

Перший європейський університет
ННІ «Європейська школа бізнесу»
Кафедра інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ

Доцент кафедри інформаційних технологій ННІ «Європейська школа
бізнесу», кандидат технічних наук

П. П. Яцук

«___» _____ 2026 р.

КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

зі спеціальності 121 «Інженерія програмного забезпечення»

**ТЕМА: «Розробка системи управління контентом (CMS) для військової
сфери»**

Виконавець: Савастюк Андрій Андрійович

(ПБ)

Науковий керівник: Яцук П. П., кандидат технічних наук, доцент.

(ПБ)

Консультанти з окремих розділів пояснювальної записки:

(ПБ)

(ПБ)

Нормоконтролер:

(ПБ)

Перший європейський університет
ННІ «Європейська школа бізнесу»
Кафедра інформаційних технологій

Ступінь вищої освіти – перший (бакалаврський) рівень

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Інженерія програмного забезпечення»

Освітньо-кваліфікаційний рівень: бакалавр

ЗАТВЕРДЖУЮ:

Доцент кафедри інформаційних
технологій ННІ «Європейська школа
бізнесу», кандидат технічних наук

П. П. Яцук

« ____ » _____ 2026 р.

ЗАВДАННЯ

НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ БАКАЛАВРСЬКОЇ РОБОТИ

Савастюк Андрій Андрійович

1. Тема:

**«Розробка системи управління контентом (CMS) для військової сфери»,
затверджена наказом ректора від «24» грудня 2025 р. № 207-к.**

2. Термін виконання роботи:

з «26» травня 2026 р. по «22» червня 2026 р

3. Дата подання роботи на випускню кафедру:

«12» червня 2026 р.

4. Вихідні дані роботи:

Методології об'єктно-орієнтованого та веборієнтованого програмування, мова програмування PHP, технології HTML, CSS, JavaScript, система керування базами даних SQLite, принципи проектування систем управління контентом (CMS), UML-діаграми, архітектурні підходи до розробки програмного забезпечення, технології створення користувацького інтерфейсу, методи забезпечення інформаційної безпеки, засоби тестування програмного забезпечення, сучасні підходи до організації та управління інформаційним контентом у військовій сфері.

5. Зміст пояснювальної записки:

Вступ

- Розділ 1. Аналіз предметної області та постановка задачі
- Розділ 2. Проектування та архітектура системи управління контентом для військової сфери
- Розділ 3. Реалізація та тестування системи управління контентом для військової сфери

Висновки

Список використаних джерел

Додатки

6. Перелік обов'язкового графічного матеріалу:

Схема архітектури системи Military CMS.

ER-діаграма бази даних системи.

Схема структури бази даних.

Матриця прав доступу користувачів.

Схема ролей та взаємодії користувачів із системою.

Макети користувацького інтерфейсу.

Скріншот сторінки авторизації.

Скріншот головної панелі керування (Dashboard).

Скріншоти модуля керування матеріалами.

Скріншоти модуля керування категоріями.

Скріншоти модуля керування користувачами.

Скріншот журналу подій.

Фрагменти SQL-коду створення структури бази даних.

Фрагменти PHP-коду реалізації основних модулів системи.

Результати тестування програмного засобу.

7. Календарний план - графік виконання КБР

№ з/п	Завдання	Термін виконання	Відмітка про виконання
1.	Складання і затвердження індивідуальних завдань на виконання кваліфікаційної бакалаврської роботи (КБР)	28.02.26	
2.	Підготовка вступу і розділу 1 КБР	17.03.26	
3.	Підготовка розділу 2 КБР	10.04.26	
4.	Підготовка розділу 3 КБР, висновків і переліку використаних джерел	10.05.26	
5.	Подання студентом завершальної КБР науковому керівнику для перевірки на плагіат та оформлення відгуку	09.06.26	
6.	Попередній розгляд КБР на комісії від кафедри	12-13.06.26	
7.	Доопрацювання роботи, прийняття кафедрою рішення про допуск роботи до захисту в ЕК, оформлення та зовнішнє рецензування	16-20.06.26	
8.	Захист кваліфікаційної бакалаврської роботи в ЕК і присвоєння випускникам кваліфікації	23-24.06.26	

Студент _____

(підпис, ПІБ)

Керівник _____

(підпис, ПІБ)

РЕФЕРАТ

Тема: «Розробка системи управління контентом (CMS) для військової сфери»

Кваліфікаційна бакалаврська робота зі спеціальності 121 «Інженерія програмного забезпечення», ОПП «Інженерія програмного забезпечення», Перший європейський університет, 2026 р.

Загальний обсяг роботи становить **82** сторінок. **Робота складається** зі вступу, трьох розділів, висновків, списку використаних джерел та додатків; містить таблиці, UML-діаграми, лістинги, ілюстрації та схеми.

Ключові слова: CMS, система управління контентом, військова сфера, вебзастосунок, PHP, SQLite, UML, інформаційна система, база даних, автоматизація.

Актуальність теми роботи зумовлена необхідністю ефективного управління інформаційними ресурсами у військовій сфері. В умовах постійного зростання обсягів цифрової інформації особливого значення набуває використання програмних засобів, що забезпечують централізоване зберігання, редагування, структурування та публікацію контенту. Системи управління контентом дозволяють оптимізувати роботу з інформаційними матеріалами, забезпечити контроль доступу до даних та підвищити ефективність інформаційної взаємодії між користувачами. Розробка спеціалізованої CMS для військової сфери сприяє впорядкуванню інформаційних процесів та забезпечує зручні механізми адміністрування контенту.

Метою роботи є розробка системи управління контентом (CMS) для військової сфери, що забезпечує централізоване керування інформаційними матеріалами, розмежування прав доступу користувачів та ефективне адміністрування контенту. Для досягнення поставленої мети у роботі вирішено такі завдання:

1. дослідити особливості систем управління контентом та принципи їх функціонування;
2. проаналізувати сучасні програмні рішення у сфері управління контентом;
3. дослідити специфіку використання інформаційних систем у військовій сфері;
4. сформулювати функціональні та нефункціональні вимоги до програмного засобу;
5. спроектувати архітектуру системи та виконати UML-моделювання;
6. розробити базу даних для зберігання користувачів, категорій та інформаційних матеріалів;
7. реалізувати програмний засіб та провести тестування його функціональних можливостей.

Об'єктом дослідження є процес управління інформаційним контентом у військових організаціях.

Предметом дослідження є методи, моделі та програмні засоби реалізації систем управління контентом для військової сфери.

Практичне значення одержаних результатів полягає у створенні програмного засобу, який може бути використаний для централізованого керування інформаційними матеріалами у військових організаціях. Розроблена система забезпечує можливість створення, редагування, зберігання та публікації контенту, підтримує розмежування прав доступу користувачів та демонструє практичне застосування сучасних підходів до розробки веборієнтованих інформаційних систем.

ABSTRACT

Topic: “Development of a Content Management System (CMS) for the Military Sector”

Qualification bachelor thesis in specialty 121 “Software Engineering”, Educational and Professional Program “Software Engineering”, First European University, 2026.

The total volume of the work is **82** pages. **The thesis consists** of an introduction, three chapters, conclusions, a list of references and appendices; contains tables, UML diagrams, listings, illustrations and schemes.

Keywords: CMS, content management system, military sector, web application, PHP, SQLite, UML, information system, database, automation.

Relevance of the research topic:

The relevance of the work is determined by the need for effective management of information resources in the military sector. Due to the continuous growth of digital information volumes, the use of software tools that provide centralized storage, editing, structuring and publication of content becomes increasingly important. Content management systems make it possible to optimize information management processes, ensure controlled access to data and improve information interaction between users. The development of a specialized CMS for the military sector contributes to the organization of information processes and provides convenient content administration mechanisms.

The purpose of the work is to develop a Content Management System (CMS) for the military sector that provides centralized management of information materials, user access control and effective content administration.

To achieve the stated goal, the following tasks were completed:

1. investigate the features of content management systems and the principles of their operation;
2. analyze modern software solutions in the field of content management;
3. study the specifics of information systems usage in the military sector;

4. define functional and non-functional requirements for the software system;
5. design the system architecture and perform UML modeling;
6. develop a database for storing users, categories and information materials;
7. implement the software system and conduct testing of its functional capabilities.

The object of the research is the process of information content management in military organizations.

The subject of the research is methods, models and software tools for implementing content management systems for the military sector.

The practical significance of the obtained results lies in the development of a software solution that can be used for centralized management of information materials in military organizations. The developed system provides capabilities for creating, editing, storing and publishing content, supports role-based access control and demonstrates the practical application of modern approaches to web-based information system development.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Позначення	Розшифрування
CMS	Content Management System (система управління контентом)
HTML	HyperText Markup Language (мова розмітки гіпертекстових документів)
CSS	Cascading Style Sheets (каскадні таблиці стилів)
PHP	PHP: Hypertext Preprocessor (серверна мова програмування для веброзробки)
SQL	Structured Query Language (мова структурованих запитів до баз даних)
SQLite	Вбудована реляційна система керування базами даних
CRUD	Create, Read, Update, Delete (створення, перегляд, оновлення та видалення даних)
UI	User Interface (користувацький інтерфейс)
URL	Uniform Resource Locator (уніфікований локатор ресурсів)
HTTP	HyperText Transfer Protocol (протокол передачі гіпертекстових даних)
ER-діаграма	Entity-Relationship Diagram (діаграма сутностей та зв'язків)
UML	Unified Modeling Language (уніфікована мова моделювання)
VS Code	Visual Studio Code (середовище розробки програмного забезпечення)
MVC	Model–View–Controller (архітектурний шаблон програмних систем)
OWASP	Open Worldwide Application Security Project (проект із безпеки вебзастосунків)
БД	База даних

ЗМІСТ

ВСТУП	12
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ.....	14
1.1 Поняття та особливості систем управління контентом	14
1.2 Аналіз існуючих систем управління контентом	15
1.3 Особливості використання систем управління контентом у військовій сфері.....	17
1.4 Формування вимог до системи	18
1.5 Постановка задачі розробки.....	20
Висновки до розділу 1	22
РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ВІЙСЬКОВОЇ СФЕРИ	24
2.1 Аналіз функціональних вимог до системи	24
2.2 Проектування архітектури системи	25
2.3 Проектування структури бази даних	29
2.4 Проектування ролей користувачів та механізмів доступу	32
2.5 Проектування користувацького інтерфейсу	35
2.6 Вибір технологій та засобів реалізації	42
Висновки до розділу 2	46
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ВІЙСЬКОВОЇ СФЕРИ.....	48
3.1 Реалізація структури бази даних	48
3.2 Реалізація модуля авторизації та автентифікації користувачів	51
3.3 Реалізація модуля керування контентом	56
3.4 Реалізація модуля керування користувачами та журналом подій	62
3.5 Тестування програмного засобу	65
Висновки до розділу 3	68
ВИСНОВКИ.....	70

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТКИ.....	75

ВСТУП

У сучасних умовах цифровізації діяльності організацій особливого значення набуває ефективне управління інформаційними ресурсами. Значні обсяги даних потребують впорядкованого зберігання, своєчасного оновлення та оперативного доступу до необхідної інформації. Особливо актуальним це питання є для установ та організацій військової сфери, де достовірність, структурованість і контроль доступу до інформації мають важливе значення для забезпечення належного рівня інформаційної взаємодії між користувачами системи

Для вирішення таких завдань широко застосовуються системи управління контентом (Content Management System, CMS), які дозволяють централізовано зберігати інформаційні матеріали, керувати їхнім життєвим циклом, організувати доступ користувачів відповідно до їхніх повноважень та забезпечувати зручну роботу з даними через вебінтерфейс. Використання подібних систем сприяє підвищенню ефективності роботи з інформацією та зменшенню часу на виконання рутинних операцій.

Незважаючи на наявність великої кількості універсальних систем управління контентом, їх використання не завжди відповідає специфічним вимогам окремих предметних областей. У військовій сфері особливого значення набуває реалізація механізмів розмежування доступу між користувачами, ведення журналу дій, структуризація інформаційних матеріалів за категоріями та забезпечення централізованого адміністрування контенту. Це обумовлює доцільність розроблення спеціалізованого програмного рішення, орієнтованого на потреби даної галузі.

Метою кваліфікаційної роботи є розробка системи управління контентом для військової сфери, яка забезпечує створення, редагування, зберігання та пошук інформаційних матеріалів, підтримує рольову модель доступу користувачів та дозволяє здійснювати централізоване адміністрування інформаційних ресурсів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити особливості використання систем управління контентом та проаналізувати сучасні підходи до організації роботи з інформаційними ресурсами;
- виконати аналіз існуючих програмних рішень у сфері управління контентом;
- сформулювати функціональні та нефункціональні вимоги до майбутньої системи;
- спроектувати структуру бази даних та архітектуру програмного забезпечення;
- реалізувати веборієнтовану систему управління контентом із використанням технологій PHP та SQLite;
- реалізувати механізми авторизації, розмежування прав доступу та ведення журналу дій користувачів;
- провести тестування створеної системи та оцінити результати її роботи.

Об'єктом дослідження є процес управління інформаційними матеріалами в системах військового призначення.

Предметом дослідження є методи, моделі та програмні засоби побудови систем управління контентом із рольовою моделлю доступу.

Практичне значення одержаних результатів полягає у створенні працездатної системи управління контентом для військової сфери, яка забезпечує зручне адміністрування інформаційних матеріалів, підтримує розмежування прав доступу між користувачами та може бути використана як основа для подальшого розвитку інформаційних систем відповідного призначення.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ

1.1 Поняття та особливості систем управління контентом

Система управління контентом (Content Management System, CMS) — це програмний засіб, що забезпечує створення, зберігання, редагування та публікацію інформаційних матеріалів без необхідності безпосереднього втручання у програмний код вебресурсу. Головна функція таких систем полягає у розмежуванні технічної складової від роботи з інформацією, що дозволяє користувачам різного рівня підготовки ефективно керувати контентом[1].

Поява та поширення CMS зумовлені стрімким зростанням обсягів цифрової інформації в організаціях різних форм власності. В умовах, коли інформаційні ресурси потребують постійного оновлення та адміністрування, ручне управління даними стає нераціональним і ненадійним. Використання спеціалізованих систем дозволяє автоматизувати більшість процесів, пов'язаних із керуванням контентом, та забезпечити централізований контроль над інформаційними ресурсами організації.

До основних функцій систем управління контентом належать: створення та редагування інформаційних матеріалів, їх класифікація за тематичними категоріями, організація зберігання та пошуку даних, а також розмежування доступу користувачів до функціональних можливостей системи. Більшість сучасних CMS також підтримують механізми журналювання подій, що забезпечують фіксацію дій користувачів і підвищують рівень контролю за роботою системи[2].

Невід'ємною складовою сучасних систем управління контентом є рольова модель доступу[9]. Відповідно до наданих повноважень користувачі отримують різний рівень доступу до функцій системи. Найбільш поширеною є триступенева модель, що передбачає ролі адміністратора, редактора та користувача з обмеженими правами. Такий підхід дозволяє забезпечити

інформаційну безпеку системи та запобігти несанкціонованому доступу до даних.

Сфера застосування CMS охоплює державні установи, освітні заклади, комерційні організації та спеціалізовані інформаційні системи. Незалежно від контексту використання, основним призначенням таких систем залишається централізоване керування інформаційними ресурсами, підвищення ефективності їх адміністрування та забезпечення контрольованого доступу до даних.

1.2 Аналіз існуючих систем управління контентом

На сьогодні ринок програмних рішень для управління контентом представлений значною кількістю платформ різного рівня складності та функціональності. Найбільш поширеними серед них є WordPress, Joomla та Drupal, які широко застосовуються для створення вебсайтів, інформаційних порталів і корпоративних ресурсів.

WordPress є найпоширенішою CMS у світі. Платформа відрізняється низьким порогом входу, великою бібліотекою готових шаблонів і розширень та розвиненою спільнотою розробників. Завдяки цим характеристикам WordPress дозволяє швидко розгортати інформаційні ресурси без значних витрат часу на програмування. Разом із тим широке використання сторонніх модулів може негативно впливати на продуктивність системи, а підтримка актуального рівня безпеки вимагає регулярного оновлення компонентів[21].

Joomla займає проміжне місце між простотою WordPress і функціональною складністю Drupal. Платформа надає ширші можливості для налаштування структури сайту та управління правами користувачів, підтримує розмежування доступу і дозволяє реалізовувати складніші інформаційні системи. Водночас адміністрування Joomla потребує більш глибоких технічних знань, що може ускладнювати її використання для невідготовленого персоналу[22].

Drupal орієнтований на розробку масштабних інформаційних ресурсів із підвищеними вимогами до безпеки та гнучкості. Система підтримує розширені моделі розмежування доступу і добре масштабується під зростаючі потреби організації. Однак висока складність налаштування та адміністрування суттєво збільшує вимоги до кваліфікації розробників і технічного персоналу[23].

Для наочного порівняння основних характеристик популярних систем управління контентом доцільно використати таблицю.

