

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

_____ **Ігор БОЛБОТ**
(підпис)

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри комп'ютерних наук

_____ **Михайло ШВИДЕНКО**
(підпис)

«___» _____ 2026 р.

«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

**на тему: «Розробка веб-платформи “SmartLibrary” для управління
електронними навчальними ресурсами»**

Спеціальність «126 Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні системи та технології»

Гарант освітньої програми

К.Т.Н., доцент

_____ **Максим МОКРІЄВ**
(підпис)

**Керівник бакалаврської
кваліфікаційної роботи**

К.Т.Н., С.Н.С.

_____ **Вікторія СМОЛІЙ**
(підпис)

Виконав

_____ **Матвій ТИМОЩУК**
(підпис)

Київ-2026

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук

к.т.н., доцент _____ Михайло

ШВИДЕНКО
(підпис)

«__» _____ 2026 р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ**

Тимощуку Матвію Володимировичу

(прізвище, ім'я, по батькові)

Спеціальність «126 Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні системи та технології»

Тема бакалаврської кваліфікаційної роботи

«Розробка веб-платформи “SmartLibrary” для управління електронними навчальними ресурсами»

затверджена наказом від «19» грудня 2025 р. №3205 «С»

Термін подання завершеної роботи на кафедру «22 травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):

Перелік питань, що підлягають дослідженню:

Перелік графічних документів (за необхідності).

Дата видачі завдання «19» грудня 2025 р.

**Керівник бакалаврської
кваліфікаційної роботи**

(підпис)

Вікторія СМОЛІЙ

**Завдання взяв до
виконання**

(підпис)

Матвій ТИМОЩУК

ЗМІСТ

ВСТУП.....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВЕБ-ПЛАТФОРМИ SMARTLIBRARY ДЛЯ УПРАВЛІННЯ ЕЛЕКТРОННИМИ НАВЧАЛЬНИМИ РЕСУРСАМИ.....	8
1.1 Опис предметної області управління електронними навчальними ресурсами	8
1.2 Аналіз існуючих рішень й інформаційних систем.....	10
1.3 Моделювання предметної області.....	14
1.4 Аналіз вимог інформаційної системи.....	18
1.5 Постановка завдання.....	20
1.6 Висновки до першого розділу.....	21
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	22
2.1 Логічна модель даних веб-платформи SmartLibrary.....	22
2.2 Діаграма класів і кооперації веб-платформи SmartLibrary.....	25
2.3 Діаграма компонентів.....	29
2.4 Діаграма пакетів веб-платформи SmartLibrary.....	31
2.5 Висновки до розділу 2.....	33
3.1 Обґрунтування вибору технологічних засобів реалізації.....	35
3.2 Розробка фізичної моделі бази даних.....	36
3.3 Алгоритмізація роботи програмних модулів системи.....	39
3.4 Реалізація основних програмних модулів.....	44
3.5 Опис інтерфейсу користувача.....	46
3.6 Висновки до третього розділу.....	47
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ВЕБ- ПЛАТФОРМИ SMARTLIBRARY.....	49
4.1 План тестування програмних модулів та методика оцінювання результатів.....	49

4.2 Тестування веб-платформи SmartLibrary	51
4.3 Оцінювання ефективності роботи системи.....	60
4.4 Висновки до четвертого розділу	62
ВИСНОВКИ.....	64
ДОДАТОК А.....	66
ДОДАТОК Б.....	68
ДОДАТОК В.....	71
ДОДАТОК Г	75

ВСТУП

Сучасний розвиток освітнього процесу безпосередньо пов'язаний із використанням електронних навчальних ресурсів, цифрових бібліотек, інтерактивних курсів, дистанційних платформ і засобів автоматизованого доступу до навчальних матеріалів. У закладах освіти постійно зростає кількість електронних підручників, методичних матеріалів, лабораторних практикумів, тестових завдань, відеолекцій, презентацій та інших цифрових ресурсів, які потребують систематизації, зручного пошуку, контролю доступу й аналізу використання.

Актуальність теми зумовлена необхідністю створення зручного програмного середовища для централізованого управління електронними навчальними ресурсами. У багатьох навчальних закладах матеріали зберігаються у різних файлових сховищах, месенджерах, хмарних папках або окремих навчальних курсах, що ускладнює їх пошук, оновлення, перевірку актуальності та контроль доступу. Відсутність єдиної платформи призводить до дублювання матеріалів, втрати даних, неузгодженості прав користувачів і неможливості швидко отримати статистику використання ресурсів.

Веб-платформа SmartLibrary орієнтована на вирішення цих проблем шляхом створення єдиного каталогу електронних навчальних матеріалів із підтримкою ролей користувачів, заявок на доступ, закладок, переглядів, оцінювання ресурсів, журналу дій та аналітичних звітів. Така система може використовуватися студентами, викладачами, бібліотекарями, модераторами та адміністраторами для організації навчального контенту й забезпечення контрольованого доступу до нього.

Метою кваліфікаційної роботи є розроблення веб-платформи SmartLibrary для управління електронними навчальними ресурсами, яка забезпечує збереження, пошук, перегляд, фільтрацію, модерацію, контроль доступу, ведення персональної бібліотеки та формування аналітичної звітності щодо використання навчальних матеріалів.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. проаналізувати предметну область управління електронними навчальними ресурсами;
2. розглянути існуючі програмні рішення для організації електронного навчання та цифрових бібліотек;
3. визначити функціональні, нефункціональні, технічні та безпекові вимоги до системи;
4. розробити UML-моделі предметної області та основних процесів роботи платформи;
5. спроектувати логічну та фізичну модель бази даних;
6. розробити архітектуру програмного забезпечення та визначити склад основних модулів;
7. реалізувати програмний прототип платформи SmartLibrary;
8. провести тестування основних функцій системи та оцінити її ефективність.

Об'єктом дослідження є процес організації, збереження, пошуку та використання електронних навчальних ресурсів у навчальному середовищі.

Предметом дослідження є методи, моделі та програмні засоби розроблення інформаційної системи для управління електронними навчальними ресурсами, контролю доступу до них і формування аналітичної звітності.

Методи дослідження включають системний аналіз предметної області, порівняльний аналіз існуючих рішень, UML-моделювання, логічне та фізичне проектування бази даних, об'єктно-орієнтований підхід до розроблення програмного забезпечення, алгоритмізацію основних процесів, функціональне тестування та оцінювання результатів роботи системи.

Практичне значення роботи полягає у створенні програмного прототипу, який може бути використаний як основа для впровадження інформаційної системи управління електронними навчальними ресурсами в закладі освіти. Розроблена платформа забезпечує централізований каталог матеріалів, швидкий пошук за назвою, дисципліною, автором і тегами, підтримку різних рівнів

доступу, оброблення заявок, ведення персональної бібліотеки користувача, журналювання подій і формування звітів.

Наукова новизна роботи полягає в узгодженому поєднанні функцій електронної бібліотеки, системи керування доступом, персонального освітнього кабінету та аналітичного модуля в межах єдиної програмної платформи. Запропонована структура даних дозволяє пов'язати користувачів, навчальні ресурси, заявки на доступ, закладки, оцінки, події використання та журнал аудиту, що забезпечує цілісне управління електронними матеріалами.

Програмне забезпечення реалізовано з використанням мови Python, бібліотеки PyQt6 для створення графічного інтерфейсу та SQLite для локального збереження даних. Такий підхід дає змогу створити автономний програмний прототип SmartLibrary Pro, який може працювати на персональному комп'ютері користувача без складного серверного розгортання. Надалі система може бути розширена до повноцінної веб-платформи з клієнт-серверною архітектурою, багатокористувацьким доступом і хмарним сховищем навчальних матеріалів.

Структура кваліфікаційної роботи складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел і додатків. У першому розділі виконано аналіз предметної області та існуючих рішень. У другому розділі розроблено інформаційне та програмне забезпечення системи. У третьому розділі описано реалізацію програмних модулів, фізичну модель бази даних та алгоритми роботи. У четвертому розділі проведено тестування системи й оцінювання ефективності її використання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ВЕБ-ПЛАТФОРМИ SMARTLIBRARY ДЛЯ УПРАВЛІННЯ ЕЛЕКТРОННИМИ НАВЧАЛЬНИМИ РЕСУРСАМИ

1.1 Опис предметної області управління електронними навчальними ресурсами

Предметна область веб-платформи SmartLibrary охоплює процеси централізованого зберігання, систематизації, пошуку, перегляду та контрольованого надання доступу до електронних навчальних ресурсів. У сучасному освітньому середовищі навчальні матеріали створюються у різних форматах: електронні підручники, методичні вказівки, презентації, відеолекції, тестові матеріали, посилання на зовнішні джерела, інтерактивні завдання та архіви навчальних курсів. За відсутності єдиного каталогу такі матеріали розміщуються у різних хмарних сховищах, месенджерах, поштових повідомленнях або локальних папках викладачів, що ускладнює їх пошук, актуалізацію та повторне використання.

Веб-платформа SmartLibrary розглядається як інформаційна система, що забезпечує взаємодію між студентами, викладачами, бібліотекарями або модераторами та адміністратором. Студент використовує систему для пошуку та перегляду доступних навчальних ресурсів, формування власної електронної бібліотеки, подання заявки на доступ до обмежених матеріалів і перегляду історії роботи з ресурсами. Викладач або автор може додавати навчальні матеріали, описувати їх метадані, оновлювати версії та аналізувати використання ресурсів. Бібліотекар або модератор відповідає за перевірку якості опису, класифікацію, редагування каталогу та опрацювання заявок. Адміністратор керує користувачами, ролями, правами доступу, довідниками та системними звітами.

На рисунку 1.1 подано DFD-діаграму першого рівня, яка відображає основні інформаційні потоки веб-платформи SmartLibrary. У моделі виділено чотири ключові процеси: авторизація та ролі, каталог ресурсів, заявки на доступ,

аналітика та звіти. Кожен процес взаємодіє з відповідним сховищем даних: користувачі, електронні ресурси, заявки доступу, події і звіти.

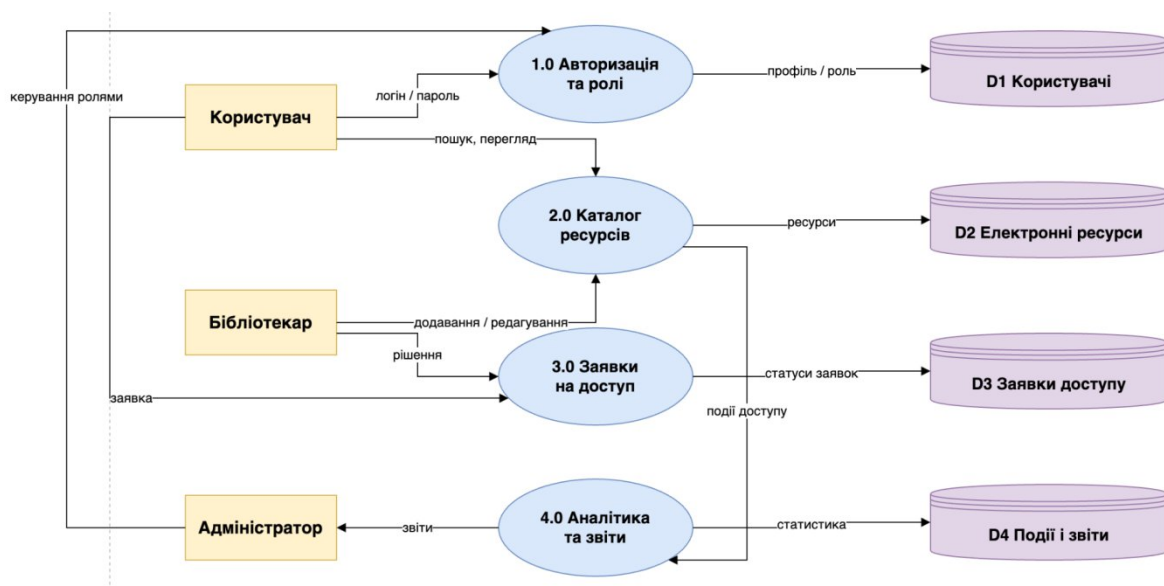


Рис. 1.1. DFD-діаграма першого рівня веб-платформи SmartLibrary

Процес авторизації та ролей приймає від користувача логін і пароль, перевіряє облікові дані та повертає профіль із визначеною роллю. Процес каталогу ресурсів забезпечує пошук, фільтрацію, перегляд та отримання інформації про навчальні матеріали. Процес заявок на доступ використовується тоді, коли ресурс має обмежений режим перегляду або потребує погодження з бібліотекарем. Процес аналітики та звітів збирає події перегляду, відкриття ресурсів, створення заявок і формує статистичну інформацію для адміністратора та модератора.

Таблиця 1.1

Основні інформаційні об'єкти предметної області SmartLibrary

Об'єкт	Зміст	Використання в системі
Користувач	Студент, викладач, бібліотекар або адміністратор, що працює з платформою	Авторизація, ролі, історія дій
Електронний ресурс	Навчальний матеріал у вигляді файлу, посилання,	Каталог, пошук, перегляд, доступ

	відео, презентації або методичних вказівок	
--	--	--

Продовження таблиці 1.1

Категорія ресурсу	Тематичний або навчальний розділ, до якого належить матеріал	Фільтрація та структуризація каталогу
Заявка на доступ	Запит користувача на відкриття обмеженого ресурсу	Погодження, відхилення, журнал статусів
Подія доступу	Факт перегляду, відкриття, завантаження або відмови в доступі	Аналітика, аудит, статистика
Звіт	Узагальнена інформація про активність користувачів та популярність ресурсів	Адміністративний контроль і планування

Сукупність зазначених об'єктів формує основу предметної області SmartLibrary. Платформа має забезпечити не тільки збереження файлів, а й повноцінне керування життєвим циклом навчального ресурсу: створення запису, перевірку, публікацію, визначення прав доступу, перегляд, оновлення, архівування та аналіз використання. Саме така структура дозволяє зменшити дублювання матеріалів, підвищити якість навчального контенту та спростити доступ студентів до актуальних ресурсів.

1.2 Аналіз існуючих рішень й інформаційних систем

Для визначення функціональних вимог до веб-платформи SmartLibrary було проаналізовано сучасні рішення, які використовуються у сфері електронного навчання, керування курсами та зберігання цифрових освітніх матеріалів. До таких рішень належать Moodle, Google Classroom, Canvas LMS, Blackboard Learn та DSpace. Зазначені системи мають розвинені засоби роботи з навчальними матеріалами, однак відрізняються за цільовим призначенням,

моделлю доступу, можливостями каталогу, ролями користувачів і рівнем придатності саме для керування електронною бібліотекою навчальних ресурсів.

Moodle є однією з найбільш поширених систем керування навчанням. Вона підтримує створення курсів, розміщення ресурсів, додавання навчальних активностей, організацію груп, оцінювання та відстеження прогресу. Moodle доцільно використовувати для побудови повноцінного електронного курсу, однак за великої кількості матеріалів пошук і повторне використання окремих ресурсів між різними дисциплінами може бути складним. Приклад інтерфейсу Moodle наведено на рисунку 1.2.

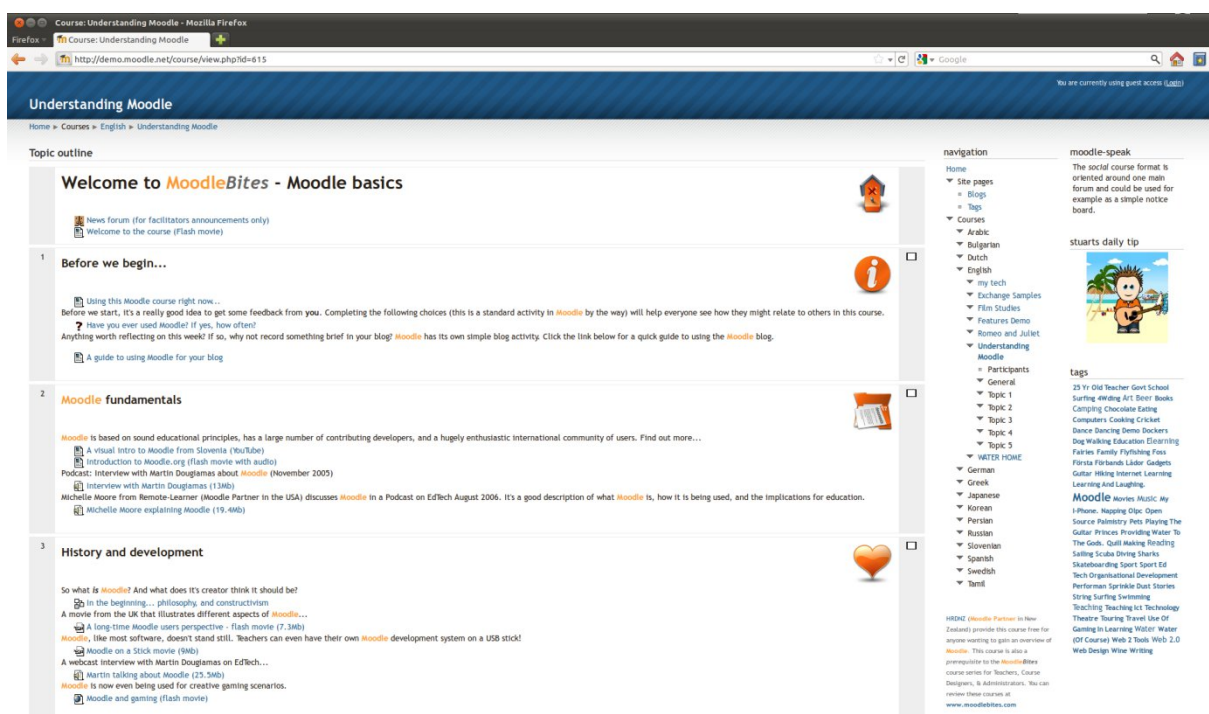


Рис. 1.2. Приклад інтерфейсу навчального курсу в Moodle

Google Classroom орієнтований на просту організацію занять, завдань і навчальних матеріалів у межах класу або групи. Викладач може додавати матеріали, завдання, теми, посилання, файли з Google Drive і відео. Перевагою Google Classroom є простота використання та інтеграція з сервісами Google, однак система більше призначена для поточного навчального процесу, ніж для побудови повноцінного каталогу електронних ресурсів із розширеними метаданими, заявками на доступ і бібліотечною модерацією. Приклад інтерфейсу наведено на рисунку 1.3.

