

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційні управляючі системи
та технології»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Автоматизована система каталогізації та класифікації
електронних видань у бібліотеці»**

Виконав (-ла):

студент (-ка) IV курсу, групи ІС-91

Ковальська Анастасія Анатоліївна _____

Керівник:

доцент кафедри ІСТ, кандидат технічних наук,

Ульяницька Ксенія Олександрівна _____

Рецензент:

доцент кафедри ІП, кандидат технічних наук,

Лісовиченко Олег Іванович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент (-ка) _____

Київ – 2023 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Ковальській Анастасії Анатоліївні

1. Тема проєкту «Автоматизована система каталогізації та класифікації електронних видань у бібліотеці», керівник проєкту Ульяницька Ксенія Олександрівна, к.т.н., доцент, затверджені наказом по університету від «31» травня 2023 р. № 2101-с

2. Термін подання студентом проєкту: «12» червня 2023 року

3. Вихідні дані до проєкту: мова програмування Python, редактор вихідного коду PyCharm, бібліотека Pandas, бібліотека Numpy, бібліотека Scikit-learn бібліотека TensorFlow

4. Зміст пояснювальної записки:

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд наявних аналогів, постановка задачі;

2. Інформаційне забезпечення: вхідні та вихідні данні, опис структури бази даних;

3. Математичне забезпечення: змістова та математична постановка задачі, обґрунтування та опис методів розв'язання;

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів;

5. Технологічний розділ: керівництво користувача, методика випробувань.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

1. Діаграми діяльності

2. Схеми варіантів використання

3. Схеми бази даних

4. Діаграми послідовностей

6. Дата видачі завдання: «13» квітня 2023

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Аналіз предметної області	19 квітня 2023	
2	Огляд наявних аналогів	21 квітня 2021	
3	Постановка та формалізація задачі	25 квітня 2023	
4	Розробка та опис структури бази даних	30 квітня 2023	
5	Опис математичного забезпечення	7 травня 2023	
6	Розробка та опис технічного рішення	18 травня 2023	
7	Тестування готового проєкту	27 травня 2023	
8	Опис керівництва користувача	28 травня 2023	
9	Подання ДП на попередній захист	29 травня 2023	
10	Подання ДП на основний захист	12 червня 2023	

Студент

Анастасія КОВАЛЬСЬКА

Керівник

Ксенія УЛЬЯНИЦЬКА

АННОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проєкту складається з 5 розділів, містить 16 рисунків, 12 таблиць, 1 додаток, 4 кресленика та посилання на 15 літературних джерел.

Ключові слова: автоматизована система, каталогізація, класифікація, електронні видання, TextRank, Наївний Баєвський класифікатор.

Метою дипломного проєкту є створення автоматизованої системи каталогізації та класифікації, що дозволить бібліотекарям ефективно керувати електронними виданнями бібліотеки, забезпечувати зручний доступ до цих ресурсів та зменшувати час та зусилля, які необхідні для їх управління.

У розділі загальні положення описано бізнес-процеси бібліотеки та процес діяльності, визначено варіанти використання та відповідні функціональні вимоги, а також визначено переваги та недоліки існуючих аналогів системи та відмінності їх від системи, що розробляється. Розділ інформаційного забезпечення присвячено опису вхідних та вихідних даних системи, проектуванню бази даних. У розділі математичного забезпечення обґрунтовано вибір методів розв'язання задачі та описано відповідні алгоритми. Розділ програмного та технічного забезпечення присвячено опису засобів розробки автоматизованої системи та архітектури проєкту. У технологічному розділі визначено та описано випробування автоматизованої системи та наведено їх результати.

Розроблена автоматизована система каталогізації та класифікації електронних видань може використовуватись як прототип для інтеграції в існуючі бібліотечні системи управління.

ABSTRACT

Structure and scope of the work. The explanatory report of the diploma project consists of 5 chapters, 16 figures, 12 tables, 1 application, 4 drawings and references to 15 literary sources.

Keywords: automated system, cataloguing, classification, electronic publications, TextRank, Naive Bayes classifier.

The aim of the diploma project is to create an automated cataloguing and classification system that will allow librarians to effectively manage the library's electronic publications, provide convenient access to these resources, and reduce the time and effort required to manage them.

The general framework chapter describes the library's business processes and activities, identifies the options for use and the corresponding functional requirements, and identifies the advantages and disadvantages of existing analogues of the system and their differences from the system under development. The chapter on information support is devoted to the description of input and output data of the system, and the design of the database. The chapter on mathematical support substantiates the choice of methods for solving the problem and describes the relevant algorithms. The software and hardware chapter describes the development tools for the automated system and the project architecture. The technological chapter identifies and describes the tests of the automated system and presents their results.

The developed automated system for cataloguing and classifying electronic publications can be used as a prototype for integration into existing library management systems.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IC91.080БАК.004 ПЗ	Пояснювальна записка	60		
6	A3	IC91.080БАК.004 Д1	Автоматизована система	1		
7			каталогізації та класифікації			
8			електронних видань у			
9			бібліотеці. Діаграми			
10			діяльності			
11	A3	IC91.140БАК.004 Д2	Автоматизована система	1		
12			каталогізації та класифікації			
13			електронних видань у			
14			бібліотеці. Схеми варіантів			
15			використання			
16	A3	IC91.140БАК.004 Д3	Автоматизована система	1		
17			каталогізації та класифікації			
18			електронних видань у			
19			бібліотеці. Схема бази даних			
20	A3	IC91.140БАК.004 Д4	Автоматизована система	1		
21			каталогізації та класифікації			
22			електронних видань у			
23			бібліотеці. Діаграми			
24			послідовностей			
25						
26						
27						
28						
Зм.	Аркуш	№ докум.	Підпис	Дата	IC91.080БАК.004 ТП	
Розроб.		Ковальська А.А.				
Керівн.		Ульяницька К.О.			Автоматизована система каталогізації та класифікації електронних видань у бібліотеці. Відомість проекту Літ. T Аркуш 1 Аркушів 1 КПІ ім. Ігоря Сікорського Група IC-91	

**Пояснювальна записка
до дипломного проєкту
на тему: «Автоматизована система каталогізації та
класифікації електронних видань у бібліотеці»**

Київ – 2023 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ	4
ВСТУП	5
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	7
1.1 Опис предметного середовища	7
1.1.1 Опис бізнес-процесів	7
1.1.2 Опис процесу діяльності	8
1.1.3 Опис функціональної моделі	10
1.2 Огляд наявних аналогів	15
1.3 Постановка задачі	17
1.3.1 Призначення розробки.....	17
1.3.2 Цілі та задачі розробки	17
Висновки до розділу.....	18
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	19
2.1 Вхідні дані	19
2.2 Вихідні дані	19
2.3 Опис структури бази даних	20
Висновки до розділу.....	25
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	26
3.1 Змістовна постановка задачі.....	26
3.2 Математична постановка задачі.....	26
3.3 Обґрунтування методів розв'язання	27

					IC91.080БАК.004 ПЗ						
Зм.	Лист	№ докум.	Підпис		Автоматизована система каталогізації та класифікації електронних видань у бібліотеці. Пояснювальна записка						
Розробив	Ковальська А.А.										
Перевірів	Ульяницька К.О.										
					Літ.	Арк.	Аркушів				
						2	60				
					КПІ ім. Ігоря Сікорського Група IC-91						

3.3.1	Методи визначення ключових слів	27
3.3.2	Методи визначення категорій.....	28
3.4	Опис методу розв’язання	31
	Висновки до розділу.....	33
4	ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	34
4.1	Засоби розробки.....	34
4.2	Загальні вимоги до технічного забезпечення	37
4.3	Архітектура програмного забезпечення	38
4.3.1	Схема архітектури програмного забезпечення	38
4.3.2	Діаграма послідовностей.....	40
4.3.3	Діаграма розгортання	40
	Висновки до розділу.....	42
5	ТЕХНОЛОГІЧНИЙ РОЗДІЛ	43
5.1	Керівництво користувача.....	43
5.2	Випробування програмного продукту	51
5.2.1	Мета випробувань	51
5.2.2	Результати випробувань	51
	Висновки до розділу.....	55
	ВИСНОВКИ.....	57
	ПЕРЕЛІК ПОСИЛАНЬ	59

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

Фреймворк – це структурована платформа, що надає розробникам програмного забезпечення готовий набір інструментів, що спрощують розробку, а також надає загальну структуру, архітектуру та правила розробки для програмного забезпечення.

СКБД – система керування базами даних.

ОС – операційна система.

Тест-кейс – сукупність кроків, які описують конкретний тестовий сценарій або тестовий випадок для перевірки певної функціональності або властивостей системи.

					ІС91.080БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			4

ВСТУП

Актуальність проблеми. Актуальність проблеми зумовлена сучасними тенденціями у розвитку інформаційних технологій та зростанням обсягу електронних ресурсів, доступних для користувачів. Завдяки швидкому розвитку Інтернету та цифрових технологій, електронні видання стають все більш популярними та доступними для широкої аудиторії.

Однак, це також призводить до виникнення проблеми організації та ефективного використання цих ресурсів у бібліотеках. Традиційні методи каталогізації та класифікації, які були розроблені для друкованих видань, не завжди ефективно працюють з електронними матеріалами. Виникає потреба у спеціалізованій системі, яка дозволить автоматизувати ці процеси та забезпечити швидкий та зручний доступ до електронних видань.

Об'єкт дослідження. Об'єктом дослідження є автоматизована система каталогізації та класифікації електронних видань у бібліотеці.

Предмет дослідження. Методи та алгоритми як засоби розробки автоматизованої системи каталогізації та класифікації електронних видань у бібліотеці.

Мета і завдання. Метою даного бакалаврського дипломного проекту є розробка автоматизованої системи каталогізації та класифікації електронних видань в бібліотеці. Для досягнення поставленої мети необхідно вирішити наступні завдання:

- 1) Описати предметну область, включаючи процеси та функціональну модель системи;
- 2) Провести огляд існуючих аналогів та виявити їх недоліки та переваги;
- 3) Сформулювати задачу розробки системи та визначити її призначення, цілі та задачі;
- 4) Розробити інформаційне забезпечення, включаючи вхідні та вихідні дані, а також опис структури бази даних. Розробити математичне забезпечення для

каталогізації та класифікації електронних видань, включаючи методи визначення ключових слів та категорій;

5) Розробити програмне та технічне забезпечення, включаючи вимоги до технічного середовища та архітектуру програмного забезпечення;

6) Провести випробування програмного продукту та надати керівництво користувача;

7) Зробити висновки про результати роботи та вказати практичне значення одержаних результатів.

Практичне значення. Розроблена автоматизована система може використовуватись як прототип для інтеграції в існуючі бібліотечні системи управління. Це дозволить розширити функціональні можливості бібліотек в галузі обробки та організації електронних ресурсів. Система може використовуватись також в освітніх установах, дослідницьких центрах та інших організаціях, які працюють з великим обсягом електронної інформації.

					ІС91.080БАК.004 ПЗ	Арк.
						6
Зм.	Лист	№ докум.	Підпис			

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Сучасний світ, який напряду пов'язаний з Інтернетом та технологіями, диктує свої правила стосовно отримання інформації. І хоча традиційні паперові носії інформації ще є досі актуальним способом отримання знань, однак їх все більше витісняють електронні носії. Електронні видання стають все більш доступними для широкої аудиторії, розширюючи можливості отримання інформації.

Тому замість традиційних бібліотек виникає необхідність в онлайн-бібліотеках, які забезпечують доступ до електронних носіїв інформації. Цифрова бібліотека спрощує пошук необхідних електронних ресурсів, полегшує процес навчання та отримання потрібної інформації.

Однією з важливих проблем, з якою почали стикатися бібліотекарі при переході від традиційного формату на електронний – необхідність ефективної класифікації та каталогізації електронних видань.

