

**МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ВНУТРІШНІХ СПРАВ**  
**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ УПРАВЛІННЯ, ПСИХОЛОГІЇ**  
**ТА БЕЗПЕКИ**

**Кафедра інформаційних технологій**

**РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ**  
**СТВОРЕННЯ ВЕБ-САЙТУ**

**Кваліфікаційна робота**  
здобувача вищої освіти  
4 курсу заочної форми навчання  
**Флоріани-Анастасії ОЛЕНЮК**

**Науковий керівник:**

**Сергій КУТАЄВ**

**Рецензент:**

\_\_\_\_\_

вчене звання, науковий ступінь

\_\_\_\_\_

(Ім'я ПРИЗВИЩЕ рецензента)

***Кваліфікаційна робота допущена до захисту***

«\_\_\_» \_\_\_\_\_ 2026 р., протокол № \_\_\_\_\_

Завідувач кафедри інформаційних технологій

\_\_\_\_\_ Олег ЗАЧЕК

(підпис)

Львів  
2026

## АНОТАЦІЯ

**ОЛЕНЮК Ф.-А.** Розробка інформаційної системи автоматизації створення веб-сайту. – Рукопис.

Дослідження на здобуття освітнього ступеня «бакалавр» за спеціальністю 126 «Інформаційні системи та технології». – Львівський державний університет внутрішніх справ, МВС України, Львів, 2026.

У процесі дослідження проаналізовано сучасні підходи до автоматизації веб-розробки, досліджено можливості фреймворка Bedrock, інструменту PHP Deployer та технологій штучного інтелекту. Розроблено концепцію системи, її архітектуру та ієрархію процесів. Реалізовано прототип системи, який забезпечує швидку ініціалізацію проєктів, автоматизований деплой та автоматичну генерацію якісного текстового контенту безпосередньо в редакторі Gutenberg.

**Ключові слова:** інформаційна система, веб-розробка, автоматизація, Roots/Bedrock, PHP Deployer, штучний інтелект, AI Engine, WordPress.

## ABSTRACT

**OLENYUK F.-A.** Development of an Information System for Automating the Creation of Websites. – Manuscript.

Research for obtaining a bachelor's degree in specialty 126 «Information systems and technologies». – Lviv State University of Internal Affairs, MIA of Ukraine, Lviv, 2026.

In the course of the study, contemporary approaches to web development automation were analyzed, and the possibilities of the Bedrock framework, the PHP Deployer tool, and artificial intelligence technologies were explored. The concept of the system, its architecture, and the process hierarchy were developed.

A functional prototype of the system was implemented, enabling fast project initialization, automated deployment to servers, and automatic generation of high-quality textual content directly within the Gutenberg editor.

**Keywords:** information system, web development, automation, Roots/Bedrock, PHP Deployer, artificial intelligence, AI Engine, WordPress.

## ЗМІСТ

ВСТУП.....	6
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ.....	9
1.1. Аналіз предметної області. Обґрунтування необхідності створення програмного продукту.....	9
1.2. Методологія пошуку наукових джерел.....	10
1.3. Огляд відомих аналогічних інформаційних систем.....	12
РОЗДІЛ 2 СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИБІР МЕТОДІВ РОЗРОБКИ ВЕБ-САЙТУ.....	14
2.1. Дерево цілей.....	14
2.2. Конкретизація функціонування системи.....	23
2.2.1. Опис функціональних можливостей системи.....	24
2.2.2. Підготовка середовища розробки.....	25
2.2.3. Процес автоматизованого деплою за допомогою PHP Deployer....	27
2.2.4. Автоматизація створення контенту через плагіни з.....	28
використанням ШІ.....	28
2.3. Побудова ієрархії процесів.....	30
2.3.1. Ієрархія процесів автоматизації розгортання сайтів.....	30
2.2.3. Ієрархія процесів автоматичного створення контенту.....	31
2.4. Вибір та обґрунтування засобів розв'язання задачі.....	33
2.5. Технічні характеристики обраних програмних засобів для розробки.....	32
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	36
3.1. Опис розробки інформаційної системи.....	36
3.2. Інструкція користувача.....	39
3.3. Аналіз розробленої інформаційної системи.....	47
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	51
ДОДАТКИ.....	53

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

**AI** – Artificial Intelligence (Штучний інтелект)

**CI/CD** – Continuous Integration / Continuous Deployment (Безперервна інтеграція / Безперервне розгортання)

**CMS** – Content Management System (Система керування контентом)

**Deployer** – PHP Deployer (інструмент автоматизації деплою)

**DevOps** – Development and Operations

**ER** – Entity-Relationship (діаграма «сутність-зв'язок»)

**Git** – Система контролю версій

**MAI** – Метод аналітичної ієрархії

**SEO** – Search Engine Optimization (Оптимізація для пошукових систем)

**SSH** – Secure Shell

**WP** – WordPress

**WP-CLI** – WordPress Command Line Interface

## ВСТУП

У сучасних умовах стрімкого розвитку цифрових технологій та веб-розробки автоматизація процесу створення та розгортання веб-сайтів є одним із ключових чинників підвищення ефективності роботи розробників, студій та компаній. Зростання попиту на швидке виведення веб-ресурсів на ринок зумовлює потребу у сучасних інструментах, які дозволяють стандартизувати процеси, мінімізувати ручну працю, підвищити безпеку та забезпечити стабільне управління залежностями. Значну допомогу в цьому надає використання спеціалізованих фреймворків, зокрема Roots/Bedrock у поєднанні з PHP Deployer та інструментами на базі штучного інтелекту.

Інформаційна система автоматизації створення веб-сайтів на базі WordPress з використанням Roots/Bedrock, PHP Deployer та плагіна AI Engine дозволяє суттєво спростити процеси ініціалізації проєктів, розгортання їх на серверах та наповнення якісним контентом. Такий програмний продукт є актуальним інструментом для веб-студій, фриланс-розробників та компаній, які регулярно створюють та підтримують сайти на WordPress, сприяючи автоматизації бізнес-процесів і значному скороченню часу від ідеї до запуску.

Необхідність розробки такої системи обумовлюється постійним зростанням обсягів веб-проєктів, високими вимогами до безпеки та швидкості розгортання, а також трудомісткістю ручного управління залежностями, оновленнями та контентом. Подібні рішення особливо корисні для невеликих команд та індивідуальних розробників, які прагнуть оптимізувати робочі процеси та зосередитися на креативній частині роботи.

Володіючи сучасним набором функцій, розроблювана інформаційна система покликана забезпечити швидке, безпечне та ефективне створення сучасних веб-сайтів.

Із викладеного випливає **актуальність теми** кваліфікаційної роботи –

зростаюча потреба у автоматизованих рішеннях для сучасної веб-розробки, які поєднують найкращі практики DevOps з можливостями штучного інтелекту.

**Аналіз останніх досліджень і публікацій.** Питанням автоматизації веб-розробки, CI/CD-процесів та використання сучасних інструментів для WordPress присвячено значну кількість наукових праць та технічної документації українських і зарубіжних авторів. Серед них можна виділити роботи, що розглядають фреймворк Bedrock [3, 12], інструменти автоматизованого деплою (PHP Deployer, Capistrano) [1, 8], а також дослідження у сфері інтеграції штучного інтелекту в процес створення контенту. Однак швидкий розвиток технологій, поява нових версій WordPress, оновлення Composer та зростання вимог до безпеки й масштабованості вимагають подальшого удосконалення комплексних рішень. У зв'язку з цим питання розробки інформаційної системи автоматизації створення веб-сайтів на базі Roots/Bedrock залишається актуальним.

**Метою** кваліфікаційної роботи є розробка інформаційної системи автоматизації створення веб-сайтів із використанням Roots/Bedrock, PHP Deployer та інструментів штучного інтелекту.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область та існуючі підходи до автоматизації веб-розробки;
- дослідити можливості Roots/Bedrock для керування залежностями та середовищем розробки;
- розробити концепцію та архітектуру інформаційної системи;
- реалізувати прототип системи та здійснити її тестування;
- проаналізувати ефективність розробленого рішення.

**Об'єктом** дослідження є процеси автоматизації, створення та розгортання веб-сайтів.

**Предметом** дослідження виступають методи та засоби автоматизації веб-розробки на базі Roots/Bedrock, PHP Deployer та штучного інтелекту.

**Методи дослідження.** У роботі застосовано загальнонаукові методи: аналіз і синтез, порівняння, системний підхід, моделювання, а також спеціальні методи веб-розробки, включаючи використання Composer, Git, PHP Deployer та інтеграцію API штучного інтелекту.

**Практичне значення одержаних результатів** полягає у значному скороченні часу на запуск нових веб-проектів, підвищенні безпеки, стандартизації процесів та зменшенні рутинної роботи розробників і контент-менеджерів. Розроблена система може бути ефективно використана веб-студіями, фрилансерами та компаніями, що працюють з WordPress.

**Структура роботи.** Бакалаврська кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Обсяг основного тексту роботи складає 42 сторінок, 23 рисунки і 8 таблиць і 20 бібліографічних джерел. Загальний обсяг роботи – 59 сторінок.

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ

#### **1.1. Аналіз предметної області. Обґрунтування необхідності створення програмного продукту**

Швидке впровадження цифрових технологій у різні сфери економіки зробило веб-сайти одним із основних інструментів взаємодії організацій з клієнтами, партнерами і суспільством. Наявність якісного веб-ресурсу сьогодні є не лише важливим аспектом репутації, а й необхідною умовою для забезпечення конкурентоспроможності бізнесу.

Попри значний розвиток засобів веб-розробки, створення та подальше супроводження сайтів залишаються складними процесами. Вони потребують залучення кваліфікованих фахівців і значних витрат часу. Ця проблема особливо відчутна для малих та середніх підприємств, ресурси яких є обмеженими.

Одним із перспективних напрямків покращення веб-розробки є автоматизація процесів створення, розгортання та обслуговування веб-ресурсів. Використання відповідних програм дозволяє зменшити вплив людського фактора, підвищити стабільність роботи системи і скоротити витрати на її підтримку.

Серед найбільш поширених проблем традиційного підходу до розроблення веб-сайтів варто виокремити:

- значну тривалість процедур підготовки та розгортання програмного середовища;
- складність контролю за актуальністю залежностей і програмних компонентів;
- необхідність постійного моніторингу стану інформаційної безпеки;
- трудомісткість процесів створення та оновлення контенту;

- ризик виникнення помилок під час внесення змін до програмного коду;
- труднощі забезпечення узгодженого керування версіями програмного забезпечення.

Усунення зазначених недоліків можливе шляхом впровадження сучасних засобів автоматизації. Одним із таких інструментів є PHP Deployer, який забезпечує автоматизоване виконання процедур оновлення та публікації веб-проектів.

Його використання надає такі переваги:

- скорочення часу виконання операцій деплою;
- стандартизацію процесів розгортання;
- можливість адаптації сценаріїв до потреб конкретного проекту;
- підтримку масштабування шляхом роботи з декількома серверами;
- інтеграцію із сучасними інструментами розроблення та супроводу програмного забезпечення.

Поєднання можливостей PHP Deployer із платформою Bedrock створює передумови для побудови безпечних, керованих і технологічно ефективних веб-систем, що відповідають сучасним вимогам до якості програмних продуктів.

## **1.2. Методологія пошуку наукових джерел**

Наукове дослідження потребує систематичного аналізу літературних і технічних джерел. Їх вивчення допомагає зрозуміти сучасний стан проблеми, визначити основні тенденції розвитку галузі та обґрунтувати необхідність дослідження.

Веб-розробка швидко розвивається, тому потрібно постійно оновлювати як теоретичні, так і практичні знання. Через це важливо обирати сучасні публікації, професійну документацію та практичні поради.

Один з основних напрямків сучасної вебінженерії – це автоматизація доставки програмного забезпечення. Дослідники відзначають, що ефективність цього процесу значно підвищується завдяки використанню

концепцій безперервної інтеграції та безперервного розгортання (CI/CD). Вони допомагають зробити впровадження змін у програмні продукти більш надійним і швидким

[1-10].

Для впровадження цих підходів використовують спеціалізовані інструменти, такі як Jenkins, GitHub Actions, GitLab CI/CD та PHP Deployer. Вони автоматизують повторювані операції, зменшують кількість помилок і роблять результати розгортання більш передбачуваними.

Ще одним способом підвищити ефективність сучасних веб-систем є використання контейнеризації, наприклад Docker і Kubernetes. Вони допомагають уніфікувати середовище виконання, спрощують масштабування застосунків і забезпечують стабільну роботу незалежно від особливостей інфраструктури.

У цьому дослідженні особливо розглядається PHP Deployer – інструмент для автоматизації розгортання PHP-застосунків. Його основні особливості такі:

- декларативний підхід до опису сценаріїв деплою;
- підтримка багатосерверного розгортання;
- інтеграція із системами контролю версій;
- можливість гнучкого налаштування процесів;
- підтримка взаємодії з іншими DevOps-рішеннями.

Окремо було проаналізовано інструменти для автоматизації роботи з WordPress. Дослідження показало, що Bedrock допомагає організувати проєкт за сучасними стандартами, Composer полегшує керування залежностями, а WP-CLI дозволяє автоматизувати адміністративні завдання.

