

Performance analysis of web applications created in the Spring and Laravel frameworks

Analiza wydajnościowa aplikacji internetowych utworzonych w szkieletach Spring i Laravel

Jakub Suchanowski*, Małgorzata Plechawska-Wójcik

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

Abstract

The article compares the performance of two test applications created using the Spring Boot and Laravel application frameworks. The aim of the study is to find an answer to the research question: which framework offers better time performance. Twelve test scenarios were created for the analysis, and they were used to conduct performance tests using the Apache JMeter tool. Additionally, application metrics and community support for both frameworks were compared. The research confirmed that the application built on the Spring Boot framework has better performance.

Keywords: spring boot; laravel; web applications; application performance

Streszczenie

W artykule porównano wydajność dwóch aplikacji testowych, które stworzono używając szkieletów aplikacji Spring Boot i Laravel. Celem pracy jest uzyskanie odpowiedzi na pytanie badawcze: który szkielet oferuje lepszą wydajność czasową. Do analizy stworzono 12 scenariuszy testowych, a następnie na ich podstawie przeprowadzono testy używając narzędzia Apache JMeter. Dodatkowo porównane zostały metryki aplikacji oraz wsparcie społeczności dla obu szkieletów. Badania potwierdziły, że aplikacja zbudowana na szkielecie Spring Boot ma lepszą wydajność.

Słowa kluczowe: spring boot; laravel; aplikacje internetowe; wydajność aplikacji

*Corresponding author

Email address: jakub.suchanowski@pollub.edu.pl (J. Suchanowski)

©Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

W dzisiejszych czasach aplikacje internetowe pełnią ważną rolę w różnych dziedzinach, takich jak media społecznościowe, bankowość, dostęp do informacji, handel internetowy. Wraz z dynamicznym rozwojem technologii internetowych i rosnącymi oczekiwaniami użytkowników, tworzenie wydajnych aplikacji internetowych stało się niezwykle istotne. Zapewnienie odpowiedniej wydajności w dużej mierze wpływa na satysfakcję użytkowników i tym samym na pozytywny odbiór przekazywanych treści [1]. Z tego powodu programiści tworzący aplikacje internetowe muszą odpowiednio dobrać używane technologie i zadbać o zoptymalizowanie ich pod kątem wydajności.

Istnieje wiele artykułów porównujących wydajności aplikacji internetowych zbudowanych na poszczególnych szkieletach aplikacji (ang. framework). Najczęściej autorzy skupiali się na porównaniu aplikacji stworzonych za pomocą jednego języka programowania. W tym artykule porównane zostały aplikacje na bazie jednych z najpopularniejszych szkieletów z dwóch różnych języków programowania. Podobne porównanie stworzył Jakub Radomski [2]. Autor przedstawił wyniki wydajności dwóch aplikacji internetowych stworzonych na bazie Javy i Vaadin oraz PHP i Laravel. Dla obu aplikacji przeprowadził testy wydajności czasowej tworząc 13 scenariuszy, w których za pomocą narzędzia Chrome DevTools oraz ApacheBench mierzył czasy trwania operacji CRUD (Create, Read, Update i Delete) na zbiorach danych. Badania pokazały, że aplikacja na bazie szkieletu Vaadin jest znacznie wydajniejsza, niż aplikacja zbudowana za pomocą platformy Laravel. Największe różnice widoczne były przy zapisie i odczycie większej liczby rekordów oraz przy symulacji pobierania zasobów dla większej liczby użytkowników. W pracy pt. „Porównanie wytwarzania oprogramowania internetowego z wykorzystaniem różnych technologii” [3] porównano dwie aplikacje internetowe stworzone na bazie PHP i Laravel oraz Javy i Spring. Artykuł przedstawia porównanie tych technologii pod względem pracy z bazą danych. We wnioskach autor stwierdził, że przy dodawaniu nowego rekordu do bazy danych szybszy jest Spring. Przy odczytywaniu pojedynczych rekordów z bazy danych oraz obliczaniu n-tego Ciągu Fibonacciego również lepiej spisuje się Spring. Jedynie przy usuwaniu pojedynczego rekordu z bazy danych aplikacja stworzona na bazie szkieletu Laravel osiąga mniejszy czas. W artykule [4] autorzy stworzyli dwie aplikacje internetowe używając języka Java oraz PHP. Miały one na celu automatyzację manualnego systemu oceny nauczycieli na ich uniwersytecie. Aplikacje zawierały pytania, na które odpowiadali studenci, następnie po przesłaniu odpowiedzi na serwer, generowano raporty. Głównym celem pracy był pomiar wydajności aplikacji. Po wykonaniu pomiarów parametrów wydajności stwierdzono, że aplikacja zbudowana w języku Java reaguje szybciej niż aplikacja zbudowana w języku PHP oraz zawiera niższe wartości odchylenia standar-