Характеристика	WordPress	Joomla	Drupal	Military CMS
Простота використання	Висока	Середня	Низька	Висока
Складність налаштування	Низька	Середня	Висока	Низька
Підтримка ролей користувачів	Так	Так	Так	Так
Керування матеріалами	Так	Так	Так	Так
Керування категоріями	Так	Так	Так	Так
Журнал дій користувачів	Через модулі	Через модулі	Так	Так
Пошук та фільтрація контенту	Так	Так	Так	Так
Спеціалізація для військової сфери	Ні	Ні	Ні	Так
Надлишковий функціонал	Високий	Середній	Високий	Низький
Простота розгортання	Висока	Середня	Низька	Висока

Таблиця 1.1 – Порівняння популярних CMS

Проведений аналіз свідчить про те, що розглянуті платформи орієнтовані переважно на універсальне використання і не завжди забезпечують достатній

рівень гнучкості для реалізації специфічних вимог окремих предметних областей. Зокрема, для військової сфери характерні підвищені вимоги до контролю доступу, фіксації дій користувачів та структурованого зберігання інформації, які у стандартних CMS або реалізовані частково, або потребують суттєвого доопрацювання. Це обумовлює доцільність розробки спеціалізованого програмного рішення в межах даної кваліфікаційної роботи.

1.3 Особливості використання систем управління контентом у військовій сфері

Застосування інформаційних систем у військовій сфері пов'язане з рядом специфічних вимог, які принципово відрізняють їх від програмних продуктів загального призначення. Ці вимоги зумовлені підвищеною відповідальністю за збереження інформації, необхідністю суворого контролю доступу до даних і критичністю стабільної роботи програмного забезпечення в умовах підвищеного навантаження.

Першочерговою вимогою є чітке розмежування прав доступу між різними категоріями користувачів. У військових структурах персонал різних рівнів працює з різними категоріями інформації, і змішування цих рівнів є неприпустимим з точки зору інформаційної безпеки. Рольова модель доступу в такому контексті є не факультативною функцією, а обов'язковою архітектурною складовою системи, що унеможливує несанкціонований доступ до даних, які виходять за межі службових повноважень користувача[29].

Не менш важливим є забезпечення повноцінного контролю за змінами інформаційних ресурсів. Механізм журналювання подій у системах військового призначення виконує роль інструменту аудиту: фіксує, хто, коли і які зміни вніс до системи. Наявність детального журналу дій дозволяє відстежувати історію операцій, виявляти порушення та підвищує загальний рівень підзвітності персоналу.

Окремої уваги потребує організація структурованого зберігання інформації. Значні обсяги документів, службових матеріалів і повідомлень, що накопичуються в системі, вимагають логічного впорядкування та забезпечення швидкого пошуку. Механізми категоризації контенту є в цьому контексті не просто зручністю, а функціональною необхідністю[30].

Важливою характеристикою є також вимога до зрозумілості інтерфейсу. Незважаючи на складність внутрішньої логіки системи, її інтерфейс повинен залишатись доступним для користувачів із різним рівнем технічної підготовки, що виключає тривале навчання персоналу і знижує ймовірність помилок при роботі з системою.

Перелічені особливості були враховані при формуванні концепції та визначенні функціональних вимог до системи Military CMS, розробленої в межах даної кваліфікаційної роботи.

1.4 Формування вимог до системи

Формування вимог є одним із найважливіших етапів розробки програмного забезпечення, оскільки саме на цьому етапі визначаються функціональні можливості майбутньої системи, її основні характеристики та обмеження. Від повноти й коректності сформованих вимог значною мірою залежить якість кінцевого програмного продукту, його відповідність потребам користувачів та ефективність практичного використання. Правильно сформульовані вимоги дозволяють уникнути неоднозначностей під час реалізації проєкту, забезпечують узгодженість між етапами проєктування та розробки, а також створюють основу для подальшого тестування програмного засобу.

На підставі проведеного аналізу предметної області було встановлено, що розроблювана система повинна забезпечувати централізоване керування інформаційними матеріалами, підтримувати роботу кількох категорій користувачів із різними рівнями доступу, надавати інструменти контролю за виконаними операціями та забезпечувати швидкий пошук необхідної

інформації [9, 10]. Крім цього, система повинна сприяти впорядкуванню інформаційних ресурсів та забезпечувати зручний механізм їх подальшого адміністрування.

Процес формування вимог до Military CMS здійснювався з урахуванням специфіки військової сфери, для якої характерними є підвищені вимоги до контролю доступу, відповідальності користувачів за виконані дії та необхідності підтримки актуального стану інформаційних матеріалів. У зв'язку з цим особливу увагу було приділено механізмам безпеки, журналювання дій та розмежування користувацьких повноважень.

До функціональних вимог системи належать авторизація користувачів, керування інформаційними матеріалами, керування категоріями, адміністрування облікових записів, ведення журналу подій та відображення зведеної статистики на головній сторінці. Після успішної авторизації кожен користувач отримує доступ лише до тих функціональних можливостей, які відповідають його ролі та рівню повноважень.

Для реалізації розмежування доступу визначено три основні ролі користувачів. Адміністратор має повний доступ до всіх модулів системи, включаючи керування користувачами, категоріями, матеріалами та журналом подій. Редактор може працювати з інформаційними матеріалами та категоріями, створювати нові записи й оновлювати існуючі, проте не має доступу до функцій адміністрування користувачів. Переглядач обмежений функціями перегляду та пошуку матеріалів без можливості їх редагування або видалення.

Запропонована рольова модель дозволяє реалізувати принцип мінімально необхідних привілеїв, відповідно до якого кожен користувач отримує лише ті можливості, які потрібні йому для виконання визначених завдань. Такий підхід сприяє підвищенню рівня безпеки системи, зменшує ризик несанкціонованих змін даних та забезпечує більш ефективний контроль доступу до інформаційних ресурсів.

Обов'язковою функціональною вимогою є автоматичне ведення журналу подій. Система повинна без участі користувача фіксувати ключові операції, пов'язані зі створенням, редагуванням і видаленням даних. Наявність такого механізму забезпечує прозорість роботи програмного засобу, дозволяє відстежувати історію змін та підвищує рівень відповідальності користувачів за виконані дії.

Серед нефункціональних вимог визначено інтуїтивно зрозумілий інтерфейс, стабільну роботу при виконанні типових операцій, коректну підтримку механізмів авторизації, надійне збереження даних та можливість подальшого супроводження програмного продукту. Особлива увага приділяється простоті використання системи, оскільки її користувачі можуть мати різний рівень технічної підготовки.

Для зберігання інформації обрано систему керування базами даних SQLite, яка не потребує розгортання окремого серверного програмного забезпечення та є доцільним рішенням для систем даного масштабу. Серверна частина реалізується засобами мови програмування PHP, яка забезпечує необхідні можливості для створення веборієнтованого програмного забезпечення та взаємодії з реляційними базами даних.

Таким чином, сформований перелік функціональних і нефункціональних вимог визначає основні характеристики Military CMS та створює основу для подальшого проектування архітектури системи, структури бази даних і програмних модулів.

1.5 Постановка задачі розробки

Результати аналізу предметної області та існуючих програмних рішень підтвердили, що наявні універсальні CMS-платформи не забезпечують достатнього рівня відповідності специфічним вимогам військової сфери в частині контролю доступу, фіксації дій користувачів та структурованого адміністрування інформаційних ресурсів. Більшість існуючих рішень орієнтована на широке коло користувачів і потребує додаткового

налаштування для використання в організаціях із підвищеними вимогами до безпеки та контролю інформації.

У зв'язку з цим актуальним завданням є розробка спеціалізованого програмного засобу, який враховуватиме особливості військової сфери та забезпечуватиме необхідний рівень функціональності без використання надлишкових компонентів. Створення власної системи дозволяє реалізувати лише ті можливості, які є необхідними для виконання поставлених завдань, що позитивно впливає на зручність використання та подальше адміністрування програмного продукту.

Основною метою кваліфікаційної роботи є розробка веборієнтованої системи управління контентом Military CMS, призначеної для централізованого керування інформаційними матеріалами, категоріями та обліковими записами користувачів із застосуванням ролівої моделі доступу.

Для досягнення поставленої мети необхідно вирішити низку взаємопов'язаних задач. Насамперед потрібно реалізувати механізм авторизації користувачів із розмежуванням прав доступу між ролями адміністратора, редактора та переглядача. Наступним етапом є створення модуля керування інформаційними матеріалами, який повинен забезпечувати виконання операцій додавання, редагування, видалення та пошуку записів. Також необхідно реалізувати модуль керування категоріями для структуризації контенту та спрощення навігації між інформаційними матеріалами.

Окремим завданням є розробка функціоналу адміністрування користувачів, що надаватиме можливість створювати нові облікові записи, редагувати інформацію про користувачів та здійснювати контроль за їхніми ролями в системі. Крім цього, необхідно впровадити механізм автоматичного журналювання дій користувачів, який забезпечуватиме фіксацію основних операцій та дозволить здійснювати аудит роботи програмного засобу.

У процесі реалізації поставлених задач передбачається використання сучасних вебтехнологій та реляційної бази даних для організації надійного

зберігання інформації. Важливими аспектами також є забезпечення цілісності даних, підтримка безпечної автентифікації користувачів і можливість подальшого розвитку програмного продукту.

Результатом виконання роботи має стати повністю функціональна система управління контентом Military CMS, яка забезпечуватиме централізоване керування інформаційними ресурсами, підтримку ролівої моделі доступу, контроль дій користувачів та ефективне адміністрування інформаційного наповнення. Створений програмний засіб може використовуватися як практичний інструмент для організації роботи з інформаційними матеріалами у військовій сфері.

Висновки до розділу 1

У першому розділі кваліфікаційної роботи проведено аналіз предметної області систем управління контентом та досліджено особливості їх застосування у військовій сфері. Встановлено, що CMS є ефективним інструментом централізованого керування інформаційними ресурсами, однак існуючі універсальні платформи не повною мірою відповідають специфічним вимогам, характерним для військового контексту використання.

Порівняльний аналіз платформ WordPress, Joomla та Drupal дозволив виявити їхні основні переваги та обмеження, а також підтвердив доцільність розробки спеціалізованого програмного рішення. Проведене дослідження показало, що використання універсальних платформ часто потребує значної кількості додаткових налаштувань і розширень для забезпечення необхідного рівня контролю доступу та журналювання дій користувачів.

Дослідження особливостей військової сфери дозволило сформулювати ключові вимоги до майбутньої системи, серед яких рольове розмежування доступу, автоматичне журналювання дій, структуроване зберігання інформаційних матеріалів, підтримка категоризації контенту та забезпечення доступного інтерфейсу для користувачів із різним рівнем технічної підготовки. На основі отриманих результатів було визначено склад

функціональних і нефункціональних вимог до Military CMS та сформульовано основні задачі її розробки.

Крім того, проведений аналіз дозволив визначити технологічні підходи, які доцільно використовувати під час створення програмного продукту. Зокрема, обґрунтовано використання рольової моделі доступу, централізованого зберігання даних та механізмів автоматичного аудиту дій користувачів. Визначені вимоги створюють основу для подальшого проєктування архітектури системи, структури бази даних та функціональних модулів.

Отже, результати першого розділу підтверджують актуальність розробки спеціалізованої системи управління контентом Military CMS та формують необхідне теоретичне підґрунтя для переходу до наступного етапу роботи, пов'язаного з проєктуванням програмного засобу та вибором технічних рішень для його реалізації.

РОЗДІЛ 2. ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ВІЙСЬКОВОЇ СФЕРИ

2.1 Аналіз функціональних вимог до системи

Проектування програмного забезпечення є одним із найважливіших етапів розробки інформаційних систем. Саме на цьому етапі визначаються функціональні можливості майбутнього програмного продукту, структура його компонентів та особливості взаємодії між окремими модулями. Правильно сформовані вимоги дозволяють створити систему, яка відповідає потребам користувачів та забезпечує виконання поставлених завдань[1].

У результаті аналізу предметної області було встановлено, що система управління контентом для військової сфери повинна забезпечувати централізоване зберігання інформаційних матеріалів, підтримувати рольову модель доступу та надавати засоби адміністрування інформаційних ресурсів через вебінтерфейс. Основною метою системи є спрощення процесів роботи з контентом та забезпечення контролю за доступом до інформації відповідно до повноважень користувачів[6].

Однією з ключових функцій системи є авторизація користувачів. Для отримання доступу до функціональних можливостей Military CMS користувач повинен пройти процедуру автентифікації шляхом введення логіна та пароля. Після успішної перевірки облікових даних система створює користувацьку сесію та надає доступ до відповідних модулів.

Важливою функціональною вимогою є підтримка рольової моделі доступу. У розробленій системі передбачено три категорії користувачів: адміністратор, редактор та переглядач. Адміністратор має повний доступ до всіх функціональних можливостей системи, включаючи керування користувачами та перегляд журналу дій. Редактор може працювати з інформаційними матеріалами та категоріями, а переглядач має доступ лише до перегляду інформації без можливості її зміни.

Центральним компонентом системи є модуль керування контентом. Його функціональні можливості передбачають створення нових матеріалів, редагування існуючих записів, видалення інформації та виконання пошуку за наявними матеріалами. Для забезпечення впорядкованого зберігання даних передбачено використання категорій, до яких можуть належати інформаційні матеріали.

Окремою функцією системи є керування користувачами. Даний модуль дозволяє створювати нові облікові записи, редагувати інформацію про користувачів, змінювати ролі доступу та виконувати адміністрування користувацьких даних. Застосування такого підходу забезпечує гнучке налаштування прав доступу та підвищує рівень інформаційної безпеки системи.

Для забезпечення контролю за роботою програмного продукту в системі реалізовано механізм журналювання подій. Усі важливі дії користувачів автоматично фіксуються в журналі, що дозволяє здійснювати моніторинг роботи системи та відстежувати історію виконаних операцій.

На основі визначених функціональних вимог було сформовано структуру майбутньої системи та визначено основні модулі, необхідні для її реалізації. Отримані результати стали основою для подальшого проектування архітектури Military CMS та структури бази даних.

2.2 Проектування архітектури системи

Після визначення функціональних вимог до програмного продукту наступним етапом розробки є проектування архітектури системи. Архітектура визначає структуру програмного забезпечення, склад його основних компонентів та механізми взаємодії між ними. Від правильності побудови архітектури залежить зручність подальшого супроводу системи, можливість її розширення та ефективність роботи програмного продукту в цілому.

Проектування архітектури є одним із найважливіших етапів створення будь-якої інформаційної системи. Саме на цьому етапі визначається спосіб

організації програмних компонентів, порядок їх взаємодії та принципи обміну даними. Грамотно спроектована архітектура дозволяє забезпечити стабільну роботу системи, спрощує процес внесення змін та створює умови для подальшого розвитку програмного забезпечення.

Під час проектування Military CMS було обрано модульний підхід до побудови системи. Відповідно до даного підходу функціональні можливості програмного продукту поділяються на окремі незалежні модулі, кожен з яких відповідає за виконання конкретного набору завдань [11]. Така організація дозволяє спростити процес розробки, тестування та подальшої модернізації програмного забезпечення.

Використання модульної архітектури забезпечує низку переваг. По-перше, кожен компонент може розроблятися та вдосконалюватися незалежно від інших модулів системи. По-друге, значно спрощується локалізація та усунення можливих помилок, оскільки функціональність кожного модуля чітко визначена. По-третє, у разі необхідності система може бути доповнена новими компонентами без суттєвого впливу на вже реалізовані функції.

До складу розробленої системи входять модуль авторизації користувачів, модуль керування користувачами, модуль керування інформаційними матеріалами, модуль керування категоріями, модуль журналу дій та модуль головної панелі керування. Усі компоненти взаємодіють між собою через серверну частину системи та використовують спільну базу даних для зберігання інформації.

Під час проектування також було враховано необхідність забезпечення централізованого зберігання інформації. Саме тому всі програмні модулі працюють із єдиною базою даних, що дозволяє підтримувати актуальність інформації та забезпечує її цілісність. Такий підхід спрощує адміністрування системи та зменшує ризик виникнення суперечностей між окремими компонентами програмного продукту.

Структуру взаємодії основних модулів системи наведено на рисунку 2.1.

