УДК 004.056.52

Огневой Ярослав Владимирович, магистрант 2 курса факультета информационных технологий, Московский политехнический университет, г. Москва, Российская Федерация

Шеховцов Артем Вадимович, магистрант 2 курса факультета информационных технологий, Московский политехнический университет, г. Москва, Российская Федерация

Рыжиков Михаил Васильевич, магистрант 2 курса факультета информационных технологий, Московский политехнический университет, г. Москва, Российская Федерация

Левицкий Вадим Ильич, магистрант 2 курса факультета информационных технологий, Московский политехнический университет, г. Москва, Российская Федерация

СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ PHP-ФРЕЙМВОРКОВ

Аннотация

В условиях стремительного роста количества веб-приложений возрастает значимость обеспечения их безопасности. Современные PHP-фреймворки предоставляют разработчикам встроенные механизмы защиты от распространённых угроз, таких как межсайтовый скриптинг (XSS), межсайтовая подделка запросов (CSRF) и SQL-инъекции. В статье проведено сравнительное исследование безопасности популярных PHP-фреймворков, включая Symfony,

Yii2 и Laravel. Рассмотрены особенности реализации защитных механизмов, проанализированы подходы к обработке пользовательского ввода, управлению доступом и защите данных. Выявлены различия в уровне безопасности и удобстве применения встроенных инструментов. Полученные результаты позволяют определить сильные и слабые стороны фреймворков и могут быть использованы при выборе технологической платформы для разработки защищённых веб-приложений.

Annotation

With the rapid growth of web applications, ensuring their security becomes increasingly important. Modern PHP frameworks provide developers with built-in protection mechanisms against common threats such as cross-site scripting (XSS), cross-site request forgery (CSRF), and SQL injection. This paper presents a comparative study of the security of popular PHP frameworks, including Symfony, Yii2, and Laravel. The study examines the implementation of security mechanisms, approaches to input validation, access control, and data protection. Differences in security levels and usability of built-in tools are identified. The results obtained make it possible to determine the strengths and weaknesses of the frameworks and can be used when selecting a technological platform for developing secure web applications.

Ключевые слова: PHP-фреймворки, веб-безопасность, Symfony, Laravel, CSRF, XSS, SQL-инъекция, безопасная разработка.

Keywords: PHP frameworks, web security, Symfony, Laravel, CSRF, XSS, SQL injection, secure development.

1. Введение

Язык программирования PHP считается одним из популярных инструментов для разработки веб-приложений благодаря своей гибкости, поддержке кроссплатформенности и работы с базами данных. PHP позволяет создавать динамические и интерактивные веб-страницы, но при использовании чистого PHP бизнес-логика часто смешивается с запросами к базе данных и представлением, что затрудняет сопровождение и повышает риск ошибок, в том числе связанных с безопасностью.

Для решения этих проблем были разработаны PHP-фреймворки [[1], [2]]. Они ускоряют процесс разработки, предоставляя базовую архитектуру, набор API, библиотек и расширений, что улучшает производительность [[3], [4]], снижает вероятность уязвимостей и облегчает написание безопасного кода.

Большинство PHP-фреймворков основаны на архитектурной модели Model-View-Controller (MVC) [[5], [6]], которая разделяет приложение на отдельные слои. Модель снижает сложность проектирования и облегчает внедрение механизмов безопасности.

В последние годы разработчики активно используют такие фреймворки, как Symfony, Yii2 и Laravel, каждый из которых предлагает свои встроенные инструменты защиты.

В работе проведено сравнительное исследование безопасности PHP-фреймворков, с целью выявления оптимального сочетания встроенных механизмов защиты, архитектурных особенностей и удобства сопровождения веб-приложений.

2. Оценка с использованием метода QSOS трёх PHP-фреймворков (Symfony, Yii2, Laravel)

2.1. Определение

QSOS (Qualification and Selection of Open Source Software) – это метод оценки программного обеспечения с открытым исходным кодом, разработанный компанией Atos Origin. Он предназначен для того, чтобы сравнивать, оценивать и выбирать программное обеспечение с учётом контекста использования [[7], [8]].