dowego. Z kolei w badaniu pt. „A Performance Comparison of RESTful Applications Implemented in Spring Boot Java and MS.NET Core” [5] autor skupił się na porównaniu oraz ocenie wydajności aplikacji internetowych opartych na RESTful. Aplikacje zaimplementowano używając szkieletów Spring Boot oraz MS.NET Core. Aplikacje testowe obsługiwały podstawowe operacje Create, Read, Update i Delete (CRUD). Do analizy wydajności użyto narzędzia Apache JMeter. Przeprowadzone badania pozwoliły autorowi stwierdzić, że MS.NET Core zapewnia szybszy czas odpowiedzi oraz zużywa mniej zasobów w porównaniu do szkieletu Spring Boot we wszystkich metodach http. Wpływ języka na wydajność REST Api zbadano również w artykule [6]. Autorzy porównali trzy języki programowania: JavaScript, Java oraz Python. Na podstawie wyników, które uzyskali, stwierdzili, że dla małych i średnich aplikacji najlepszą wydajność osiągnął język JavaScript, natomiast dla aplikacji z dużą ilością danych najlepszym wyborem okazał się język Java. W kolejnej pracy [7] autorzy porównywali szkielety aplikacji Laravel oraz Yii2 języka PHP. Badanie przeprowadzono pod względem czasu wykonania, użycia pamięci oraz wydajności. Do testów posłużono się dużą bazą danych, aby sprawdzić jak badane szkielety aplikacji zachowują się przy pracy z nimi. Po opracowaniu wyników autorzy stwierdzili, że Laravel dominował pod względem czasu wykonania oraz wydajności, gorzej spisał się jedynie w zakresie użycia pamięci. Autorzy badania [8] porównującego trzy szkielety aplikacji PHP (Laravel, Symfony, CodeIgniter) uznali, że aplikacja zbudowana za pomocą szkieletu Laravel ma najlepszą wydajność. Podobne wnioski można zauważyć w kolejnym artykule [9], w którym porównano cztery szkielety aplikacji języka PHP. W artykule [10], gdzie porównano wydajność szkieletu Laravel ze szkieletem CodeIgniter, wyniki były różne pod względem czasu i szybkości działania. CodeIgniter osiągnął dłuższy czas działania niż Laravel oraz wyższy wynik w badaniu średniej szybkości. Natomiast w porównaniu [11] wydajności aplikacji na bazie szkieletów Laravel i Slim ten drugi wypadł znacznie lepiej. W przypadku porównania [12] szkieletu aplikacji Spring do szkieletu aplikacji Play, wyniki wydajności obu szkieletów były na podobnym poziomie. W literaturze można również znaleźć inne prace porównujące zarówno Spring Boot jak i Laravel do szkieletów aplikacji języka C# i innych języków programowania, gdzie autorzy pokazują mocne i słabe strony badanych struktur [13][14][15].

2. Cel i zakres pracy

Celem pracy jest porównanie wydajności dwóch aplikacji stworzonych za pomocą najpopularniejszych szkieletów aplikacji języka Javy i PHP. Celem badania jest odpowiedź na następujące pytanie: za pomocą której technologii można budować aplikacje o lepszej wydajności. Na potrzeby badania powstały dwie proste aplikacje testowe o identycznych funkcjonalnościach. Następnie stworzone aplikacje zostały porównane pod

względem czasu ładowania zasobów oraz obciążenia aplikacji.

Teza: Aplikacja internetowa stworzona przy użyciu szkieletu Spring Boot wykazuje lepszą wydajność niż analogiczna aplikacja zbudowana przy użyciu szkieletu Laravel.

Szczegółowe pytania badawcze:

1. Czy aplikacja stworzona w szkielecie Spring Boot wyświetla szybciej zasoby strony niż aplikacja w szkielecie Laravel?
2. Czy aplikacja stworzona przy użyciu Spring Boot ma mniejszy rozmiar od aplikacji stworzonej używając platformy Laravel?
3. Czy szkielet aplikacji Spring Boot ma większe wsparcie społeczności niż szkielet aplikacji Laravel?

3. Wykorzystane narzędzia i technologie

3.1. Spring Boot

Spring Boot to popularny framework, który umożliwia tworzenie aplikacji w języku Java. Spring Boot oparty jest na Spring Framework. Wykorzystuje najlepsze techniki swojego poprzednika, a dodatkowo zapewnia domyślne ustawienia konfiguracji dla wielu komponentów i bibliotek, minimalizując potrzebę manualnej konfiguracji oraz zapewnia kontener, który pozwala w prosty sposób uruchomić aplikację bez konieczności wdrażania i ustawiania zewnętrznych kontenerów. Framework posiada system zarządzania zależnościami oparty na Maven lub Gradle. Dzięki temu w łatwy sposób można zarządzać bibliotekami i ich wersjami, co ułatwia utrzymanie aplikacji oraz aktualizacje jej zależności. Spring Boot najczęściej wykorzystywany jest do tworzenia aplikacji internetowych i mikroserwisów.

3.2. Laravel

Laravel, framework języka PHP, został wydany w 2011 roku i szybko stał się jednym z popularniejszych szkieletów aplikacji dla tego języka. Zbudowany jest w oparciu o wzorzec MVC (Model-View-Controller), co pomaga w rozdzieleniu logiki biznesowej, warstwy prezentacji oraz warstwy danych i tym samym ułatwia zarządzanie kodem. Laravel zawiera w sobie wiele funkcjonalności, które są często używane w aplikacjach internetowych, takich jak uwierzytelnianie, routing, sesje czy buforowanie. Funkcjonalności te umożliwiają programiście pominięcie pisania części kodu, co wpływa na skrócenie czasu pracy nad projektami. Dodatkowo Laravel oferuje narzędzia usprawniające obsługę baz danych. Język zapytań Eloquent pozwala w prosty sposób tworzyć zapytania do bazy danych, definiować relacje między tabelami oraz wykonywać operacje CRUD (Create, Read, Update, Delete). Do tworzenia widoków framework wykorzystuje mechanizm szablonów Blade, który wykorzystuje składnię podobną do języka PHP.

