Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

Analysis Comparative of Performance Optimization Techniques for PHP Framework Testing: Laravel, CodeIgniter, Symfony

Fauzan Prasetyo Eka Putra[1], Achmad Zulfikri[2*], Abd Rohman[3], Royfal Alim[4]
[1,2,3,4,] Fakultas Teknik, Informatika Universitas Madura Jl. Raya Panglegur No.Km 3,5, Barat, Panglegur, Kec. Tlanakan, Kabupaten Pamekasan, Jawa Timur 69371
[1]prasetyo@unira.ac.id, [2]achamadzulfikri20@gmail.com, [3]rompieslancelots79@gmail.com, [4]alimroyfal@gmail.com

open access

***Corresponding Author**

Article History:
Submitted: 31-05-2025
Accepted: 09-06-2025
Published: 24-06-2025

Keywords:
Laravel; CodeIgniter; Symfony; Comparison; Performance;

ABSTRACT

Web application performance significantly impacts digital services in sectors like e-commerce, education, and healthcare, where poor response time (>100 ms) can reduce user satisfaction by up to 20% and cause revenue losses. Despite numerous studies on PHP frameworks, comprehensive comparisons integrating optimization techniques remain limited. This research empirically evaluates the performance of Laravel, CodeIgniter, and Symfony, focusing on response time, throughput, and error rate before and after applying optimization techniques. Load and stress testing was conducted using Apache JMeter, Xdebug, and New Relic on a student management system with a MySQL database (10,000 records), simulating realistic scenarios from 1 to 100 users, mimicking e-commerce or educational platforms. Optimization techniques included caching (e.g., route caching in Laravel, query caching in CodeIgniter), database optimization (indexing, eager loading), and code refactoring. Results revealed CodeIgniter achieved the lowest response time (16.49 ms, 41.4% reduction) and highest throughput, ideal for lightweight applications like news portals. Laravel showed the greatest improvement under high load (35.1% reduction, from 118.34 ms to 76.80 ms), suitable for e-commerce platforms. Symfony demonstrated stable performance (20.0%–25.0% reduction), fitting enterprise applications with complex APIs. This study provides practical guidance for developers in selecting PHP frameworks based on project requirements and offers optimization recommendations to enhance web application efficiency and scalability.

INTRODUCTION

Web application development is a key pillar of digital services in sectors such as e-commerce, education, and healthcare, where web application performance determines user experience and operational success(Sadu et al., 2023). Metrics such as response time, throughput, and error rates are critical indicators, as response times above 100 ms can decrease user satisfaction by up to 20%, even causing revenue loss in e-commerce applications(Saputra et al., 2025). Sudden traffic spikes often worsen throughput and trigger system failures. Therefore, the development of fast, reliable, and scalable web applications is a top priority for developers(Matea, 2022; F. P. E. Putra et al., 2024).

PHP frameworks, as a popular web development technology, offer solutions to these performance challenges. Laravel, CodeIgniter, and Symfony, three popular frameworks, have different characteristics. Laravel, with its Eloquent ORM and elegant syntax, makes it easier to develop complex applications, but has a larger processing overhead. CodeIgniter, with its minimalist approach and small memory footprint, excels in response time and throughput for simple applications such as news portals. Symfony, with its modular architecture and dependency injection, is ideal for enterprise applications that require scalability, such as API-based systems(Nasution et al., 2023; Perdana et al., 2024; Siahaan & Wijaya, 2024).

Previous research has examined the performance of PHP frameworks, but with limitations. Ahmed et al. found that CodeIgniter is faster for lightweight applications, but less flexible for complex projects than Laravel(Ahmed, Gurama, et al., 2024). Gajewski and Dzieńkowski showed that Symfony is more flexible for enterprise applications, but Laravel is faster to implement on simple CRUD applications(Gajewski & Dzieńkowski, 2024). Putra et al. highlighted the advantages of CodeIgniter in memory usage and Laravel in response time, but did not test Symfony(F. P. E. Putra et al., 2025). Niarman and Candri conducted load and stress testing on PHP frameworks, showing CodeIgniter's higher throughput at low load, but did not integrate optimization techniques such as caching(Niarman & Candri, 2023). Sismadi et al. emphasized the importance of frameworks for online applications, but did not comprehensively compare performance(Sismadi et al., 2022). The lack of studies testing Laravel, CodeIgniter, and Symfony with optimization techniques in realistic load scenarios makes it difficult for developers to choose the optimal framework.

