

Symfony and Laravel – a comparative analysis of PHP programming frameworks

Symfony i Laravel – analiza porównawcza szkieletów programistycznych języka PHP

Krzysztof Kuflewski*, Mariusz Dzieńkowski

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

Abstract

This paper is a comparative analysis of PHP programming frameworks - Symfony and Laravel. The analysis was conducted on two test applications prepared for this purpose, based on the latest versions of the following technologies: Symfony 5.2 and Laravel 8. Both applications, being simple auction systems, have the same set of functionalities. They were compared in terms of selected criteria. Their implementation process, software metrics, performance and amount of community support were compared. Apache jMeter was used for performance testing. With its help, tests of several operations on databases were performed. The operations were as follows: adding auctions, retrieving auction details, editing, deleting auctions, bidding on an auction and simultaneous closing 1,000 auctions. The test results for the selected criteria were proved to be better for the Laravel framework based application.

Keywords: web applications; PHP frameworks; Symfony; Laravel

Streszczenie

Przedmiotem pracy jest analiza porównawcza szkieletów programistycznych języka PHP - Symfony i Laravela. Przeprowadzono ją na dwóch przygotowanych do tego celu aplikacjach testowych, opartych na najnowszych wersjach badanych technologii: Symfony 5.2 oraz Laravel 8. Obie aplikacje, będące prostym systemem aukcyjnym, posiadają ten sam zestaw funkcjonalności. Zostały one porównane pod względem wybranych kryteriów. Porównywano proces ich implementacji, metryki oprogramowania, wydajność oraz wielkość wsparcia społeczności. Do testowania wydajności wykorzystano program Apache jMeter. Z jego pomocą wykonano testy kilku operacji na bazie danych takich jak: dodawanie aukcji, pobieranie szczegółów aukcji, edycja, usunięcie aukcji, złożenie oferty na aukcję oraz jednoczesne zamknięcie 1000 aukcji. Wyniki badań dla wybranych kryteriów okazały się lepsze dla aplikacji opartej na szkielecie Laravel.

Słowa kluczowe: aplikacje internetowe; szkielety programistyczne języka PHP; Symfony; Laravel

*Corresponding author

Email address: krzysztof.kuflewski@pollub.edu.pl (K. Kuflewski)

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

1. Wstęp

1.1. Aplikacje internetowe i szkielety programistyczne

Ostatnie lata przyniosły zdecydowany wzrost zapotrzebowania na oprogramowanie komputerowe. Dużą popularnością cieszą się aplikacje internetowe - programy działające na serwerze internetowym, z których użytkownicy korzystają poprzez przeglądarkę internetową. Obecnie takie aplikacje stawiane są na równi z desktopowymi - powinny dostarczać jednakowych funkcjonalności, być wszechstronne, bezpieczne i szybko działające. Niewątpliwą zaletą aplikacji internetowych jest ich łatwa przyswajalność. Do użytkowania wystarczy połączenie z siecią i przeglądarka internetowa - nie ma więc potrzeby instalowania dodatkowego oprogramowania na urządzeniu. Postęp tego typu aplikacji przyczynił się do rozwoju technologii wykorzystywanych do ich tworzenia. Programiści mogą wybierać z wielu języków programowania pozwalających budować coraz bardziej zaawansowane aplikacje internetowe. W celu usprawnienia i przyspieszenia pracy nad tworzeniem

programów dla wielu języków powstały szkielety programistyczne (ang. frameworks).

Szkielet programistyczny jest platformą do konstruowania aplikacji. Definiuje on jej strukturę oraz mechanizmy działania. Pomimo, że narzuca on sposób tworzenia aplikacji, w zamian programista dostaje gotowe biblioteki i rozwiązania, co znacznie przyspiesza pracę nad programem. Deweloper nie musi trudzić się rozwiązywaniem wielu powtarzalnych problemów i może w pełni skupić się na implementacji funkcjonalności. Rynek jest obecnie bardzo bogaty w szkielety programistyczne, a programiści stają przed dylematem, który z nich wybrać. Pomimo tego, że priorytetem wszystkich szkieletów jest usprawnienie implementacji, zapewnienie bezpieczeństwa aplikacji i wsparcie społeczności, każdy szkielet oferuje inne funkcje, rozwiązania i narzędzia. Dlatego też kluczową kwestią jest wybór odpowiedniego szkieletu do budowy planowanego projektu. Kierując się tą myślą, w niniejszej pracy przeprowadzono analizę porównawczą dwóch najbardziej popularnych szkieletów programistycznych języka PHP: Symfony i Laravel.