Метод QSOS основывается на итерационном процессе, состоящем из четырёх этапов (рис. 1):

1. Определение (Define) – определение набора критериев, необходимых для последующей оценки и сравнения.
2. Оценка (Evaluate) – оценка версии программного обеспечения, включая функциональные возможности и зрелость проекта. В данном случае акцент делается на механизмы безопасности, такие как защита от XSS/CSRF, встроенная аутентификация и обработка данных.
3. Квалификация (Qualify) – взвешивание оценки с учётом конкретного контекста применения.
4. Выбор (Select) – сравнение и окончательный выбор программного обеспечения на основе данных, полученных на предыдущих этапах.

Рисунок 1. Итерационный процесс метода QSOS

Метод QSOS реализуется итерационно: процесс может применяться с разной степенью детализации, что позволяет адаптировать уровень анализа и выбора под конкретные задачи. При необходимости можно повторять циклы прохождения этапов, уточняя оценки и корректируя взвешивание критериев.

2.2. Оценочная матрица для трёх фреймворков

В рамках данного исследования метод QSOS применяется для оценки и сравнения трёх распространённых PHP-фреймворков – Symfony [9], Yii2 [10] и Laravel [11] – по критериям безопасности и надёжности.

Для каждой характеристики используется следующая шкала оценивания:

- **0** – функция отсутствует или не поддерживается;
- **1** – функция частично реализована или требует дополнительной настройки;
- **2** – функция реализована полностью и доступна «из коробки».

На основе этих правил формируется оценочная таблица (таблица 1), включающая ключевые категории метода QSOS, адаптированные для анализа уровня защищённости PHP-фреймворков.

Таблица 1. Оценочная матрица безопасности PHP-фреймворков

Категория / Критерий	**Symfony**	**Yii2**	**Laravel**
Зрелость проекта			
Возраст проекта	2 – стабилен с 2005 г.	2 – стабилен с 2014 г.	2 – стабилен с 2011 г.
Известные уязвимости и их устранение	2 – активное реагирование, публикация CVE	2 – активное реагирование, выпуск патчей	2 – активное реагирование, публикация CVE
Сообщество и поддержка			
Документация по безопасности	2 – отдельный раздел Security Guidelines	2 – отдельный раздел Security Guidelines	2 – отдельный раздел Security Docs
Форумы / каналы поддержки	2 – GitHub, Slack, SymfonyCasts	2 – форум, GitHub	2 – GitHub, Laracasts
Архитектура и механизмы защиты			
Защита от XSS	2 – автоматическое экранирование Twig	1 – рекомендуется ручное экранирование (Html::encode), но возможно автоматическое	2 – встроенное экранирование Blade
Защита от CSRF	2 – включена по умолчанию	2 – включена по умолчанию	2 – включена по умолчанию
SQL-инъекции	2 – Doctrine ORM и фильтрация запросов	2 – ActiveRecord	2 – ORM Eloquent
Шифрование и хэширование	2 – bcrypt, Argon2, OpenSSL	2 – bcrypt, OpenSSL	2 – bcrypt, Argon2, AES