This research aims to address this gap by empirically comparing the performance of Laravel, CodeIgniter, and Symfony, focusing on response time, throughput, and error rate before and after optimization(KHADRAOUA et al.,

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

2022). The optimization techniques tested include caching, such as route caching in Laravel, query caching in CodeIgniter, database optimization (indexing, eager loading), and code refactoring for efficiency. This approach has proven to be effective, with up to 30% performance improvement through database optimization and framework security features. Testing was conducted on a MySQL-based student management application (10,000 records) using Apache JMeter to simulate loads from 1 to 100 users, with code profiling via Xdebug and system monitoring via New Relic. This research is expected to provide practical guidance for developers in choosing a PHP framework according to project needs, while emphasizing the importance of performance optimization to meet the demands of modern users(Brink et al., 2022; Vargas, 2023; Zulfikri et al., 2023).

LITERATURE REVIEW

This section outlines the theory and previous research that supports the performance analysis of PHP frameworks, specifically Laravel, CodeIgniter, and Symfony, as well as relevant performance optimization techniques(Melyanto et al., 2023). This literature review provides a theoretical foundation for the research methods, by referring to current scientific references.

There have been many studies on the performance of PHP frameworks. According to Ahmed et al., Laravel excels in development productivity thanks to the Eloquent ORM and middleware, but has a larger processing overhead compared to lightweight frameworks like CodeIgniter(Ahmed, Bello, et al., 2024). CodeIgniter, as explained by Firmansyah and Sudarmilah, is designed for applications with a low memory footprint and high performance, making it ideal for small to medium-sized applications such as news portals or local management systems(Firmansyah & Sudarmilah, 2025). Meanwhile, Ewa Wieleba and Bartłomiej Wieleba highlight that Symfony, with its modularity and dependency injection support, is suitable for API-based enterprise applications or large-scale systems, although its initial configuration is more complex(Wieleba & Wieleba, 2023).

Performance optimization techniques are a major focus in web application development. Kansha showed that caching, such as route caching in Laravel and query caching in CodeIgniter, can reduce response time up to 35% in web applications(Kansha, 2023). Research by Węgrzecki K and Dzieńkowski M revealed that database optimizations, such as indexing and eager loading, improve PHP application performance by up to 30%, especially in frameworks with intensive database operations(Węgrzecki & Dzieńkowski, 2022). In addition, Pawelec K and Kopniak P emphasized that refactoring code to reduce redundancy can improve the efficiency of PHP frameworks, especially in high load testing scenarios(Pawelec & Kopniak, 2022).

Evaluation metrics such as response time, throughput, and error rate are standard in performance analysis. Sismadi et al. define response time as the main indicator of application speed, while throughput reflects the system's ability to handle concurrent requests, especially in e-commerce or educational applications(Sismadi et al., 2022). Niarman and Candri added that error rate is critical in evaluating framework stability under extreme loads. This research adopts such metrics to compare Laravel, CodeIgniter, and Symfony, taking into account the finding that lightweight frameworks such as CodeIgniter tend to excel in throughput at low loads(Niarman & Candri, 2023).

Although many studies address the performance of PHP frameworks, there is a gap in comparative analysis that integrates optimization techniques such as caching, query optimization, and code refactoring together(Barzilai & Gafni, 2023). Previous studies, such as those by Ahmed et al. and Gajewski and Dzieńkowski, focus on inter-framework comparisons without thoroughly evaluating the impact of optimization(Ahmed, Bello, et al., 2024; Gajewski & Dzieńkowski, 2024). This research aims to fill that gap by evaluating the performance of Laravel, CodeIgniter, and Symfony before and after optimization, in realistic load testing scenarios using tools such as Apache JMeter and Xdebug(Suchanowski & Plechawska-Wójcik, 2023).