Процес каталогізації забезпечує створення бібліографічних записів для кожного електронного ресурсу, а процес класифікації в свою чергу забезпечує надання цим ресурсам відповідних категорій.

Якщо вирішити цю проблему, то це дозволить не тільки полегшити організацію ресурсів та зробити її більш систематизованою, а й забезпечить простий та зручний пошук необхідної інформації та підвищить якість послуг, які надає бібліотека.

1.1.1 Опис бізнес-процесів

Бізнес-процеси бібліотеки - це послідовність дій та операцій, які забезпечують функціонування бібліотечної системи та виконання основних завдань. Вони описують ключові діяльності та процедури, які відбуваються у бібліотеці для надання послуг користувачам.

					ІС91.080БАК.004 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис			

До бізнес-процесів даної автоматизованої системи можна віднести наступне:

1) Процес створення облікових записів користувачів.

Процес охоплює реєстрацію нових користувачів у системі адміністратором. Він включає створення нового облікового запису, надання прав, перевірка та збереження даних;

2) Процес каталогізації електронних видань.

Процес включає в себе каталогізацію нових ресурсів, таких як журнал, стаття, книга тощо. Він включає визначення бібліографічних даних видання, надання йому ключових слів. Це в свою чергу спрощує організацію та пошук видання за необхідністю;

3) Процес класифікації електронних видань.

Процес включає в себе визначення каталогу за тематикою, до якого відноситься електронне видання та надання унікального ідентифікатора ресурсу. Також цей процес включає в собі збереження даних у системі;

4) Процес організації ресурсів у бібліотеці.

Процес охоплює не тільки структуроване зберігання електронних видань, що полегшує пошук, а й контроль стану видання, тобто перевірка актуальності бібліографічних даних видання, перевірка правильності віднесення до певного каталогу тощо.

Цей перелік бізнес-процесів відображає основні діяльності, які відбуваються у бібліотеці та необхідні для забезпечення ефективного функціонування системи.

1.1.2 Опис процесу діяльності

Для того, щоб вирішити проблему автоматизації процесу каталогізації та класифікації електронних видань, необхідно побудувати систему, яка дасть змогу бібліотекарям отримувати доступ до системи та повністю користуватися її ресурсами. Адміністратор в свою чергу буде контролювати та керувати цим процесом.

Відповідні діаграми діяльності наведені у графічних матеріалах.

Опишемо процес надання бібліотекарю доступу до функціоналу системи. Як

					ІС91.080БАК.004 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис			

бачимо, то для того, щоб надати бібліотекарю змогу користуватись функціоналом системи, адміністратору спочатку треба створити для нього обліковий запис, тобто надати унікальний логін та пароль. Потім необхідно надати права доступу, а саме надати можливість бібліотекарю додавати, редагувати та видаляти електронні видання. Після цього, всі необхідні дані зберігаються у системі і користувач, у нашому випадку бібліотекар, може авторизуватись та користуватись функціоналом системи. Якщо спроба авторизації невдала, то надається можливість ввести логін та пароль ще раз. Адміністратору надається інформація, коли користувач-бібліотекар авторизується у системі.

Тепер розглянемо процес отримання нового видання та додавання його до бази даних.

Спочатку у бібліотекаря перевіряються права на додання електронного видання, якщо такі є то, бібліотекар завантажує у систему електронний ресурс та заповнює бібліографічні данні про видання, такі як назва, автор тощо. Потім система опрацьовує цей текст видання, в результаті чого визначає ключові слова та категорію, до якої можна віднести це видання.

Бібліотекар перевіряє отриманий результат і якщо він з ним згоден, то зберігає його. Система перед тим як зберегти видання у базі даних спочатку перевіряє на наявність дублікатів. Якщо такі наявні, то генерує повідомлення про рішення цієї проблеми: або змінити дані про видання, або відмінити його додавання. В залежності від вибору бібліотекаря виконується або оновлення даних, або закінчення процесу.

У випадку, коли такого видання немає у базі даних, електронному ресурсу надається унікальний ідентифікатор та він зберігається у базі даних. Далі генерується звіт про додане електронне видання і відправляється адміністратору, а бібліотекар у свою чергу може переглянути видання у каталозі.

1.1.3 Опис функціональної моделі

У системі наступні дійові особи:

1. Бібліотекар – основний користувач системи, який відповідає за збір, опис та класифікацію електронних видань;

2. Адміністратор – користувач, який має розширені можливості, такі як надання прав доступу бібліотекарям, налаштування баз даних.

Розглянемо які варіанти використання системи акторами у таблиці 1.1.

Таблиця 1.1 – Варіанти використання

Актор	Варіант використання	Опис дії варіанта використання
Усі	Авторизація у системі	Усі мають можливість авторизуватися у системі.
Адміністратор	Управління користувачами системи	Адміністратор може створювати, редагувати та видаляти облікові записи користувачів, встановлювати рівні доступу.
Адміністратор	Ведення бази даних бібліотеки	Адміністратор може додавати, видаляти, редагувати всі елементи у базі даних електронних видань.
Бібліотекар	Каталогізація електронних видань	Бібліотекар може створювати, редагувати бібліографічні дані про електронне видання та використовувати систему для визначення ключових слів.
Бібліотекар	Класифікація електронних видань	Бібліотекар може використовувати систему для автоматичної класифікації та самостійно класифікувати видання.

Продовження таблиці 1.1

Актор	Варіант використання	Опис дії варіанта використання
Бібліотекар	Управління електронними виданнями	Бібліотекар має можливість додавати нове та управляти існуючими електронними виданнями.
Бібліотекар	Управління відомостями про авторів електронних видань	Бібліотекар має можливість створювати нових авторів, редагувати відомості про існуючих в системи авторів.
Бібліотекар	Управління каталогами електронних видань	Бібліотекар має можливість управляти каталогами видань, а саме редагувати відомості про них та їх вміст.
Бібліотекар	Моніторинг стану електронного видання	Бібліотекар має можливість перевіряти правильність та повноту бібліографічних даних електронних ресурсів, відстежувати посилання на недоступні, застарілі або некоректні матеріали та за потребою їх редагувати або видаляти.
Бібліотекар	Пошук електронних видань	Бібліотекар має можливість швидко знаходити необхідні матеріали за потребою.

Відповідно таблиці 1.1 у графічних матеріалах наведені схеми варіантів використання.

Тепер, відповідно визначених варіантів використання, визначимо функціональні вимоги та встановимо їм пріоритет, занесемо результат у таблицю 1.2.

Таблиця 1.2 – Функціональні вимоги

Актор	Варіант використання	Функціональна вимога	Пріоритет
Усі	Авторизація у системі	1. Система надає змогу увійти у систему за допомогою логіна та паролю. 1.1. Система перевіряє чи дані коректні і якщо так, то дає доступ до відповідного функціоналу.	Високий
Адміністратор	Управління користувачами системи	2. Система надає можливість управляти обліковими записами користувачів. 2.1. Система надає можливість адміністратору створювати нові, редагувати та видаляти існуючі облікові записи користувачів. 2.2. Система надає можливість адміністратору надавати та змінювати права доступу у користувачів.	Високий
Адміністратор	Ведення бази даних бібліотеки	3. Система надає можливість додавати, редагувати, видаляти будь-які елементи у базі даних електронних видань.	Високий

Продовження таблиці 1.2

Актор	Варіант використання	Функціональна вимога	Пріоритет
Бібліотекар	Каталогізація електронних видань	4. Система надає можливість бібліотекарю проводити та контролювати процес каталогізації. 4.1 Система надає можливість імпортувати файли у систему. 4.2. Система надає можливість бібліотекарю самостійно заповнювати бібліографічні данні про видання. 4.3. Система надає можливість бібліотекарю вносити корективи у результат каталогізації системою.	Високий
Бібліотекар	Класифікація електронних видань	5. Система надає можливість бібліотекарю використовувати її для автоматичної класифікації. 5.1. Система надає можливість редагувати результат класифікації, якщо він є незадовільний. 5.2. Система може надавати електронному виданню унікальний ідентифікатор та зберігати його у базі даних.	Високий
Бібліотекар	Управління електронними виданнями	6. Система надає можливість додавати нове та редагувати відомості про існуючі електронні видання.	Високий

Продовження таблиці 1.2

Актор	Варіант використання	Функціональна вимога	Пріоритет
Бібліотекар	Управління відомостями про авторів електронних видань	7. Система надає можливість управляти відомостями про авторів електронних видань. 7.1 Система надає можливість редагувати/додавати/видаляти відомості про авторів.	Середній
Бібліотекар	Управління каталогами електронних видань	8. Система надає можливість бібліотекарю управляти каталогами електронних видань. 8.1. Система надає можливість редагувати відомості про каталоги та змінювати їх вміст.	Низький
Бібліотекар	Моніторинг стану електронного видання	9. Система надає можливість бібліотекарю контролювати стан електронних видань. 9.1. Система надає можливість бібліотекарю редагувати та видаляти існуючі електронні видання. 9.2. Система надає можливість бібліотекарю відстежувати посилання на недоступні, застарілі або некоректні матеріали та видаляти або редагувати їх.	Високий

Продовження таблиці 1.2

Актор	Варіант використання	Функціональна вимога	Пріоритет
Бібліотекар	Пошук електронних видань	10. Система надає можливість усім користувачам системи швидко знаходити необхідні ресурси.	Середній

1.2 Огляд наявних аналогів

Існують різні інструменти та програмні рішення, які можна віднести до аналогів даної системи, до більш відомих з них можна віднести:

- Ex Libris Alma;
- Koha;
- Ex Libris Aleph.

Ex Libris Alma – це хмарне програмне забезпечення, яке включає модулі каталогізації та класифікації, а також інтеграцію з електронними ресурсами та засобами збереження електронних документів. Дане програмне забезпечення дає змогу бібліотекам ефективно управляти колекцією електронних ресурсів та забезпечувати більш точну каталогізацію.

Переваги:

- підтримка інтеграції з багатьма іншими системами;
- високий рівень безпеки даних;
- можливість інтеграції з існуючими системами;
- сучасний та зрозумілий інтерфейс користувача.

Недоліки:

- складність у використанні системи, яка вимагає додаткову підготовку для персоналу бібліотеки;
- високі вимоги до ресурсів сервера, що впливає на швидкість роботи системи загалом;

– висока вартість та складність у використанні.

Koha – це відкрите інтегроване програмне забезпечення, що надає функціональність каталогізації та класифікації електронних видань у бібліотеці, а також автоматизацію багатьох інших бібліотечних процесів.

Переваги:

- безкоштовна та доступна для всіх користувачів;
- підтримка інтеграції з існуючими системами;
- простий та зручний інтерфейс користувача;
- великий функціонал дозволяє налаштувати систему залежно від потреб користувача.

Недоліки:

- складність установки під Windows та можливі через це проблеми зі сумісністю;
- низький рівень безпеки персональних даних користувачів.

Ex Libris Aleph – це відкрите інтегроване програмне забезпечення, яка включає модулі для каталогізації, класифікації та електронного документообігу.

Ця система успішно функціонує у багатьох бібліотеках світу.

Переваги:

- підтримка інтеграції з існуючими системами;
- великий функціонал дозволяє налаштувати систему залежно від потреб користувача;
- відкритий вихідний код та гнучка архітектура;
- високий рівень безпеки даних.

Недоліки:

- високі вимоги до ресурсів сервера, що впливає на швидкість роботи системи загалом;
- старий та складний інтерфейс користувача, що робить його незручним та незрозумілим у використанні;

Отже, кожна інтегрована система має свої переваги та недоліки, вибір кожної залежить від потреб користувача та можливостей бібліотеки.

Враховуючи все вище сказане, можна прийти до висновку, що наша система буде мати:

- простий та інтуїтивно зрозумілий інтерфейс;
- високий рівень безпеки даних;
- можливість розширення та налаштування системи;
- безкоштовність та доступність для усіх користувачів.

