ЗАТВЕРДЖЕНО

Вченою радою факультету інформаційно-комп'ютерних технологій

28 серпня 2024 р., протокол №8

Голова Вченої ради

_____ Тетяна НІКІТЧУК

РОБОЧА ПРОГРАМА
вибіркової навчальної дисципліни фахової підготовки
«Розробка за допомогою PHP-фреймворків»

факультет інформаційно-комп'ютерних технологій

для здобувачів вищої освіти освітнього ступеня «бакалавр»

Схвалено на засіданні кафедри комп'ютерних наук
26 серпня 2024 р., протокол №8

Завідувач кафедри

_____ Марина ГРАФ

Розробник: старший викладач кафедри комп'ютерних наук
Олександр КУЗЬМЕНКО

Житомир
2024 – 2025 н.р.

Робоча програма вибіркової навчальної дисципліни «Розробка за допомогою PHP-фреймворків» затверджена Вченою радою факультету інформаційно-комп’ютерних технологій від 28 серпня 2024 р., протокол № 8.

1. Опис навчальної дисципліни

Найменування показників	Характеристика навчальної дисципліни
	денна форма навчання
Кількість кредитів – 4	Вибіркова
Модулів – 7	**Лекції**
Змістових модулів – 16	32 год
Загальна кількість годин – 120	**Лабораторні**
	32 год
	Самостійна робота
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 3,5	56 год
	Вид контролю: залік

Частка аудиторних занять і частка самостійної та індивідуальної роботи у загальному обсязі годин з навчальної дисципліни становить:

для денної форми навчання – 53 % аудиторних занять, 47 % самостійної та індивідуальної роботи.

2. Мета та завдання навчальної дисципліни

Метою вивчення навчальної дисципліни є формування знань і навичок у здобувачів вищої освіти щодо розробки веб-додатків за допомогою PHP-фреймворків.

Завданнями навчальної дисципліни є:

Ознайомлення з основами розробки веб-додатків за допомогою PHP та популярних PHP-фреймворків (Laravel, Symfony, Yii тощо).

Вивчення архітектурних підходів і принципів побудови веб-додатків із використанням фреймворків.

Опанування основних компонентів PHP-фреймворків: маршрутизація, контролери, моделі, шаблони.

Робота з базами даних та ORM для ефективного керування даними.

Реалізація аутентифікації, авторизації та безпеки веб-додатків.

Використання шаблонів проектування для організації коду та підвищення його ефективності.

Тестування веб-додатків і інтеграція з іншими сервісами через API.

Оптимізація продуктивності веб-додатків та використання кешування.

Застосування сучасних підходів до розробки, включаючи RESTful API, WebSocket, а також використання Composer та інші інструменти для управління залежностями.

Під час вивчення навчальної дисципліни здобувачі вищої освіти зможуть отримати наступні Soft skills:

Комунікативні навички: вміння презентувати свої веб-проекти, вести технічні дискусії, працювати в команді, пояснювати технічні рішення не технічним користувачам.

Керування часом: здатність ефективно виконувати проекти в межах дедлайнів, організовувати роботу над складними завданнями.

Гнучкість і адаптивність: уміння швидко опановувати нові технології та інструменти, адаптуватися до змін у галузі веб-розробки.

Особисті якості: аналітичне мислення для розв'язання складних задач, креативний підхід до розробки ефективних і зручних інтерфейсів, увага до деталей при розробці функціональних веб-додатків.

3. Програма навчальної дисципліни

Модуль 1: Вступ до PHP та основи роботи з фреймворками

Змістовий модуль 1. Основи PHP. Синтаксис, змінні, функції та масиви. Взаємодія PHP з базами даних.

Ознайомлення з основами мови програмування PHP, її синтаксисом, функціями та роботою з масивами. Вивчення способів взаємодії PHP з базами даних.

Змістовий модуль 2. Встановлення середовища розробки для PHP. Налаштування серверу та баз даних.

Налаштування локального середовища для розробки PHP-додатків, встановлення та конфігурація Apache, Nginx, MySQL, а також встановлення інструментів для роботи з PHP.

Змістовий модуль 3. Введення в PHP-фреймворки: Laravel, Symfony, Yii. Порівняння та вибір фреймворка.

Огляд основних PHP-фреймворків та принципи вибору відповідного фреймворка для розробки додатків.