3.3. JMeter

Apache JMeter to darmowe narzędzie do testowania wydajności aplikacji internetowych. Oprogramowanie

zostało stworzone w 1999 roku przez Apache Software Foundation. W ciągu kilku lat stało się jednym z najpotężniejszych narzędzi, które pozwala programistom i testerom wykonywać testy obciążeniowe, wydajnościowe i inne rodzaje testów aplikacji. Obecnie program JMeter używany jest na całym świecie przez największe firmy branży IT. Apache JMeter posiada interfejs graficzny, który pozwala na łatwe konfigurowanie i wykonywanie testów. Narzędzie umożliwia tworzenie scenariuszy testowych, w których definiuje się zestawy akcji, takich jak żądania http, symulowanie interakcji użytkowników, reakcji programu na poszczególne dane oraz sposób prezentacji wyników. Dodatkowo JMeter obsługuje wielowątkowość, co pomaga w badaniu wykonywania akcji przez wielu użytkowników jednocześnie.

4. Metoda badań

4.1. Plan badań

Analizę wydajnościową aplikacji utworzonych na bazie szkieletów programistycznych Laravel i Spring Boot zrealizowano na podstawie następującego planu:

1. Utworzenie dwóch identycznych aplikacji na podstawie dwóch różnych szkieletów programistycznych:
 - a) jedna używając szkieletu Laravel,
 - b) druga używając szkieletu Spring Boot.
2. Przeprowadzenie testów wydajności aplikacji:
 - a) użyte narzędzie: Apache JMeter,
 - b) kryteria badania: obsługa bazy danych (zapis, pobranie, edycja, usuwanie danych) dla różnych zbiorów danych,
3. Porównanie oraz opracowanie wyników.
4. Porównanie rozmiaru poszczególnych aplikacji:
 - a) ilość zajętego miejsca na dysku,
 - b) liczba plików,
 - c) metryki kodu.
5. Analiza wsparcia społeczności poprzez zbadanie liczby zapytań o dane szkielety na platformie stackoverflow.com [16].
6. Przedstawienie wniosków.

W celu przeprowadzania badań wykonane zostały dwie identyczne aplikacje testowe, które pozwalają użytkownikowi na wyświetlanie, dodawanie, aktualizację oraz usuwanie danych o samochodach. Jedna aplikacja została stworzona używając szkieletu Spring Boot, do drugiej użyto szkieletu Laravel. Do automatyzacji prac nad ich implementacją posłużyły narzędzia Maven w przypadku aplikacji tworzonej w języku Java oraz Composer w przypadku języka PHP. Obie aplikacje połączono z tą samą bazą danych obsługiwaną za pomocą oprogramowania MySQL. Wygląd interfejsu użytkownika stworzono przy użyciu silnika szablonów Thymeleaf dla aplikacji na bazie szkieletu Spring Boot oraz silnika szablonów Blade dla aplikacji na bazie szkieletu Laravel. Widok listy samochodów oraz formularza dodawania nowego samochodu w aplikacjach przedstawiony został na Rysunku 1 oraz Rysunku 2.

Marka	Model	Rok produkcji	Moc [KM]	Pojemność [cm3]	Rodzaj paliwa	Rodzaj napędu	Skrzynia biegów	Typ nadwozia	Liczba drzwi	Akcje
Alfa Romeo	Giulia	2021	200	1995	Gasoline	Rear wheels	Automatic	sedan	4	Edycja Usuń
Alfa Romeo	147	2003	140	1910	Diesel		Manual	compact	5	Edycja Usuń
Alfa Romeo	159	2007	120	1910	Diesel	Front wheels	Manual	sedan	4	Edycja Usuń
Alfa Romeo	159	2007	150	1910	Diesel	Front wheels	Automatic	station_wagon	4	Edycja Usuń
Alfa Romeo	Giulietta	2011	170	1400	Gasoline	Front wheels	Manual	compact	5	Edycja Usuń
Alfa Romeo	Giulietta	2014	120	1368	Gasoline	Front wheels	Manual	compact	5	Edycja Usuń

Rysunek 1: Wygląd strony wyświetlającej dane.

Dodaj nowy samochód

Marka

Model

Rok produkcji

Moc [KM]

Pojemność [cm3]

Rodzaj paliwa

Napęd

Rysunek 2: Wygląd strony z formularzem dodawania samochodu.

4.2. Środowisko testowe

Testy wydajności obu aplikacji przeprowadzone zostały w jednakowych warunkach, aby zapewnić jak najbardziej wiarygodne wyniki. Parametry środowiska testowego przedstawia Tabela 1.

Tabela 1: Parametry środowiska testowego

Sprzęt	
Procesor	Intel Core i5-8250
System operacyjny	Windows 11 Home 64-bit
Pamięć RAM	12 GB
Technologie	
PHP	8.2.0
Java	17.0.6
Laravel	10.10.0
Spring Boot	3.0.5
Apache JMeter	5.5

4.3. Scenariusze testowe

Do przeprowadzenia badania wydajności aplikacji zostały opracowane scenariusze testowe, które przedsta-

wia Tabela 2. Scenariusze dotyczą operacji wyświetlenia strony, dodania, edycji oraz usunięcia danych.

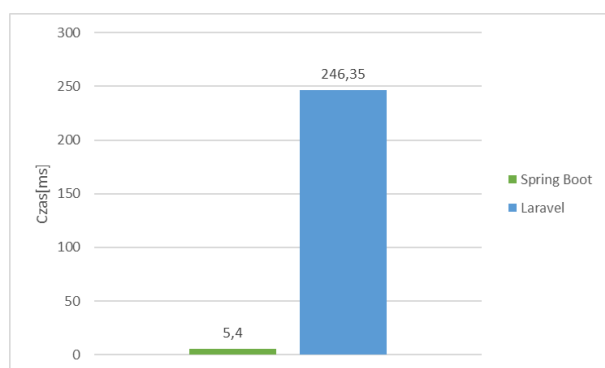
Tabela 2: Scenariusze testowe do zbadania wydajności aplikacji