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначення розробки полягає у тому, щоб розробити автоматизовану систему, яка буде дозволяти бібліотекам ефективно проводити процес каталогізації та класифікації електронних видань,

1.3.2 Цілі та задачі розробки

Мета дипломного проекту – створення автоматизованої системи каталогізації та класифікації, що дозволить бібліотекарям ефективно керувати електронними ресурсами бібліотеки, забезпечувати зручний доступ до цих ресурсів та зменшувати час та зусилля, які необхідні для їх управління.

Для досягнення мети необхідно виконати наступні задачі:

- розробка модулю експорту та імпорту, який дозволить завантажувати електронні ресурси з різних джерел та експортувати дані для зберігання на зовнішніх носіях;
- розробка модуля каталогізації, який буде дозволяти вносити бібліографічні дані про електронні ресурси та визначати ключові слова;
- розробка модуля класифікації, який дозволить віднести кожне електронне видання до певної категорії та присвоювати їм унікальний ідентифікатор;
- розробка модуля зберігання та оновлення даних, призначення якого полягає у тому, щоб зберігати дані у структурованому форматі у базі даних та оновлювати відомості про електронні ресурси у режимі реального часу, щоб підтримувати актуальну та точну інформацію;

					IC91.080БАК.004 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис			

- розробка модуля пошуку, який буде дозволяти швидко та зручно знаходити необхідні електронні ресурси;
- розробка модуля адміністрування, який дозволить адміністраторам системи керувати базою даних електронних видань та користувачів системи;
- забезпечення безпеки даних для користувачів;
- розробка інтерфейсу користувача, призначення якого полягає у тому, щоб користувачам було зручно взаємодіяти з системою та здійснювати необхідні їм дії.

Висновки до розділу

Під час написання даного розділу було зроблено опис предметного середовища, яке розглядається. Також було наведено діаграму діяльності та наведено її детальний опис.

Також було розроблено функціональну модель, в якій описано акторів системи та функції, які вони повинні виконувати. Функціям був наданий пріоритет за важливістю.

Окрім того, були описані існуючі рішення та проаналізовано їх переваги та недоліки. Як результат було встановлено які відмінності матимете система, яка розробляється.

В кінці розділу було сформовано призначення та мета розробки, визначено задачі, які необхідно реалізувати для її досягнення.

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідними даними для системи є електронні ресурси та дані про користувачів системи.

Кожне електронне видання враховує наступну інформацію про себе: бібліографічні дані та текстовий файл.

Бібліографічні дані включають наступні атрибути:

- заголовок;
- автор чи автори;
- рік видання;
- опис: короткий опис або анотація видання;
- видавництво;
- тип видання;
- мова: вказівка про мову написання;

Крім основних бібліографічних даних про видання є файл з електронним виданням, який допомагає організовувати та класифікувати це видання у базі даних. Він містить наступні дані про себе:

- текст видання;
- зображення (за наявністю).

Для системи також необхідно зберігати деяку інформацію про користувачів бібліотеки, зокрема бібліотекарів та адміністраторів. Ці дані включатимуть:

- персональні дані: ім'я та прізвище;
- логін та пароль: дані для авторизації у системі;
- роль: визначення ролі користувача (бібліотекар, адміністратор) та відповідні права доступу.

2.2 Вихідні дані

Вихідні дані – це те, що отримаємо після взаємодії з автоматизованою системою.

					ІС91.080БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			19

До основних вихідних даних системи належать:

- електронне видання у каталозі та його бібліографічні дані;
- перелік категорій, в яких містяться електронні видання, та перелік авторів;
- посилання на електронні видання, що дозволить користувачам отримати прямий доступ до необхідних матеріалів;
- звіт про активність користувачів: звіт надаватиме інформацію про активність користувачів системи.

2.3 Опис структури бази даних

Для автоматизованої системи спроектовано базу даних, яка зберігає інформацію про електронне видання, користувачів системи. У таблиці 2.1 наведено опис основних таблиць бази даних.

Таблиця 2.1 – Опис основних таблиць бази даних

Назва таблиці	Опис
User	Таблиця відповідає за зберігання даних про користувачів системи
UserAction	Таблиця відповідає за зберігання інформації про взаємодію користувачів з електронним виданням
ActionType	Таблиця, яка зберігає в собі інформацію про перелік всіх можливих дій користувача;
Publication	Таблиця відповідає за зберігання інформації про електронне видання
Author	Таблиця відповідає за зберігання інформації про авторів електронних видань
Category	Таблиця відповідає за зберігання всіх категорій та підкатегорій
Keyword	Таблиця відповідає за зберігання всіх ключових слів, які відносяться до електронних видань
Type	Відповідає за зберігання типів електронних видань

Продовження таблиці 2.1

Назва таблиці	Опис
Language	Таблиця відповідає за зберігання мов написання електронних видань
PublicationFile	Таблиця відповідає за зберігання файлів з текстом та зображенням (за наявністю) електронних видань

Також у базі даних є декілька допоміжних таблиць, опис яких наведено у таблиці 2.2.

Таблиця 2.2 – Опис допоміжних таблиць бази даних

Назва таблиці	Опис
PublicationAuthor	Таблиця відповідає за створення зв'язку між таблицями Publication і Author
PublicationCategory	Таблиця відповідає за створення зв'язку між таблицями Publication і Category
PublicationKeyword	Таблиця відповідає за створення зв'язку між таблицями Publication і Keyword

Кожна з таблиць має своє призначення. Якщо основні таблиці відповідають за зберігання конкретних сутностей у автоматизованій системі, то призначення допоміжних таблиць полягає у створенні зв'язку між деякими сутностями.

Тепер розглянемо детальний опис полів кожної з таблиць у таблиці 2.3.

Таблиця 2.3 – Опис полів таблиць бази даних

Назва таблиці	Назва поля	Опис
User	User_ID	Унікальний ідентифікатор користувача
	Username	Логін
	Password	Пароль
	FirstName	Ім'я користувача
	LasrName	Прізвище користувача
	Access_right	Права доступу
UserAction	Action_ID	Унікальний ідентифікатор дії користувача
	User_ID	Ідентифікатор користувача
	ActionType_ID	Ідентифікатор типу дії користувача
	Publication_ID	Ідентифікатор електронного видання
	Action_date	Дата дії
ActionType	ActionType_ID	Унікальний ідентифікатор типу дії користувача
	Name	Назва типу дії користувача
Publication	Publication_ID	Унікальний ідентифікатор електронного видання
	Title	Назва видання
	Publisher	Назва видавництва
	Publication_date	Дата видання
	Description	Опис видання
	Type_ID	Ідентифікатор типу видання
	Language_ID	Ідентифікатор мови написання

Продовження таблиці 2.3

Назва таблиці	Назва поля	Опис
Author	Author_ID	Унікальний ідентифікатор автора
	FirstName	Ім'я автора
	LastName	Прізвище автора
	Date_of_birth	Дата народження
	Date_of_death	Дата смерті
	Description	Опис автора
Category	Category_ID	Унікальний ідентифікатор категорії
	Name	Назва категорії
	Description	Опис категорії
	Parent_ID	Належність до якоїсь категорії
Keyword	Keyword_ID	Унікальний ідентифікатор ключового слова
	Name	Назва ключового слова
Type	Type_ID	Унікальний ідентифікатор типу видання
	Name	Назва типу видання
Language	Language_ID	Унікальний ідентифікатор мови написання
	Name	Назва мови
PublicationFile	PublicationFile_ID	Унікальний ідентифікатор файлів публікації
	ImageFile	Файл з зображенням
	TextFile	Файл з текстом

Продовження таблиці 2.3

Назва таблиці	Назва поля	Опис
PublicationAuthor	Author_ID	Ідентифікатор автора
	Publication_ID	Ідентифікатор видання
	Date	Дата
	Notes	Нотатки
PublicationCategory	Category_ID	Ідентифікатор категорії
	Publication_ID	Ідентифікатор видання
	Date	Дата
	Notes	Нотатки
PublicationKeyword	Keyword_ID	Ідентифікатор ключового слова
	Publication_ID	Ідентифікатор видання

На основі таблиці 2.3, було побудовано схему бази даних, яка представлена в графічних матеріалах.

Визначимо декілька ключових моментів, які можна побачити на діаграмі:

– кожне електронне видання може належати до однієї чи більше категорій, а кожна категорія може мати багато видань, тому реалізується зв'язок багато-до-багатьох за допомогою таблиці PublicationCategory;

– кожне видання може мати одного чи більше авторів, а також кожен автор може мати одне чи більше видань, тому тут також реалізується зв'язок багато-до-багатьох за допомогою таблиць PublicationAuthor;

– одне видання може мати декілька ключових слів, а одне ключове слово може підходити до декількох видань, тому реалізується зв'язок багато-до-багатьох за допомогою таблиці PublicationKeyword;

– категорія може мати у собі підкатегорії, тому створено поле Parent_ID, яке або дорівнює null, коли це батьківська категорія, або дорівнює id батьківської категорії;

– таблиця UserAction створює зв'язок типу багато-до-багатьох між таблицею

з виданнями та бібліотекарями. Вона відображає взаємодію користувача з електронними виданнями, що дозволяє забезпечити більший контроль та безпеку даних.

Висновки до розділу

Під час написання цього розділу були визначені вхідні та вихідні дані автоматизованої системи.

Була розроблена структура бази даних, яка відповідає поставленим задачам каталогізації та класифікації електронних видань у бібліотеці. Вона дозволяє зберігати інформацію про видання, категорії тощо, що дозволяє здійснювати ефективний пошук та організацію видань. Також у ній міститься інформація про користувачів системи та їх рівнів доступу, зберігається інформація про активність користувачів у системі для забезпечення безпеки даних.

					ІС91.080БАК.004 ПЗ	Арк.
						25
Зм.	Лист	№ докум.	Підпис			

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Система каталогізації та класифікації отримує на вхід електронні видання різних типів, таких як стаття, книга, журнали тощо, і має завдання визначити їх ключові слова та до якої категорії (каталогу) віднести.

Так як основна мета розробки полягає у тому, щоб автоматизувати процес каталогізації та класифікації й зменшити залежність від ручної обробки даних, забезпечити швидке та точне визначення необхідних параметрів, то для цього доцільним буде використання алгоритмів машинного навчання.

3.2 Математична постановка задачі

Постановка задачі полягає в розробці алгоритму автоматичного визначення ключових слів на основі текстової інформації та визначенні класу, до якого можна віднести видання, на основі тієї ж текстової інформації, але до неї вже додається визначенні ключові слова.

Тому, нехай:

- T : текст електронного видання;
- K : множина ключових слів;
- C : множина категорій, до яких можна віднести електронне видання.

Цільова функція в задачах каталогізації та класифікації електронних видань полягає в оптимізації або мінімізації певних значень, що відображає ефективність розв'язку задачі.

У випадку задачі каталогізації, а саме визначення ключових слів, цільова функція оцінюється точністю визначення ключових слів відносно тексту видання. Тобто задача може сформулюватися, як пошук оптимальної функції віднесення F : $T \rightarrow K$, яка відображає текст видання на множину ключових слів.

У випадку класифікації цільова функція визначає ефективність класифікаційної моделі на основі точності, збалансованості розподілу категорій. Тобто задача може сформулюватися, як пошук оптимальної функції віднесення G :

$K \times T \rightarrow C$, яка відображає ключові слова тексту видання та сам текст на множину категорій.

3.3 Обґрунтування методів розв'язання

3.3.1 Методи визначення ключових слів

Методи визначення ключових слів можуть включати:

- TF-IDF;
- TextRank.

Розглянемо ці алгоритми більш докладно.

