

# Performance comparison of working with a database in Spring Boot version 3.2.3 and Laravel version 8.83.27

## Porównanie wydajności pracy z bazą danych w Spring Boot v 3.2.3 i Laravel v 8.83.27

Mateusz Zajączkowski\*

*Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland*

### Abstract

The subject of the article is the analysis of the efficiency of working with a database of applications created using programming frameworks: Spring Boot version 3.2.3 and Laravel version 8.83.27. Test applications were prepared in the REST API architecture, which examined the time efficiency of CRUD operations on the original database. Project structure and performance of test applications were compared. After analyzing the obtained results, it was found that in all scenarios, Spring Boot performed data operations on average 12 times faster.

*Keywords:* Spring Boot; Laravel; CRUD

### Streszczenie

Tematem artykułu jest analiza wydajności pracy z bazą danych aplikacji wykonanych przy pomocy szkieletów programistycznych: Spring Boot w wersji 3.2.3 i Laravel w wersji 8.83.27. Przygotowano aplikacje testowe, w architekturze REST API, które badały wydajność czasową operacji CRUD na autorskiej bazie danych. Porównano strukturę projektów oraz wydajności aplikacji testowych. Po przeanalizowaniu uzyskanych wyników stwierdzono, że we wszystkich scenariuszach, Spring Boot realizował operację na danych średnio 12 razy szybciej.

*Słowa kluczowe:* Spring Boot; Laravel; CRUD

\*Corresponding author

*Email address:* [mateusz.zajaczkowski@pollub.edu.pl](mailto:mateusz.zajaczkowski@pollub.edu.pl) (M. Zajączkowski)

Published under Creative Common License (CC BY 4.0 Int.)

## 1. Wstęp

Obecnie wybór właściwego szkieletu programistycznego jest kluczowym elementem procesu tworzenia oprogramowania. W niniejszym artykule została przeprowadzona analiza porównawcza dwóch popularnych szkieletów programistycznych: Spring Boot w wersji 3.2.3, dedykowanego językowi Java oraz Laravel w wersji 8.83.27., stworzonego dla języka PHP. Oba te narzędzia są szeroko stosowane w branży programistycznej, zarówno przez małe, jak i duże firmy.

Laravel jest to szkielet programistyczny wydany na licencji open-source, przeznaczony do tworzenia aplikacji internetowych. Charakteryzuje się elegancką i ekspresywną składnią. Laravel oferuje użytkownikowi typowe funkcjonalności jak: uwierzytelnianie, sesje czy routing. Realizuje założenia wzorca MVC (ang. Model View Control), ale może być również stosowany do implementacji REST API (ang. Representational state transfer).

Spring Boot to szkielet programistyczny dla języka Java. Charakteryzuje się prostotą i szybkością tworzenia aplikacji. Zapewnia wydajne narzędzia do rozwijania mikroserwisów oraz obsługę aspektów takich jak konfiguracja, zarządzanie zależnościami i wbudowany serwer.

Celem badania było porównanie tych dwóch popularnych i znaczących technologii na rynku [1], głównie pod kątem wydajności czasowej wykonywania operacji typu CRUD (ang. create, read, update, delete). Postawiono następującą tezę badawczą: "Spring Boot zapewnia wyższą

wydajność w porównaniu do Laravel jeśli chodzi o wykonywanie prostych operacji na bazie danych".

## 2. Przegląd literatury

W artykule [2] z 2019 roku autorzy porównali ze sobą szkielety programistyczne Spring i Laravel, żeby sprawdzić, który z nich oferuje większe możliwości wytwarzania oprogramowania oraz jest bardziej wydajny. W tym celu wykonali dwie aplikacje przy użyciu wcześniej wymienionych technologii. Analiza przeprowadzili w oparciu o następujące kryteria: model tworzenia aplikacji, obsługa operacji bazodanowych, bezpieczeństwo, wydajność i wybrane metryki kodu. Do analizy wydajności skorzystali z narzędzia ChromeDev Tools oraz programu JMeter, który pozwala symulować obciążenie aplikacji. Wyniki przedstawiono w postaci punktowej. Autorzy doszli do wniosku, że Laravel jest prostszy w nauce, ma wyższy poziom abstrakcji niż Spring i lepiej nadaje się do małych i średnich projektów. Z kolei Spring lepiej sprawdza się w zadaniach wymagających dużej wydajności czasowej i oferuje więcej materiałów i poradników.