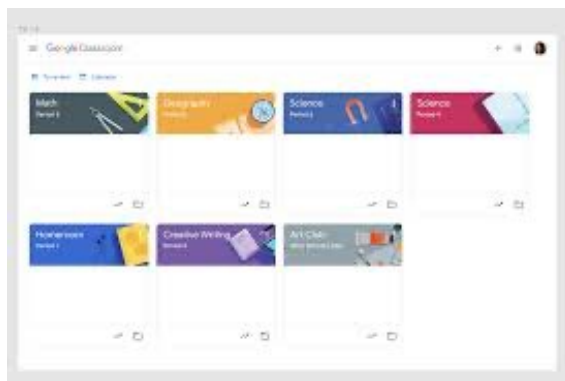


Рис. 1.3. Приклад інтерфейсу Google Classroom

Canvas LMS використовується для організації навчальних курсів, модулів, завдань, файлів, обговорень і тестів. Система має зручний модульний підхід до структурування матеріалів і підтримує роботу студентів та викладачів у цифровому навчальному середовищі. Водночас Canvas залишається передусім LMS-платформою, а не спеціалізованою бібліотекою ресурсів. Для SmartLibrary важливо використати ідею структурованих модулів, але зробити акцент на каталозі, пошуку, заявках на доступ і аналітиці використання матеріалів. Інтерфейс Canvas LMS наведено на рисунку 1.4.

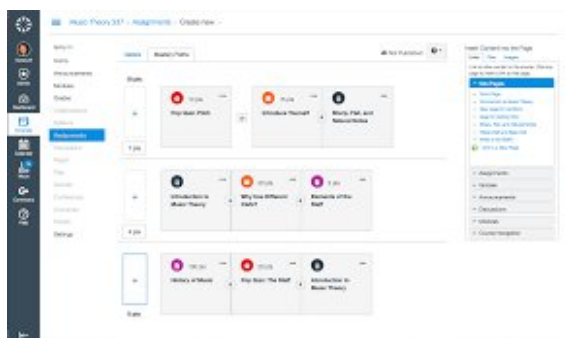


Рис. 1.4. Приклад інтерфейсу Canvas LMS

Blackboard Learn є комплексною LMS-платформою для закладів освіти, яка підтримує курси, навчальні матеріали, завдання, тести, оцінювання та взаємодію між учасниками навчального процесу. Перевагою рішення є розвинена інфраструктура для управління курсами та освітнім процесом. Недоліком у контексті розроблюваної платформи є складність, комерційна спрямованість і відсутність вузької орієнтації на простий каталог електронних

ресурсів із гнучкими заявками доступу. Приклад інтерфейсу Blackboard Learn наведено на рисунку 1.5.

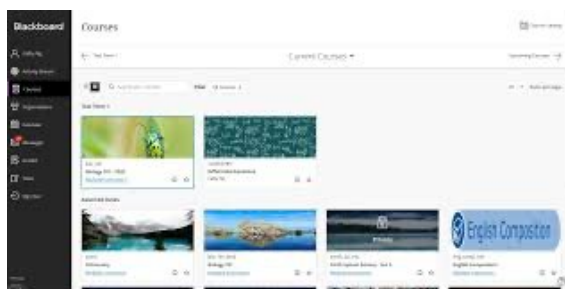


Рис. 1.5. Приклад інтерфейсу Blackboard Learn

DSpace є відкритою платформою для створення цифрових репозитаріїв, яка використовується університетами, науковими бібліотеками та організаціями для довготривалого зберігання цифрових матеріалів. DSpace добре підходить для інституційних репозитаріїв, дисертацій, наукових публікацій, наборів даних і архівних матеріалів. Водночас для повсякденної роботи зі студентськими навчальними ресурсами система може бути надмірно складною, оскільки її основний акцент зроблено на репозитарному зберіганні та метаданих, а не на швидкому навчальному сценарії пошуку, доступу і перегляду матеріалів.

Таблиця 1.2

Порівняння існуючих рішень із розроблюваною платформою SmartLibrary

Система	Розміщення ресурсів	Пошук і каталог	Заявки на доступ	Аналitика	Основне призначення
Moodle	Так	Так	Частково	Частково	Повноцінне LMS-середовище для курсів
Google Classroom	Так	Обмежено	Ні	Частково	Просте керування класами та завданнями
Canvas LMS	Так	Так	Частково	Так	Модуль на LMS-

					платформа для курсів
--	--	--	--	--	----------------------

Продовження таблиці 1.2

Blackboard Learn	Так ак	Так ак	Частково	Так	Корпоративна LMS для закладів освіти
DSpace	Так ак	Так ак	Так	Частково	Інституційний цифровий репозитарій
SmartLibrary	Так ак	Так ак	Так	Так	Спеціалізована платформа електронних навчальних ресурсів

Порівняльний аналіз показує, що більшість існуючих рішень орієнтована або на повний цикл дистанційного навчання, або на архівне зберігання цифрових матеріалів. Розроблювана веб-платформа SmartLibrary має поєднати переваги LMS і електронного репозитарію, але залишитися простішою для користувача: забезпечити каталог навчальних матеріалів, пошук, відкриття ресурсу, подання заявки на доступ, роботу бібліотекаря з ресурсами та формування аналітичних звітів.

1.3 Моделювання предметної області

Моделювання предметної області веб-платформи SmartLibrary виконано з використанням UML-діаграм, які дозволяють формалізувати ролі користувачів, основні функції системи та послідовність взаємодії між модулями. У межах моделі виділено чотири основні ролі: студент, викладач або автор, бібліотекар або модератор та адміністратор.

На рисунку 1.6 наведено діаграму прецедентів. Вона демонструє, що студент може авторизуватися, переглядати каталог ресурсів, виконувати пошук і фільтрацію, відкривати електронний ресурс, подавати заявку на доступ,

переглядати історію та власну бібліотеку. Викладач або автор має можливість працювати з ресурсами, а бібліотекар або модератор відповідає за керування ресурсами та опрацювання заявок. Адміністратор додатково керує користувачами, ролями та переглядає аналітику.

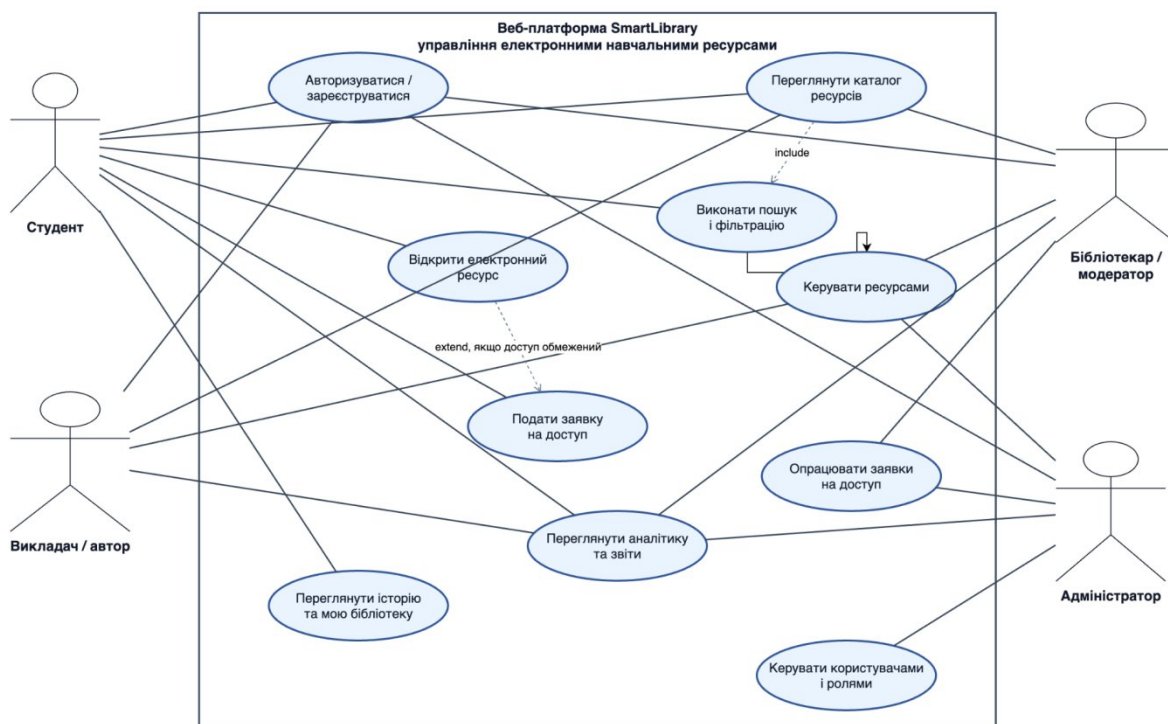


Рис. 1.6. Діаграма прецедентів веб-платформи SmartLibrary

Для деталізації сценарію доступу до ресурсу побудовано діаграму послідовності, наведену на рисунку 1.7. Вона відображає взаємодію студента, інтерфейсу, сервісу авторизації, сервісу ресурсів, сервісу заявок, бази даних і бібліотекаря. Після введення логіна і пароля система перевіряє користувача в базі даних, відкриває каталог ресурсів і надає список матеріалів. Якщо студент обирає ресурс з обмеженим доступом, система перевіряє права або створює заявку та повідомляє бібліотекаря.

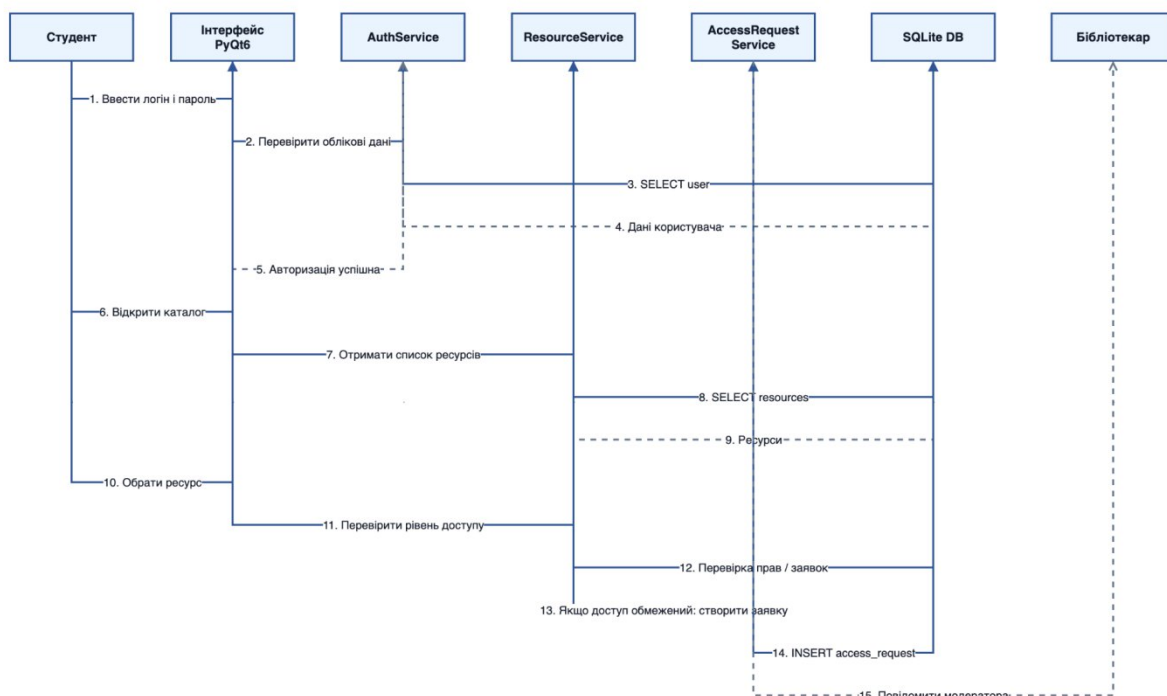


Рис. 1.7. Діаграма послідовності відкриття ресурсу та створення заявки

Діаграма активності на рисунку 1.8 описує логіку роботи користувача з платформою. Спочатку користувач відкриває програму SmartLibrary Pro, проходить авторизацію або реєстрацію, після чого переходить до каталогу електронних ресурсів. Далі виконується пошук і вибір матеріалу. Якщо доступ дозволено, ресурс відкривається, а подія записується в історію та аналітику. Якщо доступ обмежено, користувач подає заявку, яка опрацьовується модератором.

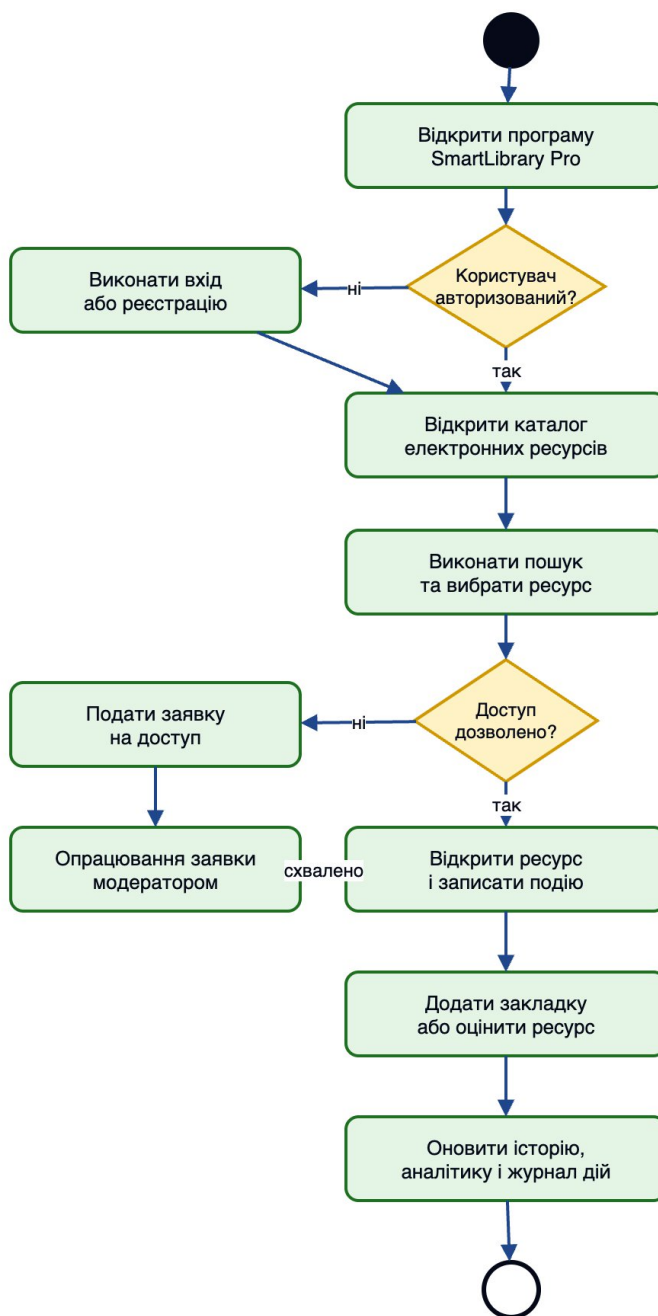


Рис. 1.8. Діаграма активності роботи користувача з платформою SmartLibrary

Побудовані UML-моделі підтверджують, що веб-платформа SmartLibrary повинна мати модульну структуру. Окремі частини системи мають відповідати за авторизацію, каталог ресурсів, управління доступом, заявки, аналітику та журнал подій. Такий підхід спрощує подальше проектування бази даних, програмних компонентів і алгоритмів роботи.

1.4 Аналіз вимог інформаційної системи

Аналіз вимог до веб-платформи SmartLibrary виконується з урахуванням функцій, визначених під час аналізу предметної області та моделювання. Система має забезпечити стабільну роботу з електронними навчальними ресурсами, контроль прав доступу, зручний пошук, можливість подання заявок і формування звітів для адміністратора та бібліотекаря.

Таблиця 1.3

Функціональні вимоги до веб-платформи SmartLibrary

№	Вимога	Опис
F1	Реєстрація та авторизація користувачів	Вхід до системи за логіном і паролем, визначення ролі користувача
F2	Керування ролями	Підтримка ролей студента, викладача, бібліотекаря та адміністратора
F3	Каталог електронних ресурсів	Перегляд навчальних матеріалів з описом, категоріями та параметрами доступу
F4	Пошук і фільтрація	Пошук ресурсів за назвою, автором, дисципліною, типом, тегами та рівнем доступу
F5	Відкриття ресурсу	Перегляд доступного електронного матеріалу та запис події доступу
F6	Заявки на доступ	Створення заявки на перегляд обмеженого ресурсу
F7	Опрацювання заявок	Погодження або відхилення заявки бібліотекарем чи модератором
F8	Керування ресурсами	Додавання, редагування, публікація, архівування й видалення ресурсів
F9	Аналітика та звіти	Формування статистики переглядів, популярності ресурсів і активності користувачів
F10	Журнал подій	Фіксація авторизації, відкриття ресурсів, створення заявок і дій адміністратора

Нефункціональні вимоги визначають якість роботи системи. Для SmartLibrary важливими є швидке відкриття каталогу, коректна робота з багатьма записами ресурсів, захист персональних даних, збереження журналу подій і зручність використання інтерфейсу. Оскільки система призначена для навчального закладу, інтерфейс має бути зрозумілим для студентів і викладачів без додаткового навчання.

Таблиця 1.4

Нефункціональні та безпекові вимоги до системи

№	Вимога	Опис
NF1	Продуктивність	Відкриття каталогу та пошуку без помітних затримок для користувача
NF2	Зручність використання	Проста навігація, зрозумілі кнопки, фільтри та структура каталогу
NF3	Надійність	Збереження даних про ресурси, заявки та події без втрати записів
NF4	Масштабованість	Можливість збільшення кількості ресурсів, користувачів і категорій
S1	Аутентифікація	Перевірка облікових даних перед доступом до функцій системи
S2	Рольовий доступ	Обмеження операцій відповідно до ролі користувача
S3	Журналювання	Фіксація важливих операцій для контролю змін і аудиту
S4	Захист ресурсів	Недоступність обмежених матеріалів без заявки або дозволу

Сформовані вимоги показують, що SmartLibrary має виконувати не лише роль сховища файлів, а й роль контрольованої освітньої платформи. Важливою особливістю є наявність заявок на доступ, оскільки частина навчальних матеріалів може бути доступною лише для певних груп, дисциплін або користувачів.

