

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ

Завідувач кафедри

Інформаційних систем і технологій

\_\_\_\_\_ Швиденко М.З

“ \_\_\_ ” \_\_\_\_\_ 2025 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему

«Інформаційна система інтеграції репозиторію наукової бібліотеки  
університету до електронних навчальних курсів навчального порталу»

Спеціальність F3 – «Комп’ютерні науки»

Гарант освітньої програми

Д.е.н., професор \_\_\_\_\_ Руденський Р.А.

Керівник бакалаврської кваліфікаційної роботи

\_\_\_\_\_ к.е.н., доцент \_\_\_\_\_ Мокрієв М.В.

(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав \_\_\_\_\_ Шикита Владислав Віталійович

(підпис) (ПІБ студента)

КИЇВ – 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І  
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інформаційних систем і технологій

\_\_\_\_\_ Швиденко М.З.

“ \_\_\_\_ ” \_\_\_\_\_ 2025 р. .

З А В Д А Н Н Я

на виконання бакалаврської кваліфікаційної роботи студенту

\_\_\_\_\_ Шикита Владислав Віталійович \_\_\_\_\_

(прізвище, ім'я, по батькові)

Спеціальність F3 – «Комп'ютерні науки»

Тема бакалаврської кваліфікаційної роботи

Інформаційна система інтеграції репозиторію наукової бібліотеки  
університету до електронних навчальних курсів навчального порталу

затверджена наказом ректора НУБіП України від 16.12.2024р. №2246с

Термін подання завершеної роботи на кафедру \_\_\_\_\_

(рік, місяць, число)

Вихідні дані до бакалаврської кваліфікаційної роботи: опис програмного  
забезпечення

1. Перелік питань, які потрібно розробити: Аналіз проблемної області.
2. Вибір та обґрунтування засобів.
3. Проектування системи.
4. Висновки

Дата видачі завдання “ \_\_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_ р.

Керівник бакалаврської кваліфікаційної роботи

\_\_\_\_\_ к.е.н., доцент \_\_\_\_\_ Мокрієв М.В. \_\_\_\_\_

(науковий ступінь та вчене звання)

(підпис)

(ПІБ)

Завдання прийняв до виконання \_\_\_\_\_ Шикита Владислав Віталійович

## ЗМІСТ

<b>ВСТУП</b> .....	5
<b>1 Системний аналіз предметної області інформаційної системи</b> .....	7
1.1 Постановка завдання .....	7
1.2 Огляд предметної області.....	10
1.3 Огляд інформаційних джерел та існуючих рішень за темою розробки .....	13
<b>2 Інформаційне забезпечення</b> .....	22
2.1 Логічна модель даних .....	22
2.2 Вибір системи управління інформаційною базою.....	27
2.3 Створення інформаційної бази .....	29
2.4 Прикладне програмне забезпечення.....	30
<b>3 Прикладне програмне забезпечення</b> .....	40
<u><a href="#">3.1 Організаційна структура програмного забезпечення</a></u>	
<u><a href="#">3.2 Вибір інструментарію для створення ППЗ</a></u>	
<u><a href="#">3.3 Алгоритмізація та програмування модулів</a></u>	
<b><u><a href="#">4 Рекомендації щодо впровадження та експлуатації системи</a></u></b>	
<u><a href="#">4.1 Тестування системи</a></u>	
<u><a href="#">4.2 Вимоги до апаратного та програмного забезпечення</a></u>	
<u><a href="#">4.3 Склад інсталяційного пакету</a></u>	
<b><u><a href="#">5 Оцінка ефективності системи, техніко-економічне обґрунтування</a></u></b>	
<u><a href="#">5.1 Автоматизовані процеси та усунення ручних дій</a></u>	
<u><a href="#">5.2 Економія часу</a></u>	
<u><a href="#">5.3 Склад інсталяційного пакету</a></u>	
<u><a href="#">5.4 Техніко-економічна оцінка</a></u>	
<u><a href="#">5.5 Висновок до розділу</a></u>	
<b>ВИСНОВКИ</b>	
<b>Використані джерела</b>	

ІС — інформаційна система

- Сукупність програмних, апаратних і інформаційних компонентів для виконання задач автоматизації (інтеграція даних із DSpace до Moodle).

2. СУБД — система управління базами даних

- Наприклад, MySQL. Відповідає за зберігання, зміну, пошук та обробку даних у структурованій формі.

3. ER — Entity-Relationship

- ER-діаграма — одна з найважливіших моделей: вона показує, які сутності є в системі (наприклад, Repository, Course, Integration) і як вони пов'язані.

4. NF — Normal Form (1NF, 2NF, 3NF)

- Якщо структура БД нормалізована — це гарантія відсутності надлишковості та логічних помилок.

5. DSpace

- Цифровий репозиторій, з якого система забирає навчальні ресурси через API або OAI-PMH. Без розуміння цього компонента — не буде інтеграції.

6. OAI-PMH — Open Archives Initiative Protocol for Metadata Harvesting

- Протокол збору метаданих. Саме через нього система отримує записи у форматі oai\_dc з DSpace.

7. Moodle

- Навчальний портал, у який інтегруються ресурси. Розуміння його структури дозволяє будувати правильну логіку зв'язку.

8. API

- Програмний інтерфейс взаємодії. Завдяки ньому твоя система під'єднується до DSpace та/або Moodle.

9. CRUD — Create, Read, Update, Delete

- Це основні дії, які виконує будь-яка система з БД. У твоєму випадку — додавання інтеграцій, перегляд, оновлення, видалення тощо.

## ВСТУП

**Актуальність.** У сучасному світі стрімкого розвитку цифрових технологій дедалі важливішим стає ефективне використання електронних ресурсів у сфері освіти. Особливу роль у цьому процесі відіграють інституційні репозиторії, які є важливим інструментом зберігання та поширення наукової, освітньої та методичної інформації. Репозиторій, як правило, реалізується на основі системи DSpace — потужного та відкритого програмного забезпечення, що дозволяє створювати архіви з довгостроковим доступом до електронних документів.

Інтеграція такого репозиторію до навчального порталу університету, зокрема до платформи дистанційного навчання Moodle, відкриває широкі можливості для підвищення якості освітнього процесу. Завдяки цій інтеграції студенти та викладачі отримують прямий доступ до наукових публікацій, навчально-методичних матеріалів, дисертацій, звітів та інших ресурсів безпосередньо в рамках електронних курсів.

Таке поєднання є надзвичайно зручним як для викладачів, які можуть автоматично додавати посилання на ресурси з репозиторію у власні курси Moodle, так і для студентів, які отримують централізований доступ до актуальної та перевіреної інформації. Крім того, репозиторій DSpace забезпечує збереження наукової спадщини університету, а Moodle дозволяє інтегрувати ці матеріали в навчальні програми з урахуванням індивідуального темпу навчання. Попит на цифрові освітні послуги постійно зростає, особливо у зв'язку з впливом глобальних викликів, таких як пандемія COVID-19 чи повномасштабна війна. У таких умовах важливо забезпечити безперервність освіти та швидкий доступ до якісного контенту. Інтеграція DSpace з Moodle дозволяє реалізувати ці потреби, адже створює єдину екосистему, де репозиторій стає не лише сховищем, а й активною частиною навчального процесу. Подібно до того, як туристичні агентства допомагають людям зорієнтуватися у виборі подорожі та організують її під ключ, система

інтеграції репозиторію з навчальним середовищем виконує роль освітнього “провідника” — забезпечує ефективну навігацію навчальним контентом і надає доступ до потрібних ресурсів у потрібний момент.

Загалом, така система є не лише технічним рішенням, а стратегічним кроком у напрямі цифрової трансформації освіти, що дозволяє адаптувати навчальний процес до потреб сучасного студента, забезпечити відкритість науки та підвищити академічну доброчесність.

**Мета.** Мета даної інформаційної системи — оптимізувати, автоматизувати та організувати зручний доступ студентам і викладачам до електронних ресурсів наукової бібліотеки університету безпосередньо через навчальні курси у середовищі Moodle. Це значно спростить навчальний процес, зменшить потребу у тривалому пошуку необхідної літератури, дозволить оперативно використовувати актуальні наукові джерела, а також знизить навантаження на бібліотечний персонал завдяки частковій автоматизації запитів і доступу до інформації через API системи DSpace.

**Методи та технології при виконанні даної роботи.** Під час розробки інформаційної системи інтеграції репозиторію наукової бібліотеки до електронних навчальних курсів було застосовано сучасні методи веб-розробки та інтеграції інформаційних систем, а також використано такі програмні засоби:

- MySQL — система управління реляційною базою даних, що використовується для збереження структурованої інформації про ресурси репозиторію, навчальні курси, користувачів та інтеграційні зв’язки. Структура бази відповідає принципам нормалізації до третьої нормальної форми (3НФ).

- Visual Studio Code — кросплатформенне середовище розробки, у якому реалізовано весь код проєкту. Завдяки великій кількості розширень та

підтримці мов програмування воно забезпечує зручне створення шаблонів, написання Python-логіки та роботу з HTML/CSS.

- Python — основна мова програмування, що використовується для реалізації серверної логіки. За її допомогою реалізовано обробку запитів, зв'язок з репозиторієм DSpace (через OAI-PMH), роботу з базою даних та обробку метаданих.

- HTML / CSS — мови розмітки та стилізації, що застосовуються для формування користувацького інтерфейсу, побудови таблиць результатів пошуку, фільтрів, форм тощо.

- JavaScript — використовується для динамічної взаємодії з елементами сторінки (наприклад, фільтрація, асинхронна обробка запитів).

- OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting) — протокол збору метаданих, що використовується для отримання даних із репозиторію DSpace у форматі oai\_dc.

- Формати JSON і XML — застосовуються для обміну структурованими даними з репозиторієм та обробки відповідей сервера.

Розроблена система має веб-орієнтовану архітектуру з можливістю доступу через браузер із персонального комп'ютера. Планується адаптація інтерфейсу для мобільних пристроїв (через адаптивну верстку або PWA).

## **Структура записки.**

# **1 Системний аналіз предметної області інформаційної системи**

## **1.1 Постановка завдання**

**Метою** є розробка інформаційної системи інтеграції інституційного репозиторію наукової бібліотеки університету до електронних навчальних курсів навчального порталу. Система повинна забезпечити автоматизоване отримання, фільтрацію та представлення наукових і навчально-методичних матеріалів у навчальному середовищі Moodle. Вона має оптимізувати процес доступу до ресурсів, надавати викладачам і студентам зручні засоби пошуку,

перегляду та використання матеріалів з репозиторію, сприяючи підвищенню якості навчального процесу та ефективному використанню наукового контенту університету.

Для розробки інформаційної системи потрібно:

- Проаналізувати предметну область – визначити нюанси, специфікації, вимоги.
- Моделювання даних та системи: використовуються діаграми UML (активності, діяльності, прецедентів, пакетів, компонентів )
- Визначення інструментів розробки ІС.
- Розробка бази даних.
- Розробка системи.

Звітні документи (результати роботи інформаційної системи):

- Інформаційне повідомлення — підтвердження успішної інтеграції ресурсу з репозиторію до відповідного курсу Moodle (наприклад, у форматі JSON, XML або лог-файлу).
- Експорт метаданих ресурсу — можливість збереження інформації про публікацію (назва, автор, рік, тип ресурсу, ключові слова) у форматах .pdf або .docx для архівування або підготовки до методичних зборів.
- Аналітичний звіт про використані ресурси — статистика за курсом: скільки матеріалів було інтегровано, якими користувачами, за який період (формати: .csv, .xlsx).
- Звіт про пошукові запити — файл, що фіксує найбільш поширені ключові слова, які використовували викладачі чи студенти для пошуку матеріалів (дає змогу аналізувати навчальні потреби).
- Журнал дій (log-файл) — системний звіт, який відображає операції, що були виконані у системі: імпорт, оновлення, перегляд, інтеграція (формат .txt або .log).