W artykule [3] z 2018 roku autorzy zbadali dwa języki programowania, Java i PHP. Badanie opierało się na zbadaniu wydajności dwóch aplikacji. Aplikacje służyły do oceny pracy nauczyciela akademickiego poprzez wypełnianie ankiet przez uczniów. Wydajność oceniano przy pomocy narzędzia JMeter. Swoje badania oparli na czterech parametrach: średni czas odpowiedzi, przepustowość, odchylenie standardowe oraz przepływność

(ilość rzeczywistych danych wysyłanych w danej jednostce czasu). Otrzymane wyniki były podobne do siebie, ale aplikacja w Java osiągała lepsze wyniki w każdej z kategorii, co doprowadziło do wniosku, że Java jest bardziej wydajna niż PHP.

W artykule [4] z 2022 autorzy porównali wydajność języków Java i PHP. Zrealizowali to poprzez zmierzenie czasu wykonania, obciążenie procesora i pamięci RAM dla czterech operacji: matematycznej sumy, metody numerycznej Newton-Raphsona, sortowania bąbelkowego i zapisu danych do bazy danych PostgreSQL. Czas wykonania okazał się mniejszy dla każdej z operacji dla języka Java. Jednak, jeśli chodzi o użycie pamięci RAM to dla każdej operacji lepszy był język PHP. W kwestii obciążenia procesora to lepszy był PHP - Java osiągnęła lepsze rezultaty tylko w przypadku wykonywania operacji numerycznej. Jednak mimo tego, że użycie procesora dla Javy było większe, to było znacznie bardziej stabilne niż w przypadku PHP, który osiągał duże wahania odczytów. Autorzy doszli do wniosku, że język Java jest bardziej wydajny czasowo a PHP lepiej (ale mniej stabilnie) zarządza pamięcią.

### 3. Badane technologie

W celu przeprowadzenia badań zostały wykonane dwie aplikacje testowe, jedną przy pomocy szkieletu Laravel i drugą przy pomocy szkieletu Spring Boot. Obie aplikacje nie posiadają interfejsu graficznego, komunikacja z bazą danych odbywa się przy pomocy protokołu HTTP (REST API) przy użyciu narzędzia JMeter.

#### 3.1. Spring Boot

Spring Boot to szkielet programistyczny oparty na języku Java, umożliwiający tworzenie aplikacji dla platformy Java Platform, Enterprise Edition. Pierwsze wydanie Spring Framework pojawiło się w październiku 2002 roku na bazie licencji „Apache 2.0 license”. Wersja 1.0 ukazała się w marcu 2004 roku [5]. Został stworzony w celu uproszczenia procesu programowania poprzez dostarczenie domyślnych ustawień i konfiguracji, dzięki którym programiści mogą skupić się na implementacji logiki biznesowej zamiast zajmować się szczegółową konfiguracją.

Zalety szkieletu Spring Boot [6]:

- wsparcie dla mikrousług,
- automatyczna konfiguracja,
- wbudowany serwer aplikacji,
- prostota i elastyczność,
- rozbudowana społeczność użytkowników.

#### 3.2. Laravel

Laravel to szkielet programistyczny do tworzenia aplikacji webowych oparty na języku PHP bazujący na wzorcu architektonicznym MVC [7]. Pierwsza wersja szkieletu Laravel ukazała się w czerwcu 2011 roku. Jest znany z czytelnego i wydajnego kodu, oferuje wiele funkcji ułatwiających rozwój aplikacji, takich jak Eloquent ORM, Blade Template Engine, czy Artisan Console. Modułowa budowa Laravela umożliwia dostosowanie szkieletu do konkretnych wymagań projektu. Wspierając testowanie