1.5 Постановка завдання

На основі аналізу предметної області, існуючих рішень, UML-моделювання та сформованих вимог визначено завдання розроблення веб-платформи SmartLibrary для управління електронними навчальними ресурсами. Метою розроблення є створення інформаційної системи, яка забезпечує централізований каталог навчальних матеріалів, контрольований доступ до ресурсів, подання та опрацювання заявок, ведення журналу подій і формування аналітичних звітів.

Вхідними даними системи є облікові дані користувачів, ролі, відомості про електронні ресурси, категорії, файли або посилання, метадані ресурсів, заявки на доступ, дії користувачів і параметри фільтрації. До метаданих ресурсу належать назва, автор, дисципліна, тип матеріалу, опис, ключові слова, дата додавання, рівень доступу та статус публікації.

Вихідними даними системи є список доступних електронних ресурсів, результати пошуку, сторінка перегляду ресурсу, статус заявки на доступ, журнал подій, статистика популярності матеріалів, звіти про активність користувачів і перелік ресурсів за категоріями. Для адміністратора та бібліотекаря система має формувати узагальнені дані про кількість ресурсів, кількість заявок, найпопулярніші матеріали та активність користувачів.

У межах розроблення необхідно реалізувати такі основні задачі: створити механізм реєстрації та авторизації користувачів; реалізувати рольову модель доступу; розробити каталог електронних ресурсів; забезпечити пошук і фільтрацію; реалізувати модуль заявок на доступ; створити модуль адміністрування ресурсів; забезпечити ведення журналу подій; реалізувати аналітичні звіти; підготувати базу даних для зберігання користувачів, ресурсів, заявок і подій.

Розроблювана веб-платформа має бути орієнтована на використання в навчальному закладі. Вона повинна спростити доступ студентів до навчальних матеріалів, підвищити контроль за актуальністю електронних ресурсів, надати

викладачам і бібліотекарям зручний інструмент керування матеріалами та забезпечити адміністратора системною аналітикою.

1.6 Висновки до першого розділу

У першому розділі проведено аналіз предметної області веб-платформи SmartLibrary для управління електронними навчальними ресурсами. Визначено, що основною проблемою є фрагментарне зберігання навчальних матеріалів у різних сервісах, складність пошуку актуальних ресурсів і відсутність єдиного механізму контролю доступу до обмежених матеріалів.

Проаналізовано існуючі рішення Moodle, Google Classroom, Canvas LMS, Blackboard Learn та DSpace. Встановлено, що вони мають розвинені можливості для дистанційного навчання або репозитарного зберігання, однак не повністю відповідають задачі створення простої спеціалізованої платформи для каталогу електронних навчальних ресурсів із заявками на доступ, бібліотечною модерацією та аналітикою використання.

Виконано моделювання предметної області за допомогою DFD-діаграми, діаграми прецедентів, діаграми послідовності та діаграми активності. Побудовані моделі дозволили уточнити ролі користувачів, основні процеси системи, інформаційні потоки та сценарії доступу до ресурсів. На основі цього сформовано функціональні, нефункціональні та безпекові вимоги до веб-платформи.

Результати першого розділу є основою для подальшого проектування інформаційного та програмного забезпечення SmartLibrary. Вони визначають структуру майбутньої бази даних, склад програмних модулів, правила доступу, алгоритми роботи з ресурсами та вимоги до тестування розробленої системи.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних веб-платформи SmartLibrary

Логічна модель даних веб-платформи SmartLibrary визначає структуру зберігання інформації про користувачів, електронні навчальні ресурси, заявки на доступ, закладки, оцінки, події доступу та журнал адміністративних дій. Вона є основою для подальшого проєктування фізичної бази даних і програмних модулів, оскільки відображає склад сутностей предметної області та зв'язки між ними.

У межах системи передбачено декілька категорій користувачів: студент, викладач або автор, бібліотекар-модератор та адміністратор. Кожна роль має окремий набір прав. Студент переглядає каталог, відкриває доступні ресурси, залишає закладки та оцінки. Викладач або автор додає навчальні матеріали та відстежує їх використання. Бібліотекар опрацьовує заявки на доступ і контролює якість описів ресурсів. Адміністратор керує користувачами, ролями, довідниками та переглядає аналітичні звіти.

На рис. 2.1 наведено логічну модель даних, що використовується для побудови інформаційної бази SmartLibrary. Центральною сутністю є Resource, оскільки саме електронний навчальний ресурс є основним об'єктом каталогу. До нього прив'язуються заявки на доступ, закладки, оцінки та події перегляду.

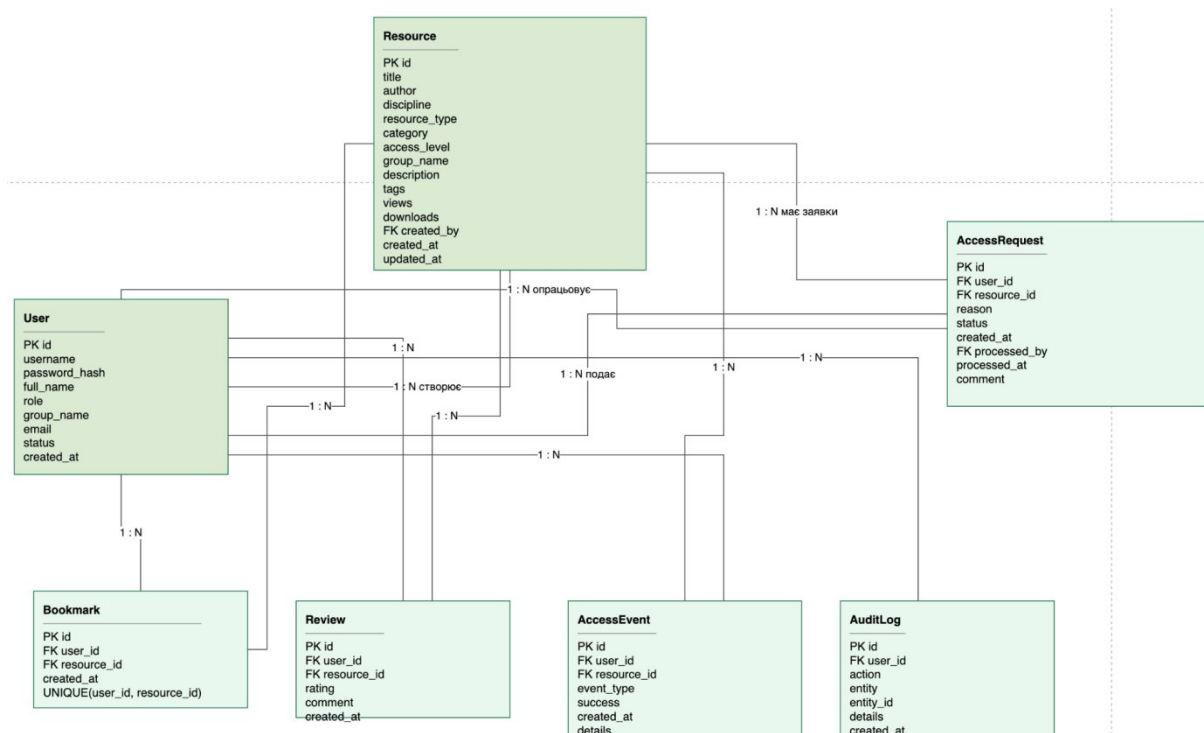


Рис. 2.1 - Логічна модель даних веб-платформи SmartLibrary

Сутність User містить облікові дані користувача, роль, групу або навчальний підрозділ, електронну адресу, статус і дату створення запису. Зв'язок між User та Resource реалізує сценарій створення матеріалів: один користувач може додати багато навчальних ресурсів, але кожний ресурс має одного автора або відповідального користувача.

Сутність Resource описує електронний навчальний матеріал. Вона містить назву, автора, дисципліну, тип ресурсу, категорію, рівень доступу, навчальну групу, опис, теги, кількість переглядів, кількість завантажень, дату створення та дату оновлення. Такий склад атрибутів дозволяє забезпечити пошук, фільтрацію, аналітику та контроль доступності ресурсу для різних груп користувачів.

Сутність AccessRequest використовується для фіксації заявок на доступ до ресурсів, які мають обмежений рівень доступу. Заявка містить посилання на користувача, ресурс, причину звернення, статус, дату створення, інформацію про користувача, який її опрацював, дату опрацювання та коментар. Завдяки цьому система підтримує не лише відкритий перегляд матеріалів, а й регламентований доступ до навчальних ресурсів.

Додаткові сутності Bookmark, Review, AccessEvent та AuditLog забезпечують персоналізацію, зворотний зв'язок, аналітику та аудит. Закладки дозволяють користувачу формувати власну бібліотеку. Відгуки зберігають оцінки ресурсів. Події доступу фіксують відкриття матеріалів, обмеження доступу або створення заявки. Журнал аудиту використовується для контролю дій користувачів і модераторів.

Таблиця 2.1

Сутності логічної моделі даних

Сутність	Призначення	Основні зв'язки
User	Зберігання облікових даних користувачів, ролей, груп і статусів	Створює ресурси, подає заявки, залишає закладки, оцінки та події доступу
Resource	Зберігання опису електронних навчальних ресурсів	Пов'язується із заявками, закладками, оцінками та подіями доступу
AccessRequest	Фіксація заявок на отримання доступу до обмежених ресурсів	Належить користувачу та ресурсу, опрацьовується бібліотекарем або адміністратором
Bookmark	Збереження персонального списку обраних ресурсів	Пов'язує користувача з ресурсом
Review	Зберігання оцінок і коментарів до ресурсів	Пов'язує користувача з ресурсом
AccessEvent	Фіксація подій перегляду, відкриття або обмеження доступу	Належить користувачу та ресурсу
AuditLog	Журналювання службових і адміністративних дій	Фіксує дії користувачів над сутностями системи

Запропонована логічна модель даних забезпечує нормалізоване зберігання інформації без дублювання основних сутностей. Вона дає змогу реалізувати

каталог ресурсів, персональну бібліотеку користувача, модуль заявок на доступ, аналітику переглядів та журналювання дій. Така структура є достатньою для подальшого створення фізичної бази даних і програмних модулів веб-платформи SmartLibrary.

2.2 Діаграма класів і кооперації веб-платформи SmartLibrary

Діаграма класів відображає об'єктно-орієнтовану структуру програмного забезпечення та показує, які класи використовуються для реалізації основних функцій веб-платформи SmartLibrary. Модель класів узгоджена з логічною моделлю даних і враховує функціональні сценарії роботи користувача, бібліотекаря та адміністратора.

На рис. 2.2 подано діаграму класів системи. Основними класами є User, Resource, AccessRequest, Bookmark, Review, AccessEvent, AuditLog та Database. Клас Database виконує роль програмного шару доступу до SQLite-бази та забезпечує виконання запитів, автентифікацію, створення користувачів і журналювання операцій.

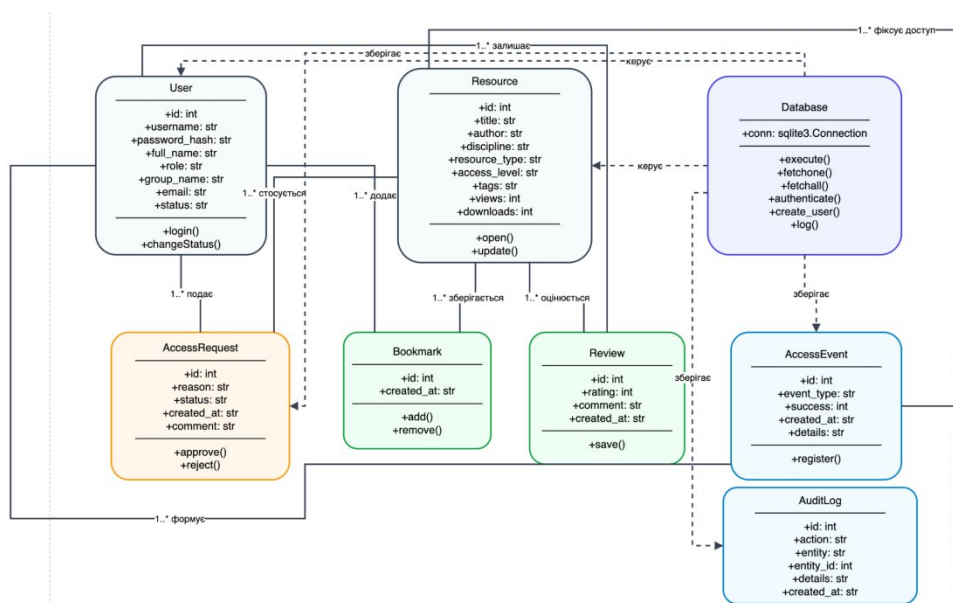


Рис. 2.2 - Діаграма класів веб-платформи SmartLibrary

Клас User описує користувача системи та містить атрибути ідентифікатора, логіна, хешу пароля, повного імені, ролі, навчальної групи, електронної пошти та статусу. Методи login() і changeStatus() забезпечують вхід до системи та зміну стану облікового запису.

Клас Resource відповідає за електронний навчальний ресурс. Він містить назву, автора, дисципліну, тип ресурсу, рівень доступу, теги, кількість переглядів і завантажень. Методи open() та update() використовуються для відкриття ресурсу та оновлення його опису.

Клас AccessRequest реалізує заявку на доступ до обмеженого ресурсу. Він містить причину, статус, дату створення та коментар модератора. Методи approve() і reject() забезпечують схвалення або відхилення заявки. Класи Bookmark і Review використовуються для персоналізації роботи користувача, а AccessEvent та AuditLog забезпечують аналітику й аудит.

Таблиця 2.2

Компоненти діаграми класів

Клас	Відповідальність	Ключові методи
User	Зберігання профілю користувача, ролі, групи та стану облікового запису	login(), changeStatus()
Resource	Опис і керування електронним навчальним ресурсом	open(), update()
AccessRequest	Опрацювання заявок на доступ до обмежених ресурсів	approve(), reject()
Bookmark	Формування персональної бібліотеки користувача	add(), remove()
Review	Збереження оцінок і коментарів до ресурсів	save()
AccessEvent	Фіксація подій доступу до ресурсів	register()
AuditLog	Журналювання адміністративних і службових операцій	create(), filter()

Database	Виконання SQL-запитів і централізований доступ до даних	execute(), fetchone(), fetchall(), authenticate(), log()
----------	--	--

Для деталізації взаємодії класів у роботі побудовано три кооперації, які описують найбільш важливі сценарії: авторизацію користувача, перегляд ресурсу з перевіркою доступу та формування аналітичного звіту.

Перша кооперація описує процес авторизації користувача. Користувач вводить логін і пароль у діалоговому вікні, після чого LoginDialog передає дані до Database. База даних виконує вибірку активного користувача з таблиці users, повертає результат перевірки та фіксує подію входу в audit_log. У разі успішної перевірки відкривається головне вікно системи. Кооперацію авторизації наведено на рис. 2.3.

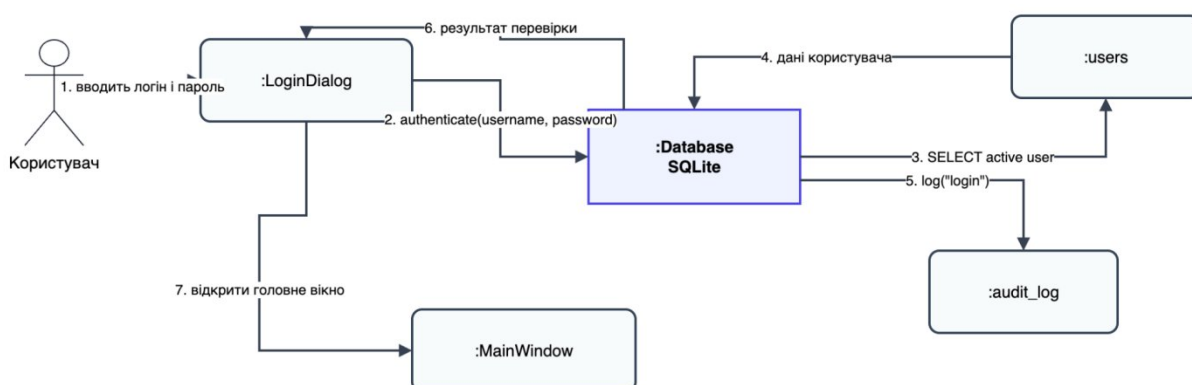


Рис. 2.3 - Кооперація класів під час авторизації користувача

Друга кооперація відображає сценарій пошуку, вибору та відкриття електронного навчального ресурсу. Студент працює зі сторінкою каталогу, обирає ресурс, після чого сервіс ресурсів звертається до бази даних і перевіряє рівень доступу. Якщо доступ дозволено, система реєструє подію перегляду. Якщо доступ обмежено, створюється запис у access_requests, а бібліотекар отримує підставу для подальшого опрацювання заявки. Кооперацію перегляду ресурсу наведено на рис. 2.4.

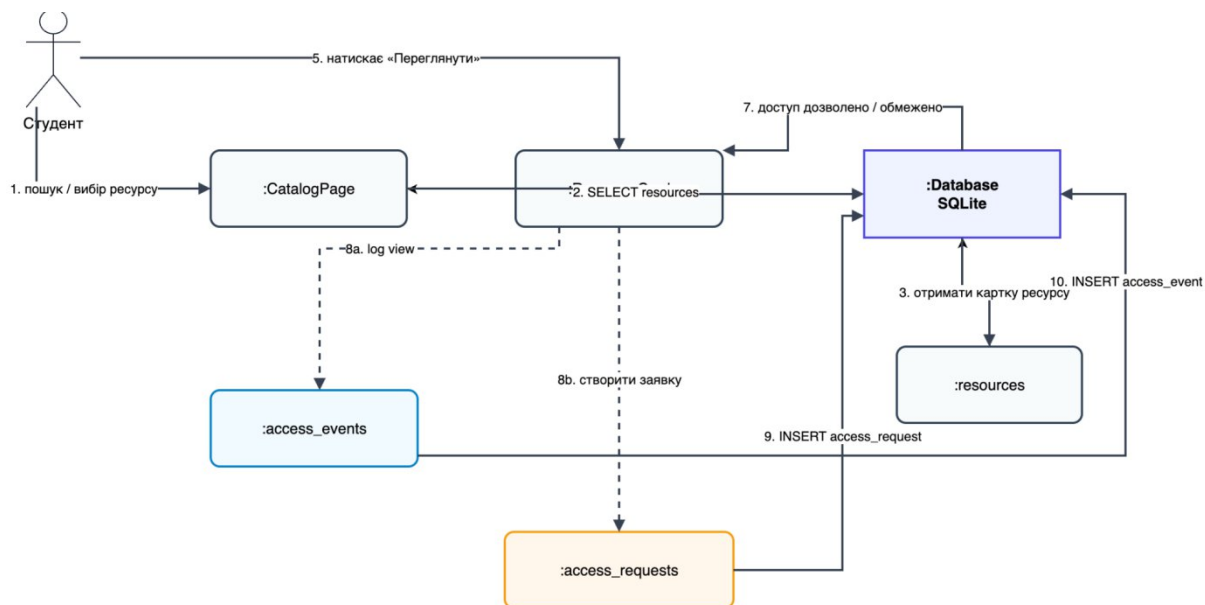


Рис. 2.4 - Кооперація класів під час перегляду ресурсу та створення заявки

Третя кооперація описує формування аналітичного звіту. Адміністратор або бібліотекар обирає функцію експорту звіту, після чого сторінка аналітики передає запит до ReportService. Сервіс отримує з бази даних показники щодо ресурсів і подій доступу, формує звіт у форматі HTML або CSV та повертає шлях до створеного файлу користувачу. Цей сценарій подано на рис. 2.5.

