

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка вебсайту електронної бібліотеки  
Тернопільської ЗОШ № 27

Виконав: студент IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

(підпис)

Савків О.І.

(прізвище та ініціали)

Керівник

(підпис)

Ясній О.П.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Марценко С.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

Яцишин В.В.

(прізвище та ініціали)

Тернопіль  
2024

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(підпис) (прізвище та ініціали)

«\_\_» червня 2024 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Савків Олегу Ігоровичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка вебсайту електронної бібліотеки Тернопільської ЗОШ №27

Керівник роботи Ясній Олег Петрович д.т.н., професор кафедри МН  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «29» квітня 2024 року № 4/7-470

2. Термін подання студентом завершеної роботи 26 червня 2024р.

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. Розділ 1. Аналіз предметної області та постановка завдання. 1.1 Аналіз предметної області. 1.2 Вимоги до функціональних можливостей електронної бібліотеки. 1.3 Аналітичний огляд існуючих рішень. 1.4 Поняття та класифікація вебсайтів. 1.5 Вибір технологій та оптимального середовища для розробки вебсайту. 1.5.1 HTML,CSS та JavaScript 1.5.2 PHP 1.5.3 Вибір бази даних для електронної бібліотеки 1.5.4 Редактор коду. 1.6 Висновки щодо першого розділу. Розділ 2. Проектування архітектури та моделей електронної бібліотеки 2.1 Вибір архітектурного підходу. 2.2 Моделювання архітектури вебсайту електронної бібліотеки школи. 2.3 Визначення з системою управління базами даних. 2.4 Розробка концептуальної моделі бази даних. 2.5 Розробка логічної моделі бази даних. Визначення сутностей та їх атрибутів. 2.6 Висновки щодо другого розділу. Розділ 3. Реалізація, тестування та впровадження вебсайту електронної бібліотеки. 3.1 Реалізація інтерфейсу користувача 3.2 Тестування функціоналу. 3.3 Впровадження вебсайту 3.4 Висновки щодо третього розділу. Розділ 4. Безпека життєдіяльності, основи охорони праці. 4.1 вплив діяльності людини на довкілля. 4.2 Особливості безпеки праці під час вантажно-розвантажувальних робіт. 4.3 Висновки щодо 4 розділу. Висновки. Перелік джерел. Додаток А. Додаток Б. Додаток В. 5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Сенчишин В.С., доцент кафедри МТ	12.06.2024	15.06.2024

7. Дата видачі завдання 29 січня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	30.01.2024	
2.	Підбір джерел про створення вебсайтів та їх проектування	31.01.2024-03.02.2024	
3.	Опрацювання джерел по темі кваліфікаційної роботи	04.02.2024-06.02.2024	
4.	Виконання дослідження щодо розробки вебсайтів	07.02.2024-11.02.2024	
	Розроблення моделей класів та архітектури вебсайту		
5.	Оформлення розділу «Аналіз предметної області та постановка завдання»	03.06.2024-05.06.2024	
6.	Оформлення розділу «Проектування архітектури та моделей електронної бібліотеки»	06.06.2024-08.06.2024	
6.	Оформлення розділу «Реалізація, тестування та Впровадження веб сайту»	09.06.2024-11.06.2024	
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»	12.06.2024-13.06.2024	
8.	Виконання завдання до підрозділу «Основи охорони праці»	14.06.2024-15.06.2024	
9.	Оформлення кваліфікаційної роботи	16.06.2024-17.06.2024	
10.	Нормоконтроль	18.06.2024-19.06.2024	
11.	Перевірка на плагіат	20.06.2024	
12.	Попередній захист кваліфікаційної роботи	21.06.2024	
13.	Захист кваліфікаційної роботи	27.06.2024	

Студент

---

  
(підпис)

Савків О.І.

---

  
(прізвище та ініціали)

Керівник роботи

---

  
(підпис)

Ясній О.П.

---

  
(прізвище та ініціали)

## АНОТАЦІЯ

Розробка веб сайту електронної бібліотеки Тернопільської ЗОШ № 27 // Кваліфікаційна робота освітнього рівня «Бакалавр» // Савків Олег Ігорович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2024 // С.78, рис. – 25, табл. – 1, бібліогр. – 35, додат. – 3.

Ключові слова: електронна бібліотека, веб-додаток, база даних, PHP, MySQL, Тернопільська ЗОШ № 27, розробка, тестування, проектування.

Кваліфікаційна робота присвячена дослідженню розробки електронної бібліотеки для Тернопільської ЗОШ № 27. В першому розділі кваліфікаційної роботи описано аналіз предметної області та існуючих рішень для розробки електронної бібліотеки. Висвітлено проблеми та недоліки існуючих систем, такі як незручний інтерфейс, недостатня динамічність та затримки у відповіді. Проаналізовано переваги та недоліки різних технологій та інструментів, які можуть бути використані для розробки електронної бібліотеки.

В другому розділі кваліфікаційної роботи досліджено архітектуру та моделі даних для електронної бібліотеки, визначено оптимальні рішення для її реалізації. Обґрунтовано вибір технологій для створення серверної та клієнтської частин системи, зокрема PHP, MySQL, HTML, CSS та JavaScript. Подано концептуальну та логічну моделі бази даних, визначено основні сутності та їх атрибути, що дозволяють зберігати необхідну інформацію.

В третьому розділі кваліфікаційної роботи описано розробку прототипу електронної бібліотеки, включаючи всі необхідні функціональні модулі. Проаналізовано методи оптимізації роботи системи для забезпечення швидкого доступу до інформації та мінімізації затримок у відповідях. Проведено тестування розробленої системи в умовах реальної експлуатації.

Об'єкт дослідження: процес розробки електронної бібліотеки для Тернопільської ЗОШ № 27.

Предмет дослідження: технології та методи, використані для створення електронної бібліотеки.

## ANNOTATION

Website Development of the Electronic Library of Ternopil Secondary School No. 27 // Qualification work of the educational level "Bachelor" // Savkiv Oleh Ihorovych // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2024 // P. 78 , fig. - 25, tabl. - 1 , annexes. – 3, references - 35 .

Keywords: electronic library, web application, database, PHP, MySQL, Ternopil Secondary School No. 27, development, testing, design.

This bachelor's thesis is dedicated to the development of an electronic library for Ternopil Secondary School No. 27. The first chapter of the thesis describes the analysis of the subject area and existing solutions for developing an electronic library. It highlights the problems and shortcomings of existing systems, such as inconvenient interfaces, insufficient dynamism, and response delays. The advantages and disadvantages of various technologies and tools that can be used to develop.

The second chapter of the thesis explores the architecture and data models for the electronic library, identifying optimal solutions for its implementation. The choice of technologies for creating the server-side and client-side parts of the system, including PHP, MySQL, HTML, CSS, and JavaScript, is substantiated. Conceptual and logical models of the database are provided, defining the main entities and their attributes that allow storing the necessary information.

The third chapter of the thesis describes the development of a prototype for the electronic library, including all necessary functional modules. Methods for optimizing the system's operation to ensure quick access to information and minimize response delays are analyzed.

Object of research: The process of developing an electronic library for Ternopil Secondary School No. 27.

Subject of research: The technologies and methods used to create the electronic library.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ЗОШ – Загальноосвітня школа.

БД – База даних.

ІТ – Інформаційні технології.

ПЗ – програмне забезпечення.

СУБД – система управління базами даних.

HTTP (англ. HyperText Transfer Protocol) – протокол передачі гіпертексту.

URL (англ. Uniform Resource Locator) – уніфікований локатор ресурсів.

API (англ. Application Programming Interface) – інтерфейс програмування додатків.

HTML (англ. HyperText Markup Language) – мова розмітки гіпертексту.

CSS (англ. Cascading Style Sheets) – каскадні таблиці стилів.

JS (англ. JavaScript) – мова програмування JavaScript.

PHP (англ. Hypertext Preprocessor) – скриптова мова загального призначення, яка широко використовується для розробки веб-додатків і веб-сайтів.

MySQL (англ. My Structured Query Language) – система керування реляційними базами даних з відкритим вихідним кодом.

SQL (англ. Structured Query Language) – структурована мова запитів.

Валідація – процес перевірки HTML-коду сайту на помилки і його відповідність стандартам.

Тестування – процес перевірки програмного забезпечення для виявлення помилок та перевірки відповідності вимогам.

## ЗМІСТ

ВСТУП .....	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ .....	10
1.1 Аналіз предметної області .....	10
1.2 Вимоги до функціональних можливостей електронної бібліотеки...	10
1.3 Аналітичний огляд існуючих рішень.....	11
1.4 Поняття та класифікація вебсайтів.....	13
1.5 Вибір технологій та оптимального середовища для розробки вебсайту.....	17
1.5.1 HTML, CSS та JavaScript .....	17
1.5.2 PHP.....	19
1.5.3 Вибір бази даних для електронної бібліотеки.....	19
1.5.4 Редактор коду .....	20
1.6 Висновки щодо першого розділу .....	21
РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛЕЙ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ.....	23
2.1 Вибір архітектурного підходу .....	23
2.2 Моделювання архітектури вебсайту електронної бібліотеки школи	27
2.3 Визначення з системою управління базами даних .....	31
2.4 Розробка концептуальної моделі бази даних .....	33
2.5 Розробка логічної моделі бази даних. Визначення сутностей та їх атрибутів .....	36
2.6 Висновки щодо другого розділу .....	41
РОЗДІЛ 3. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ВЕБСАЙТУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ .....	43
3.1 Реалізація інтерфейсу користувача .....	43
3.2 Тестування функціоналу .....	48
3.3 Впровадження вебсайту .....	53

3.4 Висновки щодо третього розділу .....	55
<b>РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ</b>	<b>56</b>
4.1 Вплив діяльності людини на довкілля .....	56
4.2 Особливості безпеки праці під час вантажно-розвантажувальних робіт .....	59
4.3 Висновок щодо 4 розділу .....	62
<b>ВИСНОВКИ</b> .....	<b>63</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ</b> .....	<b>65</b>



## ВСТУП

**Актуальність теми.** У сучасному світі стрімкий розвиток інформаційних технологій відкриває нові можливості для навчальних закладів, сприяючи підвищенню якості освіти та оптимізації навчальних процесів. Електронні бібліотеки, як невід'ємна частина цих змін, відіграють важливу роль у забезпеченні доступу до навчальних матеріалів, полегшуючи пошук і використання інформації як для учнів, так і для вчителів. Саме тому розробка електронної бібліотеки є актуальним завданням, що потребує ретельного дослідження та втілення.

Розробка сайту «Електронна бібліотека» Тернопільської загальноосвітньої школи № 27 є важливою з кількох причин. Він відповідає зростаючому попиту на цифрові освітні ресурси та забезпечує централізовану платформу, де студенти та викладачі можуть отримати доступ до великої кількості інформації будь-коли та будь-де. Цей проект не тільки покращує досвід навчання, але й заохочує самостійне навчання та дослідження, сприяючи культурі академічної досконалості та інновацій у школі.

Крім того, Електронна бібліотека служить моделлю для інших навчальних закладів, які прагнуть застосувати цифрові рішення для розширення своїх пропозицій ресурсів. Документуючи процес і результати цього проекту, ця робота має на меті зробити внесок у ширшу сферу освітніх технологій, пропонуючи ідеї та практичні рекомендації для подібних ініціатив.

**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи є розробка вебсайту електронної бібліотеки для Тернопільської загальноосвітньої школи №27 ім. Віктора Гурняка. Це дозволить створити сучасний, зручний та ефективний інструмент для організації навчального процесу, забезпечивши учнів і викладачів швидким доступом до навчальних ресурсів.

Завданнями кваліфікаційної роботи є:

- аналіз існуючих рішень у сфері електронних бібліотек;
- визначення функціональних і нефункціональних вимог до системи;

- проектування архітектури веб-сайту та його складових частин;
- розробка інтерфейсу користувача та серверної частини системи;
- інтеграція та тестування веб-сайту;
- впровадження системи у навчальний процес школи та забезпечення її подальшої підтримки.

Об'єктом дослідження є процес створення електронної бібліотеки, яка відповідатиме потребам учнів і викладачів Тернопільської середньої школи №27. Предметом дослідження є методи та технології веб-розробки, що використовуються для створення електронних бібліотек.

Методологічна основа дослідження включає в себе аналіз наукових джерел, порівняльний аналіз існуючих електронних бібліотек, використання сучасних технологій веб-розробки, а також методи тестування та впровадження програмного забезпечення.

Розробка сайту «Електронна бібліотека» для Тернопільської загальноосвітньої школи I-III ступенів №27, це прогресивний проект, який втілює в собі інтеграцію технологій в освіту. Надаючи цифрову платформу для доступу до освітніх ресурсів, цей проект спрямований на покращення результатів навчання, підтримку зусиль викладання та сприяння прогресивному освітньому середовищу. Ця кваліфікаційна робота слугуватиме детальним посібником та орієнтиром для майбутніх проектів у сфері розвитку освітніх технологій.

## **РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ**

### **1.1 Аналіз предметної області**

Аналіз предметної області є важливим кроком у розробці електронної бібліотеки для Тернопільської ЗОШ № 27 ім. Віктора Гурняка, оскільки він дозволяє визначити основні вимоги та функціональні можливості системи, які будуть задовольняти потреби користувачів.

Цільова аудиторія електронної бібліотеки складається з чотирьох груп користувачів, це учні, викладачі та бібліотекарі школи й також адміністратор системи.

Учні школи потребують доступу до навчальних матеріалів, книг, підручників та додаткових джерел інформації для підготовки до уроків та самостійного навчання.

Вчителі використовують електронну бібліотеку для підготовки до уроків, пошуку навчальних матеріалів та методичних посібників, а також для створення навчальних програм.

Бібліотекарі відповідальні за управління бібліотечним фондом, додавання нових матеріалів, оновлення існуючих записів та обслуговування запитів користувачів.

Адміністратори системи забезпечують технічну підтримку, адміністрування системи та безпеку даних.

### **1.2 Вимоги до функціональних можливостей електронної бібліотеки**

Основні процеси та функції, які повинні бути реалізовані в електронній бібліотеці, включають:

– управління контентом, а саме додавання, редагування та видалення книг, підручників, статей та інших навчальних матеріалів;

- каталогізація та пошук, створення структури бібліотеки, категоризація матеріалів за предметами, авторами, роком видання тощо, а також забезпечення ефективного пошуку та фільтрації контенту;
- забезпечення зручного доступу користувачів до матеріалів, можливість завантаження, читання онлайн та збереження в особистому кабінеті;
- створення та управління профілями користувачів, відстеження їх активності та надання персоналізованих рекомендацій;
- безпека та конфіденційність, захист особистих даних користувачів, обмеження доступу до певних матеріалів та забезпечення безпеки системи в цілому.

### **1.3 Аналітичний огляд існуючих рішень**

Стрімкий розвиток веб-технологій привів того, що організації мають власні сайти, проте це стосується в основному тільки країн з високо розвинутою або бурхливо економікою, що розвивається. Так Всесвітній економічний форум(WEF) представив звіт про розвиток інформаційних технологій в 134 країнах світу. Рівень розвитку інформаційних технологій(надалі ІТ) виражається в індексі мережевої готовності (The Networked Readiness). Проте слід враховувати, що це лише умовний показник, який має суттєві коливання всередині однієї країни.

Сайти-аналоги електронних бібліотек:

- онлайн бібліотека LoveRead (<http://loveread.ec/>);
- онлайн бібліотека Book Online (<http://book-online.com.ua/>);
- онлайн бібліотека BooksOnline (<https://booksonline.com.ua/>).

Електронна бібліотека LoveRead, що зображена на рисунку 1.1, це електронна бібліотека з великим переліком книг. З переваг можна відмітити сортування по авторах та книга. А також система авторизації. Проте у даний сайт слабо-динамічний та дизайн доволі незручний для користувача.



Рисунок 1.1 – Вебсайт LoveRead

Електронна бібліотека Book Online зображено на рисунку 1.2, вебсайт бібліотеки, на якому можна переглянути списки книг або почати читати їх. Основною перевагою даного сайту є мінімалістичний стиль та система останніх дій(додавання книг та відгуків). З недолік можна підмітити що він також мало-динамічний та погана адресація при переході на інші сторінки.



Рисунок 1.2 – Вебсайт Book Online

Електронна бібліотека Books Online, що зображена на рисунку 1.3, це вебсайт, на якому можна почитати книги. Перевагою даного сайту те, що у

любого користувача є можливість додати свою книгу. Проте головним недоліком є його час затримки відповіді.

