

Астістова Т.І.

Київський національний університет технологій та дизайну

Дроменко В.Б.

Таврійський національний університет імені В.І. Вернадського

Калашник В.Ю.

Київський національний університет технологій та дизайну

РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІНФОРМАЦІЙНОГО РЕСУРСУ БІБЛІОТЕКИ ЗАКЛАДІВ ОСВІТИ

В статті наведено аналіз використання цифрових технологій в інформаційних системах наукових бібліотек провідних університетів України. На основі проведеного аналізу визначені функціональні вимоги та завдання для розроблення інтерактивного додатку при взаємодії з бібліотекою закладу освіти за сучасними стандартами.

Архітектура розробленого web-додатку містить три основні компоненти: клієнтський компонент, серверний компонент і сервер бази даних, яка реалізована за допомогою використання шаблону MVC (Model-View-Controller).

Клієнтська частина реалізована за допомогою інтегрованого середовища розробки програмного забезпечення IntelliJ Idea та системи для побудови шаблонів і сторінок мовою розмітки HTML та стилізована за допомогою фреймворку Bootstrap та CSS та JS. Для динамічної генерації HTML-шаблонів було використано сучасний серверний шаблонізатор для Java Thymeleaf. Для реалізації серверної частини використано мову програмування Java та фреймворк Spring Boot з використанням Java бібліотек. Для управління базою даних обрано PostgreSQL. Для ORM (Object-Relational Mapping) був обраний Hibernate, який забезпечує зручний спосіб взаємодії з базою даних через об'єкти Java Hibernate, Spring Data JPA.

Розроблений web-додаток є закінченим програмним продуктом, який пройшов апробацію та використовується в Київському фаховому коледжу прикладних наук, і забезпечує підтримку мультимедійних ресурсів з субтитрами та жестовою мовою для здобувачів з вадами слуху. В проєкті реалізована генерація QR-коду для кожної книги, за допомогою якого читач може отримати завчасно додану до своїх улюблених книжку. Запроваджена система зменшує часові витрати користувача та навантаження на бібліотекаря, ліквідує паперову документацію та робить процедуру отримання книг легкою та прозорою.

Ключові слова: архітектура, сервіси, методи, класи, контролери, технологічний стек, MVC, PostgreSQL, інтегроване середовище IntelliJ Idea, Framework java Spring Boot, HTML, CSS.

Постановка проблеми. Сучасний навчальний процес у закладах освіти та сфери, які пов'язані з цим, все більше потребують використання цифрових технологій з метою підвищення якості освіти [1].

Бібліотека, як частина закладу освіти, також потребує інновацій та оновлення застарілої системи інформаційного ресурсу. Сучасні технології змінили вимоги і до організації діяльності бібліотек. Зміни відбувалися у напрямку доступності та модернізації роботи бібліотеки, у використанні сучасних технологій та фреймворків, інформаційних функцій, розробленні баз даних, поліпшені масштабованості, дизайну, інтерактивності. Сучасна бібліотека не лише забезпечує доступ до

різноманітних джерел інформації, але й створює середовище, сприятливе для навчання, розвитку та надає здобувачам та викладачам доступ до книг, наукових журналів, електронних ресурсів та інших навчальних матеріалів, що є важливими для навчання та досліджень [2]. З розвитком хмарних технологій, штучного інтелекту та інших інноваційних напрямків, потреба в програмному забезпеченні лише зростає. Для вирішення поставлених проблем, потрібно побудувати гнучку систему, що легко адаптується на основі нових принципів з використанням найсучасніших комп'ютерних та інформаційних технологій, які мають в своєму розпорядженні засоби для організації будь-яких процесів, у тому числі і в освітній сфері.

Електронні бібліотеки використовують електронний формат видань та забезпечують доступ до віддаленого ресурсу за допомогою телекомунікаційних технологій. Читач, який звертається до бібліотеки, повинен мати можливість отримати необхідну інформацію, як в електронному вигляді, так і в надрукованому [3]. Система особистого кабінету читача повинна мати користувацький інтерфейс з сучасним дизайном для зручного перегляду, замовлення, бронювання та сортування книг, наявних в бібліотеці. Найбільшим недоліком в інтерфейсі вебдодатків багатьох бібліотек закладів освіти є ігнорування потреб людей з обмеженими можливостями, що значно обмежує доступність і зручність використання, що може призвести до виключення великої кількості потенційних користувачів.

Аналіз останніх досліджень і публікацій. Існуючі платформи для розробки онлайн електронно-інформаційних ресурсів пропонують цінні інформаційні ресурси, але обмеження з точки зору доступності, функціональності та технологій перешкоджають їхній ефективності для користувачів з різноманітними потребами.

Щоб окреслити функціональні вимоги до розробки онлайн бібліотеки для закладу освіти, проведемо аналіз існуючих інформаційних систем наукових бібліотек провідних університетів України, зокрема: Київського національного університету імені Тараса Шевченка (КНУ), Львівського національного університету імені Івана Франка (ЛНУ), Національного університету «Києво-Могилянська академія» (НаУКМА), Національного університету біоресурсів і природокористування України (НУБіП), Київського національного університету технологій та дизайну (КНУТД).

1. Наукова бібліотека КНУ ім. Тараса Шевченка [4]. Електронна бібліотека одного з провід-

них київських університетів має великий книжковий фонд та багату кількість параметрів пошуку інформації. Інтерфейс каталогу має загальну продуманість для швидкого і легкого пошуку інформації, максимальну зрозумілість всіх елементів управління. До недоліків системи можна віднести використання застарілих версій мов програмування (використання php версії 4.4.9, яка була випущена ще у 2008 році) (див. рис. 1). Актуальна версія бібліотеки потребує оновлення інтерфейсу та застосування більш новітніх технологій. Користувацький інтерфейс не відповідає вимогам UI/UX дизайну, які передбачають ознайомлення з інтерфейсом майже не витрачаючи часу на те, щоб розібратись, як все працює.