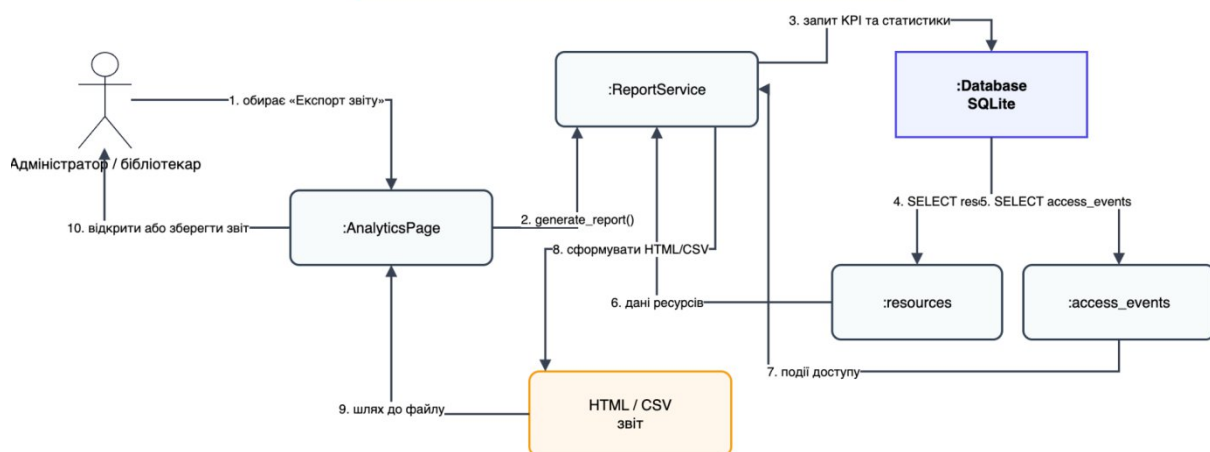


Рис. 2.5 - Кооперація класів під час формування аналітичного звіту

Побудовані кооперації демонструють, що основні сценарії роботи системи виконуються через взаємодію інтерфейсного шару, сервісних класів і бази даних. Такий підхід забезпечує розділення відповідальності між частинами

програмного забезпечення, спрощує тестування та дає змогу розширювати функціонал без повної зміни архітектури.

2.3 Діаграма компонентів

Діаграма компонентів описує склад програмного забезпечення SmartLibrary на рівні функціональних модулів. Вона показує, які частини системи відповідають за інтерфейс користувача, авторизацію, каталог ресурсів, заявки на доступ, аналітику, експорт звітів, журналювання та збереження даних.

На рис. 2.6 наведено компонентну модель веб-платформи SmartLibrary. Центральним елементом є база даних SQLite, з якою взаємодіють модулі авторизації, каталогу, заявок, аналітики, звітів та аудиту. Інтерфейсна частина реалізована як PyQt6 UI, що містить головне вікно, сторінки та діалоги системи.

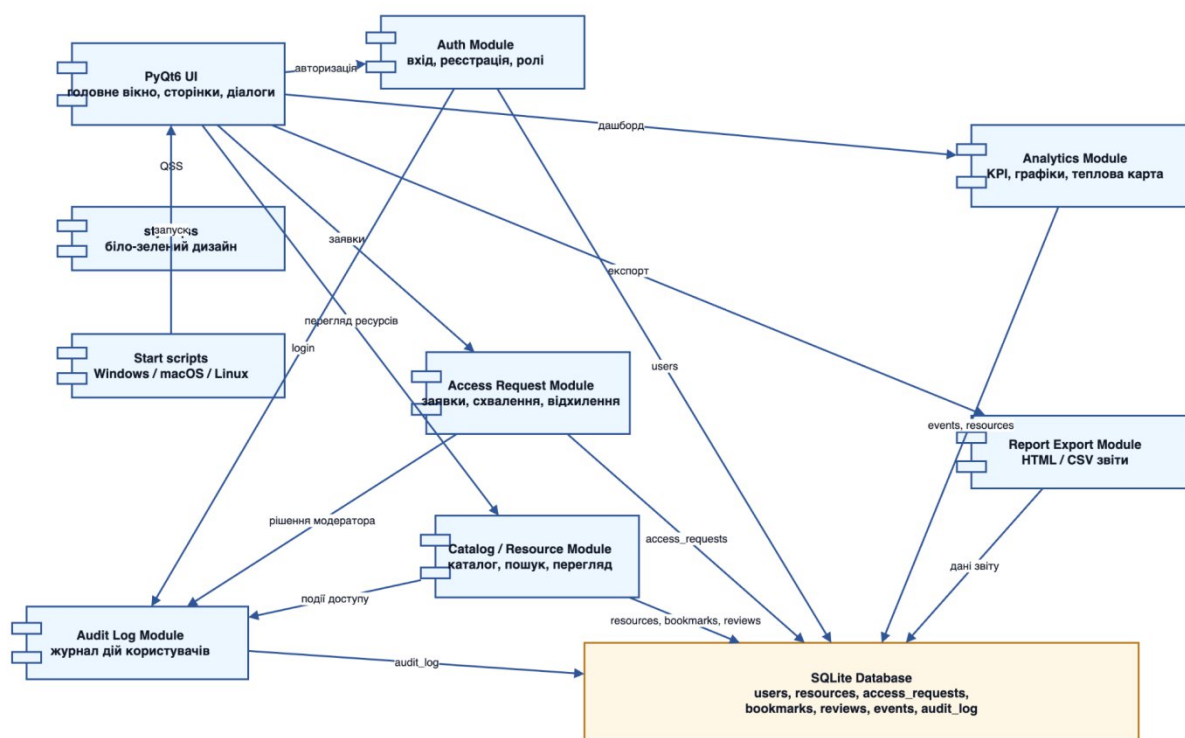


Рис. 2.6 - Діаграма компонентів веб-платформи SmartLibrary

Компонент Auth Module відповідає за вхід, реєстрацію та контроль ролей. Він отримує дані з інтерфейсу, перевіряє їх у базі даних і повертає інформацію

про користувача. Компонент Catalog / Resource Module забезпечує перегляд каталогу, пошук, фільтрацію, відкриття ресурсів, роботу із закладками та оцінками.

Компонент Access Request Module реалізує створення, схвалення та відхилення заявок на доступ. Він використовується тоді, коли студент або викладач намагається відкрити ресурс з обмеженим доступом. Компонент Analytics Module формує ключові показники використання платформи, а Report Export Module відповідає за експорт звітів у HTML або CSV.

Компонент Audit Log Module фіксує службові дії користувачів: вхід, перегляд ресурсів, створення заявок, опрацювання рішень та експорт звітів. Наявність такого компонента підвищує контрольованість системи, оскільки адміністратор може відстежити основні події роботи платформи.

Таблиця 2.3

Компоненти діаграми компонентів

Компонент	Призначення	Дані, з якими працює компонент
PyQt6 UI	Відображення сторінок, діалогів, таблиць, форм і повідомлень	Дані інтерфейсу, введені користувачем параметри
Auth Module	Авторизація, реєстрація, перевірка ролей	users, audit_log
Catalog / Resource Module	Каталог ресурсів, пошук, перегляд, закладки, оцінки	resources, bookmarks, reviews, access_events
Access Request Module	Створення та опрацювання заявок на доступ	access_requests, users, resources
Analytics Module	Побудова KPI, графіків і теплової карти активності	resources, access_events
Report Export Module	Формування HTML/CSV-звітів	Дані ресурсів, подій доступу та аналітики
Audit Log Module	Журналювання дій користувачів і модераторів	audit_log

Продовження таблиці 2.3

SQLite Database	Локальне збереження інформаційної бази системи	users, resources, access_requests, bookmarks, reviews, events, audit_log
--------------------	---	--

Компонентна модель дає змогу визначити межі відповідальності окремих частин системи. Завдяки цьому інтерфейс користувача не виконує безпосередньо складних операцій з даними, а звертається до відповідних сервісних компонентів. Це підвищує структурованість програмного забезпечення та спрощує його подальше тестування.

2.4 Діаграма пакетів веб-платформи SmartLibrary

Діаграма пакетів відображає організацію програмного коду SmartLibrary у вигляді логічних груп файлів і модулів. Такий поділ необхідний для підтримки зрозумілої структури проекту, полегшення супроводу та можливості подальшого розширення системи.

На рис. 2.7 наведено пакетну структуру програмного забезпечення. Вона включає пакети ui, dialogs, services, domain, persistence, analytics, reports та assets/style. Кожний пакет має окреме призначення і взаємодіє з іншими пакетами через визначені залежності.

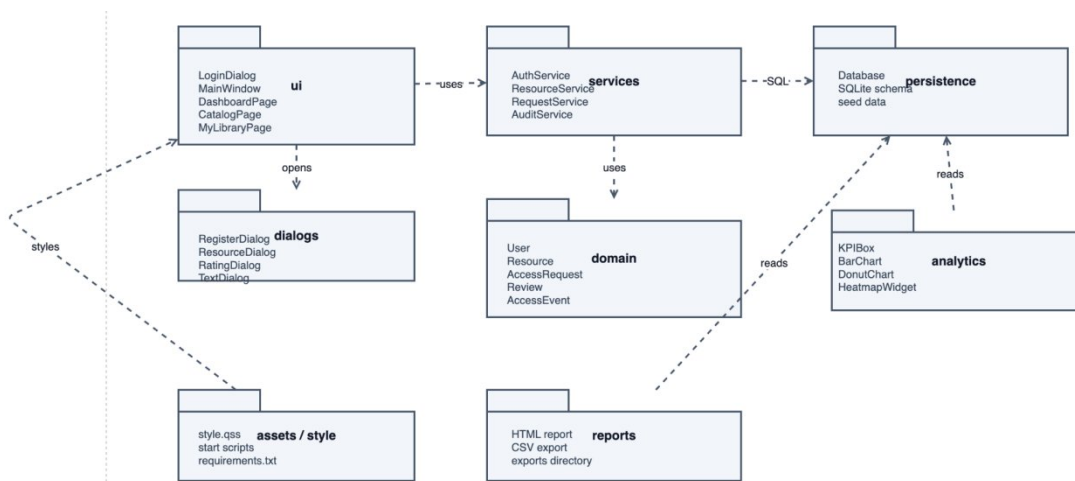


Рис. 2.7 - Діаграма пакетів веб-платформи SmartLibrary

Пакет `ui` містить основні сторінки застосунку: `LoginDialog`, `MainWindow`, `DashboardPage`, `CatalogPage` та `MyLibraryPage`. Він відповідає за відображення даних, реакцію на дії користувача та передавання запитів до сервісного шару. Пакет `dialogs` містить окремі діалогові вікна для реєстрації, перегляду ресурсу, оцінювання та введення додаткового тексту.

Пакет `services` містить прикладну логіку: `AuthService`, `ResourceService`, `RequestService` та `AuditService`. Ці класи обробляють запити з інтерфейсу та взаємодіють із доменними об'єктами й базою даних. Пакет `domain` містить основні об'єкти предметної області: `User`, `Resource`, `AccessRequest`, `Review` та `AccessEvent`.

Пакет `persistence` відповідає за фізичне збереження даних і містить модуль `Database`, SQL-схему та початкові тестові дані. Пакет `analytics` містить елементи візуалізації, зокрема `KPIBox`, `BarChart`, `DonutChart` та `HeatmapWidget`. Пакет `reports` використовується для створення HTML-звітів, CSV-експорту та збереження сформованих файлів у каталозі `exports`. Пакет `assets/style` містить файл `style.qss`, стартові скрипти та файл залежностей `requirements.txt`.

Таблиця 2.4

Пакети системи

Пакет	Склад пакета	Функціональне призначення
<code>ui</code>	<code>LoginDialog</code> , <code>MainWindow</code> , <code>DashboardPage</code> , <code>CatalogPage</code> , <code>MyLibraryPage</code>	Основний графічний інтерфейс користувача
<code>dialogs</code>	<code>RegisterDialog</code> , <code>ResourceDialog</code> , <code>RatingDialog</code> , <code>TextDialog</code>	Додаткові діалогові вікна введення та перегляду даних
<code>services</code>	<code>AuthService</code> , <code>ResourceService</code> , <code>RequestService</code> , <code>AuditService</code>	Прикладна логіка та оброблення запитів
<code>domain</code>	<code>User</code> , <code>Resource</code> , <code>AccessRequest</code> , <code>Review</code> , <code>AccessEvent</code>	Об'єкти предметної області системи

Продовження таблиці 2.4

persistence	Database, SQLite schema, seed data	Створення, оновлення та читання бази даних
analytics	KPIBox, BarChart, DonutChart, HeatmapWidget	Побудова аналітичних показників і графіків
reports	HTML report, CSV export, exports directory	Формування та збереження звітів
assets / style	style.qss, start scripts, requirements.txt	Оформлення інтерфейсу, запуск і залежності проєкту

Пакетна структура забезпечує логічне розділення інтерфейсу, бізнес-логіки, доменних об'єктів, збереження даних, аналітики та звітів. Це спрощує навігацію по проєкті, зменшує зв'язність між частинами програми та дає можливість у майбутньому замінити локальну SQLite-базу на серверну СУБД або перенести інтерфейсну частину у вебзастосунок без повної зміни логіки системи.

2.5 Висновки до розділу 2

У другому розділі виконано проєктування інформаційного та програмного забезпечення веб-платформи SmartLibrary для управління електронними навчальними ресурсами. Розроблено логічну модель даних, яка відображає основні сутності системи: користувачів, електронні ресурси, заявки на доступ, закладки, відгуки, події доступу та журнал аудиту.

Побудована логічна модель забезпечує структуроване зберігання інформації та підтримує основні сценарії роботи платформи: авторизацію, перегляд каталогу, пошук і фільтрацію ресурсів, створення заявок на доступ, ведення персональної бібліотеки, оцінювання ресурсів та формування аналітики.

Розроблено діаграму класів, яка описує програмну структуру системи на рівні об'єктів. Визначено основні атрибути та методи класів User, Resource,

AccessRequest, Bookmark, Review, AccessEvent, AuditLog і Database. Для деталізації взаємодії між класами побудовано кооперації авторизації, перегляду ресурсу з перевіркою доступу та формування аналітичного звіту.

Сформовано компонентну модель програмного забезпечення, у якій виділено модулі інтерфейсу, авторизації, каталогу ресурсів, заявок на доступ, аналітики, експорту звітів, журналювання та бази даних. Також розроблено діаграму пакетів, яка показує організацію коду за логічними групами: ui, dialogs, services, domain, persistence, analytics, reports та assets/style.

Отримані результати є основою для подальшої реалізації програмного забезпечення, створення фізичної моделі бази даних, алгоритмізації основних процесів і тестування веб-платформи SmartLibrary.

3 РОЗРОБКА ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору технологічних засобів реалізації

Розроблення програмного забезпечення веб-платформи SmartLibrary виконано з урахуванням потреб електронного навчального середовища, у якому користувачі повинні швидко знаходити навчальні матеріали, відкривати доступні ресурси, подавати заявки на обмежені матеріали, залишати закладки й оцінки, а бібліотекар або модератор має змогу керувати каталогом, заявками та звітною інформацією. Основною метою реалізації є створення працездатного програмного прототипу, який відтворює ключову логіку веб-платформи управління електронними навчальними ресурсами.

У межах роботи реалізовано прикладну систему SmartLibrary Pro, що моделює основні функції веб-платформи у вигляді локального програмного застосунку з графічним інтерфейсом. Такий підхід дає змогу продемонструвати повний цикл роботи системи без складного серверного розгортання: авторизацію користувачів, перегляд каталогу, перевірку прав доступу, створення заявок, фіксацію подій, формування аналітики та експорт звітів. Для збереження даних використано локальну базу SQLite, а основні програмні модулі структуровано відповідно до функцій, визначених на етапі проектування.

Мовою програмування обрано Python, оскільки вона має достатні засоби для швидкого створення інформаційних систем, підтримує роботу з базами даних, файлами, звітами та графічними інтерфейсами. Для побудови інтерфейсу використовується PyQt6, що дозволяє створити вікна, діалоги, таблиці, панелі навігації та форми введення даних. Для збереження інформації обрано SQLite, оскільки ця СУБД не потребує окремого сервера, зберігає дані в одному файлі та є придатною для навчального й демонстраційного програмного продукту.

Таблиця 3.1

Технологічні засоби реалізації програмного забезпечення SmartLibrary

Засіб	Призначення у системі
Python	Реалізація бізнес-логіки, алгоритмів доступу, оброблення заявок, журналювання та експорту даних
PyQt6	Побудова графічного інтерфейсу користувача, сторінок каталогу, аналітики та діалогових вікон
SQLite	Локальне збереження користувачів, ресурсів, заявок, закладок, оцінок, подій доступу та журналу аудиту
SQL	Створення таблиць, вибірка ресурсів, перевірка доступу, отримання статистики та формування звітів
HTML / CSV	Формування звітних файлів для перегляду, збереження та подальшого опрацювання
SHA-256	Хешування паролів користувачів для базового захисту облікових даних

Використання зазначених засобів забезпечує автономність програмного продукту, простоту встановлення та достатню функціональність для демонстрації повного циклу роботи веб-платформи SmartLibrary. У подальшому логіку системи можна перенести у клієнт-серверну архітектуру з веб-інтерфейсом, API-сервером та централізованою СУБД, однак для кваліфікаційної роботи обраний варіант дозволяє повністю показати структуру даних, алгоритми та роботу основних модулів.