Scenariusz	Opis
1	Wyświetlenie strony bez danych z bazy
2	Wyświetlenie strony z 10 rekordami przez 1 użytkownika
3	Wyświetlenie strony z 100 rekordami przez 1 użytkownika
4	Wyświetlenie strony z 1000 rekordów przez 1 użytkownika
5	Wyświetlenie strony z 10000 rekordów przez 1 użytkownika
6	Wyświetlenie strony z 10 rekordami gdy 10 użytkowników wyświetla ją jednocześnie
7	Wyświetlenie strony z 1000 rekordami gdy 10 użytkowników wyświetla ją jednocześnie
8	Dodanie 1 rekordu przez 1 użytkownika
9	Dodanie 1 rekordu gdy 10 użytkowników dodaje go jednocześnie
10	Aktualizacja 1 rekordu przez 1 użytkownika
11	Aktualizacja po 1 rekordzie gdy 10 użytkowników aktualizuje go jednocześnie
12	Usunięcie rekordu

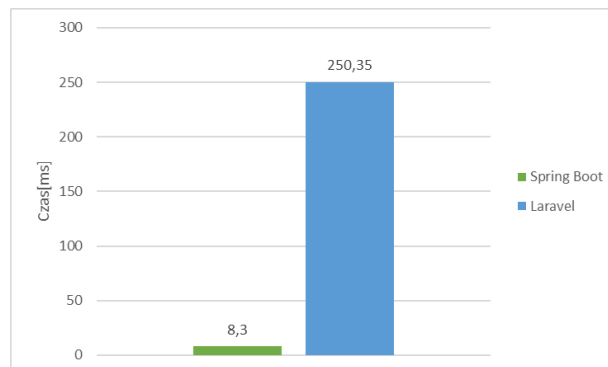
5. Prezentacja rezultatów badań

5.1. Analiza wydajnościowa

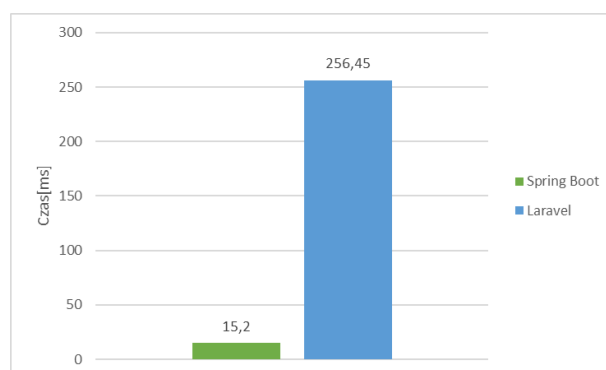
Wydajność aplikacji testowych zbadano za pomocą narzędzia Apache JMeter bazując na wcześniej przygotowanych scenariuszach. Każdy scenariusz został wykonany 20 razy dla każdej aplikacji. Następnie policzone zostały średnie czasy dla każdego scenariusza. Na Rysunkach 3-14 przedstawione są wyniki badań dla poszczególnych scenariuszy.



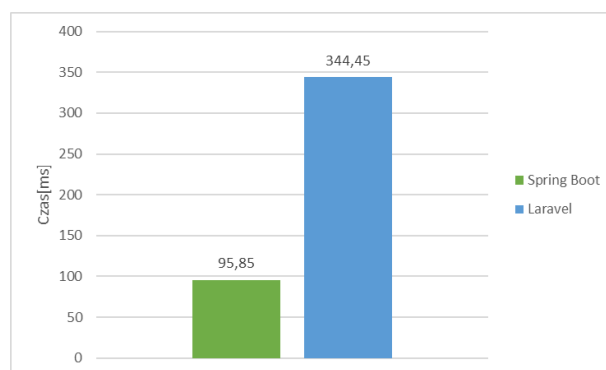
Rysunek 3: Średnie czasy operacji dla scenariusza 1 – wyświetlenie strony bez danych z bazy.



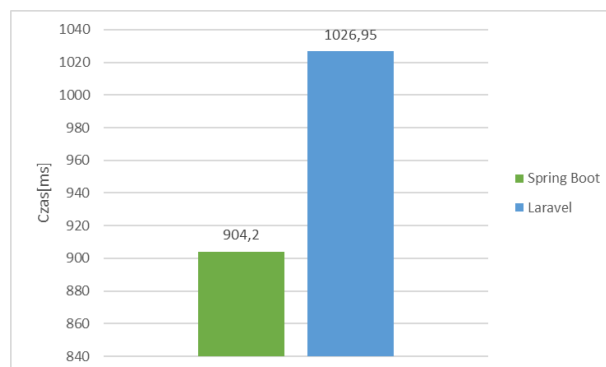
Rysunek 4: Średnie czasy operacji dla scenariusza 2 – wyświetlenie strony z 10 rekordami przez 1 użytkownika.



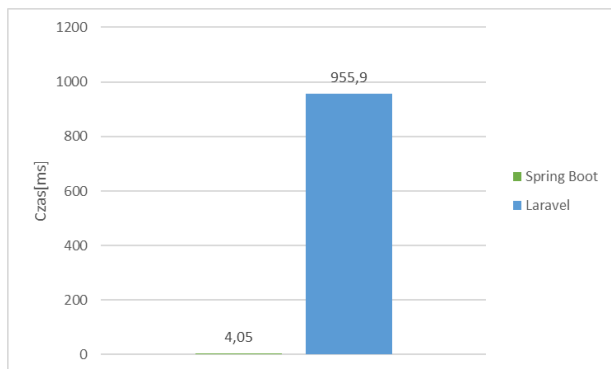
Rysunek 5: Średnie czasy operacji dla scenariusza 3 - wyświetlenie strony z 100 rekordami przez 1 użytkownika.