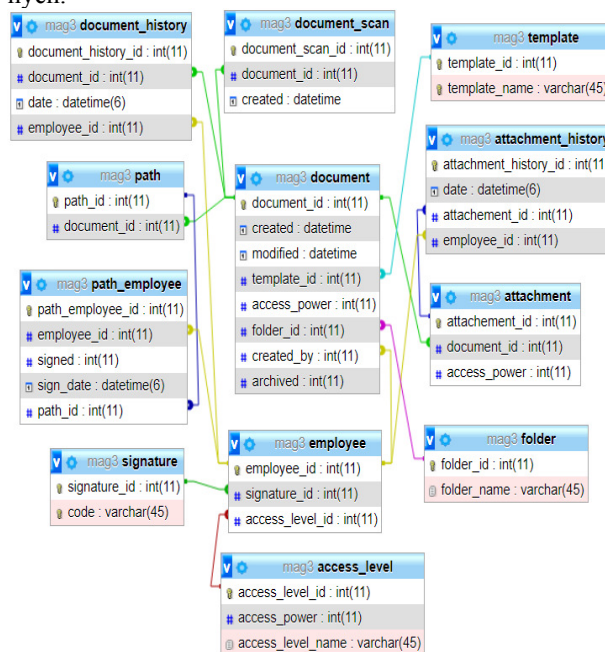
jednostkowe i funkcjonalne, Laravel zachęca do dbałości o jakość kodu oraz sprawdzania poprawności działania aplikacji. Laravel jest często używany w projektach PHP ze względu na swoją prostotę użycia i rozbudowane funkcje.

Zalety szkieletu Laravel [8]:

- czysta i elegancka składnia,
- wsparcie dla migracji baz danych,
- modularność i rozszerzalność,
- zaawansowane wsparcie dla testowania,
- Eloquent ORM (ang. Object-Relational Mapping), który upraszcza pracę z bazami danych.

### 4. Baza danych

Na potrzeby artykułu utworzono bazę danych na silniku MySQL. Na Rysunku 1 przedstawiono schemat bazy danych.



Rysunek 1: Baza danych wykorzystywana w testach.

Utworzona baza danych imituje system wspomagający obieg dokumentów w firmie. System obsługuje obieg całej dokumentacji, od tworzenia dokumentów po definowanie ich ścieżki obiegu oraz archiwizację. Każdy dokument ze zdefiniowaną ścieżką obiegu trafia do podpisu do osób, które na ścieżce figurują. System sprawdza, czy umieszczony został podpis i umożliwia przesłanie dokumentu dalej. Dodatkowo w systemie zdefiniować można strukturę różnych rodzajów dokumentów, z których można skorzystać tworząc gotowy dokument. Każdy dokument ma metadane, które określają daty utworzenia i modyfikacji dokumentu oraz całą jego historię. Dokumenty mają także zdefiniowane poziomy dostęp, określające którzy użytkownicy (będący na określonym poziomie dostępu) mogą mieć do niego dostęp. Poziomy dostęp można definiować hierarchicznie (jeden poziom może zawierać w sobie kilka innych). Dokumenty archiwalne również są klasyfikowane i przechowywane w systemie. Dokumenty mogą mieć również załączniki, do których także można definiować dostęp. Każdy

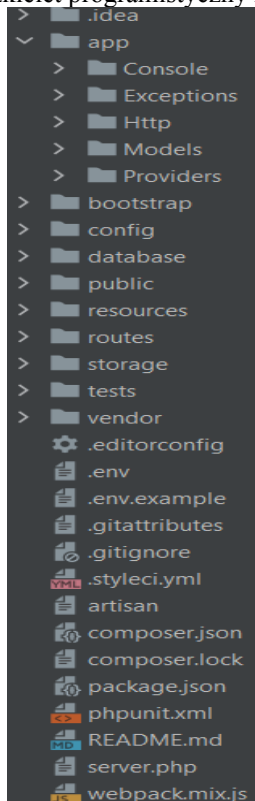
dokument (też załączniki) musi mieć przechowywaną historię i wszystkie wersje.

## 5. Struktura projektu testowego

### 5.1. Laravel

Dla wszystkich 12 tabel w bazie danych stworzono odpowiadające im modele, które odwzorowują tabele z bazy danych oraz kontrolery, w których są zaimplementowane reakcje na żądania typu CRUD. Obsługa punktów końcowych została wykonana w pliku `api.php`.

Na Rysunku 2 przedstawiono strukturę aplikacji wykorzystującej szkielet programistyczny Laravel.



Rysunek 2: Struktura aplikacji Laravel.