Також сучасна практика показує, що технології штучного інтелекту стають все важливішими у веброзробці. Інтелектуальні системи можуть створювати текстові матеріали, аналізувати стиль контенту, формувати описи товарів і підтримувати інформаційне наповнення сайтів.

Отже, аналіз показує, що поєднання інструментів автоматизації та

технологій штучного інтелекту може значно підвищити ефективність створення і підтримки веб-проектів.

### 1.3. Огляд відомих аналогічних інформаційних систем

Сучасний ринок програмних засобів автоматизації представлений значною кількістю інструментів, що відрізняються функціональними можливостями, рівнем складності та сферою застосування.

Одним із найвідоміших рішень є **Jenkins** – система автоматизації процесів безперервної інтеграції та доставки програмного забезпечення. Її перевагами виступають відкритість програмного коду, широкий вибір плагінів та висока адаптивність. Водночас значна складність конфігурації та необхідність постійного адміністрування можуть ускладнювати її використання.

**GitLab CI/CD** реалізує механізми автоматизації безпосередньо в межах платформи GitLab. Такий підхід забезпечує зручність експлуатації та тісну інтеграцію із системою керування версіями, хоча поступається окремим рішенням за рівнем гнучкості налаштувань.

**Capistrano** традиційно використовується для автоматизації деплою вебзастосунків і відзначається простотою організації сценаріїв розгортання. Разом із тим його орієнтація переважно на Ruby-проекти обмежує універсальність використання.

**Buddy** пропонує хмарну модель реалізації процесів CI/CD із використанням візуальних засобів конструювання сценаріїв. Перевагами платформи є швидке впровадження та підтримка сучасних технологій контейнеризації, тоді як фактором стримування може бути комерційна модель ліцензування [1-10].

**PHP Deployer**, своєю чергою, спеціалізується на підтримці PHP-застосунків і забезпечує ефективне автоматизоване розгортання шляхом використання SSH-з'єднань та сценаріїв конфігурації. Простота впровадження, підтримка Git та можливість роботи з декількома серверами

одночасно роблять його привабливим інструментом для веб-проектів відповідного технологічного стеку.

Таблиця 1.1

## Порівняльна таблиця аналогів деплоєра

Інструмент	Основні особливості	Переваги	Недоліки
Jenkins	Відкритий код, CI/CD, велика кількість плагінів	Гнучкість налаштувань, широка підтримка інструментів	Складність початкового налаштування, обслуговування
GitLab CI/CD	Вбудована в GitLab, простий інтерфейс, інтеграція з Git	Простота використання, інтегрованість	Менша гнучкість у порівнянні з Jenkins
Capistrano	Автоматизація деплою, проста інтеграція з Git, мультисерверний деплой	Простота налаштування, легкість використання	Орієнтація переважно на Ruby-додатки
Buddy	Хмарна платформа, візуальний редактор, підтримка Docker/Kubernetes	Просте налаштування, інтеграція з сучасними технологіями	Платна модель, не підходить для малих проектів
PHP Deployer	Орієнтація на PHP-проекти, простий синтаксис налаштувань	Простота налаштувань, гнучкість	Вузька спеціалізація (тільки PHP)

Порівняльний аналіз показав, що навіть при великій кількості схожих рішень на ринку перспективним є створення програмного продукту, який об'єднає переваги PHP Deployer, можливості Bedrock і інструменти штучного інтелекту. Це дозволить автоматизувати технічні процеси, підвищити ефективність управління контентом, зменшити час на супровід вебресурсів і покращити якість кінцевого результату.

## РОЗДІЛ 2

### СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ ТА ВИБІР МЕТОДІВ РОЗРОБКИ ВЕБ-САЙТУ

Системне дослідження процесів розробки веб-сайтів допомагає краще зрозуміти, як пов'язані між собою різні етапи життєвого циклу веб-проекту. Серед них – підготовка і налаштування робочого середовища, організація розгортання продукту на сервері, а також створення і підтримка контенту. Аналіз того, як ці етапи взаємодіють, дозволяє визначити найважливіші частини процесу, де автоматизація може суттєво підвищити ефективність роботи команди розробників.

Детальний аналіз існуючих підходів показує, що багато часу витрачається на повторювані операції, які вимагають великої уваги та часто призводять до помилок через людський фактор. Автоматизація цих процесів допомагає скоротити час виконання, зменшити навантаження на фахівців і підвищити стабільність результатів.

Аналіз сучасних методів деплою та технологій створення контенту дозволив визначити переваги й обмеження існуючих програмних рішень. Результати показують, що доцільно розробити нову інтегровану інформаційну систему, яка об'єднає сучасні інструменти автоматизації та інтелектуальної обробки даних. Для цього можна використати платформу Bedrock, засоби автоматизованого розгортання PHP Deployer і інструменти штучного інтелекту для підтримки процесів генерації та редагування контенту.

#### 2.1. Дерево цілей

Одним із базових етапів системного аналізу є формування чіткої структури цілей, яка відображає взаємозв'язок між головною метою дослідження та комплексом підпорядкованих їй завдань. Побудова дерева

цілей дозволяє систематизувати напрями роботи, визначити очікувані результати та сформувані логічну послідовність дій, необхідних для успішної реалізації проєкту.

Ключовою метою даного дослідження є проєктування та створення інформаційної системи, призначеної для автоматизації процесів розгортання вебресурсів і формування контенту шляхом інтеграції технологій Bedrock, PHP Deployer та інтелектуальних програмних модулів, заснованих на використанні штучного інтелекту.

Для досягнення зазначеної мети було визначено низку стратегічних підцілей:

- удосконалення процесів розгортання веб-сайтів шляхом їх максимальної автоматизації з метою скорочення часових витрат і мінімізації ручних операцій;
- оптимізація керування середовищем розробки, конфігураціями та залежностями проєкту із застосуванням архітектурних можливостей Bedrock;
- реалізація механізмів автоматизованого створення, редагування та вдосконалення контенту за допомогою технологій штучного інтелекту;
- забезпечення належного рівня надійності, інформаційної безпеки та стабільності функціонування веб-ресурсів завдяки використанню стандартизованих процедур;
- підвищення продуктивності праці розробників через скорочення обсягу рутинних і повторюваних завдань.

Реалізація генеральної мети потребує виконання комплексу взаємопов'язаних завдань, серед яких:

- дослідження та узагальнення сучасних підходів до автоматизації процесів веб-розробки;
- обґрунтований вибір програмних засобів і технологічних рішень, зокрема PHP Deployer, Bedrock та інструментів штучного інтелекту;
- проєктування архітектури інформаційної системи з урахуванням інтеграції всіх функціональних компонентів;

- створення програмного продукту та організація його тестування з метою перевірки працездатності;
- впровадження модулів, відповідальних за автоматичне формування контенту та виконання процедур деплою;
- апробація запропонованого рішення в умовах практичного використання та оцінювання рівня його ефективності.

Графічне представлення взаємозв'язків між генеральною метою, підцілями та завданнями наведено на рис. 2.1.

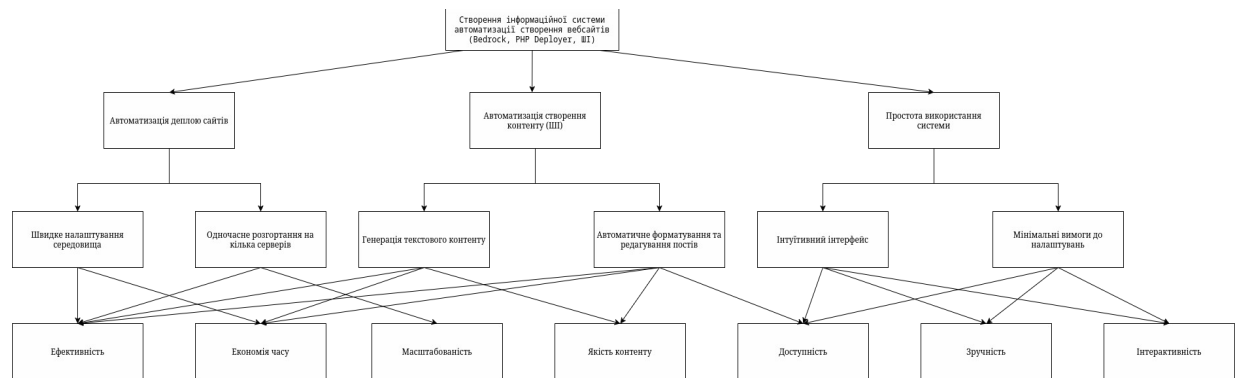


Рис. 2.1. Дерево цілей інформаційної системи

Інформаційна система являє собою сукупність взаємопов'язаних елементів, функціонування яких спрямоване на автоматизацію процесів збору, опрацювання, накопичення, збереження та поширення інформації для підтримки певної предметної області. У контексті дослідження така система орієнтована на забезпечення ефективного виконання завдань веб-розробки та супроводу веб-проектів.

Під час формування концепції інформаційної системи автоматизації створення веб-сайтів на базі Bedrock, PHP Deployer і технологій штучного інтелекту було проаналізовано різні класи інформаційних систем та визначено критерії, за якими здійснюватиметься їх оцінювання. До ключових критеріїв належать ефективність функціонування, зручність використання, доступність і рівень інтерактивності.

Серед основних типів інформаційних систем, що були розглянуті в межах дослідження, виділено такі:

– **інформаційно-консультаційні системи**, призначені для оброблення значних масивів даних та формування рекомендацій щодо оптимізації процесів підтримки й розгортання веб-ресурсів;

– **інформаційно-керівні системи**, функціональність яких забезпечує координацію, контроль і моніторинг виконання технологічних операцій, пов'язаних із деплоєм та управлінням контентом;

– **інформаційно-пошукові системи**, що надають користувачам інструменти оперативного пошуку необхідних відомостей для реалізації завдань веб-розробки та контент-менеджменту;

– **системи підтримки прийняття рішень (СППР)**, які на основі аналізу даних допомагають обирати найбільш доцільні варіанти дій та забезпечують інтерактивну взаємодію з користувачем;

– **інтелектуальні інформаційні системи**, у яких застосовуються алгоритми штучного інтелекту для автоматизації створення, аналізу й опрацювання контенту, а також для підтримки процесів розгортання веб-проектів.

Для визначення найбільш доцільної структури майбутньої системи було використано метод аналітичної ієрархії (МАІ), який належить до інструментів багатокритеріального прийняття рішень. Його застосування дає змогу виконати об'єктивне порівняння альтернатив і встановити пріоритетність окремих характеристик системи.

Реалізація цього методу передбачає проходження кількох послідовних етапів:

– декомпозицію складної проблеми шляхом побудови ієрархічної структури взаємопов'язаних елементів;

– проведення попарного оцінювання критеріїв із використанням дев'ятибальної шкали відносної важливості;

– зіставлення альтернативних типів інформаційних систем для визначення найбільш прийняттого варіанта відповідно до встановлених критеріїв оцінювання.

Застосування методу аналітичної ієрархії забезпечує науково обґрунтований підхід до вибору архітектурних і функціональних рішень, що в подальшому сприятиме створенню інформаційної системи, здатної ефективно задовольняти потреби сучасної веб-розробки та відповідати вимогам щодо продуктивності, безпеки й зручності використання.

Таблиця 2.1

## Ранжування критеріїв

№	Назва критерію	Порівняння критеріїв				Вектор пріоритету
		1	2	3	4	
1	Ефективність	1	3	7	5	0.4
2	Зручність	1/3	1	5	3	0.25
3	Доступність	1/7	1/5	1	1/3	0.15
4	Інтерактивність	1/5	1/3	3	1	0.20

Наступним етапом застосування методу аналітичної ієрархії стало проведення попарного оцінювання визначених критеріїв щодо кожного з обраних типів інформаційних систем. Таке зіставлення дало можливість встановити ступінь відповідності альтернатив поставленим вимогам та визначити їхні відносні переваги.

Для проведення аналізу були розглянуті такі типи інформаційних систем:

- інформаційно-консультаційні системи;
- інформаційно-керівні системи;
- інформаційно-пошукові системи;
- системи підтримки прийняття рішень (СППР);
- інтелектуальні інформаційні системи.

У межах дослідження було виконано порівняльне оцінювання зазначених типів систем за чотирма ключовими критеріями: ефективністю

функціонування, зручністю використання, доступністю та інтерактивністю. Результати попарних порівнянь для кожного з критеріїв подано у відповідних матрицях (табл. 2.2–2.5).

Побудова таких матриць є необхідною складовою методу аналітичної ієрархії, оскільки саме на їх основі здійснюється визначення локальних векторів пріоритетів. Кожне числове значення, наведене в матрицях, характеризує ступінь переваги одного елемента над іншим та відображає відносну значущість конкретного критерію під час оцінювання альтернативних варіантів інформаційних систем. Таким чином, отримані показники дозволяють кількісно описати внесок окремих критеріїв у формування загального результату та встановити їхню вагомість у процесі прийняття рішення.

