

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

Факультет комп'ютерно-інформаційних систем та програмної інженерії

(повна назва факультету)

Кафедра програмної інженерії

(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: веб-платформа для доступу до інформації та культурних ресурсів:  
«Бібліотека з нами» з використанням РНР

Виконав(ла): студент(ка) 4 курсу, групи СПс-41  
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис)

Бесащук М.П.

(прізвище та ініціали)

Керівник

(підпис)

Михалик Д.М.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Стоянов Ю.М.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Петрик М.Р.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль  
2024

## АНОТАЦІЯ

Кваліфікаційна робота бакалавра. Тернопільський національний технічний університет імені Івана Пулюя, кафедра програмної інженерії, спеціальність 121 «Інженерія програмного забезпечення». ТНТУ, 2024, сторінок 58 , рисунків 21, таблиць 6, презентація.

Мета роботи – проєкту є створення сайту «Бібліотека з нами» з метою культурного збагачення населення. Розроблений веб сайт повинен надавати відвідувачам можливість знайомитися з фондом бібліотеки, переліком книг та їхніми жанрами, переглядати короткий зміст книжки чи читати літературу наявну в електронному вигляді у відкритому доступі.

Метод дослідження включає аналіз та обробка інформації для опису предметної області відповідно до поставленого завдання, а також проєктування бази даних, використовуючи методи нормалізації; написання і відлагодження сайту у середовищі PHP.

Отримані результати створення веб-сайту для бібліотеки відображають значний прогрес у розвитку доступу до літературних ресурсів в місцевих громадах. Сайт дозволяє користувачам знайомитися з фондом бібліотеки, переглядати перелік доступних книг та їхні жанри, а також читати книги у відкритому доступі. Впровадження такого ресурсу сприяє культурному збагаченню мешканців сільських територій та підвищенню інформаційної грамотності в цих регіонах. Крім того, створений веб-сайт забезпечує зручний доступ до літературних ресурсів без необхідності фізичного візиту до бібліотеки. Його функціональність сприяє покращенню обслуговування читачів та забезпеченню їхніх потреб у літературних матеріалах. Завдяки сайту, бібліотечні ресурси стають більш доступними та зручними для користувачів будь-якого віку та рівня інтересів. Такий інноваційний підхід до розвитку бібліотечної сфери сприяє розширенню її впливу та ефективності у задоволенні культурних потреб громади.

БАЗА ДАНИХ, PHP, JS, HTML, CSS, БІБЛІОТЕКА, ПОШУК, КНИГА.

## ANNOTATION

Bachelor's thesis. Ivan Puluj Ternopil National Technical University, Department of Software Engineering, Specialty 121 «Software Engineering». Ternopil National Technical University, 2024, Pages 58, Figures 21, Tables 6, Presentation.

The aim of the project is to create a website called «Library with Us» with the purpose of enriching the culture of the population. The developed website should provide visitors with the opportunity to familiarize themselves with the library's collection, list of books and their genres, view a brief summary of the book, or read literature available in electronic format in open access.

The research method includes analysis and processing of information to describe the subject area in accordance with the set task, as well as database design using normalization methods; writing and debugging the website in the PHP environment.

The results obtained from creating the website for the library reflect significant progress in developing access to literary resources in local communities. The site allows users to familiarize themselves with the library's collection, view a list of available books and their genres, as well as read books in open access. The implementation of such a resource contributes to the cultural enrichment of residents of rural areas and the increase in information literacy in these regions. Additionally, the created website provides convenient access to literary resources without the need for a physical visit to the library. Its functionality improves reader service and meets their needs for literary materials. Thanks to the website, library resources become more accessible and convenient for users of all ages and interests. Such an innovative approach to the development of the library sector contributes to expanding its influence and effectiveness in meeting the cultural needs of the community.

DATABASE, PHP, JS, HTML, CSS, LIBRARY, SEARCH, BOOK.

## ЗМІСТ

ВСТУП.....	7
1 АНАЛІТИЧНА ЧАСТИНА .....	9
1.1 Аналіз предметної області .....	9
1.2 Аналіз існуючих програмних систем.....	10
2 ПРОЄКТНА ЧАСТИНА .....	15
2.1 Постановка завдання .....	15
2.2 Проєктування бази даних.....	15
2.3 Проєктування інформаційної системи.....	23
3 ПРАКТИЧНА ЧАСТИНА .....	27
3.1 Проєктування інтерфейсу користувача .....	27
3.2 Опис програмних модулів.....	30
3.3 Опис результатів тестування .....	32
4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	39
4.1 Долікарська допомога при харчових отруєннях.....	39
4.2 Перша допомога людині, яка уражена електричним струмом .....	40
ПЕРЕЛІК ПОСИЛАНЬ .....	44
ДОДАТОК А – ER-діаграма бази даних «biblioteka».....	45
ДОДАТОК Б – лістинг коду сайту .....	46
ДОДАТОК В – Диск із кваліфікаційною роботою бакалавра .....	58

## ВСТУП

Новий етап розвитку суспільства докорінно впливає на розвиток бібліотечної справи. Інформаційні технології й нові медіа стали неодмінною (умова) частиною життя кожної людини, що значно змінює спосіб отримання інформації та знань. Особливо це стосується молодого покоління, яке віддає перевагу цифровим ресурсам та електронним носіям інформації. Традиційні сільські бібліотеки не є винятком з цього процесу, і все частіше постають перед викликами, пов'язаними зі зниженням відвідуваності серед молоді. Причинами цього явища є доступність інформації в інтернеті, зручність перегляду контенту на смартфонах та персональних комп'ютерах. Окрім того, цифрові ресурси забезпечують швидкий та легкий доступ до необхідної інформації без потреби фізично відвідувати бібліотеку.

В сучасному світі, де інформація стає дедалі більше цифровою, важливо створювати умови для того, щоб бібліотеки залишалися конкурентоспроможними та привабливими для різних категорій населення. Завданням проєкту є створення сайту сільської бібліотеки з метою культурного збагачення мешканців. Розроблений вебсайт повинен надавати відвідувачам можливість знайомитися з фондом бібліотеки, переліком книг та їхніми жанрами, переглядати короткий зміст книжок чи читати літературу, наявну в електронному вигляді у відкритому доступі. Важливо, щоб сайт забезпечував зручний доступ до інформації про нові надходження, популярні книги та рекомендації від бібліотекарів.

В умовах швидкого розвитку інформаційних технологій, створення ефективного вебсайту для бібліотеки стає нагальною потребою. Для розробки сайту буде використовуватися середовище Sublime Text 3, а також технології HTML для гіпертексту, PHP для серверного кодування, CSS3 для каскадних стилів, JavaScript для скриптів, база даних MySQL та система управління базами даних phpMyAdmin. Застосування цих технологій дозволить створити ефективний та надійний вебресурс, який забезпечить стабільну роботу та швидке завантаження сторінок.

Окрім технічних аспектів, важливим є і змістовне наповнення сайту. Він повинен бути не лише функціональним, а й привабливим та корисним для користувачів. Використання сайту сільської бібліотеки дозволить автоматизувати облік літератури, значно спростивши процес управління книжковим фондом. Це дозволить бібліотекарям зосередитися на важливіших завданнях, таких як організація культурних заходів та підтримка зв'язку з громадою.

Сайт зручний тим, що завжди можна спостерігати перелік подій в бібліотеці, що дозволить жителям селища бути в курсі новин бібліотеки та брати активну участь в заходах та зустрічах, що будуть відбуватися там. Це сприятиме не лише підвищенню інтересу до бібліотеки, але й активнішому залученню громади до культурного життя селища. Бібліотеки, які раніше були місцем збереження знань і літературних скарбів, тепер можуть стати справжніми культурними центрами, де відбуваються зустрічі, обговорення, лекції та інші події, що сприяють культурному розвитку громади.

Впровадження таких цифрових рішень дозволить сільським бібліотекам стати сучасними інформаційними центрами, що відповідають потребам сьогодення. Вони зможуть не лише зберігати та надавати доступ до літератури, але й стати місцем спілкування та обміну ідеями, що є надзвичайно важливим для розвитку будь-якого суспільства. Завдяки інтеграції новітніх технологій, бібліотеки зможуть значно розширити свій вплив та залучити нових відвідувачів, що, своєю чергою, сприятиме їхньому розвитку та процвітанню.

Таким чином, створення сайту для сільської бібліотеки є важливим кроком у напрямку адаптації до нових реалій інформаційного суспільства. Це дозволить не лише зберегти, але й примножити культурний потенціал сільських громад, забезпечуючи їм доступ до необхідних знань та ресурсів у зручній та сучасній формі.

## 1 АНАЛІТИЧНА ЧАСТИНА

### 1.1 Аналіз предметної області

Для підтримання роботи підприємства, незалежно від типу власності, або то приватної, чи державної, необхідно рухатися в ногу з часом і намагатися автоматизувати процеси, які охоплюють його діяльність. Одним з ефективних способів автоматизації є створення інтернет-ресурсів, таких як інформаційний сайт. Інформаційні сайти, хоча й не є надто прибутковими, мають велике значення, особливо коли йдеться про розвиток культури та туризму в невеликих містах та селах, таких як Гусятин чи Вільхівчик.

Створення інформаційного сайту для бібліотеки може значно покращити її функціонування та підвищити рівень обслуговування користувачів. Сайт слугуватиме зручним інструментом для взаємодії між бібліотекою та її читачами, надаючи можливість швидко отримати необхідну інформацію, переглянути наявні книги, ознайомитися з новинами та подіями, а також залишити відгуки.

Перед тим, як починати розробку сайту, необхідно розглянути основні ідеї, що лежать в основі роботи кожної класичної бібліотеки, і які допоможуть нам сформувати базу даних, що забезпечуватиме діяльність сайту, а також визначити основні зв'язки, які потрібно сформувати між її таблицями.

Також не варто забувати про рекламу для самого сайту, адже це буде постійно збільшувати кількість відвідувачів. Більше відвідувачів означає додаткові кошти, що можуть допомогти розвитку бібліотеки. Реклама сайту може здійснюватися через соціальні мережі, місцеві засоби масової інформації, а також за допомогою партнерських програм з іншими установами культури.

Онлайн ознайомлення із послугами бібліотеки, перегляд графіку роботи, можливість прочитати деякі примірники онлайн – все це суттєво спрощує роботу для працівників бібліотеки та зацікавлених у книгах людей, адже їм не знадобиться йти до бібліотеки для того, щоб дізнатись, які книги вона має. Це значно підвищує

зручність для користувачів і робить бібліотеку доступнішою для широкого кола людей.

Сайт також сприяє притоку нових людей, які раніше не звертали уваги на бібліотеку, що, звісно, добре впливає на її стан та розвиток. Сайт виступає в ролі реклами для бібліотеки, і дає змогу швидко та зручно ознайомитися з необхідною інформацією, прочитати книгу онлайн або завантажити її.

Завдяки створенню інформаційного сайту бібліотека зможе залучити більше користувачів, покращити якість обслуговування, а також підвищити свою популярність і значущість у культурному житті громади. Інтернет-ресурс стане важливим інструментом для підтримання та розвитку бібліотечної справи, сприяючи збереженню і поширенню знань серед населення.

