Хмельницький національний університет
Факультет інформаційних технологій
Кафедра кібербезпеки

КВАЛІФІКАЦІЙНА РОБОТА

Дзіблюк Ксенії Сергіївни

на здобуття ступеня вищої освіти Бакалавра

Система криптостійкого мережевого зберігання файлів із принципом нульової довіри до сервера

Галузь знань 12 – Інформаційні технології
Спеціальність 125 – Кібербезпека
Освітня програма Кібербезпека

Шифр КРБКБ.220237.22.02.25 ПЗ

Виконала студентка 4 курсу група КБ-22-2 Ксенія ДЗІБЛЮК
Керівник д-р. техн. наук, професор Михайло КАСЯНЧУК
Нормоконтролер д-р філософії Наталія ПЕТЛЯК

До захисту допускаю:
Завідувач кафедри кібербезпеки Юрій КЛЬОЦ

8 06 2026 р.

Хмельницький 2026

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Кібербезпеки

Рівень вищої освіти Бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 125 – Кібербезпека

Освітня програма Кібербезпека

ЗАТВЕРДЖУЮ

Завідувач кафедри кібербезпеки

Юрій КЛЬОЦ

09 лютого 2026 р.

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дзіблюк Ксенії Сергіївні

1 Тема роботи Система криптостійкого мережевого зберігання файлів із принципом нульової довіри до сервера.

Керівник роботи д-р техн. наук, професор Касянчук М.М

Затверджено наказом ректора університету від 8 січня 2026 № 7

2 Строк подання студентом кваліфікаційної роботи на кафедру 25.05.2026

3 Вихідні дані до роботи Дослідити існуючі системи захищеної передачі, зберігання та поширення файлів у корпоративних інформаційних системах. Проаналізувати актуальні загрози інформаційній безпеці при роботі з файлами. Сформулювати постановку задачі та визначити вимоги до системи. Розробити архітектуру вебсистеми та структуру бази даних. Реалізувати підсистеми автентифікації, авторизації та контролю доступу. Розробити модулі захищеного завантаження, зберігання та обробки файлів. Впровадити підсистему аудиту та логування дій користувачів. Реалізувати вебінтерфейс системи та провести тестування її функціональності.

4 Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Вступ. Аналіз предметної області та існуючих систем захищеного управління файлами. Аналіз загроз інформаційній безпеці та методів захисту даних у мережевому середовищі. Порівняльний аналіз систем-аналогів. Постановка задачі. Формування технічного завдання та вимог до системи. Розробка архітектури вебсистеми та структури бази даних. Формування загальної структури системи. Програмна реалізація підсистем автентифікації, авторизації та контролю доступу. Розробка модулів управління файлами. Впровадження підсистем аудиту та логування. Тестування системи.

5 Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Загальна архітектура вебсистеми захищеного управління файлами. Схема процесу завантаження та збереження файлу. Схема взаємозв'язків між таблицями бази даних.

6 Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7 Дата видачі завдання 09 лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
Вибір і затвердження теми кваліфікаційної роботи	Січень-Лютий	
Ознайомлення з предметною областю	Лютий	
Дослідження існуючих рішень	Лютий	
Постановка задачі	Березень	
Визначення загальних принципів рішення задачі	Березень	
Деталізація принципів рішення задачі	Квітень	
Розробка проєктних рішень	Квітень	
Апробація проєктних рішень	Травень	
Оформлення пояснювальної записки згідно вимог	Травень	
Оформлення графічної частини	Травень	
Захист КР	Червень	

Студентка _____ Ксенія ДЗІБЛЮК

Керівник кваліфікаційної роботи _____ Михайло КАСЯНЧУК

АНОТАЦІЯ

Тема кваліфікаційної роботи: Система криптостійкого мережевого зберігання файлів із принципом нульової довіри до сервера.

Автор роботи: Дзіблюк Ксенія Сергіївна

Керівник роботи: д-р техн. наук, проф. Касянчук Михайло Миколайович

Пояснювальна записка: 69 с., 27 рис., 12 табл., 45 джерел

Графічна частина: 3 плакати

Ключові слова: захищене управління файлами, корпоративна інформаційна система, інформаційна безпека, контроль доступу, автентифікація, авторизація, шифрування даних, Laravel, PHP, вебсистема.

У кваліфікаційній роботі проведено аналіз особливостей функціонування корпоративних інформаційних систем та основних загроз інформаційній безпеці під час роботи з файлами у мережевому середовищі. Досліджено сучасні методи захисту даних, зокрема механізми автентифікації, авторизації, контролю доступу, шифрування інформації та журналювання подій.

Виконано порівняльний аналіз сучасних систем управління файлами та сформовано вимоги до розробки захищеної вебсистеми для корпоративного середовища.

У роботі обґрунтовано використання PHP та Laravel, розроблено архітектуру системи і реалізовано механізми безпечного зберігання файлів, контролю доступу та захисту від типових вебзагроз.

Результатом роботи є програмний прототип системи захищеного управління файлами, що забезпечує централізоване зберігання даних і підвищення рівня інформаційної безпеки.

25.05.2026 Дзіблюк К.С

ABSTRACT

Subject of qualification work: Cryptographically Secure Network File Storage System with a the Zero-Trust server Mode.

Author: Dzibliuk Kseniia Serhiivna

Head of work: Doctor of Technical Sciences, Professor Kasianchuk Mykhailo Mykolaiovych

Explanatory note: 69 pages, 27 figures, 12 tables, 45 sources

Graphic part: 3 posters

Keywords: file management protection, corporate information system, information security, access control, authentication, authorization, data encryption, Laravel, PHP, web system.

The qualification work analyzes the operational features of corporate information systems and the main threats to information security during file handling in a network environment. Modern methods of data protection were studied, including authentication and authorization mechanisms, access control, data encryption, and event logging.

A comparative analysis of modern file management systems was carried out, and requirements for the development of a secure web system for a corporate environment were formulated.

The paper substantiates the use of PHP and Laravel technologies, develops the system architecture, and implements mechanisms for secure file storage, access control, and protection against common web threats.

The result of the work is a software prototype of a secure file management system that provides centralized data storage and improves the level of information security.

25.05.2026 Дзіблюк К.С

ЗМІСТ

ВСТУП

У сучасних умовах цифрової трансформації бізнесу та державних установ особливого значення набуває ефективне управління інформацією. З розвитком корпоративних інформаційних систем обсяги оброблюваних даних постійно зростають, а разом із ними підвищуються вимоги до забезпечення їхньої безпеки, цілісності та доступності [1]. Надійне зберігання та контроль доступу до електронних документів і файлів стають одним із ключових завдань для організацій, що прагнуть захистити корпоративні ресурси та дотримуватися внутрішніх і зовнішніх нормативних вимог.

Однією з найбільш актуальних проблем у сучасних корпоративних середовищах є безпечне управління файлами. Незахищений доступ до конфіденційної інформації може призвести до витоку даних, втрати важливих документів або порушення політик організації. Крім того, поширення віддаленої роботи та активне використання хмарних технологій створюють додаткові ризики, які необхідно враховувати під час розробки сучасних систем управління файлами [1, 2].

Обрання саме корпоративної сфери обумовлене тим, що організації працюють із великими обсягами критично важливої інформації та потребують централізованого контролю доступу, аудиту дій користувачів і безпечного зберігання документів [2]. На відміну від персональних або навчальних систем, корпоративні середовища характеризуються складною структурою користувачів, наявністю різних рівнів доступу та інтеграцією з іншими бізнес-процесами, що робить задачу захищеного управління файлами більш комплексною та практично значущою.

Сучасні тенденції розвитку корпоративних систем свідчать про активну автоматизацію бізнес-процесів та інтеграцію різноманітних сервісів: від систем електронного документообігу до хмарних платформ і засобів корпоративної комунікації.

Додаткову актуальність проблемі надає постійне зростання кількості

кіберзагроз. Щороку збільшується кількість атак на корпоративні мережі, серед яких витоки конфіденційної інформації, несанкціонований доступ до даних та шифрування файлів із вимогою викупу [3]. У таких умовах особливого значення набуває впровадження комплексних рішень, що поєднують механізми автентифікації користувачів, розмежування прав доступу, шифрування даних та аудит дій у системі [4, 5].

Аналіз існуючих програмних рішень показує, що значна частина сучасних систем або має обмежений функціонал, або недостатньо адаптована до потреб корпоративного середовища. Деякі системи не забезпечують належного рівня захисту інформації, інші є складними в інтеграції або незручними у використанні. Це підкреслює необхідність розробки ефективної та гнучкої системи захищеного управління файлами, здатної поєднувати сучасні методи забезпечення інформаційної безпеки із зручністю використання та можливістю масштабування.

Актуальність розробки подібних систем також зумовлена стрімким зростанням обсягів цифрової інформації та необхідністю її безпечного зберігання й обробки. У сучасних умовах значна частина корпоративних даних представлена у вигляді електронних файлів, доступ до яких потребує чіткого контролю та захисту від несанкціонованого використання. Особливої важливості набувають питання забезпечення конфіденційності, цілісності та доступності інформації, які є базовими принципами інформаційної безпеки та визначають вимоги до сучасних систем управління файлами [1].

У роботі використано сучасні підходи до забезпечення інформаційної безпеки, зокрема механізми контролю доступу, захисту даних та фіксації дій користувачів. Це дозволяє реалізувати систему, яка відповідає основним вимогам до безпечної роботи з інформацією у корпоративному середовищі [2], [5].

Метою роботи є проектування та реалізація системи захищеного управління файлами, яка забезпечує безпечне зберігання, доступ та контроль корпоративних даних в умовах мережевої взаємодії.

Для досягнення поставленої мети необхідно виконати такі завдання:

– проаналізувати специфіку функціонування корпоративних

інформаційних систем та визначити потенційні загрози при обробці й зберіганні файлів;

– дослідити сучасні методи та технології забезпечення безпеки даних у мережевому середовищі;

– провести порівняльний аналіз існуючих систем управління файлами та визначити їхні переваги й недоліки;

– сформулювати вимоги до розробки захищеної системи управління файлами;

– розробити технічне завдання та визначити основні функціональні й нефункціональні вимоги до системи;

– створити архітектуру вебсистеми та структуру бази даних для надійного зберігання й обробки інформації;

– реалізувати основні модулі системи: автентифікації, авторизації, шифрування та безпечного зберігання файлів;

– провести тестування системи, оцінити ефективність реалізованих рішень та підготувати рекомендації щодо впровадження системи у корпоративне середовище.

Використання систем захищеного управління файлами дозволяє підвищити рівень захисту корпоративної інформації, забезпечити контроль доступу до даних та покращити організацію їх зберігання й обробки. Такі системи сприяють централізованому управлінню файлами, що особливо важливо для організацій із великою кількістю користувачів та різними рівнями доступу.

Таким чином, розробка системи захищеного управління файлами є актуальним завданням, спрямованим на забезпечення безпеки корпоративної інформації, підвищення ефективності роботи з документами та створення надійного механізму контролю доступу до даних у сучасному корпоративному середовищі.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ СИСТЕМ ЗАХИЩЕНОГО УПРАВЛІННЯ ФАЙЛАМИ

1.1 Аналіз особливостей функціонування корпоративних інформаційних систем та загроз при роботі з файлами

Спочатку виконаємо аналіз предметної області, пов’язаної із захищеним управлінням файлами у корпоративних інформаційних системах. Це дозволить визначити основні принципи організації роботи з корпоративними даними, встановити характерні особливості обробки файлів та дослідити актуальні загрози інформаційній безпеці, які необхідно врахувати під час розробки програмної системи.

В межах даної роботи основною предметною областю є корпоративні інформаційні системи, оскільки саме вони забезпечують централізоване зберігання, обробку та передачу електронних документів і файлів між користувачами організації.

Корпоративна інформаційна система (далі – КІС) являє собою комплекс програмних, технічних та організаційних засобів, призначених для автоматизації діяльності підприємства та підтримки управлінських процесів. Такі системи забезпечують збір, зберігання, обробку та аналіз інформації, необхідної для функціонування різних підрозділів організації [6].

У сучасних умовах КІС використовуються практично в усіх сферах діяльності підприємств. Вони об’єднують бази даних, мережеву інфраструктуру, програмні сервіси та користувачів у єдине інформаційне середовище. Значна частина інформації у таких системах представлена у вигляді електронних файлів і документів, доступ до яких повинен бути контрольованим та захищеним відповідно до національних стандартів безпеки [7].

Особливістю КІС є наявність багаторівневої структури користувачів та розмежування прав доступу. Працівники різних підрозділів можуть мати різні повноваження щодо перегляду, редагування, завантаження або видалення файлів. У зв’язку з цим виникає необхідність використання механізмів автентифікації,

авторизації та контролю доступу до інформаційних ресурсів [8].

Крім того, сучасні КІС часто інтегруються із зовнішніми програмними сервісами, серед яких системи електронного документообігу, CRM-системи, хмарні платформи та засоби корпоративної комунікації. Така інтеграція розширює функціональні можливості системи, однак одночасно підвищує ризики несанкціонованого доступу до корпоративних даних [9].

У межах функціонування КІС можна виділити типові операції роботи з файлами:

– створення та завантаження файлів до системи;
– перегляд і редагування документів;
– передача файлів між користувачами;
– зберігання та архівування інформації;
– видалення або відновлення файлів;
– контроль версій документів;
– резервне копіювання даних.

Для наочності основні етапи роботи з файлами у КІС доцільно представити у вигляді схеми життєвого циклу файла, як показано на рисунку 1.1.

Рисунок 1.1 – Життєвий цикл файла у корпоративній інформаційній системі

З розвитком мережевих технологій значна частина КІС функціонує із

використанням хмарної інфраструктури та віддаленого доступу. Це забезпечує можливість спільної роботи користувачів із документами та доступу до файлів із різних пристроїв [10]. Водночас використання мережевих технологій створює додаткові ризики, пов’язані з витоком інформації, перехопленням мережевого трафіку або компрометацією серверної інфраструктури [11].

У зв’язку з цим важливого значення набуває проблема забезпечення захисту корпоративних файлів. Файли можуть містити конфіденційну інформацію, персональні дані або комерційну таємницю, тому порушення їх цілісності чи конфіденційності може призвести до значних фінансових, репутаційних втрат, а в окремих випадках – до загрози національній безпеці [12].

Особливу увагу під час функціонування КІС необхідно приділяти питанням інформаційної безпеки, оскільки корпоративні файли можуть бути об’єктом різноманітних зовнішніх і внутрішніх загроз. До найбільш поширених загроз належать несанкціонований доступ до інформації, використання шкідливого програмного забезпечення, перехоплення даних у мережі, помилки користувачів та неправомірні дії співробітників, які використовують слабкі паролі або порушують політики безпеки [13].

Для кращого аналізу основні загрози інформаційній безпеці КІС та їх можливі наслідки доцільно узагальнити у таблиці 1.1.

Для мінімізації зазначених загроз у КІС застосовуються різні методи забезпечення інформаційної безпеки. До основних із них належать механізми автентифікації та авторизації користувачів, шифрування даних, резервне копіювання, аудит дій користувачів та системи моніторингу безпеки [1], [4].

Комплексне використання таких механізмів дозволяє забезпечити контроль доступу до корпоративної інформації, підвищити рівень захисту даних та зменшити ймовірність виникнення інцидентів інформаційної безпеки. Водночас ефективність захисту значною мірою залежить від правильності організації роботи системи та дотримання користувачами встановлених політик безпеки [2].

Таким чином, аналіз особливостей функціонування КІС показав, що корпоративні файли є критично важливим інформаційним ресурсом організації та

потребують надійного захисту. Зростання кількості кіберзагроз і використання мережевих технологій обумовлює необхідність створення систем захищеного управління файлами, які забезпечуватимуть контроль доступу, цілісність та конфіденційність інформації [14].

Таблиця 1.1 – Основні загрози інформаційній безпеці КІС

Загроза	Характеристика	Можливі наслідки
Несанкціонований доступ	Отримання доступу до файлів сторонніми особами	Витік конфіденційної інформації
Шкідливе ПЗ	Використання вірусів або троянів	Пошкодження або викрадення даних
Ransomware	Шифрування файлів із вимогою викупу	Втрата доступу до інформації
Перехоплення даних	Отримання інформації під час передачі мережею	Компрометація корпоративних даних
Внутрішні загрози	Неправомірні дії працівників	Видалення або копіювання файлів
Помилки користувачів	Неправильна робота із системою	Втрата або зміна даних

1.2 Аналіз сучасних методів та засобів забезпечення захисту даних у мережевому середовищі

У сучасних корпоративних інформаційних системах питання забезпечення захисту даних є одним із ключових факторів стабільного та безпечного функціонування інформаційного середовища. Значна частина корпоративної інформації передається через мережу та зберігається у вигляді електронних файлів, тому виникає необхідність використання ефективних механізмів контролю доступу, захисту каналів передачі даних та забезпечення цілісності інформації відповідно до встановлених державних нормативів [15].

Особливу актуальність проблема захисту даних набуває у веборієнтованих системах, які забезпечують віддалений доступ користувачів до корпоративних ресурсів. Використання мережевих технологій і хмарної інфраструктури створює

додаткові ризики, пов’язані з несанкціонованим доступом до інформації, перехопленням мережевого трафіку, використанням шкідливого програмного забезпечення та спробами компрометації облікових записів користувачів [16].

У межах даної роботи основна увага приділяється механізмам автентифікації та авторизації користувачів, контролю доступу до файлів, захисту каналів передачі даних, шифруванню інформації та аудиту дій користувачів. Використання комплексного підходу до забезпечення інформаційної безпеки дозволяє підвищити рівень захисту корпоративних ресурсів та мінімізувати ризики втрати або компрометації даних [17].

Одним із базових механізмів захисту інформації є автентифікація користувачів. Її основним завданням є перевірка достовірності користувача перед наданням доступу до системи. У сучасних вебсистемах найбільш поширеним способом автентифікації є використання логіна та пароля із подальшим хешуванням паролів у базі даних. Це дозволяє зменшити ризик компрометації облікових записів у випадку витоку даних [18].

Після проходження автентифікації важливу роль відіграє механізм авторизації, який визначає права доступу користувача до ресурсів системи. Для корпоративних систем особливо важливим є розмежування доступу між різними категоріями користувачів. У рамках даного дослідження для забезпечення контролю доступу використовується мандатна модель доступу, що передбачає наявність різних рівнів доступу до інформаційних ресурсів системи.

Мандатна модель доступу дозволяє встановити правила взаємодії користувачів із файлами залежно від їхніх повноважень та рівня доступу до інформації. Такий підхід забезпечує централізований контроль доступу до корпоративних даних та дозволяє зменшити ризики несанкціонованого використання інформації [19].

Важливим елементом забезпечення безпеки є захист каналів передачі даних. Для цього у сучасних вебсистемах використовується протокол HTTPS, який забезпечує шифрування мережевого трафіку за допомогою технології TLS. Це дозволяє запобігти перехопленню конфіденційної інформації під час її передачі

мережею [20, 21].