METHOD

In this methodology section, we describe the systematic steps we used to analyze performance optimization in PHP frameworks including Laravel, CodeIgniter and Symfony. The goal is to ensure that the tests are consistent, measurable, and replicable, so that the results are reliable and relevant for PHP framework performance analysis. We designed this methodology to evaluate the performance of Laravel, CodeIgniter, and Symfony in realistic test scenarios, considering factors such as user load, optimization, and system stability(Fauziah et al., 2022; Tadas & Lokulwar, 2024).

Framework Selection Criteria

We chose Laravel, CodeIgniter, and Symfony based on their popularity, documentation quality, performance features, and scalability. These three frameworks represent different approaches, namely Laravel for complex applications, CodeIgniter for lightweight applications, and Symfony for enterprise applications. The performance metrics used include average response time, throughput, and error rate, as per performance evaluation standards(Fitria & Chotijah, 2024; Suchanowski & ..., 2023).

Testing Setup

We conducted the tests on a server with hardware and software specifications designed to reflect a standard

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

production environment. Details of the test setup are presented in Table 1.

Table 1. Test Setup

Category	Specification	Testing Tools
Hardware	Intel Xeon E5-2620 v4 (8 core, 2.1 GHz), RAM 32 GB, SSD 1 TB	-
Software	Ubuntu 22.04 LTS, PHP 8.1, Nginx 1.18, MySQL 8.0	-
Testing Tools	-	Apache Jmeter 5.5 (load/stress testing), Xdebug 3.1 (profiling), New Relic APM (real-time monitoring), Blackfire (code performance analysis)

Testing Procedure

The testing flow is depicted in Figure 1 below.

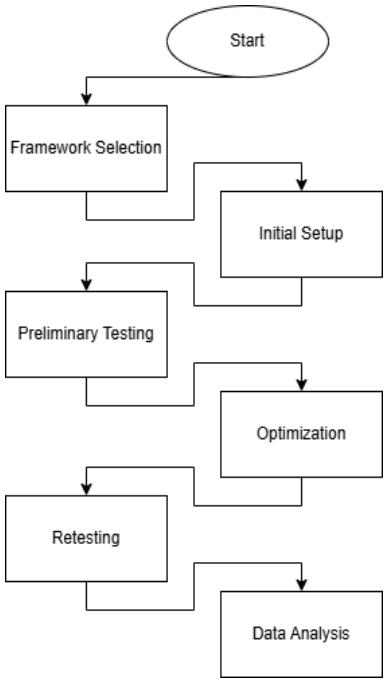

Figure 1. Framework Testing Flow

We created a test procedure to evaluate the framework's performance under normal and extreme conditions, and measure the impact of optimization techniques. The testing steps are as follows:

1. Application Development
   We built a simple web application of student management system for each framework with CRUD (create, read, update, delete) functions. This application uses MySQL database with 10,000 records to simulate realistic data.
2. Initial Configuration
   In this initial configuration, we set up the application without any optimizations for initial testing, and made sure the server and database configurations were consistent across all frameworks(Ardila, 2022; Makarim et al., 2024; Tsamiroh, n.d.).
3. Initial load testing
   In this test, we conducted load testing with three scenarios, first, 1 user/1 iteration (low load), then second, 10 users/10 iterations (medium load), and the third 10 users/100 iterations (high load). in this initial test, we used Apache JMeter software which is used to simulate user requests.
4. Stress testing
   In this test, we applied a heavy or extreme load of 100 users/1000 iterations to test the performance limits of the framework, recording the maximum response time and error rate.
5. Optimization
   At this stage, we apply various optimization techniques, these include:
   - Caching: Route caching (Laravel), view caching (Laravel, Symfony), query caching (CodeIgniter, Symfony).
   - Database Optimization: Indexing on database tables, eager loading (Laravel), and query optimization

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

(Symfony, CodeIgniter).
- Code Refactoring: Reducing code redundancy and optimizing application logic.