TF-IDF (Term Frequency-Inverse Document Frequency) [4] – статистичний показник, який використовується для визначення ключових слів на основі їх значимості у тексті. Він враховує частоту входження обраного слова в текст (term frequency) й інверсію частоти зустрічі цього слова у всьому корпусі текстів (inverse document frequency).

Отже, основна формула для знаходження частоти входжень обраного слова наступна:

$$TF = \frac{n_i}{\sum_k n_k}, \quad (3.1)$$

де n_i – кількість входжень слова в текст, $\sum_k n_k$ – сума, яка визначає загальну кількість слів в тексті.

Формула для знаходження інверсії частоти входження обраного слова:

$$IDF = \log \frac{|D|}{|(d_i \supset t_i)|}, \quad (3.2)$$

де $|D|$ – кількість текстів у корпусі, $|(d_i \supset t_i)|$ – кількість текстів, в яких зустрічається слово t_i .

Значення TF-IDF вказує на значимість слова (t) у конкретному тексті (d) в контексті всього корпусу текстів (D). Чим вище значення TF-IDF (формула 3.3), тим більша вага приділяється слову, і тим вірогідніше, що воно є ключовим для даного тексту.

$$TF - IDF = TF * IDF. \quad (3.3)$$

Алгоритм TextRank [5], який базується на концепції PageRank, використовує графову модель для визначення важливості слів у тексті. Кожне слово у тексті представляється вузлом у графі, а ребра відображають ступінь семантичного зв'язку між словами.

Початкове значення ваги кожного ребра у традиційному варіанті реалізації алгоритму дорівнює 1, але як додаткова модифікація алгоритму може визначатись за допомогою статистичного показника TF-IDF тощо.

Значення важливості (ваги) кожного слова обчислюється за допомогою ітеративного алгоритму TextRank.

$$TR(t_i) = (1 - d) + d * \sum_{(t_j, t_i) \in E} \frac{w_{ji}}{\sum_{(t_k, t_j) \in E} w_{kj}} * TR(t_j) , \quad (3.4)$$

де d – коефіцієнт демпфірування (зазвичай дорівнює 0.85), w_{ij} – вага ребра (t_i, t_j) .

Ітерація виконується поки ваги не змінюються або до досягнення певної заданої кількості ітерацій.

3.3.2 Методи визначення категорій

Методи визначення категорій тексту можуть включати наступні підходи:

- Наївний Баєс;
- метод опорних векторів;
- дерево рішень.

Метод Наївного Баєса ґрунтується на теоремі Баєса та використовує статистичні методи для класифікації тексту. Цей метод заснований на припущенні незалежності між ознаками (словами) у тексті, що дає швидкість та простоту обробки [6]. David Hand і Keming Yu вважають, що метод наївного Баєса є методом, який надзвичайно простий у реалізації, проте несправедливо вважається дурним [7], тобто метод дозволяє швидко та ефективно обробляти тексти, проте зберігати високу точність класифікації.

Формула для обчислення ймовірності належності тексту до певного класу виглядає наступним чином:

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)}, \quad (3.5)$$

де $P(C|D)$ – ймовірність належності видання D до класу C ,

$P(D|C)$ – ймовірність появи видання D , якщо він належить класу C (іноді називається ймовірністю моделі),

$P(C)$ – ймовірність зустрічі видання в класі C перед тим, як переглядати текст видання,

$P(D)$ – ймовірність появи документу D (ймовірність, що зустрічається будь-який документ).

Наївний Баєсовий класифікатор передбачає, що кожна ознака (слово) у документі незалежна від інших ознак при заданому класі.

Тому, ймовірність $P(D|C)$ розкладається на добуток ймовірностей появи окремих ознак у документі:

$$P(D|C) = P(w_1|C) * P(w_2|C) * ... * P(w_n|C), \quad (3.6)$$

де $P(w_i|C)$ – ймовірність появи ознаки w_i у документі, якщо документ належить класу C .

Наївний Баєсівський класифікатор має свої переваги та недоліки. Перевагою цього підходу є зменшення вимог до розміру вибірки з експоненційного до лінійного росту. Це означає, що класифікатор може працювати надійно, навіть якщо доступні обмежені дані.

Однак, недоліком наївного Баєсівського класифікатора є припущення про незалежність змінних. Це означає, що він вважає, що кожна змінна впливає на класифікацію незалежно від інших змінних. У реальних ситуаціях це припущення може бути неправдивим, що призводить до втрати точності класифікації. Крім того, обчислені ймовірності можуть бути неправильними, і їх сума може не дорівнювати одиниці, що вимагає нормалізації результатів.

Незважаючи на перелічені недоліки, наївний Баєсівський класифікатор залишається практично ефективним. Навіть при незначних відхиленнях від припущення про незалежність змінних, він може продовжувати корелювати з правильною класифікацією об'єктів. Крім того, він має високу швидкість роботи,

простоту і масштабованість, що робить його привабливим для застосування в багатьох практичних ситуаціях з обмеженими ресурсами.

Наступний метод, який ми розглянемо – метод опорних векторів (Support Vector Machine – SVM).

Метод опорних векторів шукає границю розділення між категоріями шляхом побудови оптимальної гіперплощини у просторі ознак. Вектори, що лежать на границі, називаються опорними векторами. За допомогою математичних оптимізаційних методів, SVM знаходить оптимальну границю, яка максимізує відстань між границею та найближчими опорними векторами. Цей метод може використовуватись для бінарної та багатокласової класифікації тексту.

Метод опорних векторів має свої переваги та недоліки, які слід враховувати при його використанні. Недолік методу є те, що для класифікації використовується лише невелика частина зразків, яка знаходиться на границях. З іншої сторони, перевагою методу є те, що для класифікації достатньо невеликого набору даних, що відрізняє його від більшості інших методів. Якщо модель коректно працює на тестовій вибірці, то можна успішно застосовувати цей метод на реальних даних.

Дерева рішень є теж одним з популярних методів класифікації та прогнозування, який використовується для аналізу та визначення класу або значення цільової змінної на основі вхідних ознак. Вони надають зрозумілу структуру, схожу на дерево, де кожен вузол представляє розгалуження на основі певної ознаки, а кожне листя дерева представляє прогнозоване значення.

Основна ідея дерев рішень полягає в тому, щоб розділити набір даних на все менші та більш однорідні підмножини, шляхом обирання найінформативніших ознак. Цей процес відбувається шляхом рекурсивного розгалуження на основі критеріїв, таких як індекс Джині, ентропія або інформаційний приріст.

Наведемо приклад знаходження індексу Джині за формулою:

$$gini(A) = 1 - \sum_{j=1}^n p_j^2, \quad (3.7)$$

де A – поточний вузол, p_j – ймовірність класу j у поточному вузлі, n – кількість класів.

					IC91.080БАК.004 ПЗ	Арк.
						30
Зм.	Лист	№ докум.	Підпис			

Основні кроки у побудові дерева рішень:

1) Вибір ознаки: на цьому кроці необхідно вибрати найбільш інформативний атрибут, за яким найкраще розділити набір даних на більш однорідні підмножини.

2) Вибір критерію зупинки: цей крок визначає, коли процес побудови дерева має бути зупинений. Наприклад, можна встановити максимальну глибину дерева чи мінімальну кількість зразків у вузлі.

3) Вибір методу обрізки гілок: під час навчання дерева рішень можуть виникати гілки, які не мають значної впливу на точність моделі або можуть привести до перенавчання. Обрізка гілок є процесом видалення таких непотрібних гілок для поліпшення ефективності моделі.

4) Оцінка точності побудованого дерева: останній крок полягає у вимірюванні точності моделі на незалежних тестових даних або за допомогою перехресної перевірки. Це допомагає оцінити якість та ефективність дерева рішень та прийняти рішення щодо його подальшого використання.

Отже, дерева рішень мають декілька переваг, таких як легка інтерпретованість, здатність працювати з числовими та категоріальними ознаками, і відсутність потреби в нормалізації даних. Однак, вони також можуть бути схильними до перенавчання, коли модель дуже точно працює на навчальних даних, але погано на нових.

3.4 Опис методу розв'язання

Обраним методом для розв'язання поставленої задачі каталогізації є алгоритм TextRank.

Опишемо алгоритм TextRank:

1) Виконати попередню обробку тексту, яка включає наступні етапи: токенизація (розбиття тексту на окремі слова або словосполучення), видалення стоп-слів (часто вживаних слів, які не несуть значущої інформації) і проведення процесу, який зводить слова до базової форми;

2) Побудувати граф, де кожне слово у тексті представлене вузлом, а ребра відображають семантичний зв'язок між словами. Ребро між двома словами (t_i, t_j) існує, якщо ці слова з'являються поруч у тексті, у вікні фіксованого розміру;

3) Ініціалізувати початкову вагу кожного слова у графі (вузла), використовуючи значення TF-IDF;

4) Виконати ітерації алгоритму TextRank для збільшення точності ранжування (формула 3.4). У кожній ітерації:

а) обчислити нову вагу для кожного вузла шляхом усереднення ваг суміжних вузлів і зваженого зв'язку між ними;

б) оновити вагу кожного вузла за новими значеннями;

5) Після закінчення ітерацій отримати значення ваги для кожного слова. Чим вище значення ваги, тим більша важливість приділяється слову, і тим вірогідніше, що воно є ключовим для тексту;

6) Після закінчення ітераційного процесу, слова з найвищими вагами вважаються ключовими словами.

Для рішення задачі класифікації було обрано реалізувати Баєвський класифікатор через його простоту та ефективність.

Загальний алгоритм для вирішення задачі класифікації наступний:

1) Зібрати набір текстів та їх ключові слова, для яких відомі категорії. Кожен текст повинен бути позначений категорією, до якої він належить. У нашому випадку буде використовуватись електронні видання бібліотеки, які були додані попередньо;

2) Розрахувати ймовірності наявності кожного ключового слова для кожної категорії. Для цього підрахувати, скільки разів ключове слово зустрічається в текстах цієї категорії та поділити на загальну кількість слів в цій категорії;

3) Після підрахунку ймовірностей для кожної категорії, можна використовувати ці значення для класифікації нового тексту. Розрахувати ймовірність належності тексту до кожної категорії, використовуючи відомі ключові слова в тексті та їх відповідні ймовірності. Категорія з найбільшою ймовірністю вважається категорією, до якої належить текст.

Висновки до розділу

У даному розділі було наведено змістовну та математичну постановку задачі. Були описані вхідні дані та вихідні дані, тобто умова задачі, що нам дано і що ми повинні отримати у результаті. Також було наведено опис можливих методів розв’язання даної задачі, проаналізовано їх переваги та недоліки. Було визначено конкретний метод розв’язання для задачі каталогізації (визначення ключових слів) та класифікації, описано алгоритм рішення.

					ІС91.080БАК.004 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис			

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

Автоматизовану систему розроблено за допомогою технологій розробки трьох основних компонентів: клієнтської та серверної частин та бази даних.

Процес створення клієнтської частини має на меті створення вебінтерфейсу, який відповідає за взаємодію з користувачем. Він включає в себе створення HTML-шаблонів, які відповідають за розмітку та вигляд сторінок, CSS-стилі для їх оформлення та JavaScript-код для забезпечення динамічної взаємодії.

HTML-шаблони використовуються для створення структури сторінок та включення даних з сервера. Вони визначають елементи, такі як заголовки, кнопки, форми та інші компоненти, які користувач може бачити та взаємодіяти з ними. CSS-стилі в свою чергу використовуються для оформлення та вигляду елементів сторінки. Вони визначають кольори, шрифти, розташування елементів та інші аспекти зовнішнього вигляду.

JavaScript-код використовується для забезпечення динамічної взаємодії з клієнтом. Він дозволяє реагувати на події користувача, виконувати асинхронні запити до сервера, змінювати вміст сторінок без перезавантаження, а також реалізовувати різноманітні функціональні можливості, такі як валідація форм, анімація та інші.

