Pozwala Autorom na przesyłanie artykułów oraz pozwala na śledzenie ich statusu.
* **System recenzowania**: Pozwala Recenzentom na przeglądanie   
  i recenzowanie artykułów oraz na komunikację z Redaktorem.
* **System redagowania**: Pozwala Redaktorom na przeglądanie oraz na redagowanie przesłanych artykułów oraz umożliwia komunikacje z Autorami oraz Recenzentami w celu ich oceny.
* **Witryna publiczna**: Ten moduł zawiera stronę internetowa, która jest dostępna dla wszystkich użytkowników i umożliwia przeglądanie artykułów oraz informacji o czasopiśmie.

## **Wzorce wykorzystane w projekcie**

W aplikacji zastosowano następujące wzorce projektowe:

* **Wzorzec Model-View-Controller (MVC)**: Wzorzec ten pozwolił na oddzielenie logiki biznesowej od logiki prezentacji. W aplikacji, backend odpowiada za logikę biznesową, natomiast frontend zajmuje się prezentacją danych.
* **Wzorzec Store**: Wzorzec ten został wykorzystany w połączeniu z React   
  i Redux. Umożliwia on centralizację stanu aplikacji i ułatwia jego zarządzanie.
* **Wzorzec Observer**: Wzorzec ten został użyty między modułami Systemem redagowania, Systemem recenzowania oraz Systemem składania prac. Pozwala on na informowanie o zmianach w statusie artykułu i umożliwia komunikację między Autorem, a Redaktorem i Recenzentem.
* **Wzorzec Singleton**: Wzorzec ten został użyty do klasy odpowiedzialnej za połączenie z bazą danych, zapewnił on, że w całej aplikacji jest tylko jedna instancja tej klasy, co umożliwia łatwiejsze zarządzanie połączeniem z bazą danych.

## **Przebieg informacji w systemie**

Przebieg informacji w aplikacji jest następujący:

1. Autor przesyła artykuł do systemu przez **System składania prac**.
2. Artykuł jest przetwarzany przez Redaktora w **systemie redagowania** i jest wysyłany do recenzentów do przeglądu.
3. Recenzenci przeglądają artykuł i wystawiają opinie za pośrednictwem modułu **System recenzowania**.
4. Redaktor przegląda opinie Recenzentów i podejmuje decyzję o publikacji artykułu lub jego odrzuceniu.
5. Kiedy podjęta zostaje decyzja o publikacji artykułu. Redaktor poprzez **System redagowania** przesyła artykuł do Wydawnictwa do **Panelu administracyjnego.** Jeśli podjęto decyzję o odrzuceniu pracy Redaktor informuje Autora, że tekst nie zostanie opublikowany.
6. Wydawnictwo poprzez **Panel administracyjny** publikuje artykuł
7. Artykuł opublikowany, jest dostępny dla czytelników na stronie internetowej   
   w **Witrynie publicznej**.
8. Autor otrzymuje informację o statusie artykułu za pośrednictwem **Systemu składania prac**.
9. W ten sposób, aplikacja umożliwia przepływ informacji między Autorami, Recenzentami i Redaktorami, a także udostępnia artykuły dla czytelników.

## **Frontend**

Frontend aplikacji został stworzony z użyciem biblioteki React. Jest ona oparta   
o język JavaScript. Komenda „**create-react-app**” z argumentem **template redux** pozwoliła na stworzenie aplikacji w React z predefiniowanym szablonem Redux. Zawiera on niezbędne pliki i konfiguracje do implementacji stanu aplikacji za pomocą biblioteki Redux

Biblioteki wymienione poniżej były najbardziej przydatne w czasie tworzenia interfejsu aplikacji oraz do integracji z backendem.

* + **React-router-dom** – pozwala na implementację nawigacji aplikacji.
  + **Axios** – służy do komunikacji z API.
  + **Moment** – pozwala na wyświetlenie daty w odpowiednim formacie.
  + **Bulma** – framework CSS daje gotowe komponenty, które pozwoliły na stworzenie solidnego i responsywnego interfejsu.
  + **@Redux.js/tollkit** – ułatwia tworzenie aplikacji z wykorzystaniem biblioteki Redux. Zawiera on skróty dla najczęściej stosowanych operacji, takich jak konfiguracja i tworzenie store'a, tworzenie akcji i reduktorów oraz ułatwia pracę z typami danych.
  + **Emailjs** – pozwala na wysyłanie wiadomości e-mail z poziomu aplikacji.

## **Backend**

Backend aplikacji został stworzony z użyciem frameworka Express.js. Wykorzystałem dodatkowe biblioteki, które pomogły w napisaniu backendu:

* + **Dotenv** – pozwala na ładowanie pliku .env z konfiguracją do projektu.
  + **Cors** – umożliwia obsługę mechanizmu CORS (Cross-Origin Resource Sharing) w aplikacji express. Jest to potrzebne, jeśli aplikacja korzysta   
    z zasobów z innych domen.
  + **Mysql2** – zezwala na połączenie z bazą danych MySQL za pomocą Node.js.
  + **Sequelize** – segment ORM (Object-Relational Mapping) dla Node.js, który ułatwia pracę z bazami danych, bez konieczności pisania zapytań SQL.
  + **Connect-session-sequeli**ze – przyzwala na trzymanie sesji użytkownika   
    w bazie danych za pomocą ORM Sequelize.
  + **Express-session** – umożliwia obsługę sesji użytkownika w aplikacji express.
  + **Express-fileupload** – ułatwia przesyłanie plików w aplikacji express.
  + **Argon2** – daje możliwość hashowanie haseł za pomocą algorytmu Argon2, który jest uważany za jeden z bezpieczniejszych sposobów hashowania.

Każdy z tych modułów pozwolił na stworzenie działającego zaplecza, które łączy się   
z bazą danych. Dają one możliwość tworzenia konta, logowania się na nie oraz umożliwiają przesyłania pliku do bazy danych.

## **Bazy danych**.

Do przechowywania danych Użytkowników oraz publikacji została wykorzystana baza danych MySQL. Aby ją uruchomić zastosowałem pakiet XAMPP co pozwoliło na proste stworzenie bazy danych potrzebnej do stworzenia aplikacji.