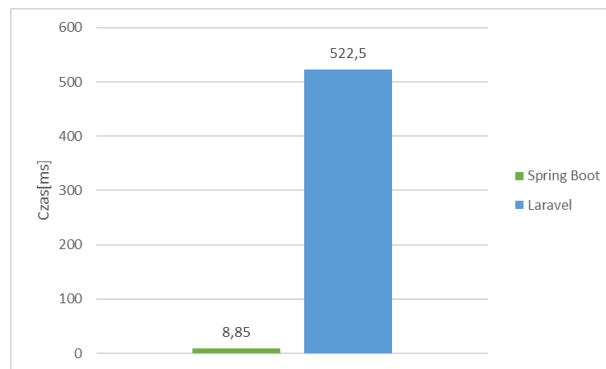
Rysunek 6: Średnie czasy operacji dla scenariusza 4 - wyświetlenie strony z 1000 rekordów przez 1 użytkownika.



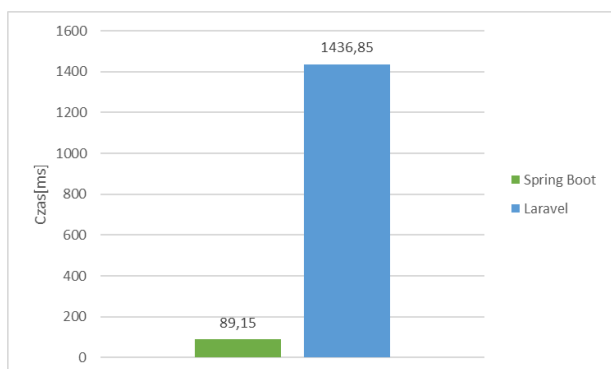
Rysunek 7: Średnie czasy operacji dla scenariusza 5 - wyświetlenie strony z 10000 rekordów przez 1 użytkownika.



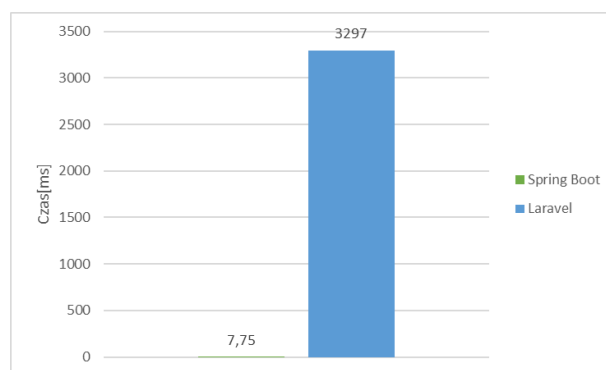
Rysunek 8: Średnie czasy operacji dla scenariusza 6 - wyświetlenie strony z 10 rekordami gdy 10 użytkowników wyświetla ją jednocześnie.



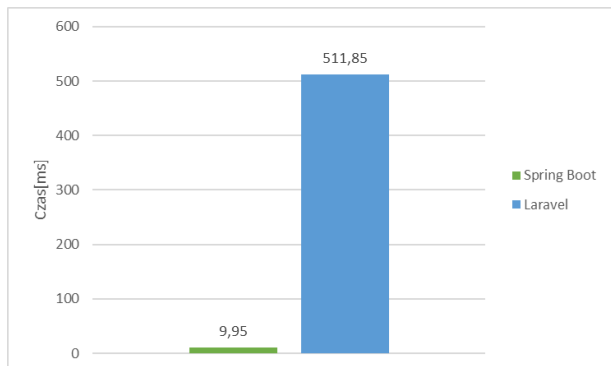
Rysunek 12: Średnie czasy operacji dla scenariusza 10 - aktualizacja 1 rekordu przez 1 użytkownika.



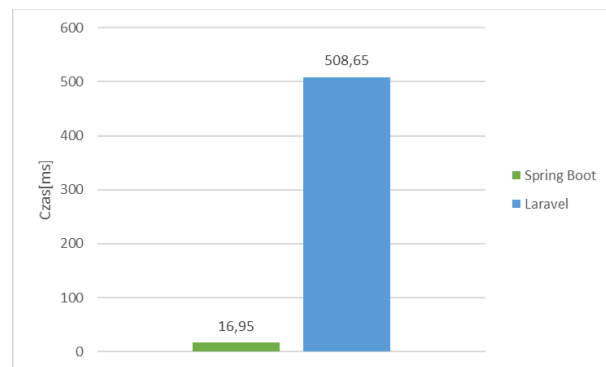
Rysunek 9: Średnie czasy operacji dla scenariusza 7 - wyświetlenie strony z 1000 rekordami gdy 10 użytkowników wyświetla ją jednocześnie.



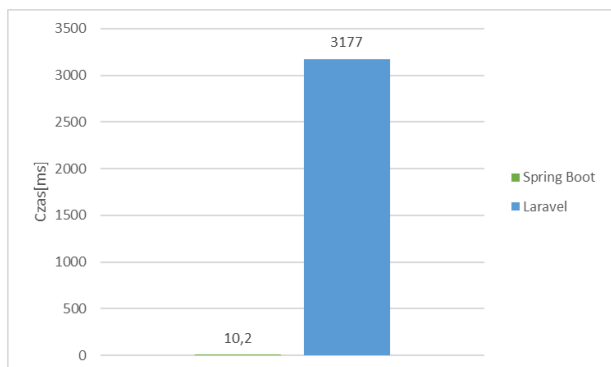
Rysunek 13: Średnie czasy operacji dla scenariusza 11 - aktualizacja po 1 rekordzie gdy 10 użytkowników aktualizuje go jednocześnie.



Rysunek 10: Średnie czasy operacji dla scenariusza 8 - dodanie 1 rekordu przez 1 użytkownika.