Таблиця 2.2

Матриця попарного порівняння систем за критерієм «Ефективність»

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1	2	4	1/3	1/5	0.38
2	Інформаційно-керівні	1/2	1	3	1/4	1/4	0.17
3	Інформаційно-пошукові	1/4	1/3	1	1/6	1/7	0.09
4	СППР	3	4	6	1	1/2	0.31
5	Інтелектуальні інформаційні	5	4	7	2	1	0.42

Таблиця 2.3

Матриця попарного порівняння систем за критерієм «Зручність»

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1	3	5	1/2	1/3	0.27
2	Інформаційно-керівні	1/3	1	2	1/4	1/4	0.15
3	Інформаційно-пошукові	1/5	1/2	1	1/3	1/5	0.11
4	СППР	2	4	4	1	1/2	0.32
5	Інтелектуальні інформаційні	3	4	5	2	1	0.43

Таблиця 2.4

Матриця попарного порівняння систем за критерієм «Доступність»

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-консультаційні	1	3	7	2	1/2	0.37
2	Інформаційно-керівні	1/3	1	2	1/2	1/3	0.14
3	Інформаційно-пошукові	1/7	1/2	1	1/4	1/6	0.08
4	СППР	1/2	2	4	1	1/3	0.26
5	Інтелектуальні інформаційні	2	3	6	3	1	0.45

Таблиця 2.5

Матриця попарного порівняння систем за критерієм «Інтерактивність»

№	Назва системи	Порівняння систем					Вектор пріоритету
		1	2	3	4	5	
1	Інформаційно-	1	2	4	1/3	1/4	0.29

	консультаційні						
2	Інформаційно-керівні	1/2	1	3	1/2	1/3	0.19
3	Інформаційно-пошукові	1/4	1/3	1	1/4	1/5	0.1
4	СППР	3	2	4	1	1/2	0.3
5	Інтелектуальні інформаційні	4	3	5	2	1	0.52

Після завершення процедури попарного порівняння було здійснено узагальнення отриманих результатів. Для цього сформовано підсумкову таблицю (табл. 2.6), у якій акумульовано значення коефіцієнтів пріоритетності для кожного типу інформаційних систем відповідно до визначених критеріїв оцінювання. Опрацювання цих даних дало змогу розрахувати інтегральні показники та встановити остаточний рівень пріоритетності досліджуваних альтернатив.

Застосування такого підходу забезпечило об'єктивність і достовірність процесу оцінювання, оскільки дозволило здійснити всебічне порівняння інформаційних систем із урахуванням різних аспектів їх функціонування. Крім того, результати аналізу дали можливість визначити, наскільки кожен тип системи здатний задовольняти потреби потенційних користувачів та відповідати вимогам предметної області.

Сформований фінальний вектор пріоритетів відображає відносну важливість досліджуваних критеріїв та дає змогу виокремити ті характеристики, які мають першочергове значення під час проектування й розроблення інформаційної системи. Отримані значення слугують підґрунтям для прийняття обґрунтованих проектних рішень та визначення напрямів подальшого вдосконалення системи з урахуванням її функціональної ефективності, зручності використання, доступності та рівня взаємодії з користувачем.

## Результати методу аналітичної ієрархії

<b>Критерій</b>	<b>Ефективність</b>	<b>Зручність</b>	<b>Доступність</b>	<b>Інтерактивність</b>
<b>Тип систем</b>				
Інформаційно-консультаційні	0.38	0.27	0.37	0.29
Інформаційно-керівні	0.17	0.15	0.14	0.19
Інформаційно-пошукові	0.09	0.11	0.08	0.1
СППР	0.31	0.32	0.26	0.3
Інтелектуальні інформаційні	0.42	0.43	0.45	0.52
<b>Фінальний вектор критеріїв</b>	<b>0.274</b>	<b>0.256</b>	<b>0.26</b>	<b>0.28</b>

Результати підсумкового аналізу, отримані на основі фінальної матриці попарних порівнянь, свідчать про неоднаковий ступінь значущості визначених критеріїв у процесі проектування інформаційної системи. Найвищий показник інтегрального вектора пріоритетів було зафіксовано для критерію інтерактивності, значення якого становить 0,28. Це дає підстави стверджувати, що саме забезпечення якісної взаємодії між користувачем і системою є визначальним чинником під час створення сучасного програмного продукту.

Дещо нижчі, проте близькі за величиною результати продемонстрували критерії ефективності та доступності, значення яких склали 0,26. Отримані показники підтверджують важливість забезпечення високої продуктивності

системи, її здатності виконувати поставлені завдання з мінімальними витратами ресурсів, а також необхідність створення умов для зручного доступу користувачів до функціональних можливостей програмного забезпечення. Найменше значення фінального вектора пріоритетів отримав критерій зручності використання – 0,256. Незважаючи на те, що цей показник є найнижчим серед досліджуваних, він також має суттєвий вплив на загальну оцінку системи та повинен враховуватися під час її розроблення.

## **2.2. Конкретизація функціонування системи**

Визначення особливостей функціонування розроблюваної інформаційної системи передбачає детальний розгляд її структурних компонентів, функціональних можливостей та механізмів взаємодії між окремими підсистемами. Такий підхід дозволяє сформулювати цілісне уявлення про принципи роботи програмного продукту, а також встановити послідовність виконання основних процесів, які забезпечують досягнення поставлених цілей.

У цьому підрозділі розкриваються ключові функції інформаційної системи автоматизації створення веб-сайтів, основою якої є інтеграція технологій Bedrock, PHP Deployer та інструментів штучного інтелекту. Особливу увагу приділено опису механізмів автоматизованого розгортання веб-проєктів, а також процесам інтелектуальної генерації та редагування контенту. Загальне уявлення про взаємодію користувача з функціональними модулями системи наведено у вигляді контекстної діаграми (рис. 2.2), яка відображає інформаційні потоки та взаємозв'язки між основними елементами програмного рішення.

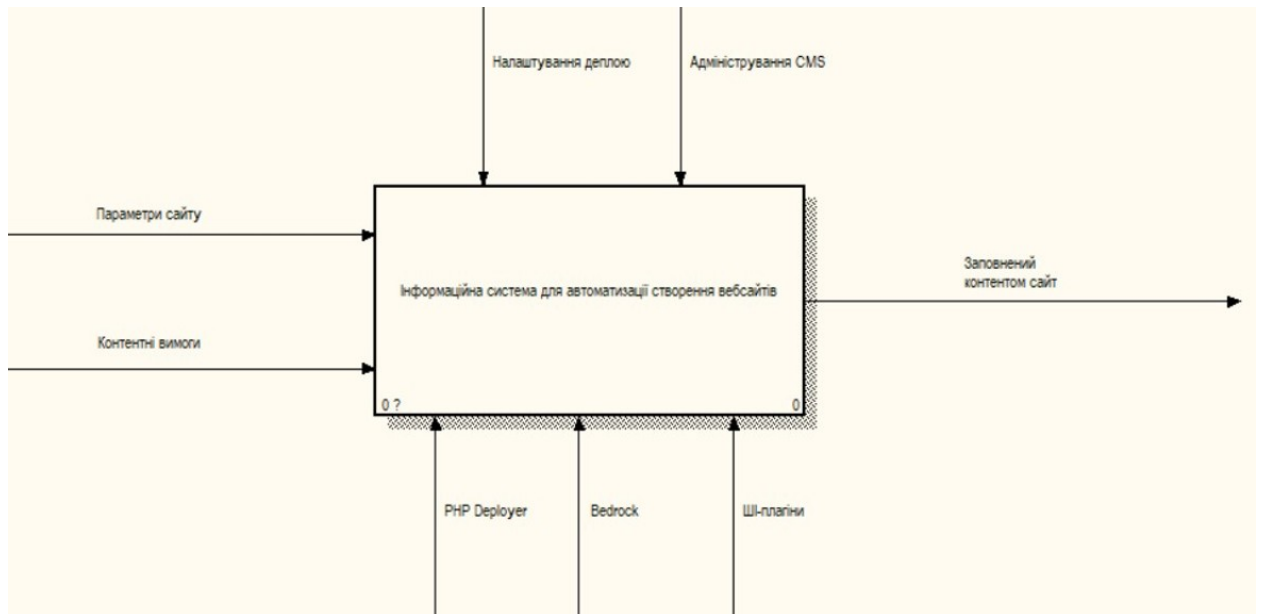


Рис. 2.2. Контекстна діаграма інформаційної системи

### 2.2.1. Опис функціональних можливостей системи

Розроблювана інформаційна система орієнтована на комплексну автоматизацію процесів створення, налаштування та подальшого розгортання веб-ресурсів. Її функціональна архітектура сформована таким чином, щоб мінімізувати участь користувача у виконанні повторюваних операцій та забезпечити високий рівень надійності всіх технологічних процесів.

До основних функціональних можливостей системи належать:

- централізоване керування середовищем розробки із використанням можливостей платформи Bedrock;
- автоматизоване виконання процедур розгортання веб-проектів за допомогою інструменту PHP Deployer;
- інтеграція спеціалізованих модулів на базі штучного інтелекту, призначених для автоматичного створення, редагування та вдосконалення контенту;
- надання користувачеві інтуїтивно зрозумілих засобів управління, контролю та моніторингу виконання основних операцій;

– забезпечення стійкості функціонування веб-сайтів, підвищення рівня їх інформаційної безпеки та підтримка можливостей масштабування відповідно до потреб проєкту.

Сукупність зазначених функцій формує основу роботи інформаційної системи та забезпечує її здатність ефективно підтримувати повний цикл створення й супроводження веб-ресурсів.

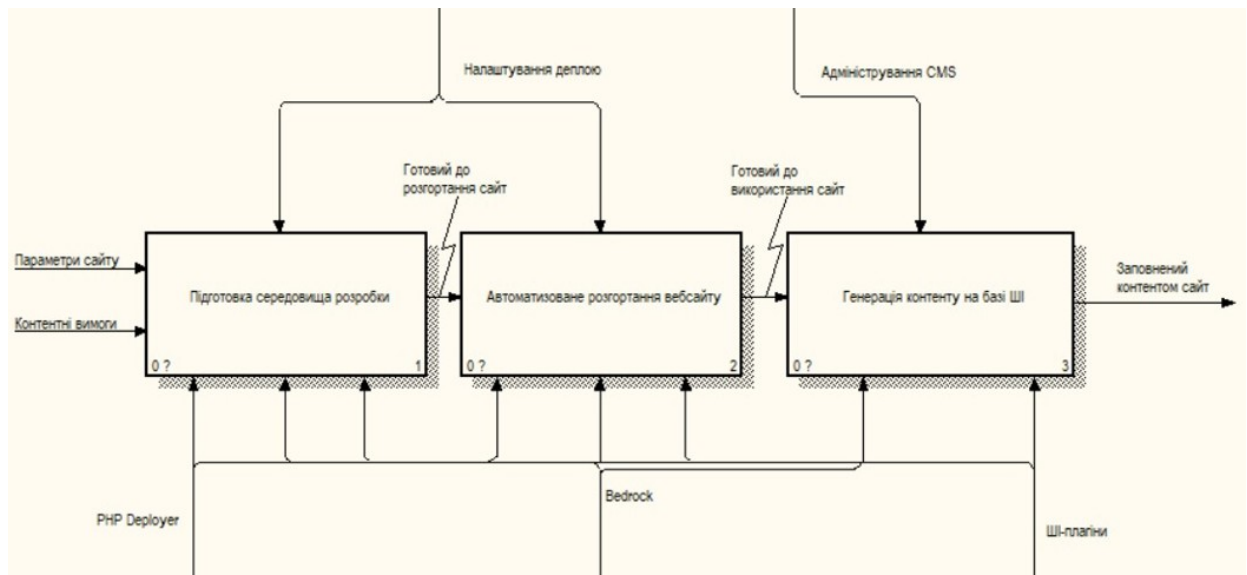


Рис 2.3. Декомпозиція першого рівня

### 2.2.2. Підготовка середовища розробки

Одним із перших етапів функціонування системи є організація середовища розробки, що передбачає виконання комплексу підготовчих процедур, необхідних для подальшої автоматизації процесів створення та розгортання веб-проєкту. Початкові дії включають ініціалізацію проєкту на базі Bedrock, встановлення необхідних програмних залежностей за допомогою Composer та виконання базового налаштування WordPress.

Застосування такого підходу дозволяє сформувати єдину стандартизовану структуру проєкту незалежно від специфіки його реалізації, що сприяє уніфікації процесів розроблення та спрощує подальше адміністрування системи (рис. 2.3–2.4). Саме на цьому етапі закладаються фундаментальні принципи організації веб-проєкту, які забезпечують можливість ефективної інтеграції інструментів автоматизації.

Підготовка середовища розробки охоплює виконання таких основних дій:

- створення та ініціалізацію WordPress-проєкту із застосуванням архітектурних можливостей Bedrock, що забезпечує впорядковану структуру каталогів і конфігурацій відповідно до сучасних практик програмної інженерії;

- встановлення та налаштування програмних розширень і плагінів, необхідних для реалізації функцій генерації контенту, оптимізації пошукової видачі, кешування даних та інших можливостей, передбачених технічними вимогами проєкту;

- виконання первинного конфігурування системи керування контентом, включаючи створення основних типів матеріалів, підготовку базових сторінок, налаштування ролей користувачів і визначення ключових параметрів функціонування адміністративної частини веб-ресурсу.