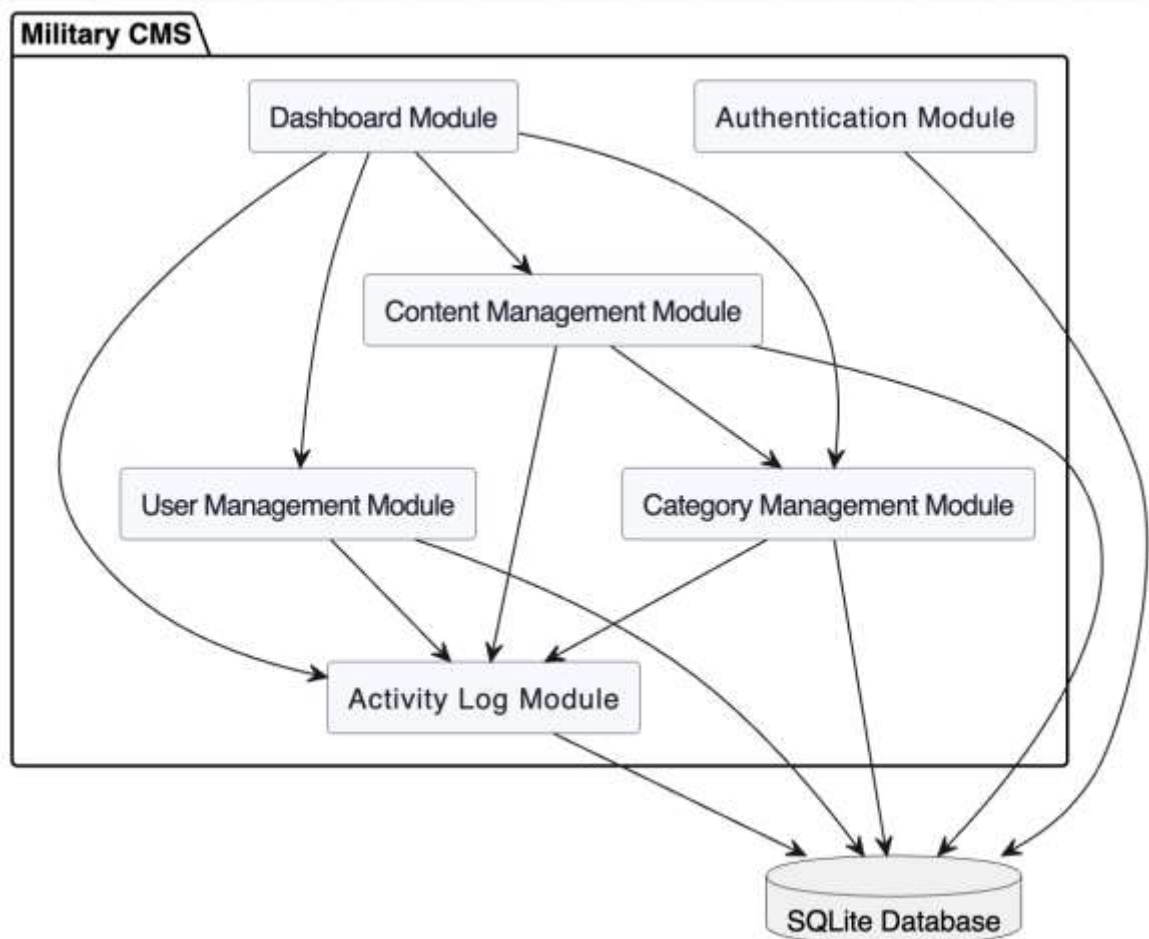


Рис. 2.1. Діаграма компонентів системи Military CMS

Рисунок 2.1 демонструє основні компоненти системи та взаємозв'язки між ними. Представлена структура відображає загальний підхід до організації роботи Military CMS і показує взаємодію між функціональними модулями та централізованим сховищем даних.

Центральним елементом архітектури виступає головна панель керування, через яку користувач отримує доступ до функціональних можливостей програмного продукту. Після проходження процедури авторизації система визначає роль користувача та відповідно до встановлених повноважень надає доступ до необхідних модулів. Такий підхід дозволяє забезпечити контроль доступу до інформації та підтримувати необхідний рівень інформаційної безпеки.

Модуль авторизації забезпечує перевірку облікових даних користувача та створення активної сесії після успішного входу до системи. Даний компонент є одним із ключових елементів системи безпеки, оскільки саме він здійснює контроль доступу до функціональних можливостей програмного продукту.

Модуль керування користувачами відповідає за створення, редагування та адміністрування облікових записів. За допомогою цього модуля адміністратор може керувати ролями користувачів, змінювати їхні персональні дані та контролювати доступ до ресурсів системи.

Модуль керування контентом забезпечує роботу з інформаційними матеріалами, включаючи їх створення, редагування, перегляд та видалення. Саме цей компонент реалізує основне призначення Military CMS як системи управління контентом і забезпечує користувачам можливість централізованої роботи з інформаційними ресурсами.

Для впорядкування інформації використовується модуль керування категоріями, який дозволяє групувати матеріали відповідно до їх тематики та призначення. Використання категорій спрощує навігацію між матеріалами та підвищує зручність роботи із системою.

Додатково в програмному продукті реалізовано модуль журналу дій, що забезпечує автоматичну фіксацію ключових операцій користувачів. Наявність журналу дозволяє здійснювати аудит роботи системи, відстежувати зміни даних та контролювати виконання критично важливих операцій.

Усі модулі взаємодіють із централізованою базою даних SQLite, у якій зберігаються відомості про користувачів, матеріали, категорії та журнали подій. Використання єдиного сховища даних забезпечує цілісність інформації, спрощує адміністрування та підвищує ефективність роботи системи [14].

Окремою перевагою запропонованої архітектури є можливість подальшого розвитку програмного продукту. У майбутньому структура системи може бути розширена за рахунок додавання нових функціональних модулів без необхідності суттєвої перебудови вже реалізованих компонентів.

Це забезпечує гнучкість програмного рішення та створює передумови для його подальшого вдосконалення.

Обрана архітектура дозволяє забезпечити логічний розподіл функціональних можливостей системи, спрощує підтримку програмного забезпечення та створює передумови для подальшого розширення функціоналу Military CMS.

2.3 Проєктування структури бази даних

Ефективність функціонування системи управління контентом значною мірою залежить від правильної організації зберігання даних. Структура бази даних повинна забезпечувати швидкий доступ до інформації, підтримувати цілісність даних та створювати умови для подальшого розширення функціональних можливостей програмного продукту. Саме тому етап проєктування бази даних є одним із ключових під час розробки інформаційних систем.

База даних виступає центральним компонентом Military CMS, оскільки саме в ній зберігаються всі відомості, необхідні для роботи програмного продукту. Від правильності побудови структури даних залежить не лише швидкодія системи, але й можливість ефективного керування інформаційними ресурсами, реалізації механізмів авторизації та підтримки взаємодії між окремими функціональними модулями.

Для реалізації Military CMS було обрано реляційну модель зберігання даних. Такий підхід дозволяє організувати інформацію у вигляді взаємопов'язаних таблиць та забезпечити логічну структуру даних. Як система керування базами даних використовується SQLite, що обумовлено простотою розгортання, відсутністю необхідності встановлення окремого серверного програмного забезпечення та достатньою продуктивністю для системи даного масштабу [16,25].

Перевагою використання SQLite є також компактність та зручність адміністрування. Усі дані зберігаються в одному файлі бази даних, що

спрощує резервне копіювання, перенесення системи між різними середовищами та виконання процедур відновлення інформації. Для навчальних і спеціалізованих вебзастосунків подібного масштабу SQLite є одним із найбільш доцільних рішень.

Під час проєктування структури бази даних особлива увага приділялася забезпеченню мінімізації дублювання інформації та підтримці логічних зв'язків між окремими сутностями. Використання принципів нормалізації дозволяє уникнути надлишкового зберігання даних, зменшує ризик виникнення суперечностей та підвищує ефективність обробки інформації.

У процесі проєктування було визначено чотири основні сутності, необхідні для функціонування системи: користувачі, інформаційні матеріали, категорії та журнал дій. Кожна з цих сутностей реалізована у вигляді окремої таблиці бази даних і виконує власне функціональне призначення.

Таблиця users призначена для зберігання інформації про зареєстрованих користувачів системи. У ній містяться відомості про ім'я користувача, логін, пароль, роль та дату створення облікового запису. Саме ця таблиця використовується під час авторизації та визначення прав доступу до функціональних можливостей системи.

Таблиця articles забезпечує зберігання інформаційних матеріалів. Для кожного запису зберігаються назва, текст матеріалу, статус публікації, дата створення та дата останнього оновлення. Крім того, таблиця містить зовнішні ключі, які забезпечують зв'язок із таблицями користувачів та категорій.

Таблиця categories використовується для класифікації інформаційних матеріалів. Використання категорій дозволяє впорядковувати контент за тематичними напрямками та спрощує процес навігації і пошуку інформації в межах системи.

Для забезпечення контролю за роботою користувачів передбачено таблицю logs, у якій зберігаються відомості про виконані операції. Кожний запис журналу містить інформацію про користувача, виконану дію та час її

здійснення. Наявність такого механізму підвищує рівень адміністрування системи та дозволяє відстежувати історію змін.

Структуру взаємозв'язків між основними сутностями бази даних наведено на рисунку 2.2.

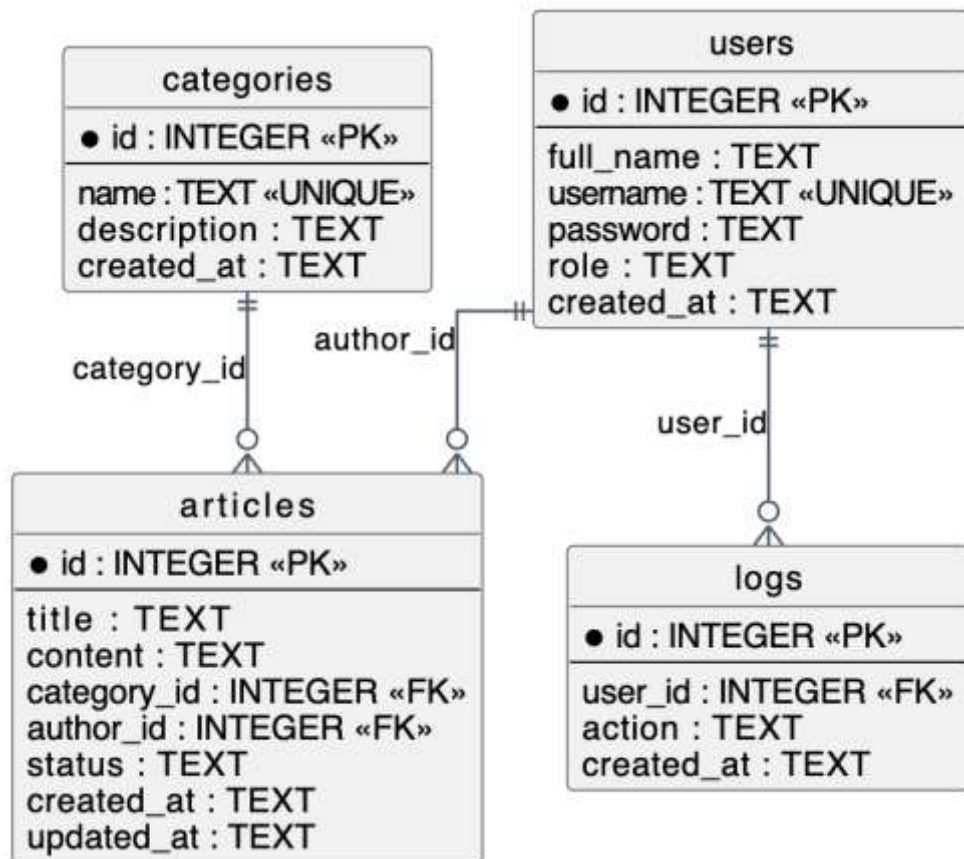


Рис. 2.1. ER-діаграма бази даних системи Military CMS

Як видно з рисунка 2.2, між таблицями бази даних існують логічні зв'язки типу «один до багатьох». Один користувач може створювати декілька інформаційних матеріалів та формувати багато записів у журналі дій. Водночас одна категорія може містити необмежену кількість інформаційних матеріалів. Така структура забезпечує відсутність дублювання даних та відповідає принципам нормалізації реляційних баз даних.

Використання зовнішніх ключів дозволяє підтримувати цілісність даних та забезпечувати коректність взаємозв'язків між окремими таблицями. Наприклад, кожний інформаційний матеріал пов'язується з конкретним

автором та певною категорією, а записи журналу подій містять посилання на користувача, який виконав відповідну операцію. Завдяки цьому система може автоматично визначати автора матеріалу, відстежувати історію змін та забезпечувати контроль за виконаними діями.

Запропонована структура бази даних також створює основу для подальшого розширення функціональних можливостей програмного продукту. У разі необхідності до системи можуть бути додані нові таблиці та зв'язки без суттєвої перебудови вже існуючої структури. Така гнучкість є важливою перевагою під час супроводу та модернізації програмного забезпечення.

Важливою особливістю розробленої моделі є забезпечення єдиного інформаційного простору для всіх функціональних модулів Military CMS. Незалежно від того, чи виконується авторизація користувача, створення нового матеріалу, редагування категорії або перегляд журналу дій, усі операції здійснюються через централізовану базу даних. Це дозволяє підтримувати актуальність інформації та забезпечує узгоджену роботу всієї системи.

Спроектowana структура бази даних створює надійне підґрунтя для роботи всіх функціональних модулів Military CMS. Використання взаємопов'язаних таблиць дозволяє підтримувати цілісність інформації, забезпечувати коректну роботу механізмів авторизації та реалізовувати операції керування контентом відповідно до визначених вимог.

2.4 Проектування ролей користувачів та механізмів доступу

Однією з ключових вимог до систем управління контентом, що використовуються у військовій сфері, є забезпечення контрольованого доступу до інформаційних ресурсів. Надання користувачам необмежених повноважень створює ризик несанкціонованої зміни або видалення інформації, що може негативно вплинути на роботу всієї системи. Саме тому під час проектування Military CMS особливу увагу було приділено реалізації механізмів розмежування прав доступу.

Забезпечення безпеки інформації є одним із базових принципів функціонування сучасних інформаційних систем. У випадку систем управління контентом контроль доступу дозволяє не лише захистити дані від несанкціонованих змін, але й забезпечити відповідність виконуваних дій службовим обов'язкам конкретних користувачів. Правильно організована система повноважень сприяє підвищенню рівня керованості програмного продукту та знижує ризик виникнення помилок під час роботи з інформаційними ресурсами.

Для вирішення даного завдання було обрано рольову модель доступу, відповідно до якої кожен користувач отримує певний набір дозволених дій залежно від своєї ролі в системі. Такий підхід дозволяє централізовано керувати повноваженнями користувачів та спрощує адміністрування програмного забезпечення.

Перевагою рольової моделі є її гнучкість та простота підтримки. У разі зміни функціональних вимог адміністратор системи може коригувати перелік дозволених операцій без необхідності внесення суттєвих змін до архітектури програмного продукту. Крім того, використання ролей дозволяє реалізувати принцип мінімально необхідних привілеїв, відповідно до якого користувач отримує лише ті права, які необхідні йому для виконання своїх функцій.

У межах розробленої системи передбачено три основні ролі користувачів: адміністратор, редактор та переглядач. Кожна з них має власний набір функціональних можливостей та обмежень.

Адміністратор є користувачем із найвищим рівнем доступу. Йому доступні всі функції системи без винятку. До повноважень адміністратора належить керування користувачами, створення та редагування категорій, адміністрування інформаційних матеріалів, перегляд журналу дій та контроль загального стану системи. Саме адміністратор відповідає за підтримку працездатності програмного продукту та налаштування доступу інших користувачів.

Редактор має доступ до функцій, пов'язаних із роботою з контентом. Він може створювати нові матеріали, редагувати існуючі записи, видаляти інформацію та працювати з категоріями. Водночас редактор не має права змінювати параметри облікових записів інших користувачів та не може виконувати адміністративні операції, пов'язані з налаштуванням системи.

Переглядач є користувачем із найменшим рівнем доступу. Його функціональні можливості обмежуються переглядом інформаційних матеріалів, використанням пошуку та роботою з власним профілем. Внесення будь-яких змін до інформаційних ресурсів системи для цієї категорії користувачів заборонене.

Для наочного відображення прав доступу користувачів доцільно використати матрицю ролей, наведену в таблиці 2.2.

Функція системи	Адміністратор	Редактор	Переглядач
Авторизація	Так	Так	Так
Перегляд Dashboard	Так	Так	Так
Перегляд матеріалів	Так	Так	Так
Пошук матеріалів	Так	Так	Так
Створення матеріалів	Так	Так	Ні
Редагування матеріалів	Так	Так	Ні
Видалення матеріалів	Так	Так	Ні
Керування категоріями	Так	Так	Ні
Керування користувачами	Так	Ні	Ні
Перегляд журналу дій	Так	Ні	Ні
Зміна власного профілю	Так	Так	Так

Таблиця 2.2 – Матриця прав доступу користувачів

Наведена матриця дозволяє чітко визначити межі відповідальності кожної категорії користувачів та забезпечує зрозумілий механізм розмежування доступу до функціональних можливостей системи. Завдяки цьому користувачі працюють лише з тими компонентами програмного продукту, які необхідні для виконання їхніх завдань.

Особливістю реалізованого підходу є те, що перевірка прав доступу виконується автоматично під час відкриття відповідних сторінок системи. У випадку спроби отримання доступу до функціоналу, який не відповідає ролі користувача, система блокує виконання операції та перенаправляє користувача до дозволених розділів програмного продукту. Такий механізм дозволяє підвищити рівень захищеності системи та мінімізувати ризик випадкових або навмисних порушень встановлених правил доступу.