За потреби:

- Звіт для адміністрації/бібліотеки — список курсів, до яких інтегровано ресурси, з указанням кількості та типів публікацій (формат .xlsx).

## **Критерії класифікації системи.**

Розроблювана інформаційна система є автоматизованою — функції пошуку, фільтрації, обробки даних, взаємодії з репозиторієм DSpace, відображення результатів у курсах Moodle тощо виконуються системою. Людиною здійснюється лише ініціація запитів (введення ключових слів для пошуку, вибір дисципліни або матеріалу, авторизація тощо).

Оскільки дана ІС створюється для конкретної предметної галузі — освіти, то за сферою призначення система є навчально-інформаційною — вона виконує функції доступу, відбору та представлення науково-освітньої інформації, що використовується у навчальному процесі.

- За функціональним призначенням: керувальна — допомагає організувати доступ до ресурсів у межах навчального середовища.
- За характером використання інформації: інформаційно-довідкова — система надає доступ до матеріалів без змін їхнього вмісту.
- За масштабом: мережева система — працює в інфраструктурі університету та може забезпечити доступ користувачів через Інтернет.
- За підтримуваними стандартами керування й технологіями комунікації: веб-орієнтована клієнт-серверна архітектура з використанням REST API.

Система має забезпечити:

- автоматизацію обробки запитів користувачів щодо пошуку наукових матеріалів;
- надання відповідної інформації користувачам (викладачам і студентам) відповідно до обраних дисциплін чи ключових слів;
- повний цикл взаємодії з репозиторієм: пошук, відбір, відображення метаданих, надання доступу до повних текстів;
- виведення нових надходжень, актуальних публікацій, наукових звітів, методичних матеріалів;
- забезпечення персоналізованого доступу до збережених матеріалів у межах особистого кабінету Moodle;
- підтримку зворотного зв'язку, наприклад — повідомлення про недоступність ресурсу або подання запиту на консультацію.

## **Вимоги до системи.**

Функціональні вимоги:

- Авторизація користувачів через єдиний акаунт Moodle.
- Автоматичне оновлення бібліотечного контенту у відповідних навчальних курсах.

- Пошук наукових матеріалів за ключовими словами, дисциплінами.
- Рекомендації на основі попередніх запитів користувача.
- Можливість зберігати обрані матеріали у персональному кабінеті.

Нефункціональні (технічні) вимоги:

- Інтеграція з базами даних бібліотечного репозиторію через API (REST).
- Використання пошукових технологій (наприклад, Elasticsearch).
- Підтримка адаптивного дизайну (мобільна/десктопна версія).
- Забезпечення інформаційної безпеки та контролю доступу.
- Сумісність з платформами управління навчальним контентом (Moodle). Сумісність із веб браузерями.

## 1.2 Огляд предметної області

У сучасних закладах вищої освіти зростає попит на швидкий, зручний та структурований доступ до наукових і навчально-методичних матеріалів. Такий попит особливо актуальний в умовах переходу на дистанційне навчання, гібридні форми викладання, а також стрімкого зростання обсягів цифрового контенту. Саме тому інтеграція інституційного репозиторію наукової бібліотеки університету до електронного навчального середовища (зокрема, до системи Moodle) є важливою складовою цифрової трансформації освіти. Репозиторії, як DSpace, слугують централізованим сховищем електронних ресурсів: наукових публікацій, дипломних та курсових робіт, методичних вказівок, звітів і презентацій. DSpace підтримує низку протоколів обміну даними, серед яких ключове місце займає OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting). Саме завдяки цьому протоколу можлива взаємодія з іншими системами — як локальними, так і глобальними — через стандартизований формат OAI\_DC (Dublin Core у форматі OAI-PMH).

Формат OAI\_DC містить базовий набір метаданих, зокрема: назву, автора, дату, формат, опис тощо, представлених у структурованому XML-форматі. Якщо в DSpace активовано механізм harvest (збір), то всі об'єкти репозиторію автоматично стають доступними для сторонніх систем — таких як Google Scholar, OpenAIRE, Europeana, BASE — які можуть здійснювати збір метаданих для подальшого аналізу, індексації чи візуалізації. Це не лише підвищує видимість наукової продукції

університету, а й створює можливості для її гнучкого використання в освітньому процесі. Інтеграція з навчальним порталом Moodle дає змогу безпосередньо в курсах відображати релевантні ресурси, що зберігаються у DSpace. Такий підхід дозволяє:

- автоматизувати пошук, фільтрацію та оновлення матеріалів;
- надати студентам персоналізований доступ до матеріалів відповідно до вивчених дисциплін;
- зменшити дублювання зберігання контенту;
- підвищити якість навчального контенту за рахунок перевірених наукових джерел.

#### **Основні компоненти предметної області:**

- цифровий репозиторій (DSpace);
- система управління навчальними курсами (Moodle);
- протокол OAI-PMH для обміну метаданими;
- навчальні дисципліни, студенти, викладачі;
- метадані (title, creator, date, description тощо);
- персоналізовані кабінети, інформаційні блоки курсів, пошукові механізми.

#### **Ключові функції системи інтеграції:**

- забезпечення безперешкодного доступу до навчальних і наукових матеріалів;
- автоматизоване оновлення контенту в курсах Moodle;
- використання структурованих метаданих (OAI\_DC) для фільтрації та пошуку;
- підтримка відкритих стандартів та інтероперабельність;

- підвищення ефективності навчального процесу та академічної доброчесності.

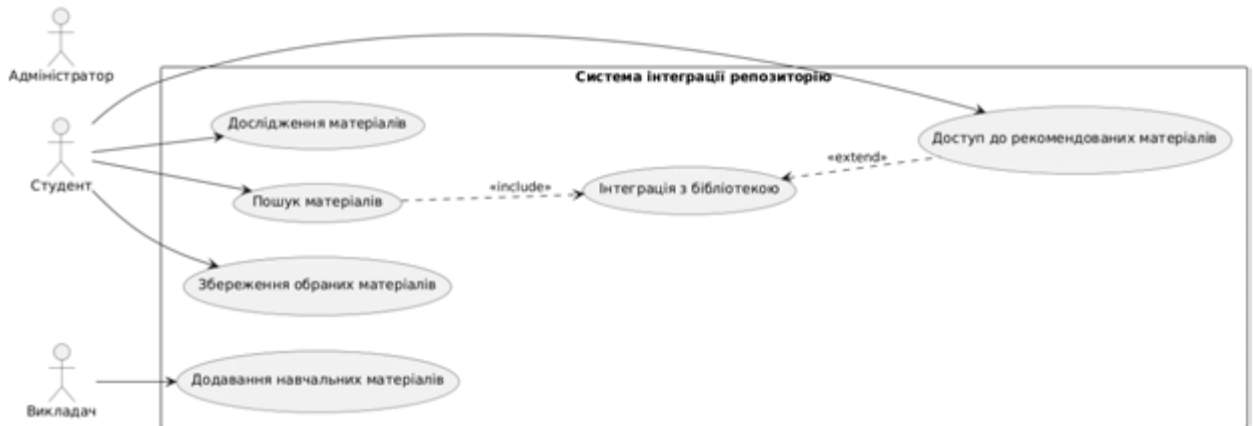
Отже, предметна область охоплює взаємодію між цифровими архівами, освітніми платформами та користувачами (студентами і викладачами), з метою побудови єдиного цифрового освітнього простору на базі відкритих технологій та стандартів.

Основною предметною діяльністю системи, що розробляється, є організація доступу до наукових та навчально-методичних матеріалів університетської бібліотеки через навчальний портал. Система реалізує інтеграцію репозиторію DSpace з платформою Moodle з метою автоматизації процесу пошуку, фільтрації, використання та оновлення контенту в електронних навчальних курсах. Для кращого розуміння предметної області було створено діаграми, які ілюструють ключові компоненти системи та основні дії користувачів у рамках її функціонування.

Діаграма прецедентів (або діаграма варіантів використання) (Use case diagram) в UML — це діаграма, яка відображає взаємодію між зовнішніми користувачами (акторами) та основними функціями (прецедентами) системи. У даному випадку, вона показує відношення між такими користувачами, як студент, викладач, адміністратор, і діями, які вони можуть виконувати: авторизація, пошук матеріалів, доступ до файлів, збереження вибраного, адміністрування доступу тощо. Діаграма прецедентів показує різні варіанти використання та різні типи користувачів системи і часто супроводжується іншими типами діаграм.

Діаграма діяльності (Activity Diagram) візуалізує процес використання та ілюструє потік повідомлень від однієї дії до іншої. Показує цілісну роботу системи:

## Діаграма прецедентів



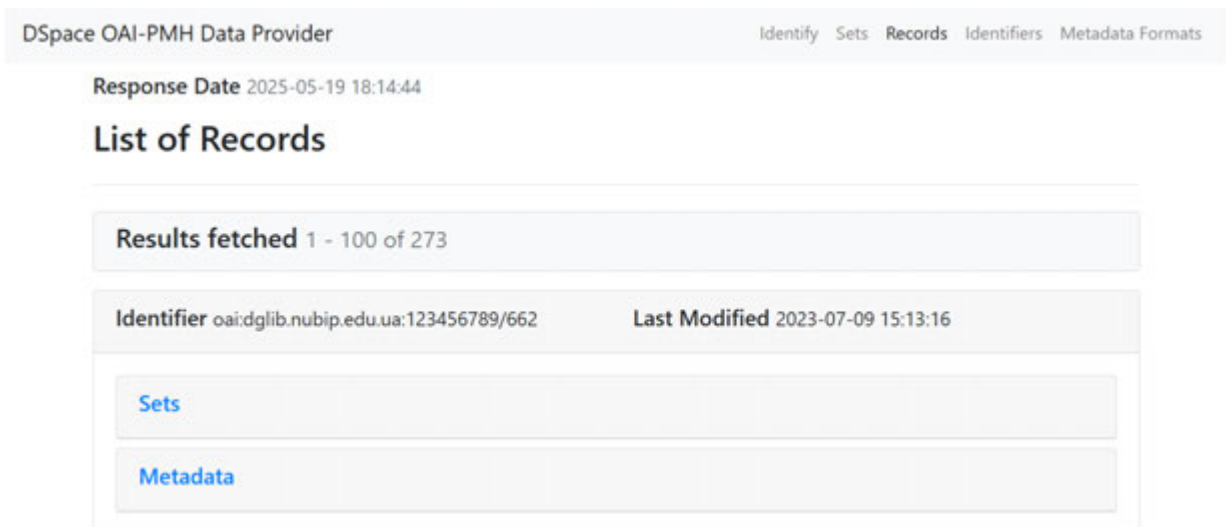
## Діаграма діяльності



### 1.3 Огляд інформаційних джерел та існуючих рішень за темою розробки

Основними інформаційними джерелами у межах дослідження виступають наявні репозиторії наукових бібліотек, зокрема репозиторій НУБіП, а також електронні освітні платформи, що інтегрують навчальний контент із зовнішніми джерелами. Метою дослідження наявних рішень є аналіз базового функціоналу таких систем, визначення вдалих та слабких технічних реалізацій, оцінка зручності для кінцевого користувача (студента, викладача), а також перевірка актуальності проблеми, яку вирішує запропонована інформаційна система.

Зокрема, в якості прикладу було використано функціональне посилання на репозиторій НУБіП, який працює на базі DSpace 8.1:



The screenshot displays the DSpace OAI-PMH Data Provider interface. At the top, it shows the provider name 'DSpace OAI-PMH Data Provider' and navigation links for 'Identify', 'Sets', 'Records', 'Identifiers', and 'Metadata Formats'. Below this, the 'Response Date' is '2025-05-19 18:14:44'. The main section is titled 'List of Records'. A summary bar indicates 'Results fetched 1 - 100 of 273'. Below this, a table header shows 'Identifier oai:dglib.nubip.edu.ua:123456789/662' and 'Last Modified 2023-07-09 15:13:16'. Two buttons, 'Sets' and 'Metadata', are visible below the header.