1.2. Przegląd literatury

Popularność aplikacji internetowych oraz duża liczba narzędzi wykorzystywanych do ich tworzenia doprowadziły do licznych rozważań nad wyborem najbardziej odpowiedniej technologii. Powstało wiele publikacji naukowych traktujących o tworzeniu aplikacji internetowych, technologiach stosowanych podczas ich programowania, czy narzędziach do testowania takich aplikacji. W wielu pracach, ich autorzy dążą do wskazania najlepszego – w danym aspekcie – rozwiązania.

Szkielety programistyczne Symfony i Laravel były już ze sobą porównywane. W publikacji *A comparative study of laravel and symfony PHP frameworks* autorzy stawiają tezę, że Symfony jest lepszym rozwiązaniem [1]. Podczas badań wzięto pod uwagę następujące kryteria: popularność, architekturę, budowę i cechy szkieletu, organizację kodu oraz wymagania systemowe. Twórcy na potrzeby analizy nie tworzyli aplikacji w badanych technologiach, a co się z tym wiąże, nie brali pod uwagę ich wydajności. Również nie są wskazane wersje badanych szkieletów, co może być istotną kwestią podczas implementacji. We wnioskach autorzy stwierdzają, iż wybór technologii zależy od potrzeb: Symfony jest lepszym rozwiązaniem do tworzenia dużych projektów, natomiast Laravel ma niższy próg wejścia, przez co może być częściej wybierany przez osoby zaczynające swoją przygodę z szkieletami programistycznymi języka PHP.

W publikacji *Practical Application of PHP Frameworks in Development of Web Information Systems* autorzy opisują kilka szkieletów programistycznych języka PHP: CakePHP2, CodeIgniter, Symfony 2, Yii i PhalconPHP, wskazując główne różnice w ich budowie i charakterystyce, a także wykonują testy wydajnościowe aplikacji wykonanych w dwóch szkieletach: Symfony2 oraz PhalconPHP [2]. Przedmiotem analizy w przypadku wydajności jest prosty system rezerwacji biletów, na którym wykonano po cztery scenariusze testowe dla każdej z technologii, a narzędziem do wykonania testów wydajnościowych był Apache Benchmark. Dokonano pomiarów czasów odpowiedzi na wysyłane żądania, zliczono żądania poprawnie obsłużone, zmierzono zużycie pamięci oraz liczbę plików, z których składała się dana aplikacja. Wyniki analizy wykazały, że PhalconPHP jest dużo wydajniejszy niż Symfony 2. Natomiast szkielet Symfony doceniono za szeroką ofertę możliwości, które znacznie ułatwiają i przyspieszają proces implementacji.

Autorzy pracy *A new model for the selection of web development frameworks: application to PHP frameworks* zbadali cztery technologie: Symfony, Laravel, Zend, CodeIgniter bez konkretnie określonych wersji [3]. Celem ich analizy była pomoc w wyborze - przez potencjalnego programistę - odpowiedniego narzędzia do wykonania danej aplikacji. Jako kryteria porównawcze wybrane zostały: ogólne informacje o szkielecie, jego cechy charakterystyczne i budowa, czas wykonania projektu opartego na danym szkielecie oraz wydajność rozumiana jako liczba obsługiwanych żądań na sekundę. W każdej z porównywanych technologii wykonana

została odrębna aplikacja wykorzystana do testów. W pracy nie ma informacji, jakie to były aplikacje. Do badań zostało użyte narzędzie ApacheBench – specjalistyczny program do wykonywania testów wydajnościowych serwerów HTTP. W podsumowaniu autorzy stwierdzają, iż każdy ze szkieletów ma swoje zalety, a wybór zależy od indywidualnych potrzeb programistów oraz specyfiki realizowanego projektu.