Аутентификация и авторизация	1 – Security Component, ACL	2 – RBAC из коробки	2 – встроенный Auth + Passport/Sanctum
Логирование и аудит	2 – Monolog встроен	1 – логирование событий (через расширения)	2 – Monolog встроен
Встроенный модуль сканирования уязвимостей	0 - Функция отсутствует	0 - Функция отсутствует	0 - Функция отсутствует
Управление обновлениями и уязвимостями			
Периодичность патчей	2 – регулярные LTS-релизы (каждые 2 года), минорные (каждые 6 мес), патчи (ежемесячно)	1 – обновления каждые 6–12 мес	2 – регулярные обновления LTS (каждые 2 года), частые патчи
Процесс отклика на уязвимости	2 – публичный security policy, GitHub advisories	2 – GitHub security advisory	2 – публичный security policy
Независимость и масштабируемость			
Совместимость с внешними средствами защиты	1 – интеграция через middleware	1 – интеграция через middleware	2 – поддержка WAF, IDS
Кроссплатформенность	2 – Windows / Linux / macOS	2 – Windows / Linux / macOS	2 – Windows / Linux / macOS
Учебные и методические ресурсы (Training & Awareness)			
Курсы по безопасной разработке	1 – SymfonyCasts, оф. обучение	1 – онлайн-курсы и оф. документация	2 – множество курсов и книг
Официальные публикации	2 – Symfony Security Best Practices	2 – Yii2 Security Guide	2 – Laravel Security Docs, OWASP

Таблица 2. Средние баллы по категориям

Категория	**Symfony**	**Yii2**	**Laravel**
Зрелость проекта	2.0	1.5	2.0
Сообщество и поддержка	2.0	2.0	2.0
Архитектура и механизмы защиты	1.6	1.5	2.0

Управление обновлениями и уязвимостями	2.0	1.5	2.0
Независимость и масштабируемость	1.5	1.5	2.0
Учебные и методические ресурсы	1.5	1.5	2.0

Таблица 3. Веса категорий

Категория	**Вес**
Зрелость проекта	0.15
Сообщество и поддержка	0.10
Архитектура и механизмы защиты	0.40
Управление обновлениями и уязвимостями	0.20
Независимость и масштабируемость	0.10
Учебные и методические ресурсы	0.05

Для наглядного сравнения по методу QSOS была построена диаграмма (рис. 1).