Додатковою перевагою рольової моделі є можливість масштабування системи. У разі необхідності до Military CMS можуть бути додані нові ролі користувачів із власними наборами повноважень без суттєвої зміни вже реалізованих механізмів контролю доступу. Це створює передумови для подальшого розвитку програмного продукту та його адаптації до нових вимог.

Використання рольової моделі дозволяє забезпечити належний рівень інформаційної безпеки та мінімізувати ризик виконання користувачами операцій, які виходять за межі їхніх службових повноважень.Dodatkowoю перевагою такого підходу є спрощення супроводу системи, оскільки всі механізми контролю доступу реалізуються централізовано та можуть бути легко модифіковані в разі зміни вимог.

Запропонована модель доступу повністю відповідає функціональним вимогам Military CMS та забезпечує безпечну взаємодію користувачів із інформаційними ресурсами системи.

2.5 Проектування користувацького інтерфейсу

Однією з важливих складових будь-якої інформаційної системи є користувацький інтерфейс. Саме через інтерфейс здійснюється взаємодія

користувача з програмним забезпеченням, тому від його зручності та зрозумілості значною мірою залежить ефективність роботи всієї системи. Під час проєктування Military CMS особлива увага приділялася забезпеченню простоти використання та логічності розташування елементів керування.

Розроблювана система орієнтована на користувачів із різним рівнем технічної підготовки, тому інтерфейс повинен забезпечувати швидкий доступ до основних функцій без необхідності тривалого навчання. Для досягнення цієї мети було використано класичну структуру вебзастосунку, що включає сторінку авторизації, головну панель керування та набір функціональних сторінок для роботи з інформаційними ресурсами.

Під час проєктування інтерфейсу враховувалися сучасні принципи побудови вебзастосунків, серед яких послідовність розташування елементів, мінімізація кількості дій для виконання типових операцій та забезпечення швидкої навігації між функціональними модулями. Важливим завданням було створення такого інтерфейсу, який не перевантажує користувача зайвою інформацією та дозволяє швидко знаходити необхідні функції.

Особлива увага приділялася уніфікації дизайну окремих сторінок системи. Використання однакових принципів побудови інтерфейсу для різних модулів дозволяє скоротити час адаптації користувачів та забезпечує більш комфортну роботу із програмним продуктом. Завдяки цьому користувач, який навчився працювати з одним модулем системи, може без додаткового навчання використовувати й інші функціональні можливості Military CMS.

Першим елементом взаємодії користувача із системою є сторінка авторизації. Вона призначена для введення логіна та пароля і забезпечує перевірку облікових даних перед наданням доступу до функціональних можливостей Military CMS. Під час проєктування даної сторінки було обрано мінімалістичний підхід, що дозволяє користувачеві швидко виконати вхід до системи без відволікання на другорядні елементи інтерфейсу.

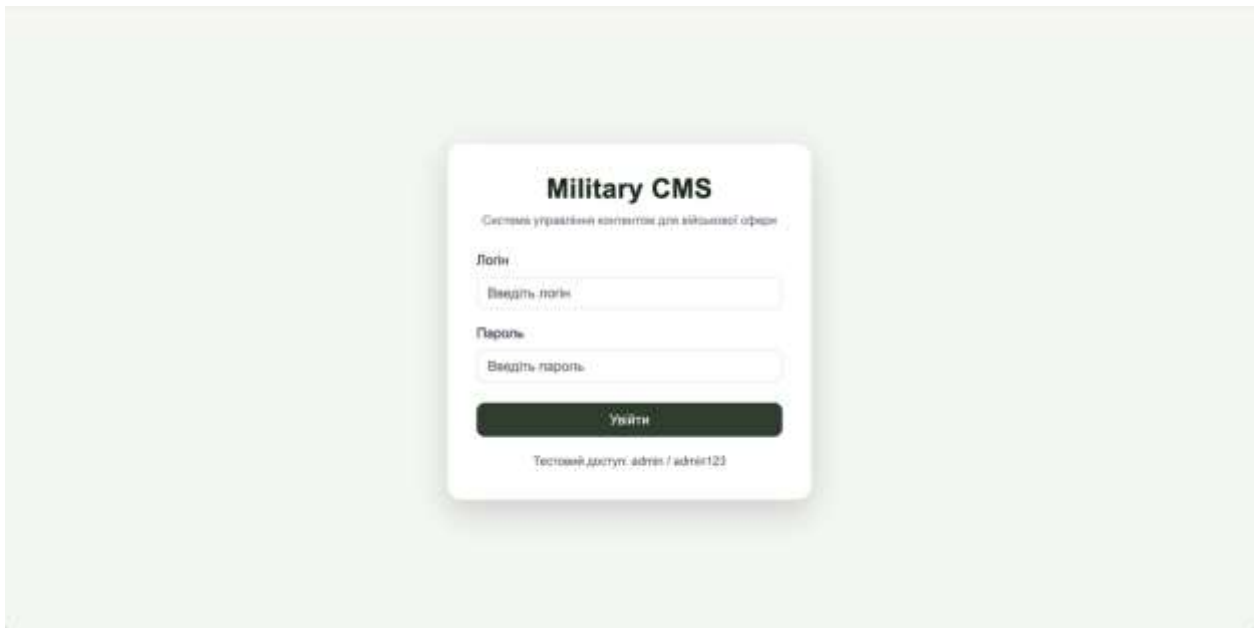


Рисунок 2.3 - Сторінка авторизації Military CMS

Представлена сторінка авторизації є початковою точкою взаємодії користувача із системою. Простий та зрозумілий інтерфейс дозволяє мінімізувати ймовірність помилок під час введення облікових даних та забезпечує швидкий доступ до функціональних можливостей програмного продукту.

Після успішного входу користувач потрапляє до головної панелі керування. Dashboard виконує роль центрального елемента навігації та надає швидкий доступ до основних модулів системи. На даній сторінці відображається інформація про поточного користувача, його роль у системі та статистичні відомості щодо наявних матеріалів, категорій і користувачів.

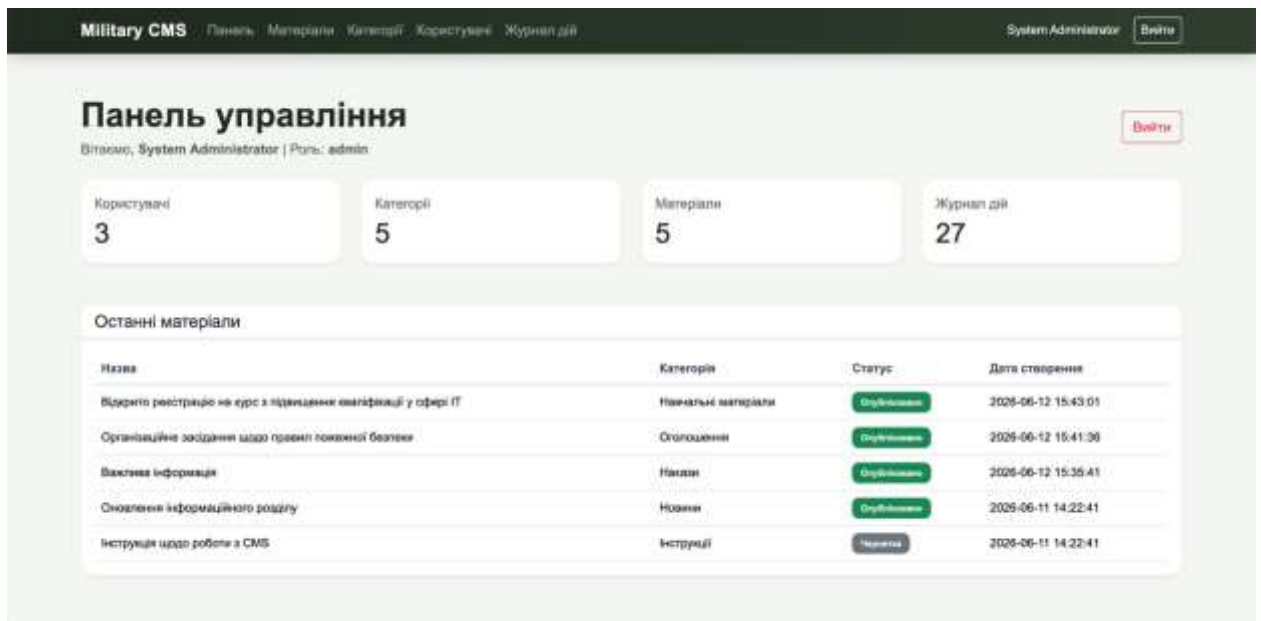


Рисунок 2.4 - Головна панель керування

Головна панель керування виконує роль інформаційного центру системи. Саме з неї користувач переходить до основних модулів Military CMS та отримує загальне уявлення про стан інформаційних ресурсів. Використання Dashboard дозволяє зменшити кількість переходів між сторінками та підвищує ефективність роботи з програмним продуктом.

Для роботи з інформаційними матеріалами передбачено окремий модуль керування контентом. Інтерфейс даного модуля дозволяє переглядати список матеріалів, здійснювати пошук необхідних записів та виконувати операції створення, редагування або видалення інформації залежно від наданих прав доступу. При проектуванні сторінки особлива увага приділялася забезпеченню зручного представлення даних у табличному вигляді та швидкому доступу до основних операцій.

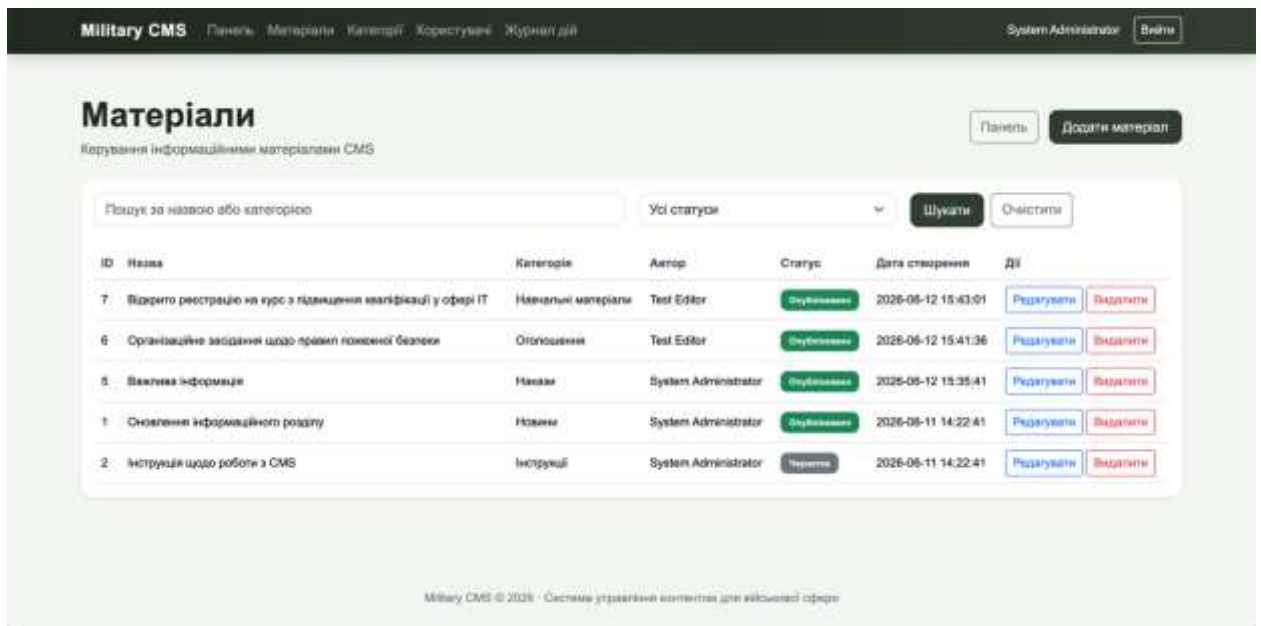


Рисунок 2.5 – Сторінка керування матеріалами

Під час проектування сторінки керування матеріалами було враховано необхідність роботи з великою кількістю інформаційних записів. Саме тому основним елементом інтерфейсу є таблиця, яка забезпечує компактне представлення даних та дозволяє швидко виконувати типові операції без переходу до додаткових сторінок.

Окремий модуль системи призначений для роботи з категоріями. Використання категорій дозволяє впорядковувати інформаційні матеріали за тематичними напрямками та спрощує процес навігації. Інтерфейс керування категоріями побудовано за аналогічним принципом до інших адміністративних сторінок системи, що забезпечує єдність користувацького досвіду.

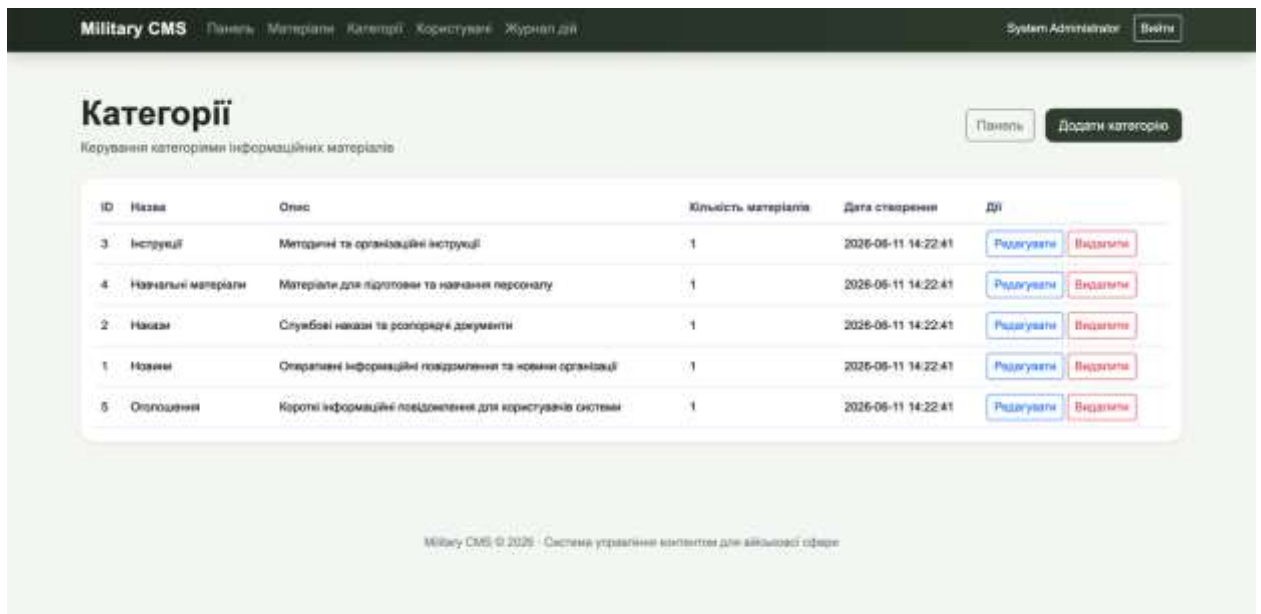


Рисунок 2.6 – Сторінка керування категоріями

Єдина стилістика оформлення сторінок керування категоріями та матеріалами забезпечує цілісність користувацького інтерфейсу. Такий підхід дозволяє зменшити навантаження на користувача та спрощує освоєння функціональних можливостей системи.

Для адміністратора системи передбачено окремий модуль керування користувачами. Даний компонент дозволяє переглядати список облікових записів, створювати нових користувачів, змінювати ролі доступу та виконувати адміністрування користувацьких даних. Завдяки цьому забезпечується централізоване управління доступом до ресурсів Military CMS.

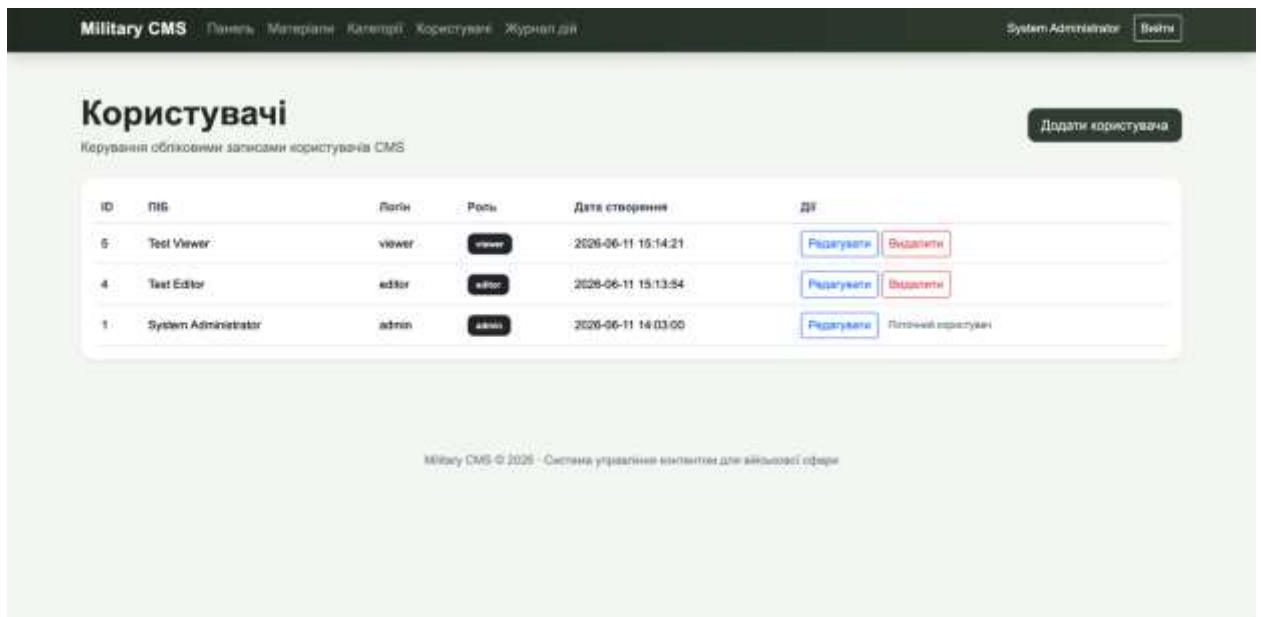


Рисунок 2.7 – Сторінка керування користувачами

Модуль керування користувачами є важливим інструментом адміністрування системи. Завдяки його використанню забезпечується централізований контроль доступу до ресурсів Military CMS та підтримується актуальність відомостей про користувачів програмного продукту.