6. Retesting
   We repeated the load and stress tests after optimization to measure the change in performance.
7. Data Analysis
   We collected data using JMeter for response time and throughput, Xdebug for profiling, and New Relic for stability. Each test was run three times, and the average was calculated for accuracy(Amin, 2025; Hasirun & Riyanto, 2024; Pusvita & Mahmudah, 2024; Putri et al., 2024).

RESULT

This section of the results and discussion will present the results of the performance testing as well as an in-depth analysis of the implications of those results. The testing focused on four key metrics, namely average response time, throughput, error rate, and optimization effectiveness. The goal was to compare the performance of Laravel, CodeIgniter, and Symfony, identify the superior framework for each metric, and provide practical insights for developers. We separate the results by test metric for clarity, followed by a discussion of factors that affect performance and recommendations for framework usage(A. Putra, 2023).

Average Response Time

The average response time was measured for three load testing scenarios, as shown in Table 2.

Table 2. Average Response Time (ms) Scenario

Scenario	Laravel	CodeIgniter	Symfony
Test 1:1	45.12	16.49	38.80
Test 10:10	463.50	123.60	206.00
Test 2:1000	88.58	13.58	51.50

CodeIgniter showed the lowest response time in all scenarios (16.49 ms in Test 1:1, 13.58 ms in Test 2:1000), making it the top framework for speed. Symfony had an intermediate performance (38.80 ms in Test 1:1), while Laravel was the slowest (463.50 ms in Test 10:10). CodeIgniter's lightweight design minimizes overhead, while Laravels advanced features like Eloquent ORM improve response time.

Throughtput

Throughput in this test is measured in requests per second (rps), as shown in Table 3.

Table 3. Throughtput (rps)

Scenario	Laravel	CodeIgniter	Symfony
Test 1:1	20.58	54.65	29.10
Test 10:10	22.35	78.83	41.20
Test 2:1000	8.47	10.69	15.45

CodeIgniter again excelled with the highest throughput (78.83 rps in Test 10:10), followed by Symfony (41.20 rps) and Laravel (22.35 rps). CodeIgniter's efficiency in handling concurrency makes it ideal for high-traffic applications. Symfony offers balance, while Laravel is more suitable for applications with complex logic.

Error Rate

In the tests conducted, all three frameworks showed an error rate of 0.000% in all scenarios, indicating that all three frameworks have high reliability. All three frameworks are very stable, but Laravel and Symfony have more advanced error handling features for example, middleware in Laravel, exception handling in Symfony, making them superior for mission critical applications.

Optimization Effectiveness

This section analyzes the impact of optimization techniques such as caching, database optimization, and code refactoring on the average response time for Laravel, CodeIgniter, and Symfony. The test results are presented in Figure 1, which compares the response time in two scenarios, testing with 1 user/1 iteration (low load) and 10 users/10 iterations (medium load). The graph uses colored bars for each framework, blue for Laravel, red for CodeIgniter, and green for Symfony. Bars with line color pattern show the response time before optimization, while bars with full color show after optimization. The number above each bar allows the reader to see the response time value directly, while the percentage reduction is explained to highlight the effectiveness of the optimization. This graph helps developers understand which frameworks are the most responsive and how optimizations improve performance in real contexts, such as e-commerce applications or news portals(NUR, 2024; Radomski, 2022; Triatmaja, 2024; Yagual et al., 2023).