## **Testy**

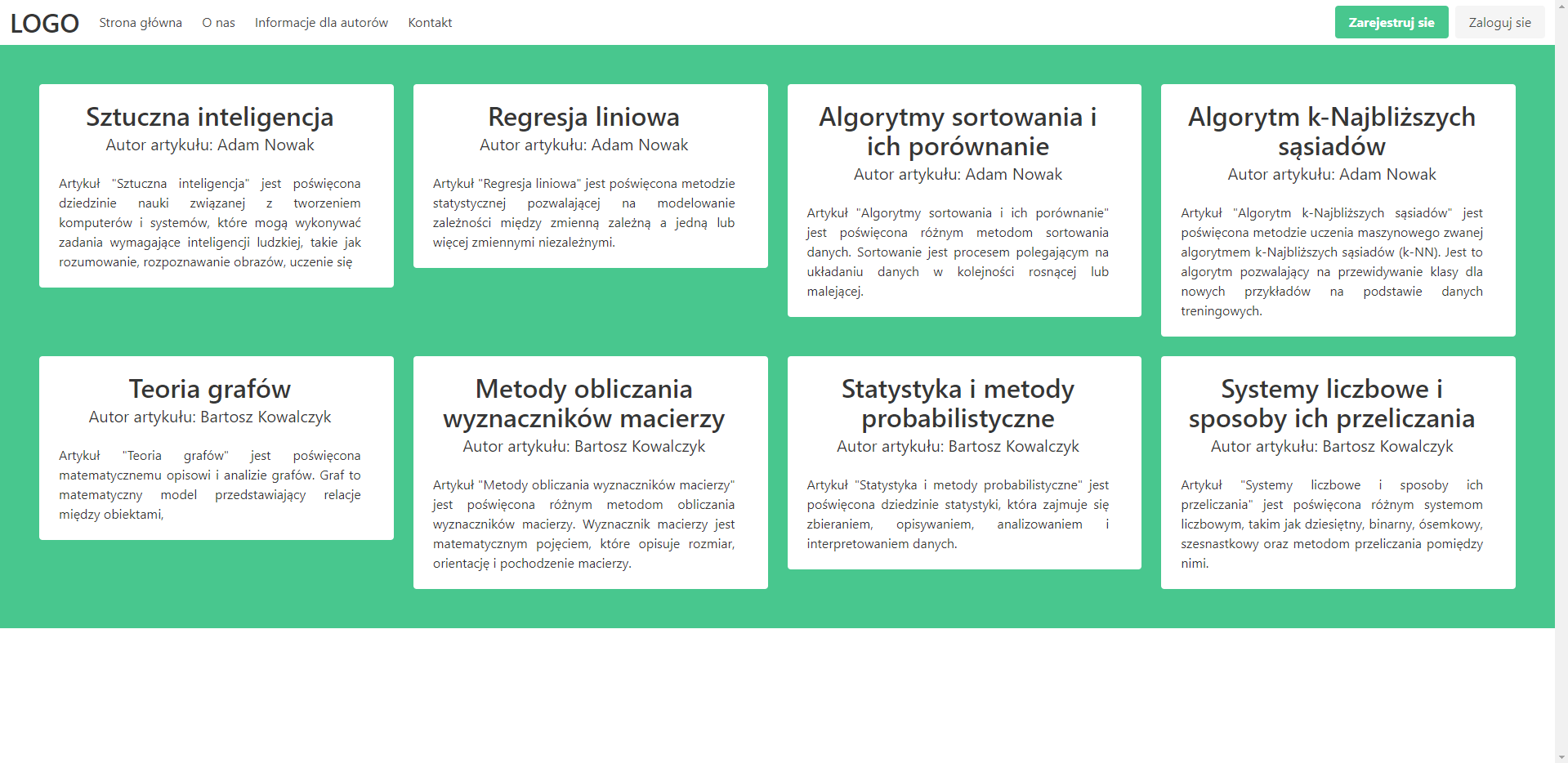
Testowanie aplikacji przeprowadziłem za pomocą narzędzia Postman. Realizowałem je na różnych etapach tworzenia aplikacji, aby upewnić się, że każda   
z funkcjonalności działa poprawnie. Dodatkowo testowanie wykonałem również poprzez używanie aplikacji przez Użytkowników końcowych.

# **Instrukcja użytkownika**

W niniejszym rozdziale widnieje instrukcja użytkowania z **Aplikacji internetowej czasopisma naukowego**. Na opisanych zrzutach ekranu przedstawiony został interfejs graficzny.

## **Interfejs strony głównej**

Po uruchomieniu aplikacji w wybranej przeglądarce internetowej pierwszym widokiem jakie użytkownik zobaczy to jest widok strony głównej, który wygląda następująco:



To jest strona główna aplikacji, gdzie są wyświetlane opublikowane teksty. Przy każdym artykule widać tytuł pracy, autora oraz opis tego o czym jest dana praca. Po wybraniu artykułu zostajemy przekierowani do strony, gdzie jest wyświetlany jego tytuł, nazwa autora, krótki opis tekstu oraz sama praca. Przykład jest na poniższym zdjęciu:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Na górze strony znajduje się nawigacja, za pomocą której zostaniemy przekierowani do odpowiednich stron:

* „O nas” – strona, w której jest zawarte informacje o czasopiśmie naukowym

Obraz zawierający tekst

Opis wygenerowany automatycznie

* „Informacje dla autorów” – strona, z wymaganiami, które musza przestrzegać autorzy by ich teksty zostały opublikowane.

Obraz zawierający tekst

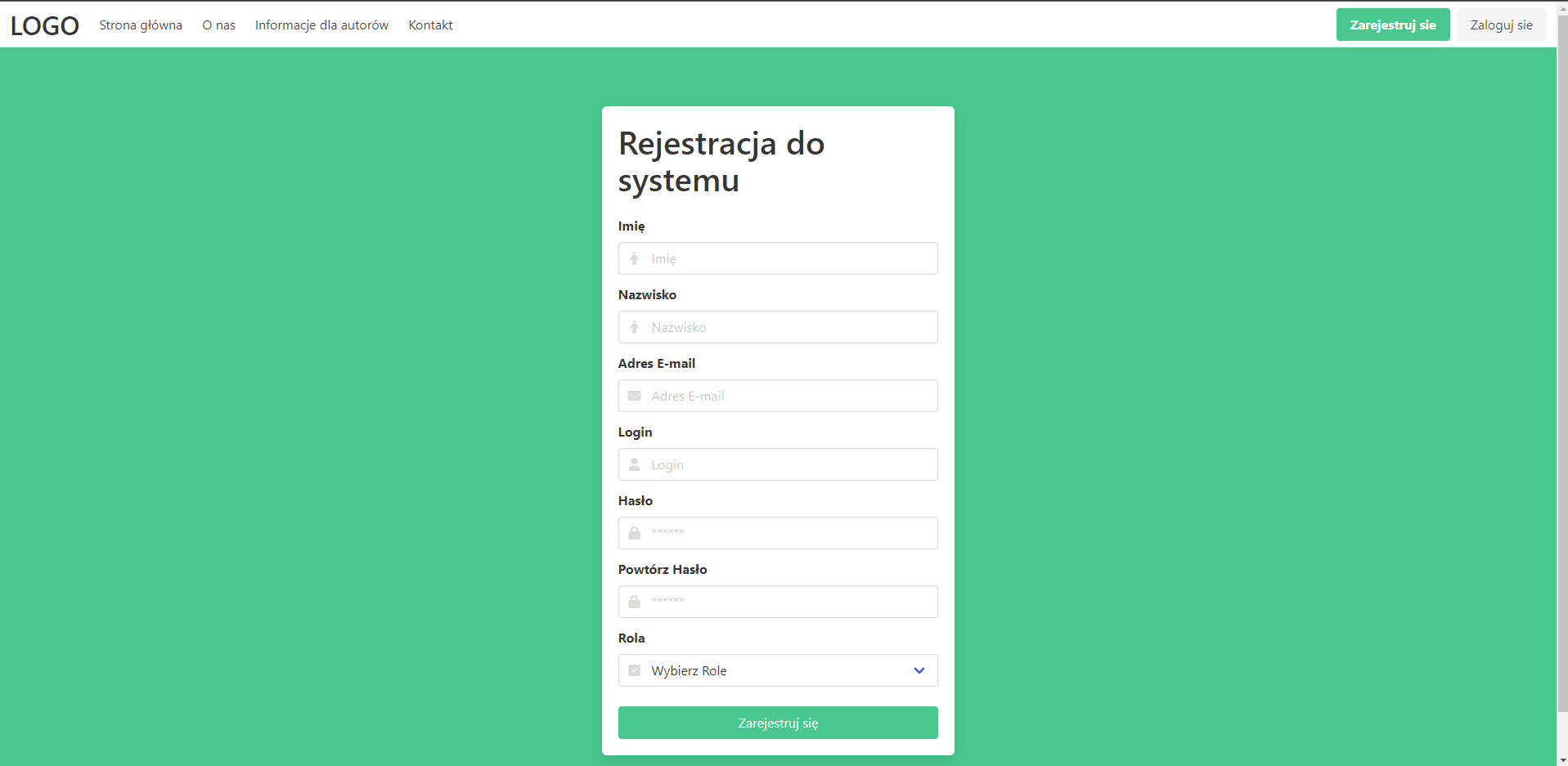
Opis wygenerowany automatycznie

* „Kontakt” – strona, na której umieszczony jest formularz kontaktowy. Za jego pomocą można wysłać wiadomość np. można zapytać się o szczegółowe informacje w sprawie publikacji.