## 1.2 Аналіз існуючих програмних систем

В наш час, коли інтернет розвивається з надзвичайною швидкістю, важливо використовувати його для зменшення затрат часу на різні дії, адже треба йти нога в ногу з можливостями.

Звичайні книги є все ще дуже популярними, проте багато бібліотек втрачають своє місце у сучасному світі. Людям можуть прийти до бібліотеки, та не знайти необхідної книги, що змушує ще більше витратити часу на пошуки.

Для того, щоб зменшити затрати часу на подібні дії, створюють сайти, які спрощують подібні процеси. Для того, щоб створити подібний проєкт важливо проаналізувати подібні рішення та виділити всі їх плюси та мінуси.

Сайт «Національна бібліотека України імені В.І Вернадського» (<http://www.nbuv.gov.ua/>) надає інформацію про бібліотеку, а також подає новини, пов'язані із нею, для звичайних користувачів та працівників [1].





Рисунок 1.1 – Головна сторінка сайту «Національна бібліотека України імені В.І Вернадського»

Дизайн сайту не є надто сучасним, кольори підібрані гармонійно. З переваг можна виділити те, що інтерфейс є зручним та зрозумілим, сайт переведено на сорок мов.

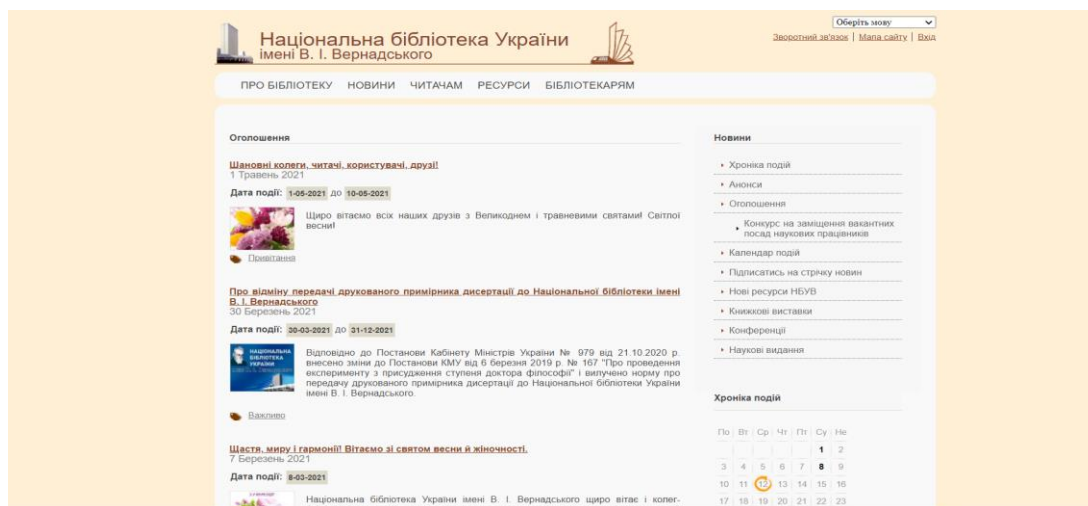


Рисунок 1.2 – сторінка оголошень «Національна бібліотека України імені В.І Вернадського»

З недоліків можна виділити малий шрифт, та трішки перевантажені інформацією сторінки.

Сайт бібліотеки «Тернопільська обласна універсальна наукова бібліотека» (<https://library.te.ua/>) надає можливість ознайомлюватись із новинами пов'язаними із бібліотекою, а також читати онлайн [2].



Рисунок 1.3 – Головна сторінка сайту «Тернопільська обласна універсальна наукова бібліотека»

Дизайн сайту виконано у стилі мінімалізму та складається із класичних, чорного та білого кольорів. Окрім інформації про бібліотеку, сайт надає змогу брати участь у віртуальних виставках.

Серед переваг можна виділити систему акцій, зрозумілий інтерфейс, серед недоліків не виділено нічого критичного.



Рисунок 1.4 – Сторінка події та новини «Тернопільська обласна універсальна наукова бібліотека»

Вебсайт було написано за допомогою HTML5, CSS3, JavaScript та PHP.

Сайт бібліотеки «Державна бібліотека України для юнацтва» (<http://4uth.gov.ua/#>) Надає інформацію зв'язану з бібліотекою, а також заходами для молодих людей, що захоплюються книгами [3].

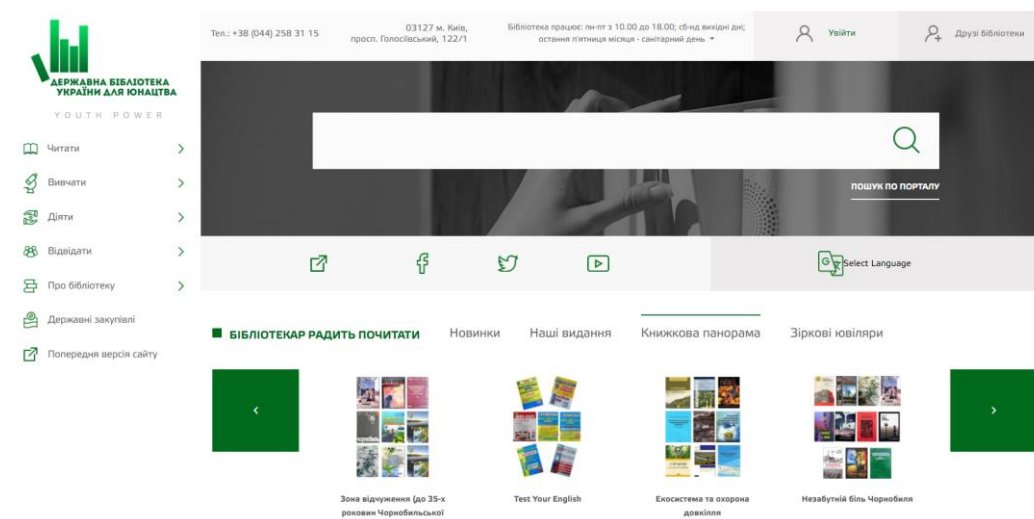


Рисунок 1.5 – Головна сторінка сайту «Державна бібліотека України для юнацтва»

Інтерфейс сайту виконано у 3 кольорах: зелений, білий, чорний. До переваг сайту входить зручна навігаційна панель та швидке завантаження, серед недоліків можна виділити елементи сайту, межі яких зливаються з білим фоном, що створює дискомфорт. З урахуванням цих особливостей, важливо регулярно оновлювати дизайн та проводити тестування на зручність користування.

Як і вищеописані, сайт створений за допомогою JavaScript, CSS3, HTML5 та PHP. Сайт «Миколаївська обласна бібліотека для юнацтва» (<http://unbib.mk.ua/>) організовує читацькі збори, а також активно просуває свої заходи в соціальних мережах для залучення більшої аудиторії.

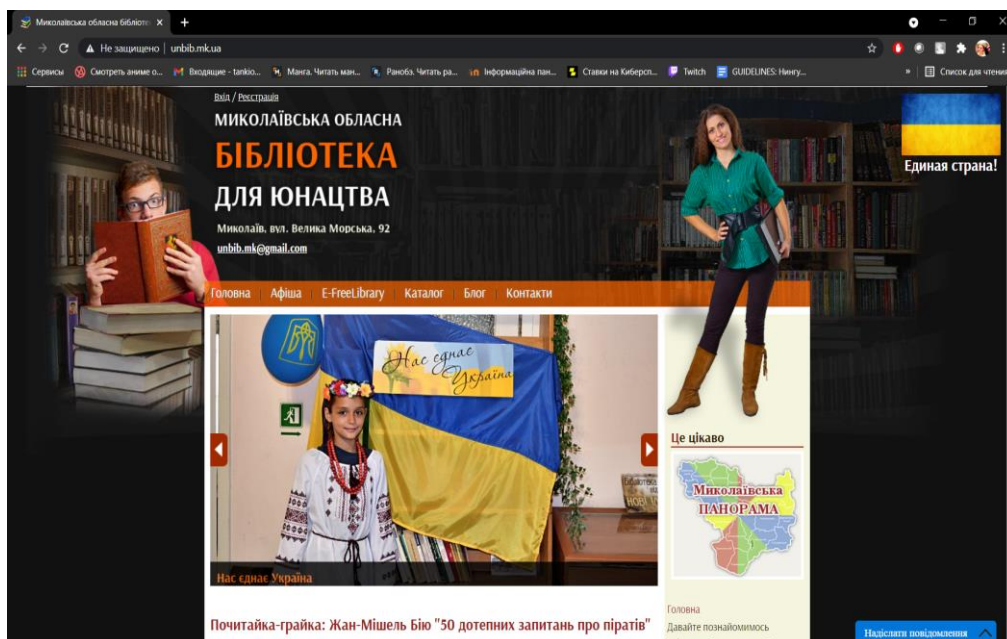


Рисунок 1.6 – Головна сторінка сайту «F1center»

Дизайн сайту застарілий. До його переваг можна віднести: всі необхідні елементи для подібних сайтів присутні та знаходяться на видному місці, зручна навігаційна панель. Серед недоліків виділяється нагромадженого інтерфейсу текстом. Також слід зазначити, що відсутність сучасних елементів дизайну може знижувати привабливість сайту для молодшої аудиторії. Деякі користувачі можуть відчувати труднощі при пошуку необхідної інформації через велике текстове наповнення. Попри це, сайт залишається функціональним та виконує свої основні завдання. Важливо регулярно оновлювати контент та дизайн, щоб відповідати сучасним вимогам та очікуванням користувачів.

Як і всі інші, сайт створено з використанням HTML5, CSS3, JavaScript і PHP. Використання цих технологій забезпечує стабільність роботи сайту та його сумісність з різними браузерами та пристроями.

## 2 ПРОЄКТНА ЧАСТИНА

### 2.1 Постановка завдання

Для організації роботи бібліотеки, покращення взаємодії між потенційними відвідувачами та робітниками бібліотеки, розробити сайт сільської бібліотеки.

Вхідними даними додатку є інформація про книги та дані кожного користувача.

Для покращення користувацького досвіду та функціональності сайту необхідно створити систему адміністрування з визначеними ролями користувачів. Сайт повинен мати зручний інтерфейс, на якому буде відображено повний перелік доступних книг і можливість легкого пошуку. Функціонал сайту повинен включати:

- Відображення доступних книг на сайті
- Надання доступу до оновлень новин
- Представлення детальної інформації про бібліотеку
- Можливість користувачам залишати відгуки
- Можливість користувачам переглядати та взаємодіяти з наявними відгуками

Крім того, важливо забезпечити цілісність та точність даних за допомогою автоматизованих методів введення та перевірки даних.

### 2.2 Проєктування бази даних

Щоб розробити базу даних, яка допомагатиме обслуговувати бібліотеку, потрібно передбачати режими ведення системного каталогу, що відображає перелік жанрів книг у бібліотеці. У систематичному каталозі бібліотеки кожен жанр може мати унікальний внутрішній номер та повну назву. Кожна книга може належати до

декількох жанрів. Кожна книга в бібліотеці може бути присутня у декількох примірниках [4].

Управління базою даних буде здійснювати адміністратор та працівник бібліотеки, а також читачі. Адміністратор матиме повний доступ до всієї бази даних, в той час, як доступ працівника буде обмежений, дозволяючи йому контролювати асортимент та замовлення книг. Читачам буде надано можливість перегляду книг та іншої доступної інформації на сайті. Після проведення аналізу предметної області ми визначимо необхідні інформаційні об'єкти бази даних. Одним з методів ідентифікації об'єктів, є вивчення вимог користувачів. ЗадOCUMENTУємо сутності в таблиці 2.1.