2. Наукова бібліотека ЛНУ ім. Івана Франка [5]. Вебсайт наукової бібліотеки пропонує користувачам широкий спектр функцій, зокрема: пошук книги за назвою, автором, темою, ISBN/ISSN та іншими критеріями. Система пошуку з автозаповненням та розширеним пошуком дає змогу користувачам знайти шукану інформацію, але в кінцевому результаті можна отримати лише інформацію про наявність книги у бібліотеці (рис. 2). Електронних джерел інформації система не пропонує, що є великим мінусом у сучасному світі, також сайт не адаптовано під мобільні телефони.

3. Наукова бібліотека Національного університету «Києво-Могилянська академія» [6]. Система онлайн бібліотеки на перший погляд має найбільш привабливий інтерфейс та функціонал на сьогоднішній день (рис. 3).

Але при роботі з функціоналом (натисканні клавіш) відбувається переадресація на застарілу інформаційну систему (рис. 4) у якій інтерфейс не відповідає сучасним вимогам.

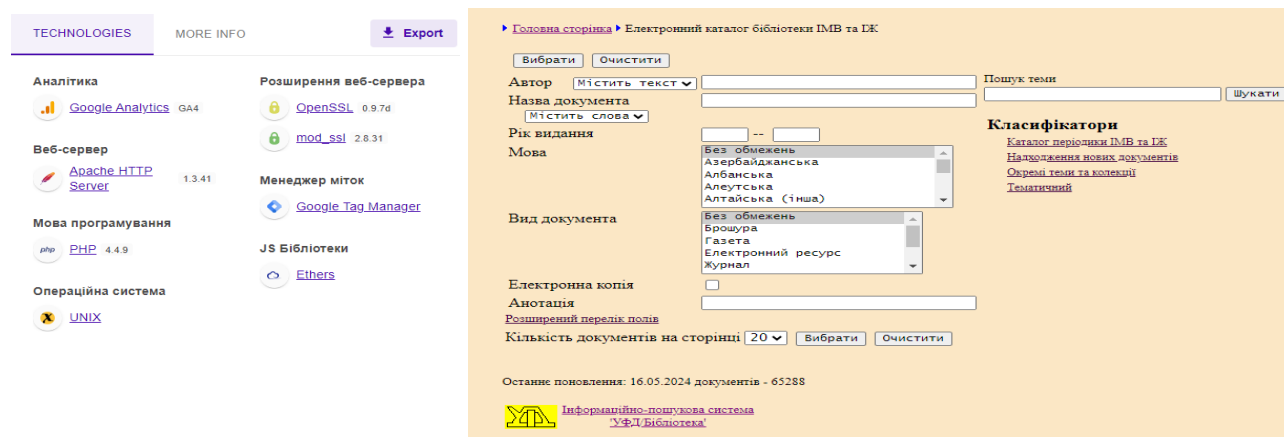


Рис. 1. Аналіз технологій та інтерфейс каталогу бібліотеки КНУ

П 308008
Шевченко, Артем.
 Слов'янськ. Початок війни [Текст] / Артем Шевченко ; [вступна стаття А.Авакова]. —
 Харків : "Фоліо", 2020. — 432 с., [24] с. кольор. фотоіл. : фотоіл. — (Хроніка). — Дар
 Bookforum 28.
 ISBN 978-966-03-9102-4

- Серія
[Хроніка.](#)

- Теми документа

- УДК // Українська література

Примірники		
Місцезнаходження	Кількість	В наявності
Книгозберігання	1	1

Рис. 2. Результат пошуку у бібліотеці ЛНУ

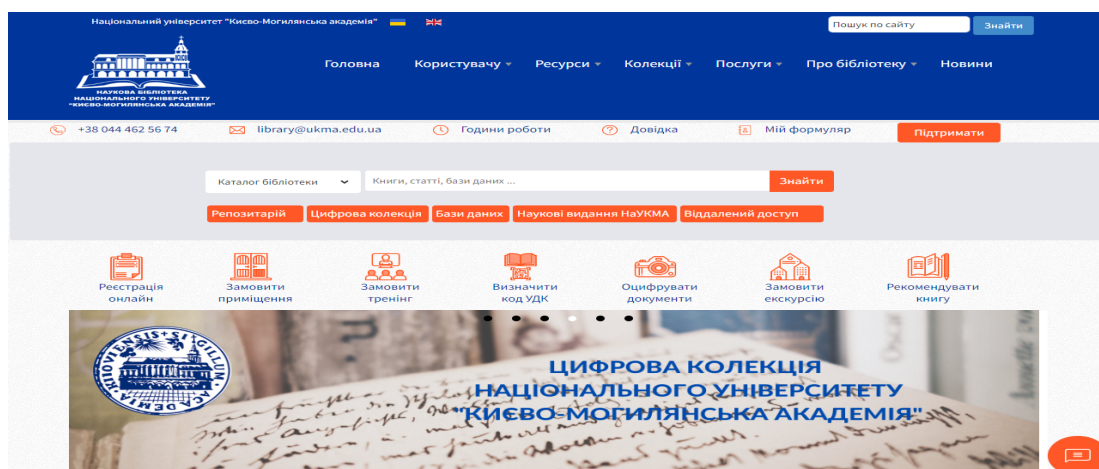


Рис. 3. Інтерфейс наукової бібліотеки НаУКМА

[Загальний каталог]		Електронний каталог Наукової бібліотеки НаУКМА				Формуляр	Опції	Вихід
Перегляд		Матеріали до курсів	Періодика	Ресурси	Нові надходження	Інші каталоги	Варіанти	Допомога
Пошук		Результат пошуку	Пошук	Пошук	Пошук	Пошук	Пошук	Пошук
Відкриті записи :		Перегляд	Відкрити всі, пошук	Сторінки набір	Пошук на "Ново-е-Поліція"	Зберегти на сервері	Зберегти на сервері	Зберегти на сервері
Всі записи:		Відкрити всі	Завантажити	Відкрити	Уточнити	Фільтрувати	Розширити	Розширити

Результати пошуку : Всі слова= Шевченко;
 Відсортовано за : Рік(3)/Автор
 Сортувати за : — Автор(3)/Рік — Назва(3)/Рік — Рік(3)/Автор — Рік(3)/Автор — Автор/Назва
 Формат : — Таблиця — Список — Список коротко (авт., назва) — Список (назва)

Записи 1 - 10 з 3436 (Максимальне відображення: 9000 записів)