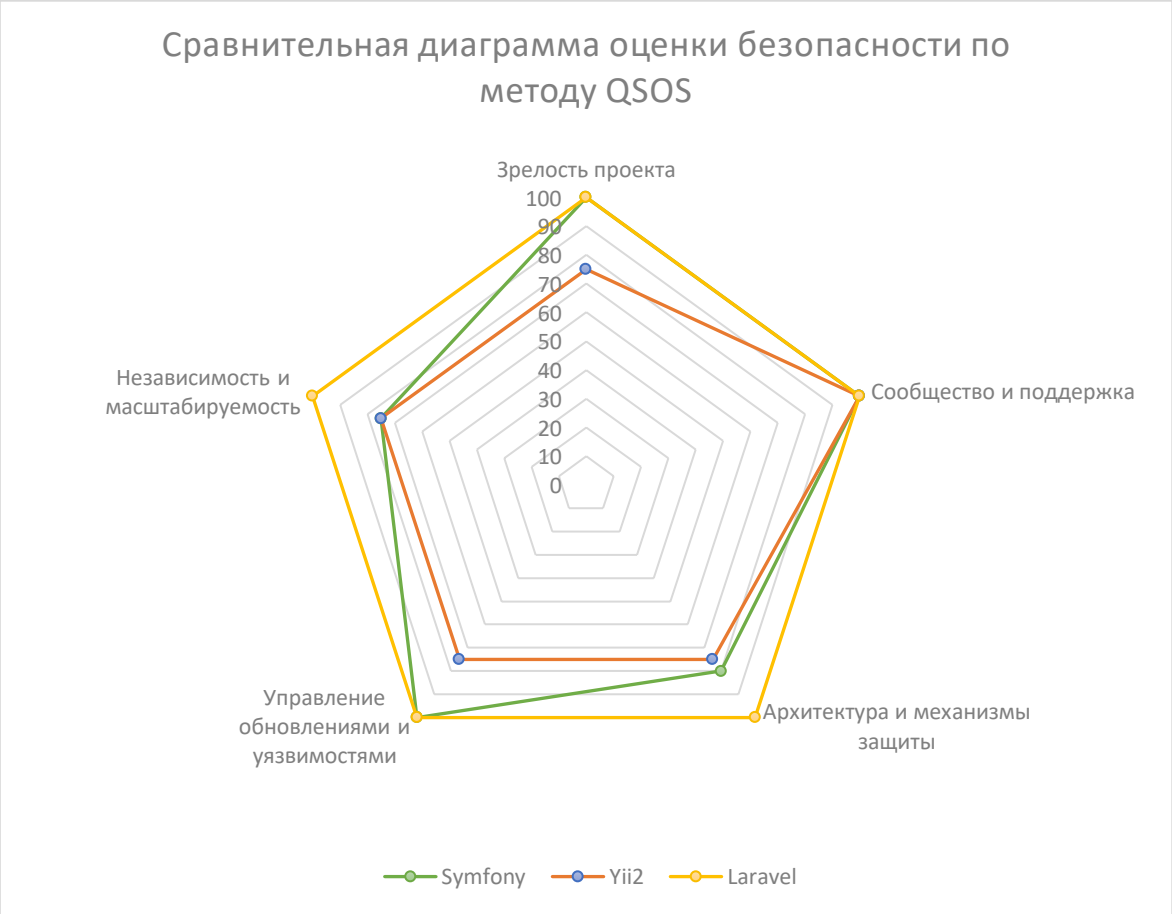

Рисунок 2. Сравнительная диаграмма оценки безопасности по методу QSOS

3. Оценка уровня безопасности

3.1. Анализ количества зарегистрированных уязвимостей

Для проведения сравнительного анализа уровня безопасности фреймворков Symfony, Yii2 и Laravel [12] были выбраны количественные критерии, отражающие практическую защищённость и зрелость подходов к управлению уязвимостями. В качестве показателей использовались:

- количество зарегистрированных уязвимостей в БДУ ФСТЭК России [13];
- количество уязвимостей, зафиксированных в международной базе NIST NVD [14];

- количество внешних зависимостей (Composer-пакетов), от которых зависит работа фреймворка.

Такие метрики позволяют оценить, насколько активно сообщество и разработчики поддерживают безопасность фреймворка, а также определить потенциальную поверхность атаки, связанную с использованием сторонних библиотек. Результаты анализа представлены на следующих диаграммах.

Рис. 3 демонстрирует количество уязвимостей, зарегистрированных в БДУ ФСТЭК по годам.