Najważniejsze katalogi i pliki tej aplikacji to:

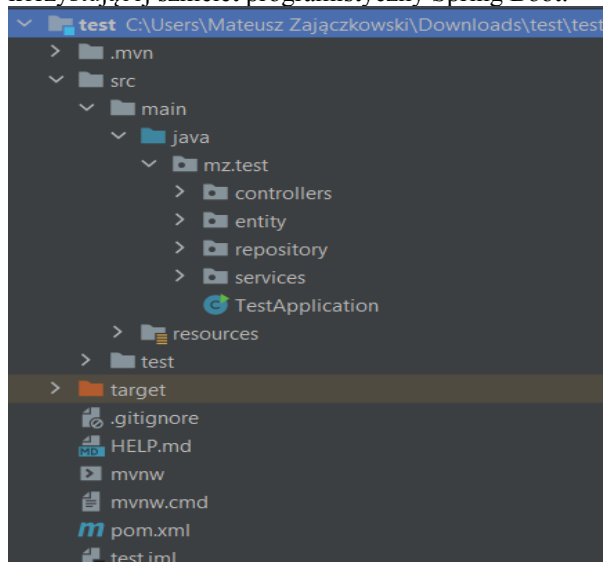
- `app`: w tym katalogu znajduje się kod aplikacji. Katalog ten składa się z podkatalogów:
  - `Console`: zawierającego wszystkie polecenia Artisan,
  - `Exceptions`: zawierającego obsługę wyjątków w aplikacji,
  - `HTTP`: w którym umieszczono kontrolery dla wszystkich modeli,
  - `Models`: zawiera modele dla każdej encji w bazie danych,
  - `Providers`: zawierającego dostawców usług aplikacji.
- `bootstrap`: w tym katalogu znajdują się pliki odpowiedzialne za ładowanie środowiska oraz szkieletu Laravel,
- `config`: w tym katalogu przechowywane są pliki konfiguracyjne aplikacji, takie jak ustawienia bazy danych, konfiguracja sesji, pliki konfiguracyjne dostawców, itp.,

- `database`: zawiera migracje, które są używane do zarządzania schematem bazy danych,
- `public`: tutaj znajduje się katalog publiczny. Zawiera on m.in. pliki statyczne,
- `resources`: w tym katalogu znajdują się pliki CSS, JavaScript, widoki (szablony HTML) oraz pliki językowe,
- `routes`: tutaj znajdują się pliki definicji tras, które określają, jak aplikacja ma odpowiadać na żądania http,
- `storage`: katalog ten przechowuje pliki generowane przez szkielet, takie jak pliki dziennika zdarzeń, pliki sesji, cache itp.,
- `tests`: w tym katalogu znajdują się testy aplikacji,
- `vendor`: katalog ten zawiera wszystkie zależności zewnętrzne pobrane przez Composer,
- `env`: jest to plik konfiguracyjny, który przechowuje zmienne środowiskowe,
- `artisan`: jest to plik wykonywalny, który jest używany do wykonywania poleceń konsoli Laravel.

### 5.2. Spring Boot

Dla wszystkich 12 tabel w bazie danych stworzono odpowiadające im encje. Dla wszystkich encji zostały również stworzone repozytoria oraz serwisy. Repozytoria odpowiadają za komunikację z bazą danych. Zaimplementowano interfejs `Jpa`, który zapewnia gotowe metody do wykonywania operacji na bazie danych (CRUD) dzięki czemu nie było potrzeby własnoręcznej ich implementacji. Serwisy zawierają logikę aplikacji. Są one odpowiedzialne za przetwarzanie danych oraz wykonywanie operacji na bazie danych. W pliku kontrolera znajduje się obsługa punktów końcowych.

Na Rysunku 3 przedstawiono strukturę aplikacji wykorzystującej szkielet programistyczny Spring Boot.



Rysunek 3: Struktura aplikacji Spring Boot.

Najważniejsze pliki to:

- `src/main/java`: w tym katalogu znajduje się kod aplikacji. Katalog ten składa się z podkatalogów:

- controllers: kontrolery obsługujące żądania HTTP,
- entity: encje reprezentujące obiekty w bazie danych,
- repository: repozytoria do interakcji z bazą danych,
- services: katalog z serwisami,
- TestApplication: główna klasa aplikacji.
- src/main/resources: ten katalog zawiera pliki konfiguracyjne oraz zasoby używane przez aplikację,
- src/test: w tym katalogu znajdują się testy,
- target: ten katalog zawiera pliki wygenerowane podczas budowania projektu,
- mvnw, mvnw.cmd, .mvn: te pliki i katalog zawierają narzędzia Maven Wrapper,
- pom.xml: plik konfiguracyjny projektu Maven, który zawiera informacje o zależnościach projektu oraz konfigurację budowania projektu.