#	Автор	Назва	Рік	Джерело/посил	Формат	Бібл./Прим.
1	Сарчин, В.	Хто такий Шевченко?	1911		[Книга]	Б-ка рідкісних вид. (1/ 0)
2	Шевченко, Тарас Григорович,	Кобзарі /	1911		[Книга]	Б-ка рідкісних вид. (1/ 0)
3	Шевченко, Тарас Григорович,	Кобзарі /	1911		[Книга]	Б-ка рідкісних вид. (1/ 0)
4	Задесенський, Р.	Критичні нариси /	1911		[Книга]	Б-ка рідкісних вид. (4/ 0) Філологічна Б-ка (2/ 0)
5	Канадське наукове товариство ім. Т.Г. Шевченка.	Збірник матеріалів наукових конференцій Канадського наукового товариства ім. Т.Г. Шевченка.	1911		[Періодика]	Філологічна Б-ка (8/ 0)
6	Одарченко, Петро Васильович,	Тарас Шевченко і Леся Українка :	1911		[Книга]	Б-ка рідкісних вид. (1/ 0) Філологічна Б-ка (1/ 0)
7	Одарченко, Петро Васильович,	Тарас Шевченко в радянській літературній критичі (1920-1960) :	1911		[Книга]	Філологічна Б-ка (1/ 0)
8	Охринович, Юліан.	Розвиток української національно-політичної думки /	1911		[Книга]	Філологічна Б-ка (1/ 0)
9	Словин, Василь Іванович,	Тарас Шевченко :	1911		[Книга]	Б-ка рідкісних вид. (1/ 0) Філологічна Б-ка (1/ 0)
10	Черкасенко, С. Ф.	Шевченко й діти :	1911		[Книга]	Б-ка рідкісних вид. (1/ 0)

Рис. 4. Застарілий інтерфейс бібліотеки НаУКМА

4. Наукова бібліотека Національного університету біоресурсів і природокористування України [7]. Сайт має достатньо привабливий, стриманий інтерфейс та наявність корисних функцій для користуванням електронним ресурсом (рис. 5).

Інтерфейс відображає весь наявний каталог, що спрощує взаємодію користувача з відображеною літературою, а панелі сортування та вибору категорій наукової літератури додають інтерактивності.

До мінусів в наявній системі можна віднести відсутність кабінету читача, що робить неможливим зберігати свої уподобання, історію пошуку та персональні налаштування, що покращило б зручність використання бібліотеки. Читач не може переглядати свої позики, замовляти та резервувати книги.

5. Наукова бібліотека Київського національного університету технологій та дизайну [8]. Бібліотека пропонує доступ до електронних каталогів, академічних баз даних та різних подій, таких як вебінари і наукові конференції тощо (рис. 6).

Головне меню дає змогу знайти основні розділи бібліотеки, що робить навігацію зрозумілою, але деякі складові підменю не інтуїтивні для нових користувачів і потребують додаткових підказок або інтерактивних елементів, які могли б покращити взаємодію з каталогом. Візуальний дизайн інтерфейсу не відволікає від основного

контенту, але потребує оновлення до відповідності сучасним стандартам.

Проведений аналіз використання цифрових технологій в інформаційних системах наукових бібліотек українських університетів показав, що кожна з них має свої переваги та недоліки, які перешкоджають ефективному використанню цих ресурсів здобувачами та викладачами. Основні проблеми стосуються застарілих технологій (системи використовують застарілі версії мов програмування та ін'єкцій залежностей, що робить їх вразливими до безпекових загроз і ускладнює їхнє подальше оновлення та масштабування), недосконалого користувацького інтерфейсу (інтерфейси бібліотек не відповідають сучасним вимогам UI/UX дизайну [9], що ускладнює навігацію та зменшує зручність використання), обмеження функціональних можливостей (вебсайти наукових бібліотек не надають доступ до електронних версій

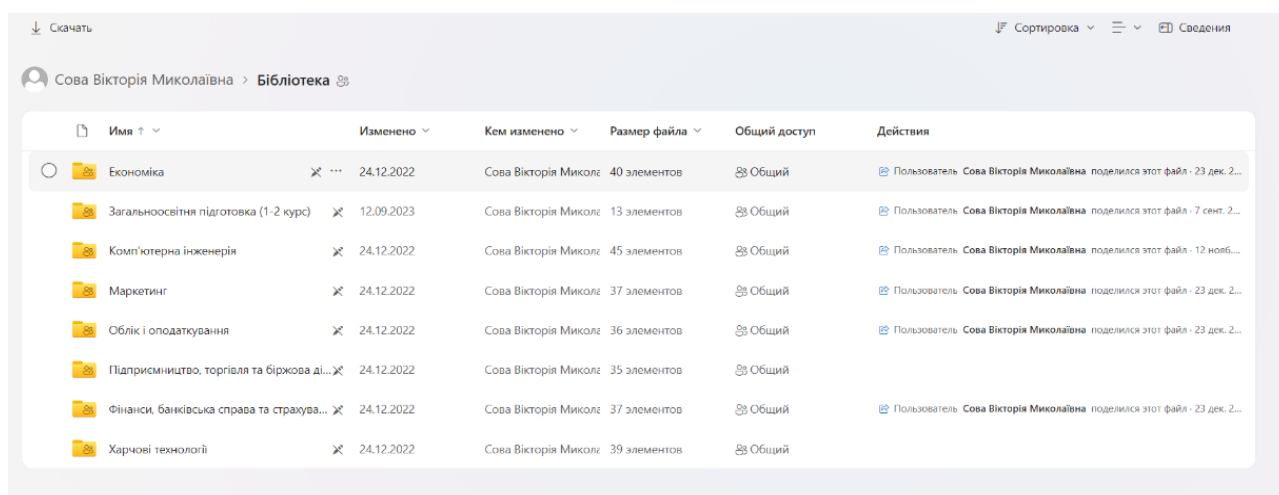


Рис. 5. Інтерфейс електронного каталогу НУБіП



Рис. 6. Інтерфейс електронного каталогу КНУТД

книг), використовуються CMS, які не дозволяють гнучко налаштовувати та масштабувати системи в майбутньому, відсутність адаптації під мобільні пристрої (бібліотеки не надають доступ до електронних версій книг, що є суттєвим недоліком у сучасному цифровому світі) і головне – у жодній з цих наукових бібліотек немає функцій, які б сприяли доступності людей з обмеженими можливостями, що є суттєвим недоліком на сьогодні.

