



**WYŻSZA SZKOŁA
INFORMATYKI i ZARZĄDZANIA**
z siedzibą w Rzeszowie

KOLEGIUM INFORMATYKI STOSOWANEJ

Kierunek: INFORMATYKA

Grupa: 6IIZ/2021-SP02

Michał Reguła
W66080, 6IIZ/2021-SP02

Aplikacja internetowa do szukania graczy do wspólnej gry w league of legends

Prowadzący: inż. Eryk Chrustek

Projekt

Rzeszów 2024

Spis treści

Wstęp	3
1 Technologie	4
1.1 Technologie	4
2 Wymagania funkcjonalne	5
2.1 Rejestracja użytkownika	5
2.2 Logowanie	5
2.3 Wylogowywanie	5
3 Wymagania Niefunkcjonalne	6
3.1 Bezpieczeństwo	6
3.2 Łatwość użytkowania	6
3.3 Wydajność	6
3.4 Skalowalność	6
3.5 Niezawodność	6
3.6 Kompatybilność	6
4 Widok aplikacji	7
5 Swagger	9
5.1 Rejestracja użytkownika	9
5.2 Odpowiedź serwera na rejestrację użytkownika	9
5.3 Logowanie użytkownika	10
5.4 Logowanie użytkownika	10
6 Struktura bazy danych	12

Wstęp

W dzisiejszych czasach, rozwój technologii internetowej i popularność gier online sprawiają, że społeczność graczy stale rośnie. Jednakże, wraz ze wzrostem liczby entuzjastów wirtualnych światów, pojawia się także wyzwanie skutecznego łączenia graczy o podobnych zainteresowaniach i umiejętnościach. W odpowiedzi na tę potrzebę powstała koncepcja aplikacji internetowej, która ma za zadanie ułatwić proces szukania partnerów do wspólnej gry online.

Niniejsza praca inżynierska skupia się na analizie istniejących problemów związanych z organizacją rozgrywek online oraz proponuje innowacyjne rozwiązanie w postaci aplikacji, która nie tylko eliminuje bariery w komunikacji między graczami, ale także dostarcza spersonalizowane narzędzia do efektywnego odnajdywania odpowiednich partnerów do gry.

Celem tego projektu jest stworzenie aplikacji, która nie tylko ułatwi proces szukania drużyny czy towarzyszy do gry online, ale także dostarczy użytkownikom nowoczesne narzędzia personalizacji, dzięki którym doświadczenie z rozgrywką stanie się bardziej satysfakcjonujące i dostosowane do indywidualnych preferencji.

W ramach wprowadzenia do problematyki, niniejsza praca rozpocznie od identyfikacji kluczowych wyzwań związanych ze znalezieniem partnerów do gry online, przeanalizuje aktualne trendy na rynku gier oraz zbadanie istniejących platform służących do organizacji rozgrywek. Następnie skoncentruje się na prezentacji koncepcji nowej aplikacji, opisując jej funkcje, założenia oraz technologie wykorzystane do jej stworzenia.

W miarę postępu pracy, czytelnik zostanie zapoznany z analizą rynkową, badaniem istniejących rozwiązań oraz szczegółowym opisem proponowanego systemu. W rezultacie, praca ta ma na celu przyczynienie się do doskonalenia procesu organizacji rozgrywek online, zwiększając satysfakcję graczy oraz umożliwiając bardziej efektywne korzystanie z dostępnych zasobów w obszarze gier online.

Rozdział 1

Technologie

1.1 Technologie

Projekt wykorzystuje PHP, HTML, CSS, MySQL ,XAMPP, phpMyAdmin, JavaScript, cURL

Rozdział 2

Wymagania funkcjonalne

2.1 Rejestracja użytkownika

Rejestracja użytkownika (User Registration) - Umożliwia nowym użytkownikom założenie konta w systemie poprzez podanie niezbędnych danych takich jak imię, nazwisko, adres e-mail, hasło, nazwa użytkownika w grze oraz tag.

2.2 Logowanie

Logowanie użytkownika (User Login) - Pozwala zarejestrowanym użytkownikom na zalogowanie się do systemu poprzez podanie adresu e-mail i hasła.

2.3 Wylogowywanie

Wylogowywanie użytkownika - Pozwala zalogowanemu użytkownikowi wylogowanie się za pomocą kliknięcia przycisku Logout.