Technologie wytwarzania aplikacji internetowych, w tym szkielety programistyczne języka PHP, były i ciągle są ze sobą porównywane. Autorzy prac w swoich badaniach skupiali się na różnych aspektach tych technologii. Badano m.in.: główne założenia technologii, ich budowę i popularność. Na potrzeby niektórych analiz tworzono aplikacje testowe, które służyły jako obiekty badań głównie wydajności. W swoich pracach autorzy dążą do wskazania lepszego – pod danym względem – narzędzia. Niniejsza praca również wpisuje się w ten trend, koncentrując się na badaniu procesu implementacji, wydajności oraz metryk kodu programu, wykorzystując do tego celu utworzoną na potrzeby badań prostą aplikację bazodanową realizującą operacje CRUD, czyli zapis, odczyt, aktualizacja, usuwanie (ang. Create, Read, Update, Delete). Analizom poddano również wielkość wsparcia społeczności danej technologii. Spośród innych publikacji pracę tą wyróżniają wersje badanych szkieletów programistycznych, zaimplementowana aplikacja testowa, opracowane scenariusze testowe, a także narzędzie użyte do testowania.

1.3. Cel badań

Celem badań jest porównanie dwóch szkieletów programistycznych języka PHP - Symfony oraz Laravel. Na potrzeby rozważań stworzono dwie jednakowe aplikacje oparte na najnowszych wersjach szkieletów - Symfony 5.2 oraz Laravel 8. Przeprowadzona na podstawie opracowanych aplikacji analiza pod kątem wybranych kryteriów pozwoli wyciągnąć odpowiednie wnioski i pomóc w wyborze właściwego narzędzia do stworzenia aplikacji internetowej.

2. Metodologia badań

2.1. Aplikacje testowe

Do wykonania badania niezbędne było przygotowanie aplikacji testowej w dwóch egzemplarzach - jednej zbudowanej na bazie szkieletu Laravel 8 oraz drugiej w Symfony 5.2. Obie aplikacje wykonano w jednakowy sposób - zawierają one taki sam zestaw funkcjonalności, korzystają z tej samej bazy danych, posiadają bardzo podobne widoki. Aplikacją jest prosty system aukcyjny, umożliwiający realizację podstawowych operacji CRUD na bazie danych oraz uwierzytelnianie użytkowników. System rozróżnia dwa rodzaje użytkowników: gościa oraz użytkownika zalogowanego.

Wymagania funkcjonalne aplikacji testowych:

- Użytkownik niezalogowany:
 - rejestracja w systemie,
 - logowanie do systemu,
 - przeglądanie listy aukcji,

- przeglądanie szczegółów aukcji.
- Użytkownik zalogowany:
 - tworzenie, edycja i usuwanie własnych aukcji,
 - składanie ofert do aukcji,
 - wylogowanie.

Struktura bazy danych

Baza danych składa się z trzech tabel:

- *users* - opisującej użytkowników zarejestrowanych w systemie,
- *auctions* - opisującej aukcje stworzone przez użytkowników,
- *offers* - opisującej złożone przez użytkowników oferty w aukcjach.

Tabele zostały powiązane odpowiednimi relacjami:

- *users* i *auctions* – **1:n** - jeden użytkownik może utworzyć wiele aukcji,
- *users* i *offers* – **1:n** - jeden użytkownik może złożyć wiele ofert,
- *auctions* i *offers* – **1:n** – jedna aukcja może posiadać wiele ofert.

2.2. Środowisko testowe

W celu zapewnienia jak najbardziej wiarygodnych wyników badań wydajnościowych dla obydwu technologii, aplikacje zostały uruchomione w jednym środowisku testowym, z wykorzystaniem pakietu XAMPP. Korzystają one również z jednakowej bazy danych MySQL. Dane środowiska testowego przedstawia Tabela 1.

Tabela 1: Konfiguracja środowiska testowego aplikacji

Procesor	Intel Core i5-5200U
System operacyjny	Windows 10 Home 64-bit
Pamięć RAM	8 GB
Przeglądarka internetowa	Google Chrome 89.0.4389.128
Pakiet XAMPP	7.4.10
Serwer Apache	2.4.46
MySQL	7.4.10
PHP	8
Narzędzie testowe	Apache JMeter 5.4.1

Narzędziem testowym do badania wydajności aplikacji był Apache JMeter. Jest to dedykowane oprogramowanie do wykonywania testów wydajnościowych, obciążeniowych i funkcjonalnych na aplikacjach internetowych. Jego przeznaczeniem jest badanie wydajności statycznych oraz dynamicznych zasobów: plików, dynamicznych języków programowania serwisów internetowych, np.: Java, PHP, ASP.NET, obiektów Java, kwerend, baz danych, serwerów FTP [4].