Реалізація зазначених процедур є важливою складовою загальної архітектури інформаційної системи, оскільки саме на етапі підготовки середовища створюються необхідні передумови для подальшого автоматизованого розгортання веб-сайтів, інтеграції інтелектуальних сервісів та забезпечення стабільної роботи всіх функціональних компонентів програмного продукту.

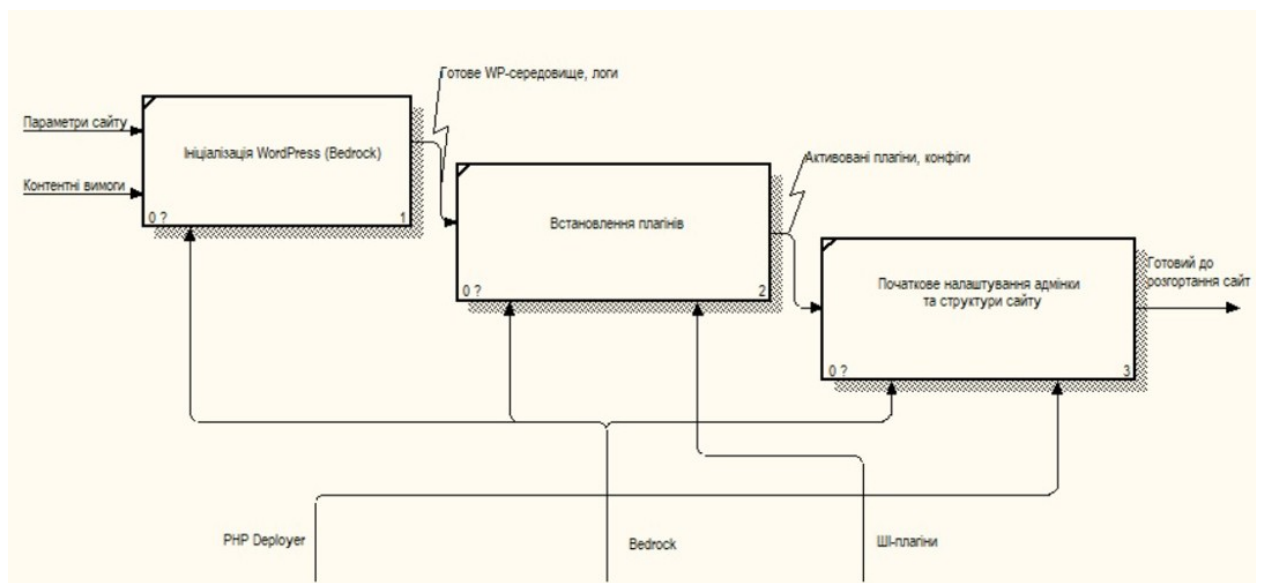


Рис. 2.4. Декомпозиція другого рівня «Підготовка середовища розробки»

### 2.2.3. Процес автоматизованого деплою за допомогою PHP Deployer

Одним із ключових компонентів розробленої інформаційної системи є механізм автоматизованого впровадження веб-проектів у робоче середовище. Для реалізації цього завдання використовується інструмент PHP Deployer, який забезпечує стандартизоване та контрольоване перенесення програмного коду на цільові сервери без необхідності виконання значної кількості ручних операцій.

Процедура автоматизованого розгортання охоплює декілька взаємопов'язаних етапів. На початковому етапі формується сценарій деплою, що містить набір конфігураційних параметрів та правил виконання операцій. Далі здійснюється підготовка програмного оточення, підключення необхідних бібліотек і пакетів керування залежностями на основі архітектури Bedrock. Після цього виконується автоматична передача програмних компонентів із локального середовища розробки до віддаленої серверної інфраструктури. Завершальною стадією є верифікація коректності виконаних операцій, контроль успішності оновлення та фіксація нової версії програмного забезпечення.

Застосування автоматизованого підходу до розгортання дозволяє суттєво підвищити швидкість впровадження змін, зменшити ризик виникнення помилок, спричинених людським фактором, а також забезпечити централізоване управління великою кількістю серверних вузлів. Крім того, використання PHP Deployer дає змогу виконувати комплексну процедуру оновлення системи в межах єдиного сценарію, що включає передачу програмного коду, інсталяцію залежностей, синхронізацію файлової структури та оновлення даних, необхідних для функціонування веб-ресурсу (рис. 2.5).

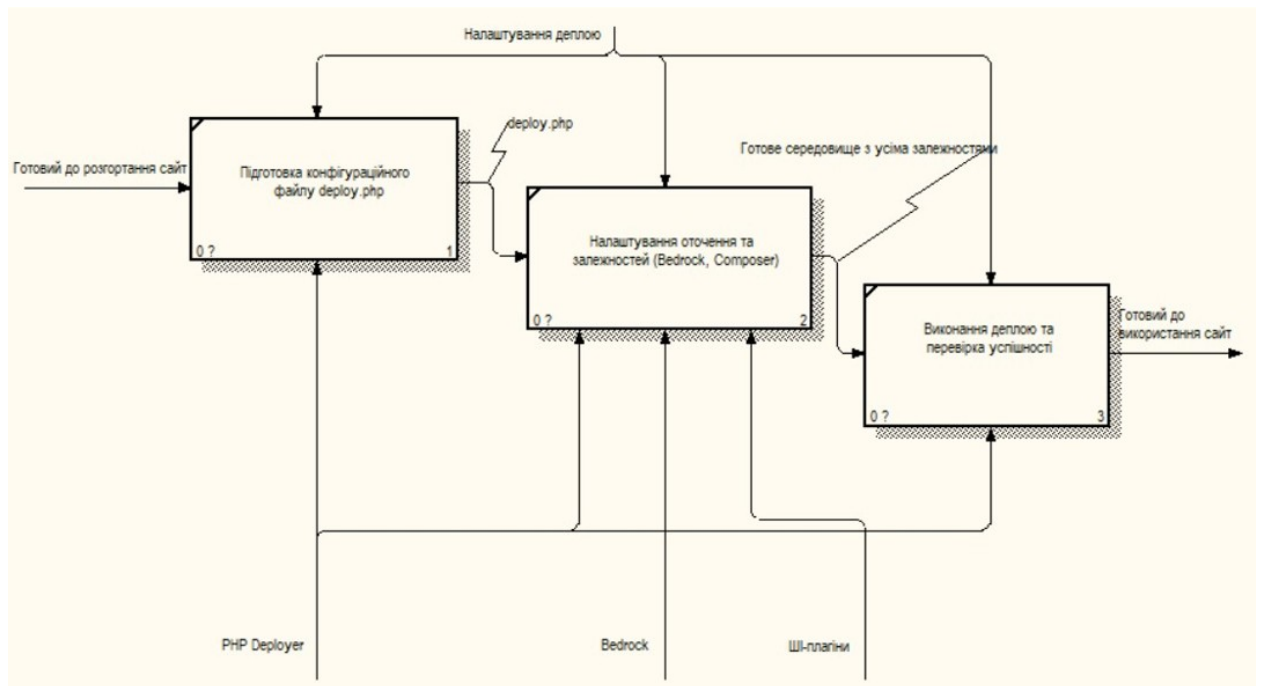


Рис. 2.5. Декомпозиція другого рівня «Автоматизоване розгортання вебсайту»

#### 2.2.4. Автоматизація створення контенту через плагіни з використанням ШІ

Сучасні підходи до управління веб-ресурсами передбачають використання інструментів штучного інтелекту для оптимізації підготовки інформаційного наповнення. У межах розробленої системи реалізовано інтеграцію спеціалізованих програмних модулів, які забезпечують автоматизоване створення текстового контенту безпосередньо в середовищі редагування веб-сайту.

Функціональні можливості такого рішення охоплюють автоматичне формування текстових матеріалів відповідно до визначеної тематики, цільової аудиторії та набору ключових параметрів. Крім генерації змістового наповнення, система виконує структурування матеріалу шляхом автоматичного створення заголовків, підзаголовків та логічних блоків тексту. Додатково реалізовано механізми лінгвістичного аналізу, які забезпечують перевірку стилістичної узгодженості тексту та його адаптацію до вимог пошукової оптимізації.

Використання інтелектуальних алгоритмів істотно скорочує часові витрати на підготовку матеріалів для веб-ресурсів, підвищує продуктивність контент-менеджерів та забезпечує відповідність створених публікацій сучасним стандартам цифрового контенту. Зокрема, інтеграція програмного рішення AI Engine надає можливість формувати SEO-орієнтовані тексти за допомогою природномовних запитів користувача безпосередньо в редакторі WordPress, що значно спрощує процес наповнення сайту актуальними матеріалами (рис. 2.6).

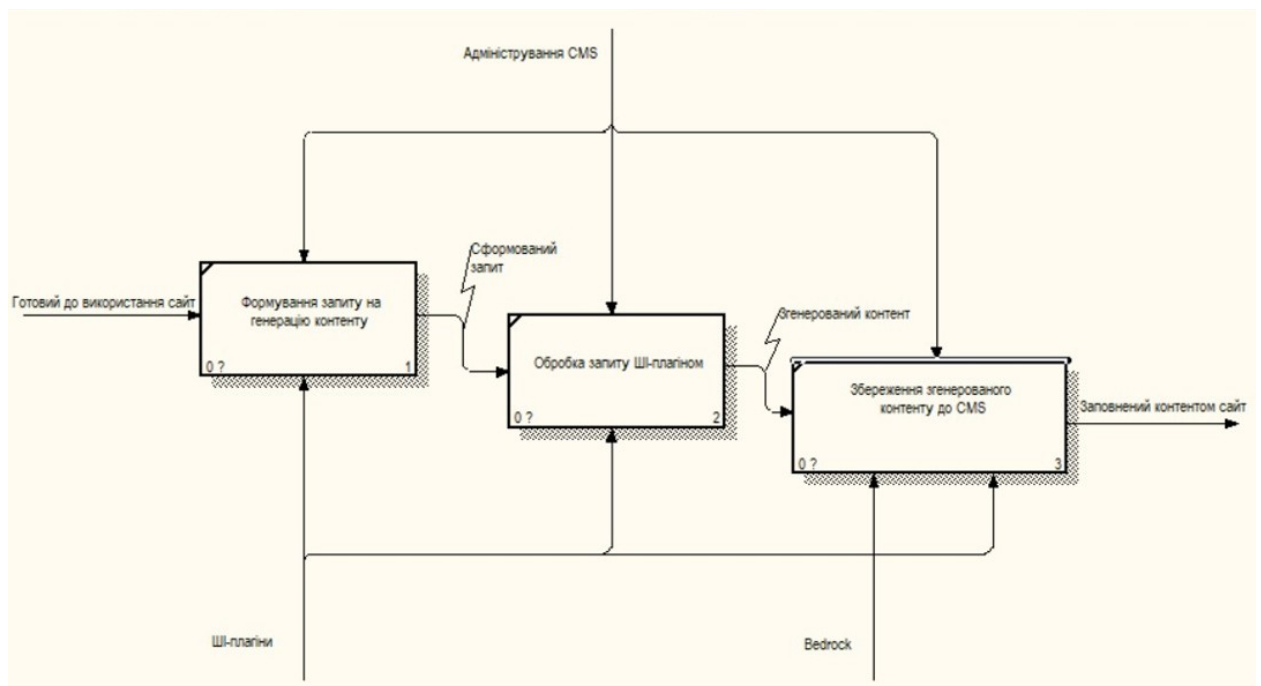


Рис. 2.6. Декомпозиція другого рівня «Генерація контенту на базі ШІ»

Отже, поєднання автоматизованого механізму розгортання веб-проектів із технологіями інтелектуальної генерації контенту формує єдине інтегроване середовище підтримки життєвого циклу веб-сайту. Запропонований підхід сприяє підвищенню рівня автоматизації процесів, скороченню експлуатаційних витрат, покращенню якості контенту та забезпеченню масштабованості інформаційної системи в умовах зростання вимог до сучасних веб-технологій.

## **2.3. Побудова ієрархії процесів**

Одним із важливих етапів системного проектування є побудова ієрархії процесів, яка відображає логічну структуру функціонування інформаційної системи та встановлює взаємозв'язки між окремими функціональними компонентами. Такий підхід дозволяє деталізувати послідовність виконання операцій, визначити роль кожного процесу в загальному життєвому циклі системи та забезпечити ефективне управління її функціонуванням. У межах розробленої інформаційної системи особливу увагу приділено двом ключовим напрямкам автоматизації: розгортанню веб-ресурсів та створенню інформаційного наповнення із застосуванням технологій штучного інтелекту.

### **2.3.1. Ієрархія процесів автоматизації розгортання сайтів**

Процес впровадження веб-проєкту у робоче середовище реалізується як послідовність взаємопов'язаних операцій, що забезпечують автоматичне перенесення програмного продукту на серверну інфраструктуру.