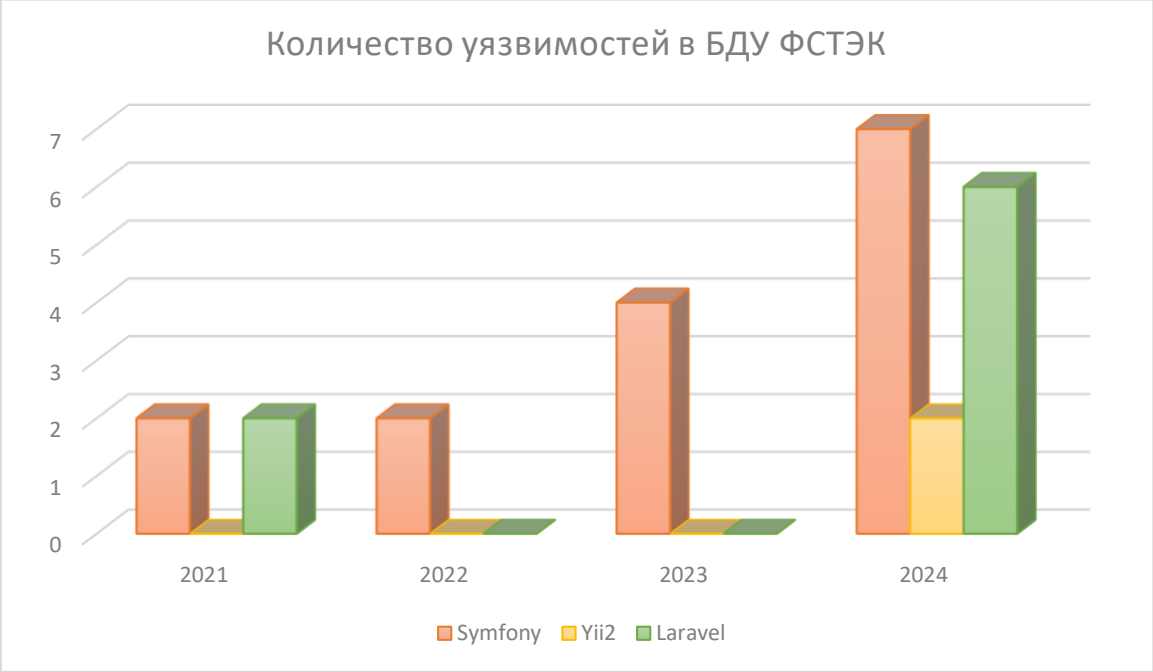

Рисунок 3. Количество уязвимостей PHP-фреймворков в БДУ ФСТЭК по годам

Рис. 4 демонстрирует количество уязвимостей, зарегистрированных в NIST NVD по годам.

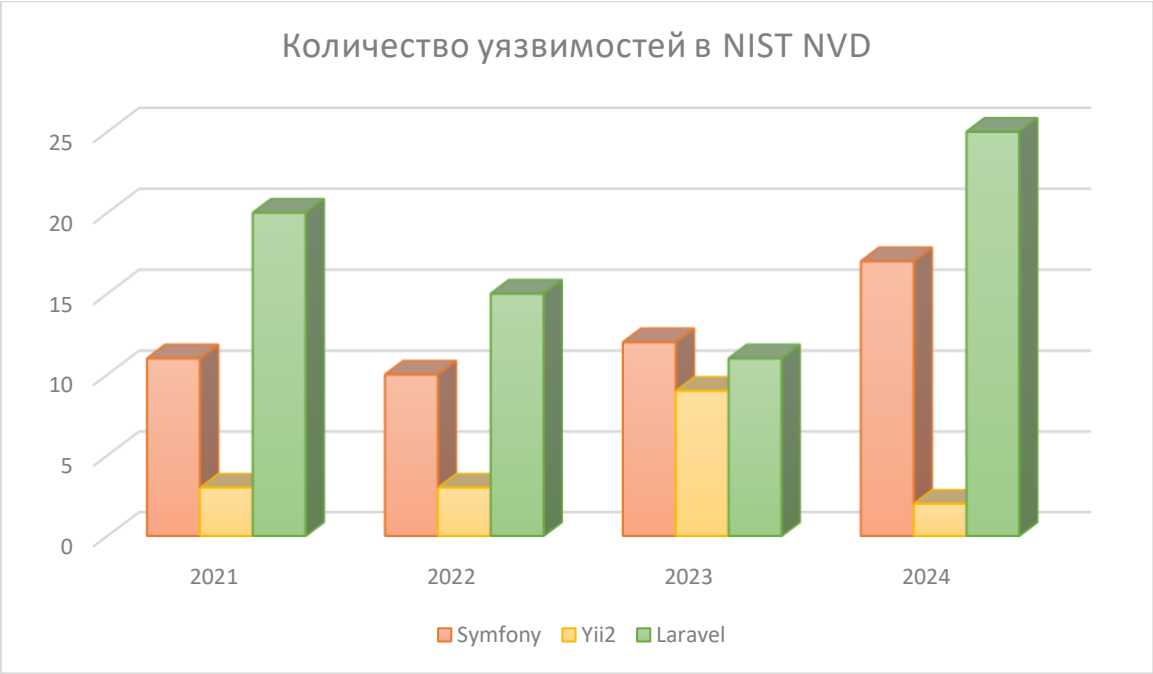

Рисунок 4. Количество уязвимостей PHP-фреймворков в NIST NVD по годам

Рис. 5 демонстрирует количество внешних зависимостей (composer-пакетов), используемых в каждом из рассматриваемых PHP-фреймворков.