Rozdział 3

Wymagania Niefunkcjonalne

3.1 Bezpieczeństwo

a) Bezpieczeństwo - Autoryzacja i autentykacja użytkowników za pomocą bezpiecznych metod takich jak JWT (JSON Web Tokens). Wdrożenie mechanizmów ochrony danych użytkowników, takich jak szyfrowanie haseł.

3.2 Łatwość użytkowania

Łatwość użytkowania - Intuicyjny interfejs użytkownika z przejrzystymi instrukcjami i nawigacją. Zapewnienie, aby wszystkie elementy interfejsu były łatwe w użyciu, z wyraźnymi komunikatami informacyjnymi i błędów.

3.3 Wydajność

Wydajność - Szybki czas odpowiedzi serwera, zoptymalizowane zapytania do bazy danych oraz efektywne zarządzanie zasobami systemu. Implementacja technik asynchronicznego przetwarzania operacji sieciowych w celu zwiększenia wydajności.

3.4 Skalowalność

Skalowalność - Architektura systemu powinna umożliwiać łatwe dodawanie nowych funkcji i modułów w przyszłości, co pozwoli na rozbudowę aplikacji bez konieczności znaczących modyfikacji istniejącego kodu.

3.5 Niezawodność

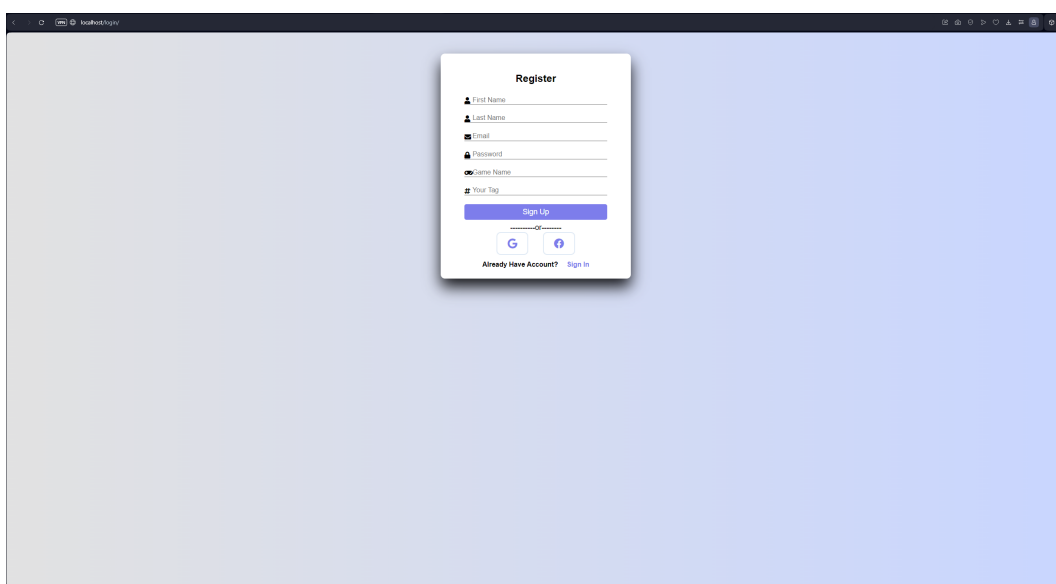
Niezawodność - System powinien być odporny na błędy i zapewniać ciągłość działania, z minimalnymi przestojami. Implementacja mechanizmów monitorowania i logowania błędów w celu szybkiego ich wykrywania i naprawiania.