Brilliance
Research of Artificial Intelligence

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

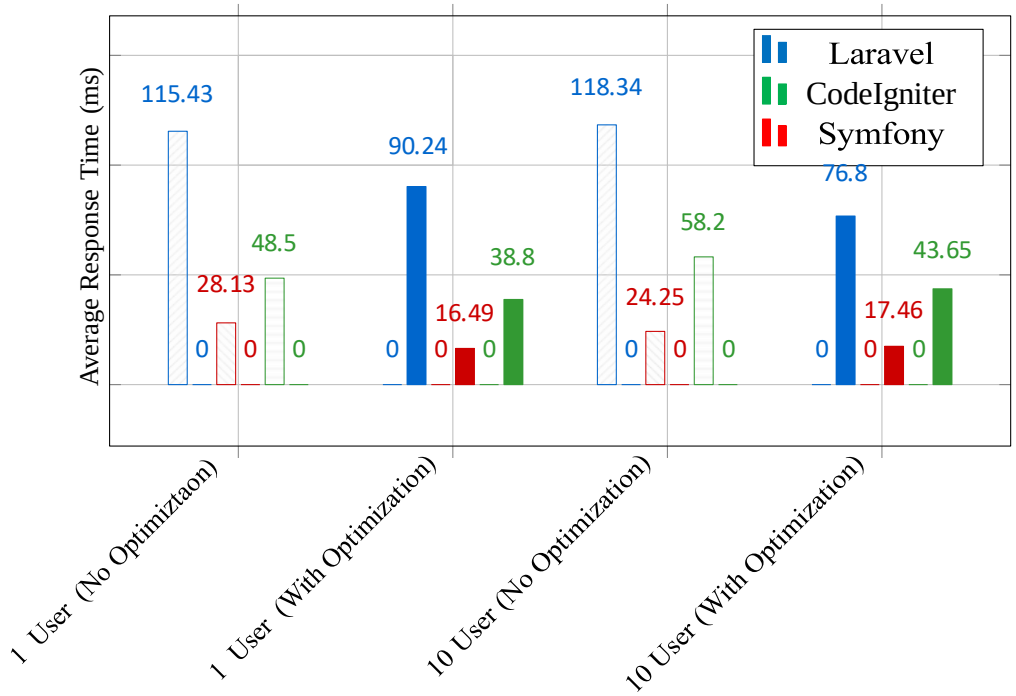

Figure 2. Comparison of response time before and after optimization

Test Scenario

The graph shows that the optimizations significantly reduced the response time for all frameworks. CodeIgniter recorded the largest decrease in the 1-user scenario (from 28.13 ms to 16.49 ms, -41.4%), thanks to its query caching and lightweight design with minimal overhead. This makes it ideal for lightweight applications such as news portals or local systems. Laravel showed the most improvement in the 10-user scenario (from 118.34 ms to 76.80 ms, -35.1%), supported by eager loading and route caching, making it a strong choice for high-load e-commerce applications. Symfony has a stable reduction (20.0% at 1 user, from 48.50 ms to 38.80 ms, 25.0% at 10 users, from 58.20 ms to 43.65 ms), with HTTP caching and database optimization providing consistent performance for enterprise applications. This graph underscores the importance of optimization to improve web application speed, with each framework excelling in different scenarios according to project needs.

DISCUSSION

These results confirm that CodeIgniter is the best choice for lightweight applications with high speed requirements, such as news portals or local apps. Laravel and Symfony are more suitable for complex applications such as e-commerce platforms or enterprise management systems, thanks to features such as ORM and modularity. Performance optimization is very important, especially for Laravel, which shows significant improvement after optimization.

This finding is in line with previous research that noted the superiority of CodeIgniter for performance and Laravel/Symfony for scalability. In the context of e-commerce, Laravel and Symfony excel due to API support and cloud scalability, while CodeIgniter is suitable for applications with simple logic. Limitations of this research include the focus on MySQL and specific test scenarios. Future research could explore metrics such as memory usage or performance with other databases.

CONCLUSION

This research shows that optimization techniques such as caching, database optimization, and code refactoring significantly improve the performance of PHP frameworks. CodeIgniter stands out with the lowest response time (16.49 ms at 1 user, 41.4% reduction) and highest throughput, making it an optimal choice for lightweight applications such as news portals, small enterprise applications, or local management systems. Laravel records the greatest performance improvement at high loads (35.1% reduction at 10 users, from 118.34 ms to 76.80 ms), supported by eager loading and route caching, making it suitable for complex applications such as e-commerce platforms or cloud-based services. Symfony offers stable performance with a response time reduction of 20.0%-25.0% (from 48.50 ms to 38.80 ms at 1 user, and 58.20 ms to 43.65 ms at 10 users), ideal for enterprise applications such as API systems or large-scale data management.

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

This finding suggests that framework selection should be tailored to the needs of the project, such as the number of users, application complexity, and server resource availability. Developers are advised to apply optimizations consistently, especially on heavy frameworks such as Laravel, to overcome overhead and maximize speed. For future research, performance evaluation with alternative databases (e.g., PostgreSQL or MongoDB) or additional metrics such as memory usage can provide more insight. This research is expected to help developers make decisions in building fast and reliable web applications.