Таблиця 2.1 – Опис сутностей бази даних

Ім'я сутності	Опис	Псевдоніми	Особливості використання
1	2	3	4
Користувачі	Узагальнюючий термін, що описує весь персонал, який користується базою даних	users	Кожен із користувачів має свій метод доступу
Читачі	Термін, що описує таблицю, де зберігається інформація про читачів	readers	Інформація про користувачів які замовляють послуги у бібліотеки
Книги	Таблиця, в якій зберігається інформація перелік книг, які надає бібліотека	books	Надання інформації про всі книги
Працівники	Таблиця, в якій зберігається інформація про працівників підприємства	worker	Кожен із працівників має свої обов'язки

Продовження таблиці 2.1 – Опис сутностей бази даних

1	2	3	4
Жанр	Таблиця, в якій зберігається інформація про різноманітні жанри	genre	Кожна книга має свій жанр
Формуляр	Таблиця, в якій зберігається інформація видачі книг читачам	form	Інформація про дату видачі списання певної книги певним читачем

Проаналізувавши виділені інформаційні об'єкти, визначимо первинні та зовнішні ключі для кожної сутності та внесемо цю інформацію у таблицю 2.2.

Таблиця 2.2 – Опис атрибутів сутностей, їхніх доменів і характеристик

Тип сутності	Атрибут	Опис	Домен (тип даних, довжина)	Обмеження	Значення за замовчуванням	Псевдонім	Значення NULL
1	2	3	4	5	6	7	8
Користувачі	Код користувача	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	id	Ні
	Логін	Логін користувача	Символьний тип	-	-	login	Ні
	Пароль	Пароль	Символьний тип	-	-	password	Ні
Читачі	Код клієнта	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	id	Ні
	Прізвище	Прізвище читача	Текстовий тип	-	-	Surname	Ні
	Ім'я	Ім'я читача	Текстовий тип	-	-	Name	Ні
	По батькові	По батькові читача	Текстовий тип	-	-	Twoname	Ні

Продовження таблиці 2.2 – Опис атрибутів сутностей, їхніх доменів і характеристик

1	2	3	4	5	6	7	8
Жанр	Код Жанру	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	Id	Ні
	Назва	Назва жанру	Текстовий тип	-	-	genre	Ні
Працівники	Код працівника	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	id	Ні
	Прізвище	Прізвище працівника	Текстовий тип	-	-	Surname	Ні
	Ім'я	Ім'я працівника	Текстовий тип	-	-	Name	Ні
	По батькові	По батькові працівника	Текстовий тип	-	-	Twoname	Ні
	Посада	Посада працівника	Текстовий тип	-	-	work	Ні
Формуляр	Код формуляру	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	id	Ні
	Читач	Унікальний номер читача	Числовий тип, цілий	Зовнішній ключ -	-	id_readers	Ні
	Дата повернення	Дата повернення книги	Дата тип	-	-	data_return	Ні
	Дата видачі	Дата отримання книги	Дата тип	-	-	data_receiving	Ні
	Шифр книги	Унікальний номер книги	Числовий тип, цілий	Зовнішній ключ	-	pressmark	Ні
Книги	Код книги	Унікальний ідентифікатор	Числовий тип, цілий	Первинний ключ	-	id	Ні
	Назва	Унікальний ідентифікатор	Текстовий тип	-	-	name	Ні
	Автор	Автор книги	Текстовий тип	-	-	author	Ні
	Рік випуску	Дата випуску книги	Тип дата	-	-	year_graduation	Ні



Продовження таблиці 2.2 – Опис атрибутів сутностей, їхніх доменів і характеристик

1	2	3	4	5	6	7	8
Книги	Видавництво	Видавництво книги	Текстовий тип	-	-	publication	Ні
	Кількість сторінок	Кількість сторінок в книзі	Числовий тип, цілий	-	-	number_pages	Ні
	Кількість книг	Кількість книг в бібліотеці	Чисельний тип, цілий	-	-	number_books	Ні
	Код жанру	Унікальний код жанру	Чисельний тип, цілий	Зовнішній ключ	-	id_genre	Ні

Щоб отримати наочне представлення основних сутностей та типів зв'язків слід використати ER-діаграму, щоб графічно передати всі елементи інформаційної моделі

Визначимо первинні ключі для кожної сутності і зовнішні ключі для створення зв'язків. У таблиці 2.3 буде представлено характеристики зв'язків виділених сутностей, а ER-діаграму розміщено в додатку А.

Таблиця 2.3 – Опис зв'язків між сутностями бази даних «biblioteka»

Ім'я сутності 1	Ім'я сутності 2	Ім'я зв'язку	Тип зв'язку
1	2	3	4
worker	form	Оформляє	1 : N
form	readers	Виконує	1 : N
form	books	Виконує	1 : N
books	genre	Постачає	1 : N

Виходячи з того, що в якості моделі даних вибрано реляційну, то розглянемо особливості спроектованої концептуальної схеми бази даних «biblioteka». Проаналізуємо зв'язки між сутностями бази даних. Зв'язків між двома таблицями розробленої схеми відсутній тип зв'язку «багато-до-багатьох». Також не знайдено жодних зв'язків між трьома чи більше сутностями.

У концептуальній моделі даних відсутні багатозначні атрибути. Всі зв'язки між сутностями у цій моделі є типу «один-до-одного» або «один-до-багатьох». Складних та рекурсивних зв'язків не виявлено [4].

Отже, розроблена концептуальна схема проєктується у логічній схемі кожна сутність перетворюється в окреме відношення. Зв'язки між відношеннями відображаються за допомогою механізму первинних та зовнішніх ключів.

Для досягнення першої нормальної форми бази даних «biblioteka», кожен атрибут відношень, таких як «readers», «books», «worker», «genre», «form» та «users», повинен бути неподільним і не містити повторюваних груп даних. Перевіривши відношення, ми встановили, що вони відповідають цим вимогам і можуть бути нормалізованими до першої нормальної форми.

Для досягнення другої нормальної форми, ми аналізуємо функціональні залежності між описовими атрибутами та первинним ключем. За результатами аналізу, ми визначили, що описові атрибути кожного відношення функціонально залежать від свого первинного ключа. Отже, вони можуть бути нормалізованими до другої нормальної форми.

Для досягнення третьої нормальної форми, ми видаляємо транзитивні функціональні залежності між атрибутами відношень. Проаналізувавши усі відношення бази даних «biblioteka», ми встановили, що транзитивних функціональних залежностей між атрибутами не існує. Тому відношення «readers», «books», «worker», «genre», «form» та «users» можуть бути нормалізованими до третьої нормальної форми.

В результаті проведення аналізу кожного відношення бази даних і виконання нормалізації до третьої нормальної форми, ми забезпечили цілісність та ефективність даних у базі даних «biblioteka». На наступному етапі ми визначимо вимоги до забезпечення цілісності даних у логічній моделі даних. Спочатку визначимо, які з атрибутів повинні містити обмеження NOT NULL, тобто дані для них повинні бути внесені обов'язково. До таких атрибутів відносяться первинні й зовнішні ключі відношення та основна інформація про книги, видавництва, читачі тощо [4].

Наступне обмеження полягає у забезпеченні цілісності посилань, що також включає в себе вимогу до встановлення значення NOT NULL. Взаємозв'язки між таблицями моделюються за допомогою первинних та зовнішніх ключів. Якщо в таблиці існує зовнішній ключ, його значення повинно відповідати значенню первинного ключа іншої таблиці або бути NULL. Навіть якщо в базі даних визначені зв'язки «один-до-багатьох», обмеження NOT NULL також встановлюється для зовнішніх ключів [4].

Правила корпоративного забезпечення цілісності встановлюються користувачами або адміністраторами бази даних. Проаналізувавши предметну область, можна визначити декілька таких обмежень:

- читач може взяти у бібліотеці декілька різних книг;
- співробітники бібліотеки відповідають за оформлення та внесення даних про формуляри читачів.

Отже, на цьому етапі проведено аналіз відношень щодо цілісності даних в базі «biblioteka».

Враховуючи розглянуту предметну область, створимо базу даних «biblioteka» та таблиці: «users», «worker», «readers», «genre», «books», «form» і встановимо зв'язок між ними.

Нехай ми створимо таблицю "users" за допомогою наступного SQL-запиту:

Лістинг 2.1 – Текст SQL-команди для створення таблиці "users" у базі даних "biblioteka".

```
Create Table users(
  id int Primary Key,
  login varchar(20) Not Null,
  password varchar(20) Not Null)
```

Створимо таблицю "worker" за допомогою наступного SQL-запиту.

Лістинг 2.2 – SQL-код для створення таблиці «worker» бази даних «biblioteka»

```
Create Table worker(
```

Продовження лістингу 2.2 –SQL-кож для створення таблиці «worker» бази даних «biblioteka»

```
id int Primary Key,
login varchar(20) Not Null,
password varchar(20) Not Null,
work varchar(20) Not Null,
PIP varchar(50) Not Null)
```

Сформуємо таблицю «readers» за допомогою наступного SQL- запиту:

Лістинг 2.3 – Текст SQL-запиту для створення таблиці «readers» бази даних «biblioteka»

```
Create Table readers (
id int Primary Key,
login varchar(20) Not Null,
password varchar(20) Not Null,
PIP varchar(50) Not Null)
```

Створимо таблицю «genre» за допомогою наступного SQL- запиту:

Лістинг 2.4 – Текст SQL-запиту для створення таблиці «genre» бази даних «biblioteka»

```
Create Table genre (
id int Primary Key,
name varchar(20) Not Null)
```

Давайте створимо таблицю «books» за допомогою наступного SQL- запиту:

Лістинг 2.5 – SQL-код для створення таблиці «books» бази даних «biblioteka»

```
Create Table books(
id int Primary Key,
name varchar(20) Not Null,
author varchar(40) Not Null,
year_graduation date Not Null,
publication varchar(20) Not Null,
number_pages int Not Null,
number_books int Not Null,
id_genre int Not Null,
Foreign Key (id_genre) References genre (id))
```

Сформуємо таблицю «form» за допомогою наступного SQL- запиту

## Лістинг 2.6 – Текст SQL-запиту для створення таблиці «form» бази даних «biblioteka»

```
Create Table form(  
id int Primary Key,  
id_readers int Not Null,  
Foreign Key (id_readers) References readers (id),  
data_return date Not Null,  
data_receiving date Not Null,  
pressmark int Not Null,  
Foreign Key (pressmark) References books (id))
```

### 2.3 Проектування інформаційної системи

Розробку сайту бібліотеки виконано з використанням стандартної мови моделювання UML (Unified Modeling Language). UML дозволяє застосовувати на будь-якому етапі життєвого циклу розробки програмного забезпечення. Це універсальна мова з відкритим стандартом, яка застосовує графічні символи використовуються для створення абстрактної моделі системи, відомої як UML-модель. При проектуванні інформаційної системи необхідно створити діаграму прецедентів (Use Case Diagram), діаграму класів (Class Diagram) та діаграму послідовності (Sequence Diagram) [5].