Важливим елементом користувацького інтерфейсу є сторінка журналу дій, призначена для моніторингу операцій, виконаних користувачами системи. Даний модуль дозволяє переглядати інформацію про створення, редагування та видалення матеріалів, а також інші події, що відбуваються під час роботи програмного продукту. Наявність журналу дій забезпечує додатковий рівень контролю за використанням системи та сприяє підвищенню рівня інформаційної безпеки.

Під час проєктування сторінки журналу особлива увага приділялася зручності представлення інформації. Дані відображаються у табличному вигляді із зазначенням користувача, виконаної дії та часу її здійснення. Такий підхід дозволяє швидко аналізувати історію роботи системи та відстежувати зміни, внесені користувачами.

ID	Користувач	Роль	Дія	Дата і час
27	Test Editor	editor	Створено матеріал: Відкрило реєстрацію на курс з підвищення кваліфікації у сфері IT	2026-06-12 15:43:01
26	Test Editor	editor	Створено матеріал: Організаційне засідання щодо правил поведінки ботів	2026-06-12 15:41:36
25	System Administrator	admin	Ведано категорію: testing viewer	2026-06-12 15:35:55
24	System Administrator	admin	Створено матеріал: Висока інформація	2026-06-12 15:35:41
23	Test Viewer	viewer	Користувач змінив власний пароль	2026-06-11 15:25:58
22	Test Viewer	viewer	Користувач змінив власний пароль	2026-06-11 15:25:33
21	Test Editor	editor	Створено категорію: testing viewer	2026-06-11 15:16:12
20	Test Editor	editor	Ведано категорію: testing editor	2026-06-11 15:16:03
19	Test Editor	editor	Створено категорію: testing editor	2026-06-11 15:15:58
18	System Administrator	admin	Оновлено користувача: editor	2026-06-11 15:15:19
17	System Administrator	admin	Створено користувача: viewer	2026-06-11 15:14:21
16	System Administrator	admin	Створено користувача: editor	2026-06-11 15:13:54
15	System Administrator	admin	Оновлено користувача: admin	2026-06-11 15:04:50

Рисунок 2.8 – Сторінка журналу дій

Сторінка журналу дій надає адміністратору можливість оперативно аналізувати події, що відбуваються в системі. Завдяки централізованому зберіганню інформації про виконані операції забезпечується можливість контролю роботи користувачів та виявлення потенційних порушень установлених правил використання програмного забезпечення.

Наявність окремого модуля журналювання є особливо важливою для систем, що використовуються у військовій сфері, оскільки дозволяє здійснювати аудит виконаних операцій та підвищує рівень відповідальності користувачів під час роботи з інформаційними ресурсами.

Таким чином, під час проектування користувацького інтерфейсу було реалізовано принципи простоти, логічності та функціональної цілісності. Запропонована структура інтерфейсу забезпечує швидкий доступ до необхідних функцій системи та створює комфортні умови для роботи користувачів різних категорій.

2.6 Вибір технологій та засобів реалізації

Важливим етапом проектування програмного забезпечення є вибір технологій та інструментів розробки. Від правильності такого вибору

залежить складність реалізації проєкту, швидкість розробки, зручність супроводу програмного продукту та можливість його подальшого розвитку. Під час створення системи Military CMS було обрано набір технологій, які забезпечують реалізацію необхідного функціоналу та відповідають поставленим вимогам.

Вибір технологічного стеку здійснювався з урахуванням особливостей розроблюваної системи, її функціонального призначення та вимог до середовища експлуатації. Одним із ключових критеріїв була можливість створення повноцінного вебзастосунку без використання складної серверної інфраструктури. Крім того, обрані технології повинні були забезпечувати простоту розгортання, достатню продуктивність та можливість подальшого розвитку програмного продукту.

Для реалізації серверної частини системи використано мову програмування PHP. Дана технологія широко застосовується під час розробки веборієнтованих інформаційних систем та забезпечує ефективну взаємодію між користувацьким інтерфейсом і базою даних. Однією з переваг PHP є простота розгортання, значна кількість доступних бібліотек та підтримка практично всіма сучасними вебсерверами. Крім того, використання PHP дозволяє реалізувати механізми авторизації користувачів, обробки форм та роботи з базами даних без необхідності застосування додаткових програмних платформ [24].

Важливим фактором вибору PHP стала також його широка поширеність та наявність великої кількості навчальних матеріалів і технічної документації. Це спрощує процес супроводу програмного забезпечення та дозволяє швидше впроваджувати нові функціональні можливості у майбутньому.

Для зберігання даних було обрано систему керування базами даних SQLite. На відміну від MySQL або PostgreSQL, SQLite не потребує встановлення окремого серверного програмного забезпечення, оскільки вся база даних зберігається у вигляді одного файлу [25]. Такий підхід значно спрощує розгортання програмного продукту та є доцільним для систем

невеликого та середнього масштабу. Використання SQLite також дозволяє зменшити вимоги до апаратних ресурсів і спростити процес резервного копіювання даних.

Під час розробки користувацького інтерфейсу використовувалися технології HTML та CSS. Мова HTML забезпечує формування структури вебсторінок, тоді як CSS відповідає за їх візуальне оформлення. Поєднання цих технологій дозволяє створювати зрозумілий та зручний інтерфейс користувача, який підтримується всіма сучасними веббраузерами.

Для прискорення розробки інтерфейсу було використано фреймворк Bootstrap. Його застосування дозволило реалізувати адаптивний дизайн сторінок та забезпечити однакове відображення елементів інтерфейсу на різних пристроях. Крім того, Bootstrap містить готові компоненти інтерфейсу, що суттєво скорочує час розробки та забезпечує сучасний зовнішній вигляд програмного продукту [26].

Використання Bootstrap також сприяло забезпеченню єдиного стилю оформлення всіх сторінок системи. Це позитивно впливає на сприйняття інтерфейсу користувачами та забезпечує більш комфортну взаємодію із програмним продуктом.

Як середовище розробки використовувався редактор Visual Studio Code. Даний інструмент підтримує велику кількість мов програмування, має розвинену систему розширень та забезпечує зручну роботу з файлами проєкту. Важливою перевагою Visual Studio Code є його доступність на різних операційних системах та можливість інтеграції з інструментами контролю версій [27].

Для запуску та тестування вебзастосунку використовувався вбудований сервер PHP. Використання такого підходу дозволило спростити процес локального розгортання системи та забезпечити швидку перевірку працездатності реалізованих функцій під час розробки.

Для наочного представлення обраних технологій доцільно використати таблицю 2.3.

Технологія	Призначення
PHP	Реалізація серверної логіки системи
SQLite	Зберігання даних та робота з базою даних
HTML	Формування структури вебсторінок
CSS	Оформлення та стилізація інтерфейсу
Bootstrap	Створення адаптивного користувацького інтерфейсу
Visual Studio Code	Середовище розробки програмного продукту
PHP Built-in Server	Локальний запуск та тестування системи

Таблиця 2.3 – Технології та засоби реалізації системи

Наведені в таблиці технології утворюють цілісний програмно-технологічний комплекс, необхідний для реалізації всіх функціональних можливостей Military CMS. Кожен із використаних інструментів виконує власну роль у процесі розробки та забезпечує взаємодію між окремими компонентами програмного продукту.

Важливою перевагою обраного набору засобів є їх сумісність між собою та відсутність потреби у використанні складних додаткових програмних платформ. Це дозволяє зменшити витрати часу на налаштування середовища розробки та спрощує процес подальшого супроводу системи.

Крім того, усі використані технології є широко розповсюдженими у сфері веброзробки та мають активну підтримку з боку професійної спільноти. Завдяки цьому забезпечується доступність технічної документації, навчальних матеріалів та готових рішень для реалізації додаткового функціоналу в майбутньому.

Проведений аналіз показав, що обраний набір технологій повністю відповідає вимогам проекту та забезпечує можливість реалізації всіх функціональних можливостей Military CMS. Використання PHP, SQLite, HTML, CSS та Bootstrap дозволяє створити повноцінну систему управління

контентом із мінімальними вимогами до середовища розгортання та зручним користувацьким інтерфейсом.

Таким чином, обрані технології та засоби реалізації створюють необхідне технічне підґрунтя для практичної розробки програмного продукту та забезпечують можливість ефективного функціонування Military CMS у межах поставлених завдань.

Висновки до розділу 2

У другому розділі кваліфікаційної роботи виконано проектування системи управління контентом Military CMS, призначеної для використання у військовій сфері. На основі вимог, сформованих у попередньому розділі, визначено структуру програмного продукту, спроектовано його архітектуру, модель зберігання даних та механізми розмежування доступу користувачів до функціональних можливостей системи.

У процесі проектування було обрано модульний підхід до побудови програмного забезпечення, що дозволив розділити функціональність системи на окремі логічно пов'язані компоненти. До складу Military CMS включено модуль авторизації користувачів, модуль керування користувачами, модуль керування інформаційними матеріалами, модуль керування категоріями, модуль журналювання подій та головну панель керування. Запропонована архітектура забезпечує зручну організацію програмного коду, спрощує подальший супровід системи та створює передумови для її подальшого розвитку.

Особливу увагу було приділено проектуванню структури бази даних. У результаті визначено основні сутності предметної області, до яких належать користувачі, інформаційні матеріали, категорії та журнал подій. Для кожної сутності сформовано відповідну структуру таблиць та встановлено логічні зв'язки між ними. Використання реляційної моделі даних дозволяє забезпечити цілісність інформації, мінімізувати дублювання даних та підтримувати коректну роботу всіх функціональних модулів системи.

У межах проектування механізмів доступу реалізовано рольову модель, яка передбачає використання ролей адміністратора, редактора та переглядача. Для кожної категорії користувачів визначено перелік дозволених операцій та обмежень. Такий підхід забезпечує контрольований доступ до інформаційних ресурсів, сприяє підвищенню рівня інформаційної безпеки та дозволяє реалізувати принцип мінімально необхідних привілеїв.

Також у розділі було розроблено структуру користувацького інтерфейсу та визначено основні принципи взаємодії користувача із системою. Запропонована організація інтерфейсу орієнтована на забезпечення простоти використання, швидкого доступу до функціональних можливостей програмного продукту та комфортної роботи користувачів із різним рівнем технічної підготовки.

Крім того, обґрунтовано вибір технологій, використаних для реалізації Military CMS. Застосування мови програмування PHP, системи керування базами даних SQLite та сучасних вебтехнологій дозволяє створити повноцінний програмний продукт із мінімальними вимогами до середовища розгортання та подальшого супроводу.

Отримані результати проектування формують необхідне підґрунтя для практичної реалізації системи управління контентом Military CMS. Розроблені архітектурні рішення, структура бази даних та механізми контролю доступу будуть використані на наступному етапі роботи під час програмної реалізації та тестування функціональних можливостей системи.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ СИСТЕМИ УПРАВЛІННЯ КОНТЕНТОМ ДЛЯ ВІЙСЬКОВОЇ СФЕРИ

3.1 Реалізація структури бази даних

Після завершення етапу проєктування наступним кроком стала практична реалізація системи управління контентом Military CMS. Одним із перших завдань було створення структури бази даних, яка забезпечує зберігання всієї інформації, необхідної для функціонування програмного продукту. Сама база даних виступає центральним компонентом системи, оскільки через неї здійснюється взаємодія між окремими програмними модулями та забезпечується збереження інформації незалежно від роботи користувацького інтерфейсу.

Для реалізації сховища даних було використано систему керування базами даних SQLite. Вибір даного рішення обумовлений його простотою, відсутністю необхідності встановлення окремого серверного програмного забезпечення та зручністю використання під час розробки вебзастосунків невеликого та середнього масштабу. На відміну від серверних СУБД, SQLite зберігає всю інформацію у вигляді одного файлу, що значно спрощує процес розгортання системи та резервного копіювання даних[25].

Структура бази даних була реалізована відповідно до моделі, розробленої на етапі проєктування. До її складу увійшли чотири основні таблиці: users, articles, categories та logs. Кожна таблиця відповідає окремій сутності предметної області та виконує визначене функціональне призначення.

Таблиця users призначена для зберігання інформації про користувачів системи. У ній містяться відомості про повне ім'я користувача, логін, пароль, роль доступу та дату створення облікового запису. Особливістю реалізації є використання унікального логіна для кожного користувача, що унеможливорює створення двох облікових записів з однаковими даними авторизації.

Поле	Тип даних	Призначення
id	INTEGER	Унікальний ідентифікатор користувача
full_name	TEXT	Повне ім'я користувача
username	TEXT	Логін користувача
password	TEXT	Хешований пароль користувача
role	TEXT	Роль користувача в системі
created_at	TEXT	Дата створення облікового запису

Таблиця 3.1 – Структура таблиці users

Таблиця categories використовується для зберігання інформації про категорії матеріалів. Категоризація контенту дозволяє впорядковувати інформаційні ресурси відповідно до їх тематики та спрощує подальший пошук необхідних матеріалів. Для кожної категорії зберігається її назва, опис та дата створення.

Поле	Тип даних	Призначення
id	INTEGER	Унікальний ідентифікатор категорії
name	TEXT	Назва категорії
description	TEXT	Опис категорії
created_at	TEXT	Дата створення категорії

Таблиця 3.2 – Структура таблиці categories

Основним сховищем інформаційних матеріалів виступає таблиця articles. У ній містяться відомості про назву матеріалу, текстовий вміст, категорію, автора, статус публікації та часові мітки створення і редагування запису. Використання зовнішніх ключів дозволяє пов'язати матеріал із конкретним користувачем та категорією, що забезпечує цілісність даних і спрощує виконання запитів.

Поле	Тип даних	Призначення
id	INTEGER	Унікальний ідентифікатор матеріалу
title	TEXT	Назва матеріалу
content	TEXT	Текстовий вміст матеріалу
category_id	INTEGER	Посилання на категорію
author_id	INTEGER	Посилання на автора матеріалу
status	TEXT	Статус матеріалу (draft або published)
created_at	TEXT	Дата створення матеріалу
updated_at	TEXT	Дата останнього редагування

Таблиця 3.3 – Структура таблиці articles

Для реалізації механізму аудиту дій користувачів була створена таблиця logs. У ній автоматично фіксуються основні операції, виконані в системі. Кожний запис містить ідентифікатор користувача, опис виконаної дії та дату її здійснення. Наявність журналу подій дозволяє контролювати використання системи та відстежувати історію внесених змін.

Поле	Тип даних	Призначення
id	INTEGER	Унікальний ідентифікатор запису
user_id	INTEGER	Посилання на користувача
action	TEXT	Опис виконаної дії
created_at	TEXT	Дата та час виконання дії

Таблиця 3.4 – Структура таблиці logs

Для створення таблиць використовувався SQL-скрипт ініціалізації бази даних, який автоматично формує необхідну структуру під час першого запуску системи. Такий підхід дозволяє значно спростити розгортання програмного продукту та виключає необхідність ручного створення таблиць адміністратором.

```

$pdo->exec("
CREATE TABLE IF NOT EXISTS users (
id INTEGER PRIMARY KEY AUTOINCREMENT,
full_name TEXT NOT NULL,
username TEXT NOT NULL UNIQUE,
password TEXT NOT NULL,
role TEXT NOT NULL,
created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP
)
");

$pdo->exec("
CREATE TABLE IF NOT EXISTS categories (
id INTEGER PRIMARY KEY AUTOINCREMENT,
name TEXT NOT NULL UNIQUE,
description TEXT,
created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP
)
");

$pdo->exec("
CREATE TABLE IF NOT EXISTS articles (
id INTEGER PRIMARY KEY AUTOINCREMENT,
title TEXT NOT NULL,
content TEXT NOT NULL,
category_id INTEGER NOT NULL,
author_id INTEGER NOT NULL,
status TEXT NOT NULL DEFAULT 'draft',
created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
updated_at TEXT
)
");

```

Лістинг 3.1 – Фрагмент SQL-коду створення таблиць бази даних

У результаті виконаної роботи було сформовано повноцінну структуру бази даних, яка забезпечує зберігання інформації про користувачів, матеріали, категорії та журнал подій. Реалізована модель даних стала основою для подальшого створення функціональних модулів Military CMS та забезпечила можливість реалізації всіх передбачених проектом функцій.