Крім захисту мережевого трафіку, значну роль відіграє безпечне зберігання файлів у системі. Для мінімізації ризиків втрати або компрометації інформації використовуються механізми шифрування даних, резервного копіювання та обмеження прямого доступу до файлів на сервері [22].

Для підвищення рівня безпеки вебсистем також застосовуються механізми захисту від типових вебзагроз. До найбільш поширених із них належать SQL-ін’єкції, міжсайтовий скриптинг (XSS) та підробка міжсайтових запитів (CSRF). Використання сучасних фреймворків дозволяє реалізувати вбудовані механізми захисту від подібних атак та підвищити загальний рівень безпеки системи [23], [24, 25].

Окрему увагу під час розробки системи необхідно приділяти аудиту дій користувачів. Журналювання подій дозволяє фіксувати операції завантаження, редагування, видалення та перегляду файлів, що забезпечує можливість контролю активності користувачів та виявлення потенційних інцидентів інформаційної безпеки [26].

Для реалізації вебсистеми захищеного управління файлами необхідно обрати сучасні програмні засоби та технології, які забезпечують необхідний рівень безпеки, продуктивності та масштабованості системи. У процесі виконання роботи для серверної частини системи використовується мова програмування PHP та фреймворк Laravel [27].

PHP є однією з найбільш поширених мов програмування для створення вебзастосунків. Основними перевагами PHP є простота інтеграції із серверними технологіями, підтримка роботи з реляційними базами даних та велика кількість готових бібліотек і засобів для розробки вебсистем [28].

Laravel є сучасним PHP-фреймворком, побудованим на основі архітектурного шаблону MVC. Використання Laravel дозволяє реалізувати механізми маршрутизації, авторизації, роботи з базою даних та захисту від типових вебатак. Крім того, фреймворк містить вбудовані засоби автентифікації користувачів, middleware-механізми контролю доступу та підтримку ORM

Eloquent для безпечної взаємодії з базою даних [27, 29].

Для зберігання інформації у системі використовується реляційна база даних, яка забезпечує структуроване зберігання інформації, підтримку зв'язків між таблицями та контроль цілісності даних. Використання SQL дозволяє ефективно організувати роботу з користувачами, файлами, журналами подій та іншими компонентами системи [30].

У процесі розробки програмного забезпечення важливу роль також відіграють системи контролю версій. Для цього використовується Git, який забезпечує контроль змін у програмному коді, спрощує командну роботу та дозволяє зберігати історію змін проєкту [31].

Для обґрунтування вибору серверного фреймворку доцільно провести порівняльний аналіз сучасних технологій веброзробки, як показано в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз серверних фреймворків

Критерій	Вага	Laravel	Django	ASP.NET
Рівень безпеки	0,3	5	5	5
Простота розробки	0,2	5	4	3
Документація та підтримка	0,15	5	4	4
Підтримка MVC	0,15	5	5	5
Інтеграція з БД	0,1	5	5	4
Швидкість розробки	0,1	5	4	3
Інтегральна оцінка	1	5	4,5	4,2

Для оцінювання ефективності використання технологій використовується інтегральний показник:

$$s = \sum_{i=1}^{n} w_i \cdot x_i, \qquad (1.1)$$

де w_i – ваговий коефіцієнт критерію оцінювання,

x_i – оцінка технології за відповідним критерієм.

Для фреймворку Laravel інтегральний показник визначається як:

$$S = 0{,}30 \cdot 5 + 0{,}20 \cdot 5 + 0{,}15 \cdot 5 + 0{,}15 \cdot 5 + 0{,}10 \cdot 5 + 0{,}10 \cdot 5 = 5{,}00 \quad (1.2)$$

Для фреймворку Django:

$$S = 0{,}30 \cdot 5 + 0{,}20 \cdot 4 + 0{,}15 \cdot 4 + 0{,}15 \cdot 5 + 0{,}10 \cdot 5 + 0{,}10 \cdot 4 = 4{,}55 \quad (1.3)$$

Для фреймворку ASP.NET:

$$S = 0{,}30 \cdot 5 + 0{,}20 \cdot 3 + 0{,}15 \cdot 4 + 0{,}15 \cdot 5 + 0{,}10 \cdot 4 + 0{,}10 \cdot 3 = 4{,}15 \quad (1.3)$$

За результатами проведеного аналізу найбільше значення інтегрального показника отримав фреймворк Laravel. Це обумовлено високим рівнем безпеки, наявністю вбудованих механізмів автентифікації та авторизації, підтримкою MVC-архітектури, зручною інтеграцією з реляційними базами даних та високою швидкістю розробки вебзастосунків.

Таким чином, проведений аналіз сучасних методів та засобів забезпечення захисту даних показав, що для створення системи захищеного управління файлами доцільно використовувати комплексний підхід, який поєднує механізми автентифікації, контролю доступу, шифрування даних та аудиту дій користувачів. Обрані технології та програмні засоби забезпечують необхідний рівень безпеки, масштабованості та ефективності функціонування вебсистеми у корпоративному середовищі.

1.3 Порівняльний аналіз існуючих систем-аналогів управління файлами

Під час проєктування системи захищеного управління файлами важливим етапом є аналіз існуючих програмних рішень, які використовуються для зберігання, синхронізації та контролю доступу до інформації. Проведення такого аналізу дозволяє визначити переваги та недоліки сучасних систем, оцінити рівень

реалізації механізмів безпеки та сформувати вимоги до розроблюваної вебсистеми.

Сучасні системи управління файлами представлені широким спектром рішень, які відрізняються за архітектурою, функціональністю та рівнем захисту інформації. Найбільш поширеними є хмарні сервіси, системи приватного розгортання та спеціалізовані інструменти для синхронізації та резервного копіювання даних [32].

Одними з найпоширеніших систем є хмарні сервіси Google Drive, Dropbox та Microsoft OneDrive. Дані рішення забезпечують можливість зберігання файлів у хмарному середовищі, доступ до інформації з різних пристроїв та підтримку спільної роботи користувачів у режимі реального часу. Основними перевагами таких систем є простота використання, масштабованість та інтеграція з іншими програмними сервісами [33-35].

Крім зазначених рішень, у корпоративному середовищі також активно використовуються сервіси Box, Mega та Sync.com, які підтримують хмарне зберігання даних, синхронізацію файлів і механізми захищеного обміну інформацією. Особливістю сервісів Mega та Sync.com є використання наскрізного шифрування даних, що дозволяє підвищити рівень конфіденційності інформації під час її зберігання та передачі мережею. Система Box орієнтована переважно на корпоративний сектор та забезпечує інтеграцію із засобами командної роботи й централізованого управління доступом до файлів [36-38].

Водночас використання публічних хмарних сервісів має певні недоліки. Зокрема, організації не мають повного контролю над серверною інфраструктурою та політиками зберігання даних. Крім того, зберігання конфіденційної інформації на сторонніх серверах може створювати додаткові ризики для інформаційної безпеки [39].

Альтернативою публічним хмарним сервісам є системи приватного розгортання, зокрема Nextcloud, Seafile та ownCloud. Подібні рішення дозволяють організаціям самостійно керувати інфраструктурою зберігання даних,

налаштовувати механізми контролю доступу та використовувати власні політики безпеки [40-42].

Системи приватного розгортання забезпечують вищий рівень контролю над інформацією та дозволяють інтегрувати додаткові механізми захисту даних. Однак їх використання потребує додаткових ресурсів для адміністрування серверів, технічного супроводу та резервного копіювання інформації.

Також до корпоративних систем управління файлами можна віднести Oracle Cloud File System, яка орієнтована на використання у великих корпоративних середовищах та хмарній інфраструктурі. Дана система забезпечує масштабоване зберігання інформації, підтримує розподілену обробку даних та дозволяє інтегрувати механізми резервного копіювання і контролю доступу. Основними перевагами Oracle Cloud File System є висока відмовостійкість, продуктивність та підтримка корпоративних сервісів, однак використання подібних рішень супроводжується складністю налаштування та високими вимогами до інфраструктури [43].

Окрему категорію складають спеціалізовані інструменти синхронізації та резервного копіювання даних, серед яких можна виділити Rclone. Такі інструменти орієнтовані переважно на адміністраторів та досвідчених користувачів і дозволяють автоматизувати процеси міграції, синхронізації та резервного копіювання файлів між різними хмарними платформами [44].

Крім централізованих систем, існують також розподілені файлові системи, наприклад Tahoe-LAFS, які використовують децентралізовані підходи до зберігання інформації. Подібні системи забезпечують високу відмовостійкість і підвищений рівень захисту даних, однак характеризуються складністю налаштування та адміністрування [45].

Для кращого порівняння сучасних систем управління файлами доцільно узагальнити їх основні характеристики у таблиці 1.3.

Таблиця 1.3 – Порівняльний аналіз сучасних систем управління файлами

№ з/п	Система	Тип рішення	Основні можливості	Переваги	Недоліки
1	Google Drive	Хмарна система	Зберігання, синхронізація, спільний доступ	Простота, інтеграція з сервісами Google	Обмежений контроль над даними
2	Dropbox	Хмарна система	Синхронізація файлів, резервне копіювання	Висока швидкість синхронізації	Обмеження безкоштовного тарифу
3	Microsoft OneDrive	Хмарна система	Інтеграція з Office, спільна робота	Зручність для користувачів Windows	Залежність від екосистеми Microsoft
4	Nextcloud	Приватне розгортання	Контроль доступу, шифрування, зберігання	Повний контроль над даними	Потребує адміністрування
5	Seafile	Приватне розгортання	Синхронізація, версіонування, шифрування	Висока продуктивність	Складність налаштування
6	Oracle Cloud FS	Корпоративна хмарна система	Масштабованість, резервне копіювання	Відмовостійкість, підтримка інфраструктури	Високі вимоги до ресурсів
7	Rclone	CLI-інструмент	Синхронізація та резервне копіювання	Гнучкість та автоматизація	Потребує технічних знань
8	Tahoe-LAFS	Розподілена файлова система	Децентралізоване зберігання та захист	Висока безпека та відмовостійкість	Складність адміністрування
9	Box	Корпоративна хмарна система	Спільна робота, контроль доступу, інтеграція	Високий рівень безпеки	Висока вартість корпоративних тарифів
10	ownCloud	Приватне розгортання	Локальне зберігання, контроль доступу	Повний контроль над даними	Потребує адміністрування
11	Mega	Хмарна система	Хмарне зберігання та обмін файлами	Наскрізне шифрування	Обмеження швидкості передачі
12	Sync.com	Хмарна система	Синхронізація та резервне копіювання	End-to-end шифрування	Менша інтеграція з офісними сервісами

Проведений аналіз існуючих систем управління файлами показав, що сучасні рішення забезпечують широкий набір можливостей для зберігання та синхронізації даних, однак мають певні обмеження з точки зору контролю доступу, гнучкості налаштування та забезпечення інформаційної безпеки. Особливо це стосується корпоративного середовища, де важливими є централізоване управління користувачами, аудит дій та контроль доступу до конфіденційної інформації.

Таким чином, виникає необхідність розробки власної системи захищеного управління файлами, орієнтованої на забезпечення безпечного зберігання даних, реалізацію механізмів контролю доступу та підтримку корпоративних вимог до інформаційної безпеки.

1.4 Постановка задачі на розробку захищеної системи

У результаті проведеного аналізу предметної області встановлено, що корпоративні інформаційні системи відіграють важливу роль у забезпеченні діяльності сучасних організацій та потребують надійних механізмів захисту інформації. Значна частина корпоративних даних зберігається у вигляді електронних файлів, доступ до яких повинен бути контрольованим та захищеним від несанкціонованого використання.

Під час дослідження особливостей функціонування корпоративних інформаційних систем було визначено, що використання мережевих технологій та хмарної інфраструктури створює додаткові ризики для інформаційної безпеки. До основних загроз належать несанкціонований доступ до інформації, шкідливе програмне забезпечення, перехоплення мережевого трафіку, помилки користувачів та внутрішні загрози, пов’язані з неправомірними діями співробітників.

Проведений аналіз сучасних методів та засобів захисту даних показав, що ефективне забезпечення інформаційної безпеки можливе лише за умови

комплексного використання механізмів автентифікації, авторизації, контролю доступу, шифрування даних та аудиту дій користувачів. Крім того, встановлено, що сучасні вебтехнології та програмні фреймворки дозволяють реалізувати необхідні механізми захисту та забезпечити безпечну роботу із файлами у мережевому середовищі.

У ході порівняльного аналізу існуючих систем управління файлами визначено, що публічні хмарні сервіси забезпечують високий рівень доступності та зручності використання, проте мають обмеження щодо контролю над даними та налаштування механізмів безпеки. Приватні системи та корпоративні рішення, навпаки, забезпечують вищий рівень контролю та конфіденційності, однак потребують додаткових ресурсів для розгортання та адміністрування.

Таким чином, результати проведеного аналізу підтверджують доцільність розробки власної захищеної системи управління файлами, орієнтованої на використання у корпоративному середовищі. Розроблювана система повинна поєднувати зручність роботи з файлами, підтримку мережевої взаємодії та сучасні механізми забезпечення інформаційної безпеки.

Метою кваліфікаційної роботи є проєктування та реалізація захищеної вебсистеми управління файлами для корпоративного середовища, яка забезпечує безпечне зберігання, передачу та контроль доступу до інформації.

Для досягнення поставленої мети необхідно послідовно виконати такі задачі:

– проаналізувати особливості функціонування корпоративних інформаційних систем та визначити основні загрози під час роботи з файлами;

– дослідити сучасні методи та засоби забезпечення захисту даних у мережевому середовищі;

– провести порівняльний аналіз існуючих систем управління файлами та визначити їх переваги і недоліки;

– сформувати вимоги до захищеної системи управління файлами;

– розробити архітектуру вебсистеми та структуру бази даних;

– реалізувати механізми автентифікації та авторизації користувачів;

– реалізувати механізми контролю доступу до файлів;

– забезпечити безпечне завантаження та зберігання файлів;

– реалізувати шифрування даних та захист каналів передачі інформації;

– впровадити систему журналювання дій користувачів;

– провести тестування системи та оцінити ефективність реалізованих рішень.

Для забезпечення необхідного рівня інформаційної безпеки система повинна підтримувати перевірку файлів перед завантаженням, обмеження типів та розміру файлів, захист від несанкціонованого доступу до інформації, а також механізми шифрування та аудиту дій користувачів.

Крім того, система повинна забезпечувати зручний користувацький інтерфейс, який дозволяє виконувати основні операції з файлами без необхідності спеціальної технічної підготовки користувачів.

Додатково під час розробки системи необхідно забезпечити можливість подальшого розширення функціональності та адаптації системи до потреб корпоративного середовища. Це передбачає використання сучасних підходів до проєктування вебзастосунків, модульної структури програмного забезпечення та засобів централізованого управління даними. Такий підхід дозволить підвищити гнучкість системи, спростити її супровід та забезпечити можливість інтеграції з іншими корпоративними сервісами у майбутньому.

Послідовне виконання поставлених задач дозволить досягти мети кваліфікаційної роботи у повному обсязі та створити захищену вебсистему управління файлами, орієнтовану на використання у корпоративному середовищі.

2 ПРОЄКТУВАННЯ СИСТЕМИ ЗАХИЩЕНОГО УПРАВЛІННЯ ФАЙЛАМИ

2.1 Формування технічного завдання та основних вимог до системи захищеного управління файлами

На основі проведеного аналізу предметної області, сучасних методів забезпечення інформаційної безпеки та існуючих систем управління файлами сформовано технічне завдання на розробку захищеної вебсистеми управління файлами для корпоративного середовища. Основною метою розробки є створення системи, яка забезпечуватиме безпечне зберігання, передачу та контроль доступу до корпоративних файлів із використанням сучасних механізмів захисту інформації.

У процесі формування технічного завдання враховано специфіку функціонування корпоративних інформаційних систем, характер основних загроз інформаційній безпеці та необхідність організації безпечної роботи користувачів із файлами у мережевому середовищі. Особлива увага приділяється забезпеченню конфіденційності, цілісності та доступності інформації, оскільки корпоративні файли можуть містити конфіденційні дані, внутрішню документацію або службову інформацію організації.

Для забезпечення контролю доступу до інформації у системі використовується мандатна модель доступу. Дана модель передбачає розмежування користувачів та інформаційних ресурсів за рівнями доступу і дозволяє централізовано контролювати можливість виконання операцій із файлами.

В межах системи передбачено п’ять рівнів секретності інформації, як показано на рисунку 2.1.

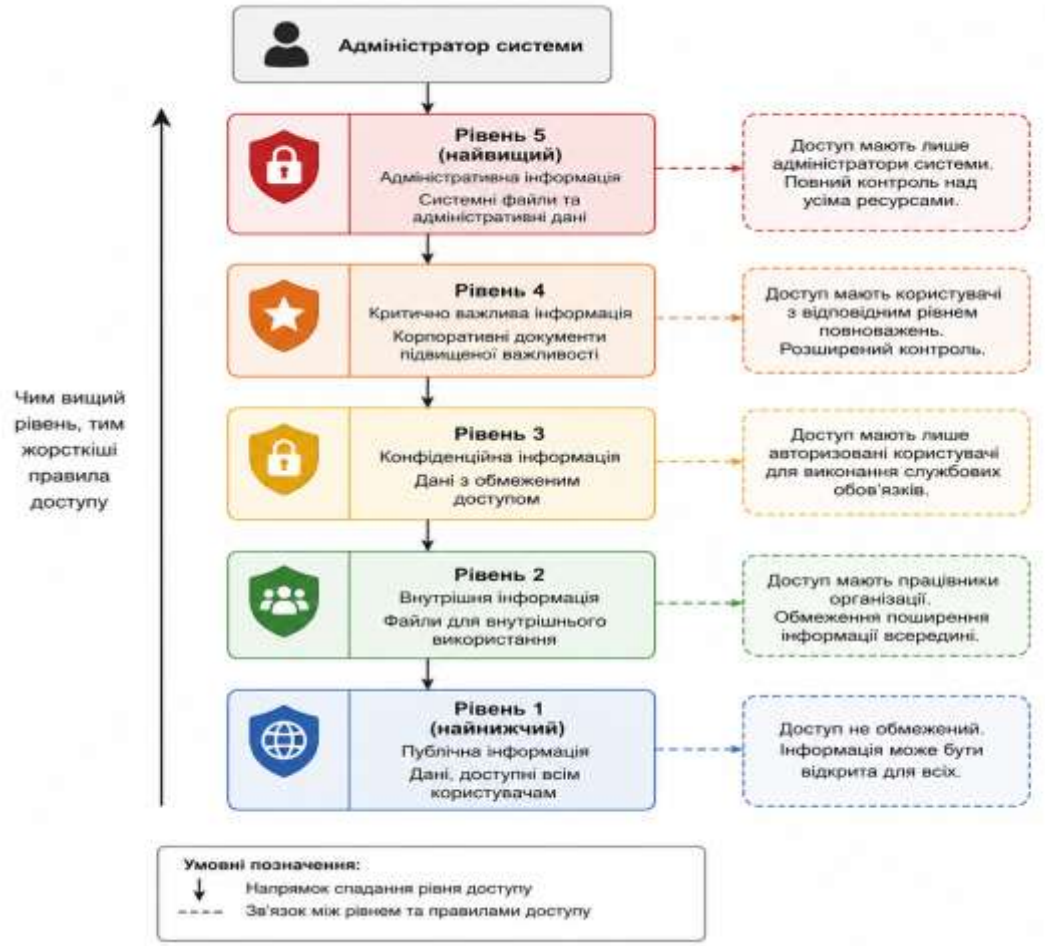


Рисунок 2.1 – Мандатна модель доступу до файлів у системі

Чим вищий рівень секретності інформації, тим жорсткіші вимоги висуваються до механізмів її захисту. Для файлів із високим рівнем секретності передбачаються додаткові механізми контролю доступу, журналювання дій та шифрування даних.