Діаграма прецедентів показує взаємодію між акторами та прецедентами в системі. Вона зображує систему як набір сутностей або акторів, які взаємодіють із системою через варіанти використання. Актори «Користувач» та «Авторизований користувач» можуть використовувати навігаційне меню для переходу на всі сторінки сайту, переглядати інформацію та користуватися функцією пошуку. Крім цього, «Авторизований користувач» має можливість переглядати асортимент книг та залишати відгуки про сайт. «Адміністратор» має доступ до ресурсів сайту та бази даних для виконання адміністративних завдань і роботи з інформацією [5]. Діаграму прецедентів представлено на рисунку 2.1.

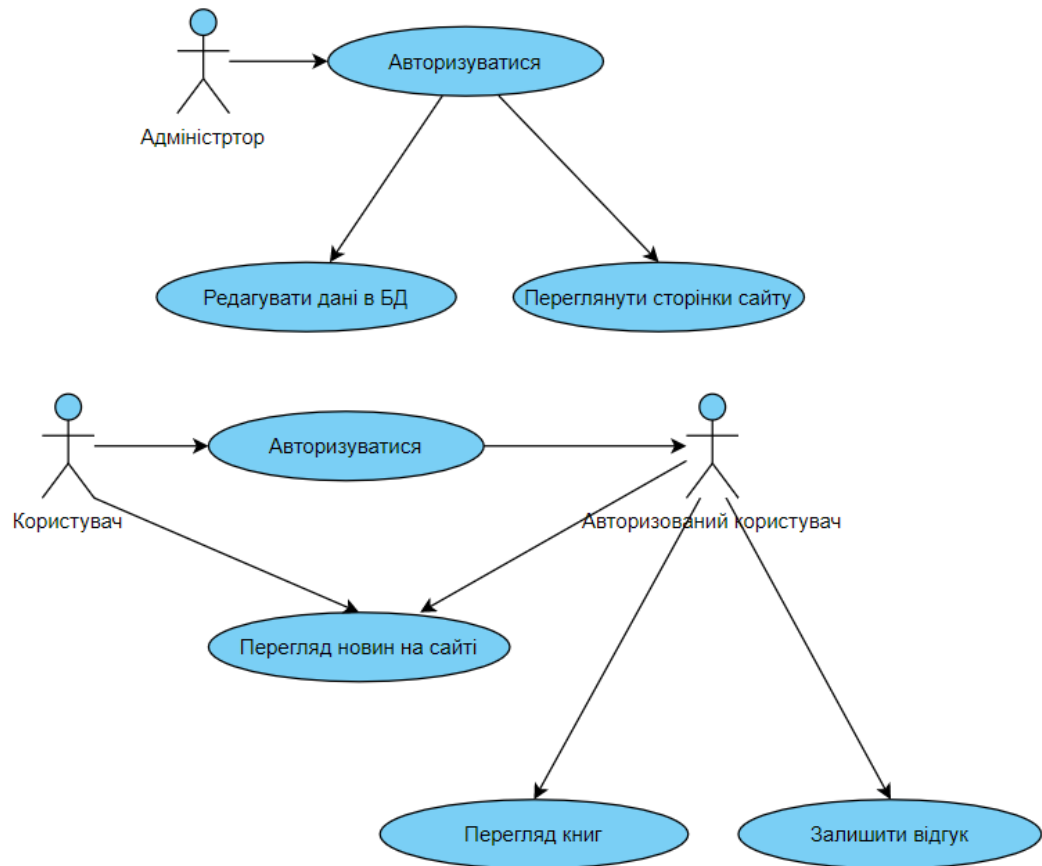


Рисунок 2.1 – Діаграма прецедентів сайту бібліотеки

Після створення діаграми прецедентів була побудована діаграма класів. Вона складається з прямокутників, які представляють класи, та зв'язків між ними. Прямокутник класу може включати ім'я класу, список атрибутів та операцій класу, причому ці частини можуть бути присутніми або відсутніми. Діаграма класів використовується для зображення структурної організації системи в контексті класів об'єктно-орієнтованого програмування.

Діаграма класів являє собою граф, у якому вершинами є класифікатори, з'єднані різними типами структурних відносин. Вона також може включати інтерфейси, групування, зв'язки та конкретні випадки, такі як об'єкти та їх взаємозв'язки. Це графічне зображення структурних зв'язків у логічній моделі системи, що залишаються незмінними в часі [6].

Робота з сайтом починається з головної сторінки, яка підключається до бази даних для інтеграції її із сайтом. Інформація на деяких сторінках сайту отримується з бази даних, яка взаємодіє з усіма основними сторінками, надаючи необхідну інформацію та забезпечуючи функцію пошуку. Діаграма класів для сайту сільської бібліотеки представлена на рисунку 2.2.

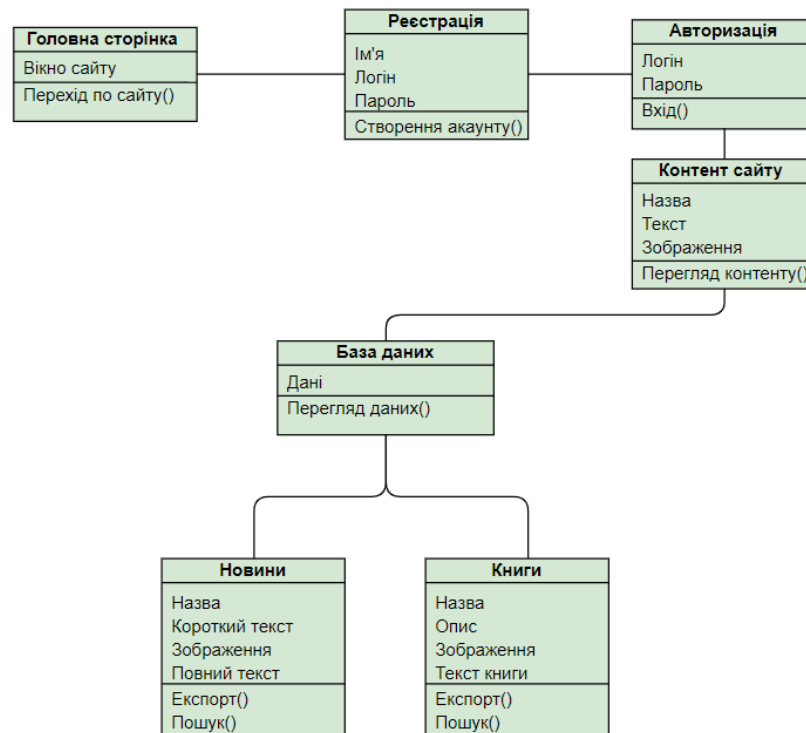


Рисунок 2.2 – Діаграма класів сайту сільської бібліотеки

Тепер перейдемо до створення діаграми послідовності, яка наочно відображає взаємодію об'єктів у часі. На цій діаграмі показуються лише ті об'єкти, які безпосередньо беруть участь у взаємодії, без урахування можливих статичних асоціацій з іншими об'єктами. Для кожного об'єкта сценарію призначається лінія життя, що відображає перебіг подій від моменту створення до знищення об'єкта. Графічно фігура прямокутника розташована у верхній частині лінії життя об'єкта, де вказані ім'я об'єкта та ім'я класу, розділені двокрапкою [7].

Діаграма послідовності показує, як працює функція пошуку протоколів на сайті. Користувач вводить дані для пошуку на сторінці пошуку, що створює

критерій для пошуку та перевіряє його у пошуковій системі. Пошукова система, звернувшись до бази даних, отримує список протоколів і порівнює критерій пошуку з елементами списку, виводячи результат. Якщо збіг виявлено, запис буде показаний користувачеві, в іншому випадку система повідомить про відсутність інформації за заданим критерієм. Діаграма послідовностей для сценарію пошуку на сайті сільської бібліотеки зображена на рисунку 2.3.

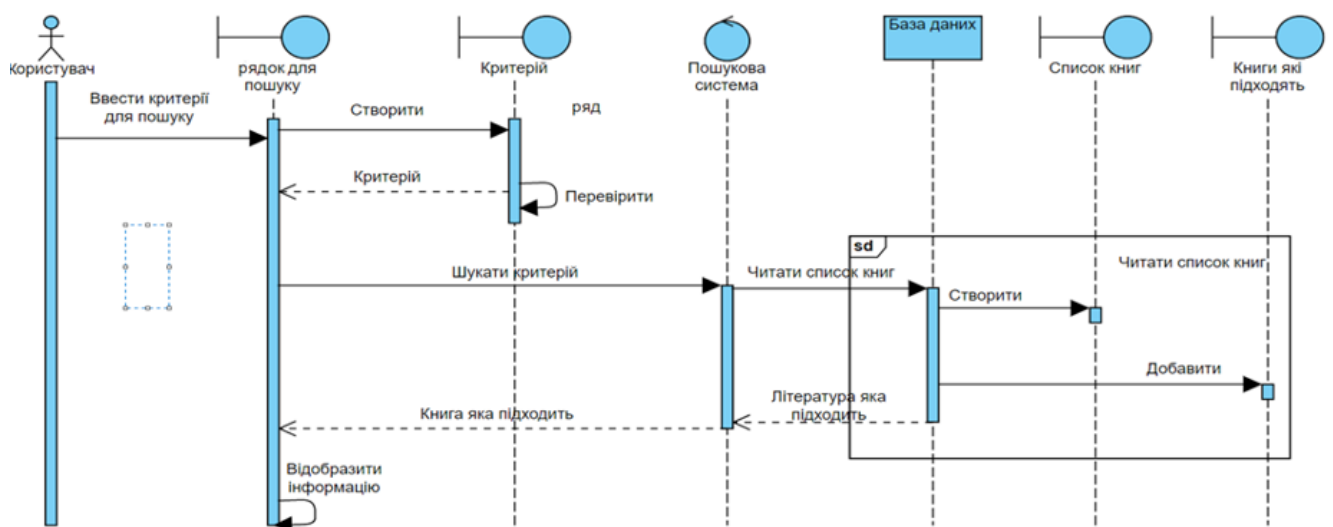


Рисунок 2.3 – Діаграма послідовності для сценарію пошуку книги

На даному етапі було розроблено діаграму прецедентів, а також діаграми класів та послідовності за допомогою уніфікованої мови моделювання (UML). Ці кроки визначили структуру та взаємодію компонентів інформаційної системи для створення веб-сайту [8].



## 3 ПРАКТИЧНА ЧАСТИНА

### 3.1 Проєктування інтерфейсу користувача

Головна сторінка сайту сільської бібліотеки відображає навігаційне меню яка включає: Головна, Новини, Книги, Адмін панель, Відгуки (див. Рис. 3.1).



#### Про бібліотеку

Рисунок 3.1 – Навігаційне меню

На сторінці «Новини» розміщено новини про головні події та заходи Вільхівчицької бібліотеки (див. Рис. 3.2).