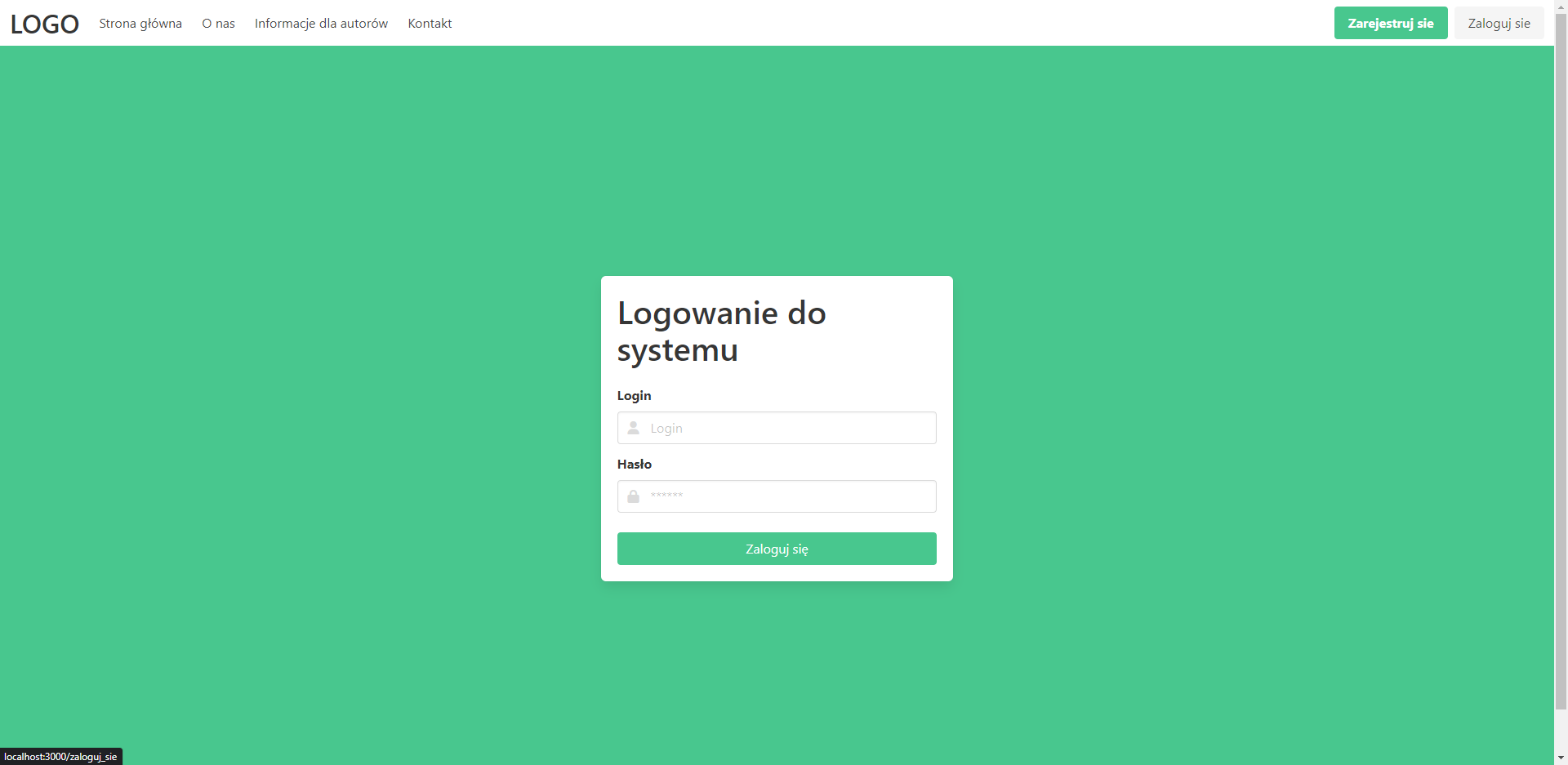
Obraz zawierający tekst

Opis wygenerowany automatycznie

* „Zarejestruj się” – strona, na której jest formularz do rejestracji nowego konta