На основі проведеного аналізу визначено найбільш критичні загрози для системи захищеного управління файлами, як показано в таблиці 2.1.

Початок таблиці 2.1 – Основні загрози для системи

№ з/п	Загроза	Можливі наслідки	Пріоритет
1	2	3	4
1	Несанкціонований доступ до файлів	Витік конфіденційної інформації	Критичний

Кінець таблиці 2.1

1	2	3	4
2	SQL-ін'єкції	Компрометація бази даних	Критичний
3	XSS-атаки	Отримання доступу до сесій користувачів	Високий
4	Перехоплення мережевого трафіку	Компрометація інформації	Високий
5	Шкідливе програмне забезпечення	Пошкодження або викрадення файлів	Високий
6	Видалення файлів	Порушення доступності інформації	Високий
7	Помилки користувачів	Втрата або зміна даних	Середній

Оскільки з урахуванням ресурсних, часових та технічних обмежень у ході дослідження неможливо реалізувати засоби протидії всім можливим загрозам, запропоновані механізми захисту будуть орієнтовані насамперед на протидію найбільш критичним загрозам, які можуть призвести до компрометації корпоративної інформації або порушення роботи системи.

Для протидії визначеним загрозам у системі передбачено використання відповідних механізмів захисту, як показано в таблиці 2.2.

Початок таблиця 2.2 – Механізми протидії загрозам

№ з/п	Загроза	Механізм захисту
1	2	3
1	Несанкціонований доступ	Автентифікація та мандатний контроль доступу
2	SQL-ін’єкції	Використання ORM Eloquent та валідація вхідних даних
3	XSS-атаки	Екранування даних засобами Blade
4	Перехоплення мережевого трафіку	Використання захищених протоколів шифрування HTTPS та TLS
5	Компрометація паролів	Хешування паролів за допомогою стійкого алгоритму bcrypt
6	Видалення або втрата файлів	Регулярне резервне копіювання та керування версіями

Кінець таблиці 2.2

1	2	3
7	Неправомірні дії користувачів	Детальне журналювання подій та аудит усіх дій у системі

Крім зовнішніх загроз, під час розробки системи необхідно враховувати й можливі дії потенційного порушника. У межах даної роботи під порушником розуміється особа або програмний засіб, який намагається отримати несанкціонований доступ до інформації або порушити функціонування системи. Приклад показано на рисунку 2.2.

Рисунок 2.2 – Модель порушника системи управління файлами

Аналіз моделі порушника показує, що найбільшу небезпеку для системи становлять спроби несанкціонованого доступу до інформації, компрометація облікових записів користувачів та порушення цілісності даних. Особливу увагу необхідно приділити як зовнішнім загрозам, пов’язаним із мережевими атаками

та використанням шкідливого програмного забезпечення, так і внутрішнім загрозам, спричиненим перевищенням повноважень або помилками користувачів. Це обумовлює необхідність реалізації комплексних механізмів захисту, зокрема мандатного контролю доступу, автентифікації користувачів, журналювання дій та захисту каналів передачі інформації.

На основі визначених загроз та моделі порушника сформовано основні технічні вимоги до системи, їх показано в таблиці 2.3.

Таблиця 2.3 – Основні технічні вимоги до системи

Код вимоги	Технічна вимога
ТВ1	Система повинна забезпечувати автентифікацію користувачів
ТВ2	Система повинна реалізовувати мандатний контроль доступу
ТВ3	Паролі користувачів повинні зберігатися у хешованому вигляді
ТВ4	Передача даних повинна здійснюватися через HTTPS
ТВ5	Система повинна підтримувати безпечне завантаження файлів
ТВ6	Повинна виконуватися перевірка типу та розміру файлів
ТВ7	Система повинна підтримувати журналювання дій користувачів
ТВ8	Повинен бути реалізований механізм шифрування файлів
ТВ9	Система повинна забезпечувати розмежування прав доступу
ТВ10	Повинна бути реалізована можливість резервного копіювання

Для виконання вимог ТВ1 та ТВ2 у системі буде реалізовано механізми автентифікації користувачів та мандатного контролю доступу. Виконання вимог ТВ3 забезпечується використанням алгоритму хешування bcrypt. Для виконання вимог ТВ4 використовуватиметься протокол HTTPS із підтримкою TLS. Вимоги ТВ5 та ТВ6 забезпечуються реалізацією механізмів перевірки типу, розміру та вмісту файлів перед завантаженням до системи. Виконання ТВ7 реалізується шляхом ведення журналу подій користувачів, а ТВ8 – шляхом використання механізмів шифрування файлів. Для виконання вимог ТВ9 у системі буде реалізовано розмежування прав доступу між різними категоріями користувачів відповідно до рівня секретності інформації. Для забезпечення ТВ10 передбачено використання резервного копіювання даних.

Крім функціональних вимог, система повинна відповідати низці нефункціональних вимог, серед яких стабільність роботи, масштабованість, зручність використання та можливість подальшого розширення функціональності.

Таким чином, сформоване технічне завдання визначає основні функціональні та безпекові вимоги до системи захищеного управління файлами та створює основу для подальшого проєктування архітектури, структури бази даних і програмної реалізації вебсистеми.

2.2 Розробка архітектури вебсистеми

На основі сформованого технічного завдання було розроблено архітектуру вебсистеми захищеного управління файлами, яка забезпечує безпечну роботу користувачів із корпоративними даними, підтримує мандатну модель доступу та реалізує механізми захисту інформації відповідно до визначених технічних вимог.

Під час проєктування архітектури системи особливу увагу приділено питанням інформаційної безпеки, розмежування доступу до файлів та контролю дій користувачів. Архітектура системи побудована з використанням багаторівневого підходу, який передбачає розподіл функціональності між клієнтською частиною, серверною логікою та рівнем зберігання даних. Такий підхід забезпечує масштабованість системи, спрощує її супровід та дозволяє ізолювати окремі компоненти системи.

Серверна частина вебсистеми реалізується із використанням архітектурного шаблону MVC, який забезпечує розмежування програмної логіки, інтерфейсу користувача та механізмів роботи з даними. Це дозволяє підвищити структурованість системи та спростити реалізацію механізмів контролю доступу й аудиту дій користувачів.

Загальну структуру архітектури вебсистеми наведено на рисунку 2.3.

Рисунок 2.3 – Загальна архітектура вебсистеми

У межах розроблюваної системи центральне місце займає мандатна модель доступу, відповідно до якої кожному користувачу та файлу призначається певний рівень доступу. Доступ до інформації надається лише у випадку, якщо рівень допуску користувача відповідає рівню секретності файлу.

Для реалізації мандатної моделі доступу у системі передбачено п’ять рівнів секретності інформації, як показано на рисунку 2.4.

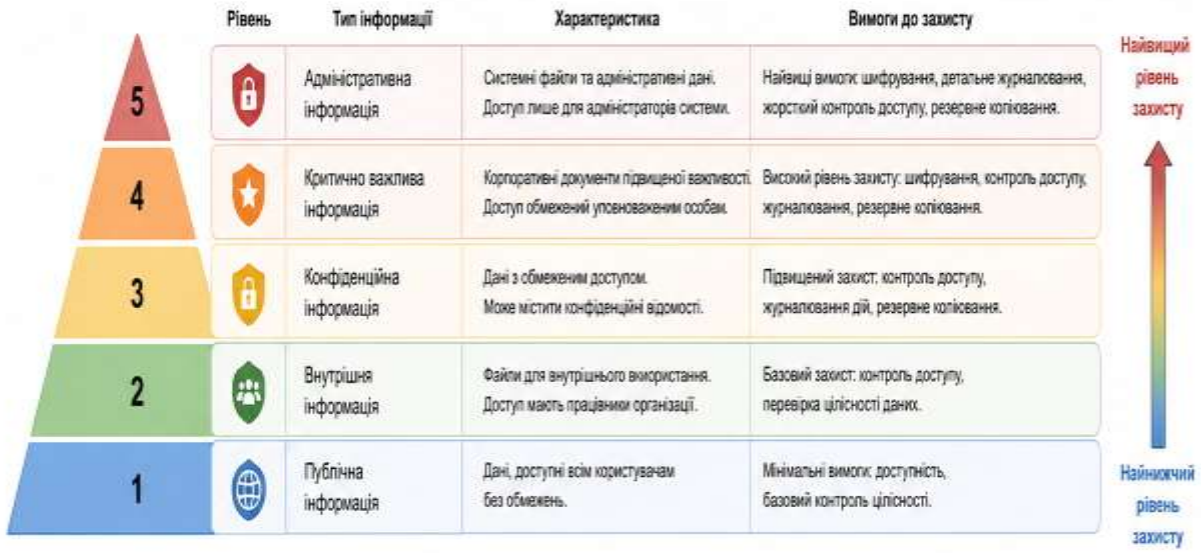


Рисунок 2.4 – Рівні секретності інформації у системі

Кожен файл у системі має власний рівень секретності, а кожен користувач – визначений рівень доступу. Це дозволяє централізовано контролювати можливість перегляду, завантаження, зміни або видалення файлів.

Для підвищення рівня інформаційної безпеки файли різного рівня секретності зберігаються у відокремлених директоріях файлової системи. Такий підхід дозволяє ізолювати дані різних категорій та зменшити ризики компрометації інформації у випадку несанкціонованого доступу. Організацію зберігання файлів відповідно до рівнів секретності наведено в таблиці 2.4.

Таблиця 2.4 – Організація зберігання файлів за рівнями секретності

Рівень секретності	Категорія файлів	Директорія зберігання
1	Публічні файли	public_storage
2	Внутрішні документи	internal_storage
3	Конфіденційні файли	confidential_storage
4	Критично важливі дані	critical_storage
5	Адміністративні файли	admin_storage

Такий підхід дозволяє забезпечити ізоляцію інформації різного рівня секретності та підвищити рівень захисту корпоративних даних.

У межах архітектури системи передбачено використання окремих механізмів захисту для протидії найбільш критичним загрозам, визначеним у технічному завданні. Основні архітектурні рішення щодо забезпечення інформаційної безпеки системи наведено в таблиці 2.5.

Початок таблиці 2.5 – Архітектурні рішення щодо забезпечення безпеки

Загроза	Архітектурне рішення
1	2
Несанкціонований доступ до файлів	Мандатна модель доступу (MAC)
Компрометація паролів	Хешування паролів алгоритмом bcrypt
SQL-ін’єкції	Використання ORM та валідації даних
XSS-атаки	Екранування даних у шаблонах
Перехоплення мережевого трафіку	Шифрування HTTPS та протокол TLS

Кінець таблиці 2.5

1	2
Втрата файлів	Регулярне резервне копіювання (Backup)
Неправомірні дії користувачів	Журналювання подій та аудит

Запропоновані механізми захисту дозволяють знизити ризик реалізації найбільш критичних загроз та забезпечити виконання основних вимог інформаційної безпеки.

Для забезпечення виконання функціональних можливостей системи було визначено основні сутності, які використовуються у структурі вебсистеми. Основні сутності, які використовуються у структурі вебсистеми, наведено в таблиці 2.6.

Таблиця 2.6 – Основні сутності вебсистеми

Сутність	Призначення
User	Зберігання інформації про користувачів (логін, хеш пароля, роль)
File	Зберігання метаданих про файли (назва, шлях, власник, розмір)
AccessLevel	Визначення прав доступу, що призначаються користувачам
SecurityLevel	Визначення рівня секретності, який присвоюється файлу
AuditLog	Реєстрація всіх подій у системі (вхід, читання, видалення, зміна прав)

Використання зазначених сутностей дозволяє реалізувати механізми автентифікації, контролю доступу, управління файлами та журналювання дій користувачів у межах розроблюваної системи.

Сутність User використовується для зберігання інформації про користувачів системи та їх рівні доступу. Сутність File містить інформацію про файли, їх власників та рівень секретності. Сутність AuditLog забезпечує журналювання дій користувачів, що дозволяє здійснювати аудит подій та контролювати активність у системі.

Клієнтська частина вебсистеми реалізується у вигляді вебінтерфейсу, який забезпечує взаємодію користувача із системою через браузер. Інтерфейс системи підтримує адаптивне відображення та дозволяє виконувати основні операції з

файлами: завантаження, перегляд, завантаження на пристрій, перейменування та видалення.

Взаємодія між клієнтською та серверною частинами здійснюється через захищене мережеве з'єднання із використанням протоколу HTTPS. Це дозволяє забезпечити захист інформації під час передачі даних між користувачем та сервером.

Для забезпечення контролю дій користувачів у системі реалізовано механізм журналювання подій. У журналі фіксуються операції входу до системи, завантаження файлів, перегляд інформації, видалення файлів та інші критично важливі дії користувачів. Це дозволяє підвищити рівень контролю та забезпечити можливість виявлення підозрілої активності.

Таким чином, розроблена архітектура вебсистеми забезпечує реалізацію мандатної моделі доступу, підтримує механізми захисту інформації та дозволяє організувати безпечну роботу користувачів із корпоративними файлами. Запропонований підхід забезпечує відповідність сформованим технічним вимогам та створює основу для подальшої реалізації структури бази даних і програмних компонентів системи.

2.3 Створення структури бази даних системи

Одним із ключових етапів проєктування системи захищеного управління файлами є створення структури бази даних, яка забезпечує зберігання інформації про користувачів, файли, рівні доступу та журнал подій системи. Від правильності побудови структури бази даних залежить ефективність роботи вебсистеми, швидкість обробки запитів, цілісність інформації та можливість подальшого розширення функціональності системи.

Під час проєктування структури бази даних особливу увагу приділено реалізації мандатної моделі доступу та забезпеченню механізмів захисту інформації відповідно до технічних вимог, сформованих у попередньому

підрозділі. Для реалізації системи використовується реляційна модель бази даних, яка дозволяє організувати зберігання інформації у вигляді взаємопов'язаних таблиць та забезпечити контроль цілісності даних.

У межах системи передбачено використання трьох основних таблиць: users, files та audit_logs. Таблиця users призначена для зберігання інформації про користувачів і містить дані, необхідні для автентифікації та визначення рівня доступу. Таблиця files зберігає метадані файлів, зокрема шлях до розташування та рівень секретності. Таблиця audit_logs використовується для фіксації дій користувачів і забезпечення аудиту подій у системі.

Центральне місце у структурі бази даних займають таблиці users та files, оскільки саме вони забезпечують реалізацію мандатної моделі доступу. Поле access_level у таблиці users визначає максимальний рівень доступу користувача, тоді як поле security_level у таблиці files визначає рівень секретності файлу.

Поля access_level та security_level є ключовими елементами реалізації мандатної моделі доступу. На основі значень цих параметрів система визначає можливість виконання операцій над файлами та доступу користувачів до інформації відповідного рівня секретності.

Для підвищення рівня інформаційної безпеки у системі використовується логічне розділення даних залежно від рівня секретності інформації. Дані користувачів, журнали подій та файли різного рівня секретності зберігаються у відокремлених структурах зберігання. Такий підхід дозволяє ізолювати інформацію різних категорій та зменшити ризики компрометації даних.

Організацію зберігання даних у системі наведено в таблиці 2.10.

Фізичні файли не зберігаються безпосередньо у базі даних. У таблиці files зберігаються лише метадані файлів, зокрема назва файлу, тип, розмір, шлях до розташування та рівень секретності. Такий підхід дозволяє зменшити навантаження на базу даних та підвищити ефективність роботи системи.

Для забезпечення контролю дій користувачів у системі використовується таблиця audit_logs, яка реалізує механізм журналювання подій. У журналі фіксуються операції входу до системи, завантаження файлів, перегляд інформації,

зміна даних та видалення файлів. Це дозволяє здійснювати аудит подій та контролювати активність користувачів у системі.

Між таблицями системи реалізовано логічні зв’язки, які забезпечують узгодженість та цілісність інформації. Один користувач може мати багато файлів, тому між таблицями users та files реалізовано зв’язок типу «один до багатьох». Аналогічно один користувач може мати багато записів у журналі подій, а один файл багато пов’язаних записів у таблиці audit_logs.

Для забезпечення цілісності даних використовуються первинні та зовнішні ключі. Це дозволяє уникнути появи некоректних або неузгоджених записів у базі даних та забезпечує правильність взаємозв’язків між сутностями системи.

Схему взаємозв’язків між таблицями бази даних наведено на рисунку 2.5.