Дослідження існуючих рішень показало, що жодне з них не в повній мірі відповідає критеріям повноцінної функціональності та інформаційної адекватності та виявило кілька проблем, які можна ефективно вирішити за допомогою розробки програмного забезпечення [10].

Постановка завдання. Метою роботи є розроблення програмного забезпечення для віддаленої взаємодії з бібліотекою у вигляді *web*-додатку, який дає змогу ознайомитись з наявними літературними джерелами та можливостями бібліотеки, підвищить швидкість опрацювання матеріалу, зробить систему кросплатформну, ліквідує паперову документацію, що зробить процедуру отримання книг легким та прозорим для всіх читачів.

Створений сервіс у вигляді *web*-додатку, дозволить перебудувати застарілу систему роботи бібліотеки завдяки інтерфейсам бібліотекар-комп'ютер та відвідувач-комп'ютер, привабить більшу кількість відвідувачів та заохотить здобувачів проводити більше часу за книгами. Увага в розробці була приділена інклюзивності та доступності до інформаційних джерел всіх категорій користувачів, забезпечуючи широке охоплення аудиторії, що буде сприяти подальшому розвитку, а можливо й інвестицій в цей проєкт [3].

Для досягнення мети, були вирішені такі завдання:

- аналіз алгоритму взаємодії між читачем та працівниками бібліотеки;
- розробка інтуїтивного та зручного інтерфейсу користувача програмного продукту для людей з обмеженими можливостями;
- вибір оптимального типу архітектури системи;
- вибір технологій та інструментів;
- розробка модулів для забезпечення функціональності бібліотеки, налагодження та тестування системи;
- тестування по зручності користування розробленою інформаційною системою.

Виклад основного матеріалу. Архітектура *web*-додатку включає три основні компоненти: клієнтський компонент, серверний компонент і сервер бази даних. Архітектурна частина додатка реалізована за допомогою використання шаблону *MVC (Model-View-Controller)*, що дозволяє розділити окремо інтерфейс користувача, базу даних (моделі) та логіку [11]. Завдяки цьому гнучкому патернові подальші зміни та розширення проєкту можуть бути здійснені без жодних проблем, а структура додатку стає більш зрозумілою як окремо для *frontend* (розробник клієнтської частини) так і для *backend* (розробник серверної частини додатку) [12].

Інтеграція з серверною частиною додатку є важливим етапом розробки *frontend*. Клієнтська частина дає можливість взаємодіяти з базою даних, обробляти запити користувачів та надавати функціональні можливості, якими користуються на серверному боці. Правильна взаємодія між клієнтської та серверною частинами гарантує ефективне і надійне функціонування всього *web*-додатку.

Крім того, розробка інтерфейсу для *web*-ресурсу включає в себе важливий етап тестування, який дозволяє виявляти та усувати помилки, адаптувати дизайн до різних величин екранів та переконатися, що користувачі можуть легко взаємодіяти з усіма елементами сторінки чи додатку. Загалом, розробка інтерфейсу для *web*-ресурсу є ключовим компонентом сучасних технологій та *web*-розробки. Вона впливає на враження користувачів від використання *web*-додатку [12–13].

Розроблення клієнтської частини *web*-додатку було виконано для Київського фахового коледжу прикладних наук. Технологічний стек обраних технологій представлено на рис. 7.

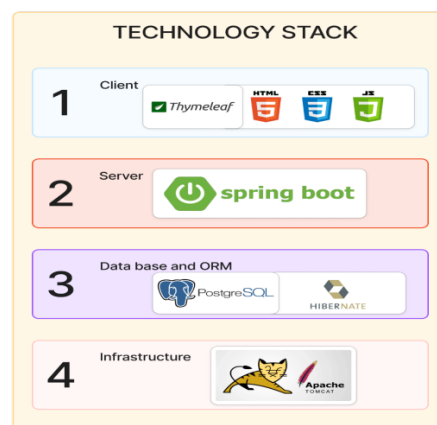


Рис. 7. Технологічний стек обраних технологій

В закладі навчаються здобувачі з вадами слуху, яким слід приділити особливу увагу, адже бібліотека забезпечує їм доступ до інформації, яка може бути недоступною з інших джерел, слід врахувати їхні потреби, що до дизайну та доступності сприйняття жестової мови. Наприклад, бібліотека може надавати послуги з перекладу інформації мовою жестів та використовувати сторонні сервіси для допомоги у користуванні бібліотекою. Використання сучасних цифрових сервісів та мультимедійного контенту для людей з порушенням слуху є невід'ємною складовою при розробці клієнтської частини. На рис. 8 наведено головну сторінку клієнтської частини розробленого додатку.

Забезпечити підтримку мультимедійних ресурсів з субтитрами та жестовою мовою для здобувачів з вадами слуху вдалося за допомогою сервісу УТОГ (Українське товариство глухих), яке надає послуги для людей із вадами слуху.

В розробленні клієнтської частини особливу увагу приділено створенню якісного функціонування та взаємодії між читачем та бібліотекарем. Для цього клієнтську частину *web*-додатку було розділено на дві частини: інтерфейс читача та

Admin-панель бібліотекаря, який здійснює керування системою (рис. 9).

Клієнтська частина була реалізована за допомогою системи для побудови шаблонів та сторінок мовою розмітки *HTML* та стилізована за допомогою фреймворку *Bootstrap* та *CSS* та *JS* (*JavaScript*). *Bootstrap* – це клієнтський фреймворк з безкоштовним набором інструментів з відкритим кодом, призначений для створення вебсайтів та вебдодатків, який містить шаблони *CSS* та *HTML* для типографіки, форм, кнопок, навігації та інших компонентів інтерфейсу, а також додаткові розширення *JavaScript*. Він спрощує розробку динамічних вебсайтів і вебдодатків [13].

Для динамічної генерації *HTML*-шаблонів було використано *Thymeleaf* – сучасний серверний шаблонізатор для *Java*, який дозволяє зручно створювати *HTML*-сторінки, використовуючи природний синтаксис *HTML*. Він легко інтегрується з *Spring Boot*, що робить його ідеальним вибором для створення вебдодатків [14].