3.2 Розробка фізичної моделі бази даних

Фізична модель бази даних веб-платформи SmartLibrary розроблена на основі логічної моделі, поданої у другому розділі. Вона відображає реальні таблиці, поля, типи даних, первинні й зовнішні ключі, а також зв'язки між сутностями. Основними об'єктами зберігання є користувачі, електронні

навчальні ресурси, заявки на доступ, закладки, відгуки, події доступу та журнал аудиту.

Фізичну модель даних SmartLibrary наведено на рис. 3.1. Центральними таблицями є `users` і `resources`. Таблиця `users` зберігає облікові записи користувачів, їх ролі, групи та статуси. Таблиця `resources` містить опис електронних навчальних ресурсів, зокрема назву, автора, дисципліну, тип ресурсу, категорію, рівень доступу, групу, опис, теги, посилання на файл, кількість переглядів і завантажень. Інші таблиці деталізують взаємодію користувачів з ресурсами.

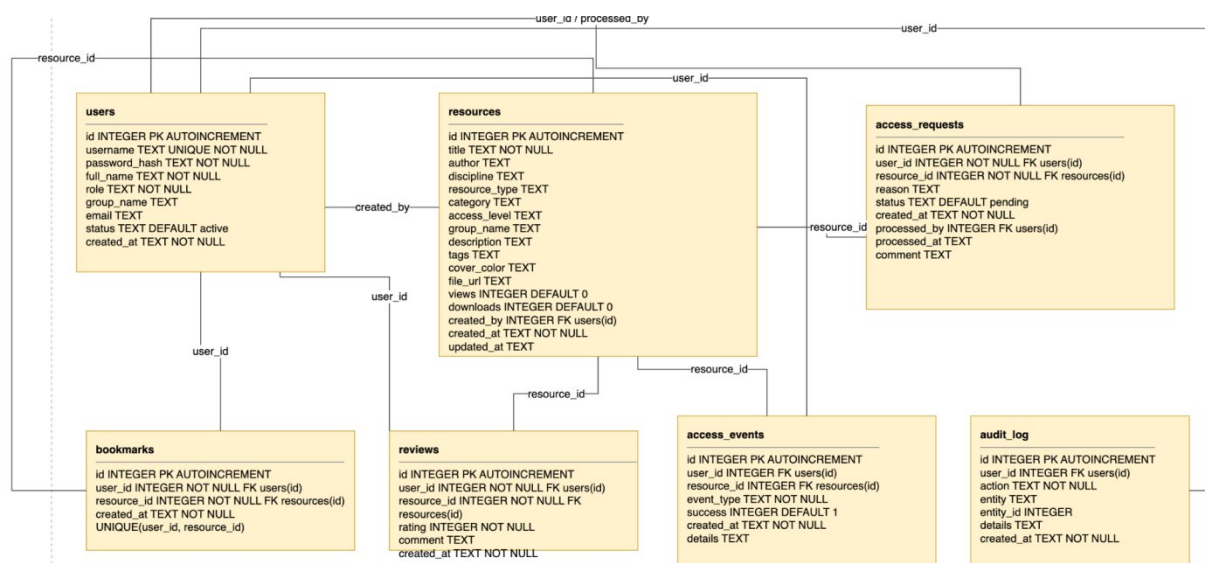


Рис. 3.1. Фізична модель бази даних веб-платформи SmartLibrary

Таблиця `access_requests` використовується для збереження заявок на отримання доступу до ресурсів, які не є відкритими для всіх користувачів. У ній фіксується користувач, ресурс, причина звернення, статус заявки, дата створення, користувач, який опрацював заявку, дата оброблення та коментар модератора. Така структура дозволяє реалізувати контрольований доступ до навчальних матеріалів.

Таблиці `bookmarks` і `reviews` забезпечують персоналізацію роботи користувача з каталогом. Закладки дозволяють сформувати власну бібліотеку навчальних матеріалів, а відгуки й оцінки дають змогу аналізувати якість ресурсів. Таблиця `access_events` фіксує події перегляду, відкриття, завантаження

або відмови в доступі, а `audit_log` використовується для службового журналювання важливих операцій у системі.

Таблиця 3.2

Основні таблиці фізичної моделі бази даних SmartLibrary

Таблиця	Призначення
users	Зберігання облікових записів користувачів, ролей, навчальних груп, електронної пошти та статусу
resources	Зберігання електронних навчальних ресурсів, їх опису, типу, категорії, рівня доступу, тегів і посилань
access_requests	Фіксація заявок користувачів на доступ до обмежених ресурсів та рішень модератора
bookmarks	Збереження персональних закладок користувача на навчальні ресурси
reviews	Збереження оцінок і коментарів користувачів щодо ресурсів
access_events	Ресстрація подій доступу, перегляду, завантаження та відмов
audit_log	Журналювання службових операцій адміністратора, бібліотекаря та користувачів

Для забезпечення цілісності даних у фізичній моделі використовуються зовнішні ключі. Поле `created_by` у таблиці `resources` посилається на `users.id`, що дозволяє визначити автора або модератора, який додав ресурс. Поля `user_id` і `resource_id` у таблицях `access_requests`, `bookmarks`, `reviews` та `access_events` забезпечують зв'язок між користувачами, ресурсами і діями. Наявність унікального обмеження `UNIQUE(user_id, resource_id)` у таблиці `bookmarks` запобігає дублюванню закладок для одного й того самого ресурсу.

Фрагмент SQL-коду створення основних таблиць бази даних наведено нижче.

```
CREATE TABLE users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  username TEXT UNIQUE NOT NULL,
  password_hash TEXT NOT NULL,
  full_name TEXT NOT NULL,
  role TEXT NOT NULL,
  group_name TEXT,
  email TEXT,
  status TEXT DEFAULT 'active',
  created_at TEXT NOT NULL
```

```
);

CREATE TABLE resources (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    author TEXT,
    discipline TEXT,
    resource_type TEXT,
    category TEXT,
    access_level TEXT,
    group_name TEXT,
    description TEXT,
    tags TEXT,
    file_url TEXT,
    views INTEGER DEFAULT 0,
    downloads INTEGER DEFAULT 0,
    created_by INTEGER REFERENCES users(id),
    created_at TEXT NOT NULL,
    updated_at TEXT
);

CREATE TABLE access_requests (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    user_id INTEGER NOT NULL REFERENCES users(id),
    resource_id INTEGER NOT NULL REFERENCES resources(id),
    reason TEXT,
    status TEXT DEFAULT 'pending',
    created_at TEXT NOT NULL,
    processed_by INTEGER REFERENCES users(id),
    processed_at TEXT,
    comment TEXT
);
```

Запропонована фізична модель є достатньою для реалізації основних функцій SmartLibrary: авторизації, управління каталогом, перевірки доступу, створення заявок, ведення закладок і відгуків, фіксації подій та побудови звітів. Модель не перевантажена надлишковими сутностями, однак дозволяє надалі розширити систему модулями рекомендацій, інтеграцією з LMS або зберіганням файлів у хмарному сховищі.

3.3 Алгоритмізація роботи програмних модулів системи

Алгоритмізація програмних модулів SmartLibrary виконана для основних сценаріїв роботи системи. До них належать авторизація користувача, перевірка доступу до електронного ресурсу, створення заявки на доступ, оброблення заявки модератором, формування аналітики та експорт звітів. Побудова алгоритмів дає змогу формалізувати логіку роботи програмного забезпечення й визначити послідовність взаємодії інтерфейсу, сервісів і бази даних.

Першим ключовим процесом є авторизація користувача. Алгоритм авторизації наведено на рис. 3.2. Він починається з відкриття вікна входу SmartLibrary Pro, після чого користувач вводить логін і пароль. Система перевіряє заповнення полів, обчислює SHA-256 хеш пароля та виконує пошук активного користувача в таблиці users. Якщо облікові дані правильні та статус користувача дорівнює active, відкривається головне вікно з правами відповідної ролі. Якщо дані неправильні, система відмовляє у вході та виводить повідомлення про помилку.

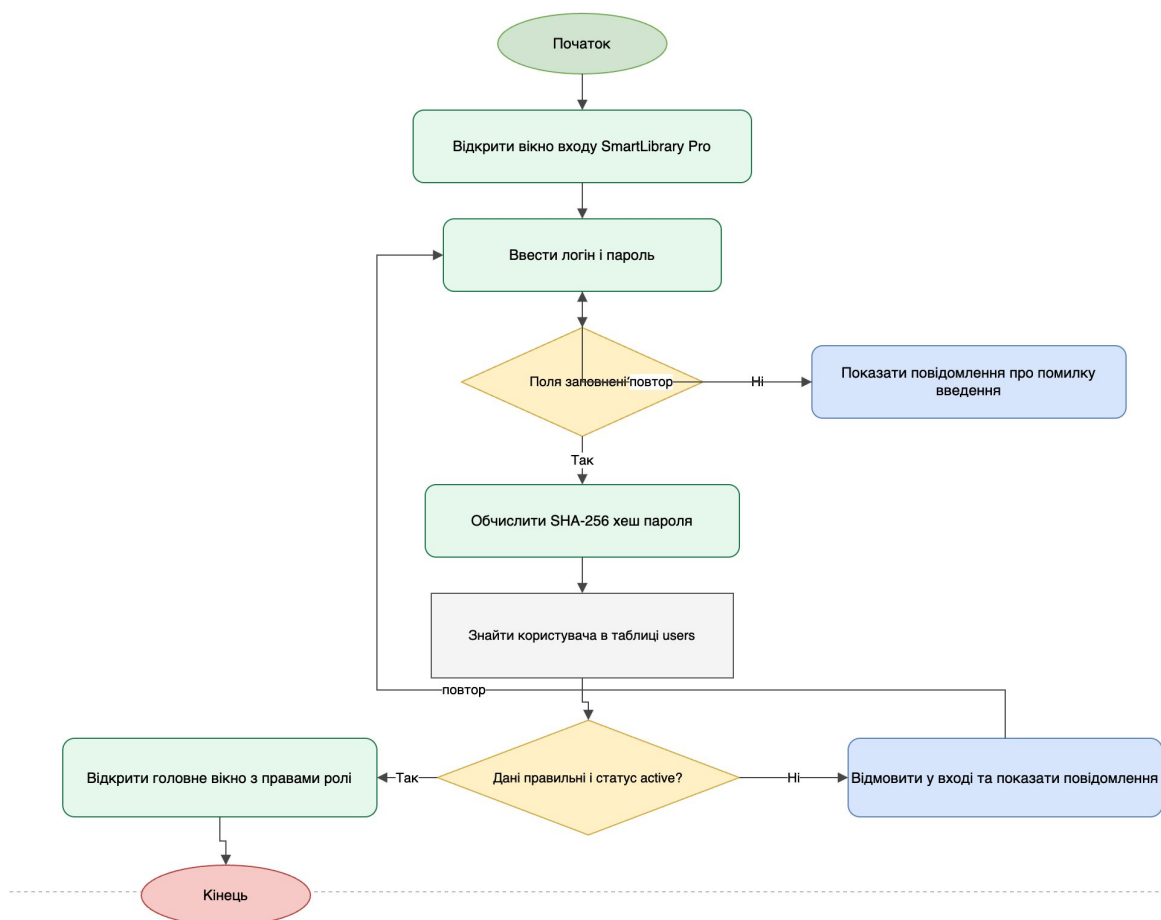


Рис. 3.2. Блок-схема алгоритму авторизації користувача

Таблиця 3.3

Етапи алгоритму авторизації користувача

Етап	Операція	Результат
1	Відкриття форми входу	Користувач отримує доступ до полів логіна й пароля

Продовження таблиці 3.3

2	Перевірка заповнення полів	Виявлення порожніх або некоректних значень
3	Обчислення хешу пароля	Пароль перетворюється у захищене представлення
4	Пошук користувача в таблиці users	Отримання облікового запису за логіном
5	Перевірка статусу та пароля	Підтвердження або відхилення входу

Другий важливий алгоритм стосується відкриття електронного ресурсу та опрацювання обмеженого доступу. Його блок-схему наведено на рис. 3.3. Користувач обирає ресурс у каталозі, після чого система зчитує рівень доступу ресурсу та роль користувача. Якщо ресурс має рівень open, він відкривається одразу, а подія перегляду записується в access_events. Якщо ресурс призначений для певної групи або ролі, система перевіряє відповідність користувача умовам доступу. За наявності дозволу ресурс відкривається, інакше створюється заявка зі статусом pending.

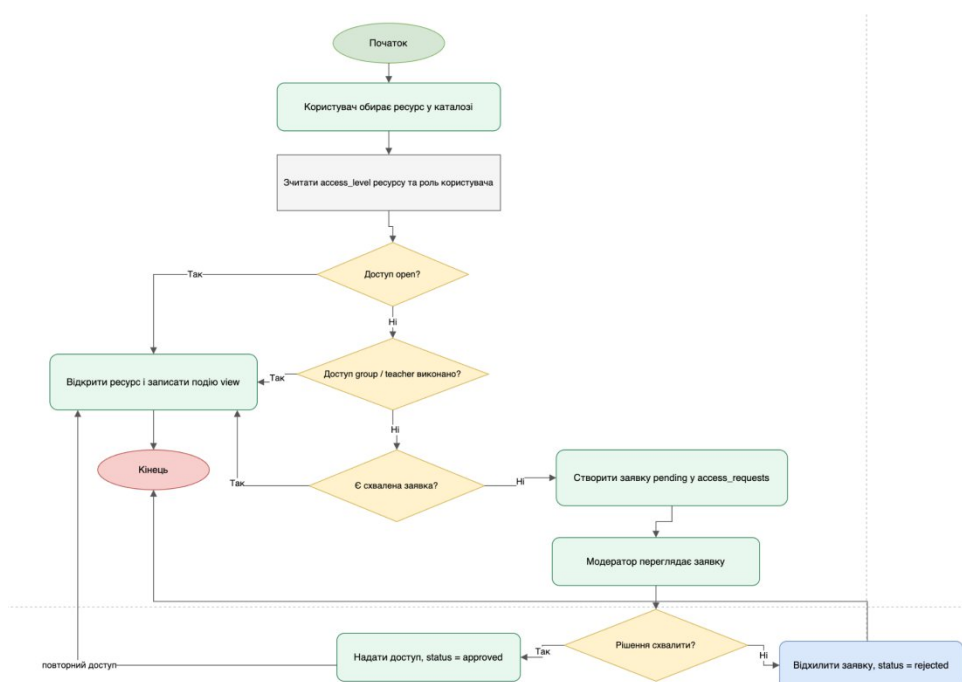


Рис. 3.3. Блок-схема алгоритму перевірки доступу до електронного ресурсу

Таблиця 3.4

Логіка перевірки доступу до електронного навчального ресурсу

Умова	Дія системи	Результат
Ресурс відкритий	Система відкриває ресурс без додаткового погодження	Подія перегляду записується в журнал доступу
Ресурс доступний групі користувача	Перевіряється group_name користувача та ресурсу	Ресурс відкривається для перегляду
Ресурс доступний викладачу або бібліотекарю	Перевіряється роль користувача	Доступ дозволяється відповідно до ролі
Доступ обмежений	Створюється заявка у таблиці access_requests	Модератор отримує заявку для опрацювання
Заявку схвалено	Статус заявки змінюється на approved	Користувач отримує повторний доступ до ресурсу
Заявку відхилено	Статус заявки змінюється на rejected	Користувач отримує відмову з коментарем

Третій алгоритм описує формування аналітики та звітів. Його блок-схему наведено на рис. 3.4. Користувач відкриває розділ звітів, після чого система зчитує дані про ресурси, заявки та події доступу. На основі цих даних обчислюються показники KPI: загальна кількість ресурсів, кількість заявок, кількість переглядів, кількість завантажень, активність користувачів і популярність навчальних матеріалів. Після цього формуються таблиці, графіки та теплова карта активності. За потреби користувач може експортувати звіт у формат HTML або CSV.