Rysunek 14: Średnie czasy operacji dla scenariusza 12 - usunięcie rekordu.



Rysunek 11: Średnie czasy operacji dla scenariusza 9 - dodanie 1 rekordu gdy 10 użytkowników dodaje go jednocześnie.

W scenariuszu 1, badano czasy wyświetlenia strony bez danych z bazy. Dla aplikacji napisanej w Spring Boot, najkrótszy czas wyniósł 5 ms, a najdłuższy 6 ms, przy odchyleniu standardowym 0,5026. Natomiast dla aplikacji napisanej w Laravel, najkrótszy czas wyniósł 234 ms, najdłuższy 294 ms, a odchylenie standardowe wyniosło 14,0611. Różnica między średnimi czasami obu aplikacji wyniosła około 4462%. W scenariuszu 2, badano czas wyświetlenia strony z 10 rekordami przez 1 użytkownika. Dla aplikacji napisanej w Spring Boot, najkrótszy zmierzony czas wyniósł 7 ms, a najdłuższy 13 ms, przy odchyleniu standardowym 1,6255. Dla aplikacji napisanej w Laravel, najmniejszy czas wyniósł 234 ms, a największy 264 ms, przy odchyleniu standardowym 9,9857. Różnica w średnich czasach wynosi około 2887%. W scenariuszu 3, zbadano czas wyświetlenia strony z 100 rekordami przez 1 użytkownika. Dla

aplikacji napisanej w Spring Boot, najkrótszy czas wyniósł 14 ms, a najdłuższy 18 ms, przy odchyleniu standardowym 1,0563. Czasy dla aplikacji napisanej w Laravel mieściły się w przedziale 242 ms do 291 ms, przy odchyleniu standardowym 13,2961. Różnica między średnimi czasami wyniosła około 1587%. W scenariuszu 4, zbadano czas wyświetlenia strony z 1000 rekordami przez 1 użytkownika. Dla aplikacji stworzonej w Spring Boot, najkrótszy czas wyniósł 92 ms, najdłuższy 108 ms, a odchylenie standardowe 3,3916. Dla aplikacji stworzonej w Laravel, najkrótszy czas wyniósł 311 ms, najdłuższy 423 ms, a odchylenie standardowe 26,6705. Różnica między średnimi wyniosła około 259%. W kolejnym scenariuszu, dotyczącym czasów wyświetlania strony z 10000 rekordami, dla aplikacji stworzonej w Spring Boot, najkrótszy zbadany czas wyniósł 885 ms, a najdłuższy 964 ms, przy odchyleniu standardowym 25,2662. Dla aplikacji napisanej w Laravel, najkrótszy czas wyniósł 1009 ms, najdłuższy 1080 ms, a odchylenie standardowe 15,4936. Różnica między średnimi wynosi około 14%. W scenariuszu 6, badano czasy wyświetlenia strony z 10 rekordami, gdy 10 użytkowników wyświetla ją jednocześnie. Dla aplikacji stworzonej w Spring Boot, najkrótszy czas wyniósł 4 ms, najdłuższy 5 ms, a odchylenie standardowe wyniosło 0,2236. Dla aplikacji stworzonej w Laravel, najkrótszy czas wyniósł 914 ms, najdłuższy 1017 ms, a odchylenie standardowe to 29,0171. Różnica między średnimi wyniosła około 23502%. W scenariuszu 7, badano czasy wyświetlania strony z 1000 rekordami, gdy 10 użytkowników wyświetla ją jednocześnie. Dla aplikacji napisanej w Spring Boot, minimalny czas wyniósł 87 ms, a maksymalny 94 ms, przy odchyleniu standardowym 1,8715. Dla aplikacji napisanej w Laravel, czasy mieszczą się w przedziale 1326 ms do 1646 ms, przy odchyleniu standardowym 107,0378. Różnica między średnimi wynosi około 1512%. W scenariuszu 8, badano czasy dodania rekordu przez 1 użytkownika. Dla aplikacji napisanej w Spring Boot, najkrótszy czas wyniósł 7 ms, najdłuższy 13 ms, a odchylenie standardowe 1,1459. Dla aplikacji napisanej w Laravel, najkrótszy czas wyniósł 493 ms, najdłuższy 577 ms, a odchylenie standardowe 20,8030. Różnica między średnimi czasami aplikacji wynosi 5044%. W scenariuszu 9, badano czasy dodania rekordu przez 10 użytkowników jednocześnie. Dla aplikacji stworzonej w Spring Boot, czasy mieszczą się w przedziale 10 ms do 11 ms, przy odchyleniu standardowym 0,4104. Dla aplikacji stworzonej w Laravel, najmniejszy czas wyniósł 3073 ms, a największy 3295 ms, przy odchyleniu standardowym 69,9165. Różnica między średnimi czasami wynosi 31047%. W scenariuszu 10, badano czasy aktualizacji 1 rekordu przez 1 użytkownika. Dla aplikacji napisanej w Spring Boot, czasy wynosiły od 8 ms do 13 ms, a odchylenie standardowe wyniosło 1,1367. Dla aplikacji napisanej w Laravel, czasy znajdowały się w przedziale od 493 ms do 560 ms, przy odchyleniu standardowym 17,2977. Różnica między średnimi czasami dla obu aplikacji wynosi 5804%. W scenariuszu 11, badano czas aktualizacji po 1 rekordzie, gdy 10 użytkowników aktu-