Змістовий модуль 4. Встановлення PHP-фреймворків. Створення базового проекту з використанням фреймворка.

Інструкція щодо встановлення Laravel, Symfony, або Yii, та створення базового проекту для подальшої розробки веб-додатків.

Модуль 2: Основи архітектури та роботи з базами даних

Змістовий модуль 5. Архітектура MVC. Розділення логіки на моделі, представлення та контролери.

Ознайомлення з архітектурою MVC та її застосуванням у PHP-фреймворках для кращого розділення логіки програми.

Змістовий модуль 6. Роутинг у PHP-фреймворках. Створення маршрутів та обробка HTTP-запитів.

Вивчення принципів маршрутизації в PHP-фреймворках, налаштування маршрутів для обробки різних типів запитів.

Змістовий модуль 7. Робота з базами даних через ORM. Використання Eloquent у Laravel та Doctrine у Symfony.

Вивчення технології ORM для роботи з базами даних, використання Eloquent та Doctrine для управління базами даних.

Змістовий модуль 8. Міграції та сидери для баз даних. Створення та модифікація бази даних через фреймворк.

Робота з міграціями для створення та оновлення бази даних у PHP-фреймворках, використання сидерів для заповнення бази даних тестовими даними.

Модуль 3: Розширене використання PHP-фреймворків та безпека

Змістовий модуль 9. Аутентифікація та авторизація користувачів. Вбудовані механізми безпеки у PHP-фреймворках.

Реалізація аутентифікації користувачів, створення системи входу та реєстрації,

а також налаштування авторизації для обмеження доступу до певних частин додатку.

Змістовий модуль 10. Валідація форм і обробка помилок. Забезпечення коректності введених даних.

Розбір методів валідації даних, введених через форми, обробка помилок та повідомлень для користувачів.

Змістовий модуль 11. Захист від SQL Injection, XSS, CSRF атак. Вбудовані механізми безпеки у PHP-фреймворках.

Забезпечення безпеки веб-додатків за допомогою захисту від SQL Injection, Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF).

Змістовий модуль 12. Шифрування даних та безпечне зберігання паролів.

Ознайомлення з техніками шифрування, використання хешування паролів (bcrypt) для забезпечення безпеки даних користувачів.

Модуль 4: Тестування, деплоймент та оптимізація

Змістовий модуль 13. Юніт-тестування PHP-додатків. Використання PHPUnit для написання та виконання тестів.

Знайомство з принципами юніт-тестування в PHP, використання PHPUnit для написання тестів на окремі компоненти додатку.

Змістовий модуль 14. Інтеграційне тестування та тестування API. Використання Postman та інших інструментів.

Вивчення тестування цілісності додатку та API за допомогою Postman і аналогічних інструментів для перевірки взаємодії між компонентами.

Змістовий модуль 15. Оптимізація продуктивності PHP-додатків. Кешування та управління пам’яттю.

Вивчення методів оптимізації PHP-додатків, використання кешування, фреймбуферів та інших стратегій для покращення продуктивності.

Змістовий модуль 16. Деплоймент додатків на сервери. Використання Docker, CI/CD для автоматизації деплою.

Знайомство з процесами деплою додатків на реальні сервери, використання Docker для контейнеризації додатків, налаштування процесу CI/CD для автоматизації деплою.

4. Структура (тематичний план) навчальної дисципліни

	Змістовні модулі	Кількість годин			
		Всього	Лекції	Лабораторні роботи	Самостійна робота
1	2	3	4	5	6
1	**Модуль 1. Вступ до PHP та основи роботи з фреймворками**				
	Змістовий модуль 1.				
	Тема 1. *Основи PHP. Синтаксис, змінні, функції та масиви. Взаємодія PHP з базами даних.* *Ознайомлення з основами мови програмування PHP, її синтаксисом, функціями та роботою з масивами. Вивчення способів взаємодії PHP з базами даних.*		***2***	***1***	***3***
	Разом змістовий модуль 1				
	Змістовий модуль 2.				
	Тема 2. *Встановлення середовища розробки для PHP. Налаштування серверу та баз даних.* *Налаштування локального середовища для розробки PHP-додатків, встановлення та конфігурація Apache, Nginx, MySQL, а також встановлення інструментів для роботи з PHP.*		***2***	***2***	***3***
	Разом змістовий модуль 2				
2	**Модуль 2. Основи архітектури та роботи з базами даних**				
	Змістовий модуль 3.				
	Тема 3. Введення в PHP-фреймворки: Laravel, Symfony, Yii. Порівняння та вибір фреймворка. Огляд основних PHP-фреймворків та принципи вибору відповідного фреймворка для розробки додатків.		***2***	***3***	***3***
	Разом змістовний модуль 3				
	Змістовий модуль 4.				
	Тема 4. Встановлення PHP-фреймворків. Створення базового проекту з використанням фреймворка. ***Інструкція щодо встановлення Laravel, Symfony, або Yii, та створення базового проекту для подальшої розробки веб-додатків.***		***2***	***2***	***3***
	Разом змістовний модуль 4				
	Змістовий модуль 5.				
	Тема 5. Архітектура MVC. Розділення логіки на моделі, представлення та контролери. Ознайомлення з архітектурою MVC та її застосуванням у PHP-фреймворках для кращого розділення логіки програми.		***2***	***2***	***3***
	Разом змістовний модуль 5				
3	**Модуль 3. Розширене використання PHP-фреймворків та безпека**				
	Змістовий модуль 6.				