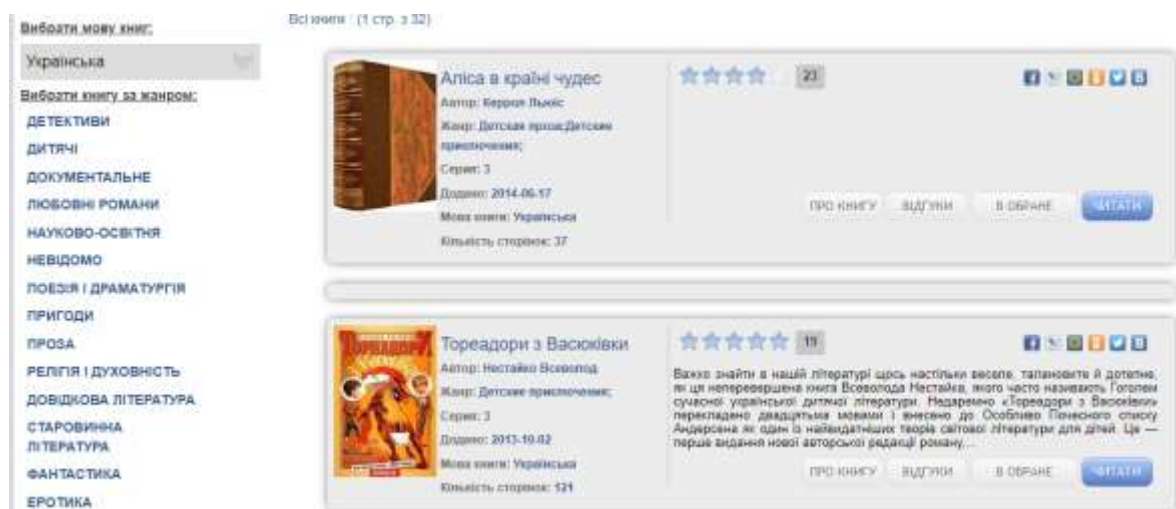


Рисунок 1.3 – Вебсайт Books Online

Підводячи підсумки, потрібно підмітити що проект має містити простий, чіткий та зрозумілий контент, який підходить для аудиторії різного віку. Також проект має містити хорошу оптимізацію, щоб користувачу не потрібно було довго чекати. У запланованого сайту повинна бути система авторизації.

#### 1.4 Поняття та класифікація вебсайтів

Вся інформацію, з якою можуть працювати користувачі Інтернету, зберігається на ПК (серверах). На серверах встановлено відповідне програмне забезпечення(надалі ПЗ). Більшість даної інформації перетворена на веб-сайти. Веб сервіс обов'язково має мати своє ім'я (адреса за доменом) в Internet.

Сайт – це сукупність сторінок, які об'єднані за змістом та навігацією(за умови, що вони під одним доменним ім'ям), доступних у мережі Інтернет, представлені користувачеві у відповідному вигляді, які розташовані на сайті і мають свою адресу. Фізично сайт може розміщуватися як на одному, так і на декількох серверах. Для отримання інформації з веб-сайтів користувачеві необхідно інстальовати браузер, які дають можливість переглядати контент на

сайті. Відповідно від того, який сайт потрібно знайти, необхідно ввести адрес веб-сайту в рядку пошуку браузера, та через деякий час буде завантажено у браузерне вікно потрібну інформацію.

В більшості випадків, сайт реалізується через зв'язаних між собою сторінках. Сторінка сайту є файлом з розширенням .html, який зберігає певну інформацію і спеціальний набір команд HTML-коду. Всі графічні, аудіо-файли та відео-файли відтворюються на веб-сайті, проте вони не завантажуються при кожному перезапуску веб-сторінки. Мова розмітки HTML містить не самі файли, а лише вказівки на самі файли з даними.

Для знаходження потрібного сайту, необхідно ввести його адресу, зображено на рисунку 1.4. Всі сайти у веб-просторі обов'язково мають мати свій Internet-адрес, який містить саму адресу веб-сайту і домену, по якому можна дізнатись деяку інформацію по даній сторінці.

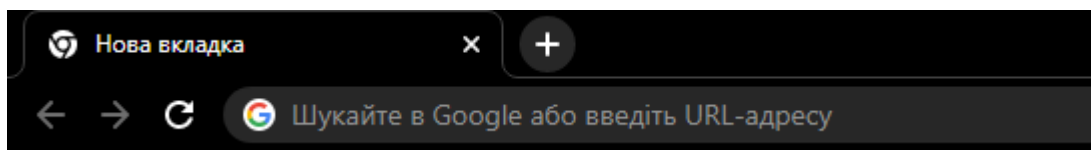


Рисунок 1.4 – Зовнішній вигляд адресного рядка

Веб-сервіси за доступністю поділяються на відкриті, напіввідкриті та закриті. Відкриті сервіси – це сервіси, які доступні для будь-яких користувачів, тобто не потребують зайвих дій для отримання доступу до інформації. На половину відкриті сервіси – це сервіси, на яких потрібно зареєструватись, щоб отримати повний доступ до інформації. Закриті сервіси призначені для використання приватними особами. Тобто про такі сайти знають обмежена кількість людей. Доступ нових користувачів до закритого сайту зазвичай відбувається через інвайт-посилання.

Веб-сервіси поділяються на динамічні та статичні, зображено у таблиці 1.1. Інформацію не потрібно підготовлювати, вона створюється відповідними скриптами. Динамічні сайти отримали велику перевагу над статичними сайтами

через можливість працювати з великим обсягом даним та можливістю розвивати сайт, відповідно до потреб з часом. На статичних веб-сайтах вся інформація створена відповідно строго плану. Відвідувачу відображається інформація у тому вигляді, у якому вона була збережена на сайті. На відмінну від статичних сайтів, динамічні веб-сервіси є більш зручніші.

Таблиця 1.1 – Відмінності динамічних та статичних вебсайтів

<b>Параметри</b>	<b>Динамічні сайти</b>	<b>Статичні сайти</b>
<b>Визначення</b>	Сайти, контент яких генерується в режимі реального часу за запитом користувача.	Сайти, контент яких фіксований та не змінюється після завантаження.
<b>Технології</b>	Використовують серверні скрипти та бази даних	Використовують статичні файли, які зберігаються на сервері без змін.
<b>Гнучкість та масштабованість</b>	Легко адаптуються до потреб користувачів, дозволяють інтерактивність, персоналізацію та роботу з базами даних.	Швидкість завантаження завдяки простоті і мінімальній кількості запитів до сервера.
<b>Обслуговування контенту</b>	Зручне управління великим обсягом даних та контенту, можливість автоматизації оновлень.	Легкість у розробці та підтримці, не потребують складних серверних налаштувань та баз даних.
<b>Функціональні можливості</b>	Підтримують складні функції, такі як системи управління контентом (CMS), інтернет-магазини, форуми тощо.	Висока безпека через відсутність серверних скриптів, менша вразливість до хакерських атак.
<b>Ресурсоміткість</b>	Потребують більше ресурсів сервера для обробки запитів та генерації контенту.	Обмеженість у функціональності та інтерактивності, потреба в ручному оновленні контенту.



Продовження таблиці 1.1

<b>Безпека</b>	Вразливі до різних видів атак (SQL-ін'єкції, XSS) через складність та взаємодію з базами даних і скриптами.	Менша вразливість до серверних атак через відсутність баз даних та серверної логіки.
<b>Вартість</b>	Вищі витрати на розробку та підтримку через складність та необхідність використання серверних технологій.	Нижчі витрати на хостинг і обслуговування завдяки простоті структури та мінімальним вимогам до сервера.

Також вебсайти поділяються за розташуванням. Вони поділяються на сайти з локальною мережею та загальнодоступні сайти, які знаходяться в просторах Інтернету. Доступ до загальнодоступних сайтів у просторі Internet може отримати любий користувач.

Сайти за схемою подання інформації та її обсягом поділяються на сайти-візитки, корпоративні сайти та інтернет-магазини. Сайт-візитка відображає загальні дані про роботу власника сайту(рідше роботу організації). Працівники на такому сайті можуть розміщувати власне резюме. З даної інформації можна зробити висновок, що сайт-візитка це докладна візитна карта в просторах Інтернету. На відмінну від сайту-візитки, корпоративний сайт має містити повну інформацію про діяльність компанії, його власника та продукцію або послуги, які пропонує дана організація. Часто такі сайти містять додаткові функціональні інструменти для роботи з інформацією (пошук, сортування, фотогалереї, календарі подій). Також може мати закриті розділи, відповідно до рівня доступу до інформації працівника.

Отже, розробка сайту включає такі стадії:

- затвердження технічного завдання розробки сайту;
- визначення структурної схеми сайту, розташування розділів, контенту і навігації;

- вебдизайн - створення графічних елементів макету сайту, стилів і елементів навігації;
- тестування програмного коду, модулів, бази даних і інших елементів сайту необхідних в проекті;
- розміщення сайту в мережі Інтернет.

## **1.5 Вибір технологій та оптимального середовища для розробки вебсайту**

Вибір оптимального середовища для розробки вебсайту є критичним етапом, оскільки він визначає технологічну базу, інструменти та платформи, які будуть використовуватися в процесі створення електронної бібліотеки. Враховуючи вимоги проекту та аналіз існуючих технологій, було прийнято рішення використовувати наступні технології: HTML, CSS та JavaScript для фронтенду, PHP для бекенду, MySQL для управління базами даних. Такий підхід забезпечує надійність, масштабованість та ефективність вебсайту.

### **1.5.1 HTML, CSS та JavaScript**

HTML (HyperText Markup Language) є основною мовою розмітки для створення структури вебсторінок. Вона дозволяє визначити елементи вебсторінки, такі як заголовки, абзаци, посилання, зображення та інші медіа. HTML забезпечує базову структуру вебсторінки, яка може бути стилізована та функціоналізована за допомогою CSS та JavaScript відповідно [1-2]. HTML є простим у використанні та вивчені, він підтримується усіма веб браузерями а також містить велику кількість ресурсів та документацій.

CSS (Cascading Style Sheets) використовується для оформлення та дизайну вебсторінок. Вона дозволяє розділити контент і презентаційний шар, забезпечуючи чистоту і підтримуваність коду. CSS визначає стилі для HTML елементів, включаючи кольори, шрифти, відступи, макети та інші аспекти

вигляду вебсторінок [3-4]. CSS має низку переваг, він гнучкий у стилізації та дизайні вебсторінок, дозволяє реалізувати адаптивність дизайну до різних пристроїв, також має підтримку анімацій та трансформації.

Для розробки вебсайту електронної бібліотеки для школи, було підключено фреймворк Bootstrap.

Bootstrap це найпопулярніший фреймворк для створення адаптивних і мобільних вебсайтів. Він містить набір інструментів для розробки інтерфейсів, включаючи готові CSS-класи та JavaScript-компоненти для оформлення форм, кнопок, навігаційних меню та інших елементів. Bootstrap дозволяє швидко створювати сучасні та зручні інтерфейси з мінімальними витратами часу[5].

JavaScript є мовою програмування, яка надає функціональність і інтерактивність вебсторінкам. Вона дозволяє реалізовувати динамічні елементи, обробку подій користувача, взаємодію з сервером та багато іншого[6-7]. JavaScript є основною мовою для розробки інтерактивних вебзастосунків.

Також було використано jQuery, популярну JavaScript бібліотеку, яка значно спрощує розробку вебсайтів[8-9]. Вона надає зручний API для роботи з елементами DOM, обробки подій, анімацій та взаємодії з сервером через AJAX. Використання jQuery дозволяє зменшити обсяг коду та підвищити його читабельність і підтримуваність.

Переваги jQuery:

- простота використання та зменшення обсягу коду;
- сумісність з усіма основними браузерами;
- велика кількість плагінів та розширень.

Використання цих технологій для фронтенду є стандартним підходом у веброботці, що дозволяє створювати сучасні, зручні та швидкі інтерфейси для користувачів.

## 1.5.2 PHP

PHP (Hypertext Preprocessor) є популярною серверною мовою програмування, яка спеціально розроблена для створення динамічних вебсторінок. Вона інтегрується з HTML, що дозволяє легко впроваджувати серверний код у розмітку сторінок. PHP є відкритим кодом, що робить його доступним і широко використовуваним серед розробників. Крім того, він підтримується більшістю вебхостингів, що забезпечує його широке розповсюдження та надійність[10-11].

Для розробки вебсайту електронної бібліотеки Тернопільської ЗОШ № 27 PHP стане оптимальним вибором з наступних причин:

- використовуючи PHP, можна легко створювати модульні системи, які дозволять додавати нові функціональні можливості до вебсайту в майбутньому;
- мова програмування PHP забезпечує ефективну роботу з базами даних, що є критично важливим для електронної бібліотеки, яка міститиме велику кількість книг та інших ресурсів;
- використовуючи сучасні фреймворки та бібліотеки, можна забезпечити високий рівень безпеки вебсайту, захищаючи його від поширених загроз.

## 1.5.3 Вибір бази даних для електронної бібліотеки

Вибір бази даних є важливим аспектом у розробці будь-якого вебсайту, оскільки вона відповідає за зберігання та управління даними. Для електронної бібліотеки необхідна надійна, швидка та масштабована система управління базами даних (СКБД), яка забезпечить зберігання інформації про книги, користувачів та інші ресурси. У кваліфікаційній роботі, я обрав MySQL як оптимальну СКБД для вебсайту електронної бібліотеки Тернопільської ЗОШ № 27.

MySQL – це одна з найпопулярніших реляційних систем керування базами даних, яка широко використовується у веб-розробці. MySQL відома своєю високою продуктивністю та швидкістю обробки запитів, що робить її ідеальною для вебсайтів, які потребують швидкої взаємодії з базою даних. Крім того, MySQL забезпечує надійне зберігання даних та їх відновлення у випадку збоїв, що є критично важливим для безперебійної роботи електронної бібліотеки.

Велика та активна спільнота розробників MySQL забезпечує постійне оновлення, підтримку та наявність великої кількості документації, що спрощує розробку та усунення можливих проблем. База даних легко інтегрується з PHP, що дозволяє створювати динамічні вебсайти та додатки. PHP має вбудовані функції для роботи з MySQL, що спрощує процес розробки. Крім цього, вона підтримує масштабування як вертикальне (збільшення потужності серверу), так і горизонтальне (додавання нових серверів), що дозволяє обробляти зростаючі обсяги даних та трафіку.

Для вебсайту електронної бібліотеки Тернопільської ЗОШ № 27 MySQL стане оптимальним вибором завдяки своїм характеристикам. Її використання дозволяє ефективно організувати зберігання даних про книги, авторів, жанри, а також інші бібліотечні ресурси, забезпечуючи швидкий доступ до них. MySQL забезпечує зручне зберігання даних про користувачів бібліотеки, їхні профілі, історію відвідувань та взяття книг. Також, використовуючи MySQL, можна реалізувати швидкий пошук та фільтрацію книг за різними параметрами, що покращить користувацький досвід[12-13].

#### **1.5.4 Редактор коду**

Visual Studio Code (VS Code) - це потужний, безкоштовний редактор коду, розроблений компанією Microsoft, який надає численні функції для веброзробників, доступний для операційних систем Windows, macOS і Linux. Програма має багату екосистему розширень для інших мов і середовищ виконання [14].

Використання VS Code для розробки вебсайту надає наступні переваги:

- велика кількість розширень для різних мов програмування, включаючи HTML, CSS, JavaScript, PHP та MySQL, які додають функціональність, покращують автозаповнення та допомагають у відлагодженні коду;
- можливість запускати команди та скрипти безпосередньо з редактора, що спрощує робочий процес і зменшує потребу у перемиканні між різними програмами;
- розширені можливості автозаповнення коду та IntelliSense допомагають швидко писати код з меншою кількістю помилок.

Інтеграція HTML, CSS, JavaScript з PHP та MySQL забезпечує створення повноцінного вебсайту з динамічним контентом та інтерактивними функціями. Такий підхід дозволяє розділити відповідальності між фронтенд та бекенд частинами системи, забезпечуючи гнучкість і масштабованість рішення.

## **1.6 Висновки щодо першого розділу**

Аналіз предметної області та існуючих рішень дозволив визначити основні вимоги та функціональні можливості електронної бібліотеки. Вибір технологій для розробки забезпечить надійність, масштабованість та ефективність системи, що задовольнить потреби користувачів.

Проведений аналіз показав, що існуючі аналоги електронних бібліотек мають низку недоліків, таких як незручний інтерфейс, мала динамічність, погана адресація при переході на інші сторінки, та затримки у відповіді. Це підкреслює важливість створення такої системи, яка б уникала подібних проблем і надавала користувачам зручний та інтуїтивно зрозумілий інтерфейс. Крім того, система повинна забезпечувати швидкий та надійний доступ до інформації, з мінімальними затримками та високою швидкістю обробки запитів.