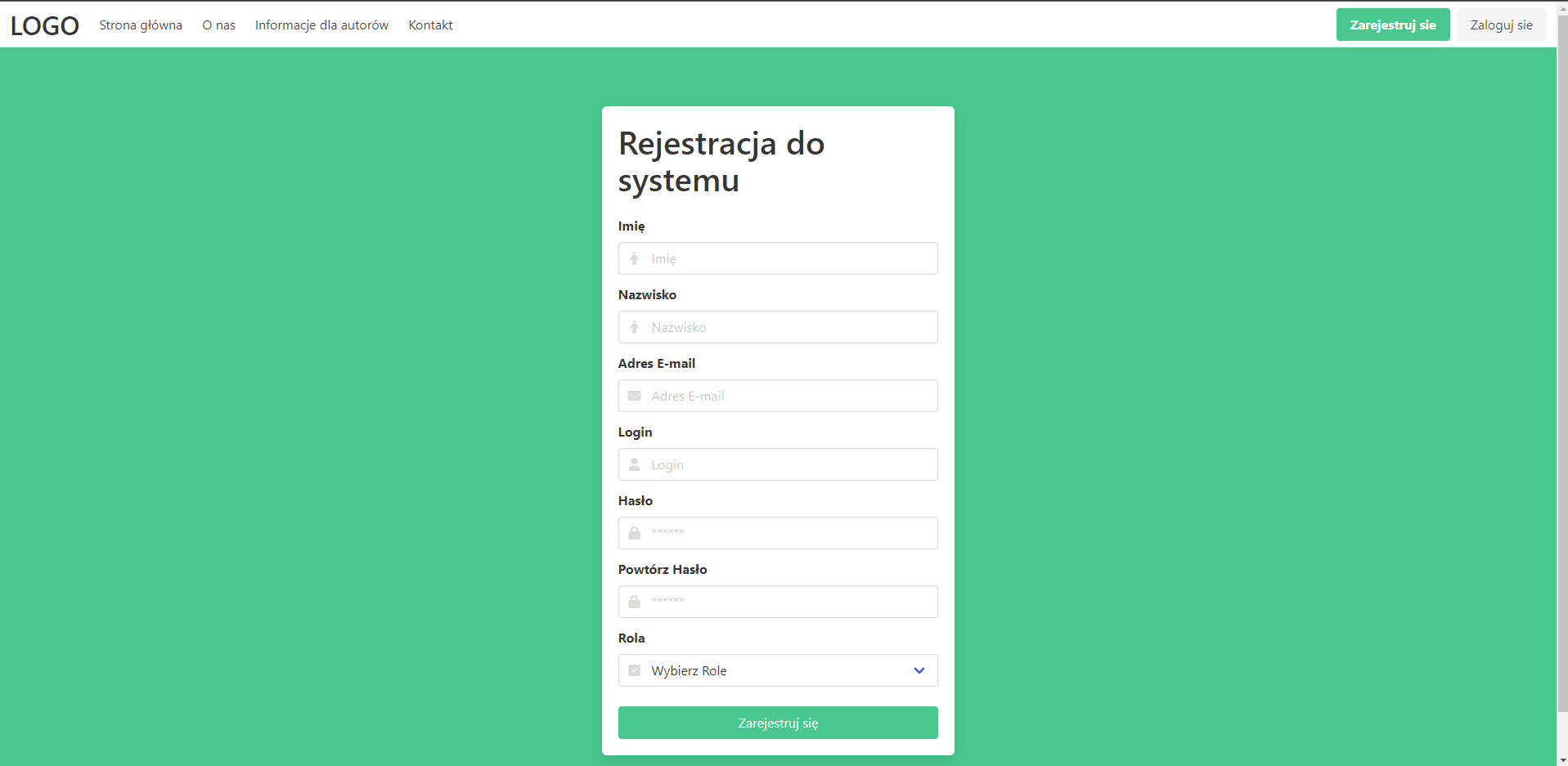


* „Zaloguj się” – strona, z formularzem do logowania się do swojego konta



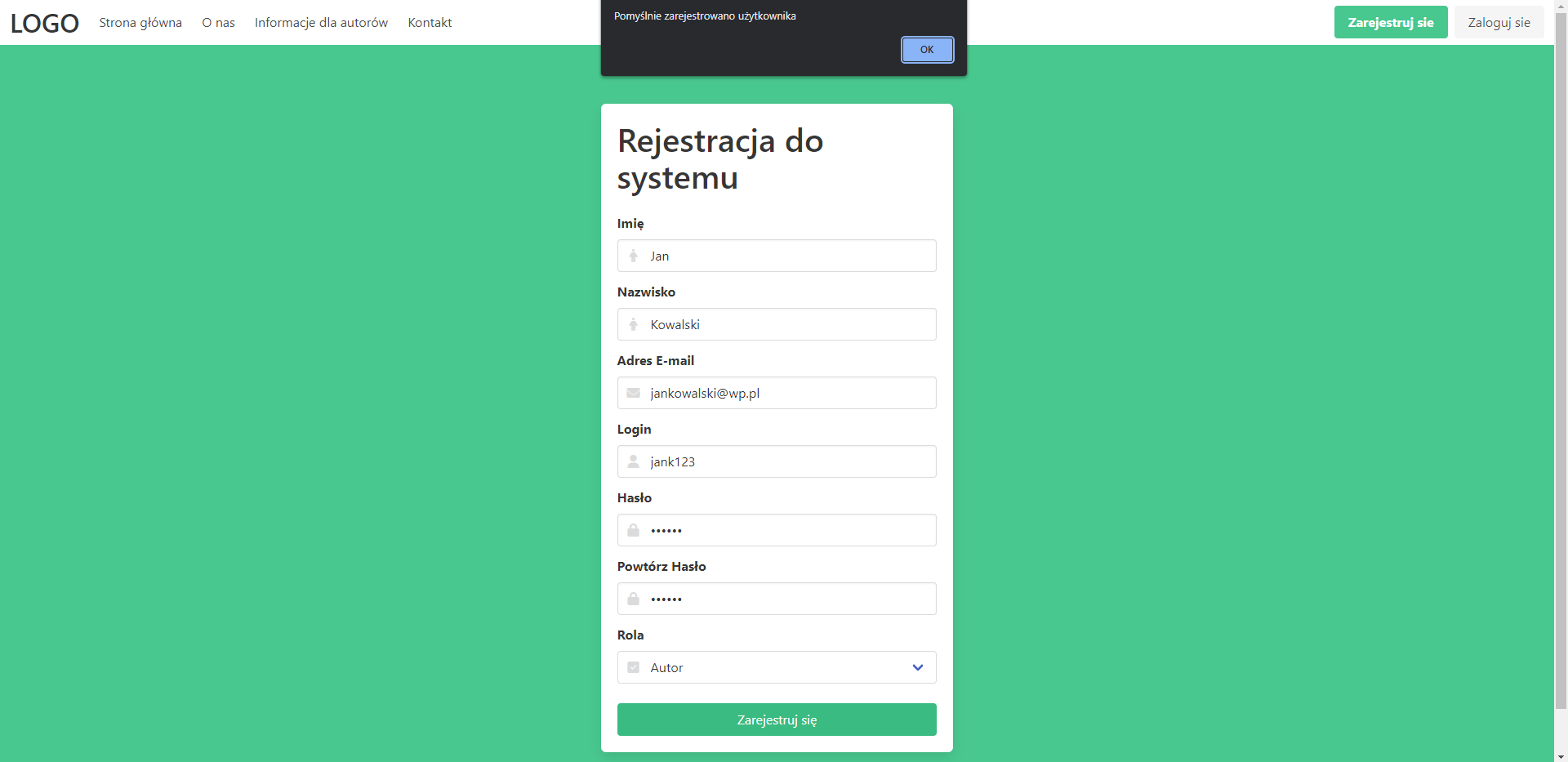
## **Konto użytkownika**

Jeżeli Użytkownik zdecyduje się na utworzenie konta na przykład konto „Autor” do dodawania publikacji, Użytkownik wybiera na stronie głównej przycisk „Zarejestruj się”. Zostanie wówczas otwarty formularz do rejestracji konta.

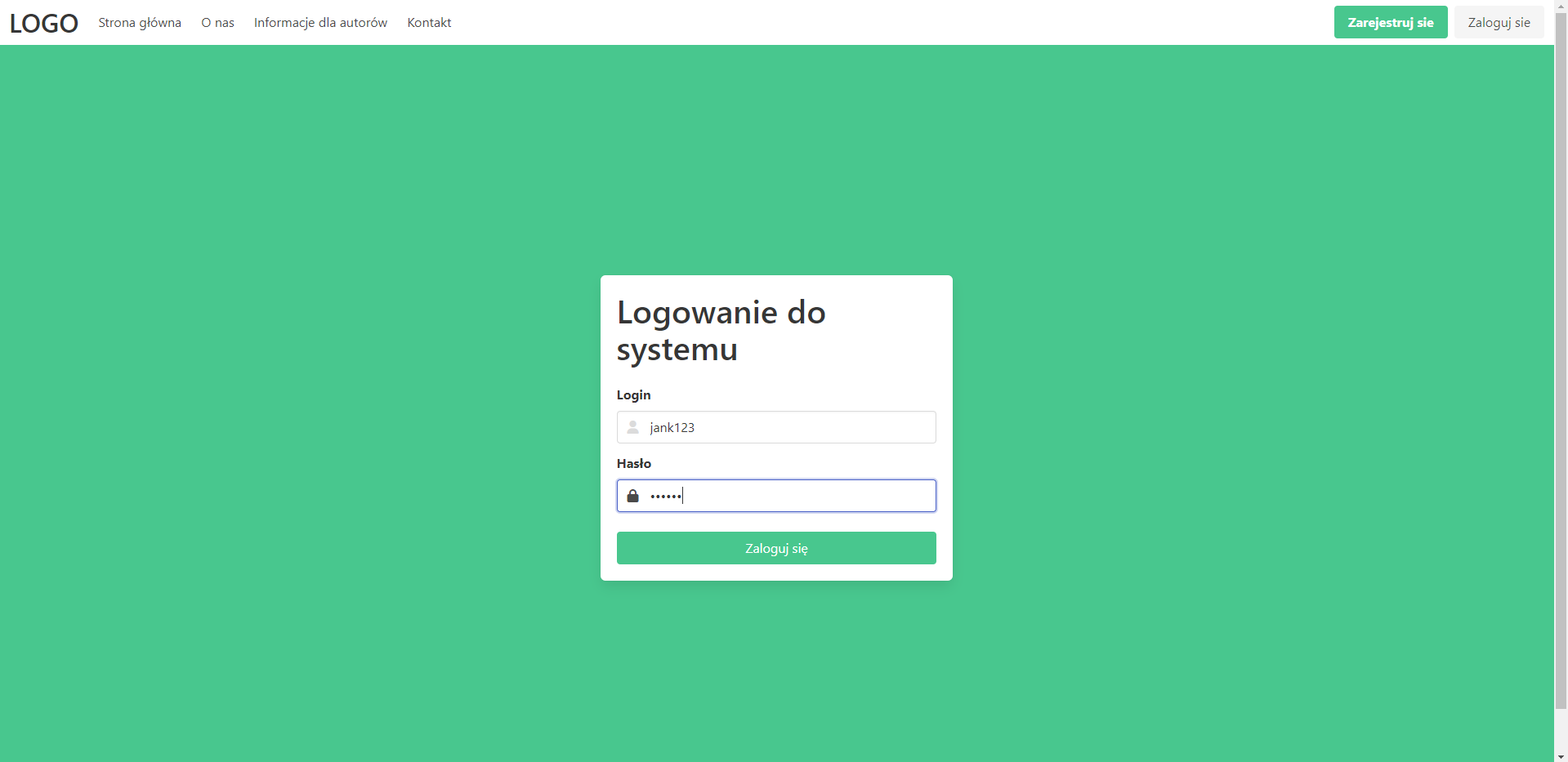


Następnie w formularzu tym wypełnia on niezbędne dane oraz wybiera rolę jaka będzie przypisana do konta. Do wyboru jest Autor, Recenzent oraz Redaktor.   
Na poniższym przykładzie pokazałem utworzenie konta „Autor” dla Użytkownika Jan Kowalski