	Тема 6. Роутинг у PHP-фреймворках. Створення маршрутів та обробка HTTP-запитів. ***Вивчення принципів маршрутизації в PHP-фреймворках, налаштування маршрутів для обробки різних типів запитів.***		***2***	***2***	***3***
	Разом змістовний модуль 6				
	Змістовий модуль 7.				
	Тема 7. Робота з базами даних через ORM. Використання Eloquent у Laravel та Doctrine у Symfony. Вивчення технології ORM для роботи з базами даних, використання Eloquent та Doctrine для управління базами даних.		***2***	***2***	***4***
	Разом змістовний модуль 7				
	Змістовий модуль 8.				
	Тема 8. Міграції та сидери для баз даних. Створення та модифікація бази даних через фреймворк. Робота з міграціями для створення та оновлення бази даних у PHP-фреймворках, використання сидерів для заповнення бази даних тестовими даними.		***2***	***2***	***3***
	Разом змістовний модуль 8				
	Змістовий модуль 9.				
	Тема 9. Аутентифікація та авторизація користувачів. Вбудовані механізми безпеки у PHP-фреймворках. Реалізація аутентифікації користувачів, створення системи входу та реєстрації, а також налаштування авторизації для обмеження доступу до певних частин додатку.		***2***	***2***	***3***
	Разом змістовний модуль 9				
4	**Модуль 4. Тестування, деплоймент та оптимізація**				
	Змістовий модуль 10.				
	Тема 10. Валідація форм і обробка помилок. Забезпечення коректності введених даних. Розбір методів валідації даних, введених через форми, обробка помилок та повідомлень для користувачів.		***2***	***2***	***2***
	Разом змістовний модуль 10				
	Змістовий модуль 11.				
	Тема 11. Захист від SQL Injection, XSS, CSRF атак. Вбудовані механізми безпеки у PHP-фреймворках. Забезпечення безпеки веб-додатків за допомогою захисту від SQL Injection, Cross-Site Scripting (XSS) та Cross-Site Request Forgery (CSRF).		***2***	***2***	***2***
	Разом змістовний модуль 11				
	Змістовий модуль 12.				
	Тема 12. Шифрування даних та безпечне зберігання паролів. Ознайомлення з техніками шифрування, використання хешування паролів (bcrypt) для забезпечення безпеки даних		***2***	***2***	***2***

	користувачів.				
	Разом змістовний модуль 12				
	Змістовий модуль 13.				
	Тема 13. Юніт-тестування PHP-додатків. Використання PHPUnit для написання та виконання тестів. Знайомство з принципами юніт-тестування в PHP, використання PHPUnit для написання тестів на окремі компоненти додатку.		***2***	***2***	***4***
	Разом змістовний модуль 13				
	Змістовий модуль 14.				
	Тема 14. Інтеграційне тестування та тестування API. Використання Postman та інших інструментів. Вивчення тестування цілісності додатку та API за допомогою Postman і аналогічних інструментів для перевірки взаємодії між компонентами.		***2***	***2***	***4***
	Разом змістовний модуль 14				
	Змістовий модуль 15.				
	Тема 15. Оптимізація продуктивності PHP-додатків. Кешування та управління пам'яттю. Вивчення методів оптимізації PHP-додатків, використання кешування, фреймбуферів та інших стратегій для покращення продуктивності.		***2***	***2***	***4***
	Разом змістовний модуль 15				
	Змістовий модуль 16.				
	Тема 16. Деплоймент додатків на сервери. Використання Docker, CI/CD для автоматизації деплою. Знайомство з процесами деплою додатків на реальні сервери, використання Docker для контейнеризації додатків, налаштування процесу CI/CD для автоматизації деплою.		***2***	***2***	***4***
	Разом змістовний модуль 16				
	ВСЬОГО	***150***	***32***	***32***	***56***

5. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
		денна форма
1.	Лабораторна робота №1. Встановлення середовища розробки. Встановлення PHP, веб-сервера (Apache/Nginx), бази даних MySQL/PostgreSQL. Ознайомлення з Composer, PHP-фреймворками (Laravel, Symfony, Yii).	4
2.	Лабораторна робота №2. Основи роботи з маршрутизацією у PHP-фреймворках. Створення маршрутів, обробка HTTP-запитів, передача параметрів у маршрутах.	4
3.	Лабораторна робота №3. Робота з базами даних через ORM. Ознайомлення з Eloquent (Laravel) або Doctrine (Symfony). Створення моделей, взаємозв'язків та виконання SQL-запитів через ORM.	4
4.	Лабораторна робота №4. Міграції та сидери бази даних. Створення, оновлення та наповнення бази даних за допомогою інструментів міграцій та сидерів.	4
5.	Лабораторна робота №5. Аутентифікація та авторизація у PHP-фреймворках. Реалізація реєстрації, входу користувачів, керування ролями та правами доступу.	4
6.	Лабораторна робота №6. Валідація форм та обробка помилок. Налаштування валідації введених даних, створення повідомлень про помилки.	4
7.	Лабораторна робота №7. REST API у PHP-фреймворках. Створення RESTful API, обробка запитів, повернення JSON-даних.	4
8.	Лабораторна робота №8. Деплоймент PHP-додатків. Налаштування серверного середовища, завантаження додатка на сервер, налаштування CI/CD, використання Docker.	4
РАЗОМ		**32**

6. Завдання для самостійної роботи

№	Назва теми	Кількість годин
		денна форма
1.	Створення власного проекту на PHP-фреймворку. Вибір Laravel, Symfony або Yii, створення каркасу проекту.	3
2.	Реалізація CRUD-операцій через ORM. Створення, читання, оновлення та видалення записів у базі даних за допомогою ORM.	3
3.	Створення ролей користувачів та керування правами доступу. Реалізація системи ролей (адміністратор, користувач тощо) та перевірка прав доступу.	4
4.	Робота з API: інтеграція сторонніх сервісів. Використання API (наприклад, OpenWeather, Google Maps, Stripe) у PHP-додатку.	6
5.	Оптимізація продуктивності PHP-додатка. Використання кешування, оптимізація запитів до бази даних, налаштування кешу (Redis, Memcached).	6
6.	Тестування PHP-додатків. Написання юніт-тестів та інтеграційних тестів за допомогою PHPUnit.	6
7.	Розширене логування та моніторинг додатку. Використання Laravel Telescope, Symfony Monolog, налаштування логування помилок та запитів.	10
8.	Автоматизація процесів розгортання. Використання Docker, CI/CD (GitHub Actions, GitLab CI), налаштування автоматичного деплою.	12
РАЗОМ		**56**

7. Індивідуальні завдання

Індивідуальні заняття не передбачено навчальним планом

8. Методи навчання

Під час викладання навчальної дисципліни використовуються наступні методи навчання.

- Вербальні методи (лекція, пояснення)
- Наочні методи (спостереження, демонстрація, ілюстрація)
- Практичні методи (виконання практичних завдань)
- Дискусійний метод
- Метод активного навчання (мозковий штурм)
- Ситуаційний метод
- Методи самостійної роботи (проведення розрахунків)

9. Методи контролю

Перевірка досягнення результатів навчання здійснюється з використанням наступних методів.

- Усне опитування, участь у дискусії, відповіді на проблемні запитання
- Перевірка виконання та захист лабораторних робіт
- Експрес-тестування
- Перевірка виконання завдань модульного контролю
- Залік

10. Оцінювання результатів навчання здобувачів вищої освіти

Оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни здійснюється відповідно до Положення про оцінювання результатів навчання здобувачів вищої освіти у Державному університеті «Житомирська політехніка» та розподілу балів, що наведений нижче.