2.3. Kryteria porównawcze i scenariusze testowe

W celu wykonania badań oraz dokonania analizy porównawczej omawianych w pracy szkieletów programistycznych, niezbędne było wybranie kryteriów, pod względem których pośrednio szkielety, a bezpośrednio aplikacje zostały sprawdzone. Do badania wydajności aplikacji opracowano scenariusze testowe, które pozwo-

liły zweryfikować wydajność aplikacji w trakcie wykonywania określonych zadań.

Proces implementacji aplikacji

Sposób tworzenia aplikacji w danej technologii ma duże znaczenie przy jej wyborze do realizacji określonego projektu. Porównany został mechanizm tworzenia aplikacji bazodanowej przy użyciu analizowanych szkieletach programistycznych na przykładzie systemu aukcyjnego. Wynikiem tych porównań jest zestawienie podobieństw oraz różnic. Nie skupiono się na przebiegu wszystkich etapów budowy aplikacji testowych, ale na tych kluczowych - będących charakterystycznymi elementami w obrębie badanego szkieletu.

Metryki aplikacji - rozmiar projektu, objętość kodu źródłowego

Rozmiar projektu ma wpływ na jego wydajność, a objętość kodu źródłowego jest czynnikiem wpływającym na pracochłonność implementacji aplikacji. Mniejsza ilość kodu to także większa czytelność i krótszy czas niezbędny do wytworzenia oprogramowania, co w konsekwencji sprowadza się do niższego kosztu.

Wydajność

Wydajność jest bardzo istotnym elementem aplikacji internetowych. Twórcom oprogramowania często zależy na sprawnym działaniu aplikacji i szybkim wykonywaniu zadań. W celu zbadania wydajności przygotowano scenariusze testowe, które pozwolą sprawdzić, jak w tym zakresie radzą sobie szkielety Laravel 8 i Symfony 5.2.

Opracowane scenariusze testowe dotyczyły prostych działań realizowanych przez użytkowników na aplikacjach testowych, podczas których rejestrowany był czas ich wykonania. Scenariusze obejmowały następujące czynności:

- dodanie aukcji,
- pobranie szczegółów aukcji,
- edycja aukcji,
- usunięcie aukcji,
- złożenie oferty do aukcji,
- zamknięcie 1000 aukcji.

Ponadto przygotowano trzy scenariusze, które wykorzystano do analizy czasów wyświetlania stron zawierających dane:

- wyświetlenie listy 100 aukcji,
- wyświetlenie szczegółów pojedynczej aukcji,
- wyświetlenie formularza dodawania aukcji.

Wsparcie społeczności

Specjalistyczne forum skupiające wokół siebie dużą liczbę osób korzystających z danej technologii, to grupa która może pomóc w sytuacjach, gdy dokumentacja z jakichś powodów okaże się niewystarczająca. Osoby te, dzięki swojemu zaangażowaniu, mogą odpowiadać na nurtujące pytania, rozwiązywać trudne problemy oraz wyjaśniać różne wątpliwości. Badanie wsparcia społeczności polegało na sprawdzeniu liczby zapytań dotyczących porównywanych technologii. Wykorzystano do tego celu jedno z najbardziej popularnych forów

programistycznych – StackOverflow.com [5]. Dodatkowo pod uwagę wzięto liczbę pytań posiadających co najmniej jedną odpowiedź, co zwiększa prawdopodobieństwo znalezienia rozwiązania dla danego problemu.

2.4. Proces implementacji

Implementacja aplikacji opartych na szkieletach Symfony 5.2 i Laravel 8 odbywa się w podobny sposób – językiem bazowym obydwu z nich jest PHP, obydwie oparte są o wzorzec projektowy MVC (ang. Model-View-Controller), struktura projektu jest także analogiczna. Zarówno Symfony jak i Laravel posiadają aplikację konsolową umożliwiającą tworzenie m.in. kontrolerów, modeli i widoków. Obydwie szkielety korzystają z systemu migracji i posiadają swoje pakiety do uwierzytelniania użytkownika, które wystarczy zainstalować i dopasować do swoich potrzeb. Szablony Symfony (twig) są bardzo podobne do szablonów Laravel'a (blade). Definicja rutingów w pliku kontrolera bezpośrednio przy funkcji obsługującej daną akcję w Symfony wydaje się bardziej czytelna niż rutingi w oddzielnym pliku w przypadku Laravela. Laravel poprzez brak repozytoriów, bardziej zwięzły kod przy działaniach z bazą danych, oraz brak definicji formularzy jako obiektów PHP jest z pewnością lepszą opcją dla początkującego programisty rozpoczynającego pracę ze szkieletami języka PHP niż Symfony.