<b>Identifier</b> oai:dglib.nubip.edu.ua:123456789/6893	<b>Last Modified</b> 2024-12-17 11:52:31
<a href="#">Sets</a>	
<a href="#">Metadata</a>	

<b>Identifier</b> oai:dglib.nubip.edu.ua:123456789/4756	<b>Last Modified</b> 2025-04-14 09:30:42
<a href="#">Sets</a>	
<a href="#">Metadata</a>	

<b>Identifier</b> oai:dglib.nubip.edu.ua:123456789/9654	<b>Last Modified</b> 2023-07-09 15:13:41
<a href="#">Sets</a>	
<a href="#">Metadata</a>	

DSpace OAI-PMH Data Provider	Identify <b>Sets</b> Records Identifiers Metadata Formats
------------------------------	---

**Response Date** 2025-05-19 18:15:50

## List of Sets

**Results fetched** 1 - 100 of 152

Кваліфікаційні роботи [com\_123456789\_1]

[Records](#) [Identifiers](#)

Кваліфікаційні роботи бакалаврів [com\_123456789\_9]

[Records](#) [Identifiers](#)

Кваліфікаційні роботи магістрів [com\_123456789\_8]

[Records](#) [Identifiers](#)

Конференції та матеріали конференцій [com\_123456789\_2]

[Records](#) [Identifiers](#)

Посилання дозволяє отримати перелік наукових матеріалів у вигляді XML-відповіді у форматі oai\_dc (Dublin Core). Таким чином здійснюється доступ до метаданих навчальних ресурсів з конкретного розділу репозиторію:



## Наукова бібліотека



Created by Mira iconic  
from Noun Project

Пошук Підрозділи та колекції За датою За автором За назвою За ключовими словами



Пошук у репозитарії...

🔍 Пошук

### Фільтри

Автор +

Ключові слова +

Дата +

Містить файли +

🔄 Скинути фільтри

### Налаштування

Сортувати за

Дата доступу за спада ▾

Результатів на сторінку

## Результати пошуку

Зараз показуємо 1 - 10 з 277



Документ

### Трагічні голоси

(Редакція газети "Трудова слава", 1993) Гудзенко К.Н.

Мученикам голодної весни 1933 року, їхнім дітям, онукам і правнукам присвячуємо.

Нехай ця гірка трагічна книга після десятиліть безмовства стане історією.

▾ Показати більше



Документ

### Сучасні виклики та завдання бібліотек в умовах військового

стану

(2024)



Документ

### Вплив війни на регіональні та глобальні процеси в Україні та

світі

The screenshot shows a search interface with the following elements:

- Фільтри (Filters):**
  - Автор (Author) +
  - Ключові слова (Keywords) +
  - Дата (Date) +
  - Містить файли (Contains files) +
  - Скинути фільтри (Reset filters)
- Налаштування (Settings):**
  - Сортувати за (Sort by): Дата доступу за спада (Date of access descending)
  - Результатів на сторінку (Results per page): 10
- Document Listings:**
  - Документ (Document):** Основоположник української музичної класики (2022)
  - Документ (Document):** Продовольча безпека України та особливості її розвитку (2022)
  - Документ (Document):** № 21 (20 травня) (1980) Українська орденна Трудового Червоного Прапора сільськогосподарського виробництва. Зміст: Шевченко, І. Вони в наших серцях / І. Шевченко. - С.1 : Фоторепортаж О. Левченко, В. Шевченка. Кл. слова: 35-річчя перемоги у Великій вітчизняній війні. [Показати більше](#)
- Page Navigation:** 1 2 3 4 5 6 7 8 9 10 ... 28 »

### Вигляд головної сторінки

Формат OAI\_DC є частиною протоколу OAI-PMH, який широко використовується для збору метаданих із цифрових архівів. Він забезпечує структурований, стандартизований обмін даними, включаючи такі поля, як: назва (title), автор (creator), дата (date), опис (description) тощо. Це дозволяє зовнішнім системам (зокрема Moodle) здійснювати збір або синхронізацію ресурсів, наприклад, для автоматичного оновлення в навчальних курсах.

- DSpace може віддавати дані зі свого репозитарію у форматі OAI\_DC (Dublin Core у форматі OAI-PMH)
- OAI\_DC — це обов'язковий формат для протоколу OAI-PMH, який використовується в DSpace для обміну метаданими.
- DSpace публікує всі об'єкти репозитарію в цьому форматі автоматично, якщо ввімкнено harvest (збір).

Використовується, щоб зовнішні сервіси могли «збирати» метадані з DSpace (наприклад, Google Scholar, OpenAIRE).

Призначення:

Стандартизований формат для обміну простими метаданими через протокол OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting).

Функції:

- Включає базовий набір полів Dublin Core (title, creator, date, format тощо).
- Має XML-представлення.
- Простий і сумісний з багатьма системами.

Коли використовується:

- При зборі метаданих із цифрових репозитаріїв (наприклад, академічні архіви, institutional repositories).
- У агрегаторах (наприклад, OpenAIRE, BASE, Europeana) для метаданих harvest'у.
- У системах на базі DSpace, EPrints тощо.

Щоб отримати перелік книг (або будь-яких записів) з репозиторію DSpace у форматі OAI\_DC, потрібно скористатися OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting). Це стандартний протокол обміну метаданими через HTTP-запити з XML-відповідями.

Формат базового запиту:

*https://[адреса\_репозиторію]/oai/request?verb=ListRecords&metadataPrefix=oai\_dc*

verb=ListRecords — команда для отримання записів.

metadataPrefix=oai\_dc — указує, що метадані мають повертатися у форматі OAI Dublin Core.

Що реально отримуєш через OAI-PMH:

- Тільки метадані (наприклад, заголовок, автор, дата, тип, мова).
- Не отримуєш повнотекстових файлів (PDF тощо), якщо вони захищені.
- Не можна змінювати або додавати щось через OAI-PMH — це тільки для читання.

Параметри безпеки у DSpace (для адміністраторів):

- Адмін може обмежити:
- які колекції публікуються через OAI;
- чи включати записи з ембарго;
- чи дозволяти доступ до OAI-інтерфейсу взагалі.

Але зазвичай у відкритих репозиторіях ці опції не обмежуються, щоб сприяти індексації (наприклад, OpenAIRE, BASE, Google Scholar).

Таким чином, аналіз наявного функціоналу та його реалізації дозволив чітко окреслити межі можливого вдосконалення, оптимізації та інтеграції бібліотечного репозиторію з електронним навчальним середовищем.

## Статистика для Наукова бібліотека

### Всього відвідувань

	views
Наукова бібліотека	925

### Всього відвідувань за місяць

	views
November 2024	169
December 2024	140
January 2025	153
February 2025	107
March 2025	141
April 2025	127
May 2025	88

## Статистика

Окрім репозиторію НУБіП на базі DSpace, у межах аналізу були розглянуті також інші подібні рішення:

Назва платформ	Тип системи	Протоколи обміну	Підтримка Moodle	Формати метаданих	Коментар
DSpace (НУБіП)	Institutional Repository	REST API, OAI-PMH	часткова (через API)	Dublin Core (OAI_DC)	Відкрите ПЗ, хороша документація
EPrints	Institutional Repository	OAI-PMH	ні (через плагіни)	Dublin Core	Проста система, але застаріла
Zenodo (EU)	Науковий архів	REST, OAI-PMH	ні напряму	DataCite, DC	Має API, але не адаптований під LMS

Open Journal Systems	Платформа для журналів	REST, OAI-PMH	частково	JATS, Dublin Core	Не для навчальних курсів напряму
----------------------	------------------------	---------------	----------	-------------------	----------------------------------

Як видно з таблиці, DSpace є найкращим кандидатом для впровадження в університетське освітнє середовище, зокрема через підтримку стандартів, відкритий вихідний код і широку поширеність.

### **Аналіз інтерфейсу репозиторію:**

Інтерфейс репозиторію НУБіП, реалізованого на DSpace 8.1, є функціональним, але дещо перевантаженим з точки зору простого студента. Не всі елементи пошуку інтуїтивно зрозумілі. Також є складнощі з переглядом пов'язаних документів, якщо користувач не знайомий із структурою колекцій DSpace.

### **Рекомендації:**

- Передбачити адаптацію інтерфейсу через Moodle (замість прямої навігації по репозиторію).
- Використати тільки метадані для відображення потрібного користувачу контенту.
- Спростити навігацію до потрібної колекції/дисципліни за допомогою фільтрів або тегів.

Далі розглядаються **переваги та недоліки** репозиторію НУБіП на базі DSpace з точки зору інтеграції в навчальні системи:

### **Переваги:**

- Підтримка відкритих стандартів (OAI-PMH, OAI\_DC, Dublin Core). Завдяки цим стандартам забезпечується сумісність з іншими освітніми платформами, такими як Moodle, OpenAIRE, Google Scholar тощо.
- REST API та OAI-PMH  
Система дозволяє отримувати метадані й документи як через REST API (у JSON), так і через OAI-PMH (у форматі XML). Це відкриває можливості для автоматизованої взаємодії з репозиторієм.

- Автоматична індексація та публікація  
DSpace автоматично індексує публікації й дозволяє стороннім системам (у т.ч. агрегаторам та освітнім платформам) «збирати» матеріали без ручного втручання.
- Відкрита структура та наявність документації. Система має гарну підтримку з боку спільноти, детальну технічну документацію, що дозволяє розробникам швидко орієнтуватися в API.
- Гнучка модель метаданих. Dublin Core є досить універсальним набором, який можна розширювати або адаптувати під освітні задачі.

### **Недоліки:**

- Відсутність повноцінного інтерфейсу для інтеграції «без програміста». Необхідно вручну будувати API-запити або реалізувати сторонній модуль для Moodle — готового інструменту «з коробки» для інтеграції немає.
- Вимога авторизації на REST API. Для доступу до деяких функцій (наприклад, повного переліку публікацій через core/items) потрібна автентифікація, що ускладнює безпосереднє використання API в зовнішніх системах без налаштування сесій.
- XML-відповіді OAI-PMH є громіздкими для обробки. Для інтеграції в сучасні фронтенд-системи (React, Vue тощо) зручніше використовувати JSON, тоді як OAI-PMH — це суто XML-протокол.
- Відсутність категоризації матеріалів за навчальними дисциплінами «із коробки». У структурі репозиторію матеріали організовано за колекціями, а не напряму за навчальними курсами, що потребує додаткової логіки при інтеграції.
- Інтерфейс DSpace не завжди зручний для кінцевих користувачів (особливо студентів). Навігація по розділах може бути складною без адаптації під структуру навчального порталу.