Система оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни включає поточний та підсумковий контроль.

Поточний контроль проводиться для оцінювання рівня засвоєння знань, формування умінь і навичок здобувачів вищої освіти впродовж вивчення ними матеріалу модуля (змістових модулів) навчальної дисципліни. Поточний контроль здійснюється під час проведення навчальних занять.

Підсумковий контроль проводиться для підсумкового оцінювання результатів навчання здобувачів вищої освіти з навчальної дисципліни. Підсумковий контроль здійснюється після завершення вивчення навчальної дисципліни. Підсумковий контроль проводиться у формі заліку. Процедура складання заліку визначена у Положенні про організацію освітнього процесу у Державному університеті «Житомирська політехніка».

Розподіл балів з навчальної дисципліни

Види робіт здобувача вищої освіти	Кількість балів за семестр
	денна форма
Виконання завдань поточного контролю	100
Підсумкова семестрова оцінка	**100**

Розподіл балів за виконання завдань під час навчальних занять

Види робіт здобувача вищої освіти	Кількість балів за семестр	
	денна форма	заочна форма
Виконання та захист лабораторних робіт	48	–
Відповіді (виступи) за виконане домашнє завдання	12	–
Разом за виконання завдань поточного контролю	**60**	**–**

З метою застосування цілих чисел для оцінювання активностей здобувачів вищої освіти під час навчальних занять протягом семестру використовується 100бальна шкала оцінювання кожного окремо виду робіт. Розрахунок набраних здобувачем вищої освіти балів за виконання завдань під час навчальних занять за семестр проводиться за формулою:

$$Р_{НЗ} = (Р_{ЛР100} \times ВК_{ЛР} + Р_{ПТ100} \times ВК_{ПТ} + Р_{ЛЕК100} \times ВК_{ЛЕК} + Р_{ІТЕ100} \times ВК_{ІТЕ}) \times К_{НЗ}, \qquad (1)$$

де $Р_{НЗ}$ – кількість набраних здобувачем вищої освіти балів за виконання завдань під час навчальних занять за семестр;

$Р_{ЛР100}$, $Р_{ПТ100}$, $Р_{ЛЕК100}$, $Р_{ІТЕ100}$ – кількість набраних здобувачем вищої освіти балів за семестр відповідно за виконання та захист лабораторних робіт, виконання поточних тестів, участь у лекційних заняттях, проходження курсу IT Essentials (кожний окремо вид робіт на навчальних заняттях оцінюється за 100бальною шкалою);

$ВК_{ЛР}$, $ВК_{ПТ}$, $ВК_{ЛЕК}$, $ВК_{ІТЕ}$ – вагові коефіцієнти відповідно за виконання та захист лабораторних робіт, виконання поточних тестів, участь у лекційних заняттях, проходження курсу IT Essentials. Значення вагових коефіцієнтів розраховуються шляхом ділення кількості балів, які встановлені за виконання окремого виду робіт під час навчальних занять, на сумарну кількість балів за виконання цих робіт (дані для розрахунку вагових коефіцієнтів наведено в табл. «Розподіл балів за виконання завдань під час навчальних занять»);

$К_{НЗ}$ – коригувальний коефіцієнт, який визначається шляхом ділення кількості балів, що встановлені за виконання завдань під час навчальних занять, на 100 балів.

Якщо здобувач вищої освіти набрав за поточний контроль 60 балів або більше, він може погодити дану оцінку в електронному кабінеті і вона стане семестровою оцінкою за вивчення навчальної дисципліни.

Якщо здобувач вищої освіти під час вивчення навчальної дисципліни набрав 60 балів або більше і бажає покращити свій результат успішності, він проходить процедуру підсумкового контролю у формі заліку. За складання заліку здобувач вищої освіти може набрати 100 балів. Семестрова оцінка з навчальної дисципліни формується за результатами підсумкового контролю.

Здобувач вищої освіти допускається до процедури підсумкового контролю у формі заліку, якщо за виконання завдань поточного контролю набрав 50 балів або більше.

Якщо здобувач вищої освіти за результатами поточного контролю набрав 35−49 балів, він отримує право за власною заявою опанувати окремі теми (змістові модулі) навчальної дисципліни понад обсяги, встановлені навчальним планом освітньої програми. Вивчення окремих складових навчальної дисципліни понад обсяги, встановлені навчальним планом освітньої програми, здійснюється у вільний від занять здобувача вищої освіти час.