Система має 2 типи користувачів, а отже і 2 види доступу до даних *admin* та *simple user*. До категорії *simple user* також відносяться і викладачі, оскільки в них немає привілеїв порівняно зі здобувачами, це зроблено лише для ідентифікування користувачів, як викладачів.

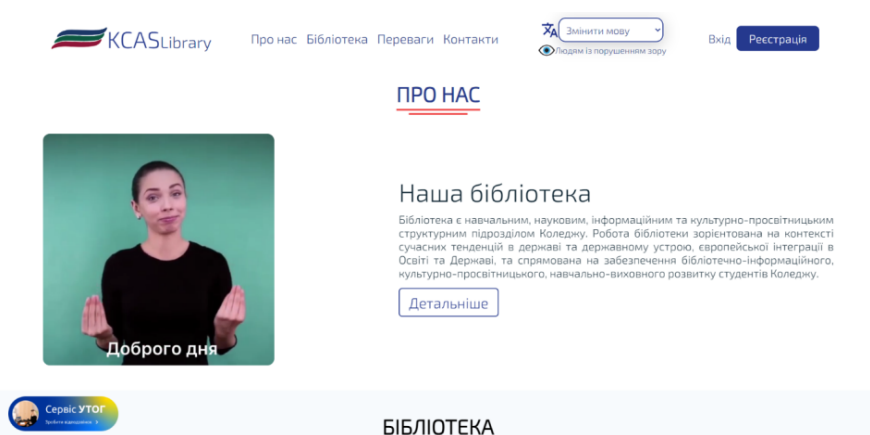


Рис. 8. Мультимедійний контент для людей з порушенням слуху

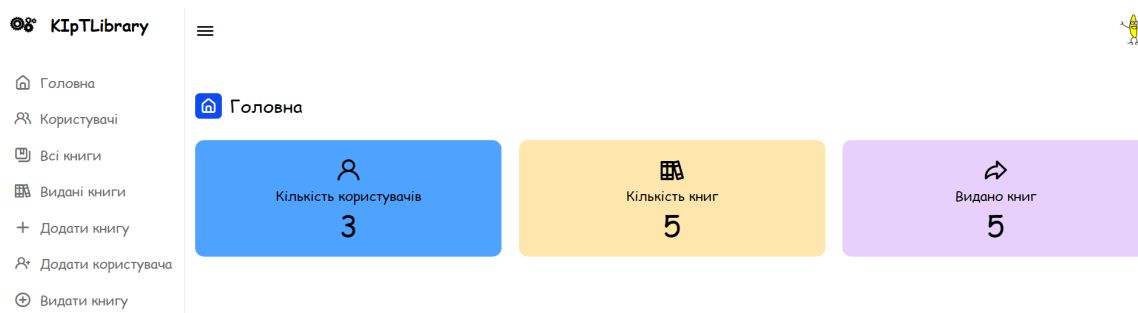


Рис. 9. *Admin*-панель бібліотекаря

В проєкті реалізована генерація *QR*-коду для кожної книги, за допомогою якого читач може отримати завчасно додану до своїх улюблених книжку (рис. 10).

Інтерфейс дає змогу адміністратору за допомогою розроблених форм (рис. 11) додавати в базу електронного каталогу нову літературу. Слід зауважити, що система має 2 основні категорії книг: паперові та електронні.

Адміністратор бібліотеки отримавши замовлення у *Admin*-панелі (рис. 9) може заздалегідь підготувати книгу для видачі. Для видачі книги без замовлення бібліотекарю потрібно обрати користувача та обрати книгу для видачі (рис. 12). Для спрощення пошуку було додано поле для пошуку користувачів та книги за ключовими словами, а також за *id QR*-коду.

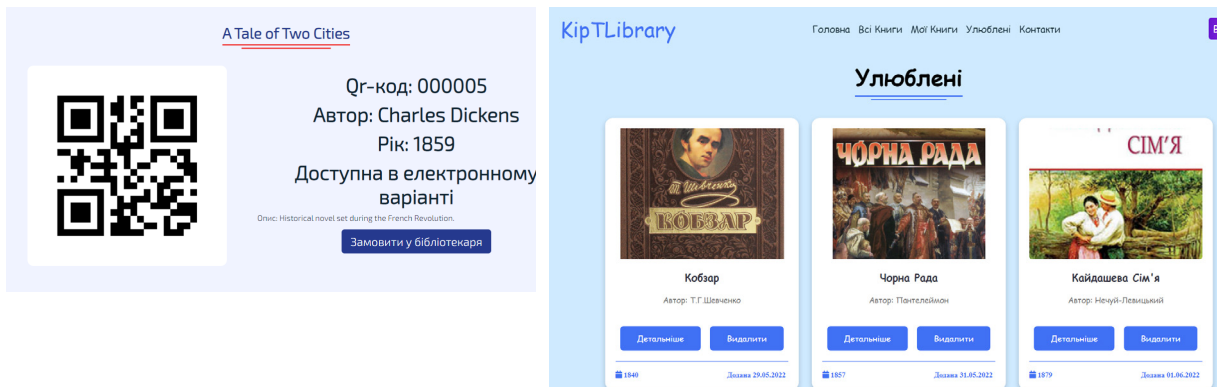


Рис. 10. Обрана книга для замовлення за допомогою *QR*-коду та улюблені книжки користувача

Рис. 11. Форми додавання в базу електронного каталогу нової літератури

Id qr-коду	Фото	Назва	Автор	Рік	Номер книги	Дії
QR1984		1984	George Orwell	1949	Електронна	Видати книгу
000005		A Tale of Two Cities	Charles Dickens	1859	Електронна	Видати книгу
QRHuckFinn		Adventures of Huckleberry Finn	Mark Twain	1884	Електронна	Видати книгу

Рис. 12. Вибір книги для користувача

Окрім адміністрування та управління основним функціоналом електронного каталогу *admin* має змогу створювати розважальний та науково-дослідницький контент розміщений на головній сторінці *web*-додатку.