1. Першим етапом є створення та початкова конфігурація проєкту, під час якої формується базова структура застосунку.
2. Наступним кроком виступає організація системи контролю версій шляхом налаштування репозиторію та визначення правил роботи з програмним кодом.
3. Особливе значення має підготовка середовища виконання, що передбачає встановлення платформи Bedrock, інтеграцію необхідних програмних пакетів і конфігурування параметрів майбутнього розгортання. На цьому ж етапі створюється файл конфігурації Deployer, де визначаються сервери призначення, доменні імена, параметри доступу та характеристики баз даних.
4. Безпосереднє розгортання включає передачу програмного коду на віддалені сервери, автоматичне встановлення залежностей, виконання службових команд і приведення системи до працездатного стану.

5. Після завершення основних операцій виконуються контрольні процедури, спрямовані на перевірку коректності роботи програмного забезпечення, моніторинг успішності оновлення та забезпечення можливості оперативного повернення до попередньої версії у разі виникнення критичних помилок.

### **2.2.3. Ієрархія процесів автоматичного створення контенту**

Автоматизація підготовки контенту реалізується шляхом інтеграції інструментів штучного інтелекту до системи керування веб-ресурсом. Логіка функціонування цього модуля передбачає кілька послідовних стадій.

1. На початковому етапі здійснюється вибір програмного компонента, який забезпечує взаємодію із мовною моделлю, а також налаштування параметрів генерації відповідно до вимог проєкту.

2. Далі користувач визначає тематичну спрямованість матеріалу, формує перелік ключових слів та додаткових інструкцій для генерації.

3. Після отримання вхідних параметрів система автоматично створює текстовий матеріал із використанням алгоритмів штучного інтелекту. Згенерований контент проходить додатковий етап обробки, який включає структурування тексту, коригування стилістики, перевірку мовних конструкцій та оптимізацію відповідно до вимог пошукових систем.

4. Завершальною стадією є експертна оцінка сформованого матеріалу користувачем, після чого контент може бути опублікований на веб-ресурсі або переданий на подальше доопрацювання.

### **2.3.1. Контекстна діаграма функціонування інформаційної системи**

Для відображення взаємодії інформаційної системи із зовнішнім середовищем побудовано контекстну діаграму, яка демонструє інформаційні потоки між основними учасниками та програмно-технічними компонентами.

До ключових зовнішніх сутностей належать адміністратори та контент-менеджери, які здійснюють управління веб-проєктами, серверна

інфраструктура, що забезпечує розміщення програмного забезпечення, а також сервіси штучного інтелекту, відповідальні за автоматичне формування контенту. Окрему роль відіграють інструменти розробки та розгортання, представлені платформою Bedrock і засобом автоматизації PHP Deployer.

Побудована модель дозволяє візуалізувати логіку взаємодії між усіма складовими системи, визначити їх функціональне призначення та оцінити інформаційні зв'язки між окремими підсистемами.

Узагальнена схема процесів охоплює підготовку проєкту, керування програмними залежностями, автоматизоване розгортання веб-ресурсу, створення контенту на основі штучного інтелекту та подальший моніторинг результатів функціонування системи. Загальну структуру процесів наведено на рис. 2.7.

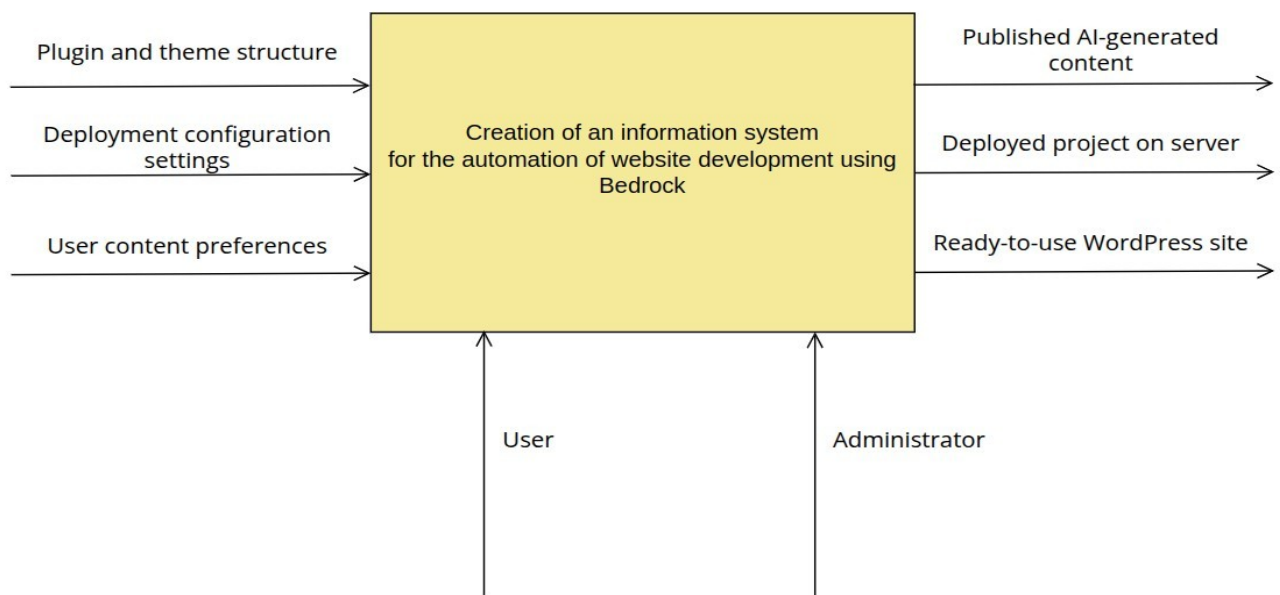


Рис. 2.7. Структура процесів інформаційної системи

Застосування методу аналітичної ієрархії забезпечило можливість обґрунтованого вибору архітектурних рішень та визначення пріоритетних напрямів розвитку інформаційної системи. Використання цього підходу сприяло формуванню програмного рішення, яке поєднує високий рівень

автоматизації, функціональну гнучкість, масштабованість і відповідність актуальним вимогам сучасної веб-розробки.

#### **2.4. Вибір та обґрунтування засобів розв'язання задачі**

Розроблення інформаційної системи потребувало добору програмних засобів, здатних забезпечити ефективну автоматизацію процесів створення, розгортання та супроводу веб-ресурсів. Вибір технологій здійснювався з урахуванням критеріїв надійності, масштабованості, безпеки та сумісності між компонентами системи.

Основою програмної архітектури обрано платформу Bedrock, яка реалізує сучасний підхід до організації WordPress-проектів. Використання цього рішення забезпечує централізоване управління залежностями через Composer, ізоляцію конфігурацій та розмежування робочих середовищ, що позитивно впливає на безпеку й керованість проєкту.

Для автоматизації процесів розгортання використовується PHP Deployer. Інструмент дозволяє стандартизувати процедури оновлення веб-ресурсів, усунути більшість ручних операцій та забезпечити повторюваність усіх етапів деплою. Завдяки підтримці багатосерверних конфігурацій рішення придатне для використання як у невеликих, так і в масштабних веб-проєктах.

Інтелектуальна складова системи реалізована за допомогою плагіна AI Engine, який інтегрує можливості сучасних мовних моделей у середовище WordPress. Це рішення дозволяє автоматизувати створення текстових матеріалів, оптимізувати контент для пошукових систем та прискорити процес інформаційного наповнення веб-ресурсів.

Як основне середовище програмної розробки використовується Visual Studio Code. Популярність цього редактора зумовлена широкими можливостями налаштування, підтримкою сучасних мов програмування та інтеграцією із системами контролю версій і засобами автоматизації.

Зберігання та обробка даних забезпечується системою керування базами даних MySQL, яка характеризується високою продуктивністю, надійністю та широкою підтримкою у веб-середовищі.

## 2.5. Технічні характеристики обраних програмних засобів для розробки

Технологічна основа розробленої інформаційної системи включає низку програмних рішень, кожне з яких виконує окремі функціональні завдання.

**Roots/Bedrock** використовується як спеціалізована платформа для структуризації WordPress-проектів. Система забезпечує централізоване керування залежностями, підвищений рівень безпеки та розділення конфігурацій між середовищами розробки, тестування й експлуатації.

**PHP Deployer 8.0** виконує роль механізму автоматизованого розгортання. Інструмент підтримує сценарне виконання операцій, багатосерверний деплой, контроль версій та механізми відновлення працездатності системи після невдалих оновлень.

**AI Engine** забезпечує інтеграцію сервісів штучного інтелекту з WordPress. Основними можливостями компонента є генерація текстового контенту, його редагування, форматування та адаптація до вимог SEO-оптимізації.

**Visual Studio Code** використовується як універсальне середовище програмування, що підтримує сучасні інструменти розробки, налагодження та контролю якості програмного коду.

**MySQL 8.0** виконує функції сховища даних інформаційної системи, забезпечуючи швидке виконання запитів, підтримку транзакцій та високий рівень цілісності інформації.

Сукупне використання зазначених технологій формує цілісне програмно-технічне середовище, здатне забезпечити ефективне функціонування системи автоматизації створення веб-сайтів, підвищити

продуктивність розробників та гарантувати стабільність роботи веб-ресурсів на всіх етапах їх життєвого циклу. Для візуалізації взаємодії компонентів розроблено архітектурну схему системи (рис. 2.8).

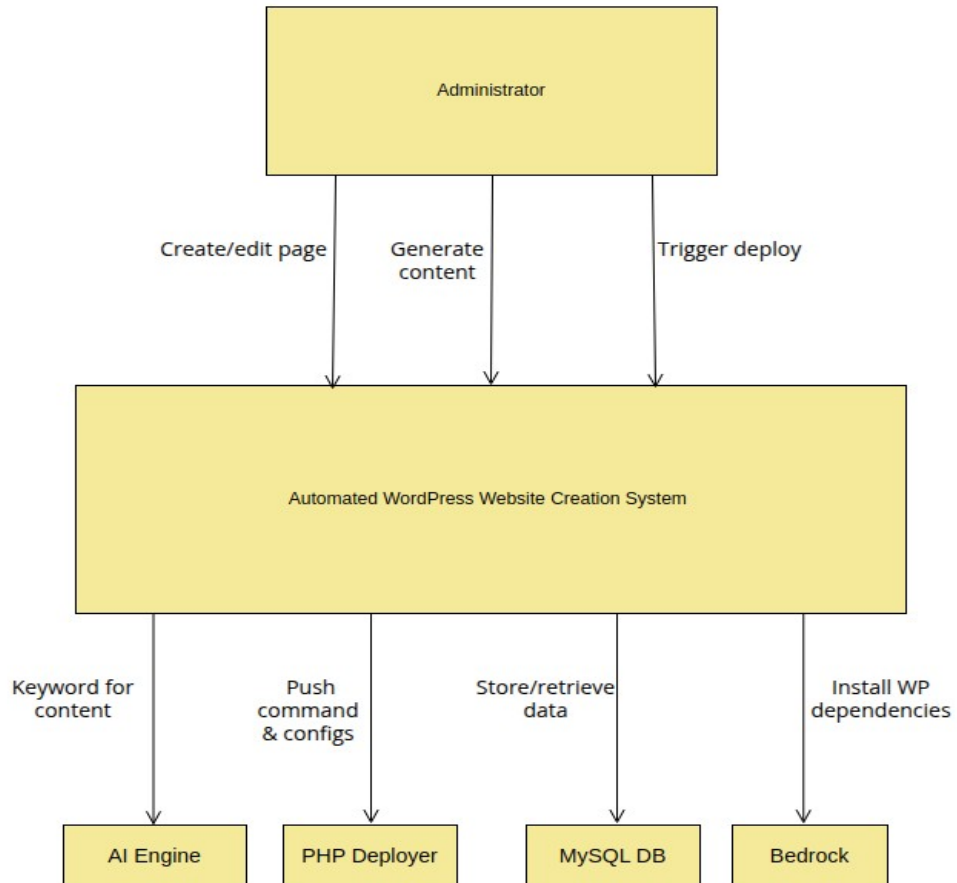


Рис. 2.8. Діаграма архітектури системи

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 3.1. Опис розробки інформаційної системи

Розроблена інформаційна система, орієнтована на автоматизацію процесів створення, супроводу та впровадження веб-ресурсів на платформі WordPress. Вона спрямована на зменшення обсягу рутинних завдань під час налаштування проєктів, розгортання програмного забезпечення та підготовки інформаційного наповнення. Архітектура системи на основі інструментів управління середовищем розробки, засобів автоматизованого розгортання, модулів штучного інтелекту для генерації контенту та механізмів адміністрування користувачів. Реалізація програмного рішення здійснена з використанням технологій PHP, HTML, CSS та JavaScript на базі системи управління контентом WordPress. Структура інформаційної системи включає такі функціональні модулі:

##### **1. Модуль підготовки та керування середовищем розробки (Bedrock):**

**Організація середовища виконання.** Використання Bedrock забезпечує розділення конфігурацій для локального, тестового та продуктивного середовищ, що підвищує рівень безпеки та спрощує адміністрування проєкту.

**Керування програмними залежностями.** Інтеграція з Composer надає можливість централізовано контролювати версії пакетів, бібліотек і розширень, забезпечуючи їх актуальність та сумісність між собою.