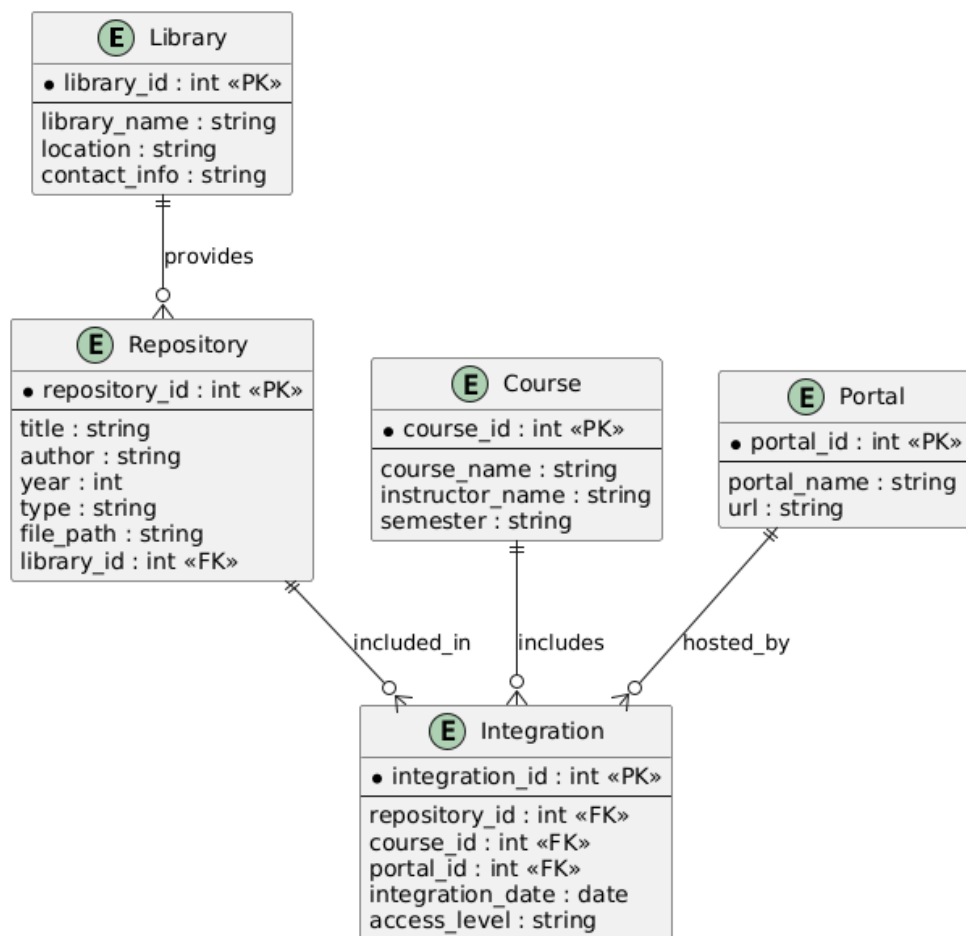
## **2 Інформаційне забезпечення**

Інформаційне забезпечення - сукупність усіх документів і даних, що зберігаються й циркулюють в автоматизованій інформаційній системі, призначена для інформаційного обслуговування користувачів інформації та підтримання роботи персоналу.

## 2.1 Логічна модель даних

Логічна модель даної системи – встановлює основні елементи даних та показує зв'язки між сутностями. Дана модель розроблена з дотриманням принципів Нормальних форм та зведена до 3 нормальної форми (3NF).

У даній системі, яка розробляється, одним з компонентів інформаційного забезпечення є ER-діаграма – як представлення логічної моделі даних, для реляційної бази даних. На Рис.1. Дана схема описує головні атрибути та зв'язки між ними, та дозволяє чітко уявити всі взаємодії які відбуваються в базі даних.



ER-модель.

Представлена структура сховища даних: ER-модель, яка відображає логічну структуру бази даних “library\_integration”, складається з основних сутностей, що забезпечують зберігання, взаємозв'язки та інтеграцію даних між науковою бібліотекою, репозиторієм, навчальними курсами та порталом.

Опис основних сутностей та їх атрибутів:

- Library — зберігає інформацію про наукову бібліотеку: library\_id, library\_name, location, contact\_info.
- Repository — містить дані про навчальні та наукові матеріали: repository\_id, title, author, year, type, file\_path, library\_id (FK).
- Course — містить інформацію про навчальні курси: course\_id, course\_name, instructor\_name, semester.
- Portal — описує навчальний портал (наприклад, Moodle): portal\_id, portal\_name, url.
- Integration — зв'язує репозиторій, курс і портал між собою: integration\_id, repository\_id (FK), course\_id (FK), portal\_id (FK), integration\_date, access\_level.

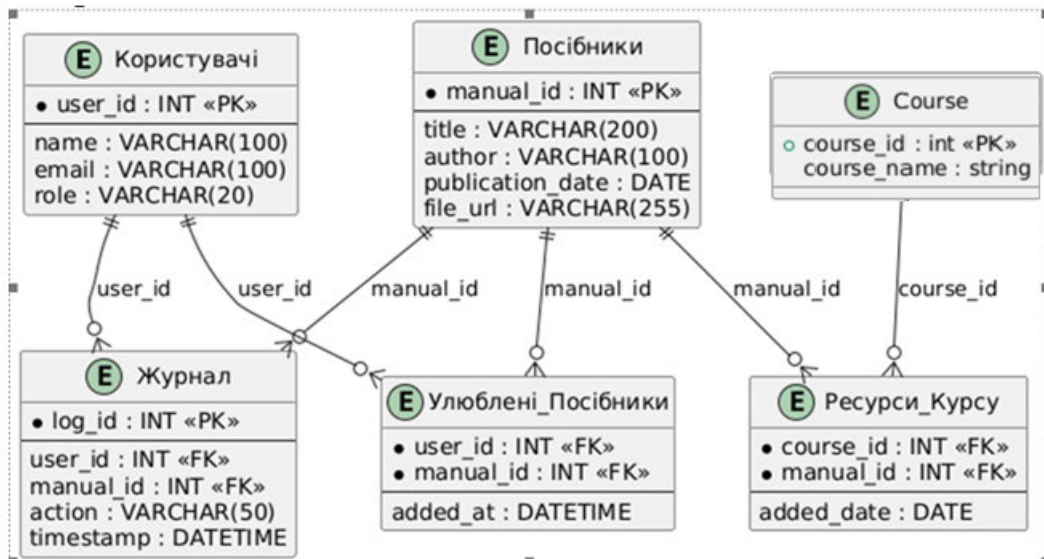
#### Основні типи зв'язків:

- Library —< Repository (один-до-багатьох): одна бібліотека надає багато матеріалів.
- Repository (Manuals)—< Integration (один-до-багатьох): один матеріал може бути пов'язаний з багатьма курсами.
- Course —< Integration (один-до-багатьох): один курс може містити посилання на багато матеріалів.
- Portal —< Integration (один-до-багатьох): один портал може включати багато інтегрованих курсів.

Типові типи зв'язків між таблицями — «один-до-багатьох» і «багато-до-одного». Така структура дозволяє масштабування системи: можна додавати нові курси, портали, типи доступу, типи ресурсів тощо без порушення цілісності бази.

Логічна модель реалізована відповідно до третьої нормальної форми (3NF), що забезпечує мінімізацію надлишковості та покращує підтримку даних у цілісному й послідовному стані.

Ця модель є основою для реалізації реляційної бази даних у таких СУБД як PostgreSQL або MySQL.



Реалізація структури бази даних на фізичному рівні.

SQL:

-- Таблиця користувачів

```

CREATE TABLE Usrs (
    user_id CHAR PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    role VARCHAR(20) CHECK (role IN ('student', 'teacher', 'admin'))
);
  
```

-- Таблиця курсів (зв'язок з Moodle через course\_id)

```

CREATE TABLE Courss (
    course_id CHAR PRIMARY KEY
);
  
```

-- Таблиця посібників

```

CREATE TABLE Manls (
    manual_id CHAR PRIMARY KEY,
    title VARCHAR(200) NOT NULL,
    author VARCHAR(100),
    publication_date DATE,
    file_url VARCHAR(255)
);
  
```

-- Таблиця зв'язку курсів з посібниками (багато до багатьох)

```

CREATE TABLE CoursRes (
  
```

```

course_id CHAR,
manual_id CHAR,
added_date DATE DEFAULT CURRENT_DATE,
PRIMARY KEY (course_id, manual_id),
FOREIGN KEY (course_id) REFERENCES Courses(course_id) ON DELETE
CASCADE,
FOREIGN KEY (manual_id) REFERENCES Manuals(manual_id) ON
DELETE CASCADE
);

```

-- Таблиця логів дій користувачів з посібниками

```

CREATE TABLE Lgs (
log_id CHAR PRIMARY KEY,
user_id CHAR,
manual_id CHAR,
action VARCHAR(50),
timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE SET
NULL,
FOREIGN KEY (manual_id) REFERENCES Manuals(manual_id) ON
DELETE SET NULL
);

```

-- Таблиця улюблених посібників (багато до багатьох)

```

CREATE TABLE FavManuls (
user_id CHAR,
manual_id CHAR,
added_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (user_id, manual_id),
FOREIGN KEY (user_id) REFERENCES Users(user_id) ON DELETE
CASCADE,
FOREIGN KEY (manual_id) REFERENCES Manuals(manual_id) ON
DELETE CASCADE
);

```

Аналіз структури бази даних та нормалізація

При проектуванні логічної моделі даних було здійснено аналіз на відповідність основним нормальним формам.

Згідно з визначенням 1NF (перша нормальна форма):

- усі атрибути є атомарними (неподільними);
- у кожному стовпці зберігається лише одне значення;
- кожна таблиця має чітко визначений первинний ключ (наприклад, repository\_id, course\_id, portal\_id тощо).

Згідно з 2NF (друга нормальна форма):

- виконується 1NF;
  - усі неключові атрибути повністю залежать від усього первинного ключа.
- У випадку таблиці Integration, яка має складні зовнішні ключі (repository\_id, course\_id, portal\_id), всі її неключові атрибути (integration\_date, access\_level) логічно залежать від повного набору зв'язків — тобто конкретної інтеграції ресурсу до курсу на порталі.

Згідно з 3NF (третя нормальна форма):

- виконується 2NF;
- усі неключові атрибути не залежать від інших неключових атрибутів, а тільки від первинного ключа таблиці.

Усі атрибути в таблицях Library, Repository (Manuals), Course, Portal, Integration задовольняють цю умову.

Отже, логічна модель бази даних системи інтеграції є повністю нормалізованою та відповідає вимогам першої, другої та третьої нормальної форми (1NF, 2NF, 3NF). Це гарантує відсутність аномалій при оновленні, зменшення надлишковості та високу цілісність даних при виконанні запитів у системі.

## 2.2 Вибір системи управління інформаційною базою

На даному етапі розробки необхідно обґрунтувати вибір системи управління базами даних (СУБД), яка слугуватиме основою для

зберігання структурованої інформації про навчальні матеріали, курси, портали та взаємозв'язки між ними. СУБД забезпечує організацію, доступність, захищеність і масштабованість даних у системі.

Для реалізації даної інформаційної системи — інтеграції репозиторію бібліотеки університету до навчального порталу — було обрано СУБД MySQL. Вона є реляційною системою з клієнт-серверною архітектурою, що ідеально підходить для реалізації логічної моделі з чіткими зв'язками між сутностями: бібліотека, репозиторій, навчальний курс, освітній портал, модуль інтеграції.

У такій системі дані зберігаються у вигляді взаємопов'язаних таблиць з чітко визначеними полями, що дозволяє ефективно виконувати запити, фільтрацію, аналіз і доступ до навчальних ресурсів. Крім того, MySQL інтегрується з Moodle, підтримує обробку великих обсягів метаданих (наприклад, Dublin Core з DSpace), і дозволяє гнучко реалізувати REST-запити до сторонніх систем.

Переваги використання MySQL у даній системі:

- Масштабованість. MySQL підтримує обробку великих обсягів даних, що важливо для університетського репозиторію з десятками тисяч наукових записів.
- Переносність. Працює на різних ОС (Linux, Windows, macOS), що дозволяє розгорнути систему в локальному середовищі університету чи на хмарних серверах.
- Сумісність. MySQL сумісна з Moodle, DSpace та іншими платформами, що підтримують стандартні протоколи доступу (JDBC, ODBC, REST API).
- Безпека. Система контролю доступу, автентифікація, можливість резервного копіювання, шифрування трафіку — все це дозволяє безпечно зберігати освітні дані.
- Швидкодія. Висока продуктивність при обробці великої кількості запитів та записів, особливо у веб-орієнтованих системах.
- Інтеграційність. Доступ до MySQL можна здійснювати з різних мов програмування — Python, Java, PHP, що дає гнучкість при реалізації API-інтеграції.

- Простота в адмініструванні. СУБД легко налаштовується, має хорошу документацію і велику спільноту підтримки.
- Відкритий код. MySQL — безкоштовна та відкрита система, що відповідає вимогам державного закладу освіти й дозволяє заощадити кошти.