3.2 Реалізація модуля авторизації та автентифікації користувачів

Одним із найважливіших компонентів системи Military CMS є модуль авторизації та автентифікації користувачів. Його основне призначення полягає у перевірці облікових даних користувача та забезпеченні доступу до

функціональних можливостей системи відповідно до встановлених прав доступу. Реалізація даного модуля дозволяє захистити інформаційні ресурси від несанкціонованого використання та забезпечує контроль доступу до адміністративних функцій програмного продукту.

Механізми автентифікації та авторизації є обов'язковими складовими сучасних веборієнтованих інформаційних систем. Їх основним завданням є ідентифікація користувача та визначення рівня його повноважень після входу до програмного продукту. Реалізація таких механізмів дозволяє забезпечити контроль доступу до інформаційних ресурсів та мінімізувати ризик несанкціонованого використання функціональних можливостей системи.

Під час розробки Military CMS особлива увага приділялася забезпеченню безпеки процедури входу до системи. Для цього було реалізовано перевірку облікових даних користувача на стороні сервера та використано механізми захищеного зберігання паролів. Такий підхід дозволяє запобігти прямому збереженню конфіденційної інформації в базі даних та підвищує загальний рівень захищеності програмного продукту.

Для входу до системи користувач повинен ввести логін та пароль у спеціальній формі авторизації. Після надсилання форми введені дані передаються на сервер для подальшої перевірки. Інтерфейс сторінки авторизації містить лише необхідні елементи керування, що спрощує взаємодію користувача із системою та забезпечує швидкий доступ до процедури входу.



Military CMS

Система управління контентом для військової сфери

Логін

admin

Пароль

Увійти

Тестовий доступ: admin / admin123

Рисунок 3.1 – Введення облікових даних користувача

Сторінка авторизації реалізована у максимально спрощеному вигляді та містить лише необхідні поля для введення логіна та пароля. Відсутність другорядних елементів дозволяє зосередити увагу користувача на процедурі входу та забезпечує швидкий доступ до функціональних можливостей системи після успішної автентифікації.

Після отримання введених даних система виконує пошук користувача в таблиці users та перевіряє відповідність введеного пароля його збереженому хешованому значенню. Для забезпечення безпеки використовуються вбудовані механізми PHP для хешування та перевірки паролів. У випадку успішної перевірки створюється користувацька сесія, яка використовується для подальшої ідентифікації користувача під час роботи із системою.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {  
    $username = trim($_POST['username'] ?? "");  
    $password = trim($_POST['password'] ?? "");  
  
    if ($username === "" || $password === "") {
```

```
$error = 'Будь ласка, заповніть усі поля.';
} else {
    $pdo = Database::connect();

    $stmt = $pdo->prepare("SELECT * FROM users WHERE username = :username LIMIT 1");
    $stmt->execute([
        ':username' => $username
    ]);

    $user = $stmt->fetch();

    if ($user && password_verify($password, $user['password'])) {
        $_SESSION['user'] = [
            'id' => $user['id'],
            'full_name' => $user['full_name'],
            'username' => $user['username'],
            'role' => $user['role']
        ];

        header("Location: dashboard.php");
        exit;
    } else {
        $error = 'Неправильний логін або пароль.';
    }
}
}
```

Лістинг 3.2 - Фрагмент коду перевірки облікових даних

Наведений фрагмент програмного коду демонструє реалізацію основної логіки перевірки облікових даних користувача. Спочатку система отримує значення логіна та пароля із форми авторизації, після чого виконує пошук відповідного користувача в базі даних. Якщо обліковий запис знайдено, здійснюється перевірка відповідності введеного пароля його хешованому значенню за допомогою вбудованої функції `password_verify()`.

Використання механізму хешування дозволяє уникнути зберігання паролів у відкритому вигляді та відповідає сучасним вимогам інформаційної безпеки. Після успішної перевірки створюється користувацька сесія, яка використовується для подальшої ідентифікації користувача під час роботи із системою.

Після успішної авторизації система зберігає інформацію про поточного користувача та його роль у сесії. Це дозволяє реалізувати механізм розмежування доступу та надавати різним категоріям користувачів лише ті функціональні можливості, які відповідають їхнім повноваженням. Адміністратор отримує доступ до всіх модулів системи, редактор може працювати з контентом і категоріями, а користувач із роллю переглядача має доступ лише до перегляду інформаційних матеріалів.

Важливою складовою роботи модуля є обробка помилок авторизації. У випадку введення неправильного логіна або пароля система не надає доступ до ресурсів Military CMS та відображає відповідне повідомлення про помилку. Такий підхід дозволяє інформувати користувача про невдалу спробу входу та запобігає несанкціонованому доступу до системи.



The image shows a login interface for 'Military CMS'. At the top, the title 'Military CMS' is displayed in a large, bold font, followed by the subtitle 'Система управління контентом для військової сфери'. Below this, a pink error message box contains the text 'Неправильний логін або пароль.'. Underneath the error message, there are two input fields: 'Логін' (Login) with the value 'admin' and 'Пароль' (Password) with the placeholder text 'Введіть пароль'. A dark green button labeled 'Увійти' (Login) is positioned below the password field. At the bottom of the form, there is a line of text: 'Тестовий доступ: admin / admin123'.

Рисунок 3.2 – Повідомлення про помилку авторизації

Реалізація механізму повідомлення про помилки є важливою складовою користувацького інтерфейсу. Відображення зрозумілого повідомлення дозволяє користувачу швидко зрозуміти причину відмови у доступі та виконати повторну спробу входу із правильними обліковими даними. При цьому система не розкриває додаткової інформації щодо конкретної причини помилки, що сприяє підвищенню рівня безпеки.

Для завершення роботи передбачено функцію виходу із системи. Після виконання даної операції поточна користувацька сесія видаляється, а доступ до захищених сторінок блокується до моменту повторного проходження процедури авторизації. Завдяки цьому забезпечується додатковий рівень захисту інформації та знижується ризик використання системи сторонніми особами.

Окремою перевагою реалізованого підходу є можливість подальшого розширення механізмів безпеки. За необхідності до системи можуть бути інтегровані додаткові засоби захисту, зокрема двофакторна автентифікація, обмеження кількості невдалих спроб входу або механізми автоматичного блокування облікового запису після підозрілої активності. Поточна архітектура модуля дозволяє реалізувати такі вдосконалення без суттєвої зміни вже існуючого програмного коду.

У результаті реалізації було створено повноцінний модуль авторизації та автентифікації користувачів, який забезпечує перевірку облікових даних, підтримує рольову модель доступу, здійснює керування користувацькими сесіями та гарантує захист функціональних можливостей Military CMS від несанкціонованого використання.

3.3 Реалізація модуля керування контентом

Основним призначенням системи Military CMS є забезпечення централізованого керування інформаційними матеріалами. Саме тому модуль керування контентом є центральним компонентом програмного продукту та реалізує ключову частину його функціональних можливостей. Даний модуль

забезпечує створення, редагування, перегляд, пошук та видалення інформаційних матеріалів, а також взаємодію з категоріями та механізмами контролю доступу.

Під час реалізації модуля було використано принцип розподілу функціональності між окремими сторінками системи. Такий підхід дозволяє спростити підтримку програмного забезпечення, підвищити читабельність програмного коду та забезпечити логічну організацію роботи з контентом. Кожна сторінка відповідає за виконання конкретної операції та взаємодіє з базою даних через серверну частину системи.

Після успішної авторизації користувач отримує доступ до сторінки перегляду матеріалів. На цій сторінці відображається перелік усіх записів, що зберігаються в системі. Для кожного матеріалу відображаються основні відомості, зокрема назва, категорія, автор та дата створення. Представлення інформації у вигляді таблиці дозволяє швидко переглядати великі обсяги даних та спрощує навігацію між записами.

ID	Назва	Категорія	Автор	Статус	Дата створення	Дії
7	Відкрито реєстрацію на курс з підвищення кваліфікації у сфері IT	Навчальні матеріали	Test Editor	Опубліковано	2026-06-12 15:43:01	Редагувати Видалити
6	Організаційне засідання щодо правил пожежної безпеки	Спогодження	Test Editor	Опубліковано	2026-06-12 15:41:36	Редагувати Видалити
5	Важлива інформація	Накази	System Administrator	Опубліковано	2026-06-12 15:35:41	Редагувати Видалити
1	Оновлення інформаційного розділу	Новини	System Administrator	Опубліковано	2026-06-11 14:22:41	Редагувати Видалити
2	Інструкція щодо роботи з CMS	Інструкції	System Administrator	Чернетка	2026-06-11 14:22:41	Редагувати Видалити

Рисунок 3.3 – Список інформаційних матеріалів

Однією з найважливіших функцій модуля є створення нових матеріалів. Для реалізації цієї можливості була розроблена спеціальна форма введення даних, що дозволяє користувачеві вказати назву матеріалу, його зміст, категорію та статус публікації. Після заповнення всіх необхідних полів

система виконує перевірку введених даних та зберігає інформацію до бази даних.



Додати матеріал
Створення нового інформаційного матеріалу CMS

Назва матеріалу

Категорія
Оберіть категорію

Текст матеріалу

Статус
Чернетка

Рисунок 3.4 – Форма створення нового матеріалу

Під час розробки особлива увага приділялася коректності введених даних. Для цього було реалізовано перевірку обов'язкових полів форми перед виконанням операції збереження. У випадку відсутності необхідної інформації користувач отримує відповідне повідомлення про помилку, що запобігає створенню неповних або некоректних записів.

Додати матеріал

Створення нового інформаційного матеріалу CMS

Будь ласка, заповніть усі обов'язкові поля.

Назва матеріалу

Категорія

Оберіть категорію

Текст матеріалу

Статус

Рисунок 3.5 – Приклад повідомлення про помилку

Для додавання матеріалу до системи використовується SQL-запит вставки нового запису до таблиці articles. Серверна частина отримує введені користувачем дані, обробляє їх та виконує відповідну операцію в базі даних.

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $title = trim($_POST['title'] ?? '');
    $content = trim($_POST['content'] ?? '');
    $categoryId = $_POST['category_id'] ?? '';
    $status = $_POST['status'] ?? 'draft';

    if ($title === '' || $content === '' || $categoryId === '') {
        $error = 'Будь ласка, заповніть усі обов'язкові поля.';
    } else {
        $stmt = $pdo->prepare("
            INSERT INTO articles (title, content, category_id, author_id, status)
            VALUES (:title, :content, :category_id, :author_id, :status)
        ");

        $stmt->execute([
            'title' => $title,
            'content' => $content,
            'category_id' => $categoryId,
            'author_id' => $_SESSION['user']['id'],
            'status' => $status
        ]);

        $logStmt = $pdo->prepare("
            INSERT INTO logs (user_id, action)
            VALUES (:user_id, :action)
        ");
    }
}

```

```
$logStmt->execute([
    'user_id' => $_SESSION['user']['id'],
    'action' => 'Створено матеріал: ' . $title
]);

header('Location: articles.php');
exit;
}
```

Лістинг 3.3 – Фрагмент коду додавання матеріалу

Не менш важливою функцією є редагування існуючих записів. Після вибору необхідного матеріалу користувач отримує доступ до форми редагування, яка автоматично заповнюється поточними даними. Такий підхід дозволяє змінювати лише необхідні параметри без повторного введення всієї інформації. Після збереження змін система оновлює відповідний запис у базі даних та фіксує виконану операцію в журналі подій.



Редагувати матеріал
Оновлення інформаційного матеріалу CMS

Назва матеріалу
Відкрито реєстрацію на курс з підвищення кваліфікації у сфері ІТ

Категорія
Навчальні матеріали

Текст матеріалу
Для військовослужбовців та цивільного персоналу відкривається безкоштовний доступ до базового онлайн-курсу з мережевої безпеки.

Статус
Опубліковано

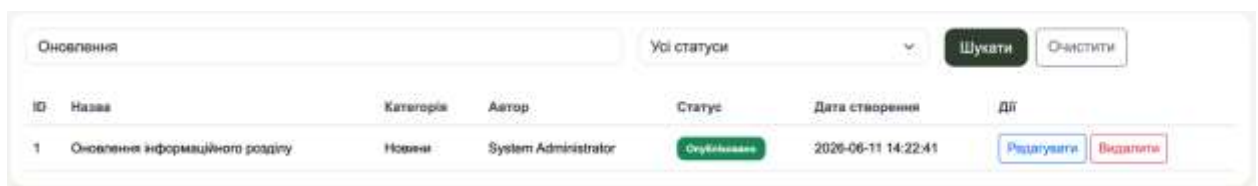
Зберегти зміни Назад

Рисунок 3.6 – Редагування матеріалу

Для підтримання актуальності інформаційних ресурсів реалізовано функцію видалення матеріалів. Доступ до цієї операції мають лише

користувачі з відповідними правами. Перед видаленням система може запитувати підтвердження виконання дії, що дозволяє уникнути випадкової втрати інформації.

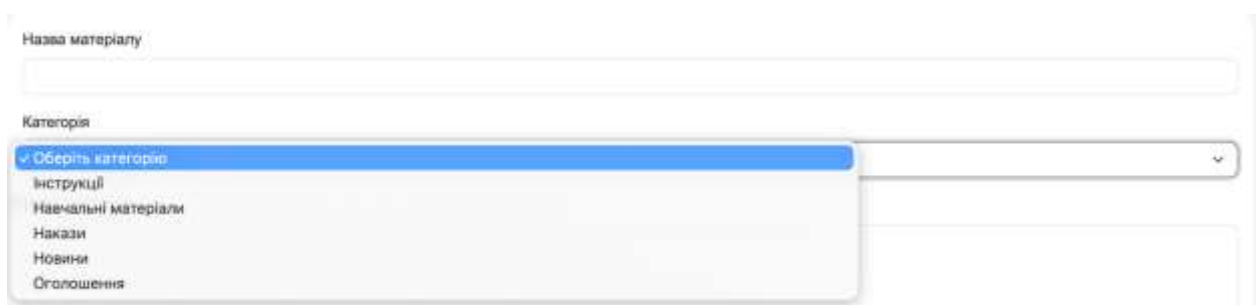
Окрему увагу приділено реалізації механізму пошуку. Зі збільшенням кількості матеріалів швидкий пошук інформації стає важливою складовою ефективною роботи системи. Реалізований механізм дозволяє здійснювати пошук за назвою матеріалу та швидко знаходити необхідні записи без перегляду повного списку контенту.



Оновлення	Усі статуси		Шукати	Очистити		
ID	Назва	Категорія	Автор	Статус	Дата створення	Дії
1	Оновлення інформаційного розділу	Новини	System Administrator	Одбувається	2026-06-11 14:22:41	Редагувати Видалити

Рисунок 3.7 – Результат роботи пошуку матеріалів

Додатковою перевагою реалізованого модуля є його інтеграція з системою категорій. Під час створення або редагування матеріалу користувач може обрати необхідну категорію, що забезпечує логічну структуру інформаційних ресурсів та спрощує їх подальше адміністрування.



Назва матеріалу

Категорія

- Оберть категорію
- Інструкції
- Навчальні матеріали
- Накази
- Новини
- Оголошення

Рисунок 3.8 – Приклад обрання необхідної категорії

Усі операції створення, редагування та видалення матеріалів автоматично фіксуються в журналі подій. Завдяки цьому забезпечується контроль за змінами в системі та можливість відстеження історії роботи користувачів із контентом.

У результаті реалізації модуля керування контентом було створено повноцінний механізм роботи з інформаційними матеріалами, який забезпечує виконання основних операцій над даними, підтримує взаємодію з іншими компонентами системи та відповідає функціональним вимогам, сформованим на етапі проєктування Military CMS.

3.4 Реалізація модуля керування користувачами та журналом подій

Важливою складовою системи Military CMS є модуль керування користувачами, який забезпечує адміністрування облікових записів та контроль доступу до функціональних можливостей програмного продукту. Реалізація даного модуля дозволяє централізовано керувати користувачами системи та забезпечувати виконання вимог щодо інформаційної безпеки.

У процесі розробки було передбачено, що доступ до функцій керування користувачами має виключно адміністратор системи. Перед відкриттям відповідних сторінок система виконує перевірку ролі поточного користувача та блокує доступ у випадку відсутності необхідних повноважень. Такий підхід дозволяє запобігти несанкціонованим змінам облікових записів та забезпечує централізований контроль доступу до ресурсів Military CMS.

Однією з основних функцій модуля є створення нових користувачів. Для цього реалізовано спеціальну форму введення даних, що дозволяє адміністратору вказати ім'я користувача, логін, пароль та роль доступу. Після перевірки коректності введених даних інформація зберігається в таблиці users, а новий користувач отримує можливість авторизації в системі.

