

# Platformy Programistyczne

Kierunek

*Informatyczne systemy automatyki*

Numer projektu

-

Imię, nazwisko, numer albumu

*Jakub Wojtala 272542, Mateusz Stochel 272524*

Data

*28.05.2025r.*

Grupa

*Grupa 1 wtorek 17:05 - 18:45*



## Sprawozdanie – aplikacja webowa

---



# **Spis treści**

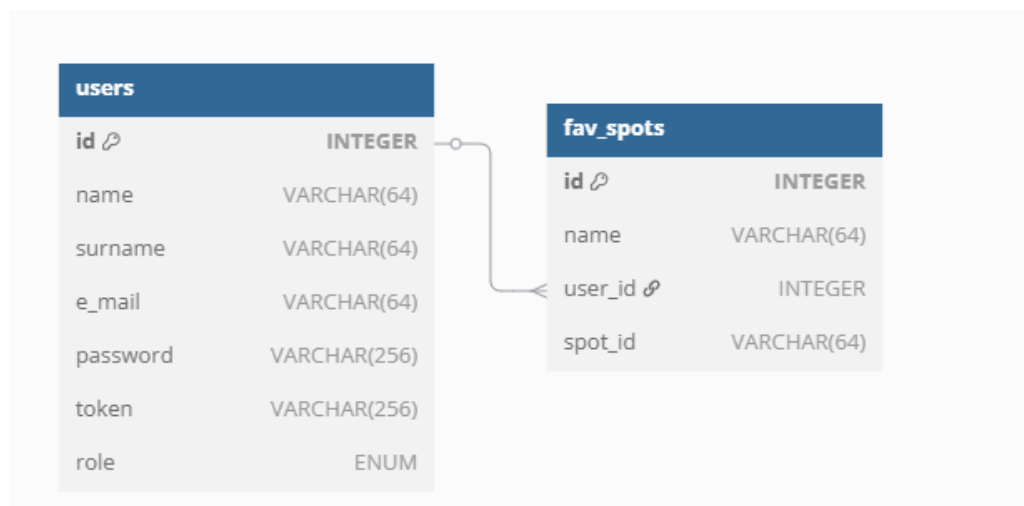
<b>1. Wstęp .....</b>	<b>2</b>
<b>2. Baza danych - MySQL .....</b>	<b>2</b>
<b>3. Backend – Java Spring Boot .....</b>	<b>3</b>
<b>4. Frontend – React .....</b>	<b>4</b>
<b>5. Konteneryzacja aplikacji – Docker .....</b>	<b>6</b>



# 1. Wstęp

Została zaprojektowana i stworzona aplikacja webowa umożliwiająca przeglądanie stacji rowerowych na całym świecie z wykorzystaniem zewnętrznego API <https://api.citybik.es/v2/networks>. Użytkownik ma możliwość wyboru konkretnego miasta, po czym wyświetlane są dostępne stacje rowerowe w wybranej lokalizacji. Dodatkowo istnieje opcja zapisywania wybranych stacji do ulubionych, które następnie dostępne są w dedykowanej zakładce. W aplikacji zaimplementowano również system rejestracji i logowania użytkowników. Projekt został zrealizowany z wykorzystaniem frameworka Java Spring Boot po stronie backendu oraz biblioteki React do budowy interfejsu użytkownika. Dane przechowywane są w relacyjnej bazie danych MySQL.

## 2. Baza danych - MySQL

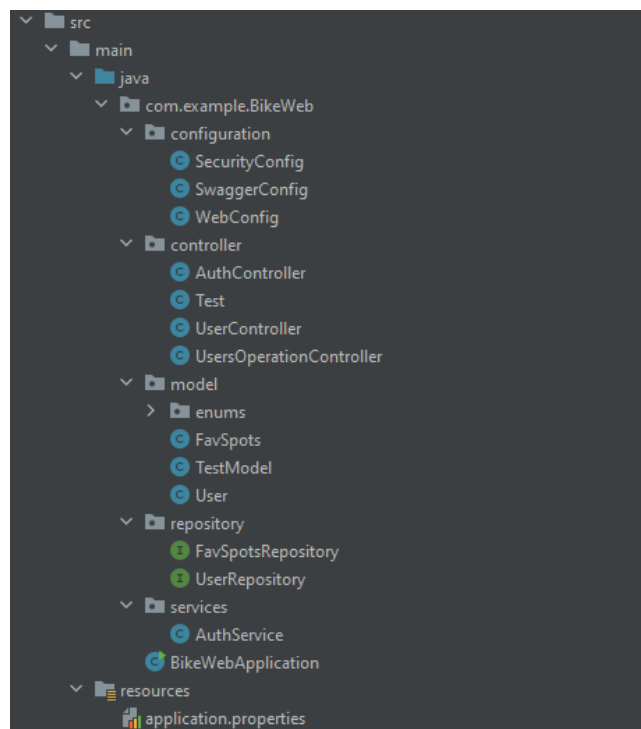


Rysunek 1 - schemat bazy danych MySQL

Została utworzona baza danych zawierająca dwie tabele: users oraz fav\_spots.