3.6 Kompatybilność

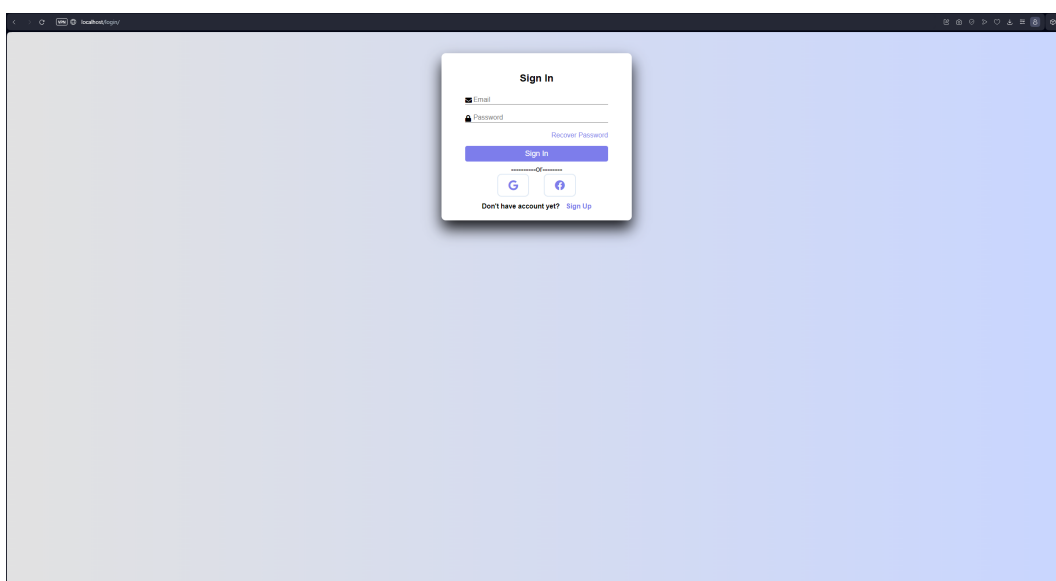
Kompatybilność - Aplikacja powinna być kompatybilna z różnymi przeglądarkami internetowymi oraz urządzeniami mobilnymi, zapewniając spójne doświadczenie użytkownika na wszystkich platformach.

Rozdział 4

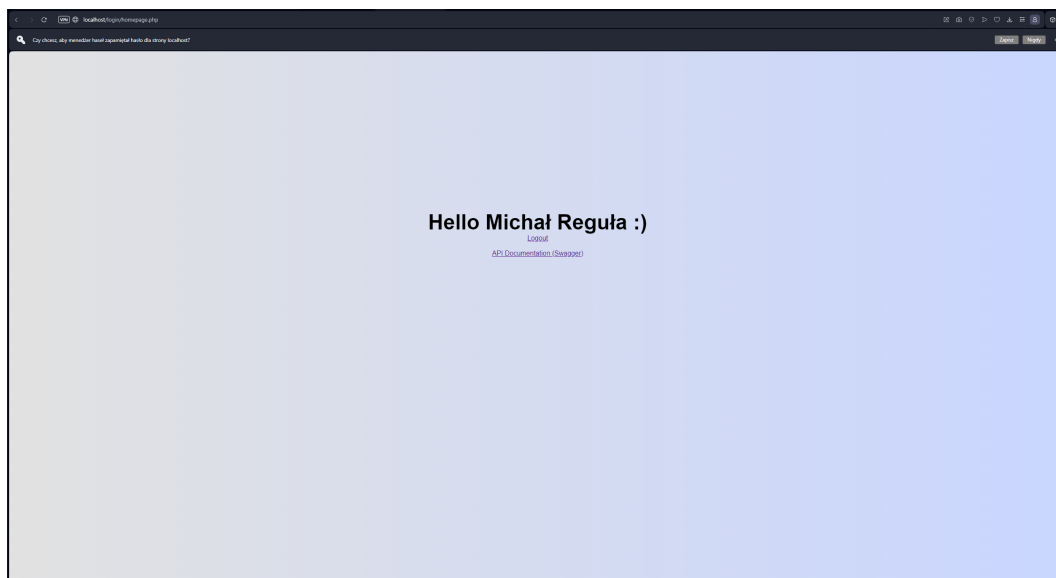
Widok aplikacji



Rysunek 4.1: Widok interfejsu rejestracji użytkownika



Rysunek 4.2: Widok interfejsu logowania użytkownika



Rysunek 4.3: Strona Główna Wylogowanie

Rozdział 5

Swagger

5.1 Rejestracja użytkownika

Na tym ekranie użytkownik wypełnia formularz rejestracyjny, podając swoje dane, takie jak imię, nazwisko, email, hasło, nazwę w grze oraz tag. Po wypełnieniu formularza i kliknięciu przycisku "Execute", użytkownik zostanie zarejestrowany.

The screenshot displays the Swagger UI for a REST API. The selected endpoint is `POST /register` with the description "Rejestracja użytkownika". The "Parameters" tab is active, showing "No parameters". The "Request body" tab is also active, showing a form with the following fields:

- `fName` (string, required):
- `lName` (string, required):
- `email` (string, required):
- `password` (string, required):
- `gameName` (string, required):
- `yourTag` (string, required):

Below the form is a large blue "Execute" button. At the bottom, the "Responses" section shows a table with one entry:

Code	Description	Links
200	Użytkownik zarejestrowany	No links

Rysunek 5.1: Rejestracja

5.2 Odpowiedź serwera na rejestrację użytkownika

Po przesłaniu formularza rejestracyjnego, serwer zwraca odpowiedź. W przypadku sukcesu zwracany jest kod 200 oraz wiadomość potwierdzająca, że użytkownik został zarejestrowany. W przypadku błędu (np. nieprawidłowy adres URL) zwracany jest kod 404 z odpowiednią wiadomością.