Додати користувача

Створення нового облікового запису CMS

ПІБ користувача
User 1

Логін
user

Пароль

Роль
Переглядач

Зберегти користувача Назад

Рисунок 3.9 – Заповнення форми створення нового користувача

Для забезпечення цілісності даних реалізовано перевірку унікальності логіна. Якщо користувач із вказаним логіном уже існує в системі, створення нового облікового запису не виконується, а адміністратор отримує відповідне повідомлення. Такий механізм запобігає дублюванню облікових записів та підтримує коректність роботи системи авторизації.

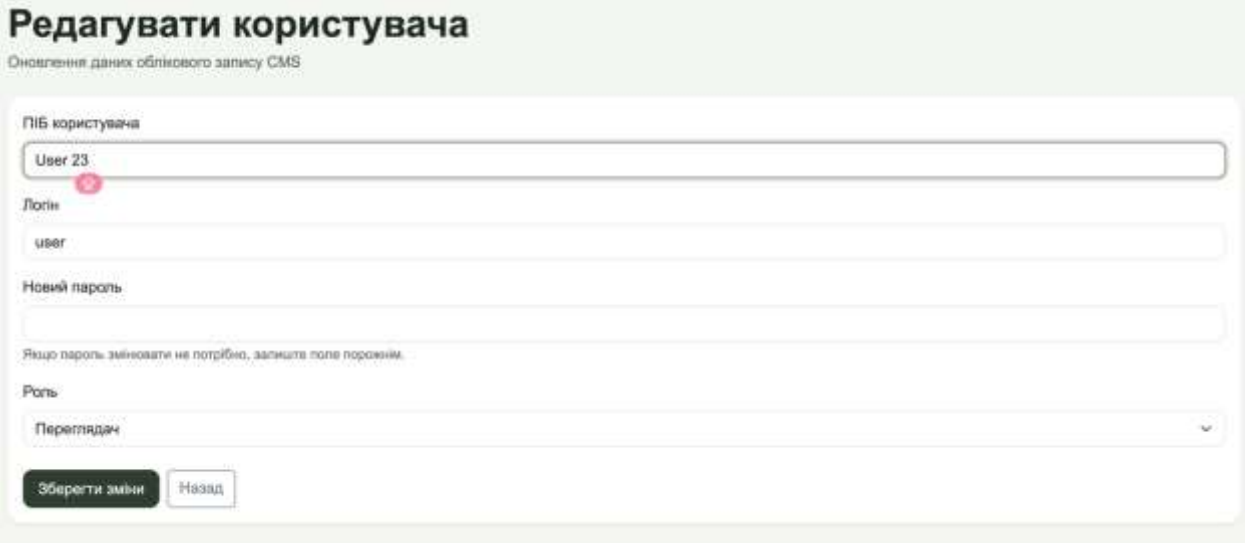
Фрагмент програмного коду, що реалізує процедуру додавання нового користувача до бази даних, наведено в лістингу 3.4.

```
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $stmt = $pdo->prepare("
        INSERT INTO users (full_name, username, password, role)
        VALUES (:full_name, :username, :password, :role)
    ");
    $stmt->execute([
        ':full_name' => $fullName,
        ':username' => $username,
        ':password' => password_hash($password, PASSWORD_DEFAULT),
        ':role' => $role
    ]);
    $logStmt = $pdo->prepare("
        INSERT INTO logs (user_id, action)
        VALUES (:user_id, :action)
    ");
    $logStmt->execute([
```

```
'user_id' => $_SESSION['user']['id'],  
'action' => 'Створено користувача: ' . $username  
]);  
}
```

Лістинг 3.4 – Фрагмент коду додавання користувача

Окрім створення нових облікових записів, адміністратор має можливість редагувати інформацію про вже зареєстрованих користувачів. Реалізований механізм дозволяє змінювати ім'я користувача, логін та роль доступу відповідно до поточних потреб організації. Усі внесені зміни автоматично зберігаються в базі даних та стають доступними відразу після оновлення запису.



Редагувати користувача
Оновлення даних облікового запису CMS

ПІБ користувача
User 23

Логін
user

Новий пароль

Якщо пароль змінювати не потрібно, залиште поле порожнім.

Роль
Переглядач

Зберегти зміни Назад

Рисунок 3.10 – Редагування облікового запису користувача

Додатково в системі реалізовано механізм зміни пароля. Для підвищення рівня безпеки нові паролі не зберігаються у відкритому вигляді, а проходять процедуру хешування перед записом до бази даних. Використання вбудованих засобів РНР забезпечує належний рівень захисту персональних даних користувачів та знижує ризик компрометації облікових записів.

Важливим компонентом Military CMS є журнал подій, який виконує функцію аудиту дій користувачів. Під час роботи системи автоматично фіксуються операції створення, редагування та видалення інформації. Кожний запис журналу містить відомості про користувача, виконану дію та час її здійснення.

Для зберігання журналу використовується окрема таблиця logs, що забезпечує накопичення історії роботи системи та дозволяє контролювати всі критично важливі зміни. Такий підхід підвищує прозорість використання програмного забезпечення та забезпечує додатковий рівень контролю за роботою користувачів.

ID	Користувач	Логін	Дія	Дата і час
29	System Administrator	admin	Створено користувача: user	2026-06-12 17:07:56
28	System Administrator	admin	Створено користувача: user	2026-06-12 17:03:12

Рисунок 3.11 – Запис про виконану дію в журналі подій

Реалізований механізм журналювання працює автоматично та не потребує додаткових дій з боку користувача. Після виконання визначених операцій система самостійно створює відповідний запис у журналі та зберігає його до бази даних. Завдяки цьому забезпечується можливість подальшого аналізу дій користувачів та контролю змін, внесених до системи.

У результаті виконаної роботи було реалізовано повноцінний модуль керування користувачами та журналом подій, який забезпечує створення та редагування облікових записів, підтримку рольової моделі доступу, безпечно зберігання паролів і аудит дій користувачів у межах системи Military CMS.

3.5 Тестування програмного засобу

Після завершення реалізації основних функціональних модулів Military CMS було проведено тестування програмного засобу з метою перевірки

коректності його роботи та відповідності вимогам, сформованим на етапі проєктування. Проведення тестування дозволяє виявити можливі помилки, оцінити стабільність функціонування системи та підтвердити працездатність реалізованих механізмів.

Тестування виконувалося в локальному середовищі розробки з використанням вбудованого вебсервера PHP та бази даних SQLite. Під час перевірки особлива увага приділялася роботі механізмів авторизації, коректності функціонування рольової моделі доступу, роботі модулів керування контентом, категоріями та користувачами, а також журналу подій.

Одним із перших етапів тестування стала перевірка процедури авторизації користувачів. Було перевірено можливість входу до системи з коректними обліковими даними, а також поведінку програмного забезпечення при введенні неправильного логіна або пароля. Результати перевірки підтвердили правильність роботи механізму автентифікації та коректне відображення повідомлень про помилки у випадку введення некоректних даних.

Наступним етапом стала перевірка рольової моделі доступу. Для кожної ролі користувача було протестовано набір доступних функцій. Адміністратор отримав доступ до всіх модулів системи, редактор — до функцій роботи з контентом і категоріями, а переглядач — лише до функцій перегляду та пошуку матеріалів. Результати тестування підтвердили коректність реалізованого механізму розмежування доступу.

Окрему увагу було приділено перевірці модуля керування контентом. Під час тестування здійснювалися операції створення, редагування, пошуку та видалення інформаційних матеріалів. Усі операції виконувалися коректно, а зміни своєчасно відображалися в базі даних. Також було перевірено взаємодію матеріалів із категоріями та правильність відображення інформації в користувацькому інтерфейсі.

Додатково проведено тестування модуля керування користувачами. Перевірено процедури створення нових облікових записів, зміну ролей

доступу, редагування даних користувачів та перевірку унікальності логінів. У ході тестування не було виявлено критичних помилок, які могли б вплинути на працездатність системи.

Перевірка журналу подій показала, що всі основні дії користувачів автоматично фіксуються в таблиці logs. Після створення, редагування або видалення інформаційних матеріалів система формувала відповідні записи журналу, що підтверджує коректність роботи механізму аудиту.

Результати виконаного тестування наведено в таблиці 3.1.

№	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
1	Авторизація з правильними даними	Успішний вхід до системи	Вхід виконано	Успішно
2	Авторизація з неправильним паролем	Повідомлення про помилку	Повідомлення відображено	Успішно
3	Створення нового матеріалу	Матеріал додається до БД	Матеріал створено	Успішно
4	Редагування матеріалу	Дані оновлюються	Дані оновлено	Успішно
5	Видалення матеріалу	Запис видаляється	Запис видалено	Успішно
6	Створення категорії	Категорія додається	Категорію створено	Успішно
7	Створення користувача	Користувач додається	Користувача створено	Успішно
8	Перевірка ролі редактора	Відсутній доступ до користувачів	Доступ заблоковано	Успішно
9	Перевірка ролі переглядача	Лише перегляд матеріалів	Обмеження працюють	Успішно
10	Перевірка журналу подій	Створюється запис у журналі	Запис створено	Успішно

Таблиця 3.5 - Результати тестування системи

Отримані результати свідчать про те, що реалізовані модулі Military CMS функціонують відповідно до поставлених вимог. У процесі тестування не було виявлено критичних помилок, які б унеможливили використання

програмного продукту. Це дозволяє зробити висновок про готовність системи до практичного використання та подальшого розвитку.

Висновки до розділу 3

У третьому розділі кваліфікаційної роботи виконано практичну реалізацію системи управління контентом Military CMS та проведено перевірку її працездатності. На основі розробленої архітектури було створено програмний продукт, який забезпечує централізоване керування інформаційними матеріалами, підтримує рольову модель доступу та надає засоби адміністрування контенту через вебінтерфейс.

У процесі реалізації сформовано структуру бази даних на основі SQLite, яка забезпечує зберігання інформації про користувачів, матеріали, категорії та журнал подій. Створена структура даних дозволяє підтримувати цілісність інформації та забезпечує коректну взаємодію між окремими модулями системи.

Було реалізовано модуль авторизації та автентифікації користувачів, який забезпечує перевірку облікових даних, керування користувацькими сесіями та контроль доступу до функціональних можливостей програмного продукту. Використання рольової моделі доступу дозволило розмежувати повноваження між адміністраторами, редакторами та користувачами з обмеженими правами.

У межах розробки створено модуль керування контентом, який підтримує операції створення, редагування, видалення та пошуку інформаційних матеріалів. Також реалізовано модуль керування користувачами та механізм журналювання подій, що забезпечує контроль за виконанням основних операцій у системі.

Проведене тестування підтвердило працездатність реалізованих компонентів Military CMS та їх відповідність функціональним вимогам, визначеним на етапі проєктування. Результати перевірки показали коректну роботу механізмів авторизації, розмежування доступу, керування контентом, адміністрування користувачів та ведення журналу подій.

Отримані результати свідчать про успішне виконання поставлених завдань та підтверджують можливість використання створеної системи для централізованого керування інформаційними ресурсами у військовій сфері.

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи було розглянуто питання розробки системи управління контентом для військової сфери. Актуальність обраної теми обумовлена зростанням обсягів інформаційних ресурсів, необхідністю їх централізованого адміністрування та підвищеними вимогами до контролю доступу в організаціях, діяльність яких пов'язана з обробкою службової інформації.

У ході дослідження було проведено аналіз предметної області систем управління контентом та розглянуто найбільш поширені програмні рішення, серед яких WordPress, Joomla та Drupal. Виконаний аналіз дозволив визначити їх основні переваги та недоліки, а також встановити, що універсальні CMS не завжди повною мірою враховують специфічні вимоги окремих предметних областей. На підставі отриманих результатів було обґрунтовано доцільність створення спеціалізованої системи управління контентом для військової сфери.

Під час виконання роботи сформовано функціональні та нефункціональні вимоги до програмного продукту, визначено основні принципи його функціонування та розроблено архітектуру системи. Було спроектовано структуру бази даних, механізми розмежування доступу між користувачами та користувацький інтерфейс програмного забезпечення. Також виконано моделювання основних компонентів системи за допомогою діаграм, що дозволило формалізувати структуру програмного продукту та взаємодію його складових.

Практичним результатом роботи стала розробка веборієнтованої системи Military CMS, реалізованої із використанням мови програмування PHP, системи керування базами даних SQLite та технологій HTML, CSS і Bootstrap. Створений програмний продукт забезпечує авторизацію користувачів, підтримує рольову модель доступу, дозволяє здійснювати керування

інформаційними матеріалами та категоріями, підтримує адміністрування користувачів і автоматичне ведення журналу подій.

У межах реалізації було створено модулі авторизації та автентифікації користувачів, керування контентом, керування категоріями, адміністрування облікових записів та журналювання дій. Розроблена структура бази даних забезпечує зберігання всієї необхідної інформації та підтримує коректну взаємодію між окремими компонентами системи.

Для перевірки працездатності програмного продукту проведено тестування реалізованих функцій. Отримані результати підтвердили коректність роботи механізмів авторизації, розмежування прав доступу, керування інформаційними матеріалами та адміністрування користувачів. У процесі тестування не було виявлено критичних помилок, які б унеможливили використання системи за призначенням.

Таким чином, поставлена мета роботи була досягнута, а визначені завдання виконані у повному обсязі. Розроблена система Military CMS забезпечує централізоване керування інформаційними ресурсами, підтримує контроль доступу до даних та може використовуватися як основа для подальшого розвитку спеціалізованих інформаційних систем військового призначення. Перспективним напрямом подальшого розвитку програмного продукту може стати впровадження розширених механізмів пошуку, системи сповіщень, багаторівневої системи погодження матеріалів та інтеграції з іншими інформаційними ресурсами.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

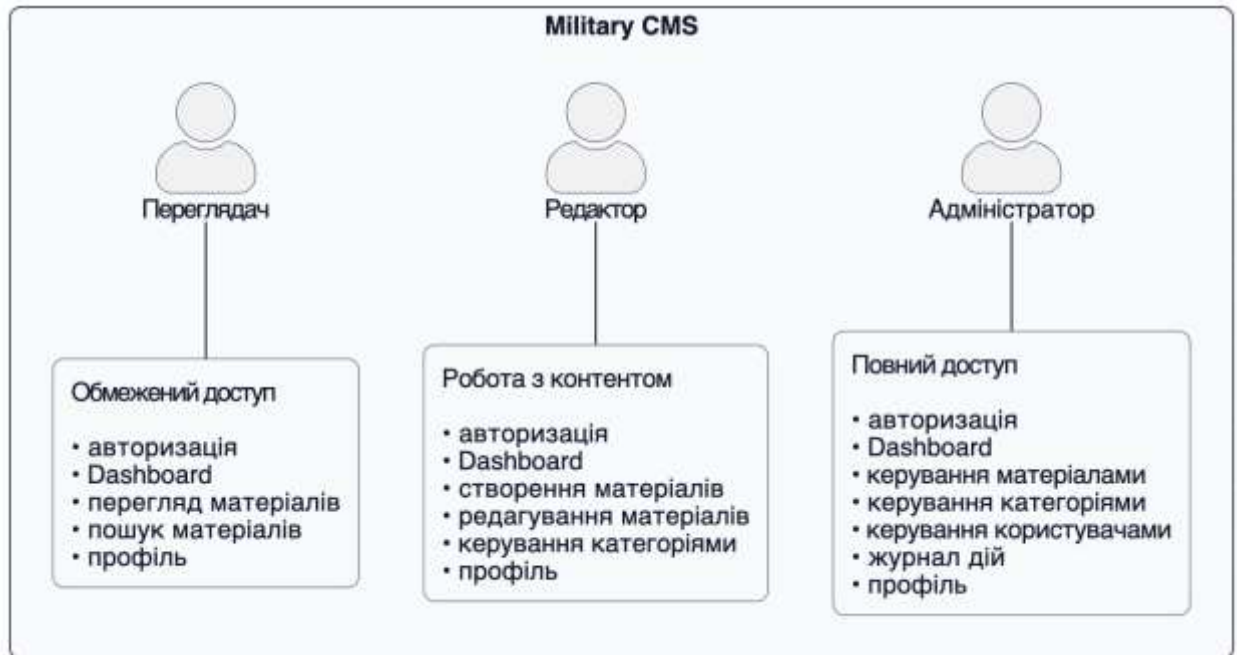
1. Басюк Т. М. Проектування інформаційних систем : навчальний посібник. Львів : Новий Світ-2000, 2021. 432 с.
2. Гужва В. М. Інформаційні системи і технології на підприємствах : навчальний посібник. Київ : КНЕУ, 2021. 400 с.
3. Копняк К. В. Інформаційні системи та технології : навчальний посібник. Вінниця : ВНТУ, 2020. 320 с.
4. Морзе Н. В. Основи інформаційних систем : підручник. Київ : Ліра-К, 2021. 384 с.
5. Павлов О. І. Проектування баз даних : навчальний посібник. Київ : Центр учбової літератури, 2020. 312 с.
6. Ситник В. Ф. Основи проектування інформаційних систем. Київ : КНЕУ, 2020. 540 с.
7. Тарасов В. О. Вебтехнології та вебдизайн : навчальний посібник. Київ : Каравела, 2021. 368 с.
8. Шаховська Н. Б. Організація баз даних та знань : навчальний посібник. Львів : Магнолія-2006, 2020. 296 с.
9. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 880 p.
10. Sommerville I. Software Engineering. 10th ed. Boston : Pearson, 2015. 816 p.
11. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2017. 432 p.
12. Martin R. C. Clean Code: A Handbook of Agile Software Craftsmanship. Boston : Prentice Hall, 2008. 464 p.
13. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3rd ed. Boston : Addison-Wesley Professional, 2003. 208 p.
14. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley Professional, 2002. 533 p.