Якщо здобувач вищої освіти за результатами поточного контролю набрав від 0 до 34 балів (включно), він вважається таким, що не виконав вимоги робочої програми навчальної дисципліни та має академічну заборгованість. Здобувач вищої освіти отримує право за власною заявою опанувати навчальну дисципліну у наступному семестрі понад обсяги, встановлені навчальним планом освітньої програми[1].

Процедура надання додаткових освітніх послуг здобувачу вищої освіти з метою вивчення навчального матеріалу дисципліни понад обсяги, встановлені навчальним планом освітньої програми, визначена у Положенні про надання додаткових освітніх послуг здобувачам вищої освіти в Державному університеті «Житомирська політехніка».

Визнання результатів навчання, набутих у неформальній та/або інформальній освіті

Визнання результатів навчання, набутих у неформальній та/або інформальній освіті в рамках окремих тем навчальної дисципліни, здійснюється викладачем за зверненням здобувача вищої освіти та представленням документів, які підтверджують результати навчання (сертифікати, свідоцтва, скріншоти тощо). Рішення про визнання та оцінка за відповідну частину освітнього компонента приймається викладачем за результатами співбесіди зі здобувачем вищої освіти.

Визнання результатів навчання, набутих у неформальній та/або інформальній освіті в рамках цілого освітнього компонента, здійснюється за процедурою, яка визначена у Положенні про організацію освітнього процесу у Державному університеті «Житомирська політехніка».

Шкала оцінювання

Шкала ЄКТС	Національна шкала	100-бальна шкала
A	Зараховано	90-100
B	Зараховано	82-89
C		74-81
D	Зараховано	64-73
E		60-63
FX	Не зараховано	35-59
F	Не зараховано	0-34

11. Глосарій

№ з/п	Термін державною мовою	Відповідник англійською мовою
1	Модель	Model
2	Представлення	View
3	ORM (Об'єктно-реляційне відображення)	Object-Relational Mapping (ORM)
4	Запит HTTP	HTTP Request
5	Відповідь HTTP	HTTP Response
6	Сесія	Session
7	Кешування	Caching
8	Middleware (Проміжне програмне забезпечення)	Middleware
9	API (Інтерфейс програмування застосунків)	Application Programming Interface (API)
10	Аутентифікація	Authentication
11	Авторизація	Authorization

12. Рекомендована література

1. Stauffer, M. (2023). Laravel: Up & Running: A Framework for Building Modern PHP Apps (3rd Edition). O'Reilly Media. – 450 p.
2. Lindeboom, C. (2022). PHP 8 Programming Tips, Tricks, and Best Practices. Packt Publishing. – 350 p.
3. Schreuder, D. (2021). Symfony 5: The Fast Track. SensioLabs. – 250 p.
4. Kuznetsov, A. (2023). Modern PHP Web Development: Building Applications with Laravel 10 and React. Packt Publishing. – 400 p.
5. Boduch, A. (2020). Mastering API Architecture: Designing Robust, Reliable, and Secure APIs with PHP and Laravel. Packt Publishing. – 380 p.
6. MacDonald, M. (2020). PHP 8 Programming: Learn PHP for Web Development. Apress. – 500 p.

Допоміжна література

7. McCool, B. (2021). PHP & MySQL: Server-side Web Development. Pearson. – 800 p.
8. Sams, T. (2019). PHP and MySQL Web Development (5th Edition). Addison-Wesley. – 700 p.
9. Shah, J. (2022). PHP 8 Solutions: Dynamic Web Design Made Easy. Apress. – 480 p.
10. Campbell, A. (2023). The Laravel Survival Guide: Best Practices for Modern PHP Development. Packt Publishing. – 350 p.
11. Adams, C. (2020). Pro Symfony: Application Development with PHP Frameworks. Apress. – 400 p.

12. Інформаційні ресурси в Інтернеті

12. Офіційна документація PHP: https://www.php.net/docs.php
13. Офіційна документація Laravel: https://laravel.com/docs
14. Офіційна документація Symfony: https://symfony.com/doc/current/index.html
15. Пакети для PHP: https://packagist.org/
16. Туторіали Laravel: https://laracasts.com/
17. Документація API Platform: https://api-platform.com/docs/
18. Форум PHP: https://www.php-forum.com/