Responses

Curl

```
curl -X 'POST' \
  'http://localhost/register' \
  -H 'accept: */*' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'Name=RichaK5382&Name=ReguK5382&email=regula.m4002.p1&password=123456&gameName=Arctic&yourTag=1234'
```

Request URL

http://localhost/register

Server response

Code	Details
404	Error: Not Found

Response body

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL was not found on this server.</p>
<hr/>
<p>address:Apache/2.4.58 (Ubuntu) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80</p>
</body></html>
```

Response headers

```
connection: Keep-Alive
content-length: 295
content-type: text/html; charset=iso-8859-1
date: Sat, 29 Jun 2024 10:00:31 GMT
keep-alive: timeout=5,max=100
server: Apache/2.4.58 (Ubuntu) OpenSSL/3.1.3 PHP/8.2.12
```

Responses

Code	Description	Links
200	Użytkownik zarejestrowany	No links

Rysunek 5.2: Rejestracja Wynik

5.3 Logowanie użytkownika

Ekran logowania umożliwia użytkownikowi zalogowanie się do aplikacji poprzez podanie adresu email oraz hasła. Po wprowadzeniu danych i kliknięciu przycisku "Execute", serwer zwraca odpowiedź potwierdzającą poprawność danych logowania (kod 200) lub informację o błędnym emailu lub hasle (kod 401).

default

POST /register Rejestracja użytkownika

POST /login Logowanie użytkownika

Parameters

No parameters

Request body

application/x-www-form-urlencoded

email * required

string

regula.m@o2.pl

password * required

string

123456

Execute

Responses

Code	Description	Links
200	Użytkownik zalogowany	No links
401	Błędny email lub hasło	No links

Rysunek 5.3: Logowanie

5.4 Logowanie użytkownika

Po przejściu formularza logującego, serwer zwraca odpowiedź. W przypadku sukcesu użytkownik zostaje zalogowany (kod 200), w przypadku niepowodzenia zwraca (kod 401) informacje o niepoprawnym emailu lub hasle.

Responses

Curl

```
curl -X 'POST' \
  'http://localhost/login' \
  -H 'accept: */*' \
  -H 'Content-Type: application/x-www-form-urlencoded' \
  -d 'email=regula.804802.pl&password=12345'
```

Request URL

http://localhost/login

Server response

CodeDetails

200

Response body

```
<link rel="stylesheet" href="style.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
</head>
<body>
  <div class="container" id="signup" style="display:none;">
    <h1 class="form-title">Register</h1>
    <form method="post" action="register.php">
      <div class="input-group">
        <i class="fas fa-user"></i>
        <input type="text" name="fname" id="fname" placeholder="First Name" required>
        <label for="fname">First Name</label>
      </div>
      <div class="input-group">
        <i class="fas fa-user"></i>
        <input type="text" name="lname" id="lname" placeholder="Last Name" required>
        <label for="lname">Last Name</label>
      </div>
      <div class="input-group">
        <i class="fas fa-envelope"></i>
        <input type="email" name="email" id="email" placeholder="Email" required>
        <label for="email">Email</label>
      </div>
      <div class="input-group">
        <i class="fas fa-lock"></i>
        <input type="password" name="password" id="password" placeholder="Password" required>
        <label for="password">Password</label>
      </div>
      <div class="input-group">
```

Response headers

```
connection: Keep-Alive
content-length: 3705
content-type: text/html; charset=UTF-8
date: Sat, 29 Jun 2024 10:01:28 GMT
keep-alive: timeout=5,max=99
server: Apache/2.4.56 (Ubuntu) OpenSSL/3.1.3 PHP/8.2.12
x-powered-by: PHP/8.2.12
```

Responses

CodeDescriptionLinks

200

Użytkownik zalogowany

No links

401

Błędny email lub hasło








No links

Rysunek 5.4: Logowanie Wynik

11

Rozdział 6

Struktura bazy danych

	login users
	Id : int(11)
	firstName : varchar(50)
	lastName : varchar(50)
	email : varchar(50)
	password : varchar(50)
	gameName : varchar(50)
	yourTag : varchar(50)

Rysunek 6.1: Struktura Bazy