REFERENCES

Ahmed, M. K., Bello, A. H., Jauro, S. S., & Dawaki, M. (2024). A comparative analysis of performance optimization techniques for benchmarking PHP frameworks: Laravel and Codeigniter. *Dutse Journal of Pure and …*. https://www.ajol.info/index.php/dujopas/article/view/281638

Ahmed, M. K., Gurama, A. U., & ... (2024). Development of a New Minimalistic PHP Micro Framework: NOAPHP. *BIMA JOURNAL OF …*. https://journals.gjbeacademia.com/index.php/bimajst/article/view/810

Amin, M. (2025). Perbandingan Kinerja Nginx dan Caddy sebagai Web Server untuk Aplikasi PHP. *Insect (Informatics and Security): Jurnal Teknik …*. http://www.ejournal.um-sorong.ac.id/index.php/insect/article/view/4223

Ardila, K. (2022). *LKP: Rancang Bangun Aplikasi Job Order Pengiriman Cargo Berbasis Website pada PT Sunan Inti Perkasa*. repository.dinamika.ac.id. https://repository.dinamika.ac.id/id/eprint/6097/

Barzilai, O., & Gafni, R. (2023). Using web frameworks in server side programming courses. *Journal of Computer Information Systems*. https://doi.org/10.1080/08874417.2022.2111378

Brink, W. Van den, Gerhold, M., & ... (2022). Deriving modernity signatures for PHP systems with static analysis. *2022 IEEE 22nd …*. https://ieeexplore.ieee.org/abstract/document/10006847/

Fauziah, A., Noer, S., Junaedi, J., & ... (2022). SYSTEM PENJUALAN SAYUR MENGGUNAKAN FRAMEWORK LARAVEL: SYSTEM PENJUALAN SAYUR MENGGUNAKAN FRAMEWORK LARAVEL. *Prosiding …*. http://prosiding.sentimeter.nusaputra.ac.id/index.php/prosiding/article/view/32

Firmansyah, S., & Sudarmilah, S. T. E. (2025). *Analisis Performa Framework Codeigniter Dan Laravel Dalam Penggunaan Bahasa Pemrograman Php*. eprints.ums.ac.id. https://eprints.ums.ac.id/id/eprint/132852

Fitria, L., & Chotijah, U. (2024). Implementasi Framework Laravel pada Sistem Manajemen Penomoran dan Arsip Surat Bawaslu Kota Surabaya. *SABER: Jurnal Teknik Informatika …*. https://jurnal.stikes-ibnusina.ac.id/index.php/SABER/article/view/1676

Gajewski, P., & Dzieńkowski, M. (2024). Comparison of Laravel and Symfony-the most popular PHP frame-works, based on a simple CRUD application. *Journal of Computer Sciences Institute*. https://ph.pollub.pl/index.php/jcsi/article/view/6146

Hasirun, H., & Riyanto, A. B. (2024). Implementasi Framework Laravel dan Enkripsi IONCUBE Encode untuk Meningkatkan Keamanan pada Perpustakaan Digital. *Jurnal Borneo Informatika Dan Teknik …*. http://jurnal.borneo.ac.id/index.php/jbit/article/view/6096

Kansha, W. M. (2023). Analisis Perbandingan Framework Codeigniter Dan Laravel Dalam Pengembangan Web Application. *Jurnal Teknik Informatika*. https://ejournal.antarbangsa.ac.id/jti/article/view/511

KHADRAOUA, N., Yagoub, R., & ... (2022). *Moteur de recherche pour les revues scientifiques*. dspace.univ-temouchent.edu.dz. https://dspace.univ-temouchent.edu.dz/handle/123456789/2210

Makarim, I. F., Edfira, A. F., Ramadhani, A. A., & ... (2024). Penerapan Arsitektur MVC pada Website Pengumpul Tugas Menggunakan PHP. *Prosiding Seminar …*. https://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/465