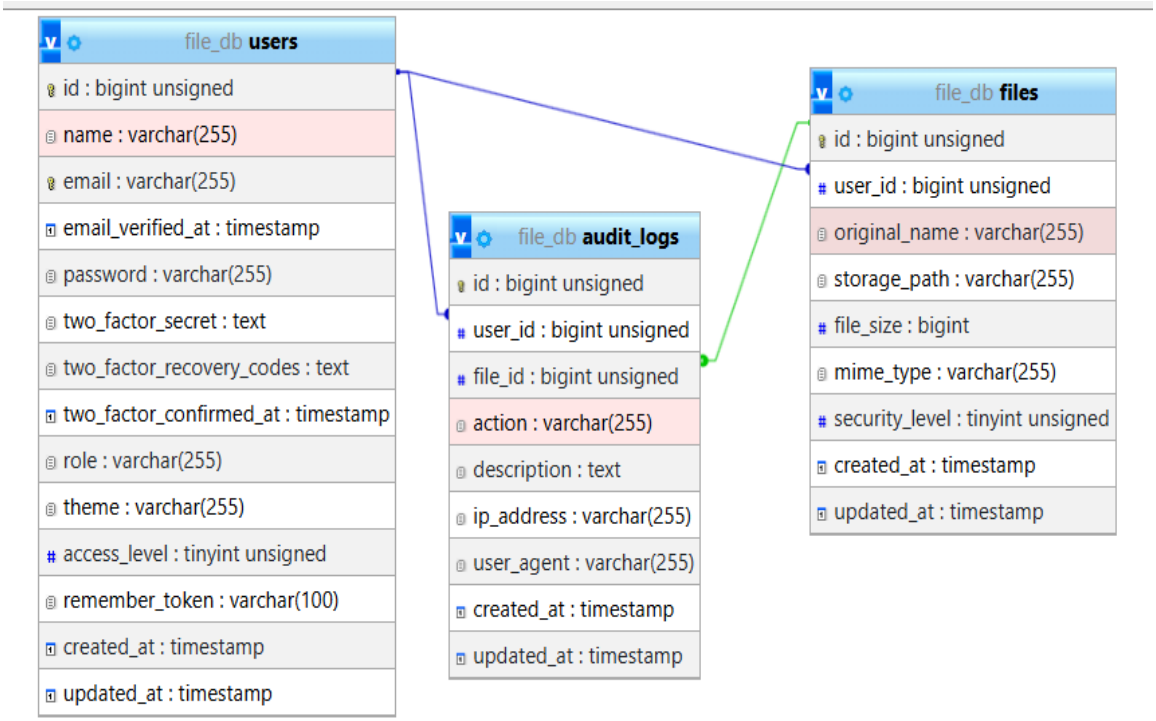


Рисунок 2.5 – Схема взаємозв’язків між таблицями

Запропонована структура бази даних забезпечує підтримку мандатної моделі доступу, реалізацію механізмів журналювання подій та безпечне зберігання інформації відповідно до рівня її секретності. Використання логічного розділення даних та контролю доступу дозволяє підвищити рівень інформаційної

безпеки системи та забезпечити захист корпоративних файлів від несанкціонованого доступу.

Таблиця 2.10 – Організація зберігання даних у системі

Категорія даних	Спосіб зберігання
Дані користувачів	users_db
Публічні файли	public_storage
Внутрішні файли	internal_storage
Конфіденційні файли	confidential_storage
Критично важливі файли	critical_storage
Адміністративні файли	admin_storage
Журнал подій	audit_storage

2.4 Формування цілісної структури вебсистеми

У попередніх підрозділах було сформовано технічне завдання на розробку системи захищеного управління файлами, визначено основні вимоги до інформаційної безпеки, розроблено архітектуру вебсистеми та структуру бази даних. Отримані результати дозволяють сформувати цілісну структуру вебсистеми, яка забезпечує взаємодію всіх компонентів системи та реалізацію необхідних механізмів захисту інформації.

Запропонована структура вебсистеми побудована з урахуванням мандатної моделі доступу та принципів багаторівневого захисту інформації. Центральними компонентами системи є модулі автентифікації користувачів, контролю доступу, управління файлами, журналювання подій та механізми захисту даних.

Користувач взаємодіє із системою через вебінтерфейс, який забезпечує можливість виконання основних операцій із файлами: завантаження, перегляду, завантаження на пристрій, перейменування та видалення. Після проходження автентифікації система визначає рівень доступу користувача та перевіряє можливість виконання операцій відповідно до мандатної моделі доступу.

У розроблюваній системі кожен файл має власний рівень секретності, а кожен користувач – визначений рівень допуску. Перед виконанням будь-якої операції система здійснює перевірку відповідності рівня доступу користувача рівню секретності файлу. Це дозволяє запобігти несанкціонованому доступу до корпоративної інформації.

Для забезпечення інформаційної безпеки у системі реалізуються механізми хешування паролів, захищеної передачі даних через HTTPS, перевірки файлів перед завантаженням, журналювання подій та резервного копіювання інформації. Крім того, використовується логічне розділення даних залежно від рівня їх секретності, що дозволяє підвищити рівень ізоляції інформації та зменшити ризики компрометації даних.

Загальну структуру взаємодії компонентів вебсистеми наведено на рисунку 2.5.

Рисунок 2.5 – Цілісна структура вебсистеми захищеного управління файлами

У межах системи взаємодія між компонентами здійснюється через серверну

частину вебзастосунку, яка забезпечує обробку запитів користувачів, перевірку прав доступу та взаємодію з базою даних і файловими сховищами. Журналювання подій дозволяє фіксувати критично важливі дії користувачів та забезпечує можливість аудиту безпеки системи.

Запропонована структура вебсистеми забезпечує реалізацію всіх основних функціональних та безпекових механізмів, визначених у технічному завданні. Використання мандатної моделі доступу, розділення даних за рівнями секретності та механізмів контролю доступу дозволяє забезпечити захист корпоративної інформації відповідно до визначених вимог інформаційної безпеки.

Таким чином, програмна реалізація системи за запропонованою структурою дозволить забезпечити виконання сформованих технічних вимог, реалізацію механізмів захисту інформації та безпечну роботу користувачів із корпоративними файлами у рамках вебсистеми.

2.5 Висновок

У другому розділі було виконано проєктування системи захищеного управління файлами відповідно до сформованих технічних вимог та визначених загроз інформаційній безпеці. На основі результатів аналізу предметної області сформовано технічне завдання, у межах якого визначено основні функціональні та безпекові вимоги до майбутньої вебсистеми.

У процесі проєктування особливу увагу приділено реалізації механізмів інформаційної безпеки та мандатної моделі доступу. Визначено рівні секретності інформації, сформовано модель потенційного порушника та встановлено пріоритетність основних загроз, що дозволило обґрунтувати вибір механізмів захисту даних і методів контролю доступу в системі. Також було враховано необхідність забезпечення конфіденційності, цілісності та доступності інформації під час роботи користувачів із файлами різних рівнів секретності.

Було розроблено архітектуру вебсистеми, яка забезпечує взаємодію між

клієнтською частиною, серверною логікою, базою даних та файловими сховищами. Запропонована архітектура підтримує механізми автентифікації, авторизації, контролю доступу, журналювання подій та безпечного зберігання файлів відповідно до рівня їх секретності. Крім того, архітектурні рішення передбачають можливість подальшого масштабування системи та інтеграції додаткових засобів захисту інформації.

У процесі проєктування структури бази даних визначено основні таблиці системи, логічні зв’язки між ними та механізми реалізації мандатного контролю доступу. Для підвищення рівня захисту інформації запропоновано логічне розділення даних і файлових сховищ залежно від рівня секретності інформації, що дозволяє мінімізувати ризики несанкціонованого доступу та витоку даних. Також передбачено механізми ведення журналів аудиту для фіксації дій користувачів і подальшого аналізу подій безпеки.

Окрему увагу приділено формуванню цілісної структури вебсистеми, яка об’єднує всі основні компоненти та забезпечує їх узгоджену взаємодію. Використання запропонованих архітектурних рішень дозволяє забезпечити ефективний контроль доступу до інформації, захист корпоративних файлів від несанкціонованого доступу, а також можливість моніторингу й аудиту дій користувачів у системі. Запропоновані підходи також сприяють підвищенню надійності та стійкості системи до потенційних кіберзагроз.

Таким чином, результати другого розділу формують завершене проєктне підґрунтя для подальшої програмної реалізації системи захищеного управління файлами. Реалізація вебсистеми за запропонованою схемою дозволить забезпечити виконання сформованих технічних вимог, реалізацію основних механізмів інформаційної безпеки у корпоративному середовищі та створення надійного програмного засобу для безпечного зберігання й обміну файлами.

3. РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБСИСТЕМИ НА БАЗІ LARAVEL

3.1 Програмна реалізація підсистем автентифікації, авторизації та розмежування прав доступу.

У процесі розробки системи захищеного управління файлами важливим етапом є реалізація підсистем автентифікації, авторизації та розмежування прав доступу. Дані підсистеми забезпечують контроль доступу до функціоналу системи та гарантують, що кожен користувач може виконувати лише дозволені йому дії. Реалізація зазначених механізмів здійснювалася відповідно до технічних вимог ТВ1, ТВ2, ТВ3, ТВ7 та ТВ9, визначених у другому розділі.

Для реалізації серверної частини вебсистеми використано фреймворк Laravel, який забезпечує підтримку механізмів автентифікації, авторизації, роботи із сесіями користувачів та взаємодії з базою даних. Використання Laravel дозволило реалізувати структуру системи відповідно до розробленої архітектури та забезпечити підтримку механізмів інформаційної безпеки.

У межах програмної реалізації використано архітектурний підхід MVC, що дозволяє розділити логіку роботи системи, інтерфейс користувача та механізми взаємодії з базою даних. Для взаємодії із базою даних використовувалися ORM-механізми Laravel Eloquent, що дозволяє мінімізувати ризик SQL-ін’єкцій за рахунок використання параметризованих запитів та автоматизованої обробки даних.

На рисунку 3.1 представлено структуру програмного проєкту, реалізованого із використанням фреймворку Laravel.

Підсистема автентифікації реалізована із використанням вбудованих можливостей Laravel. У системі передбачено механізми реєстрації користувачів, входу до системи, завершення сеансу роботи та зміни пароля користувача. Під час реєстрації користувач вводить свої облікові дані, які проходять перевірку коректності введення, унікальності електронної адреси та відповідності встановленим вимогам безпеки.

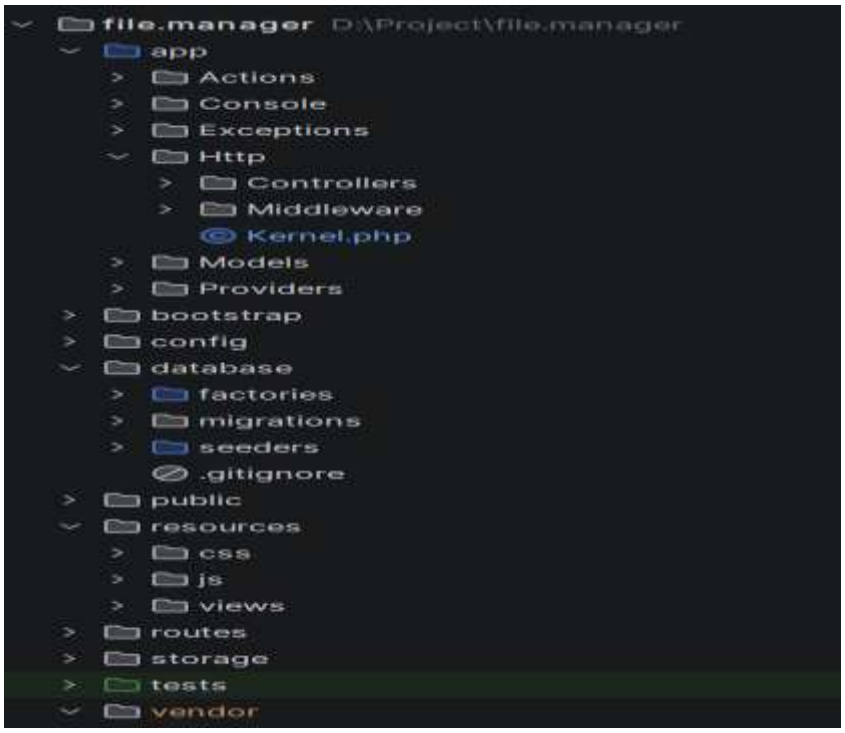


Рисунок 3.1 – Структура програмного проєкту Laravel

Для забезпечення виконання вимоги ТВ3 паролі користувачів не зберігаються у відкритому вигляді. Для їх захисту використовується алгоритм хешування bcrypt, який забезпечує безпечне зберігання облікових даних та зменшує ризик компрометації паролів у випадку несанкціонованого доступу до бази даних.

Інтерфейс сторінки входу до системи наведено на рисунку 3.2.

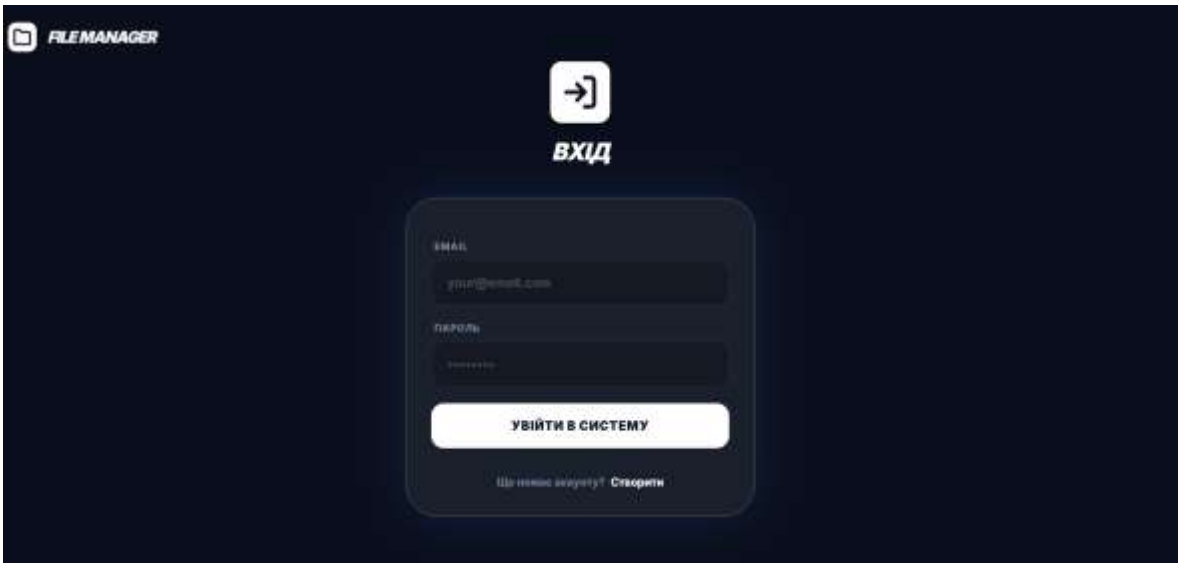


Рисунок 3.2 – Сторінка входу до системи

Форму реєстрації нового користувача, у рамках якої здійснюється створення облікового запису та перевірка введених даних, наведено на рисунку 3.3. Якщо користувач не має облікового запису в системі, він може пройти процедуру реєстрації шляхом введення імені, електронної адреси та пароля. Після успішного проходження перевірки введених даних система автоматично створює новий обліковий запис користувача та надає базовий рівень доступу відповідно до мандатної моделі безпеки.

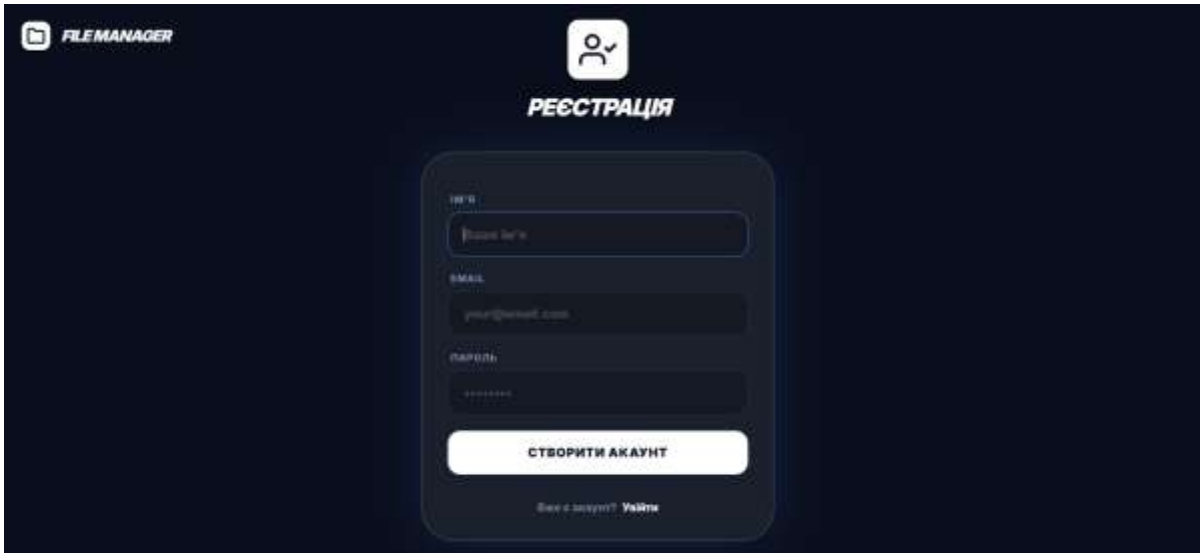


Рисунок 3.3 – Форма реєстрації користувача

Після успішної автентифікації користувач перенаправляється до основного інтерфейсу системи (dashboard), де може виконувати доступні йому операції. Для перевірки доступу до захищених маршрутів використовується middleware auth, який обмежує доступ до функціоналу системи для неавторизованих користувачів.

На рисунку 3.4 представлено головний інтерфейс системи після входу користувача.

Для розмежування доступу до файлових ресурсів у системі реалізовано мандатну модель контролю доступу. У межах даної моделі доступ користувачів до інформації визначається відповідно до рівня допуску користувача та рівня секретності файлу.

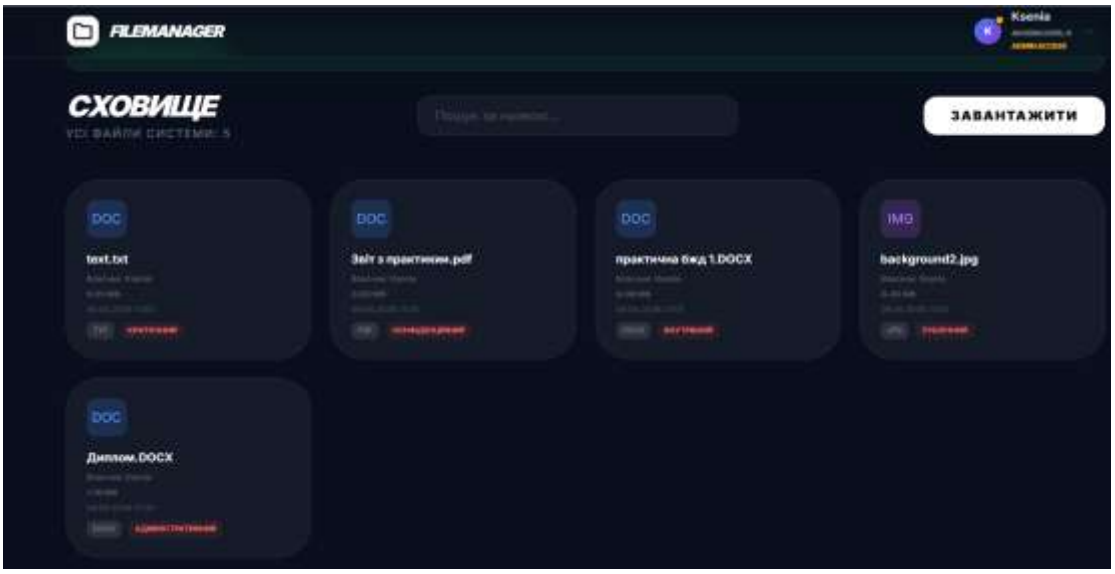


Рисунок 3.4 – Головний інтерфейс системи

Кожен користувач системи має власний рівень доступу (access_level), а кожен файл характеризується визначеним рівнем секретності (security_level). Доступ до файлів надається лише у випадку, якщо рівень доступу користувача є більшим або дорівнює рівню секретності файлу.