##### **2. Модуль автоматизації розгортання вебресурсів (PHP Deployer):**

**Підготовка сценаріїв розгортання.** Формування конфігураційного файлу `deploy.php` дозволяє визначити послідовність операцій, параметри серверів та правила виконання деплою.

**Автоматичне впровадження проєкту.** Система здійснює передачу програмного коду на сервер, виконує оновлення необхідних компонентів та проводить службові процедури підготовки середовища до роботи.

**Моніторинг результатів розгортання.** Після завершення процесу деплою виконуються перевірка працездатності вебресурсу, аналіз можливих помилок та забезпечується механізм повернення до попередньої стабільної версії.

### **3. Модуль інтелектуальної генерації контенту (AI Engine):**

**Взаємодія з редактором Gutenberg.** Інструменти штучного інтелекту інтегровані до стандартного середовища створення контенту WordPress, що забезпечує зручну роботу користувача.

**Автоматизоване формування текстових матеріалів.** Генерація контенту здійснюється на основі заданої тематики, ключових слів та інших параметрів із використанням сучасних мовних моделей.

**Підготовка матеріалів до публікації.** Передбачено засоби редагування, попереднього перегляду та подальшого розміщення створеного контенту на вебсайті.

### **4. Модуль адміністрування користувачів:**

**Керування обліковими записами.** Реалізовано стандартні механізми створення користувачів, автентифікації та доступу до функціональних можливостей системи.

**Розмежування повноважень.** Система підтримує рольову модель доступу, що дозволяє призначати права відповідно до функціональних обов'язків користувачів, зокрема адміністраторів, редакторів та авторів.

### **5. Додаткові функціональні компоненти:**

**Інформаційна панель системи.** Головна сторінка забезпечує відображення ключових відомостей про проєкт, його функціональні можливості та поточний стан роботи.

**Підсистема керування файлами.** Передбачено механізми завантаження, зберігання та обробки мультимедійних ресурсів, включаючи

графічні матеріали та інші об'єкти, що використовуються під час наповнення вебресурсу контентом.

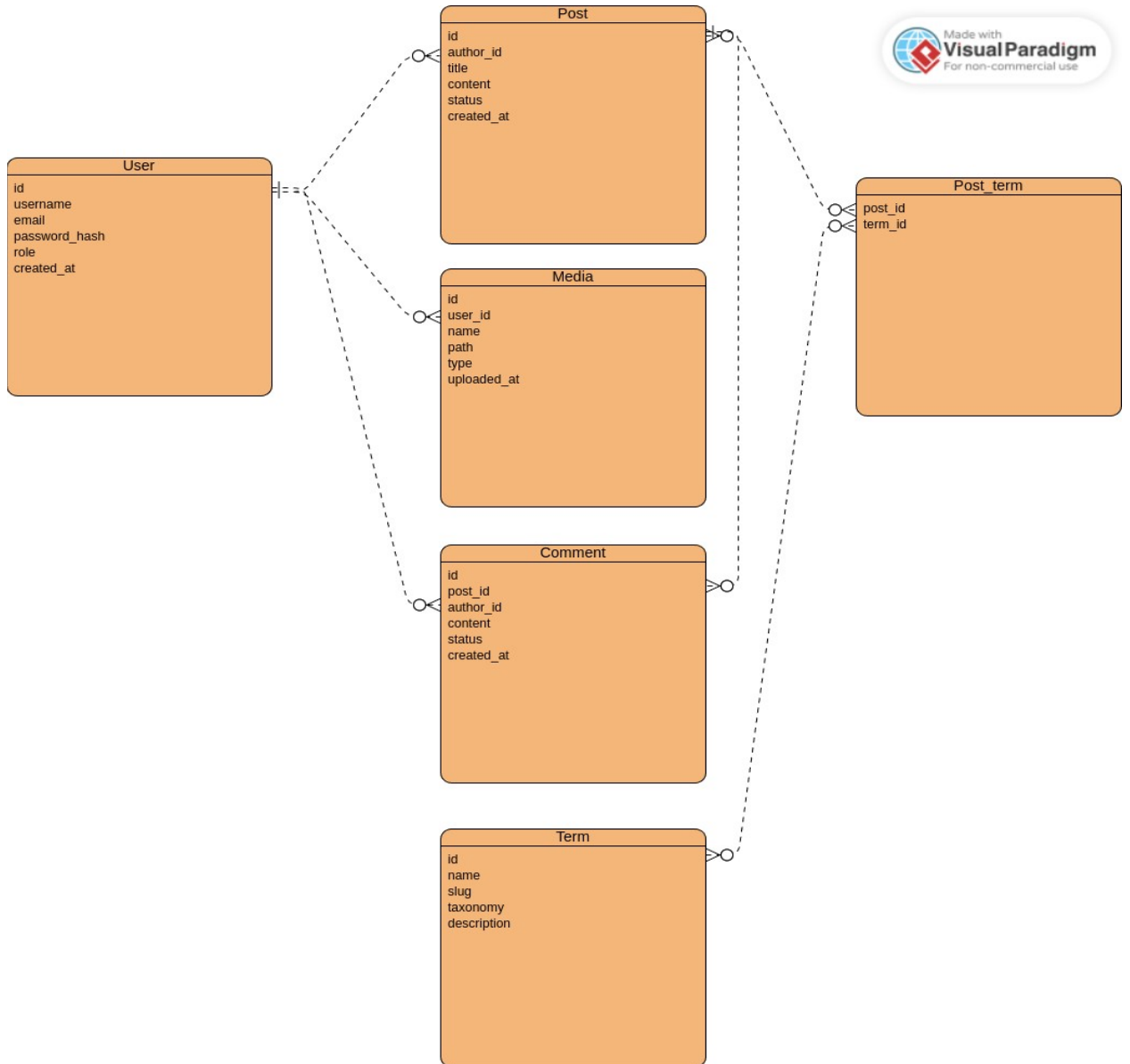


Рис. 3.1. ER-діаграма бази даних

### 3.2. Інструкція користувача

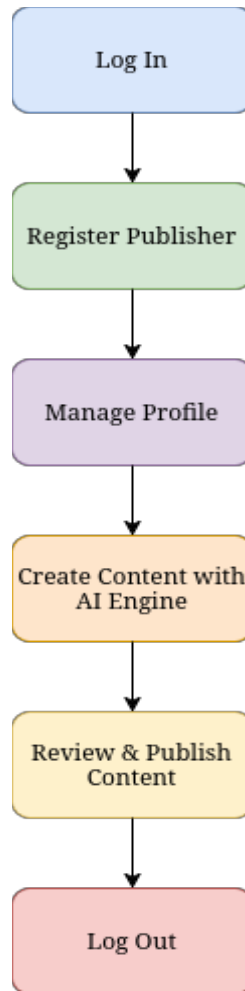


Рис. 3.2. Діаграма користувацького процесу

#### 1. Авторизація в адміністративному інтерфейсі WordPress

Для отримання доступу до функціональних можливостей системи необхідно відкрити сторінку адміністрування за адресою:

**ваш\_домен/wp/wp-admin**

Після завантаження форми входу користувач повинен вказати облікові дані, надані адміністратором системи, а саме ім'я користувача та пароль. У разі успішної автентифікації відбудеться перенаправлення до адміністративної панелі WordPress (рис. 3.3).

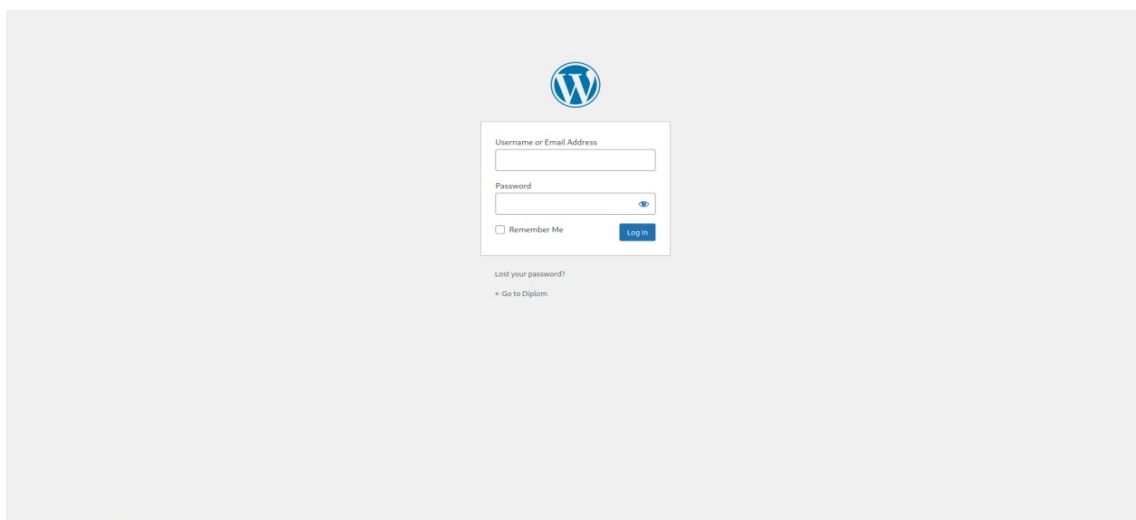


Рис. 3.3. Сторінка входу до адміністративної панелі WordPress

## 2. Створення нового облікового запису користувача (Publisher/Editor)

Для надання доступу новому користувачеві необхідно перейти до розділу «Users» та «Add New».

У відкритій формі вказати основні параметри облікового запису: ім'я користувача, адресу електронної пошти, пароль та роль. Для осіб, які будуть займатися створенням або редагуванням публікацій, доцільно призначити роль Editor або Author. Після заповнення всіх полів необхідно підтвердити створення запису натисканням кнопки «Add New User» (рис. 3.4).

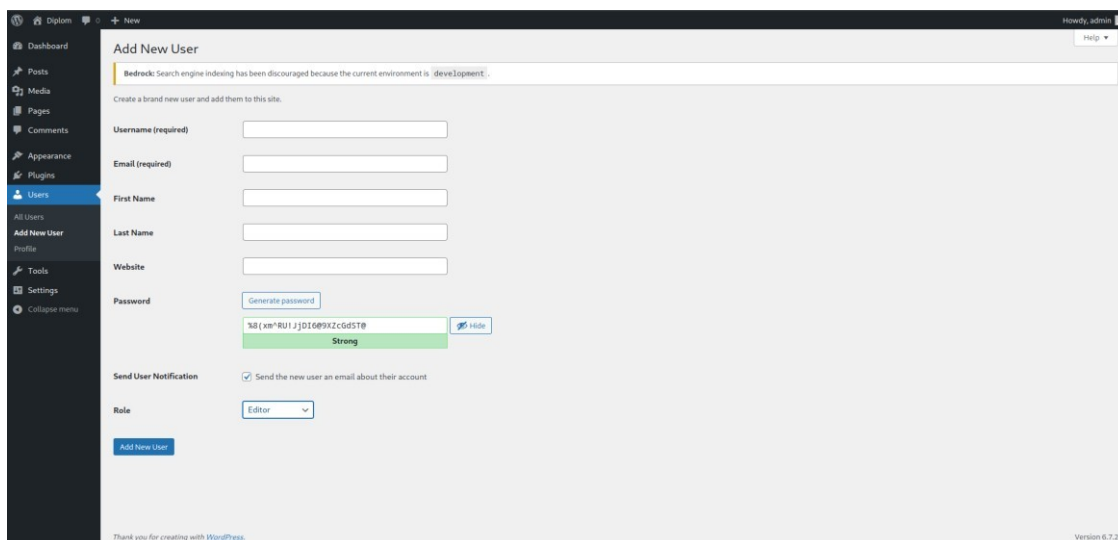


Рис. 3.4. Створення нового користувача в адміністративній панелі

### 3 Вхід користувача з редакторськими правами

Після створення нового облікового запису користувач може авторизуватися в системі за стандартною адресою адміністративної панелі:

**ваш\_домен/wp/wp-admin**

Для входу використовуються індивідуальні облікові дані, визначені під час реєстрації. Після успішної авторизації користувач отримує доступ до функціоналу відповідно до призначеної ролі (рис. 3.5).