Matea, B. (2022). *Web aplikacija za upravljanje plesnim studiom*. repozitorij.foi.unizg.hr. https://repozitorij.foi.unizg.hr/islandora/object/foi:7280

Melyanto, D. Y., Khusnuliawati, H., & Haris, F. H. S. Al. (2023). *Program Aplikasi Sistem Informasi Pemesanan Jasa Studio Foto pada Latar Belakang Studio Karanganyar dengan Framework Codeigniter*. repository.usahidsolo.ac.id. http://repository.usahidsolo.ac.id/2705/5/Dyota%20Yungyun%20M_Daftar%20Pustaka_2016063002.pdf

Nasution, N., Nasution, F. B., & ... (2023). PKM pelatihan pembuatan web berbasis framework CodeIgniter untuk siswa SMK. *J-COSCIS: Journal of …*. https://journal.unilak.ac.id/index.php/jcoscis/article/view/11001

Niarman, A., & Candri, A. K. (2023). Comparative analysis of PHP frameworks for development of academic information system using load and stress testing. *International Journal Software …*. https://www.journal.lembagakita.org/ijsecs/article/view/1850

NUR, I. (2024). *RANCANG BANGUN SISTEM INFORMASI PENERIMAAN PEGAWAI BERBASIS WEB DENGAN METODE RAPID APPLICATION DEVELOPMENT (RAD): STUDI …*. repository.nurulfikri.ac.id. https://repository.nurulfikri.ac.id/id/eprint/531/

Pawelec, K., & Kopniak, P. (2022). Comparison of frameworks for developing web applications in PHP. *Journal of Computer Sciences Institute*. https://ph.pollub.pl/index.php/jcsi/article/view/2784

E-ISSN : 2807-9035
Volume 5, **Number** 1, May 2025
https://doi.org/10.47709/brilliance.v5i1.5989

Perdana, A., Farhana, N. A., Harliana, P., & ... (2024). Web-Based Application Development using PHP-Native Framework on Agent of Change Integrity Zone Information System. *Sinkron: Jurnal Dan …*. https://jurnal.polgan.ac.id/index.php/sinkron/article/view/14118

Pusvita, E. A., & Mahmudah, L. (2024). Pendataan Sarpras (Sarana Prasarana) Berbasis PHPrad SMK Negeri 2 Teknologi dan Rekayasa Nabire Provinsi Papua Tengah. *Jurnal Teknologi Dan Informatika*. https://pesatnabire.id/index.php/jti/article/view/34

Putra, A. (2023). *APLIKASI LAYANAN PENGAJUAN PERSURATAN MAHASISWA POLITEKNIK KESEHATAN BHAKTI SETYA INDONESIA MENGGUNAKAN FRAMEWORK LARAVEL*. eprints.utdi.ac.id. https://eprints.utdi.ac.id/10019/

Putra, F. P. E., Sugi, S. A., & Mufidah, K. (2025). Comparative Literature Study of CodeIgniter and Laravel Framework Performance in Website Creation. *Jurnal Informasi Dan Teknologi*. https://www.jidt.org/jidt/article/view/617

Putra, F. P. E., Ubaidi, U., Zulfikri, A., Arifin, G., & Ilhamsyah, R. M. (2024). Analysis of Phishing Attack Trends, Impacts and Prevention Methods: Literature Study. *Brilliance: Research of Artificial Intelligence*, *4*(1), 413–421. https://doi.org/10.47709/brilliance.v4i1.4357

Putri, F. A., Nasution, F. P., Doni, R., & ... (2024). Pelatihan Pembuatan Website Menggunakan CodeIgniter Pada Siswa Madrasah Aliyah Swasta Yaspi Labuhan Batu. *Society: Jurnal …*. https://www.edumediasolution.com/society/article/view/425

Radomski, J. (2022). Comparison of web application performance on the example of Laravel and Vaadin frameworks. *Journal of Computer Sciences Institute*. https://ph.pollub.pl/index.php/jcsi/article/view/2790