- 15.Elmasri R., Navathe S. Fundamentals of Database Systems. 7th ed. Boston : Pearson, 2017. 1272 p.
- 16.Cornel C., Morris S. Database Systems: Design, Implementation and Management. 14th ed. Boston : Cengage Learning, 2022. 832 p.
- 17.Date C. J. An Introduction to Database Systems. 8th ed. Boston : Pearson, 2004. 1024 p.
- 18.Duckett J. PHP & MySQL. Indianapolis : Wiley, 2022. 928 p.
- 19.Nixon R. Learning PHP, MySQL & JavaScript. 6th ed. Sebastopol : O'Reilly Media, 2021. 832 p.
- 20.Welling L., Thomson L. PHP and MySQL Web Development. 5th ed. Boston : Addison-Wesley Professional, 2017. 912 p.
- 21.WordPress Documentation. URL: <https://wordpress.org/documentation> (дата звернення: 11.06.2026).
- 22.Joomla Documentation. URL: <https://docs.joomla.org> (дата звернення: 11.06.2026).
- 23.Drupal Documentation. URL: <https://www.drupal.org/documentation> (дата звернення: 11.06.2026).
- 24.PHP Documentation. URL: <https://www.php.net/docs.php> (дата звернення: 11.06.2026).
- 25.SQLite Documentation. URL: <https://www.sqlite.org/docs.html> (дата звернення: 11.06.2026).
- 26.Bootstrap Documentation. URL: <https://getbootstrap.com/docs> (дата звернення: 11.06.2026).
- 27.VisualStudio Code Documentation. URL: <https://code.visualstudio.com/docs> (дата звернення: 11.06.2026).
- 28.Mozilla Developer Network Web Docs. URL: <https://developer.mozilla.org> (дата звернення: 11.06.2026).
- 29.OWASP Foundation. Web Application Security Testing Guide. URL: <https://owasp.org> (дата звернення: 11.06.2026).

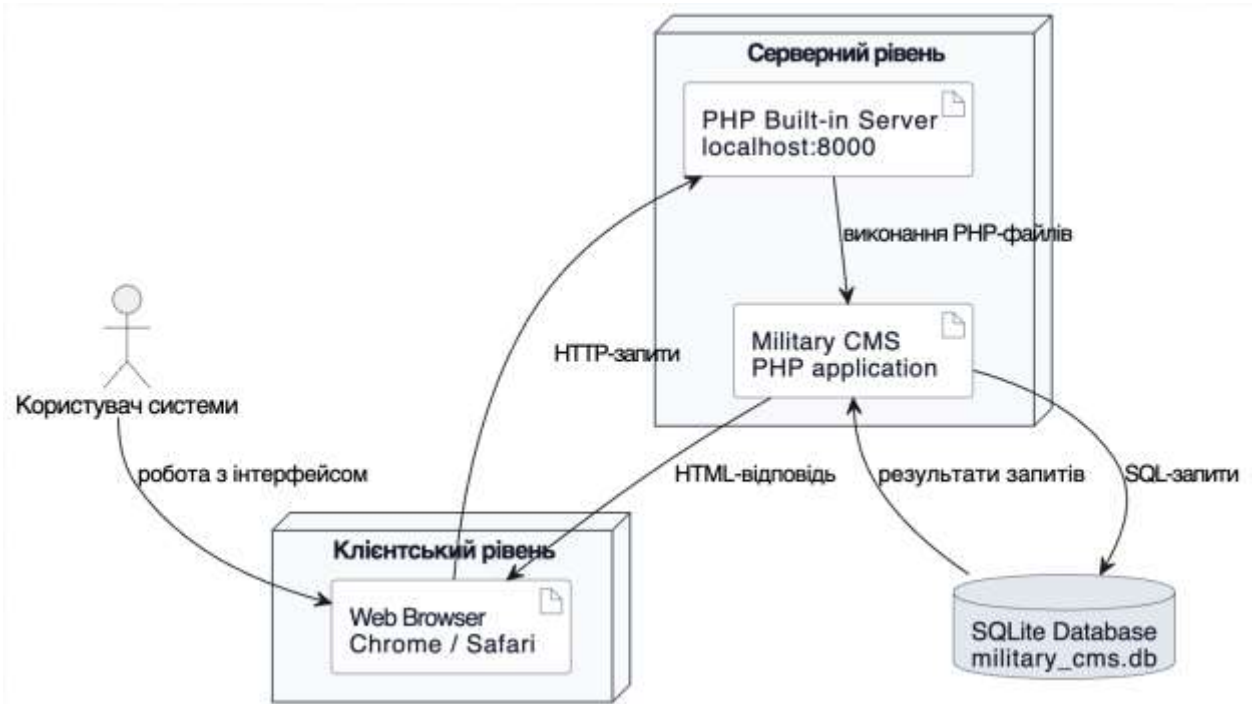
30. National Institute of Standards and Technology. Cybersecurity Framework. URL: <https://www.nist.gov/cyberframework> (дата звернення: 11.06.2026).
31. W3C HTML Standard. URL: <https://html.spec.whatwg.org> (дата звернення: 11.06.2026).
32. W3C Cascading Style Sheets (CSS). URL: <https://www.w3.org/Style/CSS> (дата звернення: 11.06.2026).
33. SQLite Tutorial. URL: <https://www.sqlitetutorial.net> (дата звернення: 11.06.2026).
34. PHP Manual. URL: <https://www.php.net/manual/en> (дата звернення: 11.06.2026).
35. Bootstrap Components Documentation. URL: <https://getbootstrap.com/docs/5.3/components> (дата звернення: 11.06.2026).

ДОДАТКИ

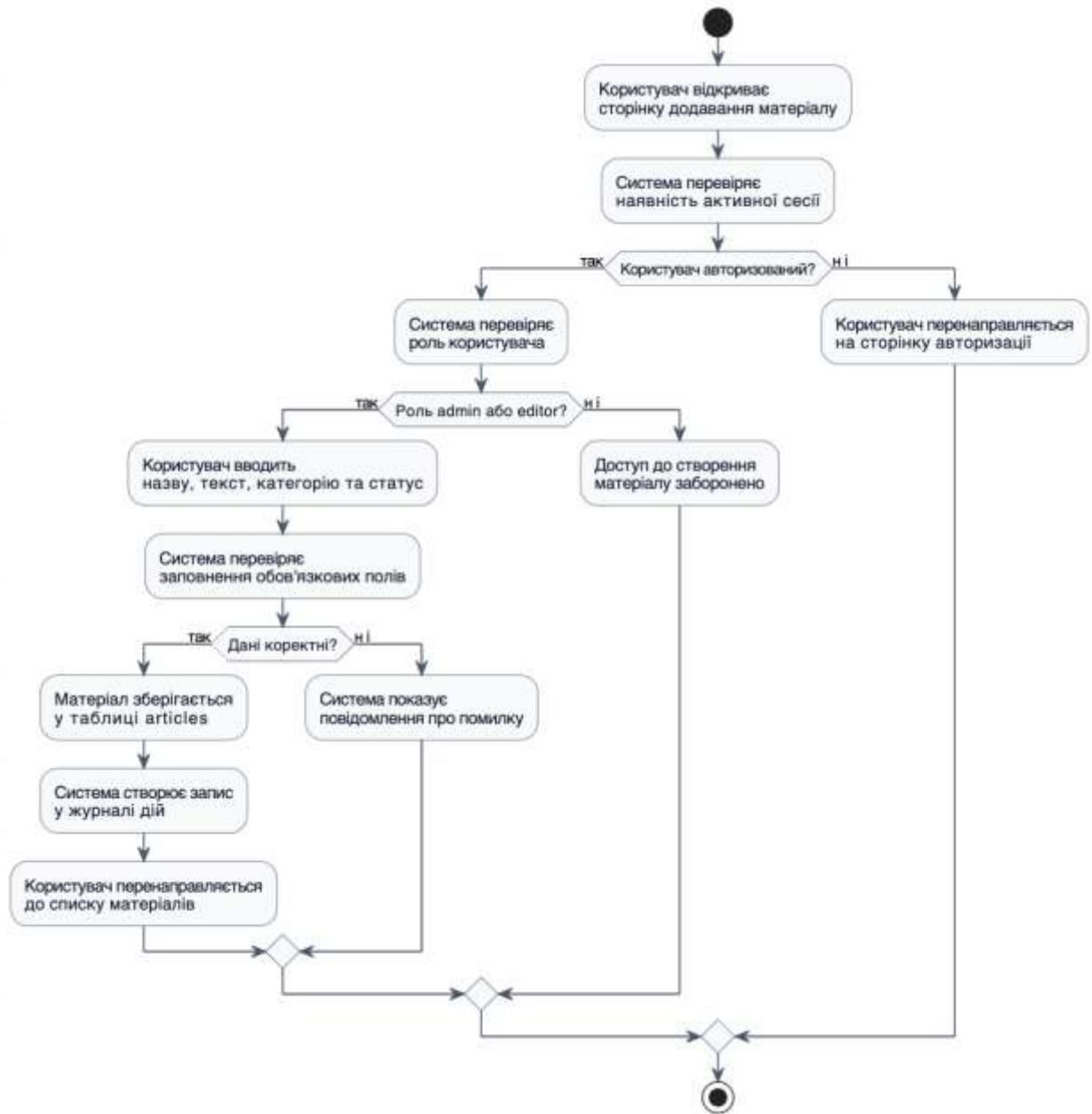
Додаток А - UML-діаграма варіантів використання системи Military CMS



Додаток Б - Схема архітектури системи Military CMS

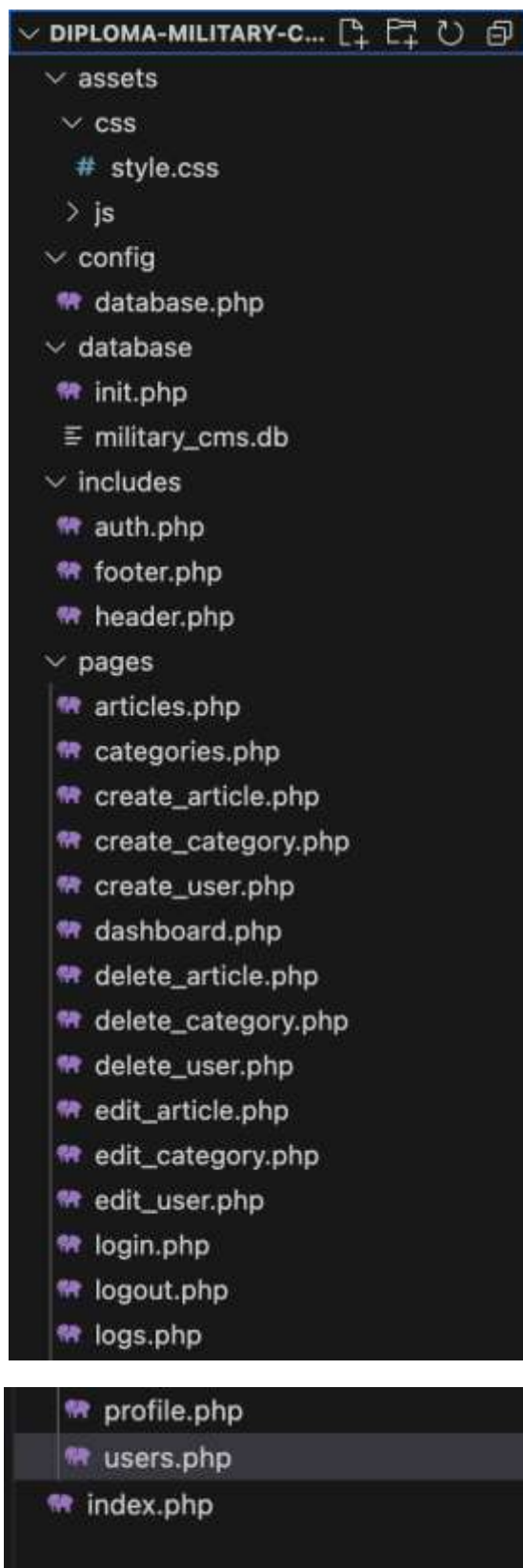


Додаток В - Діаграма діяльності процесу створення матеріалу в системі
Military CMS



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

sqlite> .tables
articles  categories  logs          users
sqlite> .schema articles
CREATE TABLE articles (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  title TEXT NOT NULL,
  content TEXT NOT NULL,
  category_id INTEGER NOT NULL,
  author_id INTEGER NOT NULL,
  status TEXT NOT NULL DEFAULT 'draft',
  created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
  updated_at TEXT,
  FOREIGN KEY (category_id) REFERENCES categories(id),
  FOREIGN KEY (author_id) REFERENCES users(id)
);
sqlite> .schema categories
CREATE TABLE categories (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT NOT NULL UNIQUE,
  description TEXT,
  created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP
);
sqlite> .schema logs
CREATE TABLE logs (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  user_id INTEGER,
  action TEXT NOT NULL,
  created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (user_id) REFERENCES users(id)
);
sqlite> .schema users
CREATE TABLE users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  full_name TEXT NOT NULL,
  username TEXT NOT NULL UNIQUE,
```

Додаток Д - Структура програмного проекту

Додаток Е – Фрагменти програмного коду Military CMS

Лістинг Е.1 – Фрагмент коду авторизації користувача

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username'] ?? '');
    $password = trim($_POST['password'] ?? '');

    if ($username === '' || $password === '') {
        $error = 'Будь ласка, заповніть усі поля.';
    } else {
        $pdo = Database::connect();

        $stmt = $pdo->prepare("
            SELECT * FROM users
            WHERE username = :username
            LIMIT 1
        ");

        $stmt->execute([
            ':username' => $username
        ]);

        $user = $stmt->fetch();

        if ($user && password_verify($password, $user['password'])) {
            $_SESSION['user'] = [
                'id' => $user['id'],
                'full_name' => $user['full_name'],
                'username' => $user['username'],
                'role' => $user['role']
            ];

            header('Location: dashboard.php');
            exit;
        } else {
            $error = 'Неправильний логін або пароль.';
        }
    }
}

```

Лістинг Е.2 – Фрагмент коду створення користувача

```

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $fullName = trim($_POST['full_name'] ?? '');
    $username = trim($_POST['username'] ?? '');
    $password = trim($_POST['password'] ?? '');
    $role = $_POST['role'] ?? 'viewer';

    if ($fullName === '' || $username === '' || $password === '') {
        $error = 'Будь ласка, заповніть усі обов'язкові поля.';
    } else {
        try {
            $stmt = $pdo->prepare("
                INSERT INTO users (full_name, username, password, role)
                VALUES (:full_name, :username, :password, :role)
            ");

            $stmt->execute([
                ':full_name' => $fullName,
                ':username' => $username,
                ':password' => password_hash($password, PASSWORD_DEFAULT),
            ]);
        } catch (PDOException $e) {
            $error = $e->getMessage();
        }
    }
}

```

```

        ':role' => $role
    });

    $logStmt = $pdo->prepare("
        INSERT INTO logs (user_id, action)
        VALUES (:user_id, :action)
    ");

    $logStmt->execute([
        ':user_id' => $_SESSION['user']['id'],
        ':action' => 'Створено користувача: ' . $username
    ]);

    header('Location: users.php');
    exit;
} catch (PDOException $e) {
    $error = 'Користувач із таким логіном уже існує.';
}
}
}

```

Лістинг Е.3 – Фрагмент коду створення таблиць бази даних

```

$pdo->exec("
    CREATE TABLE IF NOT EXISTS users (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        full_name TEXT NOT NULL,
        username TEXT NOT NULL UNIQUE,
        password TEXT NOT NULL,
        role TEXT NOT NULL,
        created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP
    )
");

$pdo->exec("
    CREATE TABLE IF NOT EXISTS categories (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL UNIQUE,
        description TEXT,
        created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP
    )
");

$pdo->exec("
    CREATE TABLE IF NOT EXISTS articles (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        title TEXT NOT NULL,
        content TEXT NOT NULL,
        category_id INTEGER NOT NULL,
        author_id INTEGER NOT NULL,
        status TEXT NOT NULL DEFAULT 'draft',
        created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
        updated_at TEXT,
        FOREIGN KEY (category_id) REFERENCES categories(id),
        FOREIGN KEY (author_id) REFERENCES users(id)
    )
");

$pdo->exec("
    CREATE TABLE IF NOT EXISTS logs (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        user_id INTEGER,
        action TEXT NOT NULL,

```

```
        created_at TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,  
        FOREIGN KEY (user_id) REFERENCES users(id)  
    )  
";
```