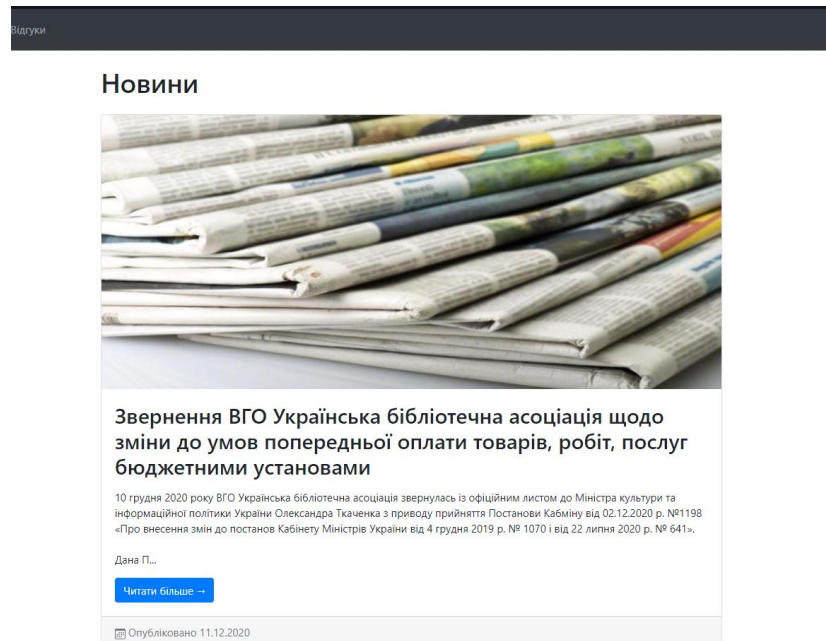


Рисунок 3.2 –Сторінка «Новини»

Для коректної роботи із сайтом необхідно авторизуватися перед початком роботи (див. Рис. 3.3).

Вхід

admin

.....

Авторизуватися Зареєструватися

Рисунок 3.3 – Форма авторизації

Вікно реєстрації наведено на рисунку 3.4.

Форма реєстрації

Введіть логін

Введіть ім'я

Введіть пароль

Зареєструвати

Рисунок 3.4 – Форма реєстрації

При переході на сторінку «Книги», представлено книги які наявні у бібліотеці для ознайомлення (див. Рис. 3.4).

Кайдашева сім'я

Автор: Іван Нечуй-Левицький

Рік випуску: 1878

Видавництво: Журнал «Правда»

Кількість сторінок: 232

Кількість книг: 7

Завантажити

Рисунок 3.4 – Сторінка «Книги»

У цьому вікні можна натиснути кнопку Завантажити, після чого файл в pdf-форматі з текстом книги відобразиться на вікні і буде доступний для перегляду (див. Рис. 3.5).

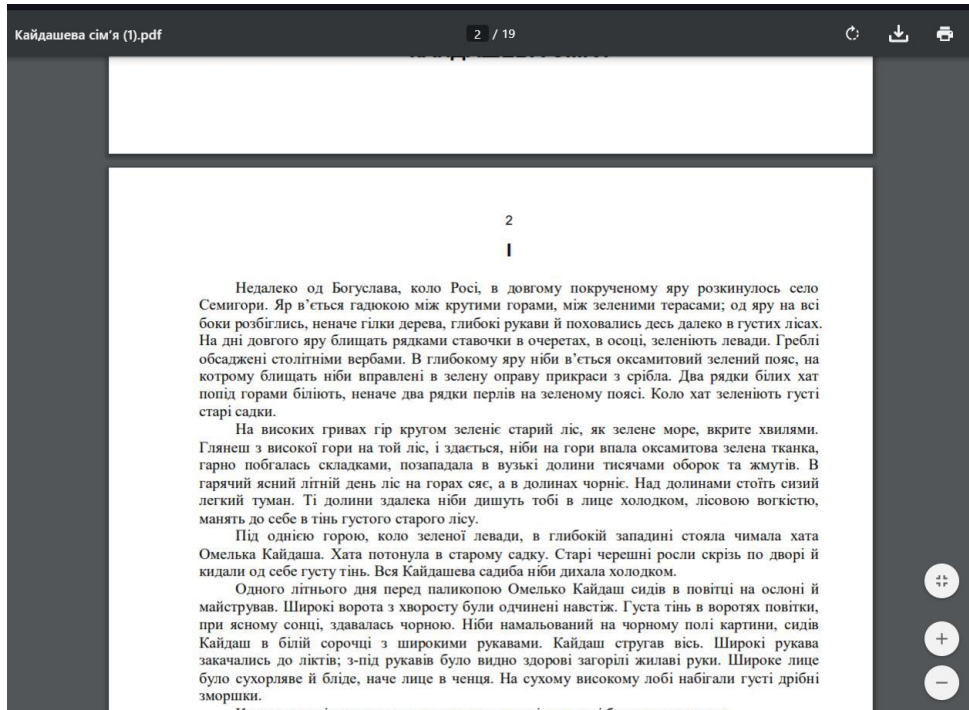


Рисунок 3.5 – Тестовий файл з текстом книги

Під час роботи з правами адміністратора, буде доступний пункт меню «Адмін панель» (див. Рис. 3.6).



## Книги

Рисунок 3.6 – «Адмін панель»

Вибір цього пункту відкриє сторінку з правами адміністратора, у якій можна редагувати дані в таблицях бази даних (див. Рис. 3.7).

## Адмін Панель

### Таблиця користувачів

Номер	Логін	Пароль	Вид користувача	
1	admin	admin	1	<a href="#">Видати</a>

### Таблиця новин

Номер	Заголовок	Зображення	Зміст	Дата	
1	Звернення ВГО Українська бібліотечна асоціація щодо зміни до умов попередньої оплати товарів, робіт, послуг бюджетними установами	img/1.png	<p>10 грудня 2020 року ВГО Українська бібліотечна асоціація звернулась із офіційним листом до Міністра культури та інформаційної політики України Олександра Ткаченка з приводу прийняття Постанови Кабміну від 02.12.2020 р. №1198 «Про внесення змін до постанов Кабінету Міністрів України від 4 грудня 2019 р. № 1070 і від 22 липня 2020 р. № 641».</p> <p>Дана Постанова вносить зміни до умов попередньої оплати товарів, робіт, послуг бюджетними установами. Під її дію підпадають і бібліотеки. Прийнята Постанова унеможливує передплату періодичних видань на 2021 рік, що призведе до незворотних негативних наслідків поповнення бібліотечних фондів. ВГО Українська бібліотечна асоціація впевнена, що в складний час пандемії громадянам необхідно мати безкоштовний доступ до актуальної періодики. А бібліотеки зобов'язані надати таку можливість кожному, хто цього потребує.</p> <p>Після кількох днів переслідування магією вибуху Мегуміна, Бельдій була роздратована і назвала її божевільною дівчиною («Дівчина з ослабленими гвинтами»). На превеликий жаль для Мегумін, так як прізвисько описує її дуже добре, «божевільна дівчина» також стала її офіційним прізвищем в Аксель. Казума заявив, що Мегумін має самий здоровий глузд серед дівчат на вечірші, але, з огляду на її конкуренцію, це не дуже комплімент. Мегумін також любить брати участь (або починати) дитячі випівки з Аквої або Темрявою, і часто діє занадто</p>	11.12.2020	<a href="#">Видати</a>

Рисунок 3.7 –Сторінка «Адміністратора»

## 3.2 Опис програмних модулів

З виокремленого технічного завдання та потреб користувачів визначимо та розглянемо основні програмні файли сайту.

Сайт було створено за допомогою HTML5 для структуризації, CSS3 для оформлення та стилізації, PHP7 для серверної логіки, JavaScript для інтерактивності, а також з використанням бази даних MySQL та інструменту керування базами даних phpMyAdmin. Для поєднання описаних було визначено та розглянуто технології разом з можливостями кінцевого користувача взаємодіяти з готовим продуктом. програмні модулі [9].

Опис основних програмних модулів наведено у таблиці 3.1.

Таблиця 3.1 – Опис програмних модулів сайту сільської бібліотеки

№	Назва програмного модуля	Короткий опис програмного модуля
1	index.php	Файл індексу веб-сайту. Є основним файлом в структурі проекту.
2	admin.php	Файл, який містить верстку та стилізацію сторінку адміністратора.
3	category.php	Файл, який містить верстку та стилізацію сторінки «Новини» та «Книги», де знаходиться інформація про новини про головні події та заходи і книги які наявні у бібліотеці для ознайомлення.
4	comen.php	Файл, який містить верстку та стилізацію сторінки користувача, де користувач може лишити відгук на сайті.
5	post.php	Файл, що містить розмітку та оформлення сторінки користувача, де знаходиться розширена інформація про певну новини, чи подію.
6	header.php	Файл, де зберігається розмітка та оформлення сторінки. верхньої частини сайту.
7	main.css	Головний файл каскадних таблиць стилів для веб-сайту.
8	footer.php	Файл, який містить верстку та стилізацію нижньої частини сайту, де знаходиться інформація про розробника.
9	function.php	Файл, який містить функціонал сайту.
10	auth.php	Файл, який містить код авторизації користувача.
11	check.php	Файл, який містить код реєстрації нового користувача.
12	conf.php	Файл, який містить підключення до бази даних.
13	register.php	Файл, який містить верстку та стилізацію сторінки «Реєстрація», надає інформацію про процес реєстрації користувача на веб-сайті.

Модулі програм, перераховані у таблиці 3.1, розкривають структуру створеного програмного засобу, надаючи інформацію про їх функції, взаємодію та можливі варіанти використання в рамках проекту [9].

### 3.3 Опис результатів тестування

Перед тим як надавати доступ користувачам сайту, необхідно протестувати сайт, щоб виправити помилки, які в процесі розробки були допущені.

Під час тестування сайту перевіряється його функціонал та відповідність технічному завданню. Залежно від початкового технічного завдання, у процесі тестування можуть бути виконані наступні перевірки:

- перевірка на адаптивність;
- перевірка html/css.
- перевірка на кросбраузерність;
- виконання перевірки роботи гіперпосилань на веб-сайті;
- аналіз usability;

Для перевірки сайту на адаптивність буде використано 3 контрольні точки, вони керують розширенням екрану пристрою у ширину [10].

Перша точка контролю - 1200px. Зображення головної сторінки сайту при першій точці контролю подано на малюнку 3.8.

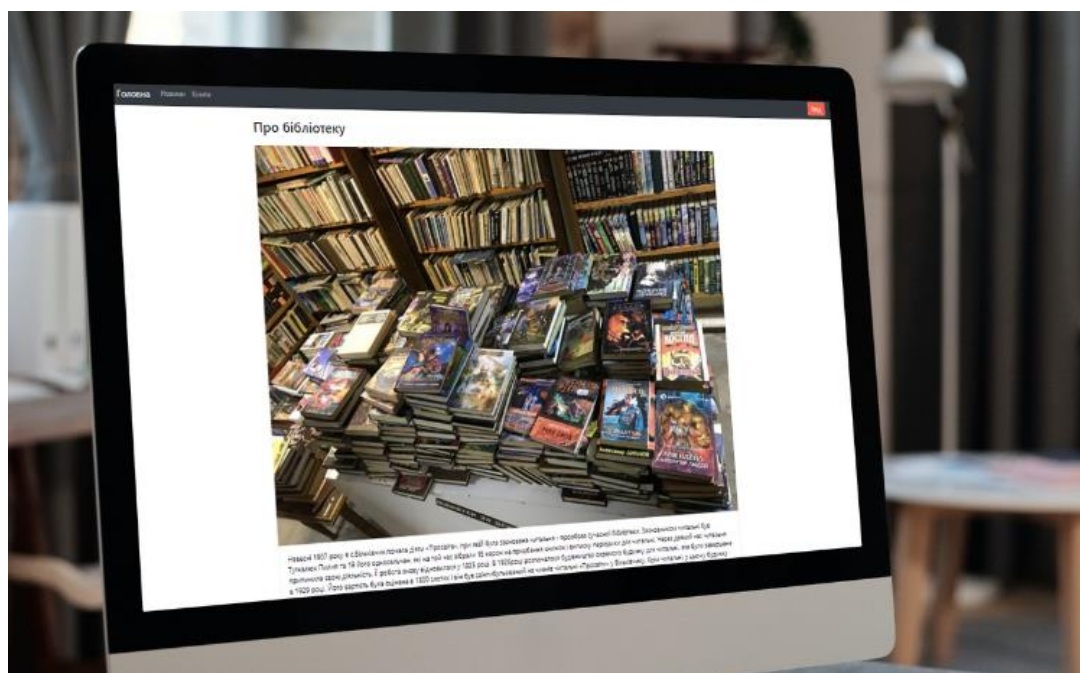


Рисунок 3.8 – Стиль головної сторінки веб-сайту при першій точці контролю.