Важливим аспектом розробки є забезпечення безпеки та конфіденційності даних користувачів. Вибір технологій, таких як PHP для серверної частини та

MySQL для керування базами даних, дозволить реалізувати ефективні механізми захисту даних та забезпечити стабільну роботу системи. Використання PHP також дозволить інтегрувати різні функціональні модулі та розширити можливості системи у майбутньому.

Застосування HTML, CSS та JavaScript для фронтенд-розробки дозволить створити адаптивний дизайн, який буде зручним для користувачів незалежно від пристрою, яким вони користуються. Підключення фреймворку Bootstrap забезпечить додаткові можливості для створення мобільних та адаптивних інтерфейсів, що є критично важливим у сучасному вебдизайні.

Вибір редактора коду Visual Studio Code обумовлений його потужними функціональними можливостями, зокрема підтримкою численних розширень та інтеграцією з різними мовами програмування. Це значно спрощує процес розробки та дозволяє швидко та ефективно впроваджувати необхідні зміни.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ АРХІТЕКТУРИ ТА МОДЕЛЕЙ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

### 2.1 Вибір архітектурного підходу

Вибір архітектурного підходу є ключовим аспектом при проектуванні системи, оскільки він визначає структуру, взаємодію компонентів та загальну надійність системи. Для розробки електронної бібліотеки Тернопільської ЗОШ №27, розглянемо декілька популярних архітектурних підходів, серед яких клієнт-серверна архітектура, мікросервісна архітектура та монолітна архітектура. Й на основі аналізу вимог проекту, особливостей кожного підходу та технологічних можливостей, оберемо оптимальну архітектуру для створення вебсайту електронної бібліотеки.

Монолітна архітектура для розробки веб-додатків, це підхід, коли вся програма будується як єдине неподільне ціле, зображено на рисунку 2.1. Це включає в себе всі компоненти програми, а саме інтерфейс користувача, бізнес-логіку, доступ до даних і будь-яку інтеграцію із зовнішніми системами, об'єднані в одну кодову базу та розгорнуті разом [15].

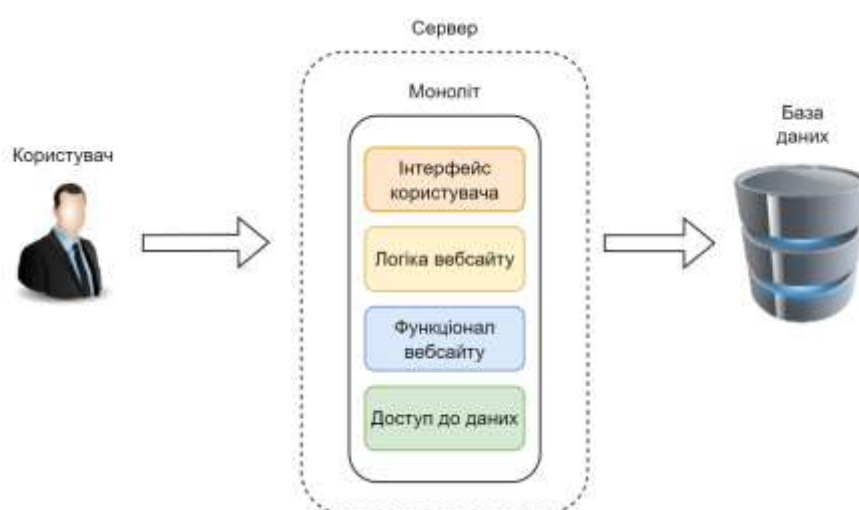


Рисунок 2.1 Монолітна архітектура

Як і будь-яка архітектура, монолітна має ряд переваг і недоліків.



Монолітна архітектура забезпечує простоту, оскільки процеси розробки та розгортання прості, і все розміщено в одному місці. Продуктивність можна підвищити за рахунок прямих викликів в одному контексті програми, які є швидшими, ніж мережеві виклики між мікросервісами. Швидкість розробки виграє від меншої кількості частин, що дозволяє швидше створювати програми. Тестування легше, оскільки тестується вся програма в цілому, а не як незалежні служби. Крім того, узгоджена розробка досягається завдяки єдиному використанню мови та інфраструктури в програмі.

Однак монолітна архітектура створює проблеми з масштабованістю, оскільки важко масштабувати окремі компоненти незалежно, що часто вимагає масштабування всієї програми. У міру зростання програми виникають труднощі з обслуговуванням, що ускладнює керування великою кодовою базою. Проблеми з розгортанням виникають, оскільки одна зміна вимагає повторного розгортання всієї програми, що призводить до потенційного простою. Обмежена гнучкість є ще одним недоліком, оскільки важко застосувати нові технології для окремих частин програми, не впливаючи на всю систему. Нарешті, більший час збірки може бути результатом більших кодових баз, типових для монолітних архітектур.

Архітектура мікросервісів, що зображена на рисунку 2.2, це підхід, за якого веб-програма розробляється як набір невеликих автономних сервісів, кожен з яких відповідає за певну частину функціональності. Ці сервіси спілкуються через мережу, і їх можна розробляти, розгортати та масштабувати незалежно [16].

Ключові особливості :

- кожен мікросервіс є окремим об'єктом із власною кодовою базою, базою даних і процесом розгортання;
- сервіси можуть бути розгорнуті незалежно, що забезпечує безперервну доставку та безперервне розгортання;
- сервіси слабо пов'язані між собою, тобто зміни в одному сервісі мінімально впливають на зміни в іншому;

- команди розробників можуть створювати, тестувати та розгортати сервіси незалежно, часто використовуючи різні мови програмування та технології;
- служби зазвичай обмінюються даними через протоколи HTTP/REST, gRPC, черги повідомлень або інші мережеві протоколи.

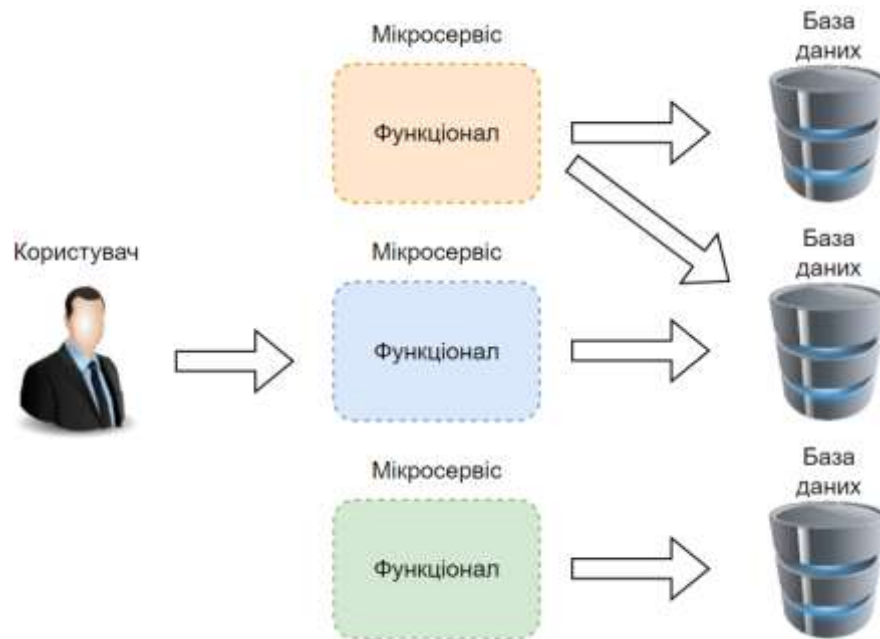


Рисунок 2.2 Мікросервісна архітектура

Архітектура клієнт-сервер, що зображена на рисунку 2.3, це модель, у якій клієнт, як правило, використовує веб-браузер або мобільний додаток, спілкується з сервером через мережу. Ця архітектура є популярною в розробці веб-додатків, де клієнт надсилає запити на сервер щодо даних або послуг, а сервер обробляє ці запити та надсилає необхідні відповіді [17].

Клієнт, це інтерфейс користувача, за допомогою якого користувачі взаємодіють із програмою. Це може бути веб-браузер, мобільний додаток або будь-який інший пристрій, здатний робити запити через мережу.

Сервер відповідає за обробку запитів клієнтів, виконання бізнес-логіки та доступ до даних у разі потреби. Він розміщує внутрішній код програми та відповідає на запити клієнтів необхідними даними або послугами.

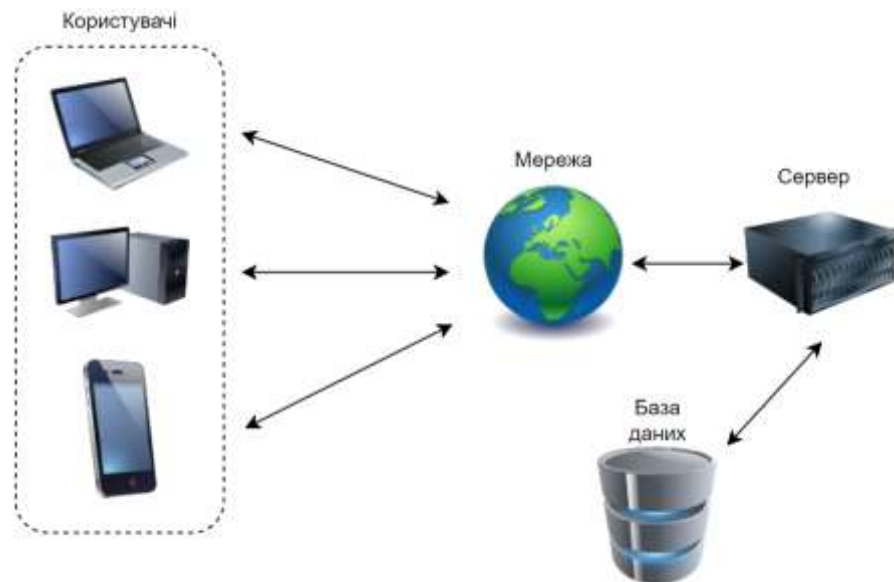


Рисунок 2.3 Клієнт-серверна архітектура

Зв'язок між користувачем і сервером зазвичай відбувається за протоколом HTTP. Коли користувач взаємодіє з клієнтським інтерфейсом, наприклад, натискає кнопку або надсилає форму, клієнт надсилає HTTP-запит на сервер. Сервер обробляє запит, виконує будь-які необхідні дії, такі як запит бази даних або виконання бізнес-логіки, і генерує відповідну відповідь. Потім ця відповідь надсилається назад клієнту, який відображає результати користувачеві.

Для проекту електронної бібліотеки ТЗОШ №27 було обрано клієнт-серверну архітектуру з наступних причин. У ній прекрасно реалізовано розподіл обов'язків. Клієнтська частина відповідає за інтерфейс користувача та взаємодію з користувачем, а серверна частина відповідає за обробку запитів, зберігання та управління даними. Вона є гнучкою у використанні технологій, наприклад на клієнтській частині використовуються HTML, CSS, JavaScript, jQuery та Bootstrap для забезпечення сучасного, адаптивного та інтерактивного інтерфейсу, а на серверній частині використовуються PHP та MySQL для реалізації бізнес-логіки та управління даними. Клієнт-серверна архітектура, забезпечує масштабованість, дозволяє легко масштабувати систему, розширюючи серверні ресурси або додаючи нові клієнтські пристрої. З точки зору безпеки, серверна частина дозволяє централізовано керувати доступом до

даних та забезпечувати їх захист. Також, серверна частина на базі PHP забезпечує ефективну взаємодію з базою даних MySQL, що дозволяє оптимізувати запити та підвищити продуктивність системи.

Обраний архітектурний підхід забезпечить баланс між простотою реалізації, масштабованістю та гнучкістю, що є критично важливим для успішної розробки та експлуатації електронної бібліотеки Тернопільської ЗОШ №27.

## 2.2 Моделювання архітектури вебсайту електронної бібліотеки школи

У кваліфікаційній роботі використовується трьохрівнева клієнт серверна архітектура, тобто файли розподілені по різних рівнях архітектури, зображено на рисунку 2.4.

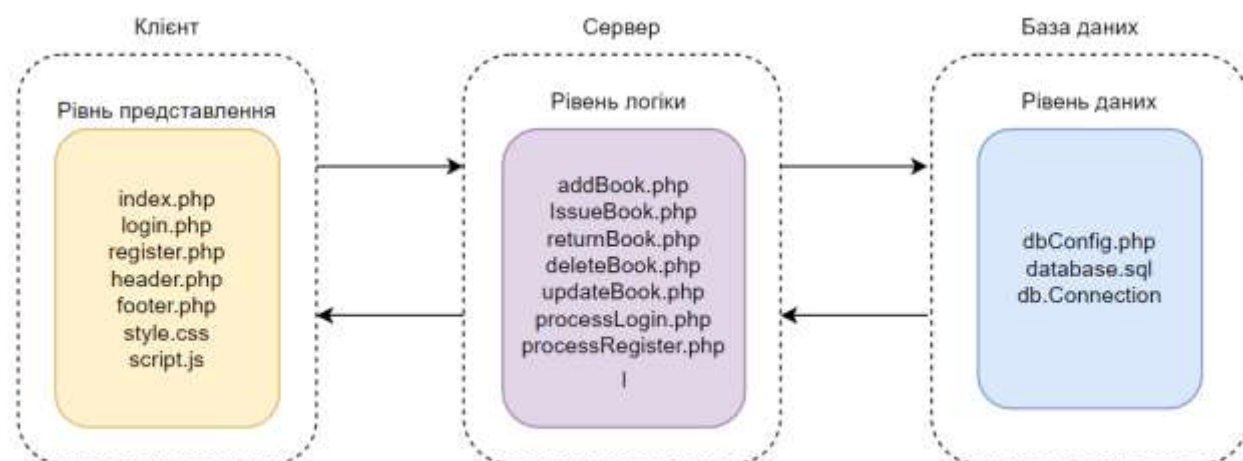


Рисунок 2.4 Трьохрівнева клієнт серверна архітектура

Трьохрівнева клієнт-серверна архітектура (Three-Tier Architecture), це специфічна реалізація клієнт-серверної архітектури, яка розділяє додаток на три окремі логічні рівні або шари: рівень представлення, рівень логіки та рівень даних. Такий підхід покращує модульність системи, полегшує обслуговування та масштабування [18].

Кожен рівень взаємодіє з сусіднім, наприклад, рівень представлення взаємодіє з рівнем логіки, отримуючи дані від користувача або викликаючи певні дії. Рівень логіки обробляє отримані дані згідно з бізнес-правилами і взаємодіє з рівнем даних для виконання необхідних операцій. Рівень даних виконує запити до бази даних та повертає результати на рівень логіки, який обробляє їх і передає на рівень представлення для відображення користувачеві.

Модульність в трьохрівневій архітектурі досягається завдяки чіткому розділенню на окремі рівні, що дозволяє легко оновлювати або замінювати окремі частини системи без впливу на інші. Масштабованість забезпечується можливістю незалежного масштабування кожного рівня, наприклад, додавання серверів для рівня логіки або оптимізації бази даних на рівні даних. Безпека підвищується через можливість впровадження різних рівнів безпеки для різних частин системи, зокрема, захисту доступу до даних на рівні даних. Підтримуваність коду покращується завдяки більш організованому та зрозумілому коду, оскільки кожен рівень виконує свою чітко визначену функцію.

Розглянемо кожен рівень побудованої моделі.

Рівень представлення (див.рис.2.4). Цей рівень відповідає за взаємодію з користувачем і є інтерфейсом між користувачем та системою. Основна функція цього рівня, це відображення даних користувачеві та отримання введених даних від користувача, цей рівень у даній кваліфікаційній роботі, складається з наступних файлів:

- index.php – головна сторінка сайту, код;
- login.php – сторінка входу для користувачів, частина коду логування користувача, наведена в лістингу 2.1;
- register.php – сторінка реєстрації для нових користувачів;
- header.php – шапка сайту, яка включає навігаційне меню;
- footer.php – футер сайту;
- style.css – CSS-файл для стилізації сторінок;

– scripts.js – JavaScript-файл для додаткової функціональності на стороні клієнта.