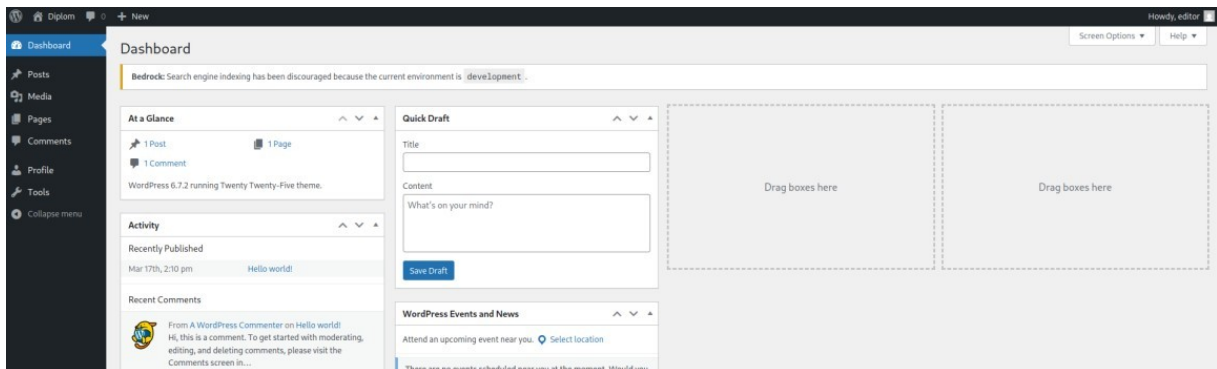


Рис. 3.5. Панель WordPress для користувача з роллю Editor

### 4. Інсталяція та конфігурація AI Engine

Для інтеграції інструментів штучного інтелекту до проєкту необхідно виконати встановлення плагіна AI Engine за допомогою менеджера залежностей Composer. У кореневій директорії проєкту виконується команда:

```
composer require wpackagist-plugin/ai-engine
```

```
~/PhpstormProjects/diplom.test/bedrock
composer require wpackagist-plugin/ai-engine
PHP Warning: PHP Startup: Unable to load dynamic library 'gd' (tried: /usr/lib/php/modules/(/usr/lib/php/modules/gd.so: cannot open shared object file: No such file or directory))
./composer.json has been updated
Running composer update wpackagist-plugin/ai-engine
Loading composer repositories with package information
Updating dependencies
Nothing to modify in lock file
Writing lock file
Installing dependencies from lock file (including require-dev)
Nothing to install, update or remove
Generating optimized autoload files
14 packages you are using are looking for funding.
Use the 'composer fund' command to find out more!
No security vulnerability advisories found.
Using version 3.5.3 for wpackagist-plugin/ai-engine
```

Рис. 3.6. Встановлення плагіна AI Engine

Після завершення інсталяції плагін необхідно активувати через розділ «Plugins» → «Installed Plugins», обравши команду «Activate» навпроти AI Engine (рис. 3.6–3.7).

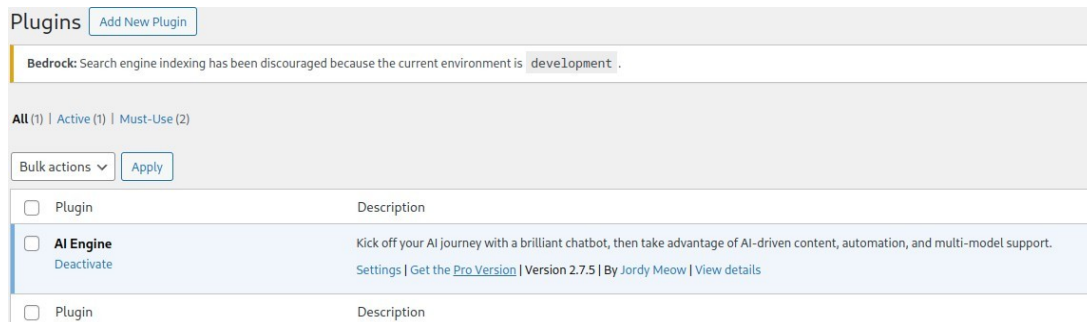


Рис. 3.7. Активація плагіна AI Engine

Подальше налаштування здійснюється через меню «Meow Apps» та «AI Engine». Для забезпечення взаємодії з моделями штучного інтелекту необхідно ввести API-ключ OpenAI у відповідне поле та зберегти внесені зміни (рис. 3.8).

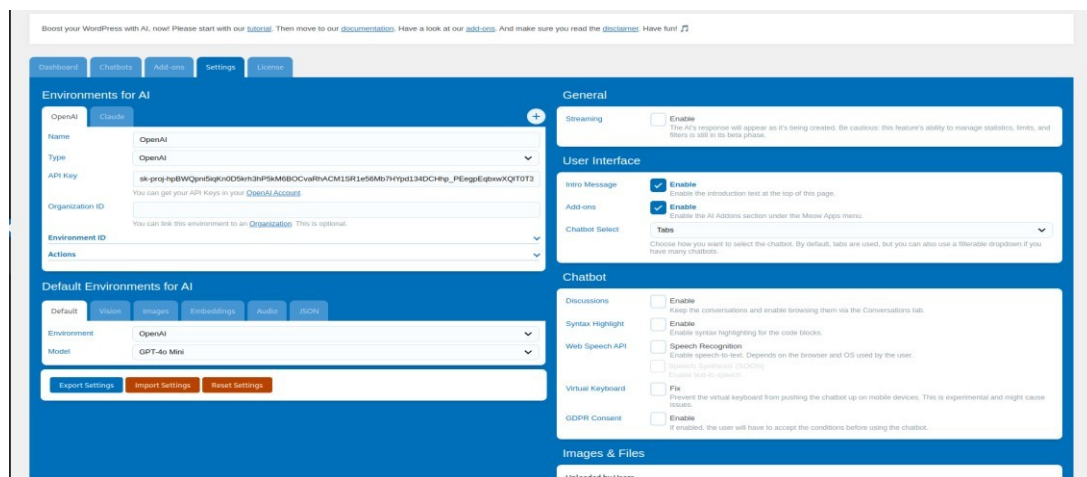


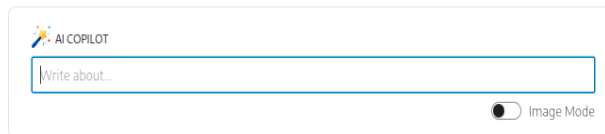
Рис. 3.8. Вставка та збереження API-ключа OpenAI

## 5. Створення та публікація контенту за допомогою AI Engine:

Для створення нового посту потрібно обрати пункт меню «Posts» та «Add New» в адміністративній панелі. Використовувати блоковий редактор Gutenberg для роботи з контентом.

Щоб автоматично згенерувати контент за допомогою AI Engine, потрібно просто поставити пробіл у новому блоці, почати вводити запит для ШІ (наприклад, опис теми або ключові слова) та натиснути клавішу **Enter**. Плагін миттєво створить текст відповідно до вашого запиту. Створений контент можна додатково редагувати.

Add title



The image shows a user interface for AI Copilot. At the top, there is a text input field with the placeholder text "Write about...". To the right of the input field is a toggle switch labeled "Image Mode", which is currently turned off. The AI Copilot logo is visible in the top left corner of the interface.

Рис. 3.9. Процес генерації контенту за допомогою AI Engine

Після завершення роботи над матеріалом його необхідно опублікувати за допомогою кнопки «**Publish**», після чого запис стане доступним на вебсайті (рис. 3.10).



Рис. 3.10. Опублікований пост на сайті

## 6. Керування існуючими публікаціями

Для перегляду та редагування раніше створених записів необхідно перейти до розділу «Posts» та «All Posts».

Обравши потрібний матеріал, користувач може відкрити його для редагування, внести необхідні зміни та зберегти оновлену версію за допомогою кнопки «Update».

У разі необхідності видалення запису використовується функція «Trash», що переносить публікацію до кошика (рис. 3.11).

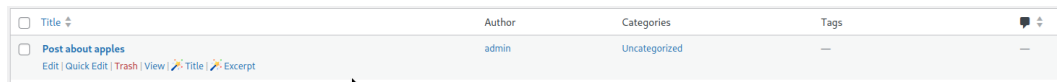


Рис. 3.11. Управління постами (редагування та видалення)

## 7. Завантаження та управління медіафайлами:

Для додавання графічних матеріалів та інших медіафайлів використовується розділ «Media» та «Add New».

Користувач може завантажити файли зі свого комп'ютера, після чого вони автоматично зберігаються в медіабібліотеці WordPress і можуть бути використані під час створення сторінок або публікацій (рис. 3.12).

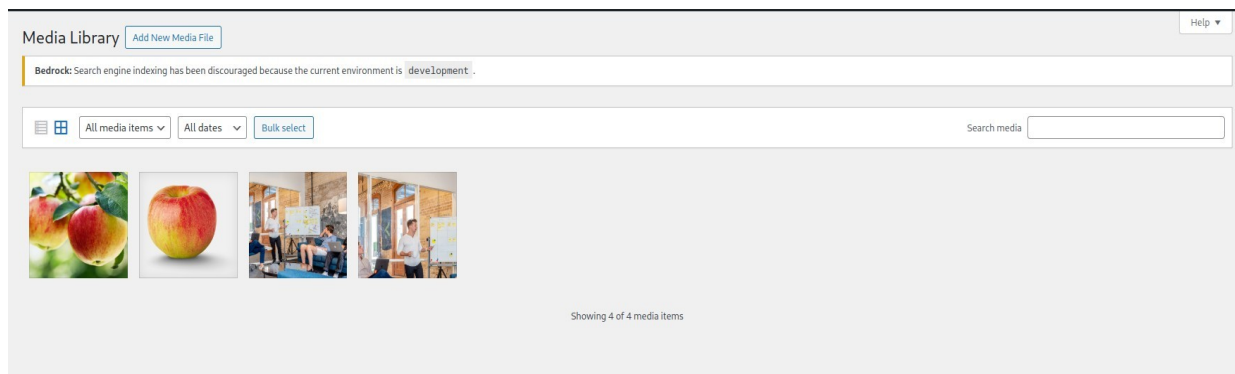


Рис. 3.12. Завантаження та управління медіафайлами

## 8. Розгортання проєкту на сервері (dep deploy):

Команда **dep deploy** забезпечує повний цикл впровадження веб-проєкту у серверне середовище без необхідності ручного виконання окремих операцій.

Під час виконання процедури автоматично здійснюються такі дії:

- створення та підготовка службових директорій на сервері;
- отримання актуальної версії коду з репозиторію Git;

- налаштування спільних ресурсів, зокрема директорії uploads та конфігураційного файлу .env;
- інсталяція залежностей Bedrock і Sage;
- перенесення згенерованих CSS та JavaScript-файлів;
- очищення тимчасових і непотрібних даних;
- активація нового релізу;
- видалення застарілих версій із збереженням останніх двох релізів;
- відображення повідомлення про успішне завершення процедури.

```

Terminal Local x Local (2) x + v
~/PhpstormProjects/diplon.test/bedrock git:[main]
vendor/bin/dep deploy
task deploy:info
# oyer@188.245.201.118's password: #
[188.245.201.118] info deploying bedrock-blog to 188.245.201.118 (release 59)
task deploy:setup
task deploy:lock
task deploy:release
task deploy:update_code
task deploy:env
task deploy:shared
task deploy:writable
task deploy:update_code
task deploy:shared
task bedrock:vendors
task sage:vendors
task sage:compile
task sage:clean_assets
task sage:upload_assets_only
[188.245.201.118] 11/11 [=====] 100% < 1 sec/< 1 sec
task deploy:clean_paths
task deploy:symlink
task deploy:unlock
task deploy:cleanup
task deploy:success
[188.245.201.118] info successfully deployed!

```

Рис. 3.13. Виконання команди dep deploy у терміналі

## 9. Передача бази даних на сервер (dep push)

Для перенесення бази даних використовується команда dep push.

Під час її виконання реалізуються такі операції:

- формування резервної копії локальної бази даних;
- передавання SQL-файлу на сервер;
- імпорт даних у віддалену базу;
- видалення тимчасових SQL-файлів після завершення процедури.

Такий механізм значно спрощує перенесення змін між середовищами розробки та експлуатації.

```
~/PhpstormProjects/diplom.test/bedrock git:[main]
vendor/bin/dep push:db
task push:db
[188.245.201.118] 📦 Exporting local DB...
[188.245.201.118] 📤 Uploading SQL dump to remote server...
  📄 oyer@188.245.201.118's password: 📄
[188.245.201.118] 1/1 [=====] 100% 4 secs/4 secs
[188.245.201.118] 📥 Importing SQL dump on remote server...

[188.245.201.118] ✅ Removing local SQL dump...
[188.245.201.118] ✅ Removing remote SQL dump...
[188.245.201.118] ✅ Database successfully pushed to remote.
```

Рис. 3.14. Приклад виконання команди `dep push:db` у терміналі

## 10. Синхронізація медіаконтенту із сервером (`dep push`)

Команда `dep push` використовується для автоматичного оновлення мультимедійних ресурсів на сервері.

У процесі виконання здійснюється:

- порівняння локальної та віддаленої директорій `uploads`;
- копіювання нових файлів на сервер;
- оновлення змінених медіа-ресурсів без повторного завантаження незмінних даних.

Завдяки цьому забезпечується актуальність усіх мультимедійних матеріалів, що використовуються веб-ресурсом.

```
~/PhpstormProjects/diplom.test/bedrock git:[main]
vendor/bin/dep push:files
task backup:remote_files
task push:files-no-bak
```

Рис. 3.15. Виконання команди `dep push:files` у терміналі

### 3.3. Аналіз розробленої інформаційної системи

Розроблена інформаційна система являє собою комплексне програмне рішення, призначене для оптимізації основних етапів життєвого циклу веб-проектів на платформі WordPress. Система об'єднує засоби керування середовищем розробки, інструменти автоматизованого розгортання та механізми генерації контенту на основі технологій штучного інтелекту, що дозволяє значно підвищити ефективність виконання типових завдань веб-розробки.