2.5. Metryki aplikacji

Rozmiar kodu projektów aplikacji testowych

Badanie rozmiaru projektów aplikacji testowych wykazało, że projekt Laravela jest o około 16% mniejszy niż Symfony (odpowiednio: 38Mb, 45Mb). Pod uwagę nie brano katalogu *node_modules* zawierającego pliki bibliotek strony klienta i potrzebnego przy użyciu technologii Tailwind.css, odpowiedzialnej za warstwę wizualną aplikacji testowych.

Objętość kodu źródłowego

Objętość kodu źródłowego aplikacji testowych zbadano przy pomocy metryki SLOC (ang. Source Lines of Code), używanej do pomiaru rozmiaru oprogramowania komputerowego. Polega ona na zliczeniu linii kodu źródłowego, co może się przełożyć na ilość pracy potrzebnej do wykonania programu, a także jego dalszego utrzymania [6]. Badanie zostało wykonane z pomocą biblioteki *npm* o nazwie *sloc*. Testom poddano tylko te obszary aplikacji, które zostały zaimplementowane przez programistę. Nie badano plików bibliotek i innych plików generowanych automatycznie.

Tabela 2: Liczba linii kodu w poszczególnych częściach aplikacji testowych w badanych szkieletach

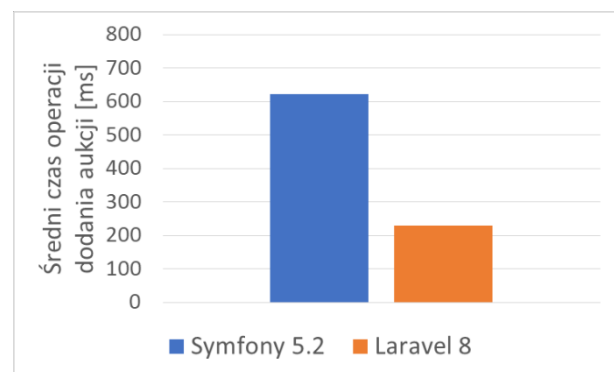
	Symfony 5.2	Laravel 8
Modele	362	75
Kontrolery	160	225
Widoki	431	456
Formularze	68	-
Rutingi	-	7
Migracje	-	91
Suma	1021	854

Wyniki zostały przedstawione w Tabeli 2. Zawiera ona liczbę linii kodu źródłowego w plikach odpowiedzialnych za poszczególne części stworzonej aplikacji. Brak plików z danego obszaru lub brak konieczności implementowania ich w danym szkielecie oznaczono znakiem "-".

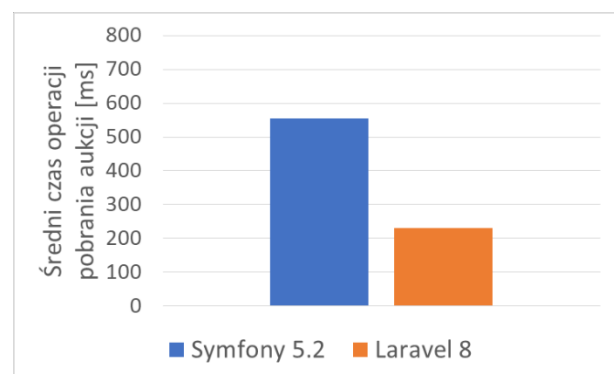
Badanie metryki SLOC wykazało, że aplikacja wykonana za pomocą szkieletu Laravel ma mniejszą ilość kodu o około 16%. Przy dużych projektach taka różnica może okazać się kluczową kwestią wpływającą na czas wykonania oprogramowania, a także jego późniejsze utrzymanie.