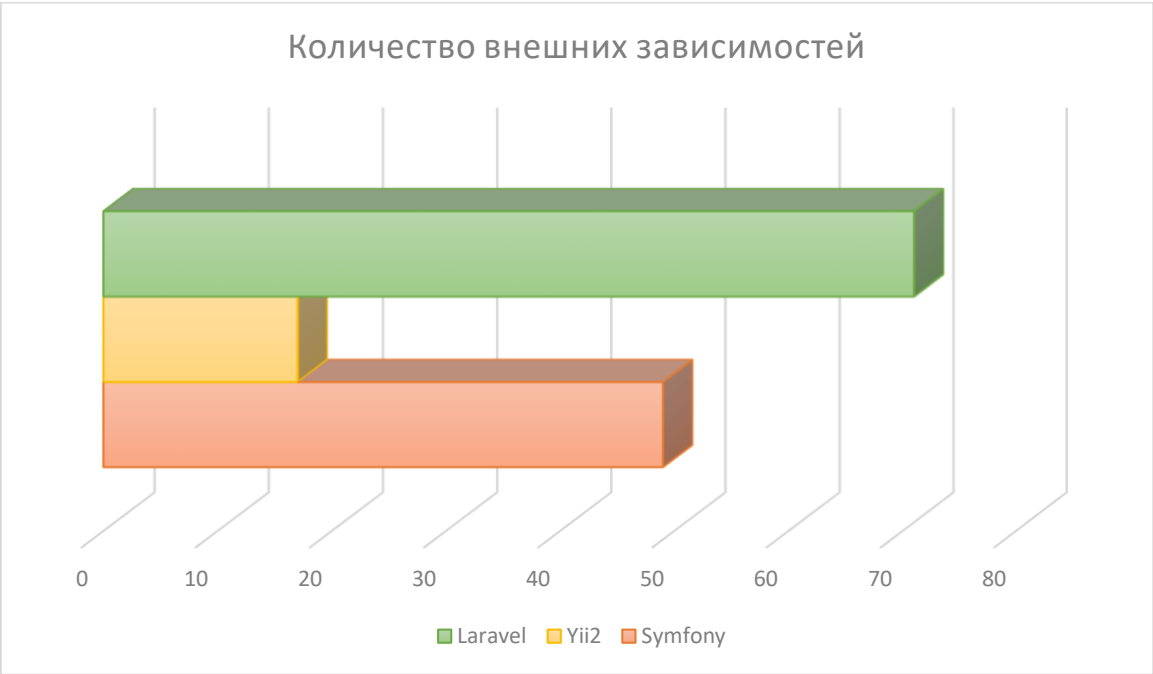

Рисунок 5. Количество внешних зависимостей PHP-фреймворков

3.2. Результаты

Анализ показателей безопасности PHP-фреймворков Symfony, Yii2 и Laravel позволил выявить ряд закономерностей, отражающих качество их поддержки и подход к обеспечению безопасности.

- На рис. 2 показано количество уязвимостей, зарегистрированных в базе ФСТЭК. Наименьшие значения продемонстрировал Yii2, тогда как у Symfony и Laravel отмечается рост уязвимостей в 2023–2024 годах.
- На рис. 3 представлено количество уязвимостей в базе NIST NVD. Наибольшее их число выявлено у Laravel, наименьшее – у Yii2, Symfony занимает промежуточное положение.
- На рис. 4 показано количество внешних зависимостей. Laravel имеет наиболее обширную экосистему (71 пакет), Symfony – умеренную (49), а Yii2 – наименее сложную структуру (17 зависимостей), что положительно влияет на безопасность.

4. Обсуждение

Согласно результатам оценки по методу QSOS и анализу показателей безопасности, фреймворки Symfony, Yii2 и Laravel соответствуют современным требованиям безопасной веб-разработки.

Symfony и Laravel обладают развитой архитектурой и широкими возможностями, что делает их востребованными для корпоративных и масштабных проектов, однако большое количество зависимостей увеличивает риск появления уязвимостей.