Створення інтерфейсу для *Admin*-панелі забезпечує бібліотекарям інтуїтивно зрозумілий та функціональний інструмент для щоденної роботи і може повністю замінити паперові формуляри у бібліотеках навчальних закладів.

Для реалізації серверної частини нашого проєкту було використано мову програмування *Java* та фреймворк *Spring Boot* з використанням *Java* бібліотек [15]. В якості середовища розробки було проаналізовано та обрано інтегроване середовище розробки (*IDE*)-*IntelliJ IDEA*.

Специфікування предметної галузі проєкту наведено на рис. 13, де зображено розроблювані класи і зв'язки між ними.

Використання зв'язку *ManyToMany* дає змогу встановити зв'язок «багато-до-багатьох» між двома об'єктами, як у випадку класів «Книга» та «Категорії книг», оскільки одна книга може бути у декількох категоріях одночасно, а також одна категорія може мати декілька книг. Зв'язок *ManyToOne* вказує на те, що між двома

об'єктами існує відношення «багато-до-одного», як у випадку класів «Книга» та «Автор», бо кожна книга може мати лише одного автора, але у одного автора може бути багато книг. Зв'язок *OneToOne* вказує на те, що об'єкт може бути пов'язаний лише з одним екземпляром іншого об'єкта, і навпаки.

Для управління базою даних було обрано *PostgreSQL*, що є потужною системою управління реляційними базами даних з відкритим вихідним кодом, яка підтримує стандарт *SQL*. Для *ORM (Object-Relational Mapping)* був обраний *Hibernate*, який забезпечує зручний спосіб взаємодії з базою даних через об'єкти *Java Hibernate*, *Spring Data JPA*, які є тісно пов'язаними технологіями, що використовуються для роботи з базами даних у *Java*-додатках [16].

Серверна частина в нашому проєкті містить 14 класів, 8 сервісів та 14 репозиторіїв. Кожен з класів має анотацію *@Entity*, яка є частиною пакету *javax.persistence* та використовується на рівні класу і позначає клас як персистентну сутність. Вона сигналізує провайдеру *JPA*, що клас слід розглядати як таблицю в базі даних.

Для коректної роботи *JPA* додана анотація *@Id*, яка використовується для позначення пер-

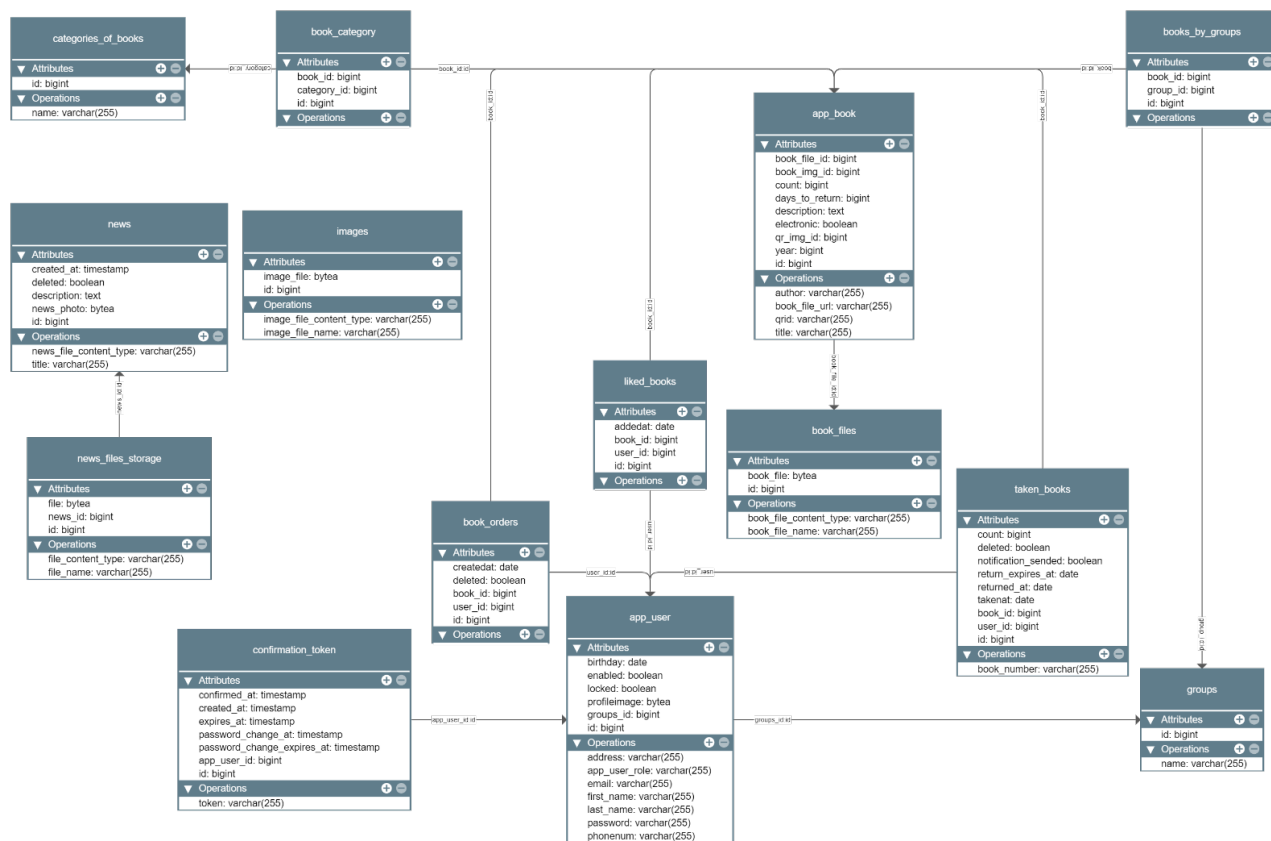


Рис. 13. Діаграма класів

винного ключа, який однозначно ідентифікує кожен екземпляр в базі даних та застосовується до поля або властивості в класі.

Точки доступу «*endpoint*», через які клієнти можуть взаємодіяти з сервером, допомагають серверу зрозуміти куди клієнт відправив запит, а контролер відповідає за обробку цього запиту та за відправку відповіді. Над кожним методом, що обробляє запит буде анотація з методом запиту, наприклад *@GetMapping* для *Get*-запиту, який може повертати різні типи даних, включаючи рядки, об'єкти, тощо та *@PostMapping*, який зазвичай приймає дані (можуть бути в різних форматах) з тіла запиту.