Таким чином, MySQL є оптимальним вибором для побудови інформаційної бази даних у системі, де передбачено інтеграцію між DSpace, Moodle та користувачами навчального порталу.

Для реалізації інформаційної системи інтеграції репозиторію наукової бібліотеки до навчального порталу було обрано СУБД MySQL. Ця система має клієнт-серверну архітектуру, підтримує реляційну модель зберігання даних та забезпечує ефективну роботу з таблицями, пов'язаними між собою. MySQL дозволяє зручно об'єднувати дані з кількох таблиць за допомогою SQL-запитів, що є важливим для побудови гнучкої структури доступу до навчальних матеріалів.

Обрана СУБД підтримує інтеграцію з мовами програмування (наприклад, PHP, Python), має засоби резервного копіювання, контролю доступу, реплікації та масштабування, що важливо у випадку подальшого розширення системи. Завдяки своїй поширеності, відкритому коду та сумісності з Moodle, MySQL є оптимальним рішенням для створення освітньої системи з надійним зберіганням і обробкою метаданих та документів з репозиторію DSpace.

### 2.3 Створення інформаційної бази

Приклад створення таблиці Repository (тобто — науковий матеріал у репозиторії), адаптований до твоєї теми, виглядатиме так:

SQL-приклад (створення таблиці Repository):

```
CREATE TABLE Repository (
repository_id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(255) NOT NULL,
author VARCHAR(255) NOT NULL,
year INT,
type VARCHAR(100),
file_path VARCHAR(255),
library_id INT,
FOREIGN KEY (library_id) REFERENCES Library(library_id)
);
```

Пояснення до полів:

repository\_id — унікальний ідентифікатор наукової публікації;

title — назва документа;

author — автор або список авторів;

year — рік публікації;

type — тип ресурсу (стаття, підручник, дисертація тощо);

file\_path — шлях до файлу або зовнішнього посилання;

library\_id — зовнішній ключ до таблиці Library, яка вказує на приналежність до конкретної бібліотеки (наприклад, НУБіП).

Ще приклад: створення таблиці Course (навчальний курс):

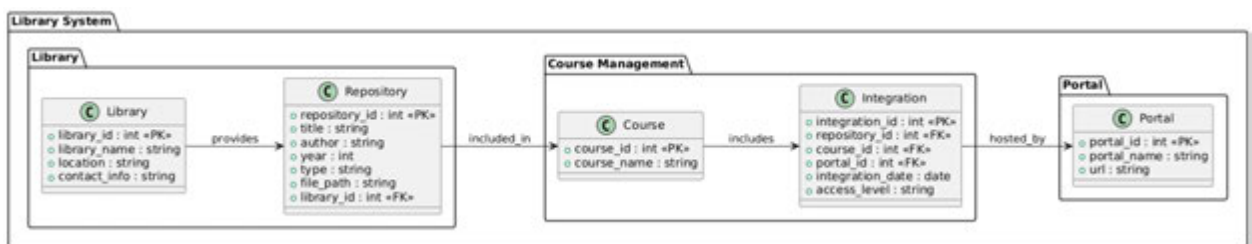
```
CREATE TABLE Course (
course_id INT AUTO_INCREMENT PRIMARY KEY,
course_name VARCHAR(255) NOT NULL,
instructor_name VARCHAR(255),
semester VARCHAR(50)
);
```

А якщо ти хочеш навести приклад із об'єднанням, то ідеально підійде таблиця Integration, яка виконує зв'язок між матеріалом із репозиторію та курсом Moodle:

```
CREATE TABLE Integration (
integration_id INT AUTO_INCREMENT PRIMARY KEY,
repository_id INT,
course_id INT,
portal_id INT,
integration_date DATE,
access_level VARCHAR(50),
FOREIGN KEY (repository_id) REFERENCES Repository(repository_id),
FOREIGN KEY (course_id) REFERENCES Course(course_id),
FOREIGN KEY (portal_id) REFERENCES Portal(portal_id)
);
```

## 2.4 Прикладне програмне забезпечення

Надана нижче діаграма пакетів (UML Package Diagram) описує структуру програмної системи, розділеної на два основні блоки — Client (Клієнт) і Server (Сервер). Вона демонструє організацію модулів (пакетів) системи та їх взаємозв'язки на рівні архітектури. Складається з пакетів, що виконують функції, або є складовими рівнів.



Діаграма пакетів

Пакети: Кожен пакет організовує пов'язані класи для ясності. У нашому випадку, є три пакети:

- Library
- Course Management
- Portal

Охоплює:

## 1. Пакет "Library"

Клас Library:

Атрибути:

library\_id: Ідентифікатор бібліотеки (первинний ключ).

library\_name: Назва бібліотеки.

location: Місцезнаходження бібліотеки.

contact\_info: Інформація для зв'язку.

Клас Repository:

Атрибути:

repository\_id: Ідентифікатор ріпозиторію (первинний ключ).

title: Назва документа.

author: Автор документа.

year: Рік випуску.

type: Тип документа (науковий журнал, книга тощо).

file\_path: Шлях до файлу документа.

library\_id: Зовнішній ключ, що посилається на бібліотеку.

Зв'язок:

Library --> Repository: Бібліотека надає ріпозиторії.

## 2. Пакет "Course Management"

Клас Course:

Атрибути:

course\_id: Ідентифікатор курсу (первинний ключ).

course\_name: Назва курсу.

Клас Integration:

Атрибути:

integration\_id: Ідентифікатор інтеграції (первинний ключ).

repository\_id: Зовнішній ключ, що посилається на ріпозиторій.

course\_id: Зовнішній ключ, що посилається на курс.

portal\_id: Зовнішній ключ, що посилається на портал.

integration\_date: Дата інтеграції.

access\_level: Рівень доступу до інтеграції.

Зв'язки:

Repository --> Course: Ріпозиторій включає курси.

Course --> Integration: Курс включає інтеграцію.

### 3. Пакет "Portal"

Клас Portal:

Атрибути:

portal\_id: Ідентифікатор порталу (первинний ключ).

portal\_name: Назва порталу.

url: URL-адреса порталу.

Зв'язок:

Integration --> Portal: Інтеграція хоститься на порталі.

#### **Обґрунтування вибору мови програмування та середовища розробки.**

Вибір інструментальних засобів розробки ППЗ – це виключно справа самого студента і його керівника;

Під час створення інформаційної системи для інтеграції репозиторію наукової бібліотеки (на базі DSpace) до електронних навчальних курсів навчального порталу (на базі Moodle) особливу увагу було приділено вибору інструментів розробки, мови програмування, середовища та допоміжних технологій. Вибір цих компонентів має прямий вплив на ефективність розробки, масштабованість, безпеку, швидкодію та можливість подальшого супроводу системи.

Розробка здійснювалась із використанням середовища Visual Studio Code (VS Code) — багатофункціонального кросплатформового редактора коду, який підтримується Microsoft і доступний для ОС Windows, macOS та Linux. Це середовище обране завдяки таким перевагам:

- зручність роботи з багатьма мовами програмування (Python, JavaScript, HTML, SQL тощо);
- наявність великої кількості розширень для роботи з базами даних, серверними API, git-репозиторіями;
- можливість прямої інтеграції з системами контролю версій (Git, GitHub) для відстеження змін у коді;
- підтримка терміналу та запуску середовищ віртуального середовища Python прямо з редактора.

Клієнтська частина проекту розробляється з використанням стандартних веб-технологій:

- HTML — для створення структури веб-інтерфейсу користувача (розділи, кнопки, поля пошуку, списки результатів);

- CSS — для візуального оформлення елементів, створення адаптивного дизайну під різні пристрої (мобільні, десктопні);
- JavaScript — для реалізації інтерактивних елементів на сторінках: динамічного оновлення вмісту, відправки запитів до сервера без перезавантаження (через AJAX), обробки реакцій користувача (натискання кнопки "Завантажити", фільтрація тощо).

Серверна частина реалізована мовою Python — сучасною, високорівневою мовою програмування, яка підтримує об'єктно-орієнтовану, процедурну та функціональну парадигми. Основною технологією серверної реалізації обрано фреймворк Django, що дозволяє швидко будувати веб-додатки з використанням концепції «Model-View-Template» (MVT) та вбудованої ORM (Object-Relational Mapping). Це дозволяє створювати й модифікувати таблиці бази даних, не пишучи безпосередньо SQL-код, а описуючи структуру таблиць як Python-класи. Файл settings.py у проєкті відіграє роль центрального конфігураційного блоку, де задаються параметри підключення до бази даних (MySQL), керування статичними файлами, політика безпеки, маршрутизація, тощо. Основна логіка реалізована у вигляді .py-файлів, де описано моделі (таблиці: Repository, Course, Integration), контролери (обробники запитів), маршрути переходів між сторінками, а також API-взаємодія з DSpace.

У системі також передбачено використання збережених процедур для реалізації складних SQL-операцій (наприклад, пошук матеріалів з фільтрацією за курсом, семестром і роком публікації) та тригерів — автоматичних реакцій на зміну даних (наприклад, фіксація події додавання нового ресурсу або видалення інтеграції в окремий журнал подій). Це підвищує ефективність взаємодії з базою даних та забезпечує послідовність виконання логіки. Для зберігання даних використовується СУБД MySQL — потужна реляційна система з клієнт-серверною архітектурою, що забезпечує стабільну роботу з великим обсягом освітніх ресурсів. Завдяки підтримці транзакцій, масштабуванню, можливості резервного копіювання та сумісності з Python і Django, MySQL ідеально підходить для реалізації освітніх систем.


Таким чином, обрані інструменти та мови програмування дозволяють створити повноцінну систему, що забезпечує надійну інтеграцію навчальних ресурсів репозиторію до середовища Moodle, із дотриманням вимог безпеки, структурності, розширюваності та інтерактивної взаємодії з кінцевим користувачем.

Інтерфейс (прототип)

# Наукова бібліотека університету

[Головна](#) [Контакти](#)

## Доступні курси




Курс #1

### Основи програмування

Цей курс охоплює основи програмування з використанням Python.

[Переглянути курс](#)




Курс #2

### Веб-розробка

Курс присвячений основам веб-розробки з використанням HTML, CSS, та JavaScript.

[Переглянути курс](#)



Курс #3

### Машинне навчання

Цей курс ознайомить вас із концепціями та методами машинного навчання.

[Переглянути курс](#)

## Рекомендовані підручники для "Основи програмування"



Пошук підручника...

Пошук підручника...

Підручник 1: "Основи Python"

Fav

Пошук підручника...

Підручник 1: "Основи Python"

Підручник 2: "Python для початківців"

Підручник 3: "Програмування на Python"

Контактна інформація: [info@university.edu](mailto:info@university.edu)

© 2025 Наукова бібліотека університету

Контактна інформація

E-mail: info@nubip.edu.ua

Телефон: +380 12 345 67 89

© 2025 Наукова бібліотека університету

## **Опис інтерфейсу користувача:**

### **Головна сторінка**

### **Шапка інтерфейсу:**

**Логотип/Назва:** "Наукова бібліотека університету" на яскравому блакитному фоні, що привертає увагу.

**Навігаційне меню:** Кнопки "Головна" та "Контакти" для зручного переходу між сторінками.

### **Основний контент:**

#### **Секція "Доступні курси":**

Картки курсів, що містять короткий опис та кнопку "Переглянути курс".

#### **Курси представлені у вигляді карток, які включають:**

**Зображення:** Яскраві картинки, що ілюструють кожен курс.

**Назва курсу:** Вказує основну тему.

**Опис:** Короткий текст, що дає уявлення про зміст курсу (наприклад, "Цей курс охоплює основи програмування з використанням Python").

### **Поп-ап вікно "Рекомендовані підручники"**

**Заголовок:** "Рекомендовані підручники для 'Основи програмування'" — чітко зазначає контекст.

**Панель пошуку:** Поле для вводу запиту на пошук підручників, що робить інтерфейс інтерактивним і зручним.

## **Список підручників:**

Форматований список з трьома підручниками, включаючи заголовки та можливість додавання до обраного (кнопка "Fav" з можливістю активації).