alizuje go jednocześnie. Dla aplikacji napisanej w Spring Boot, czasy mieszczą się w przedziale od 7 ms do 8 ms, a odchylenie standardowe wynosi 0,4443. Dla aplikacji napisanej w Laravel, czasy wynoszą od 3035 ms do 3665 ms, przy odchyleniu standardowym 193,9305. Różnica między średnimi czasami aplikacji wynosi 42442%. W scenariuszu 12, badano czasy usunięcia rekordu. Dla aplikacji napisanej w Spring Boot, najkrótszy czas wynosi 14 ms, najdłuższy 22 ms, a odchylenie standardowe wyniosło 1,9861. Dla aplikacji napisanej w Laravel, najkrótszy czas wynosi 483 ms, najdłuższy 564 ms, a odchylenie standardowe to 19,5024. Różnica między średnimi czasami wynosi 2901%. W każdym z 12 scenariuszy Spring Boot osiągnął krótsze czasy. Najmniej różnic można zaobserwować w scenariuszu 5, gdzie jeden użytkownik wyświetlał stronę z dużą ilością danych z bazy. W tym przypadku średnie czasy różniły się o około 122 ms. Natomiast największe rozbieżności odnotowano w scenariuszu 11, w którym mierzono czasy aktualizacji rekordu przez 10 użytkowników jednocześnie, gdzie średnie czasy różniły się o około 3289 ms. Podczas wyświetlania strony z danymi przez jednego użytkownika wraz ze wzrostem liczby danych malała różnica między badanymi szkieletami. Scenariusze dotyczące średnich czasów dodawania rekordu przez 1 użytkownika oraz 10 użytkowników jednocześnie pokazują, że czasy dla aplikacji zbudowanej w szkielecie Spring Boot nie różnią się znacząco, natomiast czasy dla aplikacji stworzonej w szkielecie Laravel wzrosły ponad sześciokrotnie. Podobne wyniki można zaobserwować dla średnich czasów aktualizacji rekordu przez 1 użytkownika oraz 10 użytkowników jednocześnie.

5.2. Metryki aplikacji

Zestawienie metryk aplikacji przedstawia Tabela 3. Porównano w niej ogólny rozmiar projektu dla obu aplikacji, liczbę plików w obu projektach, liczbę linii kodu języka Java i PHP, które napisał autor aplikacji oraz liczbę linii kodu dla silników szablonów Thymeleaf oraz Blade, które również napisał autor aplikacji. Przed rozpoczęciem pomiarów aplikacje zostały uszczuplone o foldery z plikami konfiguracyjnymi, które mogłyby znacznie zaburzyć wyniki. W przypadku aplikacji zbudowanej przy pomocy szkieletu Spring Boot odrzucono foldery takie jak „target”, natomiast w przypadku aplikacji budowanej na bazie frameworka Laravel usunięty został między innymi folder „vendor”. Do pomiarów użyto wtyczki Statistic, która jest dostępna zarówno w IntelliJ IDEA, jak i PhpStorm.

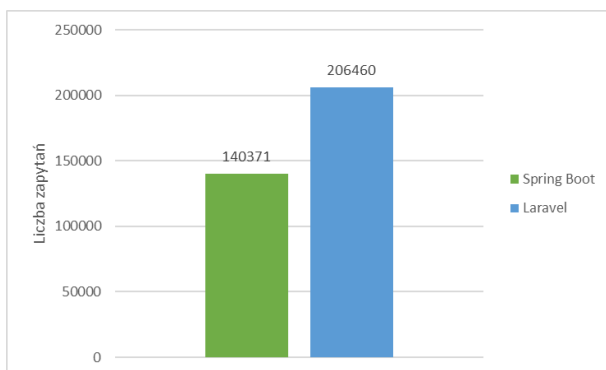
Tabela 3: Metryki kodu aplikacji testowych

	Spring Boot	Laravel
Rozmiar projektu na dysku [KB]	136	540
Liczba plików w projekcie	24	82
Liczba linii kodu Java/PHP napisanych przez autora	150	119
Liczba linii kodu Thymeleaf/Blade	241	312

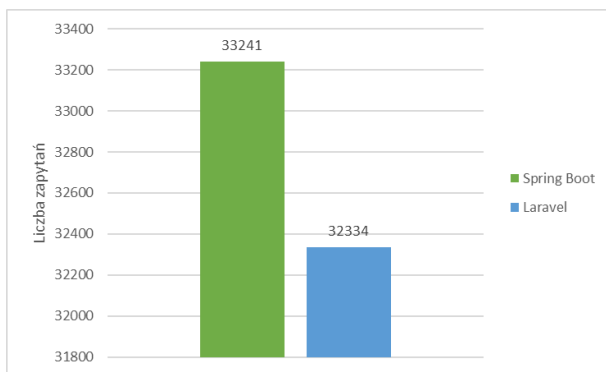
Projekt zbudowany na szkielecie Spring Boot zajął prawie czterokrotnie mniej miejsca na dysku od projektu, do którego wykorzystano szkielet Laravel. Liczba plików w projekcie oraz liczba linii kodu Thymeleaf/Blade również są mniejsze w przypadku szkieletu Spring Boot o odpowiednio 58 oraz 71. Jedynie w przypadku liczby linii kodu Java/PHP napisanych przez autora szkielet Laravel osiągnął wynik mniejszy o 31.

5.3. Wsparcie społeczności

Badanie polegało na porównaniu liczby zapytań dla obu szkieletów w serwisie społecznościowym Stack Overflow, który przeznaczony jest do zadawania pytań dotyczących wytwarzania programowania. Wyniki wszystkich zapytań o dany szkielet zostały przedstawione na Rysunku 15. Na Rysunku 16 przedstawiono wyniki dla zapytań o szkielet, które nie miały żadnej odpowiedzi.



Rysunek 15: Liczba zapytań w serwisie stackoverflow.com.



Rysunek 16: Liczba zapytań w serwisie stackoverflow.com bez odpowiedzi.