Перед виконанням будь-якої операції над файлом система автоматично виконує перевірку відповідності рівня доступу користувача рівню секретності файлу. Якщо рівень доступу є недостатнім, виконання операції блокується, а користувач отримує повідомлення про відмову в доступі.

Схему перевірки прав доступу користувача наведено на рисунку 3.5.

Рисунок 3.5 – Схема перевірки прав доступу користувача

Для реалізації контролю доступу використовуються middleware-компоненти, які виконують перевірку автентифікації користувача, ролі користувача та рівня доступу перед обробкою запиту. Такий підхід дозволяє централізовано контролювати доступ до функціональних модулів системи та забезпечує виконання вимог мандатної моделі доступу.

На рисунку 3.6 представлено реалізацію middleware-компонентів у структурі програмного проєкту.

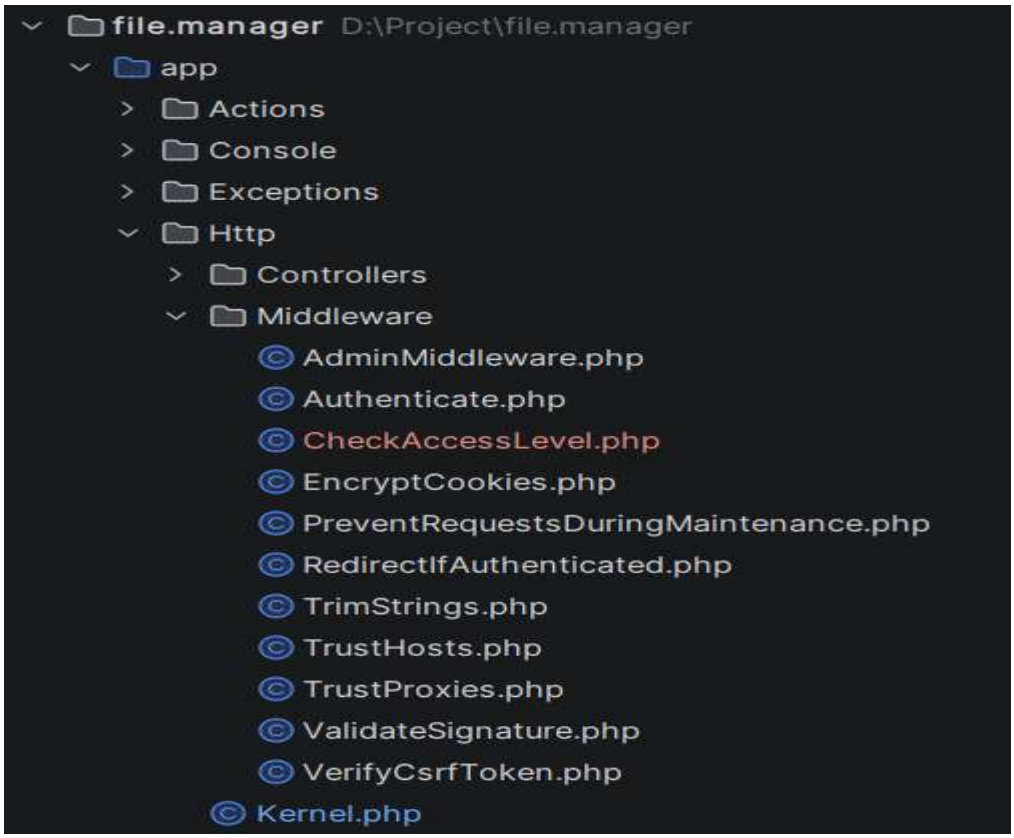


Рисунок 3.6 – Middleware-компоненти системи

Middleware-компоненти використовуються для централізованого контролю доступу до функціональних модулів системи. Зокрема, middleware Authenticate забезпечує перевірку автентифікації користувача, middleware AdminMiddleware виконує перевірку адміністративних прав доступу, а middleware CheckAccessLevel реалізує механізм мандатного контролю доступу відповідно до рівня допуску користувача та рівня секретності файлу.

У рамках даної системи реалізовано дві основні категорії користувачів –

адміністратор системи та зареєстрований користувач.

Адміністратор має доступ до управління користувачами, журналами подій, файлами та налаштуваннями системи. Зареєстровані користувачі можуть працювати лише з файлами, рівень секретності яких не перевищує їхній рівень доступу.

Для реалізації адміністративного функціоналу створено окрему адміністративну панель, доступ до якої мають лише користувачі з роллю адміністратора. У межах адміністративної панелі реалізовано механізми централізованого управління користувачами, зокрема перегляд списку акаунтів, зміну ролі користувача, зміну рівня доступу та видалення облікового запису. Також у панелі адміністратора відображається статистична інформація про кількість користувачів, файлів, загальний обсяг сховища та журнал останніх дій користувачів. Адміністративний інтерфейс системи наведено на рисунку 3.7.

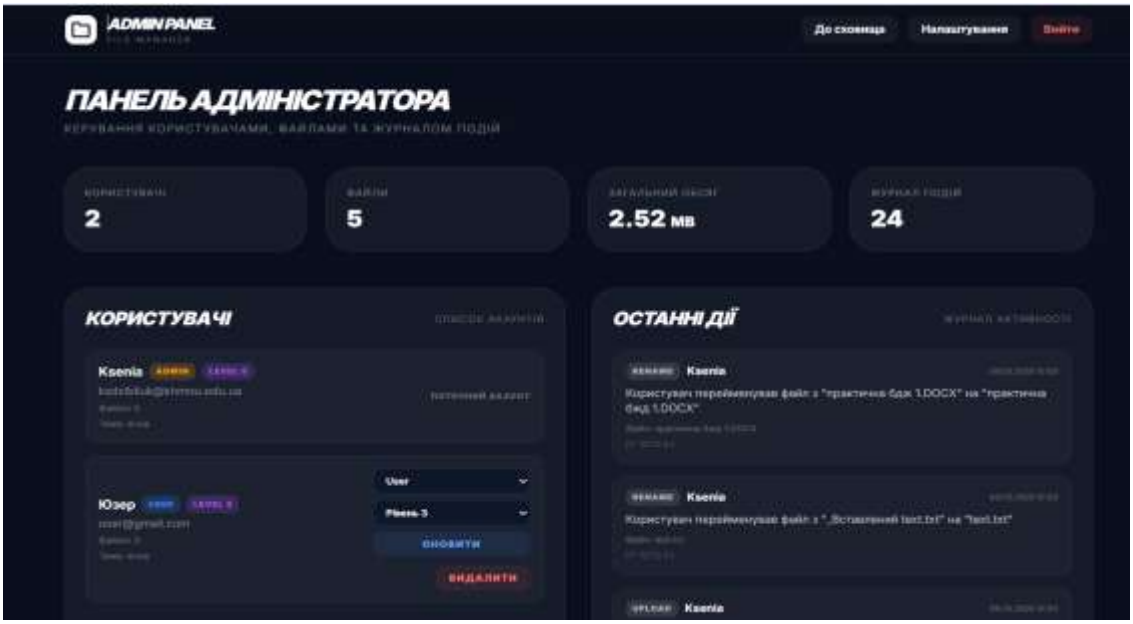


Рисунок 3.7 – Адміністративний інтерфейс системи та керування користувачами

У структурі таблиці users реалізовано поля role та access_level, які використовуються для підтримки мандатної моделі контролю доступу. Для кожного файлу у таблиці files визначається рівень секретності security_level, що використовується під час перевірки доступу користувачів до файлових ресурсів.

Структуру таблиць бази даних наведено на рисунку 3.9.

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Змінити Видалити Більше
2	name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
3	email	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
4	email_verified_at	timestamp			Так	NULL			Змінити Видалити Більше
5	password	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
6	two_factor_secret	text	utf8mb4_unicode_ci		Так	NULL			Змінити Видалити Більше
7	two_factor_recovery_codes	text	utf8mb4_unicode_ci		Так	NULL			Змінити Видалити Більше
8	two_factor_confirmed_at	timestamp			Так	NULL			Змінити Видалити Більше
9	role	varchar(255)	utf8mb4_unicode_ci		Ні	user			Змінити Видалити Більше
10	theme	varchar(255)	utf8mb4_unicode_ci		Ні	dark			Змінити Видалити Більше
11	access_level	tinyint		UNSIGNED	Ні	1			Змінити Видалити Більше
12	remember_token	varchar(100)	utf8mb4_unicode_ci		Так	NULL			Змінити Видалити Більше
13	created_at	timestamp			Так	NULL			Змінити Видалити Більше
14	updated_at	timestamp			Так	NULL			Змінити Видалити Більше

#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Змінити Видалити Більше
2	user_id	bigint		UNSIGNED	Ні	Немає			Змінити Видалити Більше
3	original_name	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
4	storage_path	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
5	file_size	bigint			Ні	Немає			Змінити Видалити Більше
6	mime_type	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Змінити Видалити Більше
7	security_level	tinyint		UNSIGNED	Ні	1			Змінити Видалити Більше
8	created_at	timestamp			Так	NULL			Змінити Видалити Більше
9	updated_at	timestamp			Так	NULL			Змінити Видалити Більше

Рисунок 3.9 – Структура таблиць users та files

Організація маршрутів у системі побудована таким чином, що доступ до більшості функцій обмежений механізмами автентифікації та контролю доступу. Доступ до користувацького функціоналу здійснюється лише після проходження автентифікації, а доступ до адміністративних функцій – виключно для користувачів із роллю адміністратора. Схему маршрутизації та контролю доступу вебсистеми наведено на рисунку 3.10.

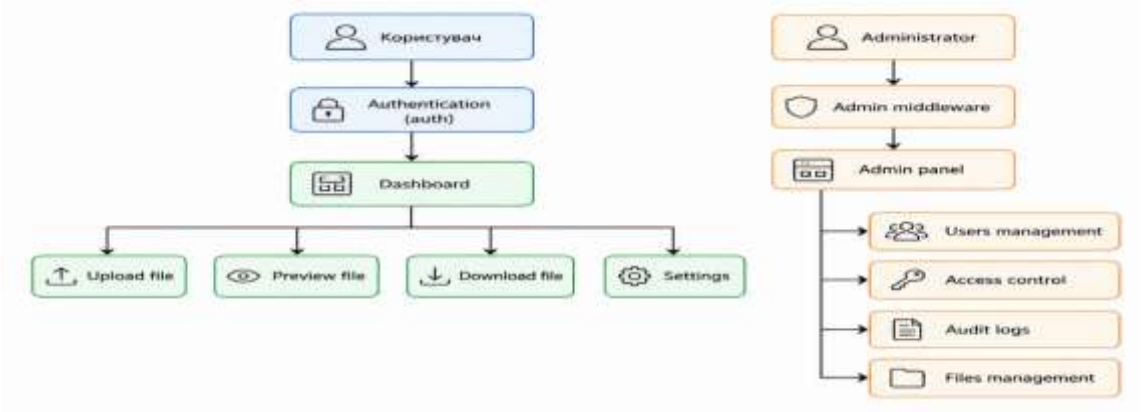


Рисунок 3.10 – Схема маршрутизації та контролю доступу системи

Додатково у системі реалізовано можливість персоналізації інтерфейсу користувача, зокрема зміну теми оформлення. Користувач може обирати між світлою та темною темами, що дозволяє адаптувати зовнішній вигляд системи відповідно до власних уподобань. Обрана тема зберігається у базі даних користувача та автоматично застосовується при наступному вході в систему.

На рисунку 3.11 представлено приклад зміни теми інтерфейсу користувача.

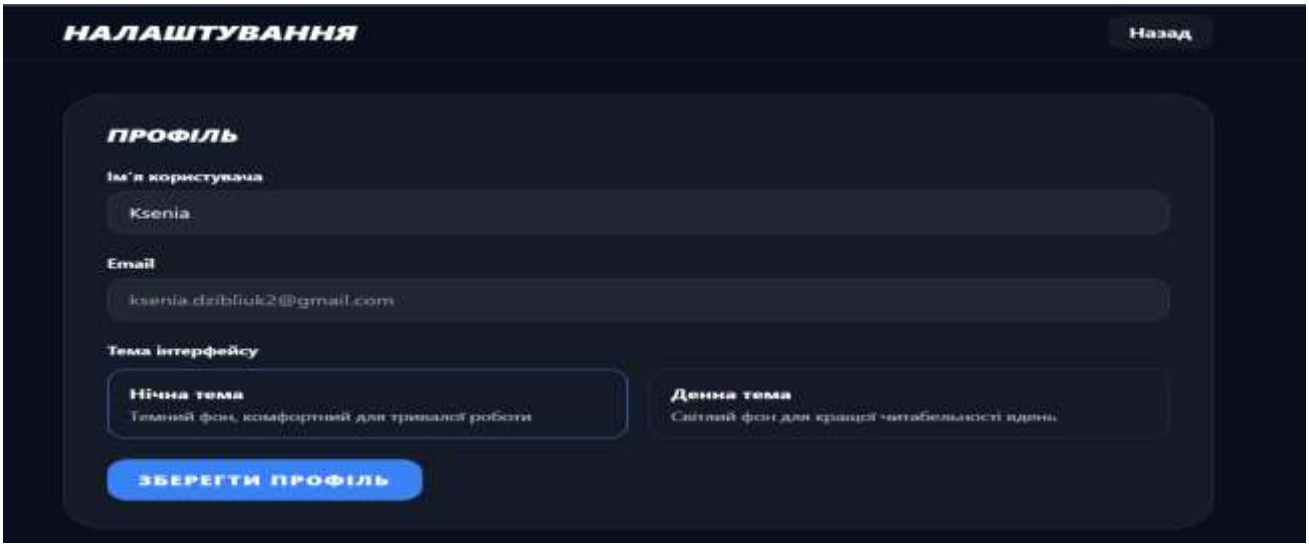


Рисунок 3.11 – Зміна теми інтерфейсу користувача

У межах підсистеми автентифікації також реалізовано можливість зміни пароля користувача. Для зміни пароля користувач повинен ввести поточний пароль, а також новий пароль із підтвердженням, для чого було запропоновано відповідну графічну форму. Перед збереженням виконується перевірка правильності введених даних та відповідність нового пароля встановленим вимогам безпеки. На рисунку 3.12 представлено інтерфейс зміни пароля користувача.

Для підвищення рівня інформаційної безпеки у системі реалізовано механізм автоматичного завершення сесії після тривалого періоду неактивності користувача. Це дозволяє зменшити ризик отримання несанкціонованого доступу до системи у випадку залишення активної сесії без нагляду.

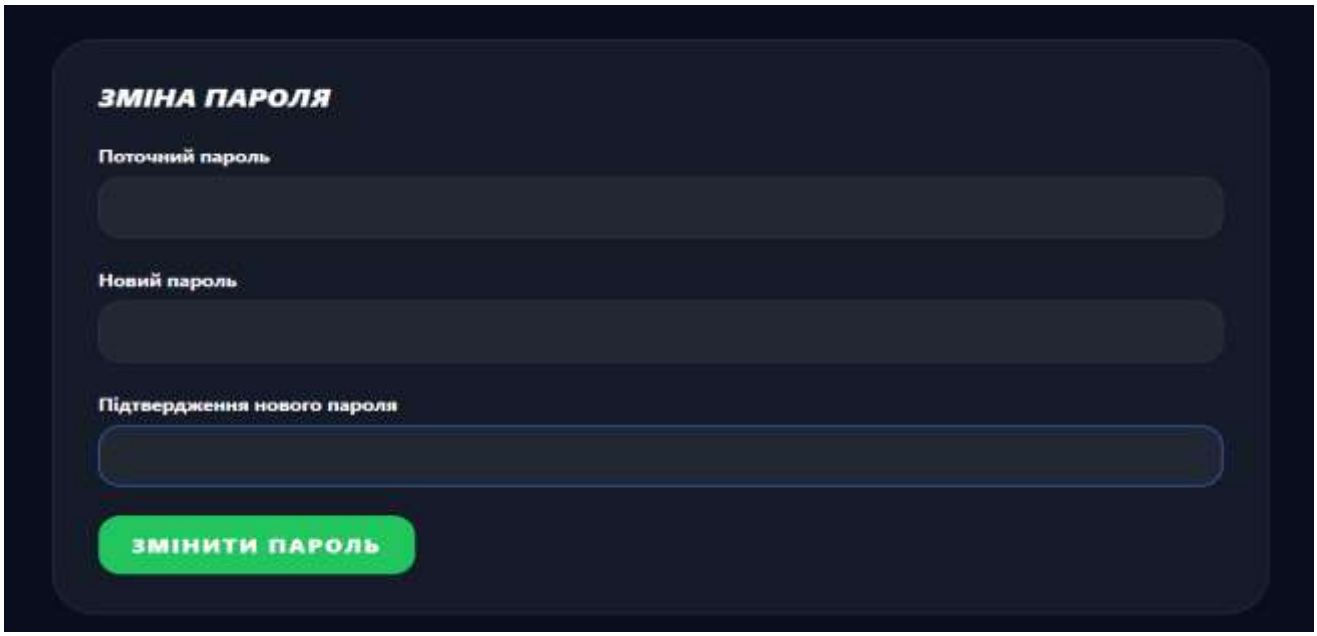


Рисунок 3.12 – Форма зміни пароля користувача

Для забезпечення захисту від типових вебзагроз у системі реалізовано перевірку вхідних даних, захист від CSRF-атак, контроль сесій користувачів та екранування даних під час відображення інформації у вебінтерфейсі. Використання зазначених механізмів дозволяє знизити ризик реалізації SQL-ін’єкцій, XSS-атак та інших загроз інформаційній безпеці.

Таким чином, у межах даного підрозділу реалізовано підсистеми автентифікації, авторизації та розмежування прав доступу, які забезпечують безпечну взаємодію користувачів із вебсистемою та підтримують виконання технічних вимог ТВ1, ТВ2, ТВ3, ТВ7 та ТВ9 щодо захисту корпоративної інформації.

3.2 Розробка модулів захищеного завантаження, шифрування та зберігання контенту

Відповідно до технічних вимог ТВ4, ТВ5, ТВ6, ТВ8 та ТВ10 у системі реалізовано підсистему захищеного завантаження, шифрування та зберігання файлів. Основним призначенням даної підсистеми є забезпечення безпечного

прийому файлів від користувачів, їх перевірки, збереження та контролю доступу до корпоративної інформації.