### Лістинг 2.1 Форма логуювання користувача

```
<div class="login-content">
  <div class="login-body">
    <h4>Форма входу </h4>
    <form action="" method="post">
      <div class="mb-20">
        <input type="text" name="username" class="form-
control" placeholder="Логін" required=""/>
      </div>
      <div class="mb-20">
        <input type="password" name="password" class="form-
control" placeholder="Пароль" required=""/>
      </div>
      <div class="mb-20">
        <input class="btn btn-info submit" type="submit"
name="login" value="Увійти">
        <a class="reset_pass" href="lost-
password.php">Втратили пароль?</a>
      </div>
    </form>
  </div>
  <div class="login-footer text-center">
    <div class="separator">
      <p class="change_link">Вперше на сайті?
      <a href="registration.php" class="text-right"> Створити
акаунт </a>
    </p>
    </div>
  </div>
</div>
```

Рівень логіки (див.рис.2.4), цей рівень відповідає за обробку бізнес-логіки програми. Він приймає дані від рівня представлення, обробляє їх згідно з бізнес-правилами і передає результати на рівень даних або назад на рівень представлення, цей рівень, складається з наступних файлів:

- addBook.php – логіка для додавання нових книг до системи;
- issueBook.php – логіка для видачі книг користувачам;
- returnBook.php – логіка для повернення книг;
- deleteBook.php – логіка для видалення книг з системи, код функції зображено в лістингу 2.2;

- updateBook.php – логіка для оновлення інформації про книги;
- processLogin.php – обробка входу користувачів;
- processRegister.php – обробка реєстрації нових користувачів.

### Лістинг 2.2 Функція видалення книги

```
<?php
session_start();
if (!isset($_SESSION["username"])) {
    ?>
    <script type="text/javascript">
        window.location="login.php";
    </script>
    <?php
}
include 'inc/connection.php';
if (isset($_GET["id"])) {
    $id = $_GET["id"];
    mysqli_query($link, "delete from add_book where id=$id");
    ?>
    <script type="text/javascript">
        window.location="display-books.php";
    </script>
    <?php
}
?>
```

Рівень даних (див.рис.2.4), цей рівень відповідає за зберігання та управління даними. Він взаємодіє з базами даних, файлами або іншими системами зберігання даних, забезпечуючи доступ до даних для рівня логіки і складається з наступних файлів:

- dbConfig.php - файл конфігурації для підключення до бази даних;
- database.sql - SQL-скрипт для створення та ініціалізації бази даних;
- dbConnection.php - файл для встановлення з'єднання з базою даних.

Цей поділ на рівні та відповідні файли допомагає краще організувати код та розділити відповідальність між різними частинами додатку.

## 2.3 Визначення з системою управління базами даних

Електронна бібліотека ЗОШ № 27 має на меті надати учням, учителям та працівникам централізовану платформу для доступу до освітніх ресурсів, управління бібліотечними фондами та полегшення навчання. Вирішальним компонентом цієї системи є база даних, яка зберігатиме та керуватиме величезною колекцією книг, статей та інших цифрових ресурсів бібліотеки. Ця кваліфікаційна робота спрямована на вибір найбільш підходящої системи управління базами даних (надалі СУБД) для електронної бібліотеки.

Щоб визначити ідеальну СУБД, ми повинні спочатку визначити вимоги до електронної бібліотеки. Вибір буде керуватися такими критеріями:

- база даних повинна мати можливість обробляти великий обсяг даних і масштабуватися в міру зростання колекції бібліотеки;
- система управління базами даних повинна забезпечувати конфіденційність, цілісність і доступність конфіденційних даних, таких як інформація про студентів і співробітників;
- база даних повинна підтримувати швидке й ефективне надсилання запитів, щоб полегшити швидкий доступ до ресурсів;
- система управління базами даних повинна підтримувати гнучку модель даних для розміщення різноманітних типів ресурсів у колекції бібліотеки;
- база даних повинна мати можливість інтегруватися з іншими системами, такими як шкільна система управління навчанням і онлайн-каталог;
- база даних має бути зручною для користувачів і легкою в обслуговуванні навіть для нетехнічного персоналу.

Було розглянуто два основні типи баз даних: реляційні та нереляційні. Реляційні бази даних використовують структури у вигляді таблиць для зберігання даних і підтримують мову запитів SQL для доступу до них[19]. Вони добре підходять для систем, де дані мають чітку структуру та взаємозв'язки.



Однією з популярних реляційних баз даних є MySQL, яка відома своєю високою продуктивністю, надійністю, підтримкою ACID-транзакцій для забезпечення цілісності даних та великою спільнотою користувачів і розробників[20].

Іншою потужною реляційною базою даних є PostgreSQL, яка підтримує складні запити та розширення, високу продуктивність і масштабованість[21].

Нереляційні бази даних використовують різні моделі зберігання даних, такі як документи, графи, ключ-значення, що робить їх більш гнучкими для певних типів даних і сценаріїв використання[22].

MongoDB є документно-орієнтованою базою даних з гнучкою структурою даних, що дозволяє зберігати дані у форматі JSON-подібних документів. Вона відома високою масштабованістю та продуктивністю[23].

Іншою нереляційною базою даних є Cassandra, яка оптимізована для масштабованості та доступності, підходить для великих розподілених систем[24].

На основі аналізу вимог та особливостей різних типів баз даних, для електронної бібліотеки ЗОШ №27 було вирішено використовувати реляційну базу даних MySQL. Це рішення було прийняте з огляду на такі фактори:

- дані, що зберігаються в електронній бібліотеці, мають чітку структуру та взаємозв'язки (наприклад, книги, автори, позичання), що добре підходить для реляційної моделі;
- забезпечує високу продуктивність при обробці запитів та має вбудовані механізми для забезпечення цілісності даних;
- підтримує масштабування як вертикальне (розширення ресурсів одного сервера), так і горизонтальне (додавання нових серверів);
- база даних має потужні засоби контролю доступу та забезпечення безпеки даних;
- велика спільнота користувачів і розробників забезпечує доступ до ресурсів, документації та підтримки.

## 2.4 Розробка концептуальної моделі бази даних

Модель даних у базі даних є концептуальним представленням структури та організації даних. Вона визначає, як дані зберігаються, пов'язуються та доступні до бази даних. Модель даних надає план або структуру для проектування та реалізації бази даних і служить інструментом спілкування між зацікавленими сторонами, включаючи дизайнерів баз даних, розробників і користувачів. Зазвичай включає такі компоненти: сутності, зв'язки та атрибути[25].

У базі даних сутність, це реальний об'єкт або концепція, актуальна та важлива в контексті бази даних, призначена для зберігання інформації, зазвичай вони представлені у вигляді таблиць або зв'язків.

Ідентифікація сутностей є важливим кроком у проектуванні бази даних, оскільки це допомагає визначити обсяг і структуру бази даних. Тому кожна сутність повинна бути визначена унікальним ідентифікатором. Також сутності повинні мати набір атрибутів або властивостей, що їх описують. Вони можуть брати участь у відносинах з іншими сутностями й можуть бути зміненими або оновленими незалежно від інших сутностей. Це допомагає переконатися, що база даних розроблена відповідно до потреб організації та її користувачів [26].

Атрибут у базі даних є властивістю або характеристикою сутності. Він являє собою поле даних, яке зберігає певну інформацію про сутність. Атрибути відповідають стовпцям у таблиці бази даних. Атрибути також можуть мати різні властивості, наприклад тип даних, який визначає у якому форматі зберігатимуться дані в атрибуті, як-от ціле число, рядок або дата, довжина описує максимальну кількість символів, які можна зберегти в атрибуті.

Зв'язки визначають взаємозв'язок між різними сутностями (таблицями) і відображають спосіб, якими ці сутності пов'язані між собою. Зв'язки відіграють ключову роль у структуруванні даних і визначенні їх взаємозалежності, що дозволяє зберігати та організовувати інформацію в базі даних.

Типи зв'язків:

- один-до-одного (One-to-One), у даному випадку кожен запис в одній таблиці (сутності) пов'язаний з одним записом в іншій таблиці, наприклад, кожен користувач може мати лише один профіль;
- один-до-багатьох (One-to-Many), кожен запис в одній таблиці пов'язаний з кількома записами в іншій таблиці, наприклад, кожен користувач може мати багато позик книг;
- багато-до-одного (Many-to-One), кілька записів в одній таблиці пов'язані з одним записом в іншій таблиці, наприклад, багато книг належать одному видавцю;
- багато-до-багатьох (Many-to-Many), кілька записів в одній таблиці пов'язані з кількома записами в іншій таблиці, цей тип зв'язку вимагає використання проміжної таблиці, що містить зв'язки між обидвома сутностями, наприклад, багато книг можуть мати багато авторів.

Зв'язки дозволяють забезпечити цілісність даних, контролюючи відповідності між пов'язаними записами. Вони дозволяють ефективно управляти даними, забезпечуючи доступ до потрібної інформації через різні таблиці. З допомогою зв'язків створюються складні запити і звіти, які включають дані з різних сутностей, пов'язаних між собою.

Взаємодія елементів у концептуальній моделі бази даних визначається через зв'язки між сутностями. Ці зв'язки забезпечують логічну цілісність даних та відображають реальні процеси, що відбуваються в бібліотеці. Наприклад, коли учень позичає книгу, створюється запис у сутності «Позичання», який пов'язує конкретну книгу із конкретним учнем. Цей зв'язок дозволяє відстежувати, які книги були позичені і коли вони мають бути повернуті.

Концептуальна модель також допомагає визначити ключові атрибути для кожної сутності, такі як первинні ключі, які однозначно ідентифікують кожен екземпляр сутності, та зовнішні ключі, які забезпечують зв'язок між сутностями. Наприклад, атрибут «ISBN» може бути первинним ключем для сутності «Книга», а атрибут «ID учня» - первинним ключем для сутності «Учень». Зовнішні ключі,

такі як «ID книги» у сутності «Позичання», забезпечують зв'язок між сутностями «Книга» і «Позичання».

Найпопулярнішим методом представлення моделі бази даних є діаграма сутність-зв'язок (Entity-Relationship Diagram або ERD). Діаграма сутність-зв'язок, це графічний інструмент для моделювання та представлення структури бази даних. Вона використовується для візуалізації взаємозв'язків між сутностями (таблицями) у базі даних, що дозволяє розробникам, аналітикам та іншим зацікавленим сторонам легше розуміти, як дані пов'язані між собою.

Розроблена діаграма, зображена на рисунку 2.5.

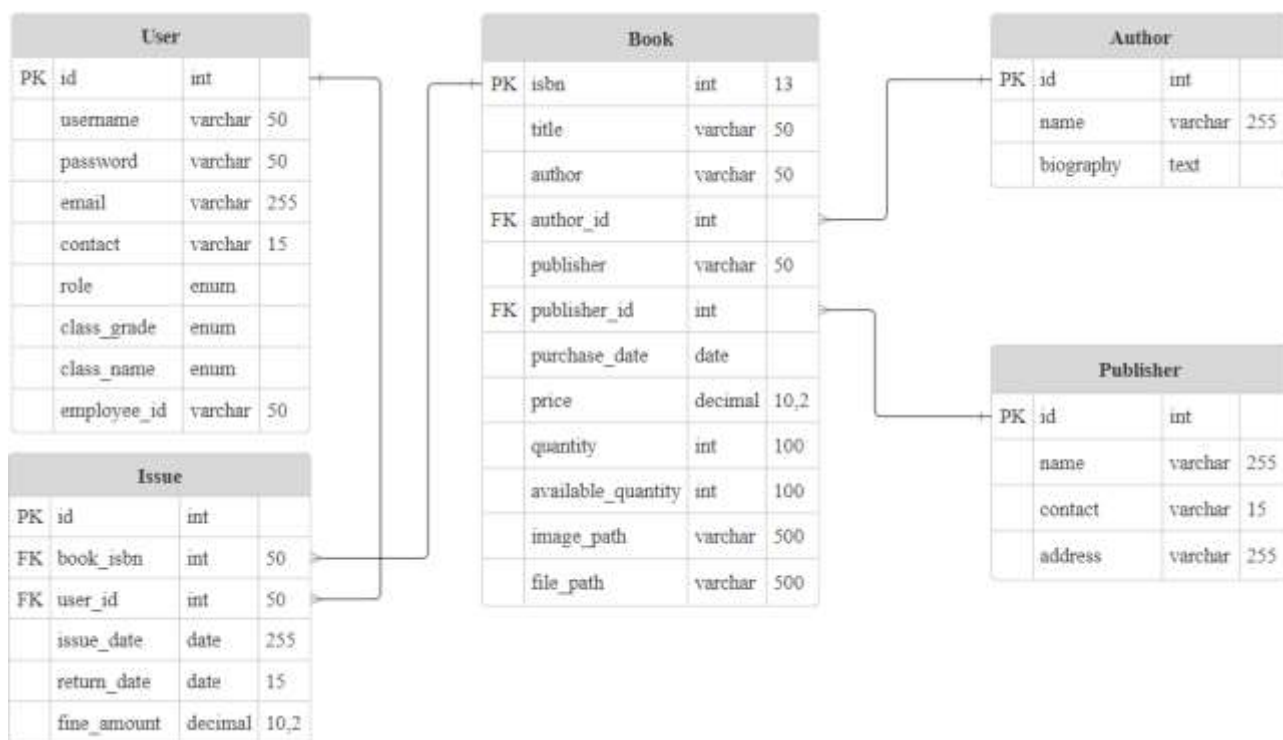


Рисунок 2.5 Діаграма сутність-зв'язок електронної бібліотеки школи

Діаграма сутність-зв'язок є важливим інструментом для розробки баз даних, діаграма має ряд переваг, вона дозволяє легко розуміти, як дані пов'язані між собою, також служить як документація для розробників, щоб зрозуміти структуру і зв'язки в базі даних, дозволяє аналізувати і оптимізувати структуру бази даних до її фізичної реалізації.

## 2.5 Розробка логічної моделі бази даних. Визначення сутностей та їх атрибутів

Проектування сутностей та атрибутів, це необхідний етап розробки бази даних. В електронній бібліотеці основні сутності можуть включати книги, користувачів (бібліотекарів, студентів або вчителів), видачі книг та інші додаткові сутності, що забезпечують управління бібліотекою[27-28].

Так як бібліотека не може існувати без книг, першою сутністю буде «Book», зображена на рисунку 2.6. Книги є основним ресурсом бібліотеки, і їх атрибути повинні відображати всю необхідну інформацію про кожну книгу. Містить наступні атрибути:

- поле «isbn», це унікальний ідентифікатор книги, тип поля – int, поле служить первинним ключем;
- поле «title», містить інформацію про назву книги, тип поля – varchar(255);
- поле «author», зберігає в собі інформацію про автора книги, тип поля – varchar(255);
- поле «author\_id», містить унікальний ідентифікатор автора книги, тип поля – int, поле є зовнішнім ключем до сутності «Author»;
- поле «publisher», зберігає інформацію про назву видавництва книги, тип поля – varchar(255);
- поле «publisher\_id», це унікальний ідентифікатор видавництва книги, тип поля – varchar(255), поле є зовнішнім ключем до сутності «Publisher»;
- поле «purchase\_date», це дата покупки книги, тип поля – date;
- поле «price», містить ціну книги, тип поля – decimal(10, 2);
- поле «quantity», дане поле необхідне, для відстеження кількості книг, які можуть бути видані або вже видані учням, тип поля – int;
- поле «available\_quantity», це поле показує кількість доступних книг, тип поля – int;

- поле «image\_path», визначає шлях до зображення обкладинки книги, тип поля – varchar(500);
- поле «file\_path», містить шлях до електронної версії книги, призначене для того, щоб користувач міг завантажити книгу на свій пристрій, тип поля – varchar(500).

Book			
PK	isbn	int	13
	title	varchar	50
	author	varchar	50
FK	author_id	int	
	publisher	varchar	50
FK	publisher_id	int	
	purchase_date	date	
	price	decimal	10,2
	quantity	int	100
	available_quantity	int	100
	image_path	varchar	500
	file_path	varchar	500

Рисунок 2.6 Сутність «Book»

Наступною спроектуємо сутність «User», зображено на рисунку 2.7. Користувачі є центральними елементами будь-якої бібліотечної системи. В електронній бібліотеці користувачі можуть бути бібліотекарами, вчителями або студентами. Містить наступні атрибути:

- поле «id», це унікальний ідентифікатор користувача, тип даних – int, дане поле є первинним ключем для сутності «User»;
- поле «username», містить ім'я (логін) користувача, тип даних – varchar(50);
- поле «password», зберігає пароль, для доступу до системи, тип даних – varchar(50);

- поле «email», це електронна адреса користувача, на яку в майбутньому можна надсилати листи про нові надходження або заборгованості книг у бібліотеці, тип даних – varchar(255);
- поле «contact», містить контактний номер телефону, для того щоб зв'язатися у разі потреби із користувачем бібліотеки, тип даних – varchar(15);
- поле «role», необхідне, щоб визначити роль користувача, чи це учень, вчитель чи працівник бібліотеки, тип даних – enum('Учень', 'Вчитель', 'Бібліотекар');
- поле «class\_grade», це клас учня, в якому він навчається, тип даних – enum(1–11);
- поле «class\_name», відображає назву класу учня, в якому він навчається, тип даних – enum('А', 'Б', 'В', 'Г');
- поле «employee\_id», містить унікальний ідентифікатор працівника школи, тип даних – varchar(50);
- поле «photo», це фото користувача, поле не є обов'язковим, тип даних – varchar(500);
- поле «verification», призначене для верифікації користувача, чи його заявка на реєстрацію в бібліотеці прийнята чи ні, тип даних – enum('Так', 'Ні').