Użytkownicy serwisu stackoverflow.com częściej pytali o szkielet aplikacji Laravel. Różnica w liczbie zapytań wyniosła 66089. Zapytania bez odpowiedzi stanowiły 24% wszystkich zapytań w przypadku szkieletu Spring i 16% w przypadku szkieletu aplikacji Laravel.

6. Wnioski

Przeprowadzone badania pokazują, że aplikacja zbudowana na szkielecie Spring Boot ma znacznie lepszą wydajność od tej zbudowanej na szkielecie Laravel. Tym samym postawiona teza: *Aplikacja internetowa stworzona przy użyciu szkieletu Spring Boot wykazuje lepszą wydajność niż analogiczna aplikacja zbudowana przy użyciu szkieletu Laravel.*

Pierwsze szczegółowe pytanie badawcze brzmiało: *Czy aplikacja stworzona w szkielecie Spring Boot wyświetla szybciej zasoby strony niż aplikacja w szkielecie Laravel?* Odpowiedź uzyskano na podstawie przeprowadzonych badań wydajności, gdzie Spring Boot osiągnął krótsze czasy wyświetlania danych dla każdego scenariusza przedstawiającego tę operację. Wyniki pokazują również, że przy większym zestawie danych wyświetlanych przez jednego użytkownika różnica w czasie wyświetlenia zasobów była mniejsza.

Drugim szczegółowym pytaniem badawczym było: *Czy aplikacja stworzona przy użyciu Spring Boot ma mniejszy rozmiar od aplikacji stworzonej używając platformy Laravel?* Analiza metryk kodu wykazała, że aplikacja zbudowana z pomocą szkieletu Spring Boot zajmuje prawie czterokrotnie mniej miejsca na dysku, niż identyczna aplikacja napisana w szkielecie Laravel. Również w przypadku liczby plików w projekcie oraz liczba linii kodu Thymeleaf/Blade aplikacja stworzona w szkielecie Spring Boot osiąga mniejsze wyniki. Natomiast aplikacja stworzona w szkielecie Laravel osiągnęła mniejszy wynik dla liczby linii kodu Java/PHP napisanych przez autora.

Trzecia hipoteza badawcza brzmiała: *Czy szkielet aplikacji Spring Boot ma większe wsparcie społeczności niż szkielet aplikacji Laravel?* Analiza została przeprowadzona w oparciu o zapytania w serwisie stackoverflow.com. Wyniki wskazały, że użytkownicy częściej pytali o szkielet aplikacji Laravel, a także chętniej odpowiadali na pytania dotyczące tego szkieletu aplikacji.

Przeprowadzone badania oraz otrzymane wyniki mogą być pomocne dla programistów w wyborze technologii do tworzenia aplikacji internetowych.

Literatura

- [1] C. L. Hsu, K. C. Chang, M. C. Chen, The impact of website quality on customer satisfaction and purchase intention: perceived playfulness and perceived flow as mediators, *Information Systems and e-Business Management* 10 (2012) 549-570.
- [2] J. Radomski, Porównanie wydajności aplikacji internetowych na przykładzie szkieletów programistycznych Laravel i Vaadin, *Journal of Computer Sciences Institute* 22 (2022) 35-39.
- [3] Paweł Rzeńca, Porównanie wytwarzania oprogramowania internetowego z wykorzystaniem różnych technologii, *Repozytorium PJATK, Warszawa*, 2022.
- [4] 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 (2018) 555-564.
- [5] H. K. Dhalla, A Performance Comparison of RESTful Applications Implemented in Spring Boot Java and MS. NET Core, In *Journal of Physics: Conference Series*, IOP Publishing (2021) 2-7.
- [6] L. R. Abbade, M. A. da Cruz, J. J. Rodrigues, P. Lorenz, R. A. Rabelo, J. Al-Muhtadi, Performance comparison of programming languages for Internet of Things

- middleware, Transactions on Emerging Telecommunications Technologies 31 (2020) 522-532.
- [7] U. Latif, T. Kusumasari, Comparison Between Yii Frameworks and Laravel in 3 Different Version for Viewing Large Data of Shipyard Industry in Indonesia, International Journal of Innovation in Enterprise System 2 (2018) 13-18.
- [8] M. Laaziri, K. Benmoussa, S. Khouliji, M. L. Kerkeb, A Comparative study of PHP frameworks performance, Procedia Manufacturing 32 (2019) 864-871.
- [9] P. R. Chavan, S. Pawar, Comparison Study Between Performance of Laravel and Other PHP Frameworks, International Journal of Research in Engineering, Science and Management 4 (2021) 27-29.
- [10] O. W. Purbo, A Systematic Analysis: Website Development using Codeigniter and Laravel Framework, Enrichment: Journal of Management 12 (2021) 1008-1014.
- [11] A. Sunardi, MVC architecture: A comparative study between laravel framework and slim framework in freelancer project monitoring system web based, Procedia Computer Science 157 (2019) 134-141.
- [12] M. Gajewski, W. Zabierowski, Analysis and comparison of the Spring framework and play framework performance, used to create web applications in Java, IEEE XVth International Conference on the Perspective Technologies and Methods in MEMS Design (2019) 170-173.
- [13] A. Poudel, A comparative study of project management system web applications built on ASP. Net core and laravel MVC frameworks, The Repository @ St. Cloud State, Minnesota, 2018.
- [14] M. Kaluža, M. Kalanj, A comparison of back-end frameworks for web application development, Computer Science 7 (2019) 317-332.
- [15] J. Muittari, Modern web back-end, Oulu University of Applied Sciences Information Technology, Oulu, 2020.
- [16] Stack Overflow – serwis społecznościowy, <https://stackoverflow.com>, [10.06.2023].