W tabeli users przechowywani są dane użytkowników w tym e\_mail i zahaszkowane hasło, za pomocą których użytkownik może się zalogować. Tabela fav\_spots zawiera dane o zapisanych stacjach rowerowych – dane te potrzebne są do pobierania zawartości z zewnętrznego API. Obie tabele połączone są relacją jeden do wielu (jeden użytkownik może mieć wiele zapisanych stacji rowerowych).

### 3. Backend – Java Spring Boot



*Rysunek 2 - struktura projektu po stronie serwera*

Backend został podzielony strukturalnie na pięć części:

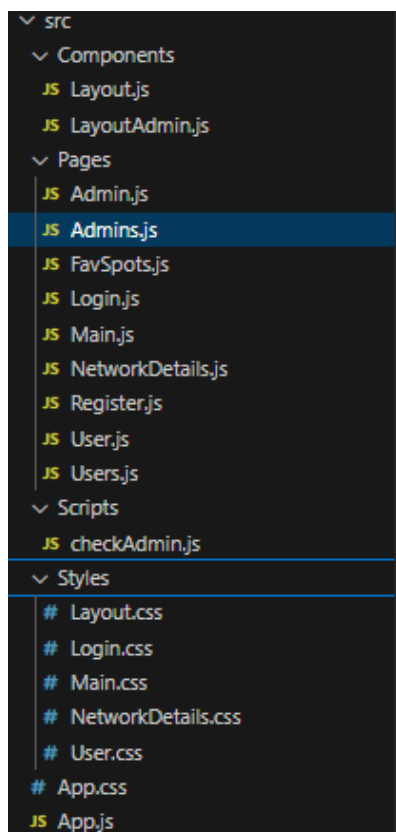
1. Configuration:  
Zawiera klasy konfiguracyjne dla encodera haseł, generatora dokumentacji end-point'ów oraz dla ustawienia polityki CORS.
2. Controller:  
Zawiera definicje end-point'ów oraz logikę po stronie serwera.
3. Model:  
Zawiera klasy odzwierciedlające stan tabel w bazie danych.
4. Repository:  
Zawiera interfejsy, który służą do obsługi operacji na bazie danych związanych z poszczególnymi encjami.
5. Services:  
Zawiera definicje metod, aby nie były powtarzane w wielu miejscach w kodzie.

Zdefiniowane zostały odpowiednie end-point'y do komunikacji z serwerem:

user-controller			^
PUT	/user/modify/{id}		🔒 ▼
GET	/user/get-all-fav-places		🔒 ▼
GET	/user/find-by-email/{email}		🔒 ▼
GET	/user/		🔒 ▼
DELETE	/user/delete/{id}		🔒 ▼
auth-controller			^
PUT	/auth/login		🔒 ▼
POST	/auth/register		🔒 ▼
GET	/auth/isadmin		🔒 ▼
users-operation-controller			^
POST	/add-fav-place		🔒 ▼
POST	/add-fav-place-by-id/{id}		🔒 ▼
DELETE	/del-fav-place		🔒 ▼
DELETE	/del-fav-place-by-id/{id}		🔒 ▼

Rysunek 3 - lista zdefiniowanych end-point'ów po stronie serwera

## 4. Frontend – React



Frontend został podzielony strukturalnie na cztery części:

1. Components:  
Zawiera definicje komponentów, aby nie były powtarzane w wielu miejscach w kodzie.
2. Pages:  
Zawiera odpowiednio strony dostępne dla użytkowników.
3. Scripts:  
Zawiera definicje skryptów, aby nie były powtarzane w wielu miejscach w kodzie.
4. Styles:  
Zawiera pliki stylów dla danej strony zdefiniowanej w Pages.

Rysunek 4 - struktura projektu po stronie klienta

Po stronie użytkownika zaimplementowane zostało pobieranie danych z zewnętrznego API <https://api.citybik.es/v2/networks> oraz przetrzymywanie tokena autoryzacyjnego niezbędnego do uzyskania dostępu m.in. do panelu użytkownika, zapisywania stacji rowerowych czy do panelu administratora. Token ten jest wysyłany w nagłówku requestu do backend'u.