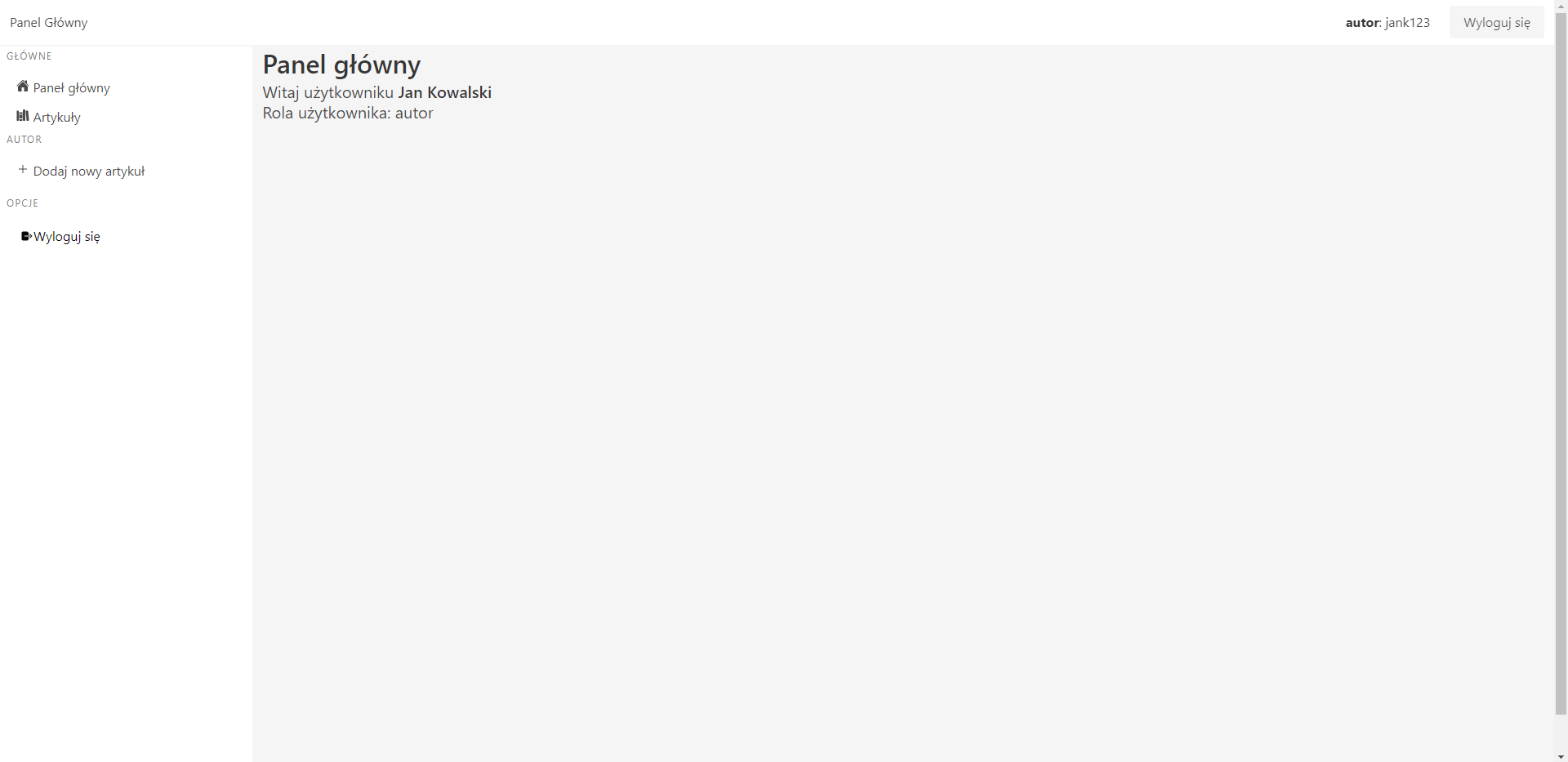
.



Kiedy konto zostanie utworzone Użytkownik Jan Kowalski wybiera na górze   
z prawej strony przycisk „Zaloguj się” i zostaje przekierowany do strony z formularzem do logowania na konto.



Po naciśnięciu przycisku „Zaloguj się” na formularzu, Użytkownik zostaje zalogowany na swoje konto.



W ten oto sposób konto Użytkownika zostało utworzone oraz zalogowano się   
na nie. Na powyższym zdjęciu znajduje się główny panel konta Użytkownika.   
Po lewej stronie jest nawigacja, która zmienia się w zależności od roli jaką wybrano   
w czasie rejestracji. W tym przykładzie znajduje się link do strony Panelu Głównego   
i link do strony z listą publikacji. Dla nowo utworzonego konta „Autor” strona   
z publikacjami wygląda następująco:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Na górze po prawej oraz po lewej znajduje się przycisk do wylogowania się ze swojego konta. Naciśniecie przycisku spowoduje wylogowanie się ze swojego konta oraz powrót do strony głównej.

## **Dodawanie artykułu**

Jeżeli Autor będzie chciał dodać nowy artykuł to przechodzimy do strony Artykuły albo wybieramy opcje po lewej stronie. Dla przykładu wykorzystane będzie konta użytkownika Jan Kowalski.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po wybraniu opcji zostajemy przekierowani do strony z formularzem, gdzie dodajemy nowy tekst.

Obraz zawierający tekst

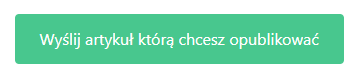
Opis wygenerowany automatycznie

Następnie Autor wpisuje tytuł pracy, jej krótki opis oraz dodaje plik PDF.