Для створення серверної частини програмного застосунку обрано мову Python. Ця мова програмування має відносно простий синтаксис, що робить код читабельним, а це в свою чергу полегшує його розуміння та роботу з ним. Python також відомий своєю потужною стандартною бібліотекою, яка надає безліч корисних функцій та модулів. Завдяки цьому, Python дозволяє легко вирішувати різноманітні завдання та задачі.

Слід зазначити, що Python – це крос-платформена мова програмування, що означає, що програмне забезпечення, написане на цій мові, можуть працювати на різних операційних системах, таких як Windows, Linux, MacOS та інші.

Також важливий фактор вибору саме Python для розробки системи – це

					IC91.080БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			34

можливість підтримки інтеграції з іншими технологіями та мовами програмування. Це дозволяє поєднання модулів автоматизованої системи каталогізації та класифікації з іншими модулями бібліотеки у майбутньому.

Виходячи з вибору мови програмування, середою розробки автоматизованою системи було обрано PyCharm.

PyCharm – це інтегроване середовище розробки (IDE) для мови програмування Python, розроблене компанією JetBrains і надає всі необхідні інструменти для розробки на обраній мові програмування.

До основних переваг IDE можна віднести:

- управління проектами: PyCharm надає можливість легко створювати, налаштовувати та управляти проектами. Воно підтримує різні типи проектів, серед яких є консольні додатки, вебзастосунки тощо, і дозволяє використовувати нові версії фреймворків та бібліотек;

- відладка: PyCharm надає можливість ставити точки зупинки, спостерігати за значеннями змінних, виконувати код по крокам та аналізувати стек викликів. Все це полегшує розробку програм і дозволяє швидко писати код;

- контроль версій: це середовище розробки дозволяє інтегрувати будь-який проект з системами контролю версій, такими як Git, і використовувати потужні інструменти для керування змінами в коді.

Для реалізації автоматизованої системи було обрано фреймворк Django.

Django – це фреймворк, розроблений на мові програмування Python, який дозволяє швидко та ефективно будувати складні вебзастосунки. Він має вбудовану адміністративну модель, велику кількість розширень та структуровану організацію проекту.

До основних особливостей Django, завдяки яким було обрано саме цей фреймворк, можна віднести:

- шаблонний двигун: Django має вбудований шаблонний двигун, який дозволяє відокремити логіку від представлення;

- безпека: Django надає вбудовану систему безпеки, яка допомагає запобігти атакам, наприклад, перехопленні сесій, внедрення SQL-запитів тощо. Також він

має вбудовану підтримку автентифікації, авторизації та контролю доступу;

– розширюваність: Django має велику кількість пакетів та бібліотек, які дозволяють легко розширювати його функціональність. Це дозволяє використовувати готові рішення для різних завдань, таких як робота зі різними сторонніми сервісами, інтеграція з базами даних тощо.

– URL маршрутизація: Django має потужну систему маршрутизації URL, яка дозволяє визначати, які вебсторінки пов’язані з якими представленнями, це в свою чергу спрощує навігацію та побудову структурованого вебзастосунку;

– ORM (Object-Relational Mapping): Django надає вбудований ORM, який дозволяє зручно взаємодіяти з базою даних без прямої роботи з SQL-запитами. ORM перетворює об’єкти моделей Django на записи бази даних та надає зручний спосіб виконання операцій з базою даних, таких як створення, читання, оновлення та видалення даних.

В проєкті будемо використовувати систему керування базами даних PostgreSQL.

PostgreSQL – це потужна та надійна об’єктно-реляційна система керування базами даних, яка працює на різних операційних системах. Ця СКБД буде доцільною у використанні, оскільки вона надійна, гнучка та сумісна з багатьма іншими інструментами.

Тепер опишемо бібліотеки, які допоможуть саме у реалізації системи каталогізації та класифікації електронних видань.

NumPy – це основна бібліотека для обчислень у Python. Вона надає підтримку для масивів великого розміру та математичних операцій з ними. NumPy є важливою складовою для багатьох інших бібліотек, зокрема для обробки даних і машинного навчання.

Pandas - це потужна бібліотека для обробки та аналізу даних. Вона надає структури даних, такі як DataFrame, що спрощує маніпулювання та аналіз даних. Pandas дозволяє зчитувати дані з різних джерел, виконувати фільтрацію, сортування, обчислення статистики та багато іншого.

Scikit-learn є відомою бібліотекою для машинного навчання. Вона надає реалізації багатьох алгоритмів класифікації, регресії, кластеризації, а також інструменти для оцінки моделей та підготовки даних. Scikit-learn має простий і зрозумілий API, що полегшує роботу з машинним навчанням.

TensorFlow є популярною бібліотекою для глибокого навчання та чисельних обчислень. Вона надає засоби для створення та тренування нейронних мереж. Keras в свою чергу є високорівневим API для побудови моделей глибокого навчання на базі TensorFlow. Ці бібліотеки дозволяють реалізувати складні моделі глибокого навчання для розпізнавання образів, обробки природної мови та багатьох інших завдань.

Окрім всіх наведених бібліотек та фреймворків, у розробці проекту буде використана система контролю версій – Git, основна мета якої – ведення історії змін вихідного коду проекту.

Основні особливості Git:

- розподілена архітектура: кожен клон репозиторію Git містить повну копію історії проекту, що дозволяє працювати локально та без підключення до центрального сервера;

- гілки: Git надає можливість створювати, керувати багатьма гілками коду. Це дозволяє працювати паралельно над різними функціоналом системи та вносити зміни у певний функціонал без впливу на головну гілку проекту;

- історія змін: Git зберігає повну історію змін вихідного коду. Кожен коміт має унікальний ідентифікатор, який дозволяє відстежувати коли і які зміни були внесені у проект;

- доступ до версій: так як Git зберігає кожен версію файлів проекту, то це дозволяє повертатись до попередніх версій, відновлювати видалені файли та відстежувати історію змін.

4.2 Загальні вимоги до технічного забезпечення

Дана автоматизована система буде представлена у вигляді вебзастосунку, тому мінімальні вимоги до технічного забезпечення:

					ІС91.080БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис			37

1) Комп'ютер:

а) комп'ютер з підтримкою будь-якої ОС, таких як Windows, macOS або Linux;

б) об'єм оперативної пам'яті не менше 2ГБ;

с) процесор з тактовою частотою не нижче 2ГГц.

2) Засоби вводу та виводу інформації:

а) наявність комп'ютерної миші, клавіатури та монітору.

3) Інтернет-підключення:

а) стабільне підключення до Інтернету для доступу до системи та взаємодії з нею.

4) Веббраузер:

а) рекомендовано використовувати останні версії веббраузерів, таких як Google Chrome, Mozilla Firefox тощо.

4.3 Архітектура програмного забезпечення

4.3.1 Схема архітектури програмного забезпечення

Для розробки системи використано трьохрівневу архітектуру, яка складається з рівня представлення, рівня бізнес-логіки та рівню доступу до даних [13]. Ця архітектура дозволяє відокремити функціональність системи на різні рівні, кожен з яких буде виконувати свої завдання.

Розглянемо більш детально кожен рівень:

1) Рівень представлення (Presentation Layer) - рівень, який відповідає за відображення інтерфейсу користувача. Він забезпечує представлення даних користувачу та отримання від нього вхідних даних. Цей рівень використовується для забезпечення зручної та ефективної взаємодії користувача з системою;

2) Рівень бізнес-логіки (Business Logic Layer) – рівень, який містить у собі логіку обробки даних та виконання бізнес правил. Він відповідає за обробку запитів користувача, виконання бізнес-логіки та взаємодію з рівнем доступу до даних;

3) Рівень доступу до даних (Data Access Layer) – рівень, який забезпечує доступ до бази даних та виконання операцій зберігання, оновлення та видалення

					ІС91.080БАК.004 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис			

даних. Він забезпечує ізоляцію даних від бізнес-логіки та представлення даних, тобто дозволяє легко змінювати джерела даних без впливу на іншу рівні архітектури.

Архітектура застосунку розробленого на Django базується на архітектурному шаблоні модель-поділена-на-компоненти (Model-View-Controller, MVC). Django використовує модифіковану версію MVC, яку часто називають модель-представлення-шаблони (Model-View-Template, MVT) (рисунок 4.1).

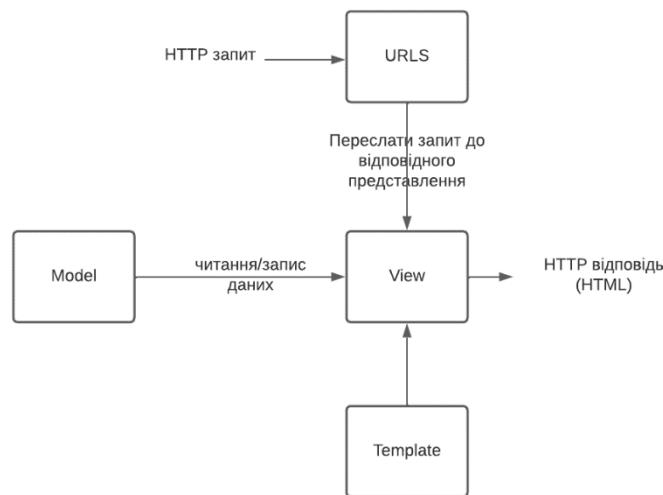


Рисунок 4.1 – Схема MVT шаблону

Основні компоненти архітектурного шаблону MVT:

- моделі (Models): моделі представляють структуру даних застосунку та відповідають за доступ до бази даних. Вони визначають таблиці, поля та взаємозв'язки між даними. Моделі в Django використовуються для збереження та отримання даних з бази даних;

- представлення (Views): представлення обробляють запити від користувача і визначають, які дані повинні бути показані. Вони отримують дані з моделей, виконують необхідні операції та передають результати до шаблонів для відображення;

- шаблони (Templates): шаблони визначають, як дані будуть відображені для користувача. Вони містять HTML-розмітку та спеціальні теги та синтаксис, що

дозволяють вставляти дані з представлень та виконувати логіку відображення;

– URL-шаблони (URL patterns): URL-шаблони визначають маршрутизацію запитів користувача до відповідних представлень. Вони вказують, які URL-адреси пов'язані з якими представленнями.

4.3.2 Діаграма послідовностей

Розглянемо діаграми послідовностей, які наведені у графічних матеріалах.

Опишемо деякі основні послідовності дій у автоматизованій системі.

Для того, щоб бібліотекар зміг користуватись функціоналом автоматизованої системи, адміністратор повинен виконати деякі попередні дії. Спочатку він повинен створити обліковий запис користувача, ввести логін та пароль, які користувач-бібліотекар буде використовувати для авторизації у системі. Далі для взаємодії з системою бібліотекар повинен авторизуватись у системі, тобто ввести логін та пароль, які йому визначив адміністратор. Після процесу автентифікації бібліотекар отримує доступ до функціоналу системи.

Після авторизації у бібліотекаря з'являється декілька можливих дій, одна з них – додавання нового видання. Для цього бібліотекарю необхідно натиснути відповідну кнопку на користувацькому інтерфейсі. Користувач вносить бібліографічні дані про видання та додає відповідний текстовий файл у форму. Далі бібліотекар натискає на відповідну кнопку и система обробляє текст, який йому надали і визначає ключові слова та категорію видання. Процес визначення ключових слів включає в собі: попередню обробку тексту, подуву графу слів та виконання алгоритму TextRank. Після цього виконується процес класифікації: на основі ключових слів визначається категорія електронного видання. Потім бібліотекар перевіряє результат автоматичної класифікації та зберігає дані у базі даних.

4.3.3 Діаграма розгортання

Тепер розглянемо фізичну архітектуру системи на рисунку 4.2, у тому числі

					ІС91.080БАК.004 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис			

розміщення апаратного та програмного забезпечення, мережеві взаємозв'язки та взаємодію зовнішніх ресурсів.