Друга точка контролю - 920px. Зображення головної сторінки веб-сайту при другій точці контролю подано на малюнку 3.9.



Рисунок 3.9 – Стиль головної сторінки веб-сайту при другій точці контролю.

Третя точка контролю - 680px. Зображення головної сторінки веб-сайту при третій точці контролю подано на малюнку 3.10.

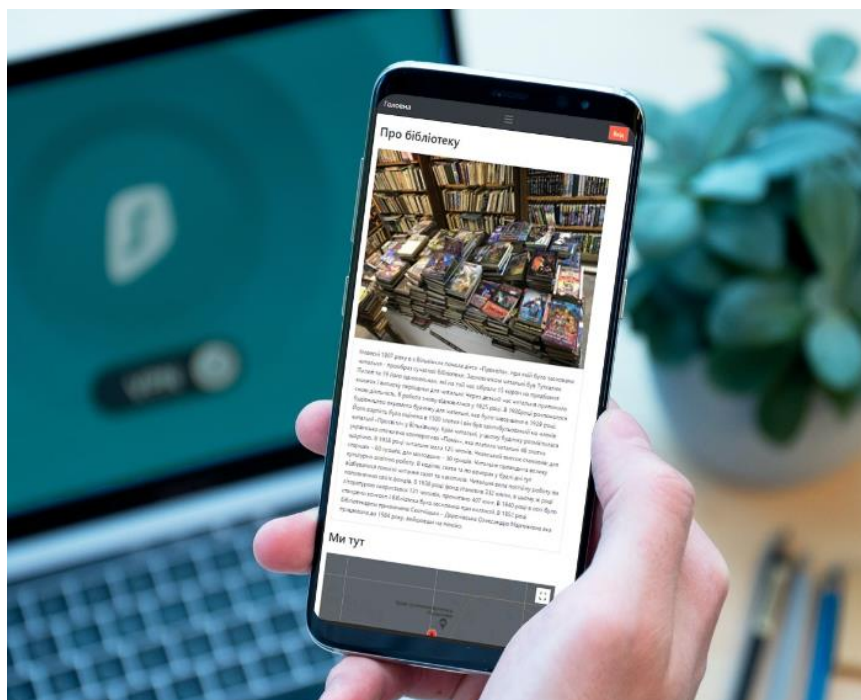


Рисунок 3.10 – Стиль головної сторінки веб-сайту при третій точці контролю.

Для оцінки HTML та CSS верстки використаємо W3C validator, який автоматично знайде помилки в коді.

При перевірці файлу головної сторінки сайту, W3C validator знайшов 3 помилки. Всі помилки пов'язані з мовою PHP, тому їх можна не брати до уваги, оскільки вони не відносяться до HTML. Результат html тестування за допомогою W3C validator наведено на рисунку 3.11

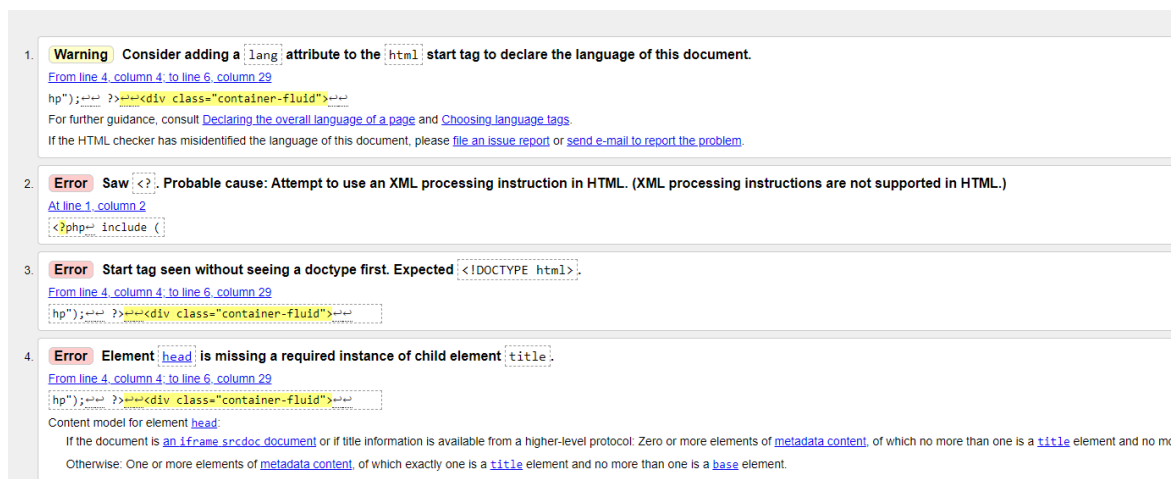


Рисунок 3.11 – Результати перевірки HTML за допомогою W3C Validator

Давайте розглянемо три види тестування сайтів:

- Ручне тестування, відоме також як ad-hoc-тестування. Результатом цього типу тестування є створення технічного звіту - Bug report;
- функціональне тестування;
- автоматизоване тестування.

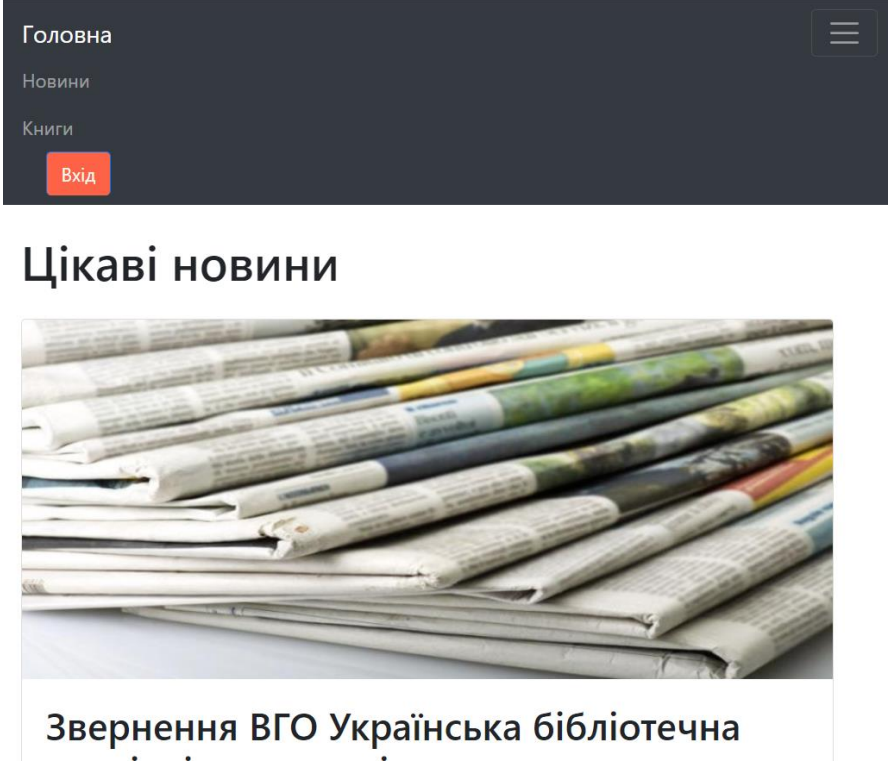
У таблиці 3.2 подано організацію звіту про помилки.

Таблиця 3.2 – Структура Bug-report

Сайт бібліотеки	
Короткий опис (Summary)	В мобільній версії сайту пункти випадаючого меню не коректно відображаються
Проект (Project)	Сайт Сільської бібліотеки
Компонент сайту (Component)	Додаткове меню



## Продовження таблиці 3.2 – Структура Bug-report

Номер версії (Version)	Версія v 1.0.1
Серйозність (Severity)	S3 Значний (Major)
Пріоритет (Priority)	P2 (Medium)
Автор (Author)	Бесащук Микола
(Assigned To)	Developer – Бесащук Микола
Середовище	
ОС / Сервіс Пак і т.д. / Браузер + версія / ...	Windows 11, Chrome 125.0.6422.142 (64 bit)
Опис	
Кроки відтворення (Steps to Reproduce)	1. Виконання веб-сайту на мобільному пристрої. 2. Натиснути на пункт з випадючими пунктами
Фактичний результат (Result)	Пункти випадючого меню відображаються не коректно
Очікуваний результат (Expected Result)	Пункти випадючого меню мають задній фон та відцентровані
Додатково	
Прикріплений файл (Attachment)	 <p>Головна Новини Книги Вхід</p> <p><b>Цікаві новини</b></p> <p>Звернення ВГО Українська бібліотечна</p>

Проблему було вирішено за допомогою додавання фрагменту коду до основного CSS файлу. Код поданий у лістингу 3.1.

Лістинг 3.1 – Частина коду, яка допомогла виправити помилку.

```
@media screen and (max-width: 767px){
.dropdown-menu, .show{
display: block;
padding: 15px 0;
text-align: center; width: 100%;}}
```

Далі ми проведемо тести сайту було протестовано з використанням функціонального тестування. Серед функціональних вимог враховано:

- Перевірка функціональної готовності: перевірка виконання всіх функцій, описаних у технічному завданні.
- Перевірка вірності: оцінка точності результатів, наданих сайтом
- Оцінка можливості взаємодії: перевірка спроможності користувачів взаємодіяти з сайтом та його елементами.
- Перевірка відповідності стандартам і правилам: оцінка того, наскільки розроблений сайт відповідає встановленим стандартам і правилам.
- Захищеність: перевірка рівня захищеності сайту від незаконного доступу та атак.

Під час тестування веб-сайту ми розглянемо всі функції, методи та компоненти, застосовані для переконання у їх відповідності вимогам, які реалізовані в системі. Результати функціонального тестування системи подані в таблиці 3.3.