## 6. Opis stanowiska badawczego

W Tabeli 1 przedstawiono konfigurację środowiska badawczego służącego do przeprowadzania badań.

Tabela 1: Specyfikacja środowiska testowego

Pamięć RAM	8 GB
Dysk	SSD 512 GB
Procesor	Intel Core i5-9300H CPU 2.40GHz
System operacyjny	Windows 11

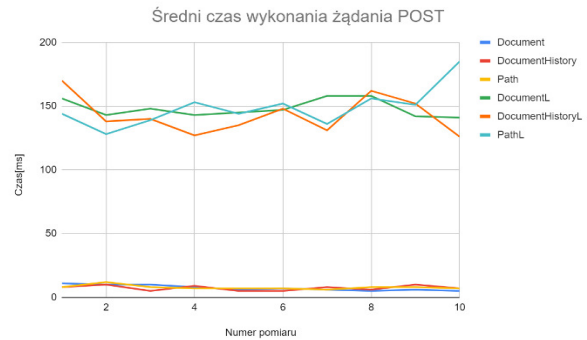
## 7. Scenariusze badawcze

W każdym z scenariuszy wykonano po 10 pomiarów. Wszystkie badania zostały przeprowadzone na trzech tabelach: Document, DocumentHistory i Path. Badania zostały wykonane przy użyciu narzędzia JMeter. Opracowano 4 scenariusze badawcze:

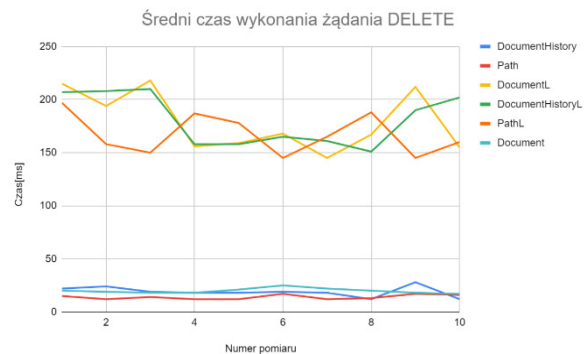
- scenariusz 1: badanie czasu dodania rekordu do bazy danych,
- scenariusz 2: badanie czasu usunięcia rekordu z bazy danych,
- scenariusz 3: badanie czasu edycji rekordu w bazie danych.
- scenariusz 4: badanie czasu pobrania rekordów z bazy danych.

## 8. Wyniki badań

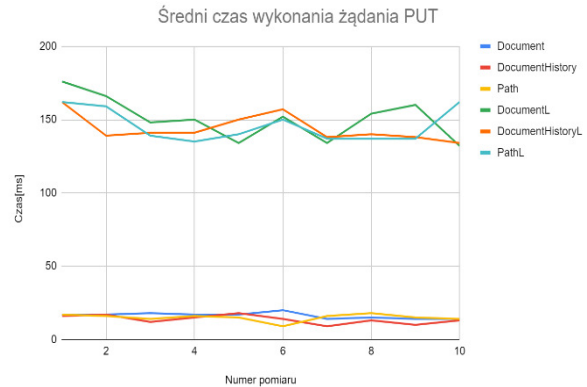
Wyniki pomiarów przedstawiono na Rysunkach 4-7. Literą L na końcu oznaczono czasy realizacji poszczególnych scenariuszy dla aplikacji wykorzystującej szkielet programistyczny Laravel. Pozostałe czasy dotyczą aplikacji wykorzystującej szkielet programistyczny Spring Boot.



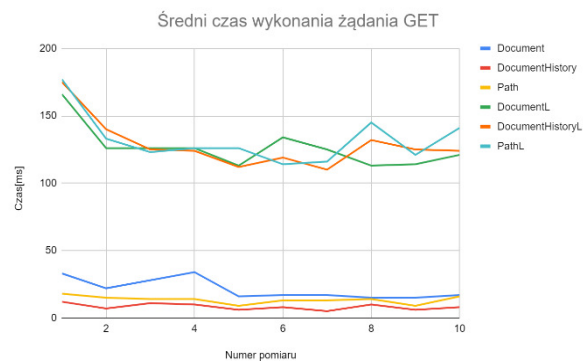
Rysunek 4: Średni czas realizacji operacji dodawania danych (scenariusz 1).



Rysunek 5: Średni czas realizacji operacji usuwania danych (scenariusz 2).