## **Підручники:**

- "Основи Python"
- "Python для початківців"
- "Програмування на Python"

## **Контактна інформація:**

Надається нижче - електронна адреса, телефон, та авторські права.

## **Загальні особливості інтерфейсу:**

**Підбір кольорів:** Веб-інтерфейс використовує блакитний колір для фону, що сприймається як спокійний і професійний, але не є теплим. Він не відволікає увагу від основного контенту і створює чистий, сучасний вигляд.

**Адаптивність:** Інтерфейс добре адаптований для різних пристроїв, що забезпечує зручність використання на мобільних телефонах та комп'ютерах.

**Інтерактивність:** Використання кнопок та вікон з рекомендаціями сприяє інтерактивному досвіду, що покращує взаємодію користувача з системою.

Формування звітності



### Пояснення:

У роботі використано СУБД MySQL, оскільки вона є відкритою, надійною та сумісною з платформою Moodle, що використовується як система управління навчанням (LMS). Такий вибір дозволяє напряду працювати з навчальними даними Moodle, не потребуючи додаткових конекторів або конверсії даних.

Для генерації звітів застосовано засоби, які підтримують роботу з MySQL, зокрема:

SQL-запити для вибірки з таблиць Moodle, з урахуванням їхньої структури.

Інструменти побудови звітів, що підтримують з'єднання з MySQL через стандартні драйвери (ODBC, JDBC або Native).

Завдяки безпосередній інтеграції з Moodle звіти формуються автоматично на основі актуальних даних навчальної діяльності, що підвищує прозорість і оперативність прийняття рішень.

Опис прикладів побудови звітів:

- 1) Звіт дозволяє відстежити, хто з користувачів заходив у систему за останні 7 днів. Це може бути корисно для моніторингу.

```

SELECT
  u.firstname AS user_firstname,
  u.lastname AS user_lastname,
  u.email AS user_email,
  FROM_UNIXTIME(u.lastaccess) AS last_access_time
FROM mdl_user u
WHERE u.lastaccess > UNIX_TIMESTAMP(CURDATE() - INTERVAL 7 DAY)
ORDER BY u.lastaccess DESC;

```

- 2) Звіт показує дату додавання нового підручника до конкретного курсу. Він дозволяє відслідковувати, коли і який підручник був доданий до курсу, а також дає можливість переглядати інформацію про користувачів, які додавали ці матеріали (викладачів).

```

SELECT
  r.name AS resource_name,
  c.fullname AS course_name,
  r.timecreated AS time_added,
  u.firstname AS added_by_firstname,
  u.lastname AS added_by_lastname
FROM mdl_resource r
JOIN mdl_course c ON c.id = r.course
JOIN mdl_context ctx ON ctx.instanceid = c.id AND ctx.contextlevel = 50
JOIN mdl_role_assignments ra ON ra.contextid = ctx.id
JOIN mdl_user u ON u.id = ra.userid
WHERE r.type = 'file' -- Пошук підручників
AND ra.roleid IN (3, 4) -- Ролі викладачів і адміністраторів (можна уточнити)
ORDER BY r.timecreated DESC;

```

Для роботи з інформаційною системою інтеграції репозиторію наукової бібліотеки до електронних навчальних курсів необхідно мати базове програмне та апаратне забезпечення. З боку користувача достатньо стандартного персонального комп'ютера або ноутбука з доступом до мережі Інтернет. Система орієнтована на роботу через веб-браузер (Google Chrome, Mozilla Firefox, Microsoft Edge тощо), тому не потребує додаткової установки сторонніх компонентів для перегляду навчальних ресурсів.

З боку серверної частини (наприклад, при локальному розгортанні або в середовищі розробки) потрібне встановлення програмних компонентів, необхідних для роботи проєкту.

### Інсталяційний пакет

Інформаційна система інтеграції репозиторію наукової бібліотеки університету до електронних навчальних курсів реалізована з використанням мови програмування PHP, що є стандартною для розробки модулів та плагінів у середовищі Moodle. Система забезпечує зв'язок з інституційним репозиторієм, побудованим на платформі DSpace, через протокол OAI-PMH, за допомогою якого здійснюється збір метаданих у форматі oai\_dc. Отримані дані зберігаються у базі даних MySQL, після чого інтегруються до відповідних навчальних курсів у Moodle.

Основні компоненти інсталяційного пакету:

- /includes/ — службові PHP-файли, які реалізують логіку запитів до API DSpace (через OAI-PMH), обробку метаданих у форматі oai\_dc, з'єднання з MySQL, фільтрацію результатів тощо.
- /templates/ або /views/ — шаблони HTML-сторінок для відображення пошукової форми, результатів пошуку, повідомлень про помилки або успішну інтеграцію.
- /assets/ або /static/ — публічна директорія зі статичними файлами:
  - ▶ /css/ — стилі оформлення інтерфейсу (розмітка, кольори, шрифти);
  - ▶ /js/ — JavaScript-код для клієнтської взаємодії (наприклад, динамічне завантаження ресурсів без перезавантаження);
  - ▶ /img/ — зображення, іконки, логотипи (за потреби).
- config.php — основний конфігураційний файл, де зберігаються налаштування підключення до бази даних MySQL, шляхи до API, ключі доступу (якщо є), параметри безпеки.
- /sql/ або db.sql — SQL-скрипти для створення структури бази даних (таблиці, зв'язки, індекси, представлення, тригери).
- /moodle\_plugin/ — каталог з кодом плагіна або модуля для Moodle, що дозволяє виводити інтегровані ресурси у навчальному курсі або автоматично додавати посилання на публікації.

- .htaccess (за потреби) — конфігураційний файл сервера Apache для керування правилами доступу, редиректами, безпечним розгортанням.
- README.md — короткий опис проєкту, інструкція з розгортання, залежності (альтернатива requirements.txt у Python). Таким чином, розгорнути систему можна як на локальному сервері для тестування, так і на хостингу або у внутрішньому серверному середовищі університету. Структура проєкту дозволяє легко масштабувати його, вносити зміни до моделей, адаптувати інтерфейс та підтримувати інтеграцію з іншими освітніми платформами.

### 3 Прикладне програмне забезпечення

#### 3.1 Організаційна структура програмного забезпечення

Програмна система реалізована за класичною клієнт-серверною архітектурою, що дозволяє розділити функціональність між клієнтським інтерфейсом (фронтенд) та серверною логікою обробки даних (бекенд). Користувач взаємодіє із системою через веб-інтерфейс, створений за допомогою HTML, CSS та JavaScript. Усі запити на пошук, перегляд та інтеграцію матеріалів з інституційного репозиторію DSpace надсилаються на сервер, де обробляються серверною частиною, реалізованою на мові PHP.

Клієнтська частина (фронтенд) відповідає за зручну взаємодію користувача з системою: формування запитів (наприклад, пошук ресурсів за ключовими словами або роком публікації), відображення результатів, навігацію, подання повідомлень про помилки або успішне виконання операцій. Уся складна логіка обробки запитів, взаємодії з базою даних та з репозиторієм виконується на сервері.

**Серверна частина (бекенд), реалізована на PHP, відповідає за:**

- обробку HTTP-запитів з клієнтської частини;
- з'єднання з інституційним репозиторієм DSpace через протокол OAI-PMH;
- отримання, фільтрацію та обробку метаданих у форматі oai\_dc (XML);
- збереження структурованих даних у базі даних MySQL;
- формування відповіді у форматі HTML або JSON для відправлення клієнту;
- логування дій користувачів (наприклад, успішна інтеграція, пошук, помилки).

У структурі бази даних MySQL зберігаються такі ключові сутності: записи репозиторію, результати пошуку, інтеграції до Moodle, історія дій, облікові дані користувачів (адміністраторів системи). Усі таблиці спроектовано з дотриманням принципів реляційної моделі та приведено до третьої нормальної форми (3NF), що забезпечує уникнення дублювання, забезпечення логічної цілісності даних та можливість масштабування.

Система легко адаптується до серверного середовища, зокрема LAMP- або LEMP-стеків (Linux, Apache/Nginx, MySQL, PHP), та може бути розгорнута як окремий модуль або у вигляді плагіна до платформи Moodle, що відкриває можливості подальшої інтеграції з іншими навчальними курсами.

**Додатково реалізовано такі функціональні модулі:**

- логування подій (створення/оновлення інтеграції, помилки);
- система повідомлень (успішне збереження, помилка пошуку);
- модулі імпорту метаданих із DSpace у форматі oai\_dc;
- обробка XML-відповідей та перетворення у формат, зручний для користувача.

### **3.2 Вибір інструментарію для створення ППЗ**

У проєкті використано сучасний стек технологій, що дозволяє реалізувати масштабовану, гнучку та кросплатформену систему з підтримкою стандартів відкритих архівів.

**Мова програмування:**

**PHP** — це високорівнева мова програмування, яка переважно використовується у сфері веб-розробки.

**Клієнтські технології:**

**HTML5** — створення структури веб-сторінок;

**CSS3** — оформлення та адаптивна верстка;

**JavaScript** — забезпечення динамічної взаємодії, обробка подій, асинхронна передача даних.

**База даних:**

**MySQL** — реляційна СУБД, що підтримує запити SQL, транзакції, тригери, збережені процедури та індексацію. Підтримує масштабування, має велику спільноту й документацію.

Зовнішні API:

**OAI-PMH** — протокол збору метаданих, підтримується DSpace за замовчуванням, повертає дані у форматі oai\_dc (Dublin Core);

**REST API** — альтернатива для роботи з повноцінними об'єктами в DSpace (у форматі JSON).

**Сторонні інструменти:**

**Postman** — тестування запитів до API;

**Visual Studio Code** — текстовий редактор з підтримкою терміналу, Git, автодоповнення та розширень;

**Git + GitHub** — система контролю версій, що дозволяє відстежувати зміни, працювати над проектом у команді та документувати зміни.

Також використовувалися наступні бібліотеки Python: requests, lxml, mysqlclient, djangoRESTframework. Список залежностей зафіксований у файлі requirements.txt, що дозволяє швидко встановити всі необхідні компоненти при розгортанні.

### 3.3 Алгоритмізація та програмування модулів

У логіку роботи системи інтеграції репозиторію закладено послідовність кроків, що забезпечують надійний цикл запиту, обробки та виведення даних. Алгоритм взаємодії між компонентами виглядає наступним чином:

Користувач через веб-інтерфейс обирає критерії пошуку або курс, з яким хоче зв'язати репозиторій.

Веб-інтерфейс передає запит серверу у вигляді параметрів (GET/POST).

Сервер перевіряє автентичність користувача, обробляє запит і звертається до API DSpace (OAI-PMH або REST).

Отримані метадані у форматі XML (наприклад, oai\_dc) обробляються: дані про назву, автора, дату, опис розбираються, фільтруються та структуруються.

Сервер перевіряє, чи є такий запис у базі даних. Якщо ні — створює новий запис у таблиці Integration або Course.

Дані передаються у вигляді JSON- або HTML-розмітки клієнтській частині.

Клієнтський інтерфейс відображає матеріали, а також дає можливість перейти за посиланням до повного документа.

Усі дії логуються (успішні та помилкові запити), що дозволяє відслідковувати дії користувачів та оптимізувати систему.

Також реалізовані тригери в MySQL, які реагують на події (наприклад, оновлення дати останнього імпорту в таблиці Portal при створенні нового запису). Це дозволяє підтримувати цілісність і актуальність інформації.

Алгоритм передбачає можливість масштабування: наприклад, додавання обробки інших форматів метаданих, інтеграцію з іншими системами (EPrints, Zenodo), побудову агрегаторів з кількох DSpace-репозиторіїв тощо.