Таблиця 3.3 – Створення набору test-case

ID	Тип	Опис	Очікувано	Реально	Pass/ Fail
1	2	3	4	5	6
01	positive	Тестування сумісності веб-сайту з різними браузерами.	Сайт має виглядати однаково у всіх доступних браузерах.	Відображення сайту має бути однаковим у різних браузерах.	Pass

Продовження таблиці 3.3 – Створення набору test-case

1	2	3	4	5	6
02	Positive	Перевірка правильності посилань у панелі навігації.	При кліці на посилання відбувається перехід на відповідну сторінку.	При натисканні на посилання користувач переходить на очікувану сторінку.	Pass
03	positive	Провірка проходження авторизації.	При авторизації користувача вертає на головну сторінку з можливістю входу в особистий кабінет	Авторизація працює правильно	Pass
04	positive	Тестування адаптивного дизайну.	Дизайн пристосовується до різних розмірів екранів.	Макет адаптується під різні розширення екранів.	Pass
05	positive	Перевірка кнопки перегляду книг	Перегляд книг по кнопці яка появляється після авторизації	Кнопка реагує правильно	Pass

Під час розробки сайту рекомендується використовувати автоматичне тестування разом з ручним, оскільки не завжди можна виявити всі проблеми та баги вручну. Для автоматизації тестування веб-сайту будемо використовувати Selenium IDE.

Selenium - це інструмент для автоматизації тестування веб-сайтів, що дозволяє автоматизувати стандартні дії, виконуючи їх через браузер. Ви можете встановити його як розширення браузера або завантажити з офіційного сайту SeleniumHQ.

Перед початком тестування потрібно увімкнути журналювання всіх тестів, які будуть виконуватися. Після цього переходимо на веб-сайт і виконуємо необхідні дії з функціоналом сторінок. Selenium IDE автоматично перевіряє валідність посилань та інші функціональні вимоги веб-сайту.

На рисунку 3.12 показано приклад автоматизованого тестування в Selenium IDE.

The screenshot shows the Selenium IDE interface. At the top, it says 'Project: Сільська бібліотека\*'. Below that, there's a toolbar with 'Run current test Ctrl+R'. A table lists 16 commands with their targets and values. A status bar at the bottom indicates 'Runs: 1 Failures: 0' and 'qwe2\* completed successfully'.

Command	Target	Value
6 mouse out	css= btn:nth-child(2)	
7 click	id=login	
8 type	id=login	qwerty
9 click	id=pass	
10 type	id=pass	q1w2e3r4
11 click	css= btn-success	
12 click	linkText=Книги	
13 click	linkText=Завантажити	
14 click	linkText=Новини	
15 click	linkText=Читати більше →	
16 click	linkText=Головна	

Log Reference

- 11. click on css= btn-success OK
- 12. click on linkText=Книги OK
- 13. click on linkText=Завантажити OK
- 14. click on linkText=Новини OK
- 15. click on linkText=Читати більше → OK
- 16. click on linkText=Головна OK

'qwe2\*' completed successfully

Рисунок 3.12 – Тестування автоматизовано за допомогою Selenium IDE.

Після виконання декількох різних тестів, Selenium IDE не виявив жодних помилок. Під час створення сайтів програмісти нерідко допускають помилки, які в майбутньому можуть спричинити серйозні проблеми в роботі сайту. Це можуть бути як технічні помилки в коді, так і логічні недоліки у функціонуванні певних елементів сайту. Тому тестування сайту є невід’ємною частиною введення його в експлуатацію [10] .

Як показали тести, майже весь програмний код написаний без помилок, а функції працюють правильно. Було виявлено декілька некритичних помилок, які були одразу виправлені. Проте, враховуючи складність сучасних веб-додатків і постійну взаємодію з користувачами, лишається імовірність, що на сайті залишились помилки, які будуть виявлені в подальшій експлуатації сайту.

## 4 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 4.1 Долікарська допомога при харчових отруєннях

Харчові отруєння - це поширена причина гострих захворювань, що виникають внаслідок вживання низькоякісних або забруднених харчових продуктів. Цей стан потребує негайного реагування для запобігання серйозним ускладненням. Медична допомога на початкових етапах отруєння є критично важливою, оскільки вона допомагає зменшити всмоктування токсинів, підтримати життєво важливі функції організму та запобігти ускладненням [10]. Основні заходи медичної допомоги при харчових отруєннях включають:

1. Визначення симптомів: Загальні симптоми харчового отруєння включають нудоту, блювоту, діарею, болі в животі, підвищення температури тіла, головний біль та загальну слабкість. У деяких випадках можуть спостерігатися також запаморочення, озноб та втрата свідомості [10].

2. Припинення вживання їжі: Перш за все, необхідно негайно припинити вживання їжі або напоїв, які могли спричинити отруєння.

3. Промивання шлунку: Якщо симптоми отруєння з'явилися протягом 2-3 годин після вживання їжі, можна провести промивання шлунку. Для цього необхідно випити велику кількість теплої води (1-2 літри) з додаванням слабого розчину соди (1 чайна ложка на літр води) та викликати блювання. Цей процес можна повторити кілька разів [10].

4. Прийом сорбентів: Для зменшення всмоктування токсинів слід прийняти сорбенти, такі як активоване вугілля або інші адсорбенти (наприклад, смекта, ентеросгель). Дозування активованого вугілля: 1 таблетка на 10 кг маси тіла.

5. Гідратація: Важливо забезпечити достатню гідратацію організму, щоб компенсувати втрату рідини внаслідок блювання та діареї. Рекомендується пити невеликими порціями, але часто, використовувати розчини для пероральної регідратації (наприклад, Регідрон), які містять електроліти [10].

6. Забезпечення спокою: Постраждалого необхідно укласти в ліжку, забезпечити спокій і тепло. Рекомендується уникати фізичного навантаження до повного одужання.

7. Контроль симптомів: Потрібно уважно стежити за станом постраждалого, звертати увагу на будь-які зміни симптомів. Особливо важливо контролювати частоту серцевих скорочень, дихання та рівень свідомості.

8. Звернення за медичною допомогою: При появі серйозних симптомів, таких як висока температура, нестримне блювання, діарея з домішками крові, значне зневоднення або порушення свідомості, необхідно негайно звернутися за медичною допомогою [10].

Медична допомога при харчових отруєннях є важливим етапом надання допомоги постраждалим, що може значно зменшити ризик ускладнень та прискорити процес одужання. Вчасно надані заходи допомоги допомагають стабілізувати стан постраждалого та підготувати його до подальшого лікування під наглядом медичних фахівців.

#### 4.2 Перша допомога людині, яка уражена електричним струмом

Ураження електричним струмом може бути дуже небезпечним і навіть смертельним, тому перша допомога має бути надана негайно і грамотно. Основні принципи надання першої допомоги при ураженні електричним струмом включають [11]:

1. Оцінка безпеки: забезпечте безпеку для себе, перш ніж підійти до постраждалого, переконайтеся, що джерело електричного струму вимкнено або відключено від живлення. Якщо це неможливо, використовуйте ізольовані предмети для відсунення джерела струму від постраждалого. Уникайте контакту з постраждалим, якщо він все ще перебуває під дією струму, не торкайтесь його руками або будь-якою частиною тіла без належного захисту [11].

2. Відключення електроживлення: вимкніть електроживлення, витягнувши вилку з розетки або вимкнувши автоматичний вимикач. Якщо це неможливо, використовуйте ізольований предмет, щоб відсунути постраждалого від джерела струму.

3. Перевірка стану постраждалого: перевірте, чи постраждалий свідомий і дихає. Якщо постраждалий непритомний, але дихає, переведіть його в стабільне бокове положення. Серцево-легенева реанімація, якщо постраждалий не дихає або у нього немає пульсу, негайно розпочніть серцево-легеневу реанімацію. Використовуйте комбінацію компресій грудної клітини та штучного дихання (30 компресій і 2 вдихи).

4. Надання першої допомоги: ураження електричним струмом часто супроводжується опіками в місцях входу та виходу струму. Накрийте опіки стерильними пов'язками або чистою тканиною. Якщо є кровотеча, зупиніть її, наклавши чисту тканину або стерильний бинт та натискаючи на рану [11].

5. Виклик швидкої допомоги: негайно зателефонуйте до швидкої допомоги, повідомивши про ураження електричним струмом та описавши стан постраждалого.

6. Подальші дії: продовжуйте спостерігати за станом постраждалого до прибуття медиків. Якщо свідомість постраждалого відновила, заспокойте його та не дозволяйте рухатись, оскільки може бути пошкодження внутрішніх органів або хребта. Якщо стан постраждалого погіршується (втрата свідомості, припинення дихання), продовжуйте надавати серцево-легеневу реанімацію до прибуття швидкої допомоги або до відновлення життєвих функцій.

Важливі моменти:

- Не намагайтеся гасити електричну пожежу водою або торкатися постраждалого мокрими руками, оскільки вода проводить електрику.

- Використовуйте гумові або інші ізольовані рукавиці та взуття при наданні допомоги в умовах можливої небезпеки електричного ураження.

- Постраждалий може перебувати в стані шоку. Підтримуйте його морально, заспокоюючи і запевняючи, що допомога вже на шляху.

Дотримання цих рекомендацій допоможе знизити ризик серйозних ускладнень та забезпечити безпеку як постраждалого, так і тих, хто надає першу допомогу [11].

У разі отримання ураження електричним струмом, важливо діяти оперативно та розсудливо, дотримуючись вищезазначених кроків. Правильно надана перша допомога може врятувати життя та запобігти серйозним ускладненням. Необхідно також пам'ятати про власну безпеку і уникати будь-яких дій, які можуть загострити ситуацію [11].

Після надання першої допомоги постраждалому від ураження електричним струмом важливо звернутися до медичних фахівців для подальшого лікування та обстеження. Можливість серйозних ускладнень, таких як травми внутрішніх органів або довготривалі наслідки від дії електричного струму, підкреслює важливість професійної медичної допомоги. Тому, після надання першої допомоги, необхідно негайно викликати швидку медичну допомогу і передати постраждалого у руки кваліфікованих медичних працівників.

Під час очікування прибуття швидкої допомоги, важливо продовжувати спостерігати за станом постраждалого і надавати необхідну допомогу з урахуванням його стану. Дотримання рекомендацій щодо надання першої допомоги та швидка реакція можуть значно полегшити перебіг ураження електричним струмом та покращити шанси на повне одужання постраждалого [11].



## ВИСНОВОК

Розвиток суспільства впливає на еволюцію бібліотечної справи, зокрема на сільські бібліотеки, які зазнають змін у відвідуванні молоддю. Сучасна молодь віддає перевагу цифровим ресурсам та інформації, доступ до яких можна отримати через інтернет і смартфони, замість фізичного відвідування бібліотеки. Такий підхід робить знання більш доступними та швидкодійними, але зменшує активність відвідування традиційних бібліотек.

Створення сайту сільської бібліотеки має на меті залучення мешканців до культурного збагачення. Розроблений веб-сайт має надавати доступ до фонду бібліотеки, переліку книг та їхніх жанрів, а також забезпечувати можливість читати книги у електронному вигляді. Важливою є зручність використання сайту для отримання інформації про нові надходження та рекомендації від бібліотекарів. Такий веб-ресурс стає важливим інструментом у зближенні між бібліотекою та спільнотою, сприяючи культурному розвитку регіону.

У процесі розробки веб-сайту будуть використовуватись сучасні технології, такі як HTML5, CSS3, PHP7, JavaScript, MySQL і phpMyAdmin. Це дозволить створити ефективний та надійний веб-ресурс, який забезпечить стабільну роботу та швидке завантаження сторінок. Використання таких інструментів дозволяє створити інтуїтивно зрозумілий та ефективний інтерфейс для користувачів, сприяючи зручній навігації та доступності інформації.