User			
PK	id	int	
	username	varchar	50
	password	varchar	50
	email	varchar	255
	contact	varchar	15
	role	enum	
	class_grade	enum	
	class_name	enum	
	employee_id	varchar	50
	photo	varchar	500
	verification	enum	

Рисунок 2.7 Сутність «User»

Для того, щоб уникнути ситуації, коли учень хоче зареєструватися як вчитель або бібліотекар і отримати додаткові права користувача бібліотеки, на рівні бази даних було створено тригер для валідації ролей та пов'язаних з ними атрибутів, зображено в лістингу 2.3

### Лістинг 2.3 Код тригера валідації ролей

```
DELIMITER $$
CREATE TRIGGER validate_user_role
BEFORE INSERT ON User
FOR EACH ROW
BEGIN
    IF NEW.role = 'student' THEN
        IF NEW.class_grade IS NULL OR NEW.class_name IS NULL THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Учень повинен мати номер і назву
класу ';
        END IF;
    ELSIF NEW.role = 'teacher' OR NEW.role = 'librarian' THEN
        IF NEW.employee_id IS NULL THEN
            SIGNAL SQLSTATE '45000'
            SET MESSAGE_TEXT = 'Вчителі та бібліотекарі повинні мати
код працівника';
        END IF;
    END IF;
END$$
DELIMITER ;
```

Сутність «Issue» представляє операції з видачі книг користувачам і містить інформацію про кожну видачу, зображене на рисунку 2.8. Ця сутність містить наступні поля :

- поле «id», це унікальний ідентифікатор видачі, тип даних – int, даний атрибут є первинним ключем для сутності «Issue»;
- поле «book\_isbn», дане поле є зовнішнім ключем, що посилається на сутність «Book», тип даних – int;
- поле «user\_id», це зовнішній ключ, що посилається на сутність «User», тип даних – int;
- поле «issue\_date» містить дату видачі книги, користувачу бібліотеки, тип даних – date;



- поле «return\_date» це дата повернення книги, тип даних – date;
- поле «fine\_amount» призначене для нарахування штрафу за прострочення повернення книги, тип даних – decimal(10,2).

Issue			
PK	id	int	
FK	book_isbn	int	50
FK	user_id	int	50
	issue_date	date	255
	return_date	date	15
	fine_amount	decimal	10,2

Рисунок 2.8 Сутність «Issue»

Сутність «Author», автори книг повинні бути окремою сутністю, щоб забезпечити можливість зберігання детальної інформації про кожного з них, зображено на рисунку 2.9. Сутність містить наступні поля:

- поле «id», це унікальний ідентифікатор автора, тип даних – int, дане поле є первинним ключем для сутності «Author»;
- поле «name», ім'я автора, тип даних – varchar(255);
- поле «biography» містить короткі відомості та біографію автора, тип даних – text.

Author			
PK	id	int	
	name	varchar	255
	biography	text	

Рисунок 2.9 Сутність «Author»

І остання сутність для електронної бібліотеки ЗОШ №27, це «Publisher». Видавці книг також мають бути окремою сутністю, що дозволить зберігати інформацію про видавця кожної книги, їх контактні номери та адресу, зображено на рисунку 2.10. Атрибути сутності :

- поле «id», це унікальний ідентифікатор видавця, тип даних – int;
- поле «name», містить назва видавця, тип даних – varchar(255);
- поле «address», зберігає адресу видавця, тип даних – varchar(255);
- поле «contact», вказує номер видавця, щоб зв'язатися, для закупівлі нових підручників, тип даних – varchar(15).

Publisher			
PK	id	int	
	name	varchar	255
	contact	varchar	15
	address	varchar	255

Рисунок 2.10 Сутність «Publisher»

Кожна сутність має добре визначені атрибути, що відображають всі необхідні властивості. Зв'язки між сутностями чітко показують, як дані пов'язані між собою, що дозволяє ефективно управляти інформацією.

## 2.6 Висновки щодо другого розділу

У другому розділі кваліфікаційної роботи було розглянуто процес проектування вебсайту, включаючи моделювання бази даних, визначення сутностей, зв'язків та атрибутів електронної бібліотеки для Тернопільської ЗОШ №27.

Створення вебсайту здійснювалося за допомогою трьохрівневої клієнт-серверної архітектури, що включала рівні представлення, логіки та даних. Це дозволило забезпечити чіткий розподіл функцій між різними компонентами системи, що спрощує її підтримку та масштабування в майбутньому.

На клієнтській стороні були використані HTML, CSS, JavaScript, jQuery та Bootstrap для створення сучасного та адаптивного інтерфейсу, який є зручним для користувачів. Серверна частина була реалізована за допомогою PHP та MySQL, що дозволило забезпечити ефективне управління даними та високу продуктивність системи.

В результаті розробки було створено вебсайт, який включає функціональні можливості для входу, реєстрації, перегляду доступних книг та пошуку по базі даних. Це забезпечує зручний доступ до освітніх матеріалів для учнів, вчителів та працівників школи, сприяючи поліпшенню навчального процесу.

Особливу увагу було приділено моделюванню бази даних. Було визначено основні сутності, такі як «Book», «User», «Author», «Issue» та «Publisher». Для кожної сутності було визначено атрибути, що дозволяють зберігати необхідну інформацію, також було розроблено діаграму сутність-зв'язок, яка є ключовим елементом, що визначає, як дані організовані та як вони взаємодіють між собою.

Другий розділ демонструє, що спроектований вебсайт електронної бібліотеки Тернопільської ЗОШ №27 відповідає всім визначеним вимогам та готовий до впровадження і використання.

## РОЗДІЛ 3. РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ ВЕБСАЙТУ ЕЛЕКТРОННОЇ БІБЛІОТЕКИ

### 3.1 Реалізація інтерфейсу користувача

Інтерфейс користувача (UI або User Interface) і досвід користування (UX або user Experience) – це два поняття, які часто використовуються в дизайні веб-і додатків. UI охоплює всі елементи, які користувач бачить і взаємодіє з ними на екрані (наприклад, кнопки, меню, колірна схема)[29]. UX визначає, як користувачі відчують взаємодію з продуктом, це включає в себе відчуття зручності, задоволення від використання і загальне враження від продукту[30]. Для досягнення успішного дизайну важливо забезпечити якісне поєднання зручності і естетики (UI) з приємним і задовільним досвідом використання (UX).

Стартова сторінка електронної бібліотеки Тернопільської ЗОШ №27, зображена на рисунку 3.1.

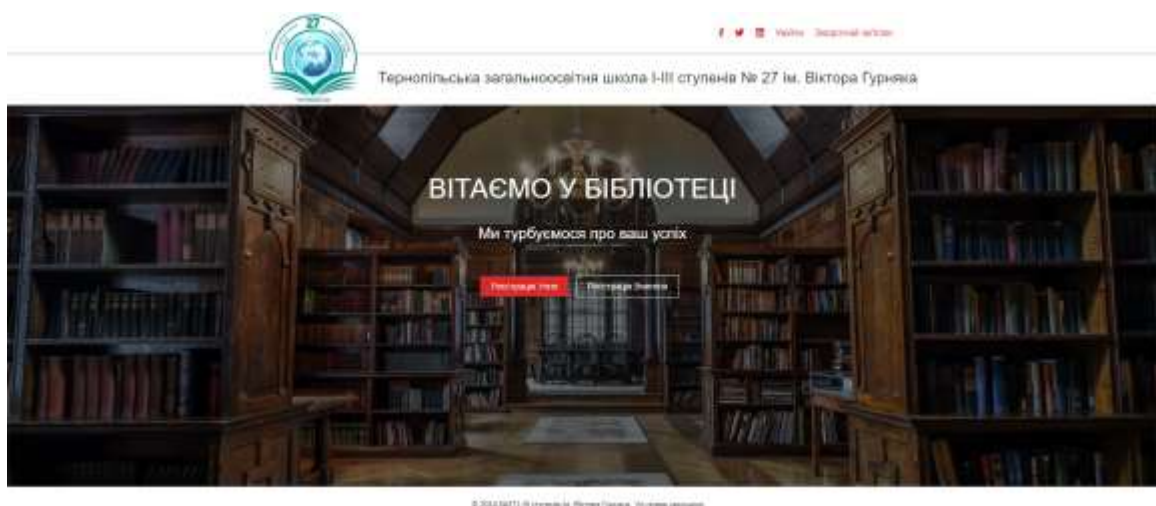


Рисунок 3.1 Стартова сторінка електронної бібліотеки школи

Вгорі сторінки розміщено логотип школи, посилання на соціальні мережі, кнопка “Увійти” для входу в бібліотеку, та кнопка “Зворотній зв'язок”, для того щоб повідомити про певну проблему або задати питання, щодо бібліотеки. В

основній частині вікна розміщено слайдер, з посиланнями на реєстрацію користувачів, частина коду слайдера наведена в лістингу 3.1.

### Лістинг 3.1 Частина коду слайдера

```
<div class="slide-carousel owl-carousel">
  <div class="item" style="background-
image:url(dist/img/3.jpg);">
    <div class="overlay"></div>
    <div class="text">
      <div class="this-item">
        <h2>Вітаємо у бібліотеці</h2>
      </div>
      <div class="this-item">
        <h3>Ми турбуємося про ваш успіх</h3>
      </div>
      <div class="this-item">
        <p><a href="student/registration.php">
Реєстрація учня</a></p>
        <p><a href="teacher/registration.php">
Реєстрація вчителя</a></p>
      </div>
    </div>
  </div>
```

Натиснувши на кнопку “Увійти”, перейдемо на сторінку логування, що зображена на рисунку 3.2. На сторінці розміщено форму входу користувача, також можна відновити пароль при його втраті або створити акаунт, у разі його відсутності.



Рисунок 3.2 Сторінка логування користувача

Увійшовши в систему як учень школи, ми переходимо на сторінку dashboard.php, зображено на рисунку 3.3. Перше що ми бачимо в центральній

частині вікна, це таблицю, в якій відобразатимуться книги, які ми отримали від бібліотеки. У верхньому куті вікна відображається фото профілю та його назва. Ліва частина вікна містить меню навігації, тут ми можемо перейти на сторінку «Профіль», для редагування своїх даних, можемо переглянути книги в бібліотеці, перейшовши на «Доступні книги», також можемо надіслати запит на видачу бажаної книги, перейшовши по посиланню «Запит на книгу».

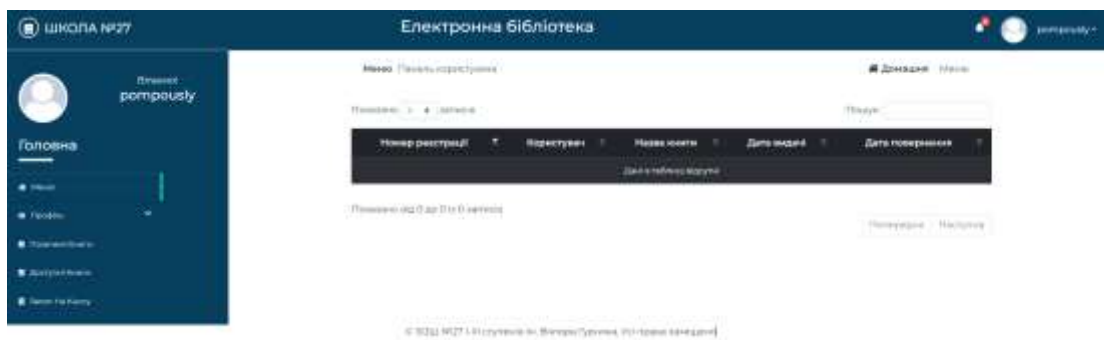


Рисунок 3.3 Панель користувача

Якщо ж увійти як бібліотекар, то головне меню містить статистику бібліотеки, а саме кількість зареєстрованих користувачів, загальну та кількість виданих книг, бачить на яку суму заборговані книги. Також на головному меню для зручності є кнопки “Добавити книгу” та “Добавити користувача”. Перейшовши на посилання “Статус”, можна побачити користувачів, які бажають зареєструватися у бібліотеці і підтвердити або скасувати їхню заявку на реєстрацію. Кнопка “Запит на книги”, призначена щоб розглянути заявки користувачів, на бажану ними книгу. У бічній панелі у бібліотекара теж з’явиться додатковий функціонал, окрім функціоналу користувача бібліотекар може :

- отримати інформацію про учнів або вчителів, які зареєстровані в бібліотеці, кнопка “Інформація про учнів/вчителів”;
- добавляти, видаляти або редагувати книги в електронній бібліотеці перейшовши по кнопці “Книги”;
- власноруч добавити користувача, перейшовши по відповідній кнопці “Добавити користувача”;

- оглянути залишені користувачами заявки на видачу книги, кнопка “Оглянути запити на книги”;
- за потреби може надіслати повідомлення користувачу бібліотеки, кнопка “Надіслати повідомлення”.

Вигляд панелі бібліотекаря, зображено на рисунку 3.4.



Рисунок 3.4 Панель бібліотекаря

Увійдемо, як учень і перейдемо у вкладку “Доступні книги”, зображено на рисунку 3.5.

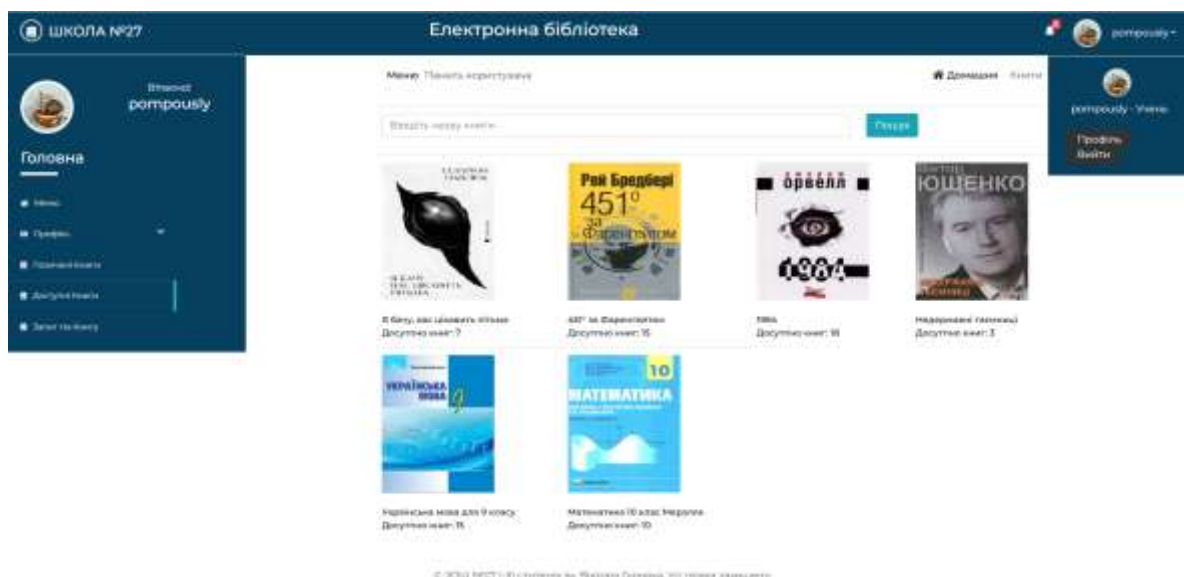


Рисунок 3.5 Вкладка “Доступні книги” для учня

Як бачимо з рисунку (див. рис. 3.5), максимальна кількість книг у ряді – 4, учень бачить обкладинку книги, назву та кількість доступних книг, якщо б учень хотів взяти книгу у бібліотеці, над доступними книгами також реалізовано можливість пошуку книги, оскільки в бібліотеці сотні різноманітних книг. Частина коду відображення книг у таблиці, зображена в лістингу 3.2

### Лістинг 3.2 Відображення книг у таблиці

```

$i=0;
$res = mysqli_query($link,"select * from add_book where
books_availability>0");
echo "<table id='dtBasicExample' class='table control-books'>";
echo "<tr>";
while ($row = mysqli_fetch_array($res)){
    $i=$i+1;
    echo "<td>";
    ?> <a href=" ../.. /?php echo $row["books_file"]; ?>"
target="_blank">
" alt=""></a> <?php
    echo "</br>";
    echo "</br>";
    echo "<b>".$row["books_name"]; "</b>";
    echo "</br>";
    echo "<b>.";
    "Досутпно книг: ".$row["books_availability"]; "</b>";
    echo "</td>";

    if ($i>=4) {
        echo "</tr>";
        echo "<tr>";
        $i=0;
    }
}
echo "</tr>";
echo "</table>";
}

```

Відображення книг виконано в таблицях і книга відображається, якщо є хоча б одна книга в бібліотеці.