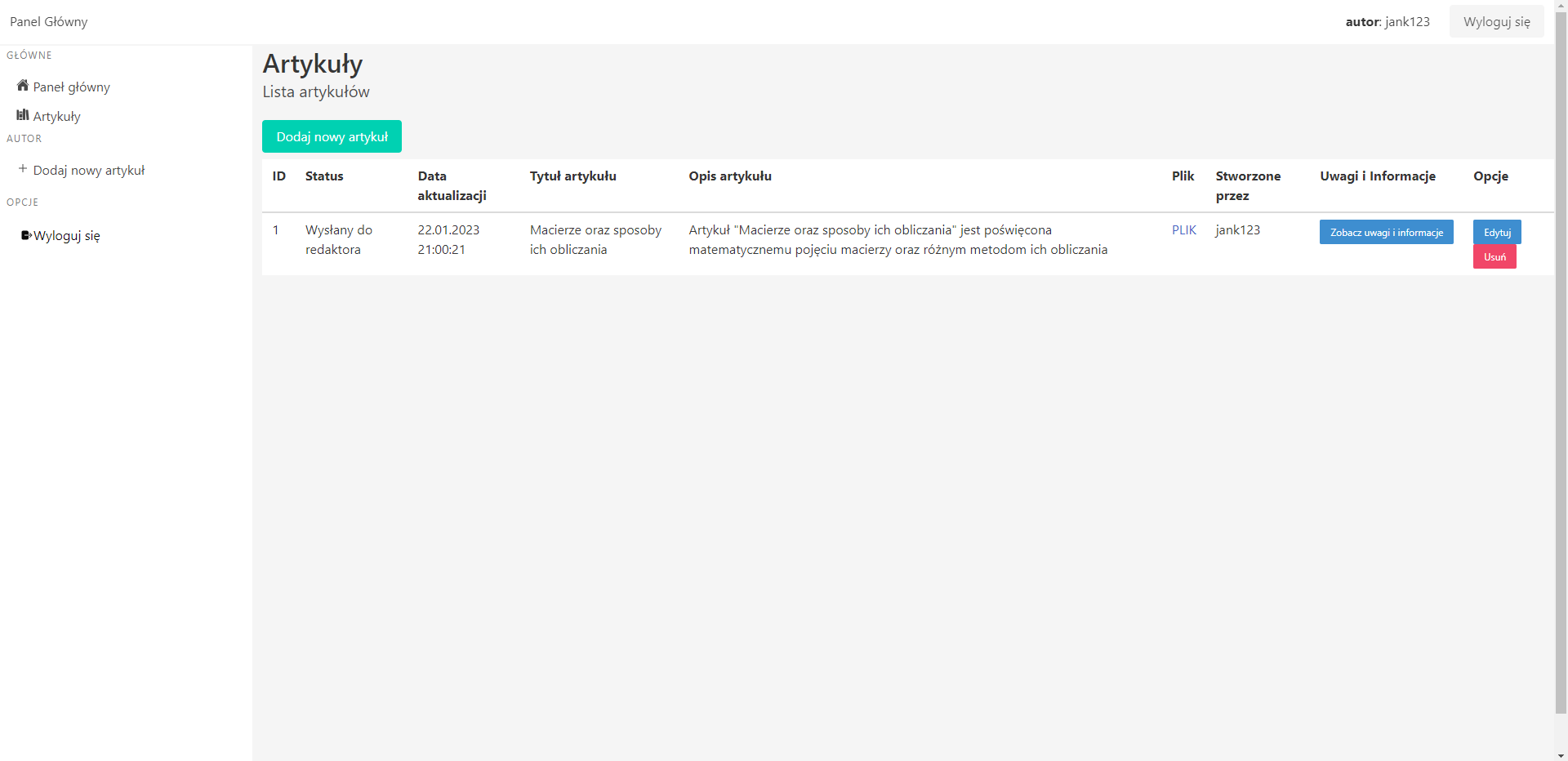
Obraz zawierający tekst

Opis wygenerowany automatycznie

Po dodaniu pliku istnieje możliwość podejrzenia jego zawartości, aby upewnić się, że dodany plik jest prawidłowy. Po wypełnieniu formularza naciskany zostaje przycisk „Wyślij artykuł którą chcesz opublikować”

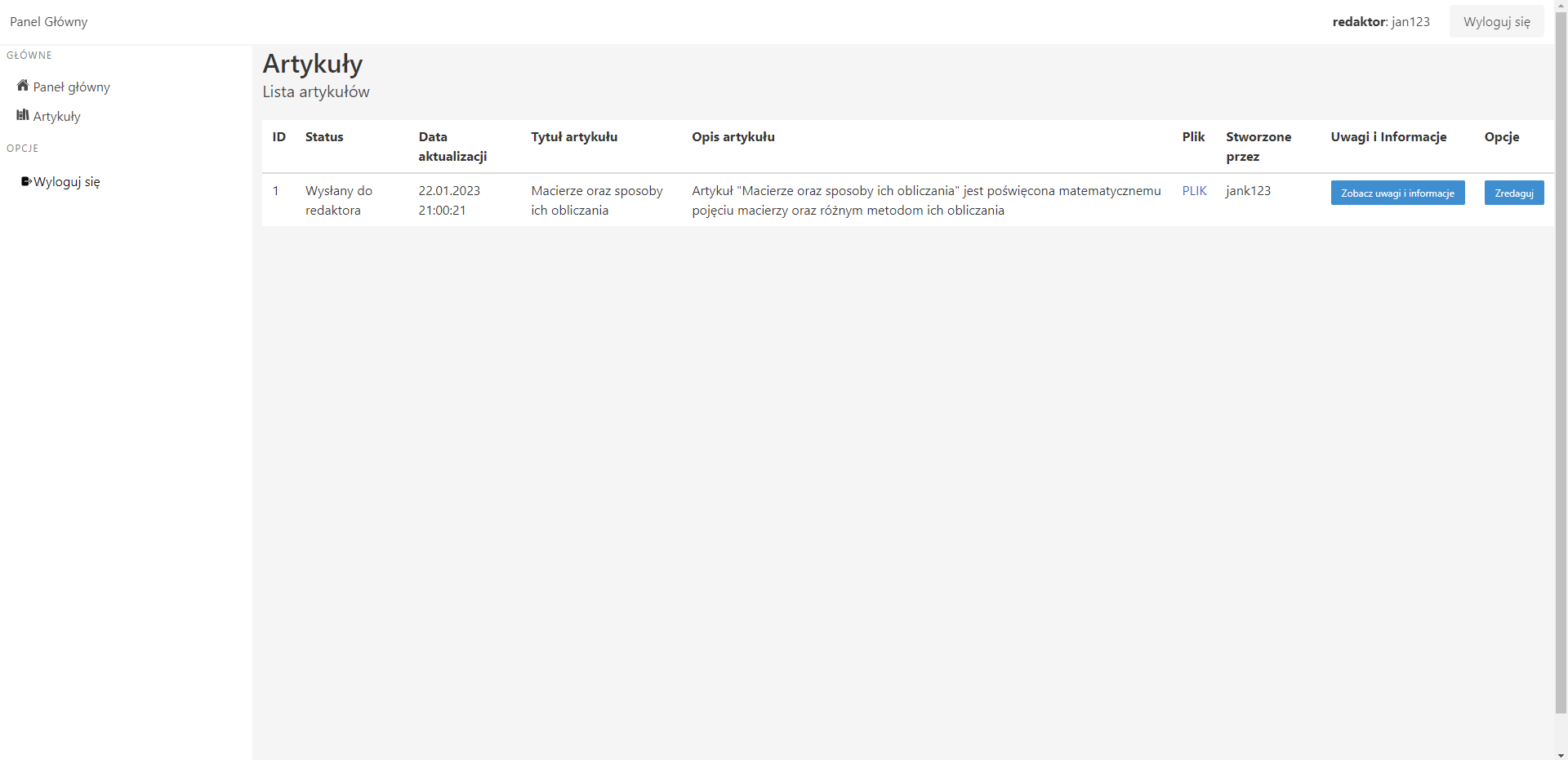


Po jego naciśnięciu zostajemy przekierowani do strony z listą tekstów autora, gdzie widoczna jest dodana praca ze statusem „Wysłany do Redaktora”. Artykuł zostanie przesłany do Redaktora celem dalszego procedowania, które zostanie opisane   
w następnym podrozdziale.



## **Przygotowanie do publikacji**

Po wysłaniu tekstu przez Autora do Redaktora praca zostanie poddana procesowi przygotowania do publikacji. W tym przykładzie utworzyłem konta: Redaktora oraz Recenzenta (przyjmujemy, że praca zostanie opublikowana po poprawieniu uwag). Po zalogowaniu się na konto Redaktora przechodzimy do strony z listą tekstów, które Redaktor widzi w momencie, kiedy ich status jest „Wysłany do Redaktora” oraz „Wysłany do recenzji”.



Redaktor może zobaczyć status artykułu, kiedy został wysłany, jego tytuł, opis   
i kto jest jej autorem oraz ma możliwość otwarcia pliku PDF po naciśnięciu   
opcji „PLIK”. Ma także możliwość zobaczyć uwagi:

1. od Recenzenta, które będą pomocne w podjęciu decyzji,
2. informacje, które napisał Redaktor,
3. odpowiedź od Autora w momencie, kiedy Autor musi poprawić swój tekst.