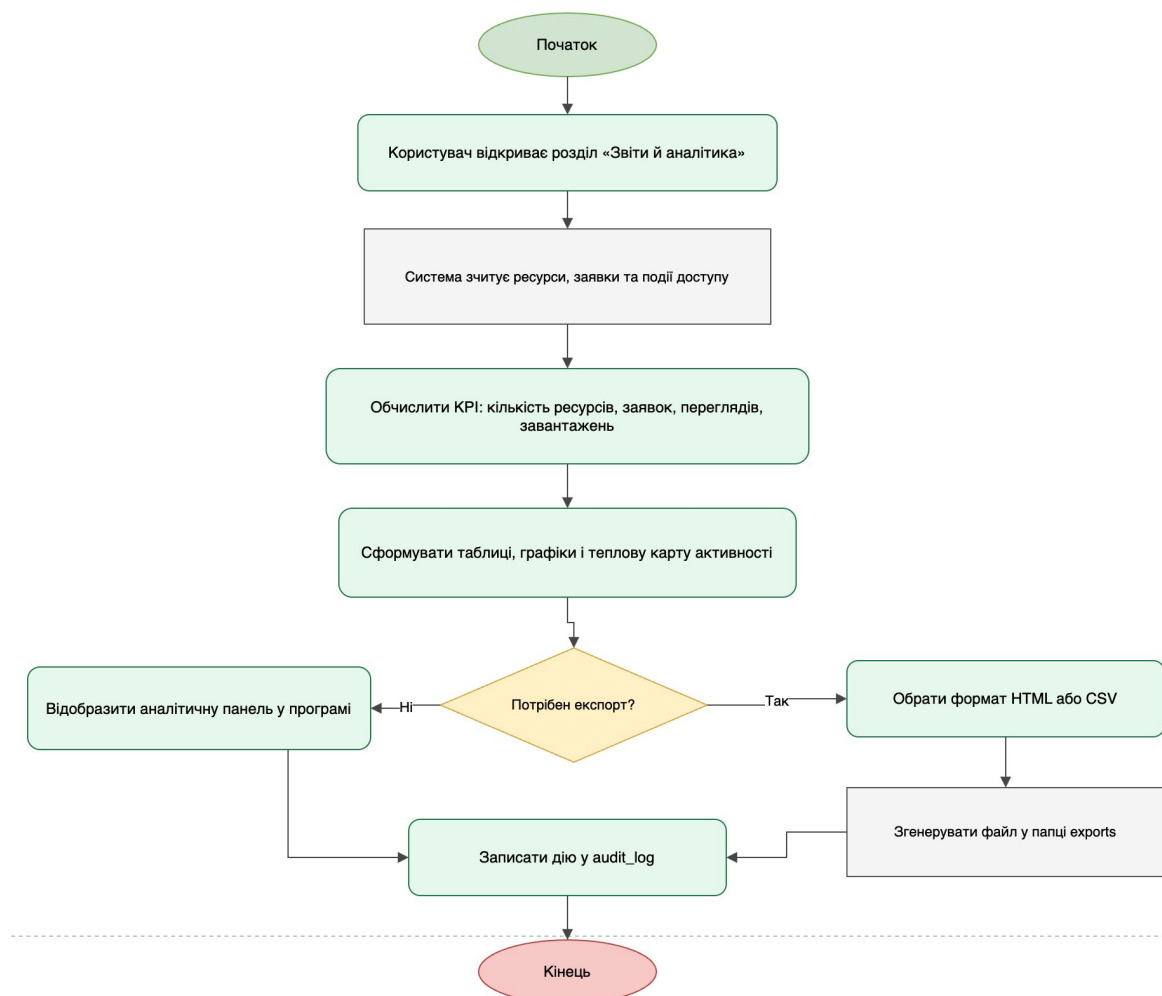


Рис. 3.4. Блок-схема алгоритму формування аналітики та звітів

Таблиця 3.5

Основні показники аналітичного модуля SmartLibrary

Показник	Джерело даних	Призначення
Кількість ресурсів	resources	Оцінювання наповнення електронного каталогу
Кількість заявок	access_requests	Контроль обмежених матеріалів і роботи модератора
Кількість переглядів	access_events, resources.views	Визначення активності використання ресурсів
Кількість завантажень	resources.downloads	Оцінювання попиту на навчальні матеріали
Популярні дисципліни	resources.discipline	Аналіз напрямів найбільшого використання ресурсів

Розроблені алгоритми враховують типові сценарії роботи веб-платформи, а також помилкові або неповні ситуації: незаповнені поля авторизації, неактивний користувач, обмежений доступ до ресурсу, відсутність схваленої заявки або потребу в експорті звіту. Це підвищує стабільність програмного забезпечення та робить логіку роботи системи прозорою для подальшої реалізації й тестування.

3.4 Реалізація основних програмних модулів

Програмне забезпечення SmartLibrary реалізовано за модульним принципом. Кожний модуль відповідає за окрему функціональну частину системи, що спрощує супровід коду та дає змогу надалі розширювати систему без повної зміни архітектури. Основні модулі відповідають проєктним компонентам, визначеним у другому розділі: інтерфейс, авторизація, каталог ресурсів, заявки на доступ, аналітика, експорт звітів, журнал дій та база даних.

Таблиця 3.6

Основні програмні модулі SmartLibrary

Модуль	Призначення
AuthService	Реєстрація, авторизація, перевірка ролі та статусу користувача
ResourceService	Отримання каталогу, пошук, фільтрація, відкриття та оновлення ресурсів
RequestService	Створення, схвалення та відхилення заявок на доступ
AuditService	Запис службових дій користувачів у журнал аудиту
ReportService	Обчислення KPI, формування аналітики та експорт HTML/CSV-звітів
Database	Виконання SQL-запитів, створення таблиць, вибірка й оновлення даних
UI Pages	Відображення форм входу, каталогу, заявок, аналітики, звітів і персональної бібліотеки

Модуль авторизації виконує перевірку облікових даних користувача. Після введення логіна й пароля система обчислює хеш пароля та порівнює його зі значенням, збереженим у таблиці `users`. У разі успіху завантажується роль користувача, що впливає на доступні елементи інтерфейсу. Наприклад, студент може переглядати каталог і подавати заявки, викладач або автор може працювати з власними ресурсами, бібліотекар опрацьовує каталог і заявки, а адміністратор керує користувачами та переглядає аудит.

```
def authenticate(username: str, password: str):
    password_hash = sha256(password.encode('utf-8')).hexdigest()
    row = db.fetchone(
        'SELECT * FROM users WHERE username=? AND password_hash=? AND status=?',
        (username, password_hash, 'active')
    )
    if row:
        db.log(user_id=row['id'], action='login', entity='user', entity_id=row['id'])
        return dict(row)
    return None
```

Модуль каталогу ресурсів відповідає за відображення навчальних матеріалів і пошук за ключовими атрибутами. Користувач може фільтрувати ресурси за назвою, автором, дисципліною, категорією, типом ресурсу, тегами або рівнем доступу. Під час відкриття ресурсу модуль перевіряє, чи має користувач право доступу. Якщо доступ обмежений, формується заявка, яка надалі опрацьовується бібліотекарем або модератором.

```
def can_open_resource(user: dict, resource: dict) -> bool:
    if resource['access_level'] == 'open':
        return True
    if resource['access_level'] == 'group' and user['group_name'] ==
resource['group_name']:
        return True
    if resource['access_level'] == 'teacher' and user['role'] in ('teacher', 'librarian',
'admin'):
        return True
    return db.exists(
        'SELECT id FROM access_requests WHERE user_id=? AND resource_id=? AND status=?',
        (user['id'], resource['id'], 'approved')
    )
```

Модуль заявок забезпечує контроль доступу до ресурсів, які не можуть бути відкриті без погодження. Користувач формує причину звернення, після чого запис створюється в таблиці `access_requests` зі статусом `pending`. Бібліотекар або модератор переглядає заявку, приймає рішення та додає коментар. Після схвалення користувач може повторно відкрити ресурс, а система вже надає доступ на підставі запису із статусом `approved`.

Модуль звітів використовує дані з таблиць `resources`, `access_requests` та `access_events`. Він формує узагальнену статистику щодо кількості ресурсів, популярності дисциплін, активності переглядів, кількості заявок і частки схвалених запитів. Для зручності результати можуть бути переглянуті в інтерфейсі або експортовані у файл.

```
def export_report_csv(path: str, rows: list[dict]) -> None:
    with open(path, 'w', newline='', encoding='utf-8-sig') as file:
        writer = csv.writer(file)
        writer.writerow(['Ресурс', 'Дисципліна', 'Перегляди', 'Завантаження', 'Заявки'])
        for row in rows:
            writer.writerow([
                row['title'], row['discipline'], row['views'], row['downloads'],
                row['request_count']
            ])
```

Важливою частиною реалізації є журналювання. Система фіксує в `audit_log` дії, пов'язані з входом, створенням ресурсу, поданням заявки, опрацюванням заявки, відкриттям ресурсу та експортом звіту. Це дозволяє адміністратору контролювати зміни в системі, а також аналізувати активність користувачів.

3.5 Опис інтерфейсу користувача

Інтерфейс `SmartLibrary` побудований за принципом розділення функцій за окремими сторінками. Після запуску програми користувач проходить авторизацію, після чого система відкриває головне вікно відповідно до ролі. Навігація організована так, щоб користувач міг швидко перейти до каталогу, власної бібліотеки, заявок, аналітики або журналу дій.

Для студента основними елементами інтерфейсу є каталог електронних ресурсів, пошук і фільтрація, перегляд картки ресурсу, додавання закладок, оцінювання матеріалу та подання заявки на обмежений ресурс. Для викладача або автора важливою є можливість перегляду створених матеріалів і контролю активності щодо них. Для бібліотекаря передбачено керування ресурсами, опрацювання заявок і перегляд статистики. Адміністратор отримує доступ до управління користувачами, ролями, журналом аудиту та звітами.

Таблиця 3.7

Основні сторінки інтерфейсу SmartLibrary

Сторінка	Призначення
LoginDialog	Вхід користувача до системи та перевірка облікових даних
RegisterDialog	Створення нового облікового запису користувача
DashboardPage	Відображення узагальненої статистики та швидких переходів
CatalogPage	Перегляд, пошук і фільтрація електронних навчальних ресурсів
ResourceDialog	Перегляд картки ресурсу, опису, тегів і рівня доступу
AccessRequestPage	Створення та опрацювання заявок на доступ
MyLibraryPage	Персональні закладки, історія перегляду та обрані ресурси
AnalyticsPage	Аналітичні показники, графіки, активність і експорт звітів

Візуальна частина інтерфейсу виконана у стриманому навчальному стилі з акцентом на таблиці, картки ресурсів, кнопки дій і аналітичні блоки. Такий підхід відповідає призначенню системи, оскільки користувач повинен швидко знаходити потрібні навчальні матеріали й розуміти статус доступу до них. Інтерфейс не перевантажується зайвими елементами, а основні дії залишаються доступними з головного робочого екрана.

Під час реалізації інтерфейсу враховано рольову модель. Якщо користувач має роль *student*, у нього відсутні функції редагування каталогу й опрацювання заявок. Якщо користувач має роль *librarian* або *admin*, в інтерфейсі відображаються додаткові кнопки керування ресурсами, користувачами, заявками та звітами. Це дозволяє забезпечити логічне розмежування доступу на рівні користувацького інтерфейсу й бізнес-логіки.

3.6 Висновки до третього розділу

У третьому розділі було виконано розроблення та опис програмної реалізації веб-платформи SmartLibrary для управління електронними

навчальними ресурсами. Обґрунтовано вибір технологічних засобів реалізації, зокрема Python, PyQt6, SQLite, SQL, HTML/CSV-експорту та хешування паролів. Обрані засоби забезпечують простоту розгортання, автономність роботи й можливість демонстрації основних функцій системи.

Розроблено фізичну модель бази даних, яка включає таблиці `users`, `resources`, `access_requests`, `bookmarks`, `reviews`, `access_events` та `audit_log`. Описано призначення кожної таблиці, зв'язки між сутностями та фрагменти SQL-коду створення основних структур. Запропонована модель забезпечує цілісне зберігання користувачів, навчальних ресурсів, заявок, персональних дій та аналітичних подій.

Побудовано й описано алгоритми авторизації користувача, перевірки доступу до електронного ресурсу та формування аналітичних звітів. Алгоритми враховують рольову модель, статус користувача, рівень доступу ресурсу, наявність схваленої заявки та потребу в журналюванні операцій. Це забезпечує логічну узгодженість роботи системи та мінімізує ризик некоректного доступу до обмежених навчальних матеріалів.

Описано реалізацію основних програмних модулів SmartLibrary: авторизації, каталогу ресурсів, заявок на доступ, аналітики, експорту звітів, журналу дій і бази даних. Також охарактеризовано інтерфейс користувача, який підтримує різні ролі: студента, викладача, бібліотекаря та адміністратора. Розроблена структура програмного забезпечення є достатньою для подальшого тестування, демонстрації та можливого розвитку системи у повноцінну веб-платформу з серверною архітектурою.

РОЗДІЛ 4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ВЕБ-ПЛАТФОРМИ SMARTLIBRARY

4.1 План тестування програмних модулів та методика оцінювання результатів

Тестування веб-платформи SmartLibrary виконувалося з метою перевірки працездатності основних програмних модулів, правильності взаємодії користувачького інтерфейсу з локальною базою даних, коректності оброблення заявок на доступ, формування аналітичних показників і експорту звітних матеріалів. Особливу увагу приділено сценаріям, які безпосередньо пов'язані з управлінням електронними навчальними ресурсами: пошуку матеріалів у каталозі, відкриттю ресурсів з різними рівнями доступу, додаванню матеріалів до особистої бібліотеки, опрацюванню заявок модератором і перегляду статистики використання.

Програмна реалізація SmartLibrary Pro розглядається як демонстраційний прототип веб-платформи, реалізований у вигляді локального застосунку з графічним інтерфейсом. Такий підхід дозволяє перевірити логіку роботи майбутньої веб-платформи без складного розгортання серверної інфраструктури. Дані користувачів, ресурсів, заявок, закладок, відгуків, подій доступу та журналу дій зберігаються у базі SQLite.

Тестування проводилося за функціональним, інтерфейсним, інтеграційним і приймальним підходами. Функціональне тестування використовувалося для перевірки окремих модулів. Інтерфейсне тестування дало змогу оцінити зручність форм, таблиць, фільтрів і кнопок. Інтеграційне тестування було спрямоване на перевірку зв'язку між інтерфейсом, сервісними модулями та базою даних. Приймальне тестування виконувалося за наскрізними сценаріями роботи студента, бібліотекаря, модератора та адміністратора.

Таблиця 4.1

План тестування програмних модулів SmartLibrary

№	Модуль / функція	Вхідні дані	Очікуваний результат	Критерій приймання
1	Авторизація користувача	Логін, пароль, роль	Відкриття головного вікна з правами ролі	Доступ надається лише активному користувачу
2	Аналітична панель	Дані ресурсів, користувачів, заявок і подій	Відображення KPI, графіків, теплової карти й таблиці топ ресурсів	Показники відповідають даним у БД
3	Каталог ресурсів	Пошуковий запит, категорія, рівень доступу	Фільтрація й відображення ресурсів	Користувач бачить релевантний список матеріалів
4	Перегляд ресурсу	Вибраний навчальний ресурс	Відкриття ресурсу або створення заявки	Система враховує рівень доступу
5	Моя бібліотека	Закладки та історія переглядів	Відображення персональних ресурсів користувача	Закладки зберігаються без дублювання
6	Заявки на доступ	Заявка студента, рішення модератора	Зміна статусу pending / approved / rejected	Рішення зберігається і впливає на доступ
7	Керування ресурсами	Дані нового або редагованого ресурсу	Створення, оновлення або видалення запису	Таблиця resources оновлюється коректно
8	Звіти й аналітика	Події доступу, перегляди, завантаження, оцінки	Формування графіків і звітів HTML / CSV	Звіт містить узгоджені показники
9	Журнал дій	Операції користувачів і адміністраторів	Запис дій до audit_log	Кожна важлива операція фіксується
10	Розгортання системи	ПК користувача, Python, PyQt6, SQLite	Запуск програми та робота з БД	Програма запускається без серверного налаштування

Сформований план тестування охоплює основні сценарії використання системи і дає змогу перевірити не лише окремі функції, а й узгодженість усіх модулів між собою. Для оцінювання результатів використовувалися тестові облікові записи різних ролей, навчальні ресурси з різними рівнями доступу та набір попередньо створених подій використання.

4.2 Тестування веб-платформи SmartLibrary

Тестування веб-платформи SmartLibrary виконувалося за послідовністю основних дій користувача в системі. Спочатку перевірялась аналітична панель адміністратора, після цього каталог електронних навчальних ресурсів, персональна бібліотека студента, модуль заявок на доступ, керування ресурсами, формування звітів і коректність розгортання програмного середовища.

На першому етапі перевірено аналітичну панель системи. Вона відображає поточний стан платформи: кількість електронних ресурсів, активних користувачів, заявок, що очікують рішення, та подій використання. Також у вікні подано діаграму статусів заявок, рейтинг найактивніших дисциплін, активність за останні 14 днів, теплову карту використання та таблицю топ ресурсів за переглядами й завантаженнями. Такий екран дає адміністратору швидке уявлення про завантаженість платформи та проблемні місця у процесі доступу до матеріалів.

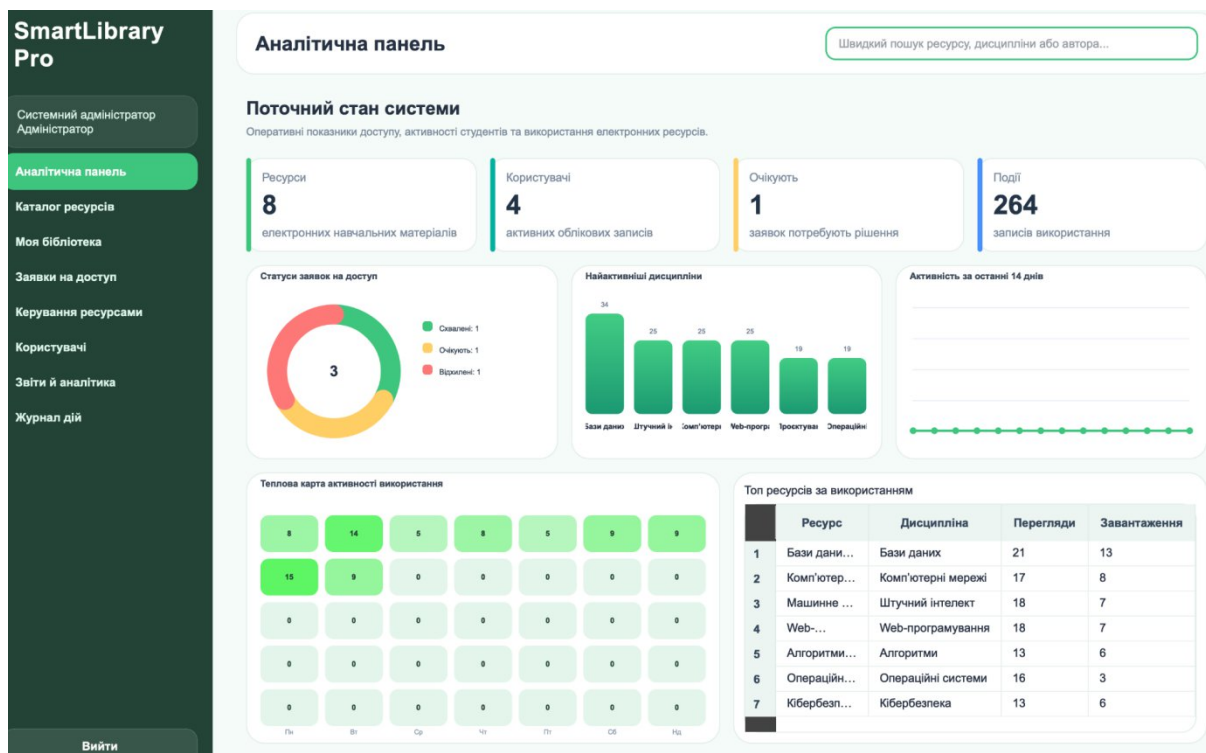


Рис. 4.1. Аналітична панель веб-платформи SmartLibrary

Під час перевірки аналітичної панелі встановлено, що система коректно виводить 8 електронних навчальних ресурсів, 4 активні облікові записи, 1 заявку, яка очікує рішення, та 264 події використання. Дані на графіках і в таблицях узгоджуються між собою, а швидкий пошук у верхній частині інтерфейсу дозволяє оперативно перейти до потрібного ресурсу, дисципліни або автора.