Функціонал роботи з файлами охоплює завантаження, перегляд, завантаження на пристрій користувача, перейменування та видалення файлів. Доступ до виконання зазначених операцій надається лише авторизованим користувачам з урахуванням їх рівня доступу та рівня секретності файлу.

Загальний процес обробки файлу під час його завантаження до системи наведено на рисунку 3.13.

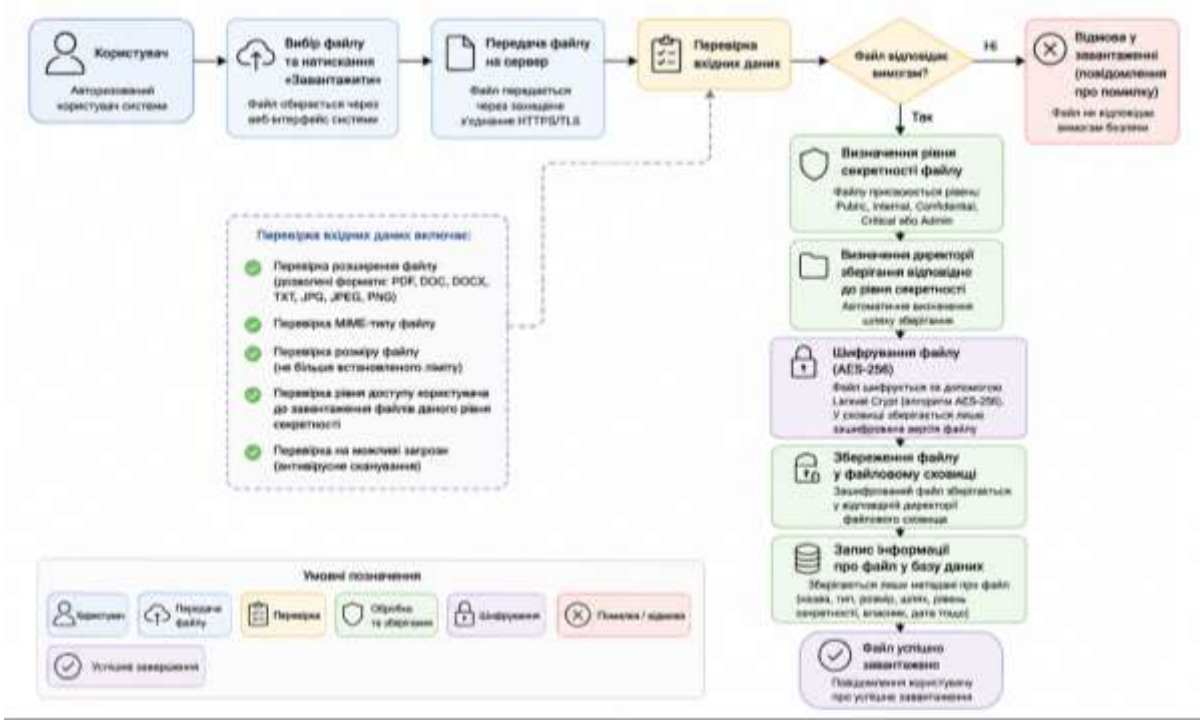


Рисунок 3.13 – Схема процесу завантаження та збереження файлу

Перед збереженням система виконує перевірку вхідних даних. Для забезпечення виконання вимог ТВ5 та ТВ6 реалізовано механізм валідації файлів, який включає перевірку дозволених форматів, MIME-типу та максимального розміру файлу. Це дозволяє запобігти завантаженню небезпечних файлів та перевантаженню файлового сховища.

У розроблюваній системі дозволено завантаження файлів форматів PDF, DOC, DOCX, TXT, JPG, JPEG та PNG. Крім цього, система виконує перевірку MIME-типу файлу, що дозволяє знизити ризик завантаження виконуваних або

потенційно небезпечних файлів.

Кожному файлу під час завантаження присвоюється рівень секретності, який визначає директорію зберігання та доступ до файлу. У системі реалізовано декілька рівнів секретності:

– Public;

– Internal;

– Confidential;

– Critical;

– Admin.

Для забезпечення ізольованого зберігання інформації реалізовано окрему структуру файлового сховища. Файли різних рівнів секретності автоматично зберігаються у відповідних директоріях файлової системи сервера.

Структуру файлового сховища системи наведено на рисунку 3.14.

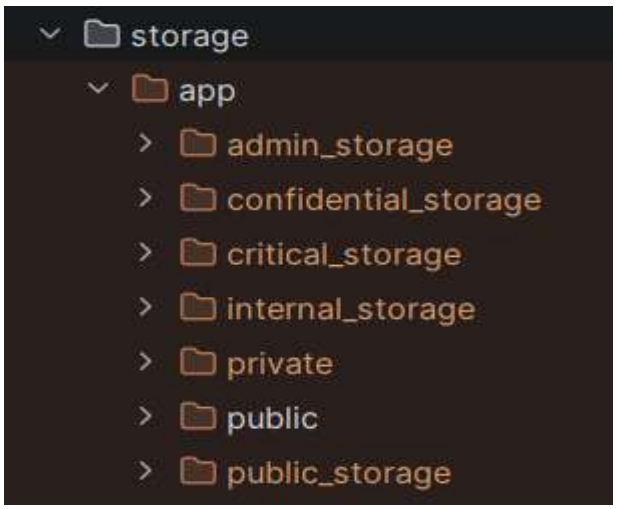


Рисунок 3.14 – Структура файлового сховища системи

Після успішного збереження файлу у файловому сховищі система автоматично фіксує службову інформацію про нього у базі даних. Для кожного файлу зберігаються назва, MIME-тип, розмір, шлях зберігання, рівень секретності та інформація про власника файлу.

Для забезпечення виконання вимоги ТВ8 у системі реалізовано механізм симетричного шифрування файлів перед їх збереженням у файловому сховищі.

Під час завантаження файл проходить процедуру шифрування, після чого у файловій системі сервера зберігається лише його зашифрована версія.

Для реалізації криптографічного захисту використано вбудований механізм Laravel Crypt, який базується на алгоритмі AES-256. Ключ шифрування формується автоматично в межах конфігурації Laravel та зберігається у захищеному середовищі застосунку. Схему процесу шифрування зображено на рисунку 3.15.

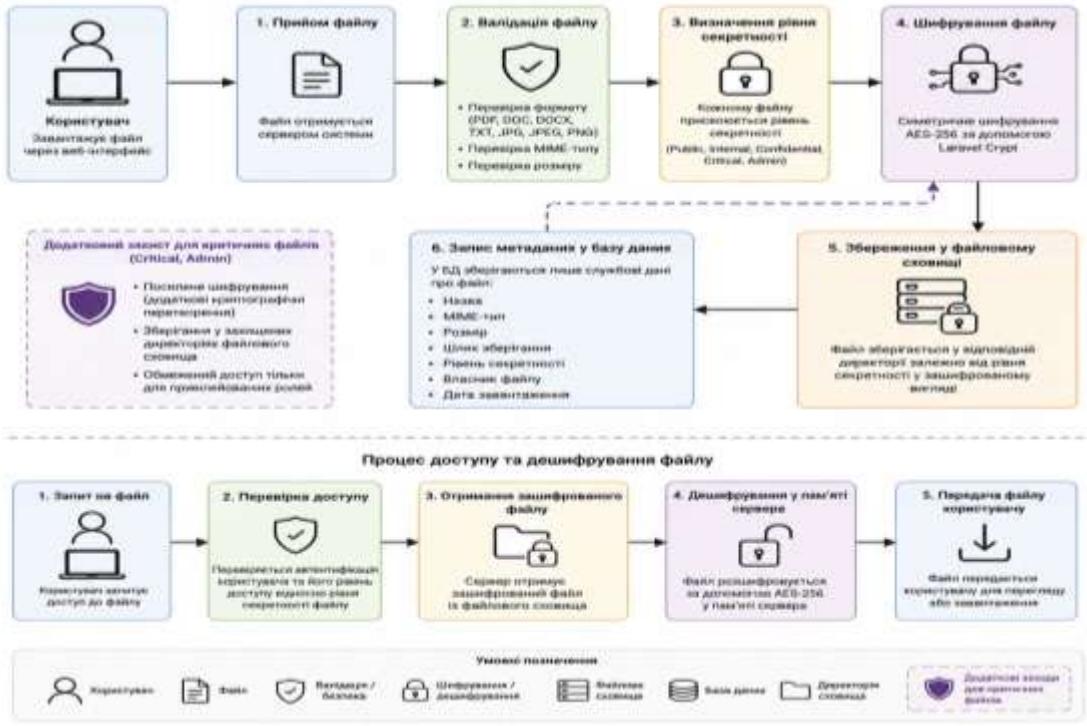


Рисунок 3.15 – Схема шифрування файлу

Для файлів із високим рівнем секретності (Critical та Admin) у системі реалізовано додатковий механізм криптографічного захисту. Такі файли проходять посилену процедуру шифрування перед збереженням та зберігаються виключно у захищених директоріях файлового сховища. Це дозволяє підвищити рівень захисту особливо критичної корпоративної інформації та зменшити ризик її компрометації у випадку несанкціонованого доступу до сервера.

Під час отримання доступу до файлу система виконує процедуру дешифрування даних у пам’яті сервера та лише після успішної перевірки прав

доступу передає файл користувачу. Такий підхід дозволяє зменшити ризик компрометації інформації у випадку несанкціонованого доступу до файлового сховища сервера.

Інформація про файлові ресурси використовується для реалізації контролю доступу та управління файлами у системі. Операції завантаження, перегляду, скачування, перейменування та видалення файлів фіксуються у журналі подій системи, що дозволяє забезпечити аудит дій користувачів та підвищити рівень контролю за використанням корпоративної інформації.

Для забезпечення виконання вимоги ТВ8 реалізовано механізм захищеного зберігання файлів. Під час доступу до файлу система виконує перевірку рівня доступу користувача відносно рівня секретності файлу. Якщо рівень доступу користувача є недостатнім, система блокує виконання операції та забороняє доступ до інформації.

Для додаткового захисту файлових ресурсів доступ до файлових маршрутів системи обмежується middleware-компонентами, що дозволяє централізовано контролювати доступ до операцій роботи з файлами. Крім цього, прямий доступ до файлів через URL-адресу сервера обмежується. Доступ до файлових ресурсів здійснюється виключно через серверну логіку застосунку після проходження перевірки автентифікації та рівня доступу користувача. Такий підхід дозволяє знизити ризик несанкціонованого отримання доступу до інформації шляхом прямого звернення до файлового сховища.

Інтерфейс завантаження файлів наведено на рисунку 3.16.

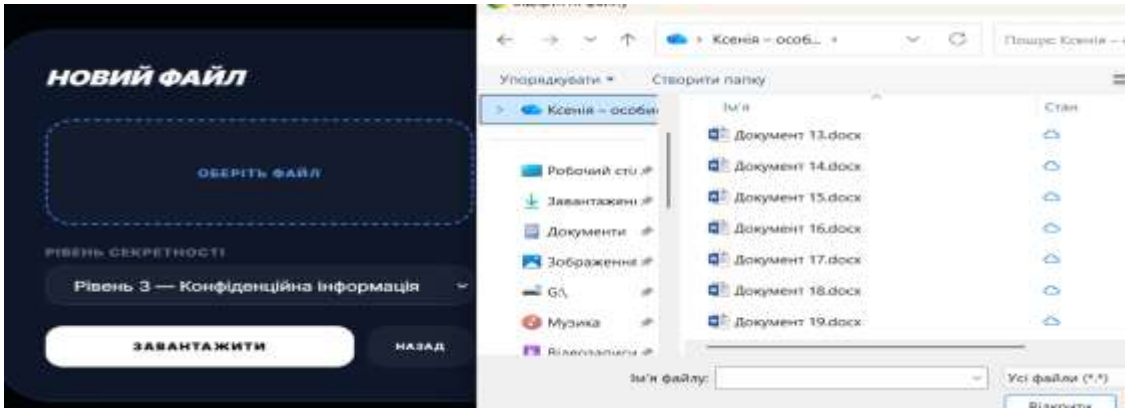


Рисунок 3.16 – Інтерфейс завантаження файлу

Після успішного завантаження користувач отримує можливість перегляду списку власних файлів та виконання дозволених операцій: перегляду, завантаження, перейменування або видалення і також здійснювати пошук файлів за їх назвою.

Інтерфейс управління файлами наведено на рисунку 3.17.

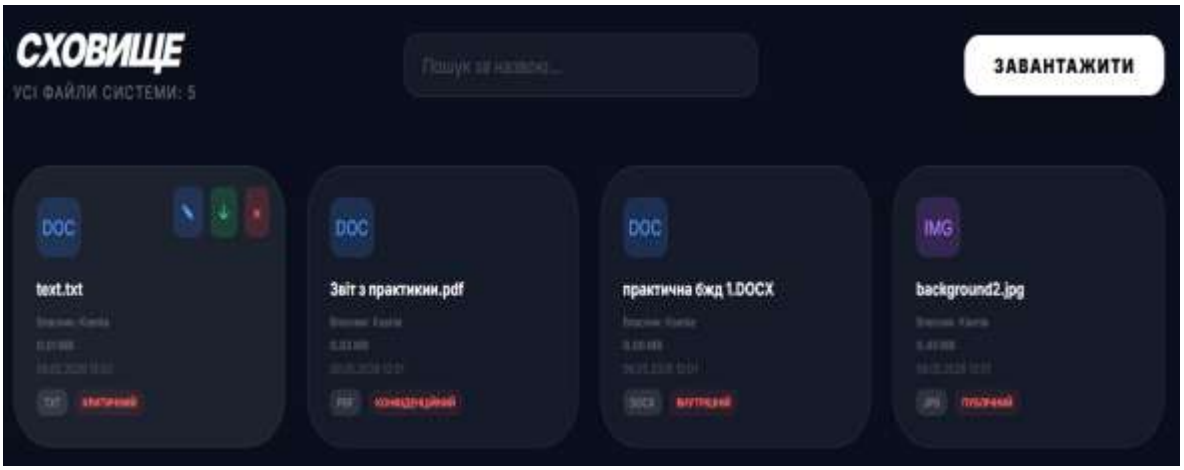


Рисунок 3.17 – Інтерфейс управління файлами

Для забезпечення виконання вимоги ТВ10 у системі реалізовано механізм резервного копіювання бази даних та файлового сховища. Резервні копії можуть використовуватись для відновлення інформації у випадку програмних або апаратних збоїв.

Крім цього, передача файлів між клієнтом та сервером здійснюється із використанням захищеного мережевого з’єднання HTTPS/TLS, що дозволяє знизити ризик перехоплення інформації під час передачі мережею. Передача даних через захищений канал зв’язку дозволяє забезпечити конфіденційність інформації та знизити ризик її перехоплення під час мережевої взаємодії.

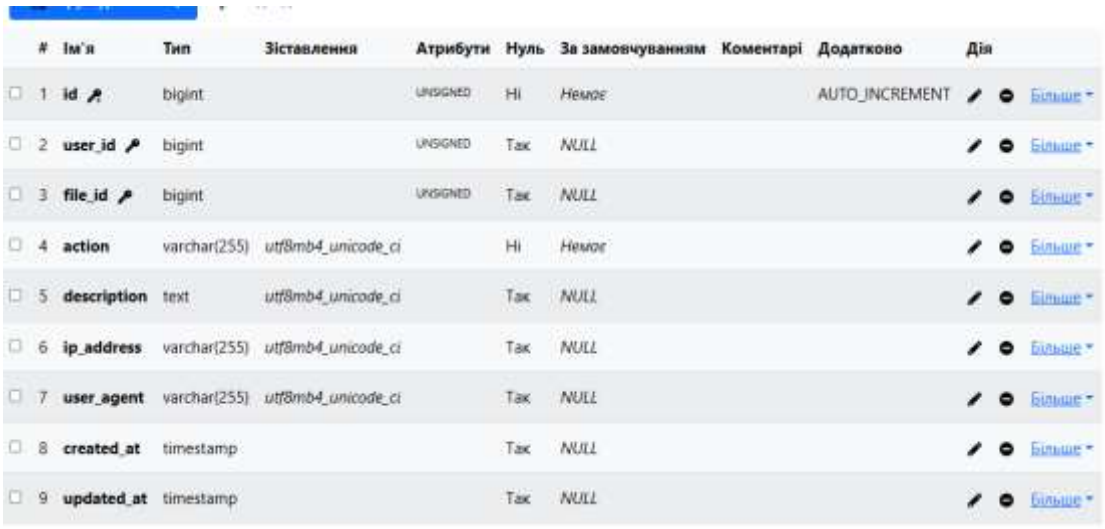
Таким чином, у межах даного підрозділу реалізовано модулі захищеного завантаження та зберігання контенту, які забезпечують безпечну роботу користувачів із файловими ресурсами та підтримують виконання технічних вимог ТВ4, ТВ5, ТВ6, ТВ8 та ТВ10 щодо захисту корпоративної інформації.

3.3 Впровадження підсистем аудиту, логування та забезпечення спостережності

Відповідно до технічної вимоги ТВ7 у системі реалізовано підсистему аудиту, логування та спостережності, призначену для фіксації дій користувачів і контролю подій, пов’язаних із безпекою вебсистеми. Реалізація даної підсистеми дозволяє відстежувати основні операції користувачів, аналізувати активність у системі та виявляти потенційно підозрілі дії.

У рамках підсистеми аудиту здійснюється журналювання основних дій користувачів під час роботи із системою. До журналу подій заносяться реєстрація користувача, успішний вхід до системи, невдала спроба входу, завершення сеансу роботи, завантаження файлів, перегляд файлів, скачування файлів, перейменування, видалення файлів, а також зміна рівня доступу користувачів адміністратором.

Для реалізації журналювання використовується окрема таблиця audit_logs, у якій зберігається інформація про користувача, пов’язаний файл, тип виконаної дії, опис події, IP-адресу, User-Agent браузера та час виконання операції. Такий підхід дозволяє централізовано зберігати інформацію про події безпеки та використовувати її для подальшого аналізу. Структуру таблиці audit_logs наведено на рисунку 3.18.



#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
1	id	bigint		UNSIGNED	Ні	Немає		AUTO_INCREMENT	Більше
2	user_id	bigint		UNSIGNED	Так	NULL			Більше
3	file_id	bigint		UNSIGNED	Так	NULL			Більше
4	action	varchar(255)	utf8mb4_unicode_ci		Ні	Немає			Більше
5	description	text	utf8mb4_unicode_ci		Так	NULL			Більше
6	ip_address	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше
7	user_agent	varchar(255)	utf8mb4_unicode_ci		Так	NULL			Більше
8	created_at	timestamp			Так	NULL			Більше
9	updated_at	timestamp			Так	NULL			Більше

Рисунок 3.18 – Структура таблиці audit_logs

Під час виконання критично важливих операцій система автоматично створює відповідний запис у журналі подій. Зокрема, під час успішної автентифікації фіксується факт входу користувача до системи, а у випадку неправильного введення облікових даних створюється запис про невдалу спробу входу. Це дозволяє адміністратору виявляти підозрілу активність, наприклад повторні спроби підбору пароля або доступу до чужого облікового запису.