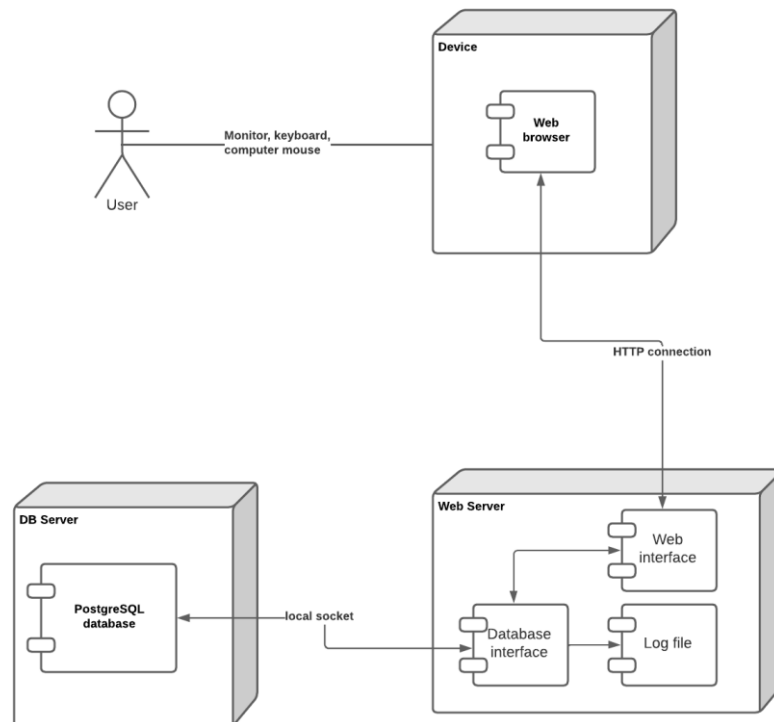


Рисунок 4.2 – Діаграма розгортання

Клієнтський браузер взаємодіє з вебсервером, надсилаючи HTTP-запити для отримання різних ресурсів та інформації з системи. Він отримує відповіді від сервера і відображає їх у зручному для користувача форматі, що дозволяє здійснювати взаємодію з системою, переглядати, додавати або редагувати дані.

Вебсервер виконує роль посередника між клієнтським браузером і базою даних. Вебсервер обробляє запити користувачів, взаємодіє з базою даних та повертає результати користувачам через вебінтерфейс.

Сервер бази даних забезпечує збереження даних та взаємодію з вебсервером за допомогою запитів. Він включає в себе всю необхідну інформацію про електронні видання, включаючи метадані, класифікацію та інші важливі атрибути.

Висновки до розділу

У даному розділі було визначено та описано засоби розробки та стек технологій для реалізації автоматизованої системи каталогізації та класифікації електронних видань у бібліотеці. Також було наведено архітектуру програмного забезпечення та конкретний архітектурний підхід. Були визначені загальні вимоги до технічного забезпечення для правильного функціонування вебзастосунку. Були наведені та описані діаграми послідовностей, компонентів та розгортання.

					ІС91.080БАК.004 ПЗ	Арк.
						42
Зм.	Лист	№ докум.	Підпис			

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

У даному розділі розглянемо функціонал системи та функції, які є у різних типів користувачів.

Для початку розглянемо можливості бібліотекаря.

Щоб користуватись функціоналом системи, бібліотекарю для початку необхідно авторизуватись. Для цього йому необхідно заповнити відповідні поля «Ім'я користувача» та «Пароль» на сторінці авторизації (рисунок 5.1).

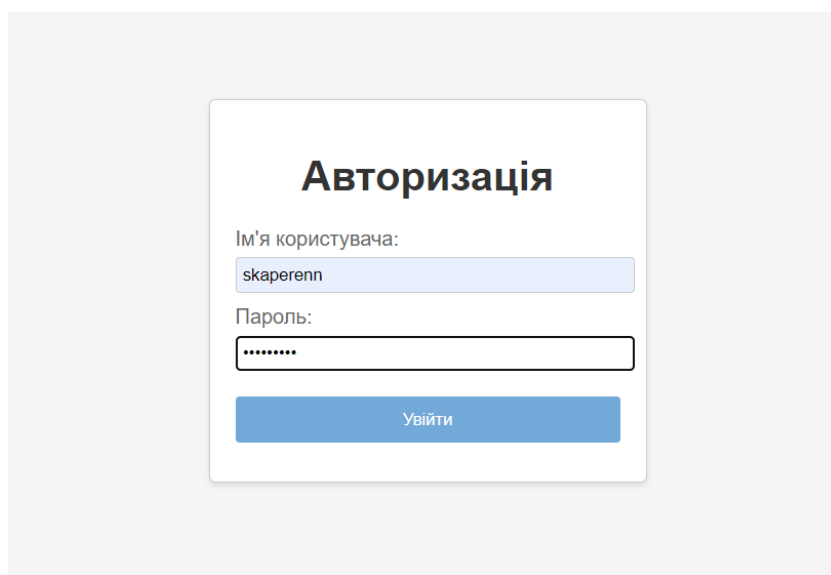


Рисунок 5.1 – Екран авторизації у системі

Якщо введені дані коректні, то користувач переходить на головну сторінку (рисунок 5.2).



Рисунок 5.2 – Екран головної сторінки автоматизованої системи

На головній сторінці ми бачимо наступне:

1) Бокову панель, на якій відображено перелік вкладок з відповідним функціоналом, на які ми можемо переходити:

- Домашня сторінка (на ній ми знаходимось);
- Категорії;
- Автори;
- Додати публікацію;
- Пошук;

А також кнопка вийти, яка перенаправляє нас знову на сторінку авторизації.

2) Текст, в якому міститься інформація, хто увійшов у систему;

3) Перелік функцій, які доступні у системі.

На сторінці з категоріями у нас є перелік всіх основних категорій електронних видань у бібліотеці та перелік всіх публікацій (рисунок 5.3).

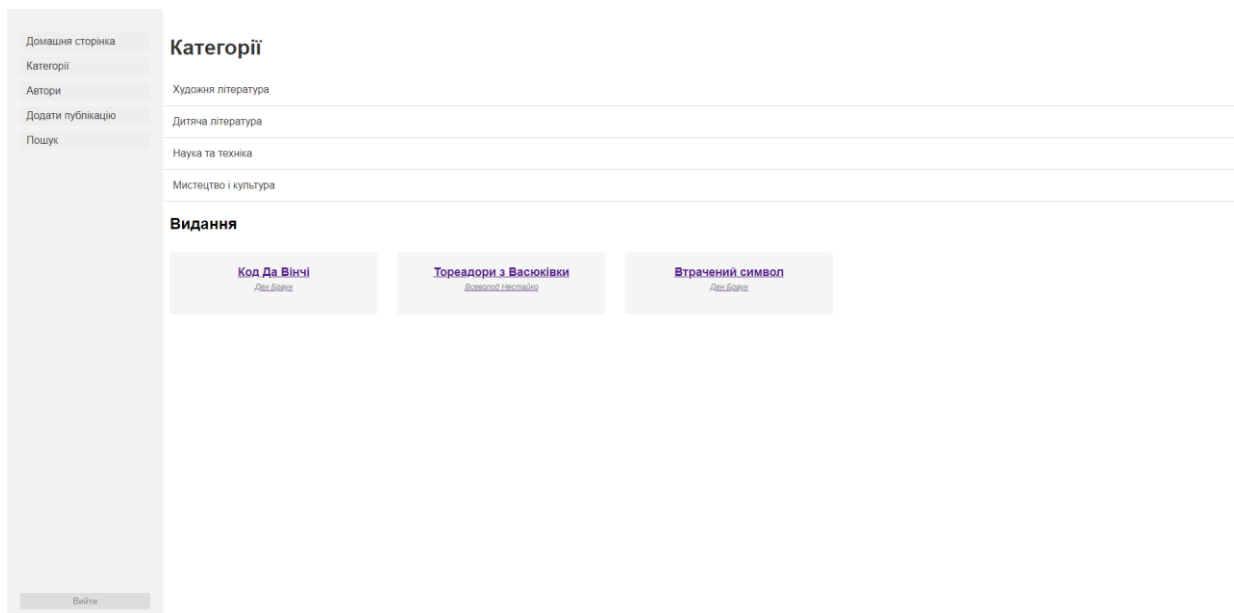


Рисунок 5.3 – Екран переліку категорій разом з всіма публікаціями

При натисканні на певну категорію у нас відкривається перелік всіх підкатегорій та нижче перелік всіх видань, які відносяться до цієї категорії. При переході на певну підкатегорію відкривається перелік всіх видань, які відносяться до неї (рисунок 5.4).

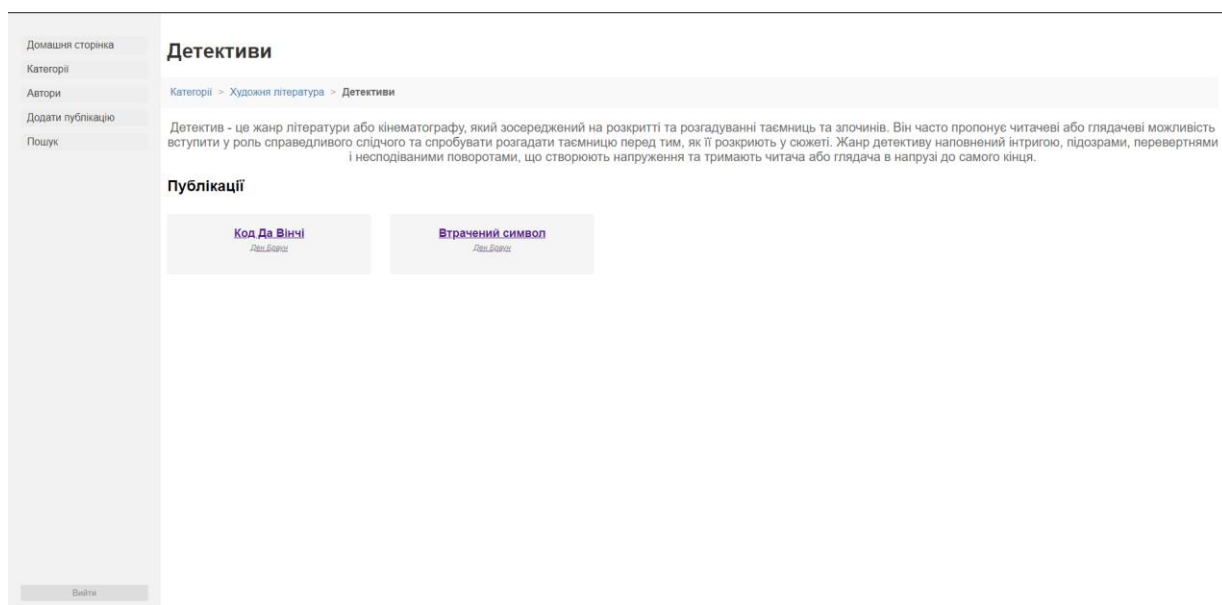


Рисунок 5.4 – Екран підкатегорії разом з виданнями, які до неї відносяться

Натиснувши на будь яку публікацію відкривається вікно з детальною інформацією про цю публікацію, де містяться всі бібліографічні данні видання та посилання на текстовий файл (рисунок 5.5).