Далі було протестовано каталог електронних навчальних ресурсів. Цей модуль є основним для студента або викладача, оскільки забезпечує пошук, фільтрацію та перегляд навчальних матеріалів. У каталозі користувач може виконати пошук за назвою, дисципліною, автором або тегом, вибрати категорію та рівень доступу. Кожний ресурс представлено у вигляді картки з назвою, дисципліною, типом матеріалу, автором, описом, тегами й доступними діями.

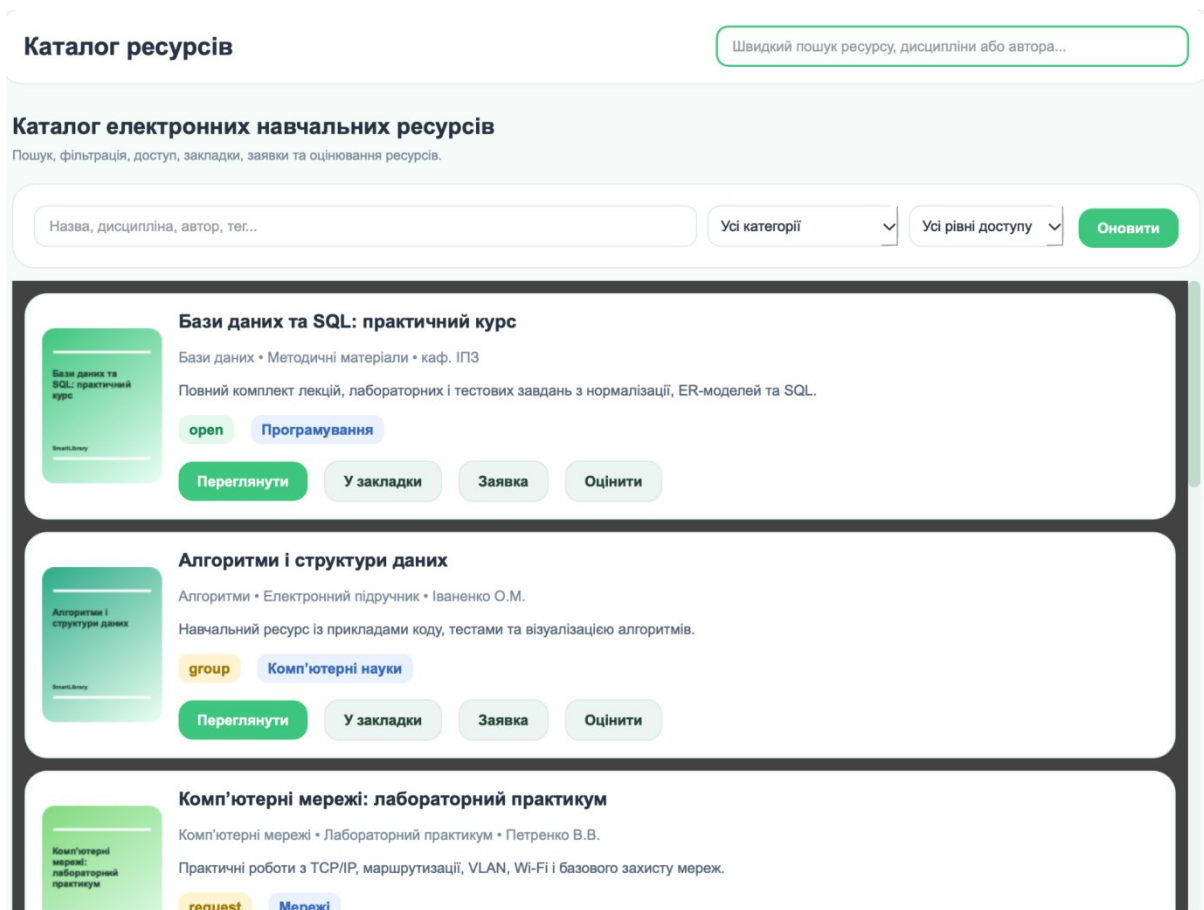


Рис. 4.2. Каталог електронних навчальних ресурсів із картками матеріалів

Перевірка каталогу показала, що система правильно розрізняє ресурси з різними рівнями доступу. Для відкритих матеріалів доступна дія перегляду, для матеріалів групового або обмеженого доступу передбачено створення заявки. Також перевірено додавання ресурсу до закладок і оцінювання навчального матеріалу. Усі дії відображаються в базі даних і можуть бути використані під час формування аналітики.

Додатково було перевірено повноекранне подання каталогу з фільтрами, яке показано на рис. 4.3. У цьому режимі інтерфейс не перевантажений зайвими елементами, а основний акцент зроблено на списку навчальних ресурсів і швидкому пошуку. Це підвищує зручність використання системи під час роботи з великою кількістю електронних матеріалів.

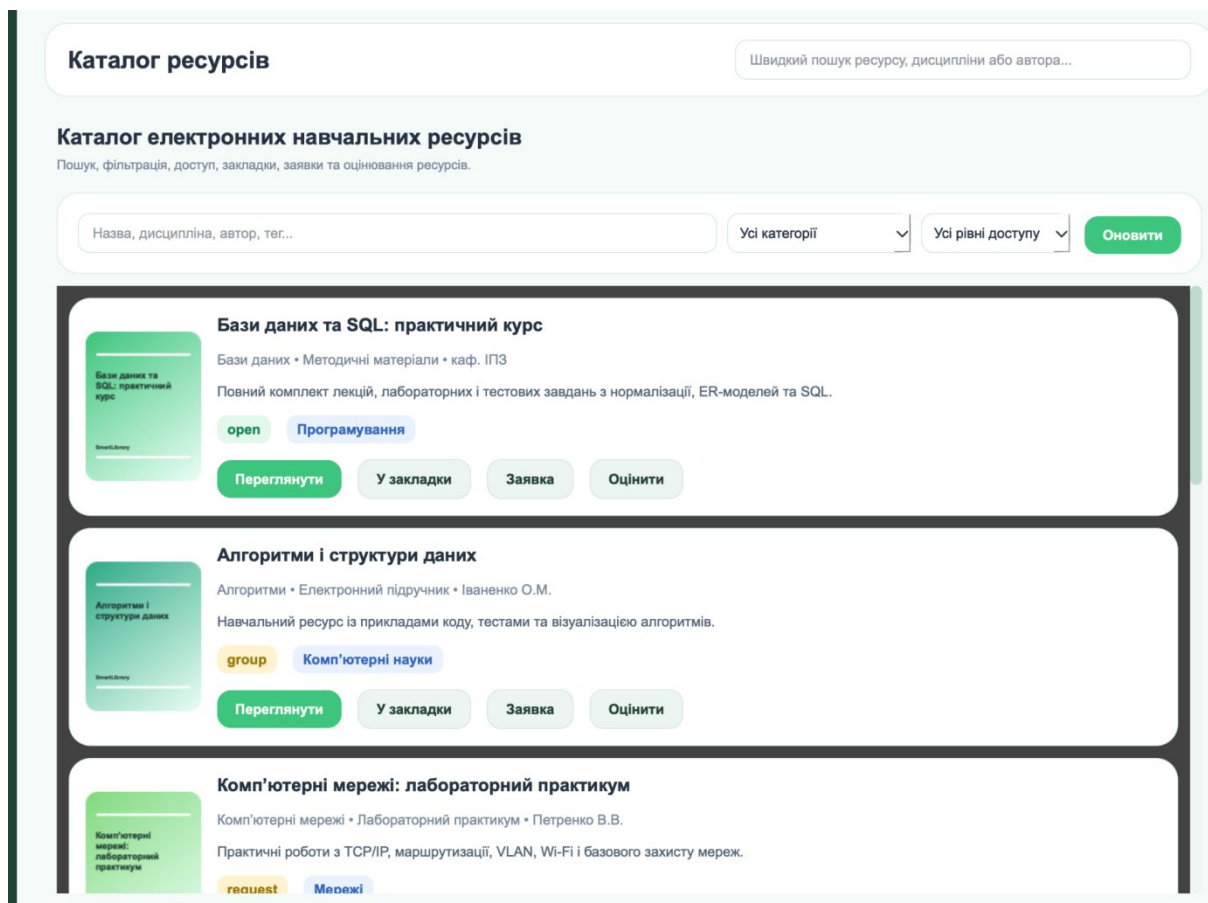


Рис. 4.3. Повноекранний режим перегляду каталогу ресурсів

Під час тестування пошуку використовувалися запити за назвами ресурсів, дисциплінами та фрагментами імен авторів. Система коректно залишала у списку лише ті записи, які відповідали введеним умовам. Така реалізація є важливою для SmartLibrary, оскільки кількість навчальних матеріалів у реальній платформі може постійно зростати.

Наступним етапом перевірено модуль “Моя бібліотека”. Він призначений для персоналізації роботи користувача і містить закладки, історію переглядів та ресурси, з якими студент працює найчастіше. На рис. 4.4 показано вкладку закладок, де відображено доданий навчальний ресурс, дисципліну, тип матеріалу, дату додавання та дію.

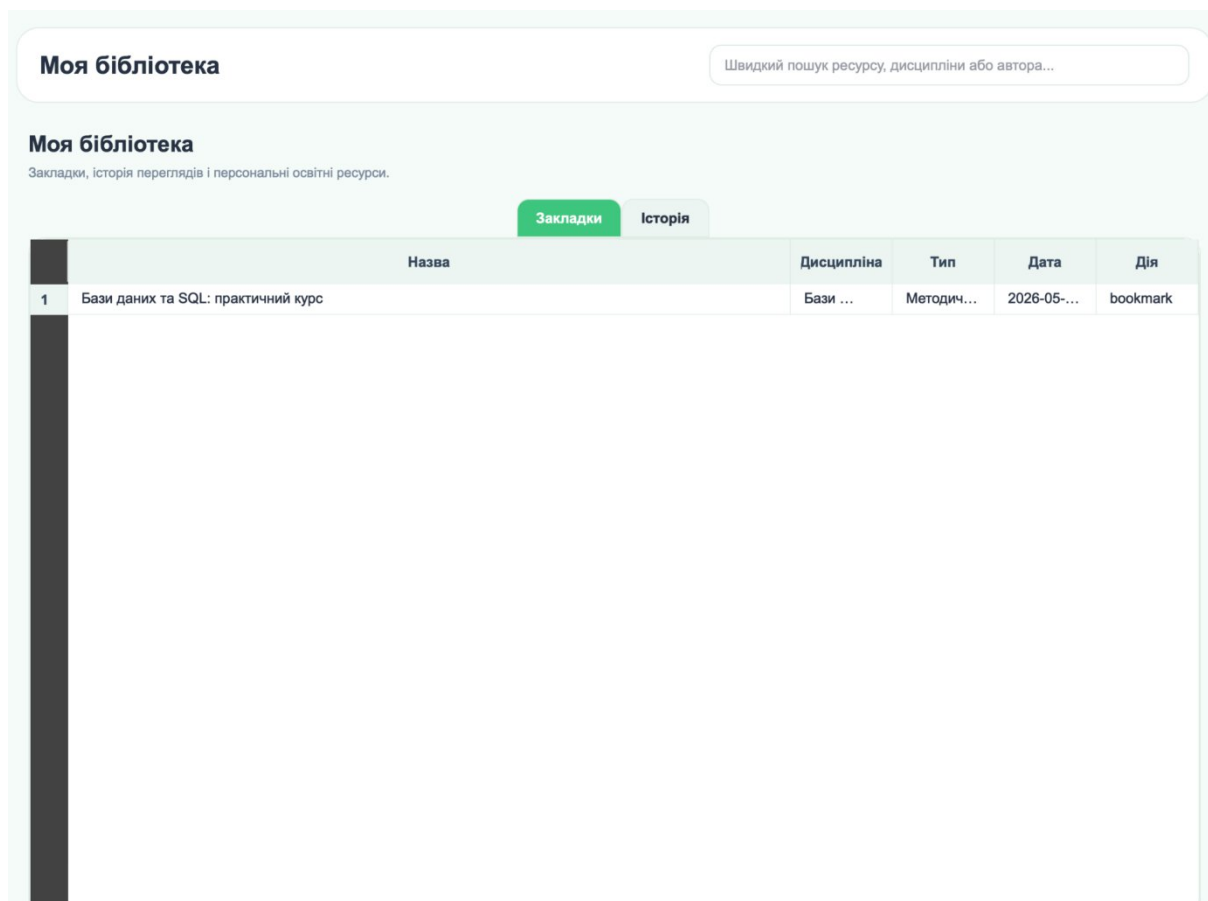


Рис. 4.4. Модуль персональної бібліотеки користувача

Під час тестування підтверджено, що після натискання кнопки “У закладки” ресурс з’являється у персональній бібліотеці. Повторне додавання того самого ресурсу не створює дублікати, оскільки у фізичній моделі даних для таблиці bookmarks передбачено унікальне обмеження для пари `user_id` і `resource_id`. Це забезпечує цілісність персональних списків користувача.

Окремо протестовано модуль заявок на доступ. Він використовується у випадках, коли студент намагається відкрити ресурс із обмеженим або груповим рівнем доступу, але не має необхідних прав. У такій ситуації система створює заявку зі статусом `pending`, а бібліотекар або модератор може її схвалити чи відхилити. Інтерфейс модуля наведено на рис. 4.5.

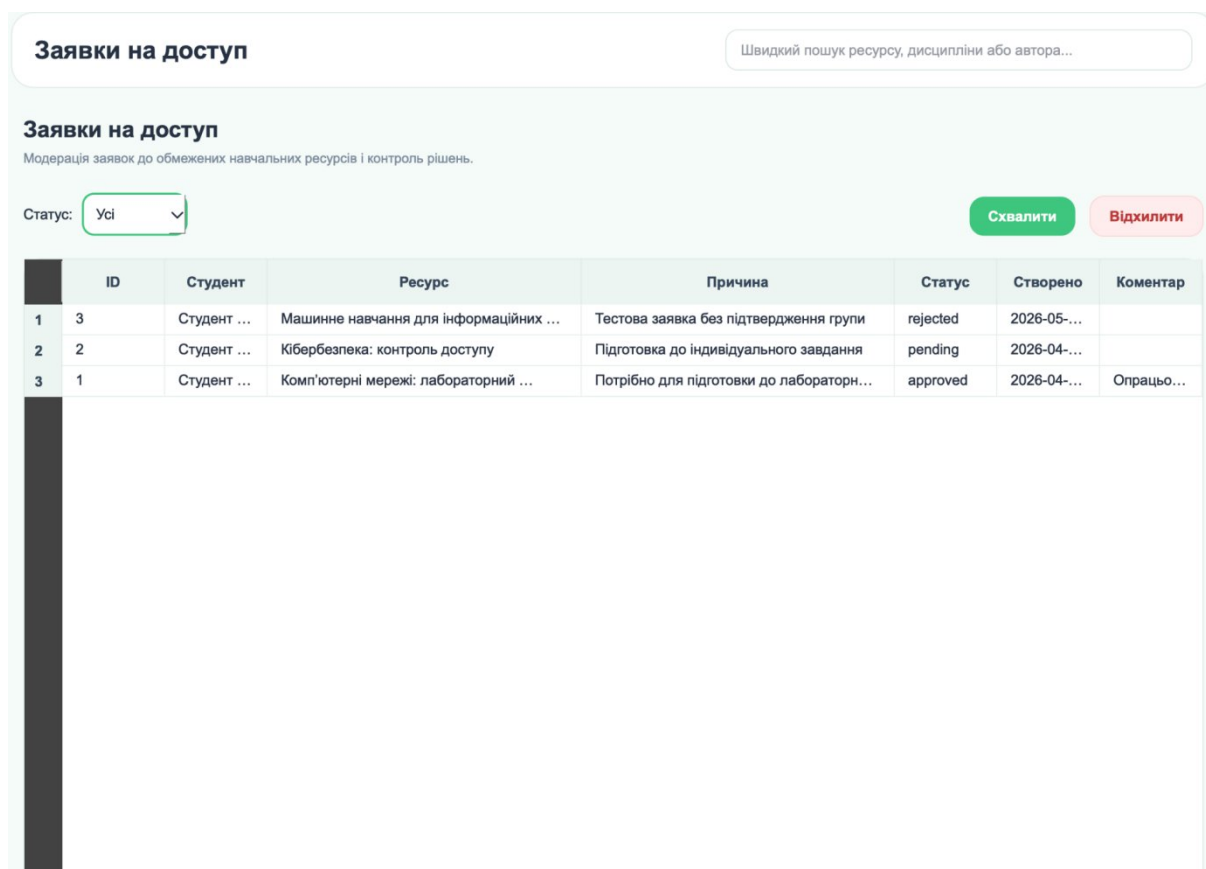


Рис. 4.5. Модуль опрацювання заявок на доступ до навчальних ресурсів

У процесі тестування перевірено фільтрацію заявок за статусом, перегляд причини звернення, зміну статусу на `approved` або `rejected`, а також збереження коментаря модератора. Після схвалення заявки користувач отримує можливість повторно відкрити відповідний ресурс. Якщо заявку відхилено, система зберігає причину відмови і не надає доступ до матеріалу.

Модуль “Керування ресурсами” перевірявся з позиції бібліотекаря або адміністратора. Він дає змогу додавати нові електронні навчальні ресурси, редагувати наявні записи, змінювати рівень доступу, встановлювати групу, тип матеріалу, дисципліну, теги та контролювати кількість переглядів і завантажень. Інтерфейс цього модуля наведено на рис. 4.6.

Керування ресурсами

Швидкий пошук ресурсу, дисципліни або автора...

Керування електронними ресурсами

Додавання, редагування, рівні доступу, теги, обкладинки та контроль використання.