### 3.2 Тестування функціоналу

У даному пункті розглянемо основні функції, необхідні для функціонування електронної бібліотеки школи, та їх реалізацію. Функції реалізовані за допомогою PHP для серверної логіки та MySQL для зберігання даних[31].

#### Додавання книг у бібліотеку

Функція додавання книг необхідна для поповнення бібліотечного каталогу новими книгами. Вона дозволяє адміністраторам або бібліотекарям вводити інформацію про нові книги, яка потім зберігається в базі даних, роблячи їх доступними для користувачів, зображена на рисунку 3.6.



Кобзар
Зображення Вибрати файл img227_17.jpg
Файл Вибрати файл 1657774373.pdf
Шеєчення
Просвіта
05.06.2024
1000
Кількість книг
Доступно книг

Рисунок 3.6 Форма додавання книги

Функція реалізована у вигляді форми, де вводяться дані про книгу. Після підтвердження ці дані надсилаються на сервер, де обробляються скриптом PHP і записуються в базу даних MySQL, скрипт php для обробки даних введених у форму, зображений у лістингу 3.3.

### Лістинг 3.3 Скрипт обробки даних внесених у форму

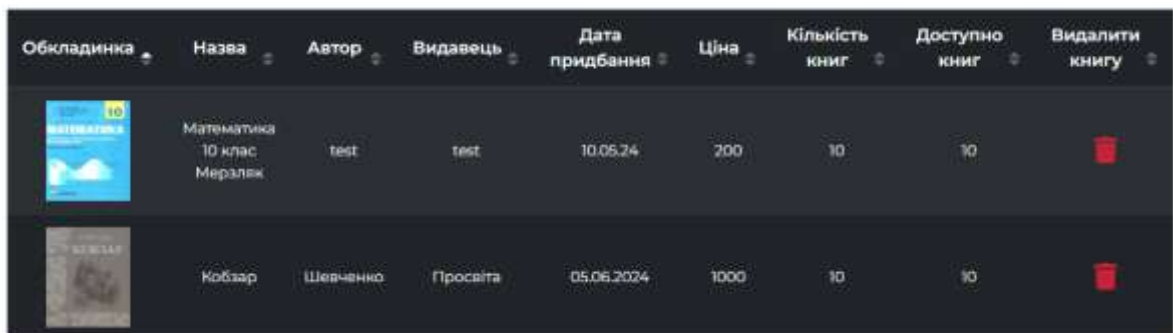
```
<?php
  if (isset($_POST["submit"])) {

    $image_name=$_FILES['f1']['name'];
    $file_name=$_FILES['file']['name'];
    $temp = explode(".", $image_name);
    $temp2 = explode(".", $file_name);
    $newfilename = round(microtime(true)) . '.' . end($temp);
    $newfilename2 = round(microtime(true)) . '.' . end($temp2);
    $imagepath="books-image/".$newfilename;
    $filepath="books-file/".$newfilename2;
    move_uploaded_file($_FILES["f1"]["tmp_name"],$imagepath);
    move_uploaded_file($_FILES["file"]["tmp_name"],$filepath);

    mysqli_query($link, "insert into add_book
values ('','$_POST[booksname]','$imagepath','$_POST[bauthorname]','$
$_POST[bpubname]','$_POST[bpurcdate]','$_POST[bprice]','$_POST[bqu
antity]','$_POST[bavailability]','$_SESSION[username]','$filepath'
)");
  }
}
```

#### Видалення книги

Бібліотекар повинен мати функцію видалення книги, для цього він переходить у вкладку “Книги”, де йому необхідно натиснути іконку смітника, навпроти книги, яку він бажає видалити, зображено на рисунку 3.7







Обкладинка	Назва	Автор	Видавець	Дата придбання	Ціна	Кількість книг	Доступно книг	Видалити книгу
	Математика 10 клас Мерзляк	test	test	10.05.24	200	10	10	
	Кобзар	Шевченко	Просвіта	05.06.2024	1000	10	10	

Рисунок 3.7 Видалення книги

Натиснувши на іконку видалення, спрацьовує php-скрипт, який приймає «id» книги і повністю видаляє її з бази даних, код реалізації даної функції наведений в лістингу 3.4

### Лістинг 3.4 Видалення книги з бібліотеки

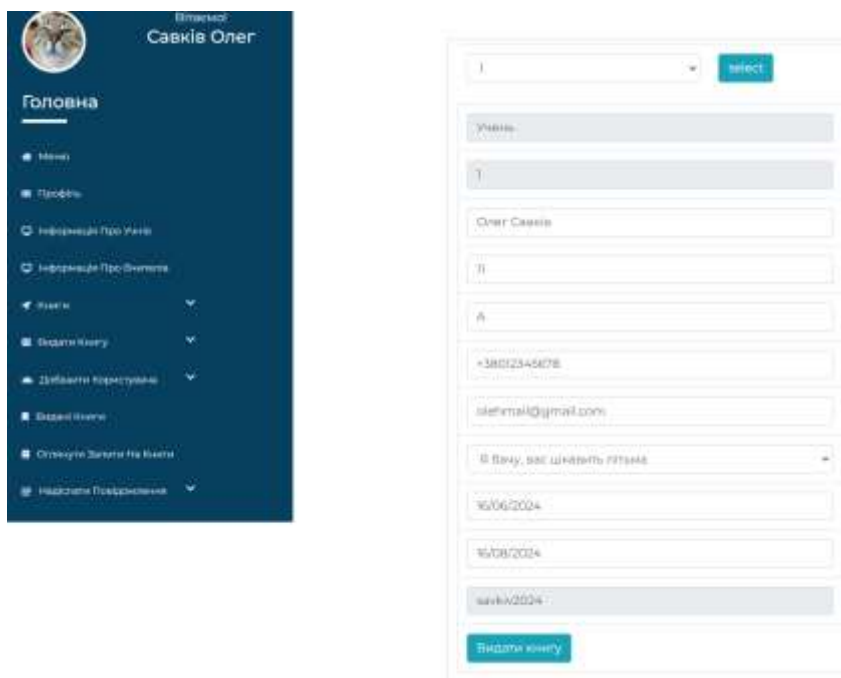
```

<?php
    session_start();
    if (!isset($_SESSION["username"])) {?>
        <script type="text/javascript">
            window.location="login.php";
        </script>
    <?php    }
    include 'inc/connection.php';
    if (isset($_GET["id"])) {
        $id = $_GET["id"];
        mysqli_query($link, "delete from add_book where id=$id");
        ?>
        <script type="text/javascript">
            window.location="display-books.php";
        </script>
    <?php    }?>

```

### Видача книги

Функція видачі книги дозволяє брати книги на певний період. Це допомагає бібліотекарям відстежувати, які книги знаходяться у користувачів, та контролювати терміни їх повернення. Бібліотекарю необхідно перейти на вкладку “Видати книгу”, де необхідно заповнити форму, ввести дані про книгу та користувача, який хоче отримати книгу, форма зображена на рисунку 3.8.



The image shows two parts of a mobile application interface. On the left is a dark blue sidebar menu for a user named 'Олег Савків' (Oleg Savkiv). The menu items include: Головна (Home), Меню (Menu), Профіль (Profile), Інформація Про Мене (Information About Me), Інформація Про Бібліотеку (Information About the Library), Книжки (Books), Видати Книгу (Checkout Book), Діагностика Користувача (User Diagnosis), Вивести Книжки (Withdraw Books), Статуси Заявок На Книжки (Book Request Statuses), and Надіслати Повідомлення (Send Message). On the right is a white form titled 'Видати книгу' (Checkout Book). The form contains several input fields: a dropdown menu with '1' selected and a 'select' button; a 'Учнів' (Students) field; a 'І' (Name) field with 'Олег Савків' entered; a 'ІІ' (Surname) field; an 'А' (Address) field; a phone number field with '+3802346678'; an email field with 'oleg10@gmail.com'; a dropdown menu for 'В яку, вас цікавить бібліотеку' (Which library are you interested in); a date field with '16/06/2024'; another date field with '16/08/2024'; a field with 'savkiv2024'; and a blue 'Видати книгу' (Checkout Book) button at the bottom.

Рисунок 3.8 Форма видачі книги

Після заповнення форми, дані користувача надсилаються на сервер, де обробляються PHP-скриптом і записуються в базу даних MySQL, код обробки даних наведений у лістингу 3.5.

### Лістинг 3.5 Функція обробки даних

```
<?php
    if (isset($_POST["submit2"])) {
        $qty=0;
        $res= mysqli_query($link, "select * from add_book where
books_name='$_POST[booksname]' ");
        while($row = mysqli_fetch_array($res)){
            $qty= $row["books_availability"];
        }
        if ($qty==0) {
            ?>
            <div class="alert alert-danger col-lg-6 col-lg-push-3">
            <strong style="">Ця книга недоступна.</strong>
            </div>
            <?php
        }
        else{
            mysqli_query($link, "insert into issue_book
values('','$_SESSION[utype]','$_SESSION[regno]','$_POST[name]','$_
POST[sem]','$_POST[dept]','$_POST[phone]','$_POST[mail]','$_POST[b
ooksname]','$_POST[booksissuedate]','$_POST[booksreturndate]','$_S
ESSION[susername]') ");
            mysqli_query($link, "update add_book set
books_availability=books_availability-1 where
books_name='$_POST[booksname]'");
            ?>
            <script type="text/javascript">
                alert("Книга успішно видана");
                window.location.href=window.location.href;
            </script>
            <?php
        }
    }
?>
```

### Повернення книги

Дана функція оновлює статус книг в базі даних і дозволяє іншим користувачам позичати ці книги, також, якщо вийшов термін на який була видана книга, то користувачу нараховується штраф, за затримання книги. Реалізація функції наведена в лістингу 3.6

### Лістинг 3.6 Функція повернення книги

```

<?php
    include 'inc/connection.php';
    $id = $_GET["id"];
    $a = date("d/m/Y");
    $fine = "50";
    $res3 = mysqli_query($link, "select * from issue_book where
id=$id");
    while($row3=mysqli_fetch_array($res3)){
        $username = $row3["username"];
        $utype = $row3["utype"];
        $email = $row3["email"];
        $booksname = $row3["booksname"];
        $brdate = $row3["booksreturndate"];}
    if($a > $brdate){
        mysqli_query($link, "insert into finezone
values('','$username','$utype','$email','$booksname','$fine')");}
    $res4 = mysqli_query($link, "select * from t_issuebook where
id=$id");
    while($row4=mysqli_fetch_array($res4)){
        $username = $row4["username"];
        $utype = $row4["utype"];
        $email = $row4["email"];
        $booksname = $row4["booksname"];
        $brdate = $row4["booksreturndate"];}
    $res = mysqli_query($link, "update t_issuebook set
booksreturndate='$a' where id=$id");
    $res2 = mysqli_query($link, "update issue_book set
booksreturndate='$a' where id=$id");
    $books_name="";
    $res = mysqli_query($link, "select * from t_issuebook where
id=$id");
    $res2 = mysqli_query($link, "select * from issue_book where
id=$id");
    while($row=mysqli_fetch_array($res)){
        $books_name = $row["booksname"];}
    while($row=mysqli_fetch_array($res2)){
        $books_name = $row["booksname"];}
    mysqli_query($link, "update add_book set
books_availability=books_availability+1 where
books_name='$books_name'");
?>

```

Дана функція доступна лише, якщо користувач увійшов під іменем бібліотекаря, у іншому випадку функція недоступна.

### 3.3 Впровадження вебсайту

Впровадження вебсайту включає в себе кілька ключових етапів, серед яких важливу роль відіграє вибір і налаштування хостингу. Хостинг є одним із фундаментальних аспектів забезпечення доступності та продуктивності вебсайту[32]. Хостинг – це послуга, що надає обчислювальні ресурси для розміщення вебсайту та його забезпечення безперебійного доступу через Інтернет. Вибір хосту впав на AwardSpace, це хостингова компанія, яка пропонує безкоштовні плани для розміщення вебсайтів.

Після вибору хостингу, необхідно виконати його налаштування, перший крок, це вибір та реєстрація доменного імені у реєстратора доменів, зображено на рисунку 3.9.

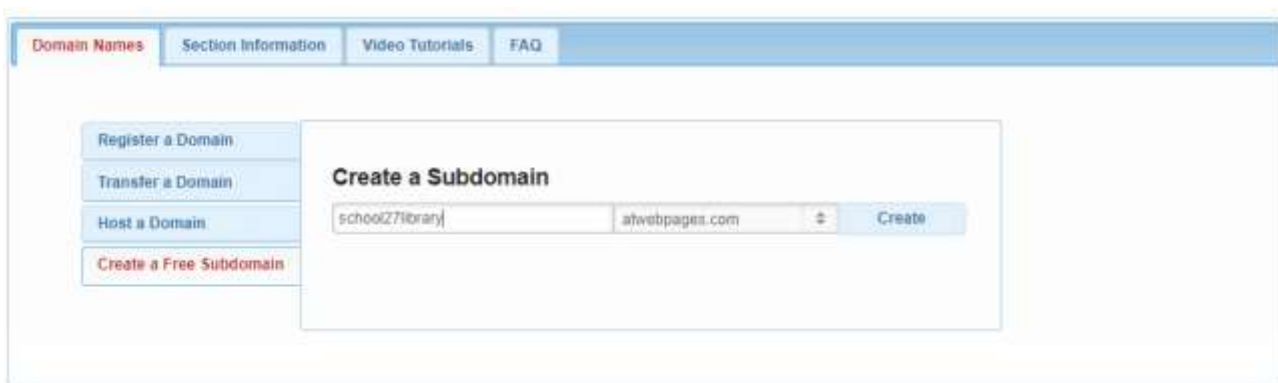


Рисунок 3.9 Реєстрація імені домену

Необхідно завантажити файли, для цього в панелі керування переходимо в розділ "File Manager" і завантажуюємо файли вебсайту. Далі створюємо базу даних, перейшовши в розділ "MySQL Databases", зображено на рисунку 3.10

Рисунок 3.10 Створення бази даних для хостингу

Після створення бази даних на хостингу, необхідно відредагувати файл конфігурації вебсайту “db\_connection.php”, потрібно вказати дані для підключення до бази даних (ім'я бази даних, ім'я користувача, пароль, хост), зображено в лістингу 3.7.