Наведемо приклад, де використовується «*endpoint*» за шляхом «/» для розроблення головної сторінки. В контролері створюємо *Get*-метод, та передаємо в нього дані, які міститимуться на головній сторінці.

```
@GetMapping("/")
public String home(Model model) {
    Page<NewsRepository.NewsNoFile> news =
        newsRepository.findAllByDeletedIsFalse(PageRequest.of(
            0, 5, Sort.Direction.DESC, «createdAt»));
    model.addAttribute(«news», news);
    return «home»;
}
```

В результаті отримуємо *html* сторінку (рис. 14) з атрибутом новин, який оброблює *Thymeleaf*.

За аналогією з головною сторінкою будемо метод у контролері для сторінки реєстрації:

```
@GetMapping(«/register»)
public String showRegistrationForm(Model model) {
    List<Groups> groups = groupsRepository.findAll();
    model.addAttribute(«groups», groups);
    model.addAttribute(«user», new AppUser());
    return «signup_form»;
}
```

В результаті отримуємо *html* сторінку (рис. 15) з атрибутами груп та атрибутом нового користувача, які також будуть оброблені за допомогою *Thymeleaf*.

Рис. 15. Реєстрація нового користувача

Після заповнення форми система відправляє ці дані на «*endpoint*» «*/process_register*», сервер отримавши запит обробляє його, перевіряє дані та додає їх до бази даних, після чого надсилає *email* для підтвердження пошти.

Підтвердивши пошту користувач має змогу зайти в свій електронний кабінет який знаходиться на «*endpoint*» «*/editprofile*» (рис. 16).

За аналогією будуються також інші контролери, наведені на рис. 17. Контролери, які починаються з слова */admin* містять методи з «*endpoints*», що допомагає у розробці безпеки.

Функціонал доступний користувачам: перегляд всіх наявних книг в бібліотеці та доступ до електронних книг, можливість вибирати книги за різними фільтрами, позначати «улюблені» та бронювати бажані. Крім цього, є можливість брати

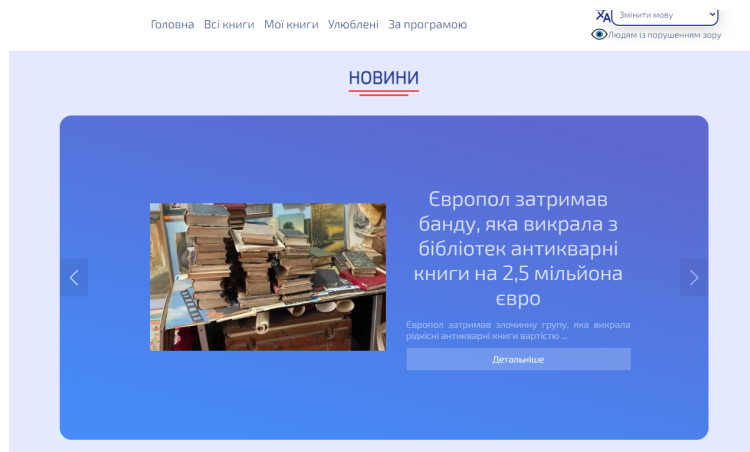


Рис. 14. Сторінка новин

Головна Всі книги Мої книги Улюблені За програмою

Змінити мову
Людям із порушенням зору



Редагувати зображення
Вибрати файл Файл не вибрано

Email	Пароль
lantuh.sasha@gmail.com	Зміна паролю
Ім'я	Прізвище
Oleksandr	Lantukh
Група	Дата народження
БІТск-22	02.02.2001
Домашня адреса	Телефон
Тараса Шевченка 23	0969023920

Зберегти

Рис. 16. Профіль користувача (електронний кабінет)

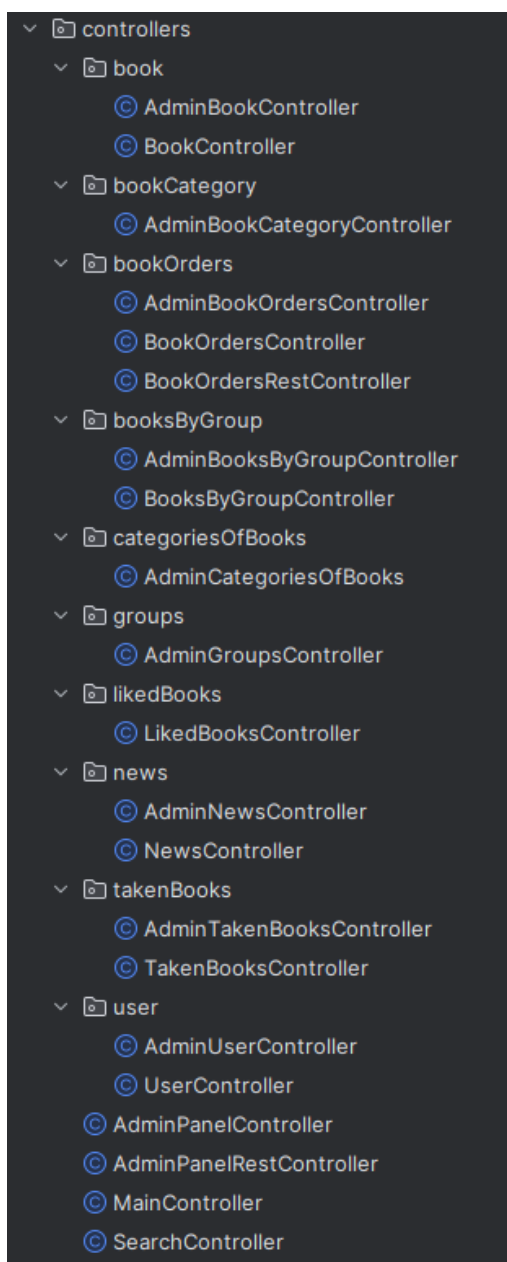


Рис. 17. Контролери проекту

участь в онлайн марафонах та заходах, які організовує бібліотека разом з партнерами, виступах тренерів, та гостей. *Web*-додаток розміщений на хмаровому сервісі *Microsoft Azure*, що дозволяє забезпечити якісну роботу сайту в будь-якому місці та в будь-який час.