Rysunek 6: Średni czas realizacji operacji edycji danych (scenariusz 3).



Rysunek 7: Średni czas realizacji operacji pobierania danych (scenariusz 4).



Wyniki badań dla czterech scenariuszy dla aplikacji Spring Boot przedstawiono w Tabeli 2.

Tabela 2: Średni czas [ms] odpowiedzi dla Spring Boot

Numer scenariusza	Tabela Document	Tabela Document History	Tabela Path
1	7,4	7,3	7,8
2	19,8	19	14
3	16,3	13,7	15
4	21,4	8	13,5

Wyniki badań dla czterech scenariuszy dla aplikacji Laravel przedstawiono w Tabeli 3.

Tabela 3: Średni czas [ms] odpowiedzi dla Laravel

Numer scenariusza	Tabela Document	Tabela Document History	Tabela Path
1	148,1	142,9	148,4
2	178,9	181	167,3
3	150,6	144,3	145,8
4	126,4	128,6	132,2

Porównując czas odpowiedzi aplikacji wykonanych przy użyciu szkieletów programistycznych Laravel i Spring Boot, zauważono, że aplikacja oparte na szkielecie programistycznym Laravel charakteryzują się wyższymi czasami odpowiedzi we wszystkich scenariuszach badawczych. W aplikacjach wykonanych przy użyciu Spring Boot zaobserwowano wzrost czasu odpowiedzi podczas operacji usuwania lub edycji rekordów w bazie danych. Te operacje trwały dłużej niż dodawanie nowych rekordów. Natomiast w przypadku pobierania rekordów z bazy danych, wyniki były najbardziej rozbieżne. Dla aplikacji Laravel czas odpowiedzi był względnie jednolity w różnych scenariuszach, chociaż przy pobieraniu rekordów z bazy danych czas był najkrótszy. Wyniki badań nie wskazywały na to, aby rozmiar tabeli w bazie danych miał istotny wpływ na otrzymane wyniki eksperymentów.

## 9. Wnioski

Celem artykułu było przeprowadzenie analizy wydajności czasowej operacji bazodanowych dwóch szkieletów programistycznych, Spring Boot i Laravel. W tym celu

stworzono dwie aplikacje, które realizowały operacje CRUD na bazie danych.

W przypadku aplikacji wykorzystującej szkielet programistyczny Spring Boot najszybciej wykonywały się żądania POST a w przypadku aplikacji wykorzystującej szkielet programistyczny Laravel najszybciej wykonywały się żądania GET. Wyniki badań wskazują na przewagę wydajnościową Spring Boot nad Laravel, operacje na danych były średnio 12 razy szybsze.

Przeprowadzona analiza porównawcza wykazała, że aplikacje wykorzystujące szkielet programistyczny Spring Boot lepiej radzą sobie z prostymi operacjami na bazie danych pod względem czasu odpowiedzi na operacje CRUD niż aplikacje wykorzystujące szkielet programistyczny Laravel.

## Literatura

- [1] Stackoverflow – ankieta popularności technologii, <https://survey.stackoverflow.co/2023/#technology-most-popular-technologies>, [10.05.2024].
- [2] S. Jędrych, B. Jędruszek, B. Pańczyk, Tworzenie aplikacji internetowych na platformie JEE i PHP – analiza porównawcza, Journal of Computer Sciences Institute 11 (2019) 86-90, <https://doi.org/10.35784/jcsi.145>.
- [3] S. Memon, R. B. Palh, M. Memon, H. S. Memon, Performance comparison of QEC network based JAVA application and web based PHP application, International Journal of Advanced Computer Science and Applications 9(8) (2018) 555-564, <https://doi.org/10.14569/IJACSA.2018.090870>.
- [4] K. Gruszczyński, W. Zabierowski, Performance Tests of Java and PHP Programming Languages for Use in Sensor Networks, In 2022 IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH) (2022) 70-74, <https://doi.org/10.1109/MEMSTECH55132.2022.10002927>.
- [5] Spring, <https://www.w3schools.com/css/default.asp>, [10.05.2024].
- [6] Cechy szkieletu Spring, <https://spring.io/why-spring>, [10.05.2024].
- [7] Laravel, <https://pl.wikipedia.org/wiki/Laravel>, [10.05.2024].
- [8] Cechy szkieletu Laravel, <https://laravel.com/>, [10.05.2024].