Znajdź swój rower

[Strona główna](#) [Zaloguj się](#) [Zarejestruj się](#)

Rejestracja

Imię:

Nazwisko:

Email:

Hasło:

Zarejestruj się

© 2025 BikeWeb. Wszystkie prawa zastrzeżone.

Rysunek 5 - strona rejestracji

Znajdź swój rower

[Strona główna](#) [Zaloguj się](#) [Zarejestruj się](#)

Logowanie

Email

Wprowadź email

Hasło

Wprowadź hasło

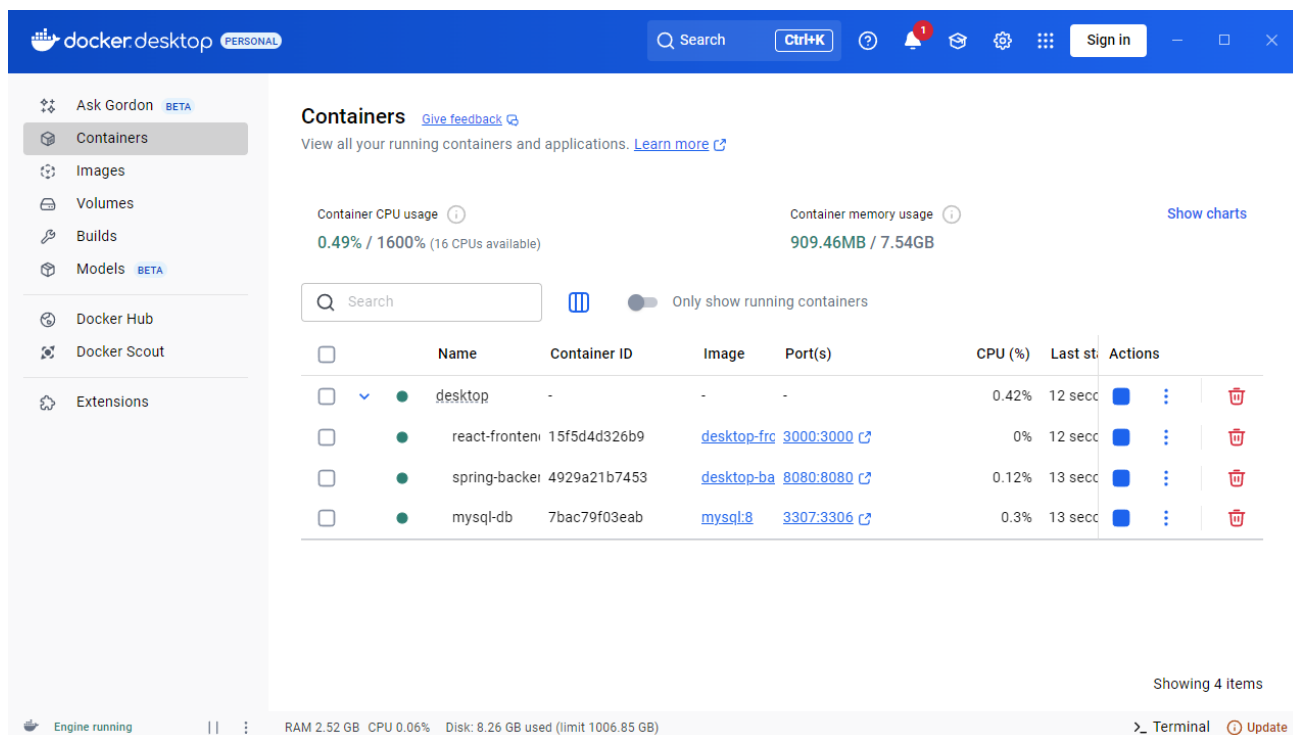
Zaloguj się

© 2025 BikeWeb. Wszystkie prawa zastrzeżone.

Rysunek 6 - strona logowania

## 5. Konteneryzacja aplikacji – Docker

Zostały utworzone pliki *Dockerfile* w katalogu zarówno backend'u jak i frontend'u definiujące zachowanie plików projektu przy budowie aplikacji oraz ogólny plik *docker-compose* określający nazwy kontenerów, porty oraz inne ustawienia środowiska. Następnie obrazy Docker'owe zostały zbudowane i uruchomione kontenery.



Rysunek 7 - podgląd na uruchomione kontenery w Docker desktop