2.6. Wydajność

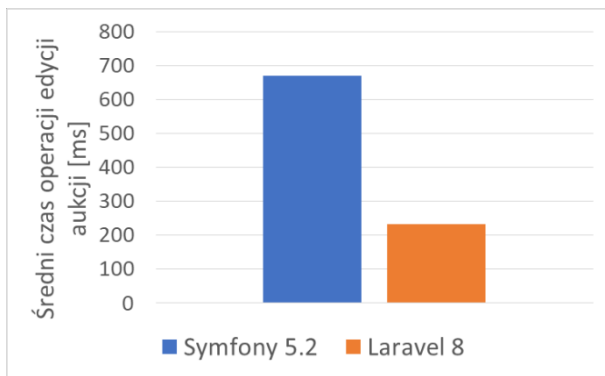
Każdy ze scenariuszy testowych został powtórzony 10 razy dla każdej aplikacji, na jednakowych zestawach danych. Na Rysunkach 1 - 6 przedstawiono wyniki w postaci średnich czasów wykonania operacji wyrażonych w milisekundach. Natomiast Rysunki 7 - 9 prezentują średnie czasy wyświetlania widoków zawierających dane (w sekundach).



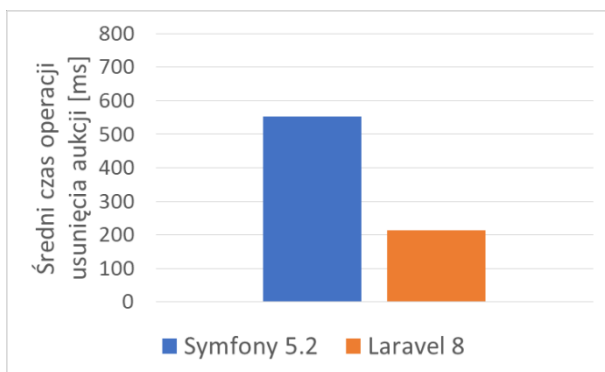
Rysunek 1: Wyniki dla scenariusza 1 – dodanie aukcji.



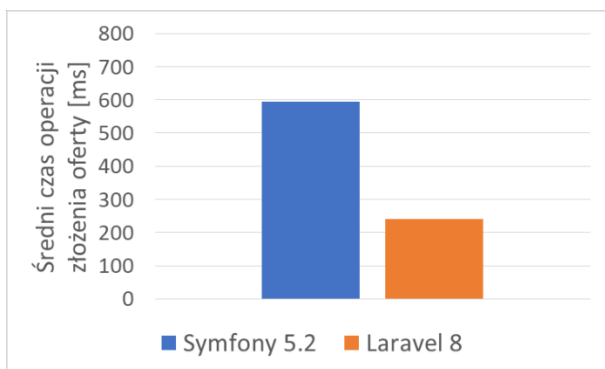
Rysunek 2: Wyniki dla scenariusza 2 – pobranie szczegółów aukcji.



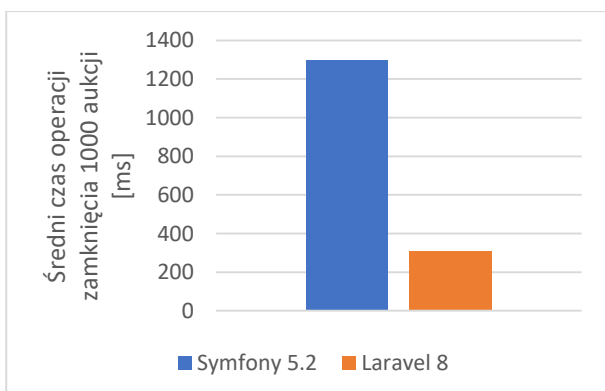
Rysunek 3: Wyniki dla scenariusza 3 – edycja aukcji.



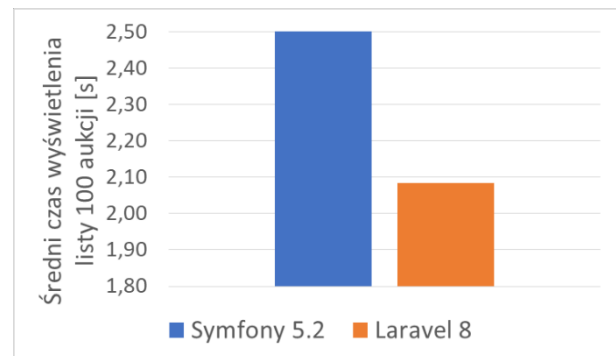
Rysunek 4: Wyniki dla scenariusza 4 – usunięcie aukcji.