### Лістинг 3.7 Підключення бази даних до хосту

```
<?php
$servername = "fdb1030.awardspace.net";
$username = "4291177_library";
$password = "QCn!hh6QwBvw6eS";
$dbname = "4291177_library";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Далі імпортуємо нашу локальну базу даних на хостинговий сервіс і відкриваємо свій вебсайт у браузері, щоб переконатися, що всі файли завантажені правильно і сайт працює без помилок, зображено на рисунку 3.11.

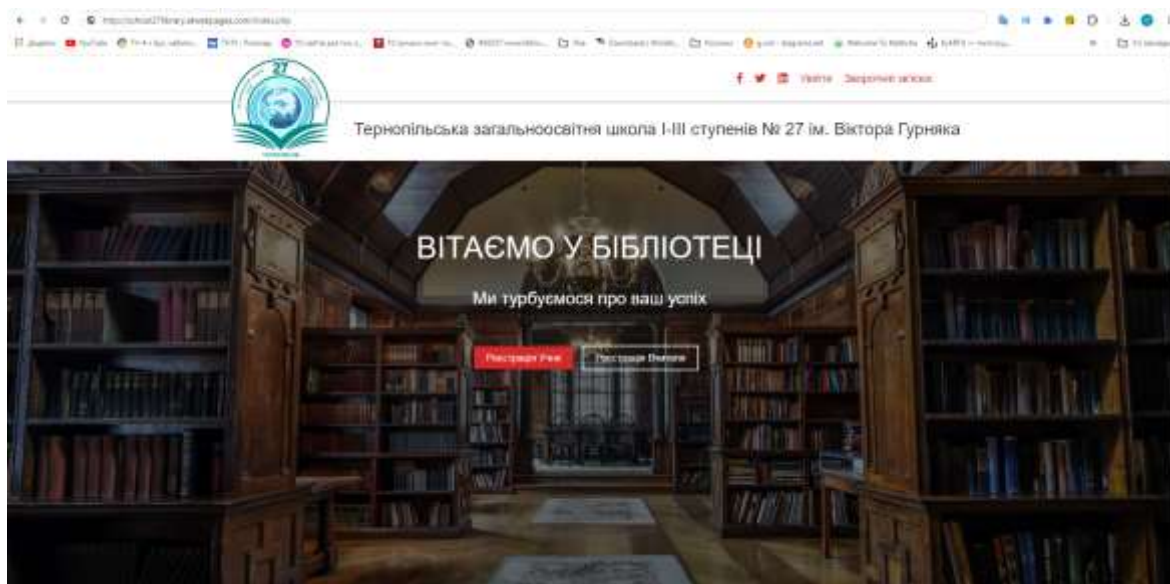


Рисунок 3.11 Електронна бібліотека завантажена на хостинг

Як бачимо з рисунку (див. рис. 3.11), електронна бібліотека завантажена на хостинговий сервіс, усі функції працюють коректно.

### 3.4 Висновки щодо третього розділу

У третьому розділі було розглянуто процес реалізації та впровадження електронної бібліотеки для ЗОШ №27. Основними етапами розробки стали створення інтерфейсу користувача, налаштування бази даних та розміщення вебсайту на хостингу. Була детально описана структура сторінок вебсайту, включаючи стартову сторінку, сторінку логування та панель користувача.

Після налаштування хостингу та імпорту бази даних, вебсайт успішно був запуснений та протестований. В результаті виконаної роботи вдалося створити функціональну електронну бібліотеку, яка дозволяє учням і вчителям зручно користуватися ресурсами шкільної бібліотеки в режимі онлайн.

Тестування підтвердило, що всі основні функції вебсайту працюють належним чином, забезпечуючи користувачам доступ до книг, можливість реєстрації та авторизації, а також управління своїми акаунтами та запитом на книги. Це свідчить про успішне виконання завдань, поставлених на початку проекту, та готовність системи до використання в освітньому процесі.



## РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Вплив діяльності людини на довкілля

В умовах науково-технічного прогресу значно ускладнились взаємовідносини суспільства з природою. Людина отримала можливість впливати на хід природних процесів, підкорила сили природи, почала опановувати майже всі доступні відновні і невідновні природні ресурси, але разом з тим забруднювати і руйнувати довкілля[33].

Втручання людини у природні процеси різко зростає і може спричиняти зміну режиму ґрунтових і підземних вод у цілих регіонах, поверхневого стоку, структури ґрунтів, інтенсифікацію ерозійних процесів, активізацію геохімічних та хімічних процесів у атмосфері, гідросфері та літосфері, зміни мікроклімату тощо. Сучасна діяльність, наприклад, будівництво гідротехнічних споруд, шахт, рудників, доріг, свердловин, водойм, дамб, деформація суші ядерними вибухами, будівництво гігантських міст, обводнення і озеленення пустель, та інші повсякденні аспекти діяльності людини, вже викликали значні видимі і приховані зміни довкілля.

З появою людей на Землі почався вплив їхньої діяльності на кругообіг речовин та енергетичний обмін у біосфері, почалося руйнування біосфери.

Існує декілька класифікацій природних ресурсів. Згідно з природничою класифікацією, ресурси поділяються на природні групи: водні, повітряні, ґрунтові, рослинні, тваринні, мінеральні, кліматичні тощо.

За природно-економічною класифікацією ресурси поділяються на ті, які використовуються в матеріальному виробництві, і ті, що використовуються в невиробничій сфері.

За іншою класифікацією природні ресурси поділяються на невичерпні і вичерпні, а останні на відновлювальні, важко відновлювальні та невідновлювальні.

Відновлюваними вважають біологічні ресурси, атмосферне повітря, поверхневі води. До важко відновлюваних можна віднести ґрунти, підземні води, деякі гірські породи, природні ландшафти. Практично невідновлювальними є переважна більшість корисних копалин та види організмів, що вже зникли на Землі.

Загальна характеристика впливу людини на природні ресурси:

а) наслідки впливу на атмосферу шляхом забруднення:

- парниковий ефект – глобальне потепління клімату за рахунок збільшення вмісту вуглекислого газу в повітрі;
- утворення озонових дір;
- зменшення прозорості атмосфери та збільшення хмарності;
- смог тобто димні тумани, які виникають внаслідок хімічних реакцій в повітрі при його забрудненні великою кількістю пилу та газів;
- кислотні дощі, які утворюються при викиданні в повітря сірчистих сполук і оксидів азоту;
- корозія металевих конструкцій;
- порушення фотосинтезу рослин.

б) наслідки впливу на водні ресурси шляхом їх забруднення:

- зменшення кількості якісної, чистої прісної води;
- порушення життєдіяльності живих організмів водою;
- вимирання окремих видів організмів (жаб, комах, риб);
- порушення ланцюгів живлення у біоценозах (історично складена сукупність тварин і рослин, що населяють територію з більш-менш однаковими умовами існування).

в) наслідки впливу на ґрунт шляхом його забруднення:

- зменшення території, що вкрита рослинністю;
- зменшення площі лісів;
- зниження родючості ґрунтів, пилові бурі, селі;
- погіршення умов росту та розвитку рослинного світу;
- міграція небезпечних речовин в гідросферу;

- накопичення небезпечних речовин в біологічних ланцюгах живлення.

г) вплив діяльності людини на корисні копалини.

Подібно до того, як люди можуть негативно впливати на навколишнє середовище, вони можуть робити це і позитивно. Діяльність, яка має позитивний вплив на навколишнє середовище - це діяльність, що приносить користь навколишньому середовищу або яка хоче виправити наявні негативні наслідки[34]. Як і попередні, вони також можуть бути незворотними, оборотними, тимчасовими або стійкими. Візьмемо кілька основних прикладів.

Лісовідновлення. Йдеться про посадку та відновлення густоти рослинності на території, яка постраждала від вирубки лісів, пожеж чи посухи. Ці наслідки спричиняють втрату продуктивності ґрунту та біорізноманіття. Завдяки відновленню лісів все це можна відновити. Цей прийом є більш корисним, якщо використовуються місцеві види, які були там до їхньої прихильності.

Ефективне зрошення. Як ми бачимо, марнотратство води - це те, що ми маємо на порядку денному. Сільське господарство - це діяльність людини, яка споживає велику кількість води. З цієї причини модифікація зрошувальних систем на інші більш регульовані та ефективніші, такі як краплинна, дозволяє не тільки підвищити продуктивність сільськогосподарських культур, але й заощадити велику кількість води.

Очищення стічних вод. Якщо ми обробляємо це спочатку перед скиданням стічних вод, наслідки будуть меншими. Поширення патогенних мікроорганізмів, змін фізичних та хімічних показників можна уникнути, і, отже, воно забруднює води або екосистеми, де воно скидається менше.

Переробка. Оскільки ми генеруємо велику кількість відходів, переробляючи їх, щоб повернути їх назад у життєвий цикл продукції, є гарною ідеєю, щоб уникнути забруднення чи витрачання такої кількості сировини при її виробництві.

Використання відновлюваної енергії. Звичайно, відновлювані джерела енергії - це майбутнє. Вони чисті і не забруднюють як у своєму виробництві, так і в процесі використання.

Існують як позитивні, так і негативні впливи на навколишнє середовище. Нам просто потрібно збільшити позитивні сторони, щоб покращити нашу планету.

#### **4.2 Особливості безпеки праці під час вантажно-розвантажувальних робіт**

Безпека під час виконання вантажно-розвантажувальних робіт залежить від групи, класу та категорії вантажу. В залежності від небезпеки, яка виникає під час навантажування, транспортування та розвантажування, всі вантажі поділяються на чотири групи: 1 - малонебезпечні (будматеріали, продукти харчування тощо); 2 - небезпечні за своїми розмірами; 3 - пилові та гарячі (цемент, крейда, вапно, асфальт, бітум і т. ін.); 4 - небезпечні за своїми властивостями (пожежо- та вибухонебезпечні, отруйні, токсичні, радіоактивні речовини тощо). Під час виконання вантажно-розвантажувальних робіт з вантажами третьої та четвертої груп необхідно використовувати засоби індивідуального захисту[35].

Вантажі, які є небезпечними за своїми властивостями відповідно до ДСТУ 4500-3-2008 поділяються на дев'ять класів: 1 - вибухові речовини; 2 - стиснені, зріджені та розчинені гази під тиском; 3 - легкозаймисті рідини, суміші рідин, які виділяють легкозаймисті пари, температура спалаху яких становить 61 °C і нижче; 4 - легкозаймисті речовини та матеріали, які здатні займатися внаслідок тертя, нагрівання, поглинання вологи, самочинних хімічних перетворень; 5 - окиснювальні речовини, які легко виділяють кисень; 6 - отруйні та інфекційні речовини; 7 - радіоактивні речовини; 8 - їдкі та корозійно активні речовини; 9 - речовини з відносно низькою небезпекою, під час перевезення та зберігання яких необхідно дотримуватись певних вимог безпеки. На упаковці з небезпечними

вантажами, крім стандартного маркування, необхідно нанести знак небезпеки. Цей знак має форму квадрата, окантованого чорною рамкою, що повернений на кут і поділений на два однакових трикутники. У верхньому трикутнику наносять символ небезпеки, а у нижньому роблять напис про небезпечність вантажу та номер класу

За масою одного місця вантажі поділяються на три категорії: 1 - масою менше ніж 80 кг, а також сипкі, дрібноштучні й такі, що перевозять навалом; 2 - масою від 80 до 500 кг; 3 - масою понад 500 кг[36].

Відповідними нормативно-правовими актами регламентовані граничні норми підймання та переміщення важких речей (вантажів) одним працівником вручну: для чоловіків, старших 18 років, - 50 кг (допускається перенесення вантажу масою до 80 кг на відстань по горизонталі не більшу ніж 25 м за умови, що вантаж піднімають на спину і знімають інші вантажники); для жінок, старших 18 років, - 10 кг при чергуванні з іншою роботою та 7 кг при постійній роботі з вантажами протягом зміни; для юнаків та дівчат 16 - 17 років при короткочасній роботі відповідно 14 та 7 кг, а 17-18 років - 16 та 8 кг[37].

Якщо маса вантажів понад 50 кг, а також у разі підймання вантажів на висоту понад 3 м, обов'язково необхідно застосовувати механізований спосіб проведення вантажно-розвантажувальних робіт, використовуючи при цьому механічні пристосування та підйимально-транспортні механізми. Виконанню таких робіт передуює складання карт технологічних процесів на вантажно-розвантажувальні роботи, визначення маршрутів руху транспортних засобів і т. ін.

Майданчики для проведення вантажно-розвантажувальних робіт повинні мати рівне та тверде покриття з ухилом не більше ніж 5°, а також природне та штучне освітлення. У місцях виконання вантажно-розвантажувальних робіт необхідно встановити знаки безпеки, відповідно до ДСТУ 7010-2018.

Вантажно-розвантажувальні роботи необхідно здійснювати під керівництвом відповідальної особи, призначеної в установленому порядку. Така особа перевіряє (до початку роботи і в процесі її виконання) справність

підйимально-транспортних машин та механізмів, такелажного та іншого інвентарю, інструктує працівників, пояснює послідовність виконання операцій, стежить, щоб у зоні проведення робіт не було сторонніх осіб тощо. У разі виникнення небезпечних ситуацій особа, що відповідає за здійснення вантажно-розвантажувальних робіт повинна негайно вжити запобіжних заходів, а якщо необхідно - припинити роботи до усунення небезпеки.

До роботи з підйимально-транспортними механізмами та пристроями допускаються особи, не молодші 18 років, які пройшли медичний огляд і спеціальне навчання, склали іспит кваліфікаційній комісії та одержали посвідчення.

Підвищені вимоги безпеки регламентуються для вантажно-розвантажувальних робіт та транспортування небезпечних вантажів. Такі роботи належить виконувати у спеціально відведених місцях з дотриманням відповідних вимог безпеки. У разі пошкодження тари небезпечного вантажу, відсутності маркування та попереджувальних написів на ній, а також при метеорологічних умовах, що впливають на фізико-хімічні властивості вантажу (наприклад, підвищують його токсичність), забороняється проводити вантажно-розвантажувальні роботи. Небезпечні вантажі не допускається перевозити на транспортних засобах, які для цього не пристосовані. Легкозаймісті рідини належить транспортувати спеціалізованими транспортними засобами, які мають відповідні написи та заземлення у вигляді металевого ланцюга із загостренням на кінці. Балони, наповнені стисненим, зрідженим або розчиненим газом під тиском, необхідно перевозити на підресорному транспортному засобі поперек кузова у закріпленому стані, що не допускає їх співударяння. Перевозити балони у вертикальному положенні допускається лише у спеціальних контейнерах. Легкозаймісті рідини та балони з газом необхідно перевозити транспортними засобами, які обладнані іскрогасниками на вихлопних трубах.

Для забезпечення безпеки важливе значення має також дотримання встановлених правил складування вантажів. Так, кошики з бутлями агресивних речовин розміщують у складах лише в один ряд, барабани з карбідом кальцію -

не більше двох ярусів. Якщо немає відповідних застережень, то вантажі у стандартній тарі зазвичай складають у штабелі. Відношення висоти штабеля до довжини найменшої сторони тари, що штабелюється, не повинно бути більше ніж 6 - для нерозбірної тари, та 4 - для розбірної тари. Ширина штабеля не повинна бути меншою, ніж його висота. Навантаження на нижню тару не повинно перевищувати допустимих значень. Відстань між рядами штабелів визначається із урахуванням можливості встановлення тари у штабель та її зняття за допомогою вантажозахоплювальних пристроїв застосовуваних засобів механізації та забезпечення необхідних протипожежних розривів. Між рядами штабелів повинні бути проходи шириною не менше ніж 1,25 м, а ширина основного проходу - не менше 3,0 м[38].

### **4.3 Висновок щодо 4 розділу**

У четвертому розділі було проаналізовано аспекти безпеки життєдіяльності та основ охорони праці. Особлива увага приділена впливу діяльності людини на довкілля та забезпеченню безпеки праці під час вантажно-розвантажувальних робіт. Визначено необхідність дотримання встановлених правил складування вантажів та забезпечення умов безпечної роботи з небезпечними вантажами.

## ВИСНОВКИ

Розробка електронної бібліотеки для Тернопільської ЗОШ № 27 стала важливим кроком у забезпеченні сучасного та ефективного доступу до навчальних матеріалів і ресурсів для учнів та викладачів школи.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- подано аналіз предметної області та існуючих рішень для розробки електронної бібліотеки, що дозволило визначити основні вимоги до системи;
- розглянуто сучасні підходи до проектування веб-додатків, що забезпечують зручність та ефективність використання;
- висвітлено проблеми та недоліки існуючих систем, такі як незручний інтерфейс, недостатня динамічність та затримки у відповіді;
- проаналізовано переваги та недоліки різних технологій та інструментів, які можуть бути використані для розробки електронної бібліотеки.

В другому розділі кваліфікаційної роботи:

- досліджено архітектуру та моделі даних для електронної бібліотеки, визначено оптимальні рішення для її реалізації;
- обґрунтовано вибір технологій для створення серверної та клієнтської частин системи, зокрема PHP, MySQL, HTML, CSS та JavaScript;
- розроблена логічна модель бази даних, було описано основні сутності, їх атрибути та взаємозв'язок між ними.

В третьому розділі кваліфікаційної роботи:

- розроблено прототип електронної бібліотеки, включаючи всі необхідні функціональні модулі;
- запропоновано методи оптимізації роботи системи для забезпечення швидкого доступу до інформації та мінімізації затримок у відповідях;
- спроектовано структуру бази даних та логіку роботи серверної частини, що забезпечує стабільність та надійність функціонування бібліотеки;



- протестовано розроблену систему в умовах реальної експлуатації, виявлено та усунуто недоліки, проведено оцінку ефективності та зручності використання.

У розділі «Безпека життєдіяльності, основи охорони праці»:

- висвітлено аспекти безпеки життєдіяльності та охорони праці, зокрема вплив діяльності людини на довкілля та забезпечення безпечних умов праці при виконанні робіт;

- проаналізовано правила складування вантажів та забезпечення умов безпечної роботи з небезпечними вантажами.

Загалом, розробка електронної бібліотеки для Тернопільської ЗОШ № 27 виконана успішно, досягнуті результати відповідають поставленим цілям, а розроблена система забезпечує надійний та ефективний доступ до навчальних ресурсів, сприяючи покращенню освітнього процесу в школі.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Ловендер, Бен. HTML and CSS: Create and Deploy a Cutting-Edge Website. Видавництво Packt Publishing, 2022.
2. Роббінс, Дженніфер Нідерст. Вивчення веб-дизайну: Посібник для початківців з HTML, CSS, JavaScript і веб-графікою. Видавництво O'Reilly Media, 2019.
3. Мейер, Ерік А., та Естель Вейл. CSS: Повний посібник. Видавництво O'Reilly Media, 2020.
4. Веру, Лія. CSS Secrets: Кращі рішення для щоденного веб-дизайну. Видавництво O'Reilly Media, 2020.
5. Летт, Якоб. Bootstrap 5 Quick Start: Розробка адаптивних веб-сайтів з HTML, CSS і JavaScript. Видавництво Packt Publishing, 2021.
6. Хавербеке, Марейн. Елегантний JavaScript: Сучасне введення до програмування. Видавництво No Starch Press, 2018.
7. Крокфорд, Дуглас. JavaScript: Хороші частини. Видавництво O'Reilly Media, 2023.
8. Бібло, Бер, Єгуда Кац і Орельо Де Роза. jQuery в дії. Видавництво Manning Publications, 2020.
9. Чаффер, Джонатан, і Карл Сведберг. Вивчення jQuery: Кращій дизайн взаємодії і розробка веб-техніками JavaScript. Видавництво Packt Publishing, 2022.
10. Ульман, Ларрі. PHP і MySQL для динамічних веб-сайтів: Візуальний посібник для швидкої професійної підготовки. Видавництво Peachpit Press, 2018.
11. Веллінг, Люк, і Лора Томсон. Розробка веб-сайтів на PHP і MySQL. Видавництво Addison-Wesley Professional, 2018.
12. Бьюлію, Алан. Вивчення SQL. Видавництво O'Reilly Media, 2019.
13. Васвані, Вікрам. Effective MySQL: SQL Optimization & Performance Tuning. Видавництво McGraw-Hill Education, 2019.

14. Документація Visual Studio Code. Visual Studio Code, <https://code.visualstudio.com/docs>(дата звернення 09.06.2024).
15. "Монолітна архітектура: переваги та недоліки." Medium, Datamify, <https://datamify.medium.com/monolithic-architecture-advantages-and-disadvantages-e71a603eec89>(дата звернення 09.06.2024).
16. Ньюмен, Сем. Побудова мікросервісів: Проектування дрібнозернових систем. Видавництво O'Reilly Media, 2023.
17. Ретіг, Джон Пол. Programming WebAssembly with Rust: Unified Development for Web, Mobile, and Embedded Applications. Видавництво Pragmatic Bookshelf, 2020.
18. Річардсон, Кріс, і Флінн, Флойд. Microservices Patterns: With examples in Java. Видавництво Manning Publications, 2019.
19. Дейт, К.Дж. Реляційна теорія для практикуючих баз даних. Видавництво O'Reilly Media, 2020.
20. Документація MySQL. MySQL, <https://dev.mysql.com/doc/>. (дата звернення 09.06.2024).
21. Обе, Регіна О., та Лео С. Хсу. PostgreSQL: Запуск та ефективна робота: Практичний посібник до розширеного відкритого джерела баз даних. Видавництво O'Reilly Media, 2019.
22. Фаулер, Мартін, і Прамодкумар Дж. Садаладж. NoSQL: Методологія розробки нереляційних баз даних. Видавництво Addison-Wesley Professional, 2020.
23. Бредшоу, Шеннон, Євген Бразіл і Крістіна Чодоров. MongoDB: Остаточний посібник: Могутня та масштабована система зберігання даних. Видавництво O'Reilly Media, 2019.
24. Карпентер, Джефф, і Ебен Хьюїтт. Cassandra: Остаточний посібник. Видавництво O'Reilly Media, 2020.
25. Коннолі, Томас, і Керолін Бегг. Системи баз даних: Практичний підхід до проектування, реалізації і управління. Видавництво Pearson Education, 2019.

26. Гернандес, Майкл Дж. Проектування баз даних для звичайних смертних: Практичний посібник з реляційного проектування баз даних. Видавництво Addison-Wesley Professional, 2023.
27. Елмасрі, Рамез, і Шамкант Б. Наваче. Основи систем баз даних. Видавництво Pearson, 2019.
28. Черчер, Клер. Початок проектування баз даних: Від новачка до професіонала. Видавництво Apress, 2022.
29. Круг, Стів. Не заставляйте мене думати: Масштабування веб-дизайну для максимальної ефективності. Видавництво New Riders, 2021.
30. Готельф, Джефф, і Джош Сейден. Lean UX: Визначення продукту за допомогою гнучких команд. Видавництво O'Reilly Media, 2022.
31. Моліно, Іан. Мистецтво тестування продуктивності програм: Підтримка програмістів і забезпечення якості. Видавництво O'Reilly Media, 2019.
32. Фелк-Морріс, Тері. Основи веб-розробки та дизайну з HTML5. Видавництво Pearson Education, 2020.
33. Кляйн, Наомі. Це все змінює: Капіталізм проти клімату. Видавництво Simon & Schuster, 2021.
34. Кольберт, Елізабет. Шосте вимирання: Неприродна історія. Видавництво Henry Holt and Company, 2018.
35. Health and Safety Executive (HSE). Здоров'я і безпека на будівництві (HSG150).
36. "Вимоги безпеки під час перевезення вантажів транспортними засобами." ОППБ, <https://oppb.com.ua/news/vymogy-bezpeky-pid-chas-perevezennya-vantazhiv-transportnymu-zasobamy> (дата звернення 15.06.2024).
37. "Про затвердження Правил охорони праці під час вантажно-розвантажувальних робіт і розміщення вантажів." Верховна Рада України, <https://zakon.rada.gov.ua/laws/show/z0183-96#Text> (дата звернення 15.06.2024).
38. "Положення про розробку інструкцій з охорони праці." Верховна Рада України, <https://zakon.rada.gov.ua/rada/show/n0007509-05#Text>(дата звернення 15.06.2024).

# ДОДАТКИ

## Сторінка входу в електронну бібліотеку, файл «login.php»