Архітектура рішення базується на використанні платформи Bedrock для впорядкованого керування конфігураціями та програмними залежностями, інструменту PHP Deployer для автоматизації процесів розгортання, а також плагіна AI Engine, який забезпечує інтеграцію сучасних мовних моделей у середовище WordPress. Такий підхід дозволяє сформувати єдиний простір для створення, підтримки та розвитку веб-ресурсів.

Особливістю системи є підтримка багаторівневої моделі доступу, що дає змогу розподіляти функціональні обов'язки між різними категоріями користувачів. Адміністратори отримують інструменти для повного керування ресурсом, тоді як редактори та автори можуть здійснювати підготовку, редагування й публікацію матеріалів відповідно до наданих повноважень.

Використання технологій штучного інтелекту суттєво розширює можливості системи. Завдяки інтеграції AI Engine користувачі можуть автоматично формувати текстовий контент на основі коротких текстових інструкцій або наборів ключових слів. Це сприяє скороченню часу підготовки інформаційних матеріалів та підвищенню продуктивності роботи контент-менеджерів.

Основні переваги розробленої інформаційної системи:

- використання технологій штучного інтелекту для автоматизації створення контенту без залучення додаткових зовнішніх сервісів;
- автоматизоване розгортання веб-проектів, що забезпечує відтворюваність процесів та зменшує ризик виникнення помилок під час впровадження

змін;

- застосування архітектури Bedrock, яка забезпечує впорядковане керування залежностями, конфігураціями та середовищами виконання;
- реалізація механізмів рольового доступу для ефективного управління користувачами та розмежування їхніх функцій;
- використання платформи WordPress як базового середовища, що спрощує адаптацію системи до існуючих бізнес-процесів;
- підвищення швидкості підготовки та публікації матеріалів завдяки інтеграції засобів автоматичної генерації контенту безпосередньо в редактор Gutenberg.

Проведений аналіз підтвердив, що розроблене програмне рішення забезпечує високий рівень автоматизації процесів веб-розробки, сприяє зниженню витрат часу на технічне обслуговування проєктів та створює передумови для ефективного масштабування веб-ресурсів у майбутньому.

## ВИСНОВКИ

У процесі виконання бакалаврської кваліфікаційної роботи було вирішено актуальне завдання, пов'язане з автоматизацією процесів створення, розгортання та супроводу веб-сайтів на платформі WordPress. Результатом проведеного дослідження стала розробка інформаційної системи, яка поєднує сучасні інструменти керування веб-проєктами, технології автоматизованого деплою та засоби штучного інтелекту для генерації контенту.

У ході дослідження було проаналізовано сучасний стан розвитку засобів автоматизації веб-розробки, розглянуто принципи функціонування платформ керування залежностями, інструментів безперервного розгортання та рішень на основі штучного інтелекту. Проведене порівняння існуючих програмних продуктів дозволило обґрунтувати вибір технологій Bedrock, PHP Deployer та AI Engine для створення запропонованої системи.

На етапі системного аналізу сформовано дерево цілей, побудовано контекстні та функціональні моделі, визначено структуру бізнес-процесів і досліджено можливості їх автоматизації.

У межах реалізації проєкту було налаштовано середовище розробки на базі Bedrock, впроваджено механізми автоматизованого розгортання за допомогою PHP Deployer та інтегровано інструменти генерації контенту на основі штучного інтелекту.

Результати виконаної роботи дозволяють зробити такі висновки:

- запропоновано комплексне рішення для автоматизації ключових процесів веб-розробки на платформі WordPress;
- реалізовано механізми автоматизованого розгортання веб-проєктів, що забезпечують стабільність та повторюваність процедур впровадження програмного забезпечення;

- інтегровано засоби штучного інтелекту для прискорення підготовки контенту та зменшення навантаження на користувачів;
- забезпечено структуроване керування залежностями та конфігураціями проєкту за рахунок використання Bedrock;
- підвищено рівень безпеки, масштабованості та керованості веб-ресурсів;
- створено програмний продукт, придатний для практичного використання веб-студіями, контент-менеджерами та індивідуальними розробниками.

Подальший розвиток системи може бути пов'язаний із впровадженням контейнеризації на базі Docker, інтеграцією засобів безперервної інтеграції та доставки через GitHub Actions, розширенням функціональності модулів штучного інтелекту, зокрема шляхом автоматичної генерації графічного контенту та проведення SEO-аналізу, а також створенням централізованого інтерфейсу для одночасного керування декількома веб-проєктами.

Отже, розроблена інформаційна система демонструє високу практичну цінність і відповідає сучасним вимогам до автоматизації веб-розробки. Її використання дозволяє підвищити продуктивність роботи, скоротити витрати ресурсів на підтримку проєктів та забезпечити якісно новий рівень організації процесів створення і супроводу веб-сайтів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Raatikainen M., Lassenius C., Männistö T. Continuous deployment of web applications: proceedings of the 40th International Conference on Software Engineering (ICSE '18). New York: ACM, 2018. С. 529–538.
2. Zhang J., Mockus A. Measuring deployment frequency in open source projects. Empirical Software Engineering. 2022. Vol. 27, Iss. 5. С. 1–28.
3. Poole R. Modern WordPress development with Bedrock and Composer. Roots.io. 2023. URL: <https://roots.io/bedrock/>
4. Moser F. Bedrock Deployer: GitHub repository. 2022. URL: <https://github.com/florianmoser/bedrock-deployer>
5. Meow Apps. AI Engine – plugin for WordPress content generation with GPT: офіційна документація. 2026. URL: <https://meowapps.com/ai-engine/>
6. WordPress.org. Block Editor Handbook (Gutenberg). 2026. URL: <https://developer.wordpress.org/block-editor/>
7. Composer. Dependency Manager for PHP: офіційна документація. 2026. URL: <https://getcomposer.org/>
8. Deployer.org. PHP Deployment Tool: офіційна документація. 2026. URL: <https://deployer.org/>
9. Tailwind CSS. Documentation and Utility-First Styling Guide. 2026. URL: <https://tailwindcss.com/docs>
10. OpenAI. API Documentation and Usage Guidelines. 2026. URL: <https://platform.openai.com/docs/>
11. W3Techs. WordPress Usage and Market Share Statistics. 2026. URL: <https://w3techs.com/technologies/details/cm-wordpress>
12. Roots. Bedrock: modern WordPress boilerplate. 2026. <https://roots.io/bedrock/>
13. WordPress 7.0 “Armstrong” Release. 2026. URL: <https://wordpress.org/news/2026/05/armstrong/>

14. Deployer. v8.0.5 Release Notes. 2026. URL:  
<https://github.com/deployphp/deployer/releases/tag/v8.0.5>.
15. Jordy Meow. AI Engine Pro Documentation. 2026. URL:  
<https://meowapps.com/products/ai-engine-pro/>
16. Microsoft. Visual Studio Code Documentation. 2026. URL:  
<https://code.visualstudio.com/docs>
17. Oracle. MySQL 8.0 Reference Manual. 2026. URL:  
<https://dev.mysql.com/doc/refman/8.0/en/>
18. Git. Reference Manual. 2026. <https://git-scm.com/doc>
19. Docker Documentation. 2026. URL: <https://docs.docker.com/>
20. Kubernetes Documentation. 2026. URL: <https://kubernetes.io/docs/>.

## **ДОДАТКИ**

**ДОДАТОК А**

## Конфігурація деплоєра deploy.php

```
<?php
    namespace Deployer;
    require 'vendor/deployer/deployer/recipe/common.php';
    require 'vendor/florianmoser/bedrock-deployer/recipe/bedrock_db.php';
    require 'vendor/florianmoser/bedrock-deployer/recipe/bedrock_env.php';
    require 'vendor/florianmoser/bedrock-deployer/recipe/bedrock_misc.php';
    require 'vendor/florianmoser/bedrock-deployer/recipe/common.php'; require
    'vendor/florianmoser/bedrock-deployer/recipe/filetransfer.php'; require
    'vendor/florianmoser/bedrock-deployer/recipe/sage.php';
    require 'vendor/florianmoser/bedrock-deployer/recipe/trellis.php';
    // === CONFIGURATION ===
    // Git repository
    set('repository', 'https://github.com/SashaSolovey1/bedrock-blog.git');
    set('keep_releases', 2);
    // Shared directories and files set('shared_dirs', [ 'web/app/uploads',
    'web/app/cache',
    ]);
    set('shared_files', ['.env']);
    // Writable dirs
    set('writable_dirs', [
    'web/app/cache',
    'web/app/cache/acorn',
    'web/app/cache/acorn/framework/views',
    ]);
    set('writable_use_sudo', true);
    set('writable_mode', 'chmod');
```

```
// Local root
set('local_root',_DIR_);
// Sage theme path
set('theme_path', 'web/app/themes/test');
// Directories to sync (e.g. uploads)
set('sync_dirs', [
  _DIR_.'/web/app/uploads/' => '{{deploy_path}}/shared/web/app/uploads/',
]);
// Default stage
set('default_stage', 'production');
// === HOST ===
host('188.245.201.118')
->set('remote_user', 'deployer')
->set('deploy_path', '/var/www/html/test')
->set('branch', 'main')
->set('stage', 'production');
// === TASKS ===
desc('Deploy your project');
task('deploy', [
  'deploy:prepare',
  'deploy:update_code',
  'deploy:shared',
  'bedrock:vendors',
  'sage:vendors',
  'push:assets',
  'deploy:symlink',
  'deploy:unlock',
  'deploy:cleanup',
  'deploy:success',
]);
```

```
// [Optional] if deploy fails automatically unlock.
after('deploy:failed', 'deploy:unlock');

// === CUSTOM TASK: Push local DB to remote ===
task('push:db', function () {
  $timestamp = date('Ymd_His');
  $localDump = "_db_export_{$timestamp}.sql";
  $remoteDump = "{{deploy_path}}/shared/{$localDump}";
  $localPath = DIR_;
  $remotePath = '{{current_path}}';
  writeln("<info> Exporting local DB...</info>");
  runLocally("cd {$localPath} && wp db export {$localDump}");
  writeln("<info>   Uploading SQL dump to remote server...</info>");
  upload($localDump, $remoteDump);
  writeln("<info>   Importing SQL dump on remote server...</info>");
  run("cd {$remotePath} && wp db import {$remoteDump}");
  writeln("<info>   Removing local SQL dump...</info>");
  runLocally("rm {$localDump}");
  writeln("<info>   Removing remote SQL dump...</info>");
  run("rm {$remoteDump}");
  writeln("<info>   Database successfully pushed to remote.</info>");
});
```

**ДОДАТОК Б**

Залежності composer.json

```
{
  "name": "roots/bedrock", "type": "project",
  "license": "MIT",
  "description": "WordPress boilerplate with Composer, easier configuration,
  and an improved folder structure",
  "homepage": "https://roots.io/bedrock/",
  "authors": [
    {
      "name": "Scott Walkinshaw",
      "email": "scott.walkinshaw@gmail.com",
      "homepage": "https://github.com/swalkinshaw"
    },
    {
      "name": "Ben Word",
      "email": "ben@benword.com",
      "homepage": "https://github.com/retlehs"
    }
  ],
  "keywords": [
    "bedrock", "composer", "roots", "wordpress", "wp", "wp-config"
  ],
  "support": {
    "issues": "https://github.com/roots/bedrock/issues",
    "forum": "https://discourse.roots.io/category/bedrock"
  },
  "repositories": [
    {
```

```

"type": "composer",
"url": "https://wpackagist.org",
"only": ["wpackagist-plugin/*", "wpackagist-theme/*"]
}
],
"require": {
"php": ">=8.4",
"composer/installers": "^2.2",
"vlucas/phpdotenv": "^5.5",
"oscarotero/env": "^2.1",
"roots/bedrock-autoloader": "^1.0",
"roots/bedrock-disallow-indexing": "^2.0",
"roots/wordpress": "6.8.0",
"roots/wp-config": "1.0.0",
"roots/wp-password-bcrypt": "1.2.0",
"wpackagist-theme/twentytwentyfive": "^1.0",
"wpackagist-plugin/ai-engine": "^3.5.3",
"roots/acorn": "^5.0",
"log1x/acf-composer": "^3.4",
"wordpress-premium/advanced-custom-fields-pro": "^6.3",
"wpackagist-plugin/contact-form-7": "^6.0"
},
"require-dev": {
"roave/security-advisories": "dev-latest",
"laravel/pint": "^1.18",
"deployer/deployer": "^8.0",
"florianmoser/bedrock-deployer": "^1.24"
},
"config": {
"optimize-autoloader": true,

```

```
"preferred-install": "dist",
"allow-plugins": {
"composer/installers": true,
"roots/wordpress-core-installer": true
}
},
"minimum-stability": "dev",
"prefer-stable": true,
"extra": {
"installer-paths": {
"web/app/mu-plugins/{$name}/*": ["type:wordpress-muplugin"],
"web/app/plugins/{$name}/*": ["type:wordpress-plugin"],
"web/app/themes/{$name}/*": ["type:wordpress-theme"]
},
"wordpress-install-dir": "web/wp"
},
"scripts": {
"lint": "pint --test",
"lint:fix": "pint"
}
}
```