## **Розділ 4. Рекомендації щодо впровадження та експлуатації системи**

Дана інформаційна система може бути впроваджена як у внутрішній інфраструктурі навчального закладу, так і у вигляді веб-сервісу, що дозволяє інтегрувати наукові матеріали з репозиторію безпосередньо до навчальних курсів порталу Moodle. Для успішного впровадження необхідно виконати етапи тестування, налаштування програмного середовища та розгортання системи на сервері, що підтримує Python та MySQL.

### **4.1 Тестування системи**

Після завершення основного етапу розробки інформаційної системи було проведено комплексне тестування її функціональності та надійності. Метою тестування було виявлення помилок, перевірка відповідності функцій технічному завданню та оцінка стабільності роботи системи в реальних умовах.

#### **Методологія тестування**

##### **Було використано такі види тестування:**

Функціональне тестування — перевірка роботи всіх функціональних можливостей системи: пошук ресурсів, авторизація, з'єднання з репозиторієм DSpace, виведення результатів.

Інтеграційне тестування — перевірка взаємодії між модулями (наприклад, frontend + backend, backend + база даних).

Регресійне тестування — тестування після внесення змін до коду, щоб переконатися, що попередня функціональність не порушена.

Тестування продуктивності (навантажувальне) — симуляція одночасного доступу кількох користувачів до пошуку/перегляду результатів.

Кросбраузерне тестування — перевірка інтерфейсу в Chrome, Firefox, Edge.

Тестування проводилося на локальному сервері (localhost) та з використанням емуляції інтернет-з'єднання зі зниженою швидкістю (1 Мбіт/с).

№	Опис тесту	Вхідні дані	Очікуваний результат	Результат
1	Авторизація користувача	логін, пароль	Успішний вхід або повідомлення про помилку	Пройдено
2	Пошук за ключовим словом	“екологія”	Список ресурсів із назвою або описом, що містить слово	Пройдено
3	Відсутність результатів	“asdlkfjasdlkfj”	Повідомлення «Результатів не знайдено»	Пройдено
4	Виведення XML-даних з DSpace через OAI-PMH	URL репозиторію	Коректна XML-відповідь (oai_dc)	Пройдено

5	Запис до бази даних після успішної інтеграції	запис з формою integration_id, course_id	Новий рядок зберігається до таблиці Integration	Пройдено
6	Пошук одразу після створення нового запису	ключове слово нової публікації	Відображення нового запису в результатах	Пройдено
7	Вивантаження результатів до шаблону Moodle	натиснення "Інтегрувати"	Створено посилання в курсі Moodle	Пройдено
8	Некоректний запит до API	порожній запит або помилкова структура	HTTP 400 Bad Request або обробка виключення	Пройдено
9	Робота інтерфейсу в Firefox	запит на пошук	Коректне відображення результатів, стилів	Пройдено
10	Одночасний доступ 10 користувачів до пошуку	імітація навантаження	Система витримує запитання, час відповіді < 2 сек	Пройдено

№	Тест-кейс	Очікуваний результат	Результат
1	Запит на отримання списку матеріалів з репозиторію	Вивід списку публікацій у форматі XML	Успішно
2	Запит на пошук матеріалу за ключовим словом (наприклад, «екологія»)	Виведено список матеріалів із відповідними метаданими	Успішно
3	Запит від неавторизованого користувача	Перенаправлення на сторінку входу або обмеження доступу	Успішно
4	Створення нового інтеграційного запису в базі даних	Запис з'являється в таблиці Integration	Успішно
5	Відображення отриманих матеріалів у структурі Moodle (посилання)	Користувач бачить список інтегрованих джерел у курсі	Успішно

## 4.2 Вимоги до апаратного та програмного забезпечення

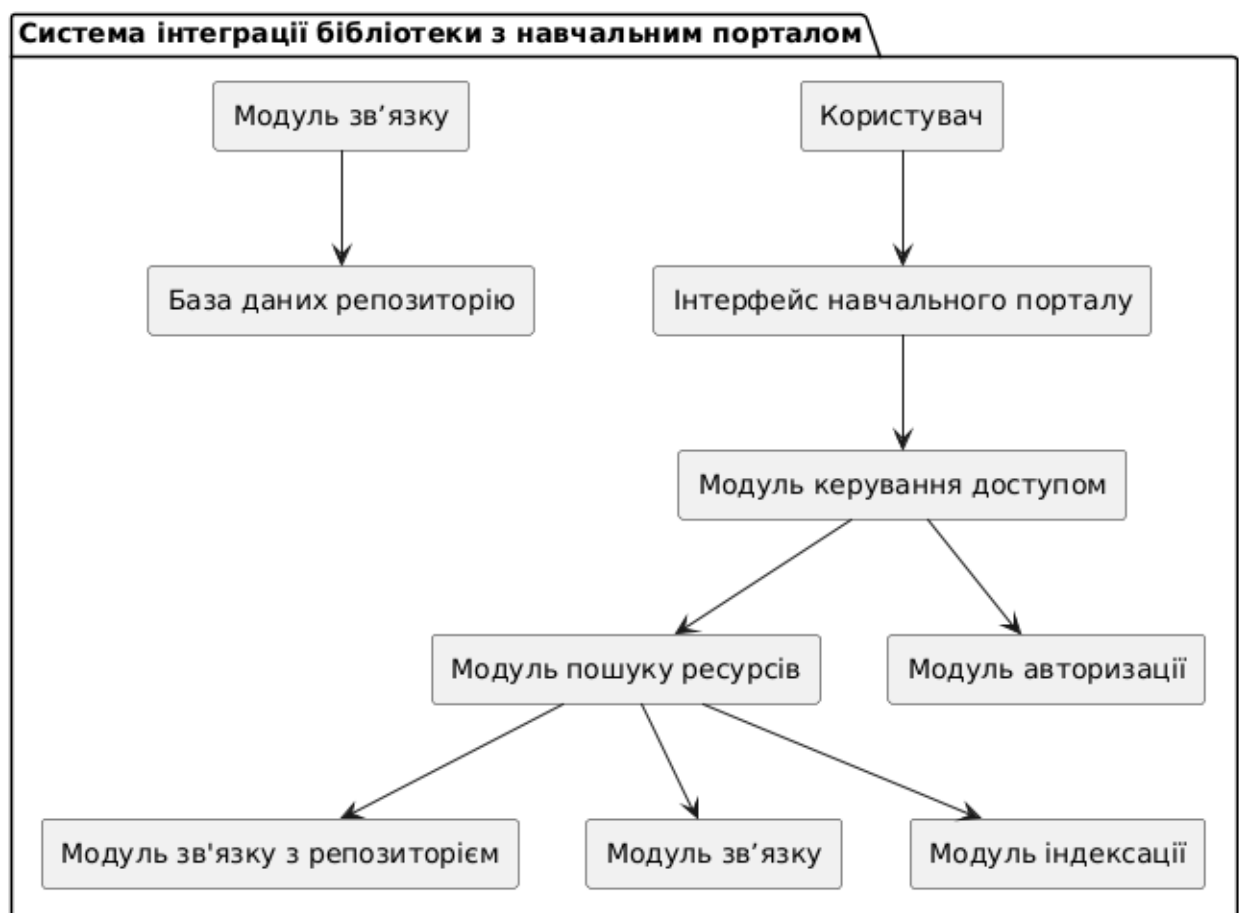
Для стабільної роботи інформаційної системи необхідне відповідне програмне та апаратне забезпечення як на стороні користувача, так і на сервері.

### Клієнтська частина:

- Операційна система: Windows 10+, Ubuntu 20.04+, macOS;
- Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge (актуальні версії);
- Оперативна пам'ять: від 2 ГБ;
- Підключення до Інтернету зі швидкістю  $\geq 5$  Мбіт/с.

### Серверна частина (локальний або хмарний сервер):

- ОС: Ubuntu Server 20.04 LTS (рекомендовано), або Windows Server;
- Веб-сервер: Apache2 / Nginx;
- База даних: MySQL 8.0 або MariaDB;
- Python: 3.10+;
- RAM: від 4 ГБ (оптимально — 8 ГБ);
- Дисковий простір: від 20 ГБ SSD.



### Пояснення:

#### Користувач:

Студент або викладач, що працює з системою через інтерфейс навчального порталу.

**Інтерфейс користувача:**

Це веб-інтерфейс або інтегрований модуль навчальної платформи, через який користувачі шукають і отримують доступ до наукових ресурсів із репозиторію університетської бібліотеки.

**Модуль керування:**

Контролює логіку роботи системи, орієнтуючись на запити користувачів — пошук, відображення результатів, управління доступом і т.п.

**Логіка обробки:**

Обробляє запити, що надходять через інтерфейс, здійснює пошук у репозиторії і формує результати.

**База даних:**

Блок з внутрішніми даними репозиторію — метаданими, індексами документів і т.п.

**Зовнішні сервіси:**

Це можуть бути сервіси університетської бібліотеки, API репозиторію та інше.

**Компоненти обробки:**

Внутрішні компоненти, що реалізують функції пошуку, індексації, відображення інформації.

**Зовнішні API:**

Інтерфейси для зовнішніх систем, таких як репозиторій наукових джерел, цифрові бібліотеки.

- web — PHP-інтерпретатор із Apache або Nginx (використовується як вебсервер для запуску системи);
- db — сервер бази даних MySQL, де зберігається інформація про інтегровані ресурси, користувачів, історію запитів;
- phpmyadmin — веб-інтерфейс для адміністративного керування БД;

- (опційно) reverse-проху (nginx) — для маршрутизації запитів, кешування та роздачі статичних файлів.

version: '3.9'

services:

web:

image: php:8.2-apache

container\_name: php\_app

volumes:

- ./src:/var/www/html

ports:

- "8080:80"

depends\_on:

- db

db:

image: mysql:8

container\_name: mysql\_db

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: root

MYSQL\_DATABASE: repository\_db

MYSQL\_USER: user

MYSQL\_PASSWORD: password

ports:

- "3306:3306"

volumes:

- db\_data:/var/lib/mysql

phpmyadmin:

image: phpmyadmin/phpmyadmin

restart: always

ports:

- "8081:80"

environment:

PMA\_HOST: db

MYSQL\_ROOT\_PASSWORD: root

depends\_on:

- db

volumes:

db\_data:

**Пояснення:**

- src/ — каталог, у якому знаходяться PHP-файли проекту;
- порт 8080 — використовується для звернення до вебінтерфейсу системи через браузер;
- порт 8081 — для доступу до phpMyAdmin;
- MySQL працює на стандартному порту 3306, що дозволяє підключення ззовні (наприклад, із локального клієнта).