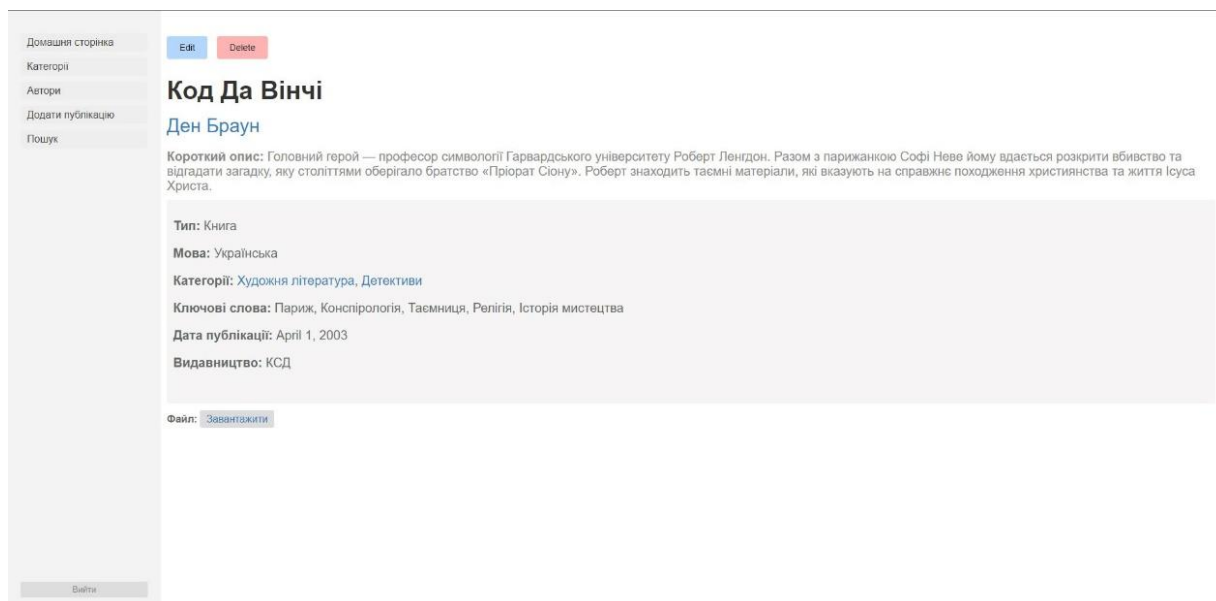


Рисунок 5.5 – Екран детальної інформації про електронне видання

Є також кнопка «Редагувати», при натисканні на неї сторінка змінюється і надається можливість редагувати всі доступні поля, зберегти зміни або видалити це видання з бібліотеки.

Тепер переглянемо вкладку «Автори». На цій вкладці є перелік всіх авторів, чий видання є у бібліотеці (рисунок 5.6).

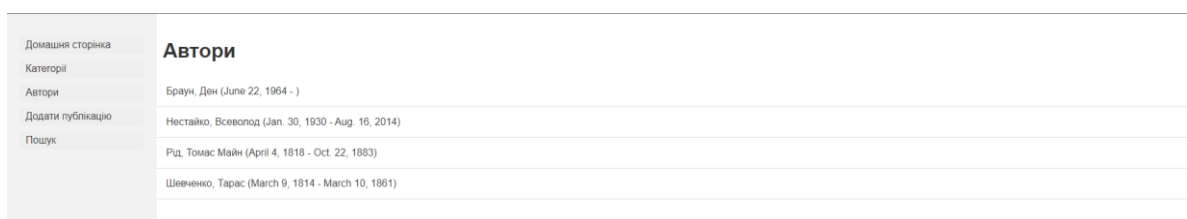


Рисунок 5.6 – Екран переліку всіх авторів, які є у системі

При натисканні на певного автора відкривається сторінка з детальною інформацією про автора, а також перелік його публікацій, які є у базі бібліотеки (рисунок 5.7). Також є аналогічні функції з редагуванням даних, як й у публікаціях.

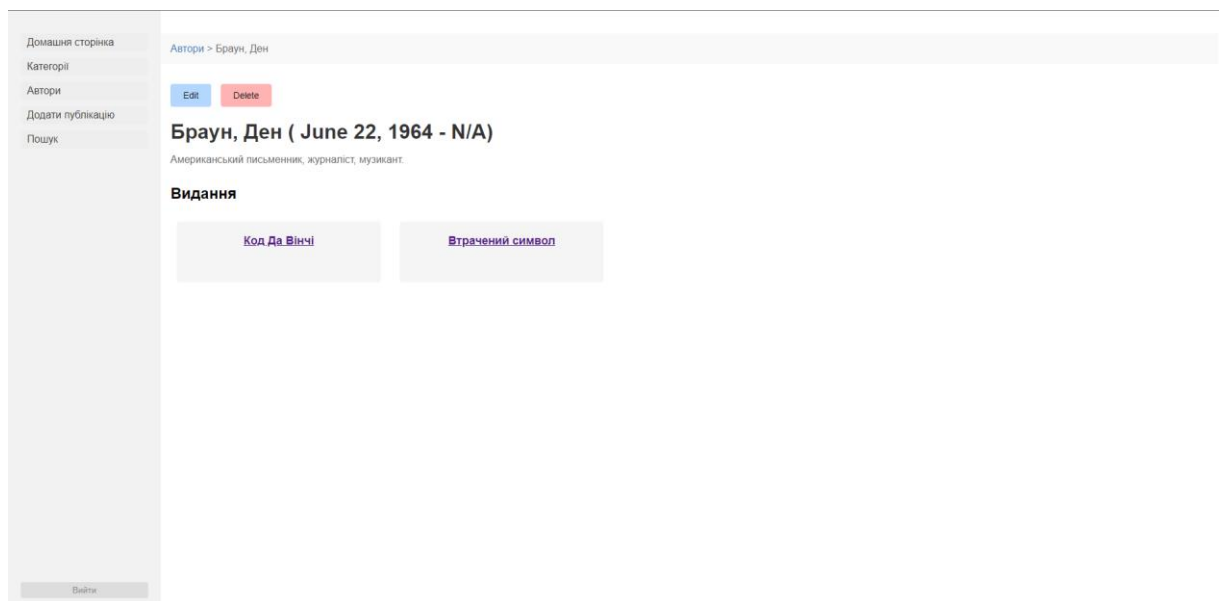


Рисунок 5.7 – Екран детальної інформації про автора

Тепер переглянемо вкладку «Додати публікацію» з основним функціоналом системи.

При натисканні на неї відкривається вікно з формою, де бібліотекарю надається можливість заповнити всі бібліографічні данні про видання та завантажити текстовий файл та файл з зображенням (за наявністю) (рисунок 5.8).

Домашня сторінка
Категорії
Автори
Додати публікацію
Пошук
Вийти

Нова публікація

Назва:

Автор:

Тип видання:

Короткий опис:

Дата публікації:

Видавництво:

Мова:

Зображення:

Файл:

Рисунок 5.8 – Екран заповнення бібліографічних даних про нове електронне видання

Після заповнення даних натискаємо кнопку «Далі». Відривається сторінка, де до заповнених бібліографічних даних додаються визначені системою ключові слова, які стосуються даного електронного видання та категорія, до якої можна віднести його. Бібліотекар, за бажанням, може ввести зміни, після цього натискає кнопку «Зберегти».

Після цього нове електронне видання з'являється у всьому переліку публікацій та у категорії, до якої його віднесли.

Останній модуль, який реалізований, - пошук видання за параметром.

На вкладці «Пошук» можна побачити поле, в якому можна ввести слово, за яким буде здійснюватися пошук та відповідний параметр, який можна обрати у дроп-лісті (рисунок 5.9). Після натискання кнопки «Пошук» відображається перелік всіх видань, які підходять під параметри пошуку.

Домашня сторінка
Категорії
Автори
Додати публікацію
Пошук

Пошук у каталозі бібліотеки

Search Категорія

Рисунок 5.9 – Екран пошуку електронних видань

Тепер опишемо, який функціонал є у адміністратора. Адміністратор так само може авторизуватись у системі (рисунок 5.10).

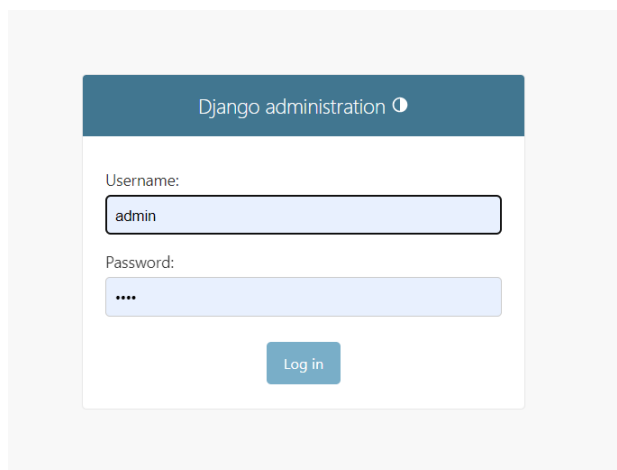


Рисунок 5.10 – Екран авторизації у системі адміністратора

Після авторизації йому відкривається адміністративна панель, на якій бачимо основні функції, які він має:

- 1) Додавання нового користувача (рисунок 5.11);
- 2) Редагування та видалення користувачів у системі (рисунок 5.12);
- 3) Додавання, редагування та видалення будь-яких позицій, які стосуються електронних видань та їх категорій (рисунок 5.13 – 5.14).

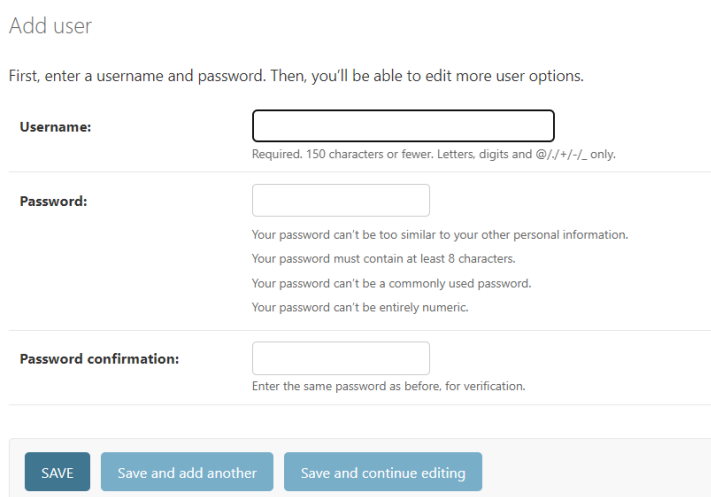


Рисунок 5.11 – Екран додавання нового користувача у системі

Change user

skaperenn HISTORY

Username:
Required: 150 characters or fewer: Letters, digits and @/./+/-/_ only.

Password: **algorithm: pbkdf2_sha256 iterations: 600000 salt: YVBH5H***** hash: CDO91*******
Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using this form.

Personal info

First name:

Last name:

Email address:

Рисунок 5.12 – Екран редагування даних про користувача у системі

CATALOG

Authors + Add Change

Categorys + Add Change

Keywords + Add Change

Languages + Add Change

Рисунок 5.13 – Екран деяких позицій, які є у базі даних електронних видань

Add author

First name:

Last name:

Date of birth: Today | Note: You are 3 hours ahead of server time.

Died: Today | Note: You are 3 hours ahead of server time.

Description:
Enter a brief description of the author

SAVE Save and add another Save and continue editing

Рисунок 5.14 – Екран додавання нового автора у базу даних

5.2 Випробування програмного продукту

5.2.1 Мета випробувань

Метою випробувань автоматизованої системи каталогізації та класифікації електронних видань у бібліотеці є перевірка її функціональності, надійності та відповідності вимогам, а також визначення її ефективності та придатності для використання.

Щоб досягти поставленої мети необхідно провести ряд тест-кейсів, які охоплюють функціонал розробленої автоматизованої системи.

5.2.2 Результати випробувань

Автоматизовану систему протестовано на відповідні функціональні вимоги. Тестування включало в собі перевірку роботи клієнтської та серверної частин, перевірку наявності та правильного функціонування бази даних.