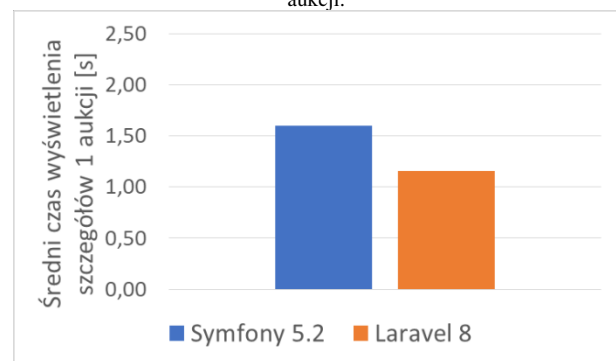
Rysunek 5: Wyniki dla scenariusza 5 – złożenie oferty na aukcję.



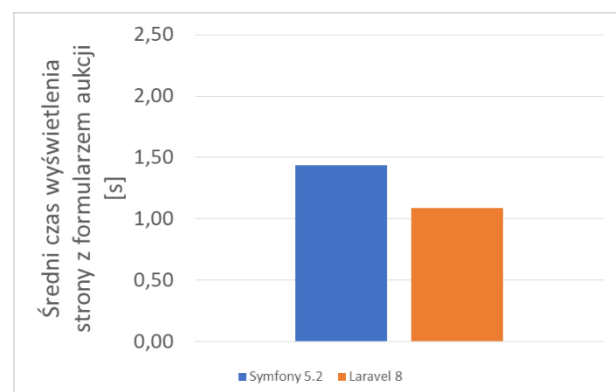
Rysunek 6: Wyniki dla scenariusza 6 – zamknięcie 1000 aukcji.



Rysunek 7: Wyniki dla scenariusza 7 – wyświetlenie strony z listą 100 aukcji.



Rysunek 8: Wyniki dla scenariusza 8 – wyświetlenie strony z pobranymi szczegółami jednej aukcji.

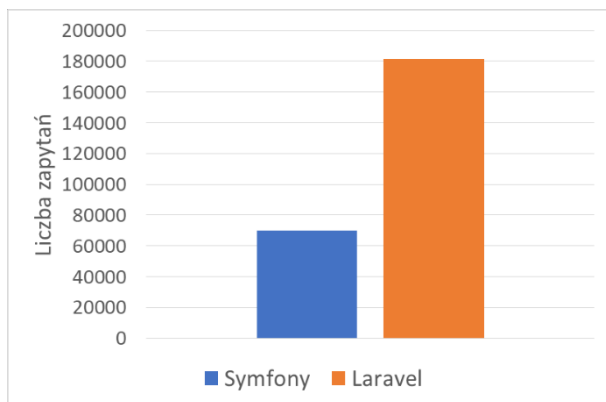


Rysunek 9: Wyniki dla scenariusza 9 – wyświetlenie strony z formularzem edycji aukcji.

W scenariuszach 1-5 dotyczących czasów wykonania operacji (dodanie aukcji, pobranie szczegółów aukcji, edycja aukcji, usunięcie aukcji, złożenie oferty na aukcję) Laravel był średnio 2,6 razy szybszy od Symfony. W przypadku scenariusza 6 (zamknięcie 1000 aukcji) różnica czasów była jeszcze większa – Laravel był ponad cztery razy szybszy od Symfony. Było to związane z brakiem w ORM (ang. Object-Relational Mapping) Doctrine, z którego korzysta Symfony, możliwości jednoczesnego zaktualizowania wielu rekordów (zamknięcie aukcji jest zmianą jej statusu w bazie danych). Z kolei taką możliwość daje ORM Eloquent, z którego korzysta Laravel. W przypadku wyświetlania stron z danymi (scenariusze 7 - 9) różnica pomiędzy badanymi szkieletami jest mniejsza, ale wciąż na korzyść Lavela. Wyświetlanie stron z danymi trwało w tym przypadku średnio o 30% krócej.

2.7. Wsparcie społeczności

W celu określenia wsparcia społeczności zbadano liczbę zapytań dotyczących obu narzędzi w serwisie StackOverflow.com. Wyniki przedstawiono na rysunku 10.



Rysunek 10: Liczba zapytań dla badanych szkieletów w serwisie StackOverflow.com.