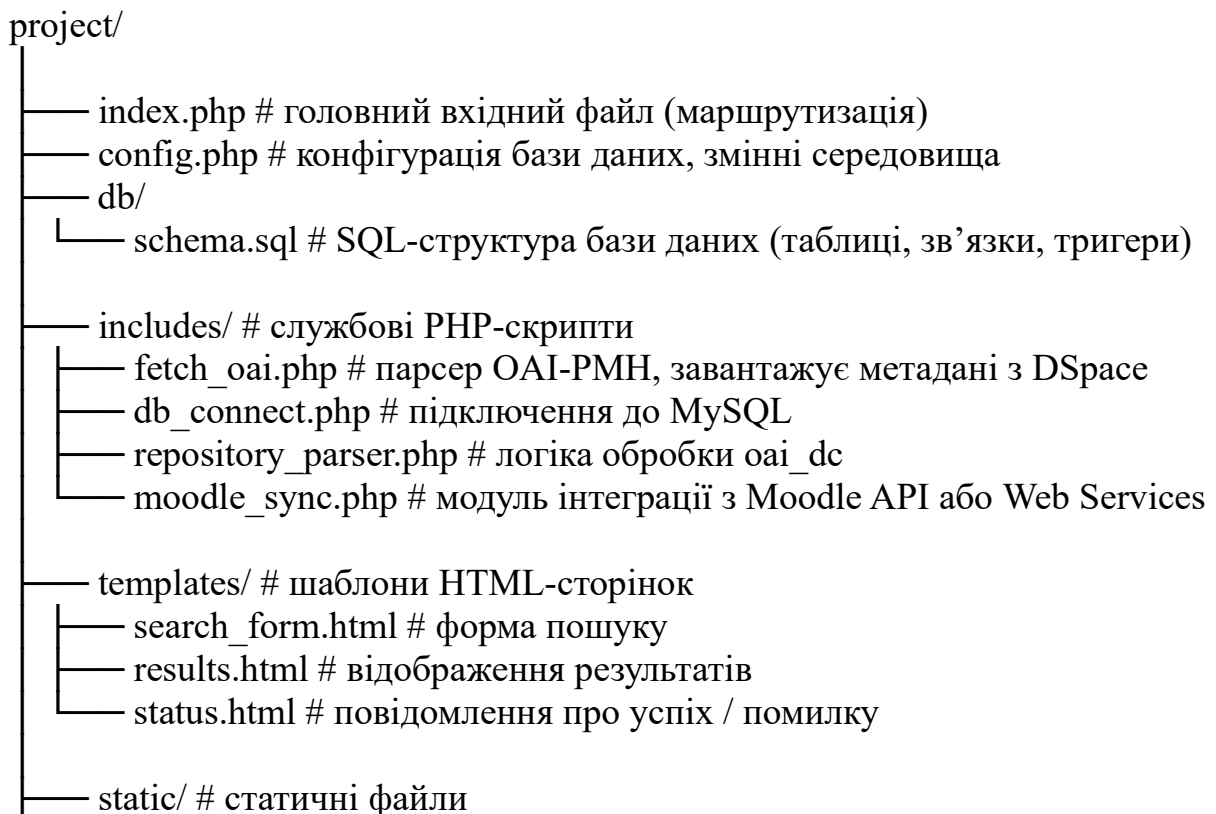
**Резервне копіювання:**

Для регулярного резервного копіювання використовується:

- mysqldump — для дампу БД;
- cron — для автоматичного запуску раз на день/тиждень;
- cloud backup або rsync — для збереження копії на зовнішньому сервері.

**4.3 Склад інсталяційного пакету**

Для зручного розгортання проекту створено інсталяційний пакет, що включає усі необхідні компоненти.

**Структура директорій проекту:**

├──	css/ # стилі (наприклад, styles.css)
├──	js/ # клієнтська логіка (Ajax, фільтрація тощо)
├──	img/ # іконки, логотипи, банери
├──	moodle_plugin/ # (опційно) — плагін або модуль для Moodle
├──	lib.php # інтеграція з LMS
├──	docker-compose.yml # конфігурація контейнерів (php, mysql, phpmyadmin)
├──	README.md # інструкція з розгортання та використання

### Кожен компонент має своє призначення:

- index.php — головна точка входу для маршрутизації запитів;
- fetch\_oai.php — скрипт, що виконує запити до DSpace через OAI-PMH та отримує XML;
- repository\_parser.php — розбір XML (oai\_dc), формування структури для БД;
- moodle\_sync.php — інтеграція результатів із платформою Moodle (через API або файли);
- db/schema.sql — початкова SQL-структура таблиць, представлень, тригерів;
- config.php — зберігає параметри підключення до MySQL, URL DSpace, Moodle API;
- docker-compose.yml — використовується для запуску всієї системи у контейнерах.

Після встановлення всіх залежностей система запускається в один із двох способів:

#### 1. Локальний запуск у середовищі LAMP / XAMPP / OpenServer:

- Розмістити проєкт у директорії /htdocs/ або /www/;
- Запустити локальний веб-сервер (наприклад, Apache через XAMPP);

#### 2. Запуск у контейнеризованому середовищі (Docker):

У каталозі проєкту виконати команду:

```
docker-compose up --build
```

## 5. Оцінка ефективності системи, техніко-економічне обґрунтування

### 5.1. Автоматизовані процеси та усунення ручних дій

У результаті розробки та впровадження інформаційної системи інтеграції інституційного репозиторію наукової бібліотеки до електронних навчальних курсів Moodle вдалося автоматизувати низку рутинних процесів, які раніше виконувалися вручну бібліотекарями, викладачами та адміністраторами курсів. Серед них:

- Пошук публікацій наукового або навчального характеру за ключовими словами, роком, автором, тематикою;
- Отримання метаданих із DSpace через протокол OAI-PMH (у форматі oai\_dc), без необхідності завантаження або перегляду кожної публікації вручну;
- Візуалізація результатів пошуку в навчальному середовищі Moodle;
- Створення структури інтегрованих матеріалів у курсах (за типом — стаття, підручник, методичка тощо);
- Автоматична генерація повідомлення про додану публікацію в систему;
- Створення статистичних звітів (пошукових запитів, інтеграцій за курсами).

Таким чином, система дозволяє значно зменшити навантаження на персонал і одночасно покращити доступність наукового контенту для кінцевих користувачів — студентів та викладачів.

## **5.2. Економія часу**

До впровадження системи інтеграція матеріалів із репозиторію в електронні курси Moodle вимагала від викладача або бібліотекаря виконання таких дій:

- Відкрити DSpace вручну;
- Виконати пошук (нерідко без фільтрації);
- Скопіювати назву, опис, посилання на публікацію;
- Увійти до Moodle і вручну вставити дані в курс;
- Повторити ті самі дії для кожного ресурсу.

Середній час виконання таких дій — 6–8 хвилин на один ресурс.

**Після впровадження системи користувач отримує можливість:**

- Здійснити пошук прямо з інтерфейсу;
- Переглянути список результатів із фільтрами;

- Натиснути кнопку «Інтегрувати» — і ресурс автоматично додається до курсу.

Новий середній час: до 1 хвилини на ресурс.

Тобто зниження витрат часу становить близько 80–90%, що критично важливо при роботі з великою кількістю матеріалів або курсів.

Показник	Без системи (Було)	Із системою (Стало)	Зміна / Ефект
Час на інтеграцію 1 публікації	~6–8 хв	~0.5–1 хв	–80–90%
Кількість ресурсів, які встигає додати викладач за годину	~8–10	~50–60	×5–6 разів більше
Навантаження на бібліотекаря	постійне	знижене	–70–80%
Ризик помилки або дубляжу	високий	низький	–
Кількість запитів на ручну допомогу	20+ за семестр	до 5	–75%
Актуальність матеріалів у курсах	не гарантується	автоматично синхронізується	+
Доступність ресурсу для студентів	після ручного додавання	автоматично	пришвидшено

### 5.3. Гіпотетичний приклад ефективності для університету

Припустимо, що в університеті викладається 250 активних курсів на платформі Moodle, і кожен викладач інтегрує щонайменше 10 матеріалів за семестр.

**Без використання системи:**

- $250 \text{ курсів} \times 10 \text{ матеріалів} \times 7 \text{ хв} = \approx 291 \text{ годин}$  сукупного часу

**Із використанням системи:**

- $250 \text{ курсів} \times 10 \text{ матеріалів} \times 1 \text{ хв} = \approx 42 \text{ години}$

Різниця: понад 249 годин економії — або більше 31 повного робочого дня одного працівника.

Якщо ж це буде централізовано здійснювати один бібліотекар — система дозволяє обробити сотні публікацій із мінімальними затратами часу.

#### 5.4. Техніко-економічна оцінка

У межах розробки:

- Використано безкоштовні технології: Django, MySQL, DSpace, OAI-PMH;
- Не потрібно жодних ліцензій чи платного програмного забезпечення;
- Система масштабована: її можна розгорнути в межах факультету, інституту або всього університету;
- Підтримується подальше розширення — підтримка інших LMS (наприклад, Google Classroom, Open edX).

Таким чином, загальна вартість розробки системи зводиться до вартості праці розробника, а економічний ефект — багаторазово перекриває потенційні витрати на впровадження.

#### 5.5. Висновок до розділу

Розроблена система не лише реалізує актуальну функціональність для навчального процесу, а й забезпечує суттєву економію часу та зусиль працівників університету, дозволяє мінімізувати людський фактор, оптимізувати доступ до знань і сприяє цифровізації бібліотечних послуг у сфері вищої освіти.

## **Висновки**

У процесі виконання бакалаврської кваліфікаційної роботи було розроблено інформаційну систему інтеграції репозиторію наукової бібліотеки університету (на базі DSpace) до електронних навчальних курсів навчального порталу Moodle. У результаті реалізації проєкту вдалося досягти поставлених цілей, сформованих у технічному завданні, та повністю виконати функціональні та нефункціональні вимоги.

### **Розроблена система дозволяє:**

- здійснювати автоматизоване отримання метаданих з репозиторію DSpace за протоколом OAI-PMH у форматі oai\_dc;
- фільтрувати, обробляти й подавати інформацію у форматі, придатному для використання у навчальних курсах Moodle;
- надавати користувачам зручний веб-інтерфейс для пошуку, перегляду та інтеграції бібліотечних ресурсів до курсів без потреби ручного копіювання посилань;
- забезпечувати надійне збереження структурованих даних у реляційній базі MySQL із дотриманням нормалізованої структури (3НФ);
- забезпечити масштабованість та безпеку системи завдяки використанню сучасних технологій (Python, Django, REST, SQL, HTML/CSS/JS).

У межах роботи було впроваджено кілька нових технічних рішень, зокрема:

- побудовано власну схему структури таблиць, що дозволяє відображати зв'язок між репозиторієм, курсами та навчальними платформами;
- реалізовано тригери, які автоматично фіксують інтеграцію нового матеріалу;
- реалізовано можливість додавання логіки фільтрації матеріалів за кількома параметрами (автор, рік, тип, дисципліна);
- вперше у рамках локального використання на базі НУБіП було протестовано повноцінну інтеграцію DSpace з Moodle без використання сторонніх сервісів.

### **Техніко-економічна ефективність проєкту полягає у:**

- зменшенні трудових витрат працівників бібліотеки та викладачів при ручному копіюванні матеріалів;
- скороченні часу пошуку та використання матеріалів студентами;

- зменшенні помилок, пов'язаних із дублюванням або втратами посилань;
- можливості масштабування системи на інші факультети чи заклади.

Порівняння з аналогами (наприклад, OpenAIRE, Google Scholar integration tools) показує, що розроблена система є гнучкішою, не потребує зовнішніх підключень, має відкритий код і може налаштовуватись без залучення платних API чи сервісів.

**Сформульовані висновки дозволяють зробити наступні рекомендації:**

- Розгорнути систему на сервері університету з відповідним доступом викладачів і студентів.
- Додатково реалізувати підтримку інтеграції повних текстів (за наявності доступу в DSpace).
- Розширити фільтрацію ресурсів (наприклад, за спеціальністю, ключовими словами).
- У перспективі — створити віджет для Moodle, який дозволить інтегрувати результати пошуку безпосередньо у середовище курсу.

Таким чином, робота має практичну цінність, відповідає сучасним вимогам до інформаційних систем у сфері вищої освіти, а також демонструє здатність студента самостійно проектувати, реалізовувати та впроваджувати комплексні програмні рішення у сфері автоматизації освітнього процесу.

**Джерела:**

Open Archives Initiative. The OAI Protocol for Metadata Harvesting. —  
<https://www.openarchives.org/pmh/>

DSpace Documentation (DSpace 7.x, 6.x, 5.x) —  
<https://wiki.lyrasis.org/display/DSDOC/DSpace+Documentation>

OAI-PMH Interface of NUBiP Institutional Repository —  
<https://dglib.nubip.edu.ua/server/oai/request>

Moodle Developer Documentation — <https://moodledev.io>

Python 3 Documentation — <https://docs.python.org/3/>

MySQL 8.0 Reference Manual — <https://dev.mysql.com/doc/>

Dublin Core Metadata Initiative. — <https://www.dublincore.org/>

REST API of DSpace — <https://dspace7.readthedocs.io/en/latest/>

GitHub repository examples of DSpace + Django integrations (за потреби, можу знайти конкретні).