Yii2 показал наименьшее количество зарегистрированных угроз и зависимостей, что свидетельствует о его устойчивости с точки зрения безопасности делает его оптимальным выбором для средних и малых проектов.

5. Заключение

Проведённый сравнительный анализ показал, что Symfony, Yii2 и Laravel в целом соответствуют современным требованиям безопасной веб-разработки, но различаются по числу выявленных уязвимостей и объёму внешних зависимостей.

Yii2 продемонстрировал наименьшее количество зарегистрированных угроз и зависимостей, что делает его рациональным выбором для средних и малых проектов с ограниченными требованиями к масштабированию и акцентом на стабильность и простоту поддержки.

Laravel и Symfony, напротив, обладают более развитой архитектурой, активным сообществом и богатой экосистемой, что делает их востребованными для корпоративных и масштабируемых решений, требующих гибкости и расширяемости, но при этом повышает важность контроля внешних зависимостей и своевременного обновления.

Таким образом, выбор фреймворка должен основываться не только на функциональных возможностях, но и на оценке зрелости механизмов защиты, регулярности выпуска обновлений безопасности и прозрачности процессов управления уязвимостями.

Список литературы

1. Кит С. В., Обзор PHP-фреймворков для разработки современных информационных систем // Статья в журнале: Инновации. Наука. Образование. - 2021. С. 1–5.
2. Natalya Prokofyeva, Victoria Boltunova, Analysis and Practical Application of PHP Frameworks in Development of Web Information Systems // ICTE 2016, Riga Technical University, Latvia - 2016. С. 51–56.

3. Косарев Н. О., Сравнительное тестирование производительности PHP-основанных программных платформ (PHP-фреймворков) // Сборник статей XIV Международной научно-практической конференции: в 2 ч. Том Часть 1. - 2018. С. 37–40.
4. Кочнев А. А., Исследование оптимизации производительности в PHP: сравнительный анализ бэкенд-фреймворков // Международный научно-исследовательский журнал. - 2023. С. 1–9.
5. Rashidah F. Olanrewaju, Thouhedul Islam, N. Ali, An Empirical Study of the Evolution of PHP MVC Framework // Lecture Notes in Electrical Engineering (LNEE,volume 315) - 2014. С. 399–410.
6. Andri Sunardi, Suharjito, MVC Architecture: A Comparative Study Between Laravel Framework and Slim Framework in Freelancer Project Monitoring System Web Based // International Advanced Research Journal in Science, Engineering and Technology, - 2022. С. 134-141.
7. R. Semeteys, Method for qualification and selection of open-source software // no. May 2008. Talent First Network - 2008.
8. Rashidah F. Olanrewaju, Thouhedul Islam, N. Ali, An Empirical Study of the Evolution of PHP MVC Framework // Polytechnic of Coimbra – ISEC, Rua Pedro Nunes, Quinta da Nora, 3030-199 Coimbra, Portugal - 2019. С. 420–426.
9. Krzysztof Kuflewski, Mariusz Dzieńkowski, Symfony and Laravel – a comparative analysis of PHP programming frameworks // Journal of Computer Sciences Institute, 21 - 2021. С. 367–372.
10. Убийко В. А., Капустина О. А., Проблемы современной разработки на PHP с использованием фреймворка Yii2 // Материалы международной студенческой научно-практической конференции. - 2023. С. 255–260.

11. Кочнев А. А., Web Development с использованием PHP и фреймворка Laravel // Издательство: "NOWA PERSPEKTYWA" spolka z ograniczona odpowiedzialnoscia=ООО "Новая Перспектива". - 2023. С. 4–11.
12. Sangay Tenzin, PHP Framework for Web Application Development // International Advanced Research Journal in Science, Engineering and Technology, - 2022. С. 144-147.
13. БДУ – Уязвимости [Электронный ресурс]. - URL: bdu.fstec.ru/vul.
14. NVD – Vulnerabilities [Электронный ресурс]. - URL: https://nvd.nist.gov/vuln.