Żeby Redaktor mógł na przykład wysłać tekst do Recenzenta w celu jego oceny wybiera opcje „Zredaguj” i zostaje przekierowany do formularza z danymi artykułu, (tytuł, opis oraz aktualny jego status). Ma możliwość zmiany statusu (Do wyboru ma: „Do publikacji”, „Wysłany do recenzji”, „Możliwość publikacji pod warunkiem”, „Brak zgody na opublikowanie”) oraz napisania prośby do Recenzenta, aby ocenił on pracę albo o poinformowaniu Autora o podjętej decyzji oraz o jej powodach. W tym przykładzie wysyłamy publikacje do Recenzenta:

Obraz zawierający tekst

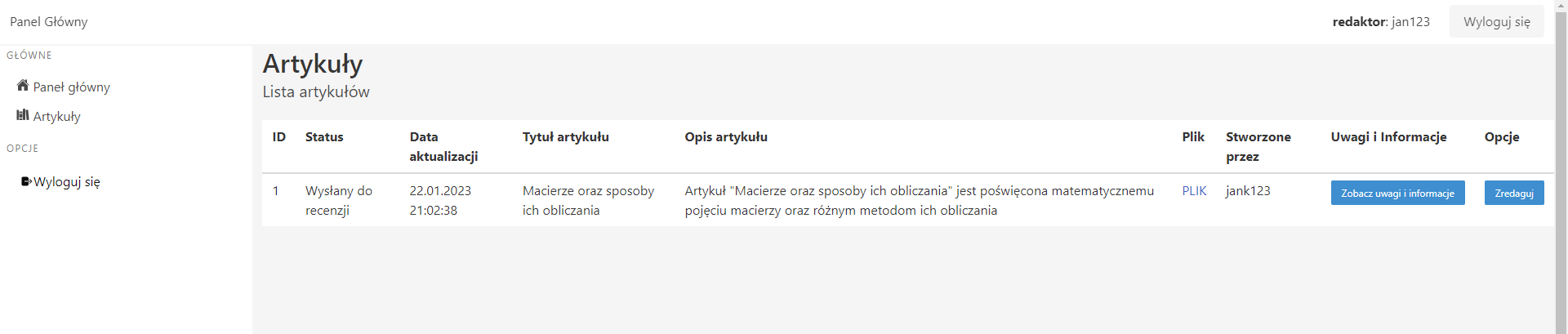
Opis wygenerowany automatycznie

Po wypełnieniu formularza naciskamy przycisk „Zaktualizuj status”

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po naciśnięciu przycisku Redaktor zostaje przekierowany do strony z listą artykułów, ale już widać, że tekst ma zaktualizowany status oraz zaktualizowaną datę.



Następnie logujemy się na konto Recenzenta, gdzie potem przechodzimy do strony z listą tekstów. Recenzent widzi wyłącznie artykuły, których status jest „Wysłany do recenzji”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

W celu oceny artykułu Recenzent wybiera opcje „Recenzja artykułu” i zostaje przekierowany do formularza, gdzie są dane o artykule, czyli tytuł, opis oraz aktualny status oraz ma możliwość zmiany statusu na „Wysłany do Redaktora” oraz wypisania uwag albo napisania informacji o braku uwag.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po wypełnieniu formularza Recenzent zatwierdza to przyciskiem „Zaktualizuj status” i zostaje przekierowany na stronę z listą prac. Jak można zauważyć na poniższym zdjęciu praca zniknęła, ponieważ Recenzent widzi teksty, kiedy ich status to „Wysłany do recenzji”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Wracamy do Redaktora i ponownie przechodzimy do strony z listą artykułów. Status jest zaktualizowany i można zobaczyć uwagi od Recenzenta np. po wybraniu opcji „Zobacz uwagi i informacje” lub „Zredaguj”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po zobaczeniu uwag od Recenzenta oraz przenalizowaniu tekstu, Redaktor podejmuje decyzję o opublikowaniu artykułu, ale pod warunkiem ze Autor poprawi uwagi od Recenzenta. W tym celu wybiera opcje „Zredaguj”. Zmienia status na   
„Możliwość opublikowania pod warunkiem” i pisze informacje do Autora.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po zaktualizowaniu statusu można zalogować się do konta Autora. Przechodźmy do strony z listą prac, gdzie Autor cały czas może monitorować co się dzieje z jego tekstem.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po spojrzeniu na status, Autor wybiera opcję „Edytuj” i przechodzi do formularza, gdzie może zobaczyć informacje od Redaktora i uwagi od Recenzenta. Aktualizuje tytuł, opis, status oraz dodaje nowy plik i pisze odpowiedź, że uwagi wszystkie zostały uwzględnione i klika przycisk „Zaktualizuj artykuł, który chcesz opublikować”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po zaktualizowaniu artykułu Autor widzi dane swojej pracy w tym jej status. Artykuł został wysłany do Redaktora.

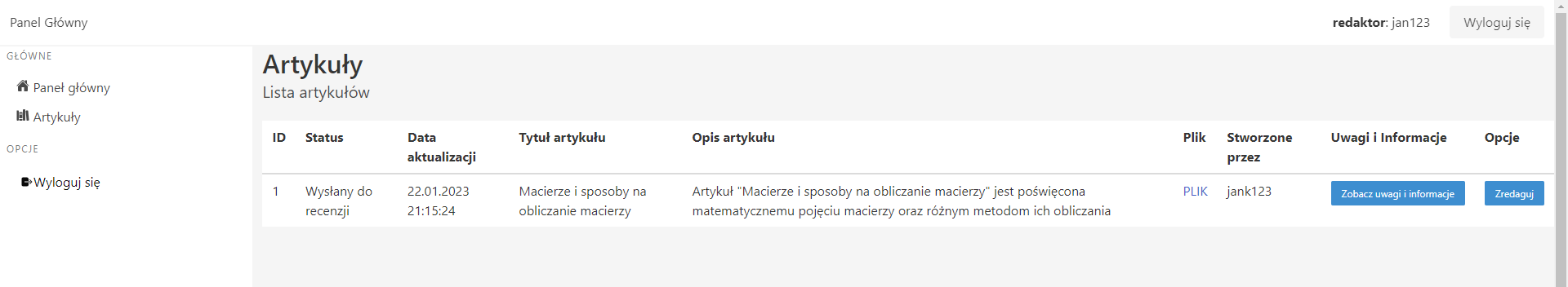
Obraz zawierający tekst

Opis wygenerowany automatycznie

Następnie logujemy się do konta Redaktora. Przechodzi on do strony z listą artykułów oraz wybiera opcję „Zredaguj” i wysyła pracę do Recenzenta w celu ponownej oceny.

Obraz zawierający tekst

Opis wygenerowany automatycznie



Po wykonaniu tego kroku logujemy się na konto Recenzenta. Przechodzimy   
do strony z listą tekstów, wybieramy opcje „Recenzja publikacji”. Recenzent pisze   
o pozytywnej ocenie.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Obraz zawierający tekst

Opis wygenerowany automatycznie

Kiedy Recenzent wykona ten krok logujemy się ponownie do Redaktora. Po przejrzeniu publikacji oraz informacji od Recenzenta o pozytywnej ocenie Redaktor wybiera opcje „Zredaguj” i zmienia status na „Do publikacji”. Pisze informację dla Autora, że spełnione zostały wszystkie wymagania i artykuł zostanie opublikowany.   
W tym przypadku praca zostaje wysłana do Wydawnictwa w celu opublikowania na stronie głównej.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Gdyby jednak się okazało, że tekst nie spełnia wymagań by go opublikować, wtedy Redaktor zmienia status na „Brak zgody na opublikowanie”. Przesyła informację Autorowi z uzasadnieniem, dlaczego jego artykuł nie zostanie opublikowany.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po pozytywnym zredagowaniu i przesłaniu artykułu do Wydawnictwa logujemy się na konto Wydawnictwa, które jest w systemie. Przechodzimy do strony z listą prac, gdzie Wydawnictwo widzi artykuły, których status jest „Do publikacji” oraz „Opublikowany”

Obraz zawierający tekst

Opis wygenerowany automatycznie

Na samym dole listy znajduje się artykuł, który Wydawnictwo musi opublikować, gdyż spełnia on kryteria określone przez Wydawnictwo. W tym celu wybiera opcję „Opublikuj” i zmienia status na „Opublikowany” i zatwierdza wybór przyciskiem „Zaktualizuj Status”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Po zaktualizowaniu statusu zostajemy przekierowani z powrotem do strony   
z listą artykułów i ostatni na liście tekst ma zmieniony status na „Opublikowany”.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Kiedy wylogujemy się z konta Wydawnictwa wrócimy do strony głównej, na której jest opublikowany artykuł i można się z nim zapoznać.