Висновки. Розроблений програмний продукт є кроком у напрямку модернізації освітнього процесу в Україні. Ця система забезпечує зручний та доступний інструмент для отримання навчальних матеріалів як здобувачам, так і викладачам.

Розроблений програмний продукт дозволяє збільшити доступність навчальних ресурсів, підвищити якість освіти через різноманітність форматів матеріалів та стимулювати самостійне навчання. План розроблення програмного продукту, який включав етапи проектування, розробки, тестування та впровадження, відображає системний та професійний підхід до створення інтерактивного додатку для взаємодії з бібліотекою закладу освіти за всіма сучасними стандартами. Заплановані мікросервіси всередині проекту дозволяють створити різноманітне та комфортне програмне забезпечення для здобувачів та викладачів, що дозволить навчатися їм більш продуктивно та комфортно. Аналітична робота дозволила визначити функціональні вимоги та особливості, необхідні для створення *web*-додатку для онлайн бібліотеки закладів освіти. Вибір правильних технологій та інструментів дозволить легко підтримувати та розширювати функціонал в майбутньому, вдосконалити архітектуру до багаторівневої, сприяє ефективній розробці, а наведені етапи розробки дозволили досягти успішного завершення проекту. Програмний продукт може бути використаний для автоматизації роботи бібліотеки будь-якого закладу освіти або іншої організації.

Розроблений проєкт пройшов апробацію та використовується в Київському фаховому коледжу прикладних наук. Використання даної системи зменшує часові витрати користувача та зменшує навантаження на бібліотекаря, забез-

печує підтримку мультимедійних ресурсів з субтитрами та жестовою мовою для здобувачів з вадами слуху, ліквідує паперову документацію та робить процедуру отримання книг легкою та прозорою.

Список літератури:

1. Моргулець О.Б., Астістова Т.І. Сучасні комп'ютерні технології в організації освітньої діяльності навчального закладу. *Вісник КНУТД*. Київ, 2016. № 5 (101), спецвипуск. С. 351–359.
2. Астістова Т.І. Розробка системи керування розкладом занять у вищому навчальному закладі. *Вісник КНУТД*. Київ, 2016. № 4 (100). С. 13–18.
3. Smorzhevskiy N.V, Astistova T.I. Development of software for interactive interaction with the library. *Інноватика в освіті, науці та бізнесі: виклики та можливості* : збірник наукових праць 1 Всеукраїнської конференції, м. Київ, 17 листопада 2020 р. Київ, 2020. С. 111–119.
4. Наукова бібліотека ім. М. Максимовича. URL: <https://library.knu.ua/>
5. Наукова бібліотека ЛНУ імені Івана Франка. URL: <https://www.lnulibrary.lviv.ua/>
6. Наукова бібліотека Національного університету «Києво-Могилянська академія». URL: <https://library.ukma.edu.ua/>
7. Наукова бібліотека Національного університету біоресурсів і природокористування України. URL: <https://nubip.edu.ua/structure/library>
8. Наукова бібліотека Київського національного університету технологій та дизайну. URL: <http://lib.knutd.edu.ua/>
9. Веб-дизайн, UI/UX, ефективні інтерфейси. URL: <https://goldwebsolutions.com/uk/uiux-uk/>
10. Дроменко В.Б., Лісовець С.М. Особливості інженерії програмного забезпечення при розробленні систем з елементами штучного інтелекту. *Вчені записки Таврійського Національного Університету імені В.І. Вернадського. Серія: Технічні науки*. 2024. Т. 35 (74), № 4. С. 74–82. DOI: 10.32782/2663-5941/2024.4/12
11. Клієнт-серверна архітектура. URL: <https://surli.li/cppxry>
12. Frontend. Інтерфейс для взаємодії між користувачем і backend. URL: <https://goldwebsolutions.com/uk/blog/frontend/>
13. Bootstrap. URL: <https://surli.cc/uivmrr>
14. Thymeleaf: Документація Thymeleaf. URL: <https://www.thymeleaf.org/doc/tutorials/3.0/usingthymeleaf.html>
15. Лантух О.М., Астістова Т.І. Розробка серверної частини для веб-додатку *Мехатронні системи: інновації та інжиніринг* : зб. тез доп. VII Міжнародної науково-практичної конференції, м. Київ, 23 листопада 2023 р. Київ, 2023. С. 165.
16. Java: Офіційна документація Oracle. URL: <https://docs.oracle.com/javase/tutorial/>

Astistova T.I., Dromenko V.B., Kalashnyk V.Yu. DEVELOPMENT OF SOFTWARE FOR INFORMATION RESOURCES OF EDUCATIONAL INSTITUTIONS LIBRARY

The article presents an analysis of the use of digital technologies in the information systems of scientific libraries of leading Ukrainian universities. On the basis of the analysis, the functional requirements and tasks for developing an interactive application for interaction with the library of an educational institution according to modern standards are defined.

The architecture of the developed web application contains three main components: a client component, a server component, and a database server, which is implemented using the MVC (Model-View-Controller) pattern.

The client side was implemented using the integrated software development environment IntelliJ Idea and a system for building templates and pages in the HTML markup language and stylised using the Bootstrap framework and CSS and JS. For the dynamic generation of HTML templates, we used a modern server-side templating engine for Java, Thymeleaf. To implement the server side, we used the Java programming language and the SpringBoot framework with Java libraries. PostgreSQL was chosen for database management. For ORM (Object-Relational Mapping), Hibernate was chosen, which provides a convenient way to interact with the database through Java Hibernate objects, Spring Data JPA.

The developed web application is a complete software product that has been tested and is used at the Kyiv Professional College of Applied Sciences, and provides support for multimedia resources with subtitles and sign language for students with hearing impairments. The project implemented the generation of a QR code for each book, with which the reader can receive a book added to their favorites in advance. The implemented system reduces user time and librarian workload, eliminates paperwork, and makes the procedure for obtaining books easy and transparent.

Key words: *architecture, services, methods, classes, controllers, technology stack, MVC, PostgreSQL, integrated environment, Integrated IDE, java Spring Boot framework, HTML, CSS.*