Додати ресурс

Редагувати

Видалити

	ID	Назва	Дисципліна	Тип	Доступ	Група	Перегляди	авантажень
1	8	Проектування інформаційних систем	Проекту...	Навчаль...	open	КН-220026	15	4
2	7	Web-розробка на Python	Web-...	Практич...	group	ІПЗ-2200...	18	7
3	6	Кібербезпека: контроль доступу	Кібербез...	Закрити...	request	КБ-220016	13	6
4	5	Операційні системи: конспект лекцій	Операці...	Конспект	open	КН-220026	16	3
5	4	Машинне навчання для інформаційних систем	Штучний...	Відеолекції	teacher	AI	18	7
6	3	Комп'ютерні мережі: лабораторний практикум	Комп'ют...	Лаборат...	request	КН-220026	17	8
7	2	Алгоритми і структури даних	Алгоритми	Електро...	group	КН-220026	13	6
8	1	Бази даних та SQL: практичний курс	Бази ...	Методич...	open	КН-220026	22	13

Рис. 4.6. Модуль керування електронними навчальними ресурсами

Під час перевірки було встановлено, що таблиця ресурсів коректно відображає ідентифікатор, назву, дисципліну, тип, рівень доступу, групу, кількість переглядів і завантажень. Операції додавання, редагування і видалення виконуються через відповідні кнопки. Після зміни даних таблиця оновлюється без перезапуску програми, а подія може бути зафіксована у журналі дій.

На наступному етапі протестовано модуль “Звіти й аналітика”. Він використовується для оцінювання активності користувачів, популярності навчальних ресурсів, кількості переглядів, завантажень і типів подій. На рис. 4.7 показано аналітичну сторінку з графіками, діаграмою типів подій, динамікою доступів і таблицею ресурсів.

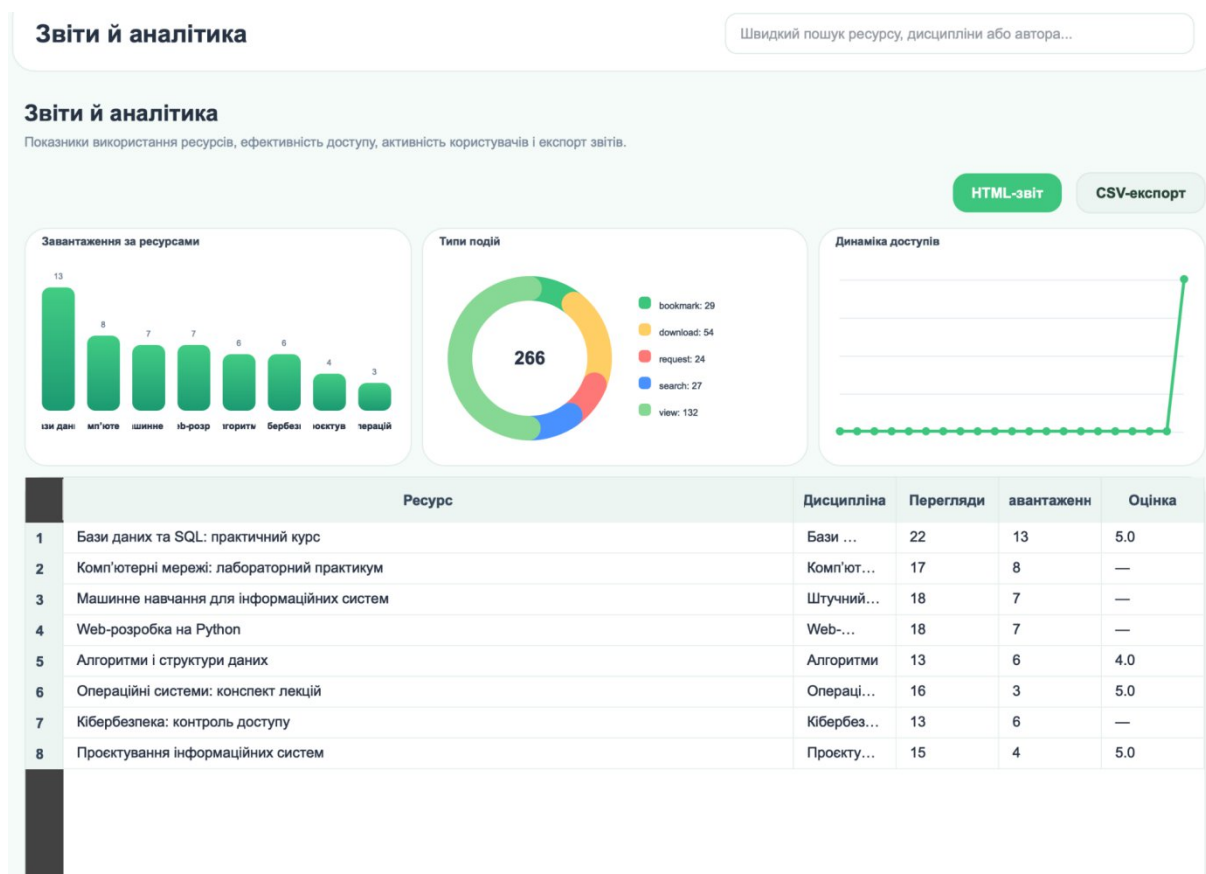


Рис. 4.7. Модуль звітів і аналітики використання ресурсів

Тестування підтвердило, що система формує звіт за даними таблиць `resources`, `access_events`, `reviews` та `audit_log`. На сторінці відображаються завантаження за ресурсами, типи подій, динаміка доступів і таблиця з переглядами, завантаженнями та оцінками. Також перевірено формування HTML-звіту і CSV-експорту, що дозволяє використовувати результати аналітики у зовнішніх документах.

Завершальним етапом тестування було перевірено розгортання програмної системи. На рис. 4.8 показано, що застосунок запускається на персональному комп'ютері користувача під керуванням Windows, macOS або Linux, використовує Python Runtime, бібліотеку PyQt6, локальний файл бази даних `smartlibrary.db` та папку `exports` для збереження HTML і CSV звітів.

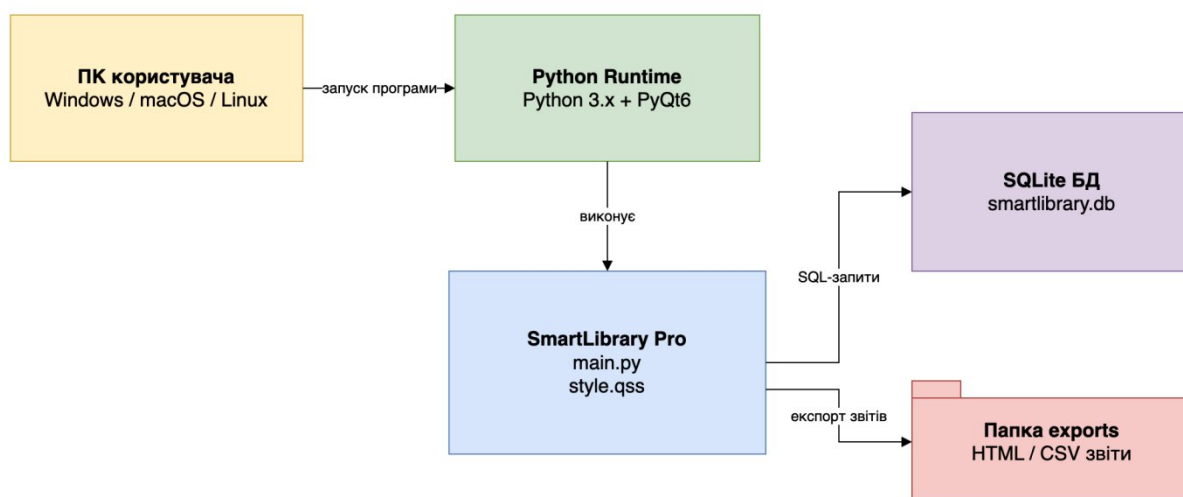


Рис. 4.8. Діаграма розгортання програмного забезпечення SmartLibrary Pro

Перевірка розгортання підтвердила, що програмна реалізація не потребує окремого сервера бази даних або веб-сервера для демонстраційного запуску. Достатньо встановленого Python, необхідних залежностей і файлів проєкту. Це спрощує демонстрацію програмної реалізації та робить її придатною для використання у навчальному середовищі.

Таблиця 4.2

Результати тестування функціональних модулів SmartLibrary

Модуль	Перевірена дія	Очікуваний результат	Результат
Аналітична панель	Завантаження КРІ, графіків і таблиць	Відображення актуальних показників системи	Виконано
Каталог ресурсів	Пошук за назвою, дисципліною й автором	Формування релевантного списку ресурсів	Виконано
Каталог ресурсів	Перегляд відкритого ресурсу	Ресурс відкривається, подія view зберігається	Виконано
Каталог ресурсів	Спроба відкрити ресурс з обмеженим доступом	Створюється заявка на доступ	Виконано
Моя бібліотека	Додавання ресурсу до закладок	Ресурс з'являється у персональному списку	Виконано
Заявки на доступ	Схвалення або відхилення заявки	Статус заявки змінюється	Виконано
Керування ресурсами	Додавання, редагування, видалення матеріалів	Таблиця resources оновлюється	Виконано
Звіти й аналітика	Формування HTML / CSV звіту	Файл створюється у папці exports	Виконано

Отримані результати свідчать, що основні модулі SmartLibrary працюють узгоджено. Дані, створені або змінені в одному модулі, використовуються іншими частинами системи. Наприклад, ресурс, доданий у модулі керування, відображається в каталозі, може бути доданий до закладок, використаний у заявці та потрапляє до аналітичного звіту.

4.3 Оцінювання ефективності роботи системи

Оцінювання ефективності веб-платформи SmartLibrary виконувалося за функціональними, інформаційними та експлуатаційними критеріями. Функціональна ефективність визначалась повнотою реалізації основних сценаріїв роботи з електронними навчальними ресурсами. Інформаційна ефективність оцінювалася за здатністю системи зберігати дані у структурованому вигляді та забезпечувати зв'язок між користувачами, ресурсами, заявками, закладками, відгуками і подіями доступу. Експлуатаційна ефективність визначалася простотою запуску, зрозумілістю інтерфейсу та можливістю локальної роботи без складного серверного налаштування.

До використання такої системи навчальні матеріали часто зберігаються у різних папках, окремих хмарних сховищах або надсилаються студентам через месенджери й електронну пошту. У такому випадку ускладнюється пошук потрібного ресурсу, контроль доступу, аналіз популярності матеріалів і ведення історії використання. SmartLibrary об'єднує ці процеси в одному програмному середовищі.

Таблиця 4.3

Порівняльне оцінювання організації роботи з навчальними ресурсами

Критерій	Без використання SmartLibrary	Після впровадження SmartLibrary
Облік ресурсів	Матеріали розміщені у різних папках або сервісах	Єдиний каталог електронних навчальних ресурсів
Пошук матеріалів	Пошук виконується вручну за назвами файлів	Пошук за назвою, дисципліною, автором і тегами

Продовження таблиці 4.3

Контроль доступу	Доступ часто надається вручну або без централізованого контролю	Рівні доступу open, group, request і teacher
Заявки на доступ	Запити погоджуються через листування	Заявки фіксуються у системі зі статусами
Персональна бібліотека	Користувач самостійно зберігає посилання або файли	Закладки й історія доступні в окремому модулі
Аналітика	Статистика переглядів і завантажень відсутня або неповна	Формуються KPI, графіки, таблиці та звіти
Аудит дій	Складно встановити, хто і коли виконував операції	Події фіксуються у журналі audit_log

Порівняльне оцінювання показує, що SmartLibrary забезпечує більш упорядковану роботу з навчальними матеріалами. Платформа не лише зберігає електронні ресурси, а й дозволяє контролювати доступ, переглядати статистику, працювати із заявками та формувати звіти для адміністратора або бібліотекаря.

Для кількісного оцінювання ефективності використано тестові дані, які відображаються на аналітичній панелі. У системі наявні 8 електронних навчальних ресурсів, 4 активні користувачі, 1 заявка у статусі очікування та понад 260 подій використання. Такі дані достатні для перевірки основних аналітичних функцій і демонструють, що платформа може накопичувати історію взаємодії користувачів з навчальними матеріалами.

Таблиця 4.4

Оцінка відповідності системи основним вимогам

Вимога	Реалізований механізм	Оцінка виконання
Авторизація та ролі	Модуль входу, перевірка користувача, рольові права	Виконано
Каталог ресурсів	Список ресурсів, картки, пошук і фільтрація	Виконано
Обмеження доступу	Рівні доступу open, group, request, teacher	Виконано
Заявки на доступ	Створення, схвалення та відхилення заявок	Виконано
Персоналізація	Закладки, історія, персональна бібліотека	Виконано
Аналітика	KPI, графіки, теплові карти, топ ресурсів	Виконано
Експорт звітів	HTML і CSV файли у папці exports	Виконано

Результати оцінювання підтверджують, що розроблена програмна реалізація відповідає основним функціональним вимогам. Найбільший практичний ефект досягається за рахунок централізованого каталогу ресурсів, автоматизованого контролю доступу та аналітичного модуля, який дає змогу оцінювати активність користувачів і популярність навчальних матеріалів.

4.4 Висновки до четвертого розділу

У четвертому розділі проведено тестування та оцінювання ефективності веб-платформи SmartLibrary для управління електронними навчальними ресурсами. Розроблено план тестування програмних модулів, який охоплює авторизацію, аналітичну панель, каталог ресурсів, персональну бібліотеку, заявки на доступ, керування ресурсами, звіти й аналітику, журнал дій та розгортання системи.

Під час тестування встановлено, що система коректно відображає поточний стан платформи, забезпечує пошук і фільтрацію електронних ресурсів, розрізняє рівні доступу, створює заявки для обмежених матеріалів, дозволяє додавати ресурси до персональної бібліотеки та формує звіти у форматах HTML і CSV. Також підтверджено, що модулі системи взаємодіють із локальною базою даних SQLite і коректно оновлюють інформацію після виконання користувацьких дій.

Оцінювання ефективності показало, що SmartLibrary спрощує організацію електронних навчальних ресурсів, скорочує кількість ручних операцій, підвищує контроль доступу до матеріалів і забезпечує аналітичну підтримку роботи бібліотекаря, викладача та адміністратора. Наявність журналу дій підвищує прозорість роботи системи, а модуль звітності дозволяє аналізувати активність користувачів і популярність ресурсів.

Отже, результати тестування підтвердили працездатність розробленого програмного забезпечення та його відповідність поставленим вимогам.

Програмна реалізація SmartLibrary може бути використана як демонстраційний прототип веб-платформи для централізованого управління електронними навчальними ресурсами, контролю доступу, обліку активності користувачів і формування аналітичної звітності.

ВИСНОВКИ

У кваліфікаційній роботі було розроблено програмне забезпечення веб-платформи SmartLibrary для управління електронними навчальними ресурсами. Розроблена система призначена для централізованого збереження, пошуку, перегляду, модерації та аналізу використання електронних навчальних матеріалів у межах освітнього середовища.

У першому розділі було проведено аналіз предметної області управління електронними навчальними ресурсами. Встановлено, що в умовах цифровізації освіти значна кількість навчальних матеріалів зберігається у різних електронних середовищах, файлових сховищах, хмарних сервісах або окремих курсах, що ускладнює їх систематизацію, пошук, контроль актуальності та керування доступом. Було розглянуто існуючі рішення, зокрема Moodle, Google Classroom, Canvas LMS, Blackboard Learn і DSpace. За результатами аналізу визначено, що доцільною є розробка власної платформи, яка поєднує каталог електронних ресурсів, персональну бібліотеку, заявки на доступ, модерацію, журнал дій та аналітичну звітність.

У другому розділі виконано проєктування інформаційного та програмного забезпечення системи SmartLibrary. Розроблено логічну модель даних, діаграму класів, кооперації класів, компонентну діаграму та діаграму пакетів. У процесі проєктування визначено основні сутності системи: користувач, електронний ресурс, заявка на доступ, закладка, оцінка, подія доступу та запис журналу аудиту. Запропонована структура забезпечує взаємозв'язок між користувачами, ресурсами, рівнями доступу, заявками та подіями використання, що дає змогу реалізувати цілісну інформаційну систему.

У третьому розділі було описано реалізацію програмного забезпечення SmartLibrary Pro. Для реалізації програмного прототипу використано мову Python, бібліотеку PyQt6 для створення графічного інтерфейсу та SQLite для локального збереження даних. Розроблено фізичну модель бази даних, яка включає таблиці users, resources, access_requests, bookmarks, reviews, access_events та audit_log. Також було розроблено алгоритми авторизації

користувача, перевірки доступу до електронного ресурсу та формування аналітичних звітів. Реалізована структура програмного забезпечення забезпечує роботу з каталогом ресурсів, особистою бібліотекою, заявками, керуванням ресурсами, аналітикою та журналом дій.

У четвертому розділі проведено тестування розробленої системи. Перевірено роботу аналітичної панелі, каталогу ресурсів, модуля «Моя бібліотека», заявок на доступ, керування ресурсами, звітів і аналітики, а також розгортання програмного продукту. Результати тестування підтвердили, що система коректно виконує основні функції, передбачені на етапі проєктування. Програмне забезпечення забезпечує пошук і фільтрацію ресурсів, перегляд відкритих матеріалів, подання заявок на обмежені ресурси, збереження закладок, модерацію заявок, облік подій використання та формування звітів.

Практична цінність розробленої системи полягає в тому, що вона дозволяє впорядкувати електронні навчальні ресурси, зменшити дублювання матеріалів, забезпечити контрольований доступ до навчального контенту та отримувати статистику його використання. SmartLibrary може бути використана як програмний прототип для подальшого впровадження в навчальному закладі або як основа для створення повноцінної веб-платформи з серверною частиною, вебінтерфейсом та централізованим сховищем файлів.

Отже, мету кваліфікаційної роботи досягнуто. У результаті виконання роботи було проаналізовано предметну область, визначено вимоги до системи, спроектовано інформаційне та програмне забезпечення, реалізовано програмний прототип SmartLibrary Pro і проведено його тестування. Розроблена система відповідає поставленим завданням і може бути використана для автоматизації процесів управління електронними навчальними ресурсами.

ДОДАТОК Г

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ****Факультет інформаційних технологій****Декларація про академічну доброчесність та використання ШІ****ДЕКЛАРАЦІЯ ЗДОБУВАЧА**

Я, Тимошук Матвій, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проект) на тему «Розробка веб-платформи “SmartLibrary” для управління електронними навчальними ресурсами» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
 - ☐ інструменти генеративного ШІ не використовувалися.
 - ☐ інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL, _____ інші (вказати назву)) були використані для:
 - ☐ перекладу іншомовних джерел;
 - ☐ стилістичного редагування та перевірки граматики;
 - ☐ допомоги у написанні/відлагодженні програмного коду;
 - ☐ інше: _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___» _____
2026 р.

_____ **Матвій ТИМОЩУК**
(підпис)