```

<?php
    session_start();
    include 'inc/connection.php';
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Бібліотека Тернопільської ЗОШ №27</title>
    <link rel="stylesheet" href="inc/css/bootstrap.min.css">
    <link rel="stylesheet" href="inc/css/fontawesome-all.min.css">
    <link rel="stylesheet" href="inc/css/pro1.css">
    <link
href="https://fonts.googleapis.com/css?family=Montserrat:400,500,6
00" rel="stylesheet">
    <style>
        .login{
            background-image: url(inc/img/3.jpg);
            margin-bottom: 30px;
            padding: 50px;
            padding-bottom: 70px;
        }
        .reg-header h2{
            color: #DDDDDD;
            z-index: 999999;
        }
        .login-body h4{
            margin-bottom: 20px;
        }
    </style>
</head>
<body>

    <div class="login registration">
        <div class="wrapper">
            <div class="reg-header text-center">
                <h2><a href="../index.php"
style="color:white">Бібліотека ЗОШ №27 I-III ступенів ім. Віктора
Гурняка</a></h2>
                <div class="gap-40"></div>
            </div>
            <div class="gap-30"></div>
            <div class="login-content">
                <div class="login-body">
                    <h4>Форма входу</h4>
                    <form action="" method="post">
                        <div class="mb-20">

```

```

        <input type="text" name="username"
class="form-control" placeholder="Логін" required=""/>
    </div>
    <div class="mb-20">
        <input type="password" name="password"
class="form-control" placeholder="Пароль" required=""/>
    </div>
    <div class="mb-20">
        <input class="btn btn-info submit"
type="submit" name="login" value="Увійти">
        <a class="reset_pass" href="lost-
password.php">Втратили пароль?</a>
    </div>
</form>
</div>
<div class="login-footer text-center">
    <div class="separator">
        <p class="change_link">Вперше на сайті?
        <a href="registration.php"
class="text-right"> Створити акаунт </a>
    </p>
    </div>
</div>
<?php
    if (isset($_POST["login"])) {
        $count=0;
        $res= mysqli_query($link, "select * from
std_registration where username='$_POST[username]' && password=
'$_POST[password]' && status='yes' && verified='yes' ");
        $count = mysqli_num_rows($res);
        if ($count==0) {
            ?>
                <div class="alert alert-warning">
                    <strong
style="color:#333">Невірний!</strong> <span style="color:
red;font-weight: bold; ">Логін або Пароль.</span>
                </div>
            <?php
        }
        else{
            $_SESSION["student"] =
$_POST["username"];
            ?>
                <script type="text/javascript">
                    window.location="my-issued-
books.php";
                </script>
            <?php
        }
    }
?>
</div>
</div>

```

```
</div>

<div class="footer text-center">
  <p>&copy; ЗОШ №27 I-III ступенів ім. Віктора Гурняка. Усі
права захищено </p>
</div>

<script src="inc/js/jquery-2.2.4.min.js"></script>
<script src="inc/js/bootstrap.min.js"></script>
<script src="inc/js/custom.js"></script>
</body>
</html>
```



## Головна сторінка електронної бібліотеки, файл «index.php»

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Бібліотека Тернопільської ЗОШ №27</title>
  <link rel="icon" type="image/png" href="dist/img/favicon.ico">
  <link rel="stylesheet" href="dist/css/bootstrap.min.css">
  <link rel="stylesheet" href="dist/css/fontawesome-
all.min.css">
  <link rel="stylesheet" href="dist/css/owl.carousel.min.css">
  <link rel="stylesheet"
href="dist/css/owl.theme.default.min.css">
  <link rel="stylesheet" href="dist/css/animate.css">
  <link rel="stylesheet" href="dist/css/main.css">
</head>
<body>
  <div class="header">
    <div class="container">
      <div class="row">
        <div class="col-3">
          <div class="logo">
            
          </div>
        </div>
        <div class="col-9">
          <div class="header-right">
            <ul>
              <li><a href=""><i class="fab fa-
facebook-f"></i></a></li>
              <li><a href=""><i class="fab fa-
twitter"></i></a></li>
              <li><a href=""><i class="fab fa-
linkedin"></i></a></li>
              <li class="dropdown"><a href="#"
class="dropdown-toggle" data-toggle="dropdown">Увійти</a>
                <ul class="dropdown-menu">
                  <li><a
href="student/login.php">Вхід учня</a></li>
                  <li><a
href="teacher/login.php">Вхід вчителя</a></li>
                </ul>
              </li>
              <li><a href="contactus.php">Зворотній
зв'язок</a></li>
            </ul>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

        </div>
    </div>
    <div class="school-name">
        <h1>Тернопільська загальноосвітня школа I-III ступенів №
27 ім. Віктора Гурняка</h1>
    </div>
    <!--Start slider-->
    <div class="slider">
        <div class="slide-carousel owl-carousel">
            <div class="item" style="background-
image:url(dist/img/3.jpg);">
                <div class="overlay"></div>
                <div class="text">
                    <div class="this-item">
                        <h2>Вітаємо у бібліотеці</h2>
                    </div>
                    <div class="this-item">
                        <h3>Ми турбуємося про ваш успіх</h3>
                    </div>
                    <div class="this-item">
                        <p><a
href="student/registration.php">Реєстрація учня</a></p>
                        <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
                    </div>
                </div>
            </div>
            <div class="item" style="background-
image:url(dist/img/2.jpg);">
                <div class="overlay"></div>
                <div class="text">
                    <div class="this-item">
                        <h2>Вітаємо у бібліотеці</h2>
                    </div>
                    <div class="this-item">
                        <h3>Ми турбуємося про ваш успіх</h3>
                    </div>
                    <div class="this-item">
                        <p><a
href="student/registration.php">Реєстрація учня</a></p>
                        <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
                    </div>
                </div>
            </div>
            <div class="item" style="background-
image:url(dist/img/1.jpg);">
                <div class="overlay"></div>
                <div class="text">
                    <div class="this-item">
                        <h2>Вітаємо у бібліотеці</h2>
                    </div>
                    <div class="this-item">

```

```

        <h3>Ми турбуємося про ваш успіх</h3>
    </div>
    <div class="this-item">
        <p><a
href="student/registration.php">Реєстрація учня</a></p>
        <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
    </div>
</div>
<div class="item" style="background-
image:url(dist/img/4.jpg);">
    <div class="overlay"></div>
    <div class="text">
        <div class="this-item">
            <h2>Вітаємо у бібліотеці</h2>
        </div>
        <div class="this-item">
            <h3>Ми турбуємося про ваш успіх</h3>
        </div>
        <div class="this-item">
            <p><a
href="student/registration.php">Реєстрація учня</a></p>
            <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
        </div>
    </div>
<div class="item" style="background-
image:url(dist/img/5.jpg);">
    <div class="overlay"></div>
    <div class="text">
        <div class="this-item">
            <h2>Вітаємо у бібліотеці</h2>
        </div>
        <div class="this-item">
            <h3>Ми турбуємося про ваш успіх</h3>
        </div>
        <div class="this-item">
            <p><a
href="student/registration.php">Реєстрація учня</a></p>
            <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
        </div>
    </div>
<div class="item" style="background-
image:url(dist/img/6.jpg);">
    <div class="overlay"></div>
    <div class="text">
        <div class="this-item">
            <h2>Вітаємо у бібліотеці</h2>
        </div>

```

```
        <div class="this-item">
            <h3>Ми турбуємося про ваш успіх</h3>
        </div>
        <div class="this-item">
            <p><a
href="student/registration.php">Реєстрація учня</a></p>
            <p><a
href="teacher/registration.php">Реєстрація вчителя</a></p>
        </div>
    </div>
</div>
</div>
</div>
</div>

<div class="footer text-center">
    <p>&copy; ЗОШ №27 I-III ступенів ім. Віктора Гурняка. Усі
права захищено</p>
</div>

<script src="dist/js/jquery-2.2.4.min.js"></script>
<script src="dist/js/bootstrap.min.js"></script>
<script src="dist/js/fontawesome.min.js"></script>
<script src="dist/js/owl.carousel.min.js"></script>
<script src="dist/js/owl.animate.js"></script>
<script src="dist/js/custom.js"></script>
</body>
</html>
```

**Функція додавання книги в бібліотеку, файл «add-book.php»**

```

<?php
session_start();
if (!isset($_SESSION["username"])) {
    ?>
    <script type="text/javascript">
        window.location = "login.php";
    </script>
    <?php
}
include 'inc/header.php';
include 'inc/connection.php';
?>

<!--dashboard area-->
<div class="dashboard-content">
    <div class="dashboard-header">
        <div class="container">
            <div class="row">
                <div class="col-md-6">
                    <div class="left">
                        <p><span>dashboard</span>Control panel</p>
                    </div>
                <div class="col-md-6">
                    <div class="right text-right">
                        <a href="dashboard.php"><i class="fas fa-
home"></i>home</a>
                        <span class="disabled">add book</span>
                    </div>
                </div>
            </div>
            <div class="bstore">
                <form action="" method="post"
enctype="multipart/form-data">
                    <table class="table table-bordered">
                        <tr>
                            <td>
                                <input type="text" class="form-
control" name="booksname" placeholder="Назва"
                                required="">
                            </td>
                        </tr>
                        <tr>
                            <td>
                                Зображення
                                <input type="file" class="form-
control" name="f1" required="">
                            </td>
                        </tr>
                    </table>
                </form>
            </div>
        </div>
    </div>

```

```

        <tr>
            <td>
                Файл
                <input type="file" class="form-
control" name="file" required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bauthorname" placeholder="Автор"
                required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bpubname" placeholder="Видавець"
                required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bpurcdate"
                placeholder="Дата купівлі
книги" required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bprice" placeholder="Ціна" required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bquantity" placeholder="Кількість книг"
                required="">
            </td>
        </tr>
        <tr>
            <td>
                <input type="text" class="form-
control" name="bavailability" placeholder="Доступно книг"
                required="">
            </td>
        </tr>
    </table>
    <div class="submit mt-20">
        <input type="submit" name="submit"
class="btn btn-info submit" value="Add Book">

```

```
        </div>
    </form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<?php
if (isset($_POST["submit"])) {

    $image_name = $_FILES['f1']['name'];
    $file_name = $_FILES['file']['name'];
    $temp = explode(".", $image_name);
    $temp2 = explode(".", $file_name);
    $newfilename = round(microtime(true)) . '.' . end($temp);
    $newfilename2 = round(microtime(true)) . '.' . end($temp2);
    $imagepath = "books-image/" . $newfilename;
    $filepath = "books-file/" . $newfilename2;
    move_uploaded_file($_FILES["f1"]["tmp_name"], $imagepath);
    move_uploaded_file($_FILES["file"]["tmp_name"], $filepath);

    mysqli_query($link, "insert into add_book
values('','$_POST[booksname]','$imagepath','$_POST[bauthorname]','$
$_POST[bpubname]','$_POST[bpurcdate]','$_POST[bprice]','$_POST[bqu
antity]','$_POST[bavailability]','$_SESSION[username]','$filepath'
)");

}
?>

<?php
include 'inc/footer.php';
?>
```