Окремо журналюються операції, пов’язані з файловими ресурсами. Під час завантаження, перегляду, скачування, перейменування або видалення файлу система зберігає інформацію про користувача, дію та файл, з яким вона була виконана. Це забезпечує можливість відстеження життєвого циклу файлу та контролю дій користувачів із корпоративною інформацією.

Для підвищення рівня спостережності система також фіксує технічну інформацію про середовище виконання запиту. Зокрема, у журналі подій зберігаються IP-адреса користувача та інформація про браузер або клієнтське середовище. Це дозволяє аналізувати джерела активності та виявляти нетипові звернення до системи. Інтерфейс перегляду журналу подій наведено на рисунку 3.19.

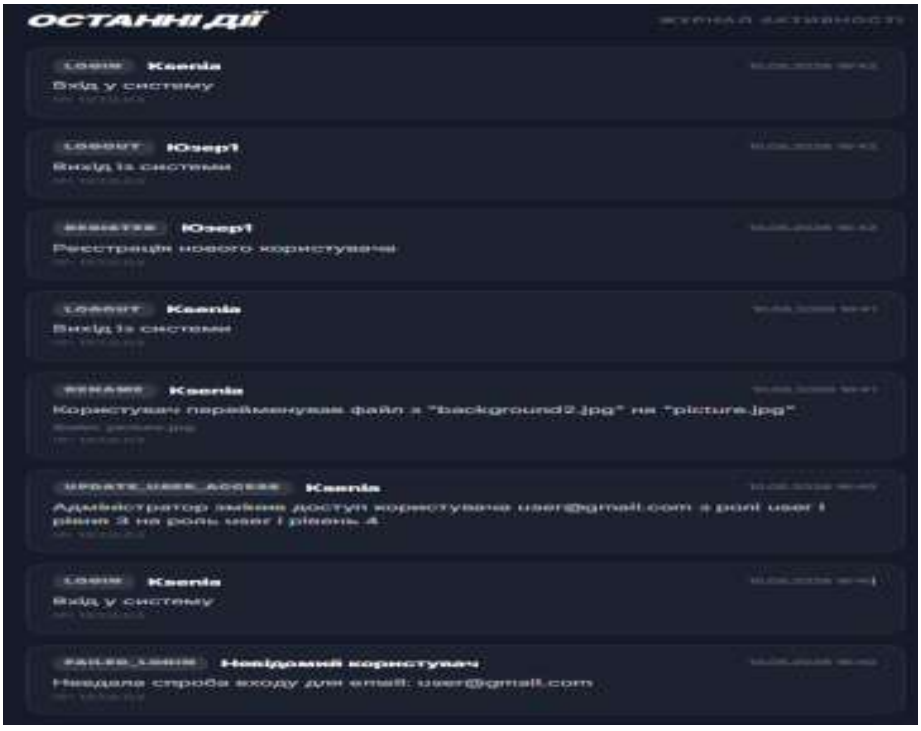


Рисунок 3.19 – Інтерфейс журналу подій системи

Доступ до журналу подій має лише адміністратор системи. Це забезпечує захист службової інформації про активність користувачів та унеможливлює перегляд журналу звичайними користувачами. У межах адміністративної панелі адміністратор може переглядати останні дії користувачів, аналізувати операції з файлами та контролювати зміни рівнів доступу.

Крім журналювання дій користувачів, у системі використовується стандартний механізм логування Laravel, який забезпечує фіксацію системних помилок, винятків та службових повідомлень. Такі журнали зберігаються у директорії storage/logs і можуть використовуватися для діагностики помилок, аналізу стабільності роботи вебзастосунку та виявлення проблем у процесі експлуатації.

Структуру директорії системних журналів наведено на рисунку 3.20.

Рисунок 3.20 – Структура директорії системних журналів

Для забезпечення спостережності також використовується статистична інформація адміністративної панелі. У ній відображається кількість користувачів, кількість файлів, загальний обсяг файлового сховища та кількість записів журналу подій. Це дозволяє адміністратору швидко оцінити поточний стан системи та активність користувачів.

Підсистема аудиту та логування також доповнює механізми контролю доступу. У випадку зміни ролі або рівня доступу користувача відповідна дія фіксується в журналі подій із зазначенням попередніх та нових параметрів доступу. Це дозволяє контролювати адміністративні зміни та забезпечує прозорість управління правами користувачів.

Таким чином, у межах даного підрозділу реалізовано підсистему аудиту, логування та спостережності, яка забезпечує фіксацію основних дій користувачів, контроль адміністративних операцій, аналіз активності у системі та підтримує виконання технічної вимоги ТВ7 щодо журналювання подій користувачів.

3.4 Проведення тестування та аналіз ефективності розроблених рішень

Після завершення реалізації основних функціональних модулів вебсистеми було проведено тестування її працездатності, механізмів захисту інформації та перевірку відповідності реалізованих рішень технічним вимогам, сформованим у другому розділі. Основною метою тестування є підтвердження коректності роботи підсистем автентифікації, контролю доступу, захищеного зберігання файлів, шифрування, журналювання подій та адміністративного управління.

У процесі тестування перевірялися функціональні можливості вебсистеми, а також ефективність реалізованих механізмів інформаційної безпеки. Особливу увагу приділено перевірці виконання технічних вимог ТВ1–ТВ10.

Для перевірки працездатності системи використовувалося ручне функціональне тестування, а також Unit- і Feature-тестування із використанням стандартних засобів Laravel. Unit-тестування застосовувалося для перевірки окремих програмних компонентів, а Feature-тестування – для перевірки повних сценаріїв взаємодії користувача із вебсистемою.

Основні напрями тестування наведено в таблиці 3.1.

Початок таблиці 3.1 – Основні напрями тестування вебсистеми

№ з/п	Напрям тестування	Що перевірялося	Очікуваний результат
1	2	3	4
1	Автентифікація	Реєстрація, вхід та вихід користувача	Доступ надається лише після успішної автентифікації
2	Контроль доступу	Перевірка рівня доступу користувачів	Користувач отримує доступ лише до дозволених файлів

Кінець таблиці 3.1

№ з/п	Напрям тестування	Що перевірялося	Очікуваний результат
1	2	3	4
3	Завантаження файлів	Формат, MIME-тип та розмір файлів	Заборонені файли не завантажуються
4	Шифрування файлів	Збереження файлів у захищеному вигляді	У сховищі зберігається зашифрована версія файлу
5	Журналювання	Фіксація дій користувачів	Події записуються до журналу аудиту
6	Адміністративні функції	Управління користувачами та правами доступу	Зміни доступні лише адміністратору

Для перевірки механізмів автентифікації були протестовані сценарії реєстрації нового користувача, успішного входу до системи, невдалої спроби входу та завершення сеансу роботи. У результаті тестування встановлено, що система коректно виконує перевірку облікових даних, не допускає неавторизованих користувачів та автоматично фіксує події автентифікації у журналі аудиту.

Окремо було перевірено реалізацію мандатної моделі контролю доступу. Для цього створено користувачів із різними рівнями доступу та файли з різними рівнями секретності. У результаті тестування підтверджено, що система дозволяє користувачу працювати лише з файлами, рівень секретності яких не перевищує його рівень доступу. У випадку недостатнього рівня доступу система блокує виконання операції та забороняє доступ до файлового ресурсу.

Під час тестування підсистеми завантаження файлів перевірялися механізми валідації форматів, MIME-типів та розмірів файлів. Результати перевірки підтвердили, що система блокує завантаження файлів, які не відповідають встановленим вимогам безпеки, та дозволяє працювати лише з дозволеними форматами даних.

Для перевірки механізму шифрування було виконано завантаження файлів різних рівнів секретності та проаналізовано їх збереження у файловому сховищі.

У результаті встановлено, що файли зберігаються у зашифрованому вигляді, а доступ до них здійснюється лише після проходження перевірки прав доступу користувача. Це підтверджує коректну роботу реалізованого механізму криптографічного захисту інформації.

Також було перевірено роботу підсистеми журналювання подій. Результати тестування підтвердили, що система автоматично фіксує основні дії користувачів, зокрема вхід до системи, невдалі спроби автентифікації, завантаження, перегляд, перейменування та видалення файлів. Для кожної події зберігається інформація про користувача, тип дії, IP-адресу, User-Agent та час виконання операції.

Для перевірки працездатності програмних компонентів було використано стандартний механізм тестування Laravel. У рамках Unit- та Feature- тестування перевірено роботу автентифікації, захищених маршрутів, обмеження доступу, файлових операцій та механізмів журналювання. Результат виконання тестів наведено на рисунку 3.21.

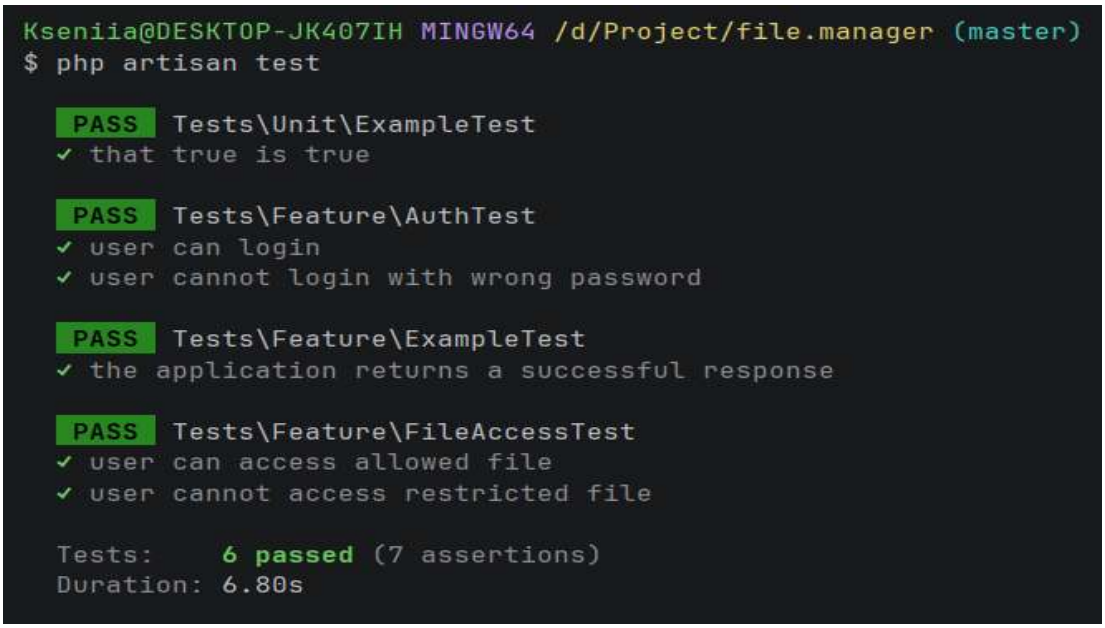


Рисунок 3.21 – Результати виконання тестів Laravel

Результати тестування підтвердили коректну роботу основних функціональних і безпекових механізмів системи. Узагальнену перевірку відповідності реалізованих рішень технічним вимогам наведено в таблиці 3.2.

Таблиця 3.2 – Перевірка виконання технічних вимог

№ з/п	Код вимоги	Назва вимоги	Реалізований механізм	Статус
1	ТВ1	Автентифікація користувачів	Впроваджено систему входу	Виконано
2	ТВ2	Мандатний контроль доступу	Реалізовано мандатний контроль доступу	Виконано
3	ТВ3	Хешування паролів	Використано алгоритм bcrypt	Виконано
4	ТВ4	Шифрування каналів зв'язку	Передача даних через HTTPS/TLS	Виконано
5	ТВ5	Завантаження файлів	Реалізовано захищене завантаження файлів	Виконано
6	ТВ6	Валідація файлів	Перевірка MIME-типів та обмеження розміру	Виконано
7	ТВ7	Аудит безпеки	Журналювання (логування) подій користувачів	Виконано
8	ТВ8	Захист збережених даних	Шифрування файлів алгоритмом AES-256	Виконано
9	ТВ9	Управління правами	Розмежування прав доступу	Виконано
10	ТВ10	Відновлення даних	Автоматичне резервне копіювання	Виконано

Аналіз ефективності розроблених рішень показав, що реалізована мандатна модель доступу забезпечує ефективне обмеження доступу до корпоративної інформації відповідно до рівня секретності файлів. Використання механізмів шифрування дозволяє забезпечити захист файлових ресурсів навіть у випадку компрометації файлового сховища, а система журналювання створює можливість проведення аудиту дій користувачів та аналізу подій інформаційної безпеки.

Таким чином, проведене тестування підтвердило працездатність розробленої вебсистеми та відповідність реалізованих програмних рішень технічним вимогам щодо забезпечення захисту корпоративної інформації.

3.5 Розробка документації та рекомендації щодо впровадження системи

Після завершення реалізації та тестування вебсистеми було сформовано рекомендації щодо впровадження та подальшої експлуатації розробленого програмного рішення у корпоративному середовищі. Основною метою даного етапу є спрощення процесу розгортання системи, її адміністрування та підтримки під час використання.

У процесі розробки визначено основні етапи встановлення та налаштування вебсистеми, порядок підключення до бази даних, конфігурацію серверного середовища та механізми запуску застосунку. Для функціонування системи використовуються вебсервер Apache або Nginx, PHP, MySQL та менеджер залежностей Composer.

Рекомендації передбачають налаштування серверного середовища, виконання міграцій бази даних, конфігурацію файлу .env, генерацію ключа застосунку Laravel та запуск вебсервера. Окремо визначено порядок організації файлового сховища, резервного копіювання бази даних та механізми відновлення інформації у випадку програмних або апаратних збоїв.

Для забезпечення безпечної експлуатації системи сформовано рекомендації щодо використання захищеного мережевого з’єднання HTTPS/TLS, регулярного оновлення серверного програмного забезпечення, обмеження доступу до серверної інфраструктури та контролю журналів подій. Також рекомендовано використовувати складні паролі для адміністративних облікових записів і періодично здійснювати резервне копіювання файлового сховища та бази даних.

Окрему увагу приділено рекомендаціям щодо адміністрування системи. Адміністратор повинен контролювати рівні доступу користувачів, аналізувати журнал подій та здійснювати моніторинг операцій роботи із файловими ресурсами. Це дозволяє своєчасно виявляти підозрілу активність та підтримувати належний рівень інформаційної безпеки.

Під час розробки також було описано структуру програмного проєкту Laravel, основних контролерів, middleware-компонентів, моделей бази даних та

механізмів взаємодії між функціональними модулями системи. Це спрощує подальшу підтримку та можливе масштабування вебзастосунку.

Крім цього, визначено можливості подальшого розвитку системи, зокрема інтеграцію із корпоративними сервісами автентифікації, впровадження багатофакторної автентифікації, використання зовнішніх файлових сховищ, централізованого резервного копіювання та інтеграції із системами моніторингу й SIEM-рішеннями.

Таким чином, сформовані рекомендації щодо впровадження та експлуатації вебсистеми забезпечують можливість її безпечного розгортання, адміністрування та подальшого використання у корпоративному середовищі.

3.6 Висновок

У третьому розділі виконано програмну реалізацію вебсистеми захищеного управління файлами для корпоративного середовища відповідно до сформованих технічних вимог і розробленої архітектури системи. Реалізовано підсистеми автентифікації, авторизації та мандатного контролю доступу, що забезпечують розмежування прав користувачів відповідно до рівня допуску та рівня секретності файлів. Також впроваджено механізми захищеного завантаження, шифрування та зберігання файлів із використанням криптографічного захисту.

Крім цього, у системі реалізовано підсистему аудиту, журналювання та спостережності для контролю дій користувачів, адміністративних операцій і подій інформаційної безпеки. Під час розробки проведено ручне функціональне тестування, а також Unit- і Feature-тестування Laravel, результати яких підтвердили працездатність основних компонентів системи та коректність реалізації механізмів захисту інформації.

Отримані результати підтверджують досягнення поставленої мети дипломної роботи та відповідність вебсистеми поставленим вимогам щодо захисту корпоративної інформації.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено вебсистему захищеного управління файлами для корпоративного середовища, основним призначенням якої є забезпечення безпечного зберігання, обробки та контролю доступу до корпоративної інформації. Актуальність роботи обумовлена необхідністю підвищення рівня захисту інформаційних ресурсів організацій в умовах зростання кількості кіберзагроз, несанкціонованого доступу до даних та ризиків витоку конфіденційної інформації.

У процесі виконання роботи проведено аналіз сучасних підходів до захисту корпоративних файлових систем, механізмів контролю доступу, журналювання подій та криптографічного захисту інформації. У результаті аналізу визначено основні загрози для корпоративних вебсистем, серед яких несанкціонований доступ до файлів, компрометація облікових записів, порушення цілісності даних та завантаження небезпечного контенту. На основі проведеного аналізу сформовано технічні вимоги до системи та обґрунтовано вибір архітектурних і програмних рішень.

Для реалізації вебсистеми використано фреймворк Laravel та архітектурний підхід MVC, що дозволило забезпечити розподіл бізнес-логіки, інтерфейсу користувача та механізмів роботи з даними. Запропонована архітектура забезпечує взаємодію між клієнтською частиною, серверною логікою, базою даних та файловими сховищами, а також підтримує можливість подальшого масштабування та інтеграції додаткових засобів інформаційної безпеки.

У межах дипломної роботи реалізовано механізми автентифікації та авторизації користувачів, а також мандатну модель контролю доступу, відповідно до якої кожному користувачу призначається рівень доступу, а кожному файлу – рівень секретності. Доступ до файлових ресурсів надається лише у випадку відповідності рівня доступу користувача рівню секретності інформації. Такий підхід дозволяє забезпечити централізований контроль доступу до корпоративних даних та підвищити рівень інформаційної безпеки системи. Для захисту облікових

даних використано алгоритм хешування bcrypt, а для забезпечення конфіденційності файлів – симетричне шифрування AES-256.

У роботі також реалізовано підсистему захищеного завантаження та зберігання файлів із перевіркою MIME-типу, формату та розміру файлів, що дозволяє запобігти завантаженню небезпечного або небажаного контенту. Для підвищення рівня спостережності впроваджено підсистему аудиту та журналювання подій, яка автоматично фіксує дії користувачів, спроби автентифікації та операції роботи з файлами. Додатково реалізовано використання HTTPS/TLS для захищеної передачі даних між клієнтом і сервером та механізми резервного копіювання для зниження ризику втрати інформації.

Для перевірки працездатності вебсистеми проведено функціональне тестування, а також Unit- і Feature-тестування із використанням стандартних засобів Laravel. Результати тестування підтвердили коректну роботу механізмів автентифікації, контролю доступу, шифрування, журналювання та управління файловими ресурсами.