Liczba wyników zapytań dla szkieletu Laravel (181341) była zdecydowanie wyższa niż w przypadku Symfony (69727). Było to około 2,6 razy więcej na korzyść Laravla w stosunku do Symfony. Odsetek pytań zawierających co najmniej jedną odpowiedź był na podobnym poziomie dla obydwu technologii: dla Laravla wynosił 84% (152326), a dla Symfony 85% (59268). Z dużo większej liczby pytań z co najmniej jedną odpowiedzią dla szkieletu Laravel można wysnuć wniosek, że platforma ta budzi większe zainteresowanie u programistów i stwarza większe prawdopodobieństwo znalezienia pomocy wśród bardzo licznej społeczności.

3. Wnioski

Rynek aplikacji internetowych rozwija się bardzo prędko – zapotrzebowanie na aplikacje rośnie, a samymi programom stawia się coraz wyższe wymagania. Zwiększa się także liczba technologii do tworzenia aplikacji www, a te istniejące, cały czas są rozwijane. Z tego względu programista staje przed trudnym wyborem – jaką technologię wybrać do swojego projektu.

Celem niniejszej pracy było porównanie dwóch najbardziej popularnych szkieletów programistycznych języka PHP: Symfony i Laravla. Do celów badawczych przygotowano dwie aplikacje testowe w najnowszych (w momencie rozpoczęcia badań) wersjach tych szkieletów. Aplikacje zostały zbudowane w podobny sposób, zgodnie z założeniami danego narzędzia. Posiadały one jednakowy zestaw funkcjonalności, a także korzystały z tych samych danych i zostały uruchomione w tym samym środowisku testowym. Aplikacje testowe poddano badaniom w obrębie wyselekcjonowanych kryteriów porównawczych i scenariuszy testowych.

Wyniki przeprowadzonych badań wypadają na korzyść szkieletu Laravel 8. Implementacja prostej aplikacji typu CRUD była łatwiejsza, a projekt zajmował mniej miejsca na dysku i składał się z mniejszej liczby linii kodu źródłowego. Analiza wydajności takiej aplikacji zrealizowana na podstawie czasów wykonania operacji CRUD oraz wyświetlania stron z danymi wykazała, że Laravel 8 okazał się szkieletem wydajniejszym niż Symfony 5.2. Kryterium wsparcia społeczności, wskazuje na szerokie zainteresowanie Laravelem. Należy przy tym dodać, że interpretacja tego kryterium może być także inna. Duża społeczność i jej aktywność może oznaczać, że platforma ta sprawia więcej problemów oraz zawiera wiele mankamentów i wymaga wielu zabiegów, aby je naprawiać.

Z przeprowadzonych w ramach pracy badań na prostych aplikacjach testowych według dobranych przez autorów kryteriów wynika, że Laravel jest lepszym rozwiązaniem do tworzenia prostych aplikacji współpracujących z bazą danych niż Symfony. Jednak, aby jednoznacznie stwierdzić, który szkielet jest lepszy należałoby wykonać testy na średnich i dużych systemach. Poza tym w badaniach należałoby uwzględnić więcej aspektów tych narzędzi, ponieważ są one bardzo rozbudowane.

Literatura

- [1] M. Laaziri, K. Benmoussa, S. Khouli, K. M. Larbi, A. El Yamami, A comparative study of laravel and symfony PHP frameworks, *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1) (2019), <http://ijece.iaescore.com/index.php/IJECE/article/view/11601/11103>, [01.09.2021].
- [2] U. K. 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(1) (2018) 02-13, <https://ijies.sie.telkomuniversity.ac.id/index.php/IJIES/article/view/12/146>, [01.09.2021].
- [3] K. Benmoussa, M. Laaziri, S. Khouli, K. M. Larbi, A. El Yamami, A new model for the selection of web development frameworks: application to PHP frameworks, *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1) (2019), <http://ijece.iaescore.com/index.php/IJECE/article/view/11586/11102>, [10.09.2021].
- [4] Apache JMeter – opis, <https://www.slideshare.net/sjsi/jmeter-narzdzie-testera>, [20.06.2021].
- [5] Stack Overflow – forum programistyczne, <https://stackoverflow.com>, [09.09.2021].
- [6] Metryka SLOC – opis, <https://pvs-studio.com/en/blog/terms/0086/>, [20.06.2021].