Tak wygląda przedstawiony proces publikowania artykułów na stronie głównej za pomocą **Aplikacji internetowej czasopisma naukowego**.

# **Licencje**

Celem niniejszego rozdziału jest przestawienie środowisk, grafiki, bibliotek zewnętrznych oraz źródeł wiedzy programistycznej jakie zostały wykorzystane w celu stworzenia aplikacji.

## **Narzędzia**

* [**Visual Studio Code**](https://code.visualstudio.com/)– darmowy edytor kodu z kolorowaniem stworzony przez firmę Microsoft, o otwartym kodzie źródłowym na licencji MIT.
* [**Google Chrome**](https://www.google.com/intl/pl_pl/chrome/)– darmowa przeglądarka internetowa na urządzenia stacjonarne i mobilne, za pomocą, której wpisując hasło w wyszukiwarce Google, możemy wyszukiwać i poruszać się po stronach internetowych oraz przeglądać na nich treści w celu znalezienia potrzebnych dla nas informacji.
* [**Postman**](https://www.postman.com/)–darmowe narzędzie, które służy do testowania i debugowania interfejsów API (Application Programming Interface). Pozwala na tworzenie, edytowanie i wysyłanie żądań HTTP oraz przeglądanie odpowiedzi   
  z serwera. Może być używane do testowania różnych typów żądań, takich jak GET, POST, PUT i DELETE, a także do wysyłania danych w różnych formatach, takich jak JSON czy XML.
* [**Node.js**](https://nodejs.org/en/) – darmowa platforma programistyczna oparta na języku JavaScript, która pozwala na tworzenie aplikacji serwerowych. Dzięki Node.js, programiści mogą korzystać z jednego języka (JavaScript) na obu stronach aplikacji (frontend i backend) co pozwala na zwiększenie produktywności   
  i jednolity styl kodowania.
* [**Pakiet XAMPP**](https://www.apachefriends.org/pl/index.html) – darmowy, wieloplatformowy, zintegrowany pakiet, składający się z serwera Apache, bazy danych MySQL Istnieje możliwość pisania skryptów PHP oraz Perl. Wykorzystany został do utworzenia bazy danych w MySQL. Jego skrót oznacza X(Cross-platform) + Apache   
  + MariaDB + PHP + Perl

## **Grafiki**

* [**Font awesome**](https://fontawesome.com/) - źródło darmowych ikon w dowolnym formacie.

## **Biblioteki zewnętrzne**

* [**React.js**](https://reactjs.org/) – biblioteka języka programowania JavaScript, która umożliwia tworzenie interfejsów graficznych w aplikacjach internetowych.
* [**Express.js**](http://expressjs.com/)– back-endowy framework aplikacji internetowych do budowania interfejsów API RESTful z Node.js, wydanym jako darmowe oprogramowanie o otwartym kodzie źródłowym na licencji MIT. Przeznaczony jest do budowania aplikacji internetowych i API.
* [**Bulma**](https://bulma.io/) - bezpłatna platforma typu open source, która zapewnia gotowe do użycia komponenty frontendowe, które można łatwo łączyć   
  w celu tworzenia responsywnych interfejsów internetowych.
* [**EmailJS**](https://www.emailjs.com/) - biblioteka JavaScript, która pomaga wysyłać wiadomości e-mail przy użyciu technologii po stronie klienta. Zaletą EmailJS jest to, że nie są wymagane żadne serwery; wszystko, co należy zrobić, to podłączyć go do jednej z obsługiwanych usług e-mail, utworzyć szablon wiadomości e-mail   
  i użyć EmailJS do uruchomienia wiadomości e-mail.

## **Źródła wiedzy programistycznej**

* **StackOverflow** – internetowa społeczność programistyczna z całego świata. Opisane tam artykuły stanowiły pomoc przy rozwiązywaniu problemów programistycznych jak i przy nauce pisania czytelnego kodu.

# **Podsumowanie**

Przedstawiona praca inżynierska dotyczyła stworzenia aplikacji internetowej czasopisma naukowego. Celem projektu było stworzenie nowoczesnego   
i funkcjonalnego narzędzia, które umożliwiłoby użytkownikom łatwy dostęp do artykułów i możliwość i publikowania.

Aplikacja internetowa czasopisma naukowego została stworzona zgodnie   
z początkowymi założeniami. Wszystkie aspekty jej funkcjonalności realizującą początkowe założenia. Aplikacja pozwala na przeglądanie artykułów na stronie głównej, rejestrację oraz logowanie się do systemu. Umożliwia także dodanie aktualizacji oraz usuniecie pracy jako Autor, pozwala na recenzowanie jako Recenzent pozwala na redagowanie jako Redaktor oraz pozwala na publikowanie oraz usuniecie publikowanej pracy jako Wydawnictwo. Dane o użytkownikach są potrzebne do sprawdzenia kto jest kim oraz do logowania do systemu i są one aktualizowane przez Wydawnictwo

Aplikacja internetowa czasopisma może zostać rozbudowana na wiele sposobów. Można na przykład dodać na stronie głównej wyszukiwarkę oraz filtry które umożliwią łatwe wyszukanie opublikowanych artykułów. Można także umożliwić komentowanie tekstów czytelnikom. Istnieje także możliwość dodania opcji stworzenia konta dla zwykłego użytkownika, który nie jest Autorem, Recenzentem czy Redaktorem i to użytkownik mógłby zapisywać przeglądane teksty do swojej listy ulubionych artykułów. Istnieje wiele możliwości rozbudowy aplikacji.

W trakcie realizacji projektu zastosowałem najnowsze technologie po stronie frontendu, takie jak React.js z narzędziami Redux Toolkit oraz frameworkiem Bulma, oraz technologie po stronie backendu takie jak Express.js, co pozwoliło na stworzenie szybkiej i responsywnej w pełni działającej aplikacji. Przeprowadzone testy poprzez korzystanie z aplikacji oraz narzędzia Postman wykazały, że aplikacja jest stabilna   
i spełnia wszystkie wymagania funkcjonalne.

Proces stworzenia aplikacji internetowej czasopisma naukowego zakończył się sukcesem. Otrzymano nowoczesne i funkcjonalne narzędzie, które umożliwia użytkownikom łatwy dostęp do artykułów. Aplikacja ta jest stabilna i spełnia wszystkie założone wymagania funkcjonalne.

# **Bibliografia**

1. Jordan Walke., rok 2023, „React – A JavaScript library for building user interfaces”,   
   [**https://reactjs.org/docs/getting-started.html**](https://reactjs.org/docs/getting-started.html)
2. TJ Holowaychuk, rok 2023, „Express.js – Node.js web application framework”,  
   [**http://expressjs.com/**](http://expressjs.com/)
3. Jeremy Thomas, rok 2023, „Bulma – A modern CSS framework based on Flexbox”,  
   [**https://bulma.io/documentation/**](https://bulma.io/documentation/)
4. Andris Reinman, rok 2023, „EmailJS – Send your emails with a simple javascript API”, [**https://www.emailjs.com/docs/**](https://www.emailjs.com/docs/)