Sadu, G. F. Z., Paturusi, S. D. E., & ... (2023). Rancang Bangun Aplikasi Layanan Ujian Berbasis Komputer di Portal Akademik Perguruan Tinggi: Design and Development of Computer-Based Examination …. *Jurnal Teknik …*. https://ejournal.unsrat.ac.id/index.php/informatika/article/view/50626

Saputra, B. A., Supriyanti, E., & Listyorini, T. (2025). IMPLEMENTATION OF AN E-COMMERCE BASED ONLINE SALES SYSTEM AT DAPOER LN MSME KUDUS. *Tekmapro*. https://tekmapro.upnjatim.ac.id/index.php/tekmapro/article/view/872

Siahaan, M., & Wijaya, R. W. (2024). Performance Comparison Between Laravel and ExpressJs Framework Using Apache JMeter. *Journal Of Informatics And …*. https://ojs.uma.ac.id/index.php/jite/article/view/10571

Sismadi, W., Martono, B. A., & Widyastuti, T. (2022). Comparative Analysis of Codeigniter, Laravel and Ktupad Frameworks: Case Study Online Exam Applications. *Indonesian Journal of …*. https://iojs.unida.ac.id/index.php/IJAR/article/view/236

Suchanowski, J., & ... (2023). Analiza wydajnościowa aplikacji internetowych utworzonych w szkieletach Spring i Laravel. *Journal of Computer …*. https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-1f03abe1-4e8f-4258-a116-a2bdf7cde7d0

Suchanowski, J., & Plechawska-Wójcik, M. (2023). Performance analysis of web applications created in the Spring and Laravel frameworks. *Journal of Computer Sciences …*. https://ph.pollub.pl/index.php/jcsi/article/view/3770

Tadas, P., & Lokulwar, P. (2024). Design and Development of Advanced CRM and Task Management System using PHP. *Grenze International Journal of …*. https://search.ebscohost.com/login.aspx?direct=true&profile=ehost&scope=site&authtype=crawler&jrnl=23955287&AN=175658246&h=Q2av7XDmtxrs4khQ309no7fN07vKuo0DQUjAeZeYM7ST8U1lSIrErSLpC5ERPCC9YOnaCM27oWAP%2FztcYJT1Yw%3D%3D&crl=c

Triatmaja, B. (2024). *RANCANG BANGUN SISTEM INFORMASI KEUANGAN MENGGUNAKAN FRAMEWORK LARAVEL PADA MASJID NURUL FIKRI DEPOK*. repository.nurulfikri.ac.id. https://repository.nurulfikri.ac.id/id/eprint/582/

Tsamiroh, M. (n.d.). Rancang bangun aplikasi web tanaman obat keluarga (toga) menggunakan metode agile pada laravel. *Repository.Uinjkt.Ac.Id*. https://repository.uinjkt.ac.id/dspace/handle/123456789/83418

Vargas, B. M. M. (2023). *Sistema web para la publicación y gestión de cursos para un centro de capacitación con el Framework Codeigniter.* 201.159.223.6. http://201.159.223.6/handle/123456789/372

Węgrzecki, K. S., & Dzieńkowski, M. (2022). Performance analysis of LARAVEL and YII2 frameworks based on the MVC architectural pattern and PHP language. *Journal of Computer Sciences Institute*. https://ph.pollub.pl/index.php/jcsi/article/view/3002

Wieleba, E., & Wieleba, B. (2023). Performance analysis of working with databases with Spring and Symfony. *Journal of Computer Sciences Institute*. https://ph.pollub.pl/index.php/jcsi/article/view/3086

Yagual, C., Andrés, C., & ... (2023). Design and comparison of data through a dynamic fluid simulation in various geometries of an intake nozzle. *Revista Científica y …*. http://scielo.senescyt.gob.ec/scielo.php?pid=S1390-76972023000100070&script=sci_abstract&tlng=en

Zulfikri, A., Putra, F. P. E., Huda, M. A., Hasbullah, H., Mahendra, M., & Surur, M. (2023). Analisis Keamanan Jaringan Dari Serangan Malware Menggunakan Filtering Firewall Dengan Port Blocking. *Digital Transformation Technology*, *3*(2), 857–863. https://doi.org/10.47709/digitech.v3i2.3379