У результаті виконання дипломної роботи досягнуто поставленої мети та виконано всі визначені завдання. Розроблена система забезпечує безпечне управління корпоративними файлами, контроль доступу до інформації, захист файлових ресурсів та підтримку механізмів аудиту подій безпеки. Практичне значення роботи полягає у можливості застосування створеної системи для підвищення рівня інформаційної безпеки підприємств та автоматизації процесів контролю доступу до корпоративних даних.

Перспективами подальшого розвитку системи є впровадження багатофакторної автентифікації, інтеграція із зовнішніми корпоративними сервісами, використання зовнішніх файлових сховищ та розширення механізмів моніторингу інформаційної безпеки. Запропоноване програмне рішення може слугувати основою для подальшого розвитку корпоративних систем захищеного файлового управління та вдосконалення механізмів захисту інформації у вебсередовищі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Корченко О. Г. Основи інформаційної безпеки : навч. посіб. Київ : Видавнича група BHV, 2019. 320 с.

2. ISO/IEC 27001:2022 Information security, cybersecurity and privacy protection — Information security management systems — Requirements. Geneva : ISO, 2022. 34 p.

3. ENISA Threat Landscape 2025 URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025 (дата звернення: 13.03.2026).

4. Stallings W. Cryptography and Network Security: Principles and Practice. 8th ed. Harlow : Pearson, 2020. 832 p.

5. OWASP Top 10 Web Application Security Risks. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 13.03.2026).

6. Татарчук М. І. Корпоративні інформаційні системи : підручник. Київ : КНЕУ, 2005. 291 с.

7. ДСТУ ISO/IEC 27001:2023 Інформаційні технології. Методи захисту. Системи управління інформаційною безпекою. Вимоги (ISO/IEC 27001:2022, IDT). Київ : ДП «УкрНДНЦ», 2023.

8. Про затвердження Мінімальних вимог до захисту інформаційних, електронних комунікаційних, інформаційно-комунікаційних та технологічних систем. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/373-2006-п#Text (дата звернення: 13.03.2026).

9. НД ТЗІ 2.5-004-99 Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу. Київ : Департамент спеціальних телекомунікаційних систем та захисту інформації СБУ, 1999.

10. Що таке хмарні технології: визначення, важливість та застосування URL: https://wezom.com.ua/ua/blog/scho-take-hmarni-tehnologiyi-viznachennya-vazhlivist-ta-zastosuvannya (дата звернення: 13.03.2026).

11. Про основні засади забезпечення кібербезпеки України. Офіційний

вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/2163-19#Text (дата звернення: 13.03.2026).

12. Про захист інформації в інформаційно-комунікаційних системах. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/80/94-вр#Text (дата звернення: 13.03.2026).

13. Людський фактор у кібербезпеці: навчання та підвищення обізнаності співробітників URL: https://surl.lt/mhxpde (дата звернення: 13.03.2026).

14. Cybersecurity Best Practices. URL: https://www.cisa.gov/topics/cybersecurity-best-practices (дата звернення: 13.03.2026).

15. Про затвердження Загальних вимог з кіберзахисту об'єктів критичної інфраструктури. Офіційний вебпортал парламенту України. URL: https://zakon.rada.gov.ua/laws/show/518-2019-п#Text (дата звернення: 17.05.2026).

16. Top Threats to Cloud Computing: Egregious Eleven. Cloud Security Alliance, 2020. 43 p. URL: https://surl.lt/osoqif (дата звернення: 14.03.2026).

17. ДСТУ ISO/IEC 15408-1:2017 Інформаційні технології. Методи захисту. Критерії оцінювання безпеки інформаційних технологій. Частина 1. Модель та загальні вимоги (ISO/IEC 15408-1:2009, IDT). Київ : ДП «УкрНДНЦ», 2018.

18. Authentication Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html (дата звернення: 14.03.2026).

19. Мандатне керування доступом. URL: https://surl.li/fturjk (дата звернення: 14.03.2026).

20. Що таке протокол HTTPS і як він працює. URL: https://alexhost.com/uk/faq/what-is-the-https-protocol-and-how-does-it-work/ (дата звернення: 14.03.2026).

21. Що таке протокол TLS, як він працює та від чого захищає. URL: https://cityhost.ua/uk/blog/scho-take-protokol-tls-yak-vin-pracyu-ta-vid-chogo-zahischa.html (дата звернення: 14.03.2026).

22. ДСТУ ISO/IEC 27002:2022 Інформаційні технології. Заходи контролю

інформаційної безпеки (ISO/IEC 27002:2022, IDT). Київ : ДП «УкрНДНЦ», 2023. 164 с.

23. SQL ін’єкції та захист від них. URL: https://foxminded.ua/sql-iniektsii/ (дата звернення: 14.03.2026).

24. XSS атаки, що це таке і чому вони дуже небезпечні. URL: https://hackyourmom.com/pryvatnist/xss-ataky-shho-cze-take-i-chomu-vony-duzhe-nebezpechni/ (дата звернення: 14.03.2026)

25. CSRF атака: небезпека підроблених запитів. URL: https://surl.lu/nkcoya (дата звернення: 14.03.2026)

26. Logging Cheat Sheet. URL: https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html (дата звернення: 14.03.2026)

27. Laravel. URL: https://laravel.com/docs/13.x (дата звернення: 14.03.2026)

28. PHP. URL: https://www.php.net/ (дата звернення: 14.03.2026)

29. Побудова динамічних вебзастосунків за допомогою MVC. URL: https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc (дата звернення: 14.03.2026)

30. Реляційні бази даних усе, що необхідно про них знати. URL: https://foxminded.ua/reliatsiini-bazy-danykh/ (дата звернення: 14.03.2026)

31. GitHub. URL: https://github.com/ (дата звернення: 14.03.2026)

32. The NIST Definition of Cloud Computing: NIST Special Publication 800-145. URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (дата звернення: 17.03.2026)

33. Google Drive. URL: https://workspace.google.com/products/drive/ (дата звернення: 17.03.2026)

34. Dropbox. URL: https://www.dropbox.com/features/cloud-storage (дата звернення: 17.03.2026)

35. OneDrive. URL: https://www.microsoft.com/uk-ua/microsoft-365/onedrive/online-cloud-storage (дата звернення: 17.03.2026)

36. Secure cloud storage solutions. URL: https://www.box.com/cloud-storage

(дата звернення: 17.03.2026)

37. MEGA Security Architecture. URL: https://mega.io/security (дата звернення: 17.05.2026).

38. Sync Privacy Policy. URL: https://www.sync.com/privacy/ (дата звернення: 17.05.2026).

39. Різні типи хмарних технологій, їх переваги та недоліки. URL: https://kub.ua/blog/oblachnyh-tehnologij-preimushhestva-i-nedostatki/ (дата звернення: 17.03.2026)

40. Nextcloud. URL: https://nextcloud.com/ (дата звернення: 17.03.2026)

41. Seafile. URL: https://www.seafile.com/en/home/ (дата звернення: 17.03.2026)

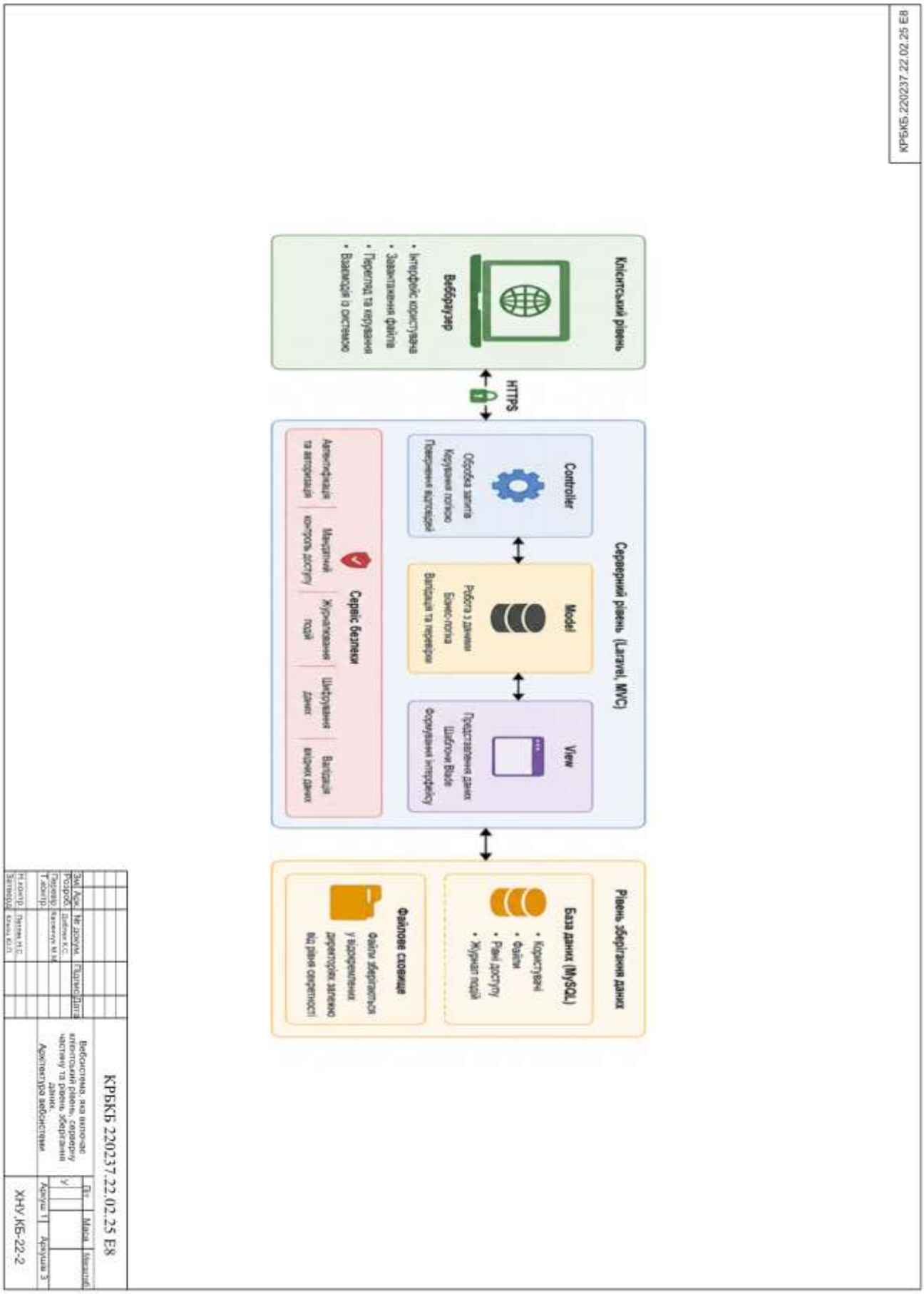
42. ownCloud Documentation Overview. URL: https://doc.owncloud.com/ (дата звернення: 17.03.2026)

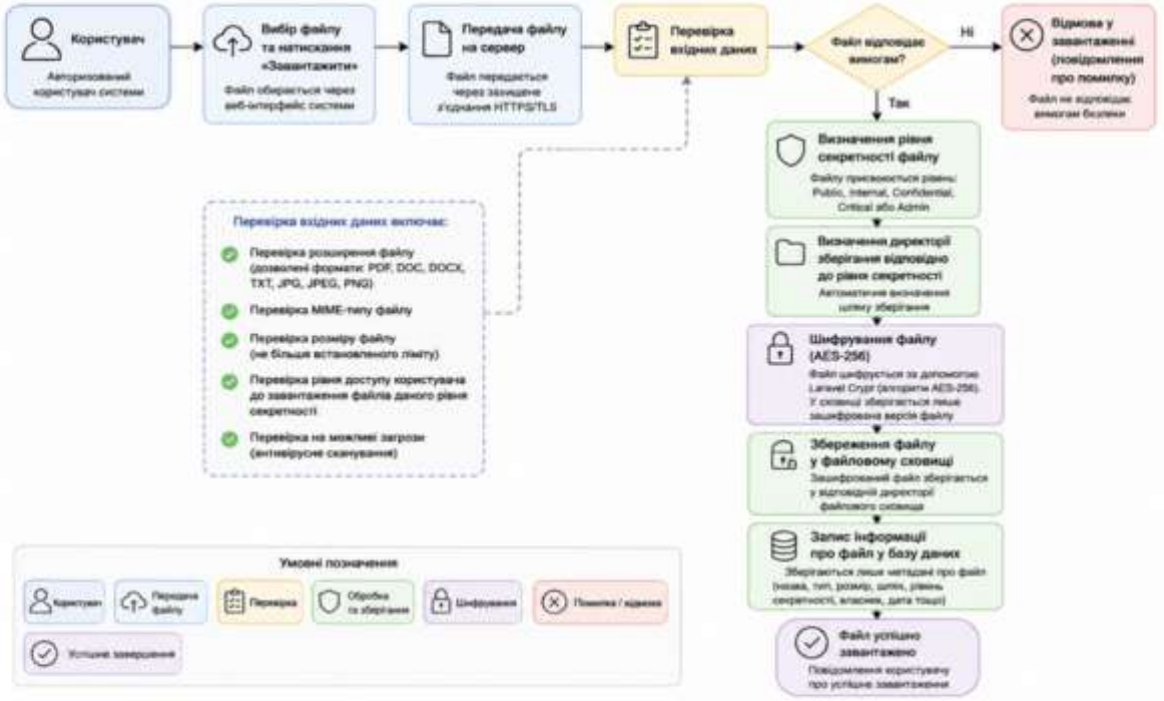
43. Oracle Cloud Infrastructure Documentation. URL: https://docs.oracle.com/en-us/iaas/Content/File/home.htm (дата звернення: 17.03.2026)

44. Rclone. URL: https://rclone.org/ (дата звернення: 17.03.2026)

45. Tahoe-LAFS. URL: https://www.tahoe-lafs.org/ (дата звернення: 17.03.2026).

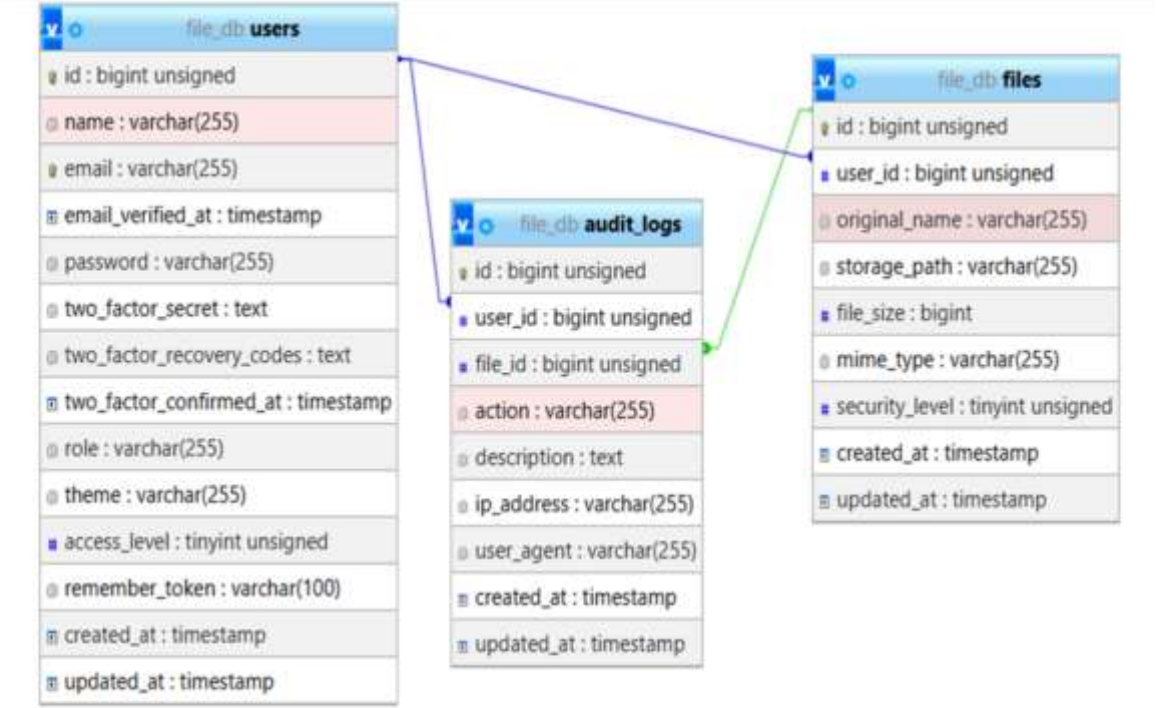
ДОДАТОК А


КРБКБ.220237.22.02.25 Е8
Клієнтський рівень
Веббраузер
• Інтерфейс користувача
• Завантаження файлів
• Перегляд та керування
• Взаємодія із системою
HTTPS
Серверний рівень (Laravel, MVC)
Controller
Обробка запита
Керування логікою
Повернення відповіді
Model
Робота з даними
Бізнес-логіка
Валідація та перевірки
View
Представлення даних
Шаблони Blade
Формування інтерфейсу
Сервіс безпеки
Автентифікація та авторизація
Мандатний контроль доступу
Журналювання подій
Шифрування даних
Валідація вхідних даних
Рівень зберігання даних
База даних (MySQL)
• Користувачі
• Файли
• Рівні доступу
• Журнал подій
Файлове сховище
Файли зберігаються у відокремлених директоріях залежно від рівня секретності
КРБКБ 220237.22.02.25 Е8
Вебсистема, яка включає клієнтський рівень, серверну частину та рівень зберігання даних.
Архітектура вебсистеми
Зм. Арк. № докум. Підпис Дата
Розроб. Дубінін К.С.
Перевір. Калінчук М.М.
Т.контр.
Н.контр. Литвин Н.С.
Затверд. Клюєв Ю.О.
Літ. Маса Масштаб
У
Аркуш 11 Аркушів 3
ХНУ.КБ-22-2



					КРБКБ 220237.22.02.25 Е8		
Зм.	Арк.	№ докум.	Підпис	Дата	Процес збереження та завантаження файлу Схема процесу завантаження та збереження файлу	Літ.	Маса	Масштаб
Розроб.		Дибляк К.С.				У		
Перевір.		Касянчук М.М.						
Т.контр.						Аркуш 2	Аркушів 3	
Н. контр.		Петлак Н.С.				ХНУ,КБ-22-2		
Затверд.		Клець Ю.П.						

КРБКБ.220237.22.02.25 Е8



						КРБКБ 220237.22.02.25 Е8	
Зм.	Арк.	№ докум.	Підпис	Дата		Літ.	Маса	Масштаб
Розроб.		Дяблов К.С.			Схема взаємозв'язків між таблицями бази даних	У		
Перевір.		Косенчук М.М.				Аркуш 3	Аркушів 3	
Т.контр.								
Н.контр.		Пилав Н.С.				ХНУ,КБ-22-2		
Затверд.		Юнусь Ю.П.						