Для початку перевіримо додавання нового користувача-бібліотекаря у систему та спроможність його увійти у систему. Відповідні тести наведені у таблицях 5.1 - 5.2.

Таблиця 5.1 – Тестування додавання нового користувача у систему

Тест	Перевірка додавання нового користувача у систему
Початковий стан системи	Адміністратор на сторінці додавання нового користувача
Вхідні дані	Дані (логін та пароль) користувача
Дія	- У полі «Login» ввести логін користувача англійськими літерами; - У полі «Password» ввести пароль; - У полі «Password confirmation» ввести пароль знову; - Натиснути кнопку «Save».

Продовження таблиці 5.1

Очікуваний результат	Створюється новий користувач.
Стан системи після тестування	У базі з'являється запис про нового користувача, користувач-бібліотекар тепер має змогу увійти у систему за допомогою даних, які йому надав адміністратор.

Таблиця 5.2 – Тестування авторизації бібліотекаря

Тест	Перевірка авторизації бібліотекаря
Початковий стан системи	Бібліотекар знаходиться на сторінці авторизації
Вхідні дані	Дані (логін та пароль) користувача
Дія	- У полі «Ім'я користувача» ввести логін користувач-бібліотекаря англійськими літерами; - У полі «Пароль» ввести пароль; - Натиснути кнопку «Увійти».
Очікуваний результат	Користувач-бібліотекар проходить авторизацію у системі.
Стан системи після тестування	Система перевіряє внесені дані та перенаправляє користувача-бібліотекаря на головну сторінку системи.
Очікуваний результат	Користувач-бібліотекар проходить авторизацію у системі.

Так як процес каталогізації та класифікації взаємопов'язані і являються частиною одного процесу – додавання нового видання у базу даних системи, то розділим випробування цих модулів на декілька тестів. Відповідні тести наведені у таблицях 5.3 – 5.4.

Таблиця 5.3 – Тестування процесу каталогізації електронного видання

Тест	Перевірка додавання нового електронного видання. Каталогізація
Початковий стан системи	Бібліотекар на сторінці додавання нового видання до каталогу
Вхідні дані	Бібліографічні дані видання та його файл
Дія	- Бібліотекар заповнює всі поля про видання (назва, автор тощо); - Бібліотекар завантажує файл з виданням до системи та його зображення (за наявністю); Бібліотекар натискає кнопку «Далі».
Очікуваний результат	Користувач-бібліотекар переходить до наступної сторінки.
Стан системи після тестування	Система починає процес визначення ключових слів та класифікації.

Таблиця 5.4 – Тестування класифікації та збереження даних

Тест	Перевірка додавання нового електронного видання. Класифікація та збереження даних
Початковий стан системи	Бібліотекар на сторінці класифікації
Вхідні дані	
Дія	- Бібліотекар перевіряє результат класифікації та визначення ключових слів, за необхідністю вносить зміни; - Бібліотекар натискає кнопку «Зберегти».
Очікуваний результат	До каталогу додається нове видання.
Стан системи після тестування	У вікні, де знаходиться каталог видань, відображається нове видання, у базі даних з'являється новий запис про видання.

Тепер перевіримо деякі функціональні вимоги, такі як оновлення даних про електронне видання, видалення електронного видання з каталогу та пошук видання за автором. Відповідні тести наведені у таблицях 5.5-5.7.

Таблиця 5.5 – Тестування оновлення даних про електронне видання

Тест	Перевірка оновлення даних про електронне видання
Початковий стан системи	Користувач-бібліотекар на сторінці з детальною інформацією про видання
Вхідні дані	Нові дані про видання
Дія	- Бібліотекар натискає кнопку «Редагувати»; - Бібліотекар обирає поле, яке хоче змінити (назву, автора тощо); - Бібліотекар вносить зміни; - Бібліотекар натискає кнопку «Зберегти зміни».
Очікуваний результат	Оновлення даних про видання.
Стан системи після тестування	Користувач знаходиться на У вікні з інформацією про видання оновлюються дані та відповідні зміни вносяться до бази даних.

Таблиця 5.6 – Тестування видалення електронного видання

Тест	Перевірка видалення електронного видання
Початковий стан системи	Користувач-бібліотекар на сторінці з детальною інформацією про видання
Вхідні дані	
Дія	- Бібліотекар натискає кнопку «Видалити»; - Бібліотекар підтверджує свій вибір.
Очікуваний результат	Видалення електронного видання з каталогу

Продовження таблиці 5.6

Стан системи після тестування	Користувач переходить до каталогу електронних видань. У вікні з каталогом зникає конкретне видання і користувач вже не може перейти та переглянути інформацію про нього.
-------------------------------	--

Таблиця 5.7 – Тестування пошуку видання за автором

Тест	Перевірка пошуку видання за автором
Початковий стан системи	Користувач-бібліотекар знаходиться на сторінці з каталогом електронних видань
Вхідні дані	Прізвище автора, видання якого необхідно знайти
Дія	- Обрати опцію пошуку за автором; - Ввести прізвище автора; - Натиснути кнопку «Пошук».
Очікуваний результат	Система надає перелік видань, які написав зазначений автор.
Стан системи після тестування	Користувач знаходиться на сторінці з переліком електронних видань, які написав зазначений у пошуку автор.

Висновки до розділу

У даному розділі дипломної роботи було розроблено та описано керівництво для користувачів системи. Воно містить інструкції, як користуватися автоматизованою системою каталогізації та класифікації електронних видань у бібліотеці.

В залежності від типу користувача (бібліотекар чи адміністратор) кожен має свій функціонал та опис дій, як взаємодіяти з автоматизованою системою. Для

бібліотекарів надано детальні інструкції з використання функціоналу, пов'язаного з каталогізацією та класифікацією електронних видань. Це включає створення нових записів про видання, редагування існуючих та інше. Для адміністраторів надано інструкції з керування користувачами, доступом до системи.

Крім того, у розділі були описані тестові сценарії для перевірки основних функцій системи, які були визначені у першому розділі дипломного проєкту. Ці сценарії включали в себе тестування різних функцій, таких як створення запису про нове видання, пошук, редагування даних про видання та інше. Після проведення випробувань були описані результати, що дозволило підтвердити, що система працює коректно та відповідає вимогам, поставленим у процесі розробки.

					ІС91.080БАК.004 ПЗ	Арк.
						56
Зм.	Лист	№ докум.	Підпис			

ВИСНОВКИ

На сьогоднішній день важливо розробляти нові методики та алгоритми класифікації та каталогізації електронних видань для забезпечення ефективного доступу до інформації в бібліотеках. Такі ресурси дозволяють не тільки полегшити пошук необхідних електронних ресурсів, а й підвищити якість послуг, які надаються користувачам бібліотекою.

Розробка системи каталогізації та класифікації тексту є складним, але цікавим завданням, яке вимагає використання різноманітних компетенцій та технологій. Головною метою проєкту є створення ефективного та автоматизованого інструменту, який допомагатиме організувати та структурувати великий обсяг текстової інформації.

В результаті проведеного аналізу аналогів та визначення функціональних вимог було з'ясовано, що подібні системи вже існують, але їх можна покращити шляхом врахування потреб користувачів та впровадження нових функціональностей. Враховуючи ці вимоги, сформульовано функціональні вимоги до системи, які описують необхідні функції та обмеження.

Розробка системи включає розробку моделі бази даних, а саме визначення основних сутностей, які зберігаються у таблицях, та їх параметрів, які є полями таблиць. Це дає змогу створити структуру, яка дозволяє зберігати та керувати електронними виданнями у системі.

Одним з головних аспектів проєкту є реалізація алгоритму TextRank для визначення ключових слів у електронних виданнях. Цей алгоритм успішно застосований після попередньої обробки тексту, такої як токенізація, видалення стоп-слів та нормалізація слів. Це дає змогу виділити найбільш інформативні елементи у тексті та поліпшити процес класифікації.

Для класифікації текстів використано Баєсівський класифікатор, що навчений на тренувальних даних. Це дозволяє системі ефективно визначати категорію тексту на основі його ключових слів та інших ознак.

У процесі реалізації системи проведено тестування, яке включає перевірку роботи кожного компонента, а також перевірку відповідності функціональним вимогам. В результаті тестування підтверджено, що система працює стабільно і ефективно, відповідно до заданих вимог.

Розроблена система має значний потенціал для застосування у різних галузях, де потрібно організувати та класифікувати текстову інформацію. Вона може бути використана у наукових дослідженнях, бібліотеках, видавництвах та інших областях, де важливо ефективно обробляти та аналізувати тексти.

Отже, мета та цілі, які поставлені на початку проєктування автоматизованої системи каталогізації та класифікації електронних видань у бібліотеці були досягнені під час реалізації дипломного проєкту.

					ІС91.080БАК.004 ПЗ	Арк.
						58
Зм.	Лист	№ докум.	Підпис			

ПЕРЕЛІК ПОСИЛАНЬ

1. Manning C. D., Raghavan, P., Schütze H. Introduction to Information Retrieval. Cambridge University Press. 2008.
2. Bilal D. Library Automation: Core Concepts and Practical Systems Analysis. 2014.
3. Ріжняк Р. Еволюція наукового обґрунтування автоматизації бібліотек вищих навчальних закладів України. Етнічна історія народів Європи. 2014. С. 207–215. [Електронний ресурс] – Режим доступу: http://nbuv.gov.ua/UJRN/eine_2014_42_33. 12.04.2023.
4. Jones K. S. A statistical interpretation of term specificity and its application in retrieval. MCB University Press, 2004. Vol. 60, No 5. P. 493—502.
5. Mihalcea R., Tarau P. TextRank: Bringing Order into Texts. [Електронний ресурс]. – Режим доступу: <https://aclanthology.org/W04-3252.pdf> 02.05.2023.
6. Rish I. An empirical study of the naive Bayes classifier. [Електронний ресурс] – Режим доступу: https://www.researchgate.net/publication/228845263_An_Empirical_Study_of_the_Naive_Bayes_Classifier. 25.04.2023.
7. Hand D.J., Yu K. Idiot's Bayes — not so stupid after all?. International Statistical Review. 2001. Vol. 69, No 3, P. 385—399.
8. Theodoridis S., Koutroumbas K. Pattern Recognition. 4th Edition. Academic Press. 2009. [Електронний ресурс] – Режим доступу: [https://darmento.akakom.ac.id/pengenalannya/Pattern%20Recognition%204th%20Ed.%20\(2009\).pdf](https://darmento.akakom.ac.id/pengenalannya/Pattern%20Recognition%204th%20Ed.%20(2009).pdf). 24.04.2023.
9. Miner G., Elder J., Hill T., Nisbet R., Delen D., Fast A. Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications. London: Academic Press. 2012. 1st Edition. P. 573-590.
10. Obe R., Hsu L. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. Sebastopol: O'Reilly Media. 2017. 3th edition.

					IC91.080БАК.004 ПЗ	Арк.
						59
Зм.	Лист	№ докум.	Підпис			

11. Müller A., Guido S. Introduction to Machine Learning with Python: A Guide for Data Scientists. Sebastopol: O'Reilly Media. 2016. 1st edition.
12. Dauzon S., Bendoraitis A., Ravindran A. Django: Web Development with Python. Packt Publishing. 2016. P. 400-425.
13. Bass L., Clements P., Kazman R. Software Architecture in Practice (SEI Series in Software Engineering). Boston: Addison-Wesley Professional. 2001. P. 220-245.
14. Shamile B. Software development with UML diagrams. Chicago: Independently published. 2022. P.50.
15. Gopalaswamy R, Desikan S. Software Testing: Principles and Practice. Canada: Pearson Education Canada. 1st edition. 2019. P. 128-155.

					IC91.080БАК.004 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис			

ДОДАТОК А

Програмний код

<https://github.com/kovlskaa/diploma>