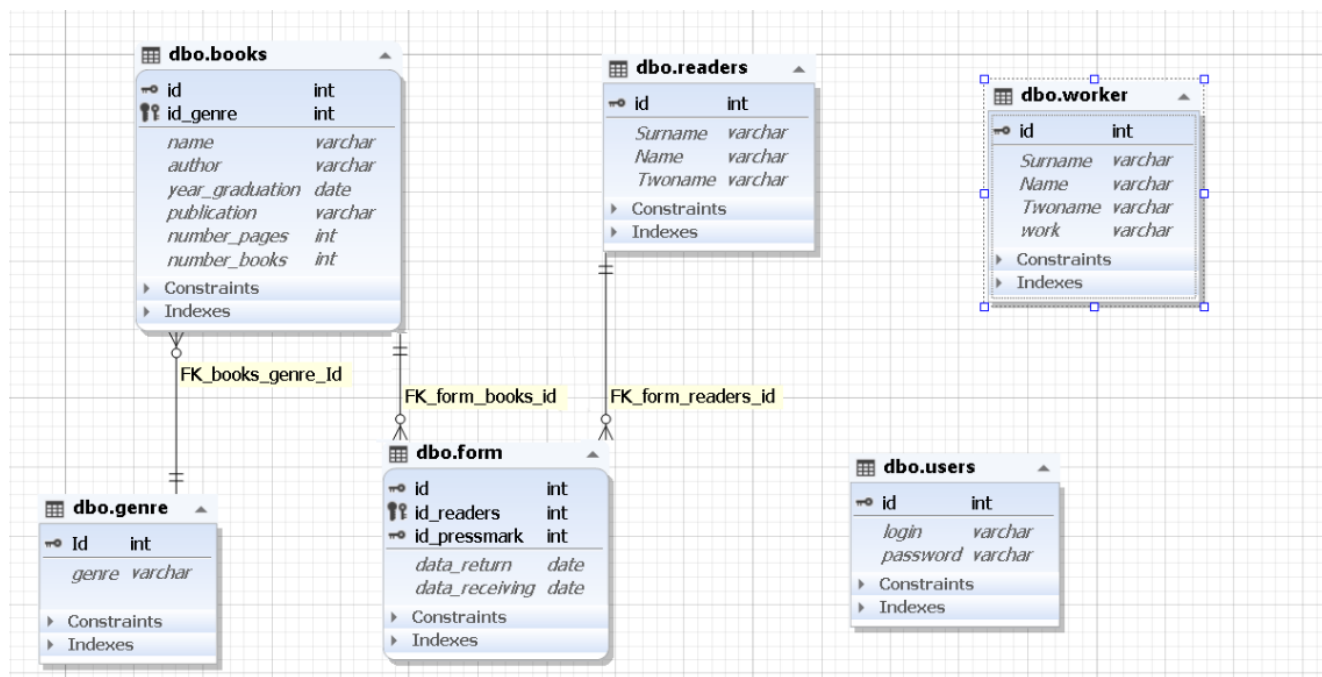
Впровадження цифрових рішень у сільські бібліотеки сприятиме їхньому розвитку та зростанню інтересу до культурного життя селища. Це дозволить бібліотекам стати сучасними інформаційними центрами, що відповідають сучасним вимогам та потребам громади. Автоматизація та цифрові рішення спростять роботу бібліотекарів та забезпечать максимальну зручність для відвідувачів.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Сайт «Національна бібліотека України імені В. І. Вернадського» URL: <http://www.nbuv.gov.ua/> (дата звернення 03.05.2024).
2. Сайт бібліотеки «Тернопільська обласна універсальна наукова бібліотека» URL: <https://library.te.ua/> (дата звернення 04.05.2024).
3. Сайт бібліотеки «Державна бібліотека України для юнацтва» URL: <http://4uth.gov.ua/> (дата звернення 04.05.2024).
4. Шаров С. В., Осадчий В. В. Бази даних та інформаційні системи : навч. посіб. Мелітополь : МДПУ ім. Б. Хмельницького, 2014. 352 с.
5. UML – діаграма прецедентів. URL: <https://www.quality-assurance-group.com/use-case-diagrams/> (дата звернення 10.05.2024).
6. UML – діаграма. Види діаграм UML. URL: <https://socscan.ru/uk/kofevarki/uml-diagramma-vidy-diagramm-uml-modelirovanie-na-uml-obshchie.html> (дата звернення 11.05.2024).
7. Діаграма послідовності. URL: <http://flash.retejo.info/cxefragho/uml/diagramma-poslidovnosti> (дата звернення 14.05.2024).
8. Основи UML-проекткування розподілених систем. URL: <http://moodle.ipk.kpi.ua/moodle/mod/resource/view.php?inpopup=true&id=44416> (дата звернення 15.05.2024).
9. PhpMyAdmin URL: <https://uk.wikipedia.org/wiki/PhpMyAdmin> (дата звернення 29.11.2023).
10. Mobile-Friendly Test. URL: <https://search.google.com/test/mobile-friendly> (дата звернення 29.11.2023).
11. Севальнев І. А. Навчальний посібник до практичних занять / І. А. Севальнев, М. П. Гребняк, Р. А. Федорченко. – Запоріжжя, 2020. – 106 с. – (ЗДМУ).
12. Зацарний В. В. БЖД навчальний посібник / В. В. Зацарний, О. В. Зацарна, О. В. Землянська. – Київ, 2016. – 230 с. – (КПІ).

## ДОДАТКИ

## ДОДАТОК А – ER-діаграма бази даних «biblioteka»



## Лістинг коду 1 – Код відображення графіків

```

<?php
  include ("include/header.php");
  ?>
<div class="container-fluid">
  <div class="row">
    <!-- Blog Entries Column -->
    <div class="col-md-2"></div>
    <div class="col-md-8">
<h2 class="my-4">Про бібліотеку</h2>
<div class="card mb-6">
  
  <div class="card-body">
    <p class="card-text">Навесні 1907 року в
с.Вільхівчик почала діяти «Просвіта», при якій була заснована читальня -
прообраз сучасної бібліотеки. Засновником читальні був Туткалюк Пилип та 19
його односельчан, які на той час зібрали 15 корон на придбання книжок і
виписку періодики для читальні. Через деякий час читальня припинила свою
діяльність. Її робота знову відновилася у 1925 році. В 1926році розпочалося
будівництво окремого будинку для читальні, яке було завершене в 1929 році.
Його вартість була оцінена в 1500 злотих і він був заінтибульований на
членів читальні «Просвіти» у Вільхівчику. Крім читальні, у цьому будинку
розмістилася українська споживча кооператива «Поміч», яка платила читальні
48 злотих щорічно. В 1938 році читальня мала 125 членів. Членський внесок
становив: для старших - 60 грошів, для молодших - 30 грошів. Читальня
проводила велику культурно освітню роботу. В неділю, свята та по вечорах у
будні дні тут відбувалися голосні читання газет та часописів. Читальня вела
постійну роботу по поповненню своїх фондів. В 1938 році фонд становив 332
книги, в цьому ж році літературою скористався 131 чоловік, прочитано 407
книг. В 1940 році в селі було створено колгосп і бібліотека була заснована
при колгоспі. В 1952 році бібліотекарем призначена Скотніцька - Деренівська
Олександра Мартинівна яка працювала до 1984 року, вийшовши на пенсію.</p>
  </div>
</div>
</div>
<script>
  let map;
  function initMap() {
    myMap = new google.maps.Map(document.getElementById("map"), {
      center: { lat: 49.096232, lng: 26.196987 },
      zoom: 17,
    });
    var marker = new google.maps.Marker(
    {
      position: { lat: 49.096232, lng: 26.196987 },
      map: myMap,
      title: "Ми тут"
    });
  }
</script>
<h2 class="my-4">Ми тут</h2>
  <div id="map" style="height: 400px; width: 100%;">
  </div>

```

```

        <script
            src="https://maps.googleapis.com/maps/api/js?key=AIzaSyABFL41mVzMWit9
n0g_0c2cuLGgKD_FyZk&callback=initMap&libraries=&v=weekly"
            async
        ></script>
    </div>
</div>
<br><br>

<?php
    include ("include/footer.php");
?>

```

## Лістинг коду 2 – Код лістингу колонки з новинами

```

<?php
include("include/header.php");
$post_id = $_GET['post_id'];
if (!is_numeric($post_id)) exit();
$post=get_post_by_id($post_id);
?>
<div class="container">
    <div class="row">
        <!-- Post Content Column -->
        <div class="col-lg-16">
            <!-- Title -->
            <h1 class="mt-4"><?=$post['title']?></h1>

            <hr>
            <!-- Date/Time -->
            <p>Опубліковано <?=$post['datetime']?></p>
            <hr>
            <!-- Preview Image -->
            

            <hr>
            <!-- Post Content -->
            <p><?=$post['content']?></p>
            <hr>
        </div>
    </div>
<?php
include("include/footer.php");
?>

```

## Лістинг коду 2 – Код лістингу списку книг

```

<?php
include("include/header.php");
$category_id = $_GET['id'];
if (!is_numeric($category_id)) exit();

$category=get_category_by_id($category_id);
if($category_id==2):
?>
<div class="container-fluid">

    <div class="row">
        <!-- Post Content Column -->
    <!-- Blog Entries Column -->
        <div class="col-md-2"></div>
        <div class="col-md-8">
            <h1 class="my-4">НОВИНИ</h1>
            <?php
                $news = get_news();
            ?>
            <!-- Blog Post -->
            <?php foreach ($news as $new):?>
                <div class="card mb-4">
                    
                    <div class="card-body">
                        <h2 class="card-title"><?=$new['title']?></h2>
                        <p class="card-text"><?=mb_substr($new['content'],0,
360, 'UTF-8').'...'?></p>
                        <a href="post.php?post_id=<?=$new["id"]?>" class=" btn
btn-primary">Читати більше &rarr;</a>
                    </div>
                    <div class="card-footer text-muted">
                        <svg class="bi bi-calendar" width="1em" height="1em"
viewBox="0 0 16 16" fill="currentColor" xmlns="http://www.w3.org/2000/svg">
                            <path fill-rule="evenodd" d="M14 0H2a2 2 0 0-2
2v12a2 2 0 02 2h12a2 2 0 02-2V2a2 2 0 0-2-2zM1 3.857C1 3.384 1.448 3 2
3h12c.552 0 1 .384 1 .857v10.286c0 .473-.448.857-1 .857H2c-.552 0-1-.384-1-
.857V3.857z" clip-rule="evenodd"/>
                            <path fill-rule="evenodd" d="M6.5 7a1 1 0 100-2 1 1
0 000 2zm3 0a1 1 0 100-2 1 1 0 000 2zm3 0a1 1 0 100-2 1 1 0 000 2zm-9 3a1 1
0 100-2 1 1 0 000 2zm3 0a1 1 0 100-2 1 1 0 000 2zm3 0a1 1 0 100-2 1 1 0 000
2zm3 0a1 1 0 100-2 1 1 0 000 2zm-9 3a1 1 0 100-2 1 1 0 000 2zm3 0a1 1 0
100-2 1 1 0 000 2zm3 0a1 1 0 100-2 1 1 0 000 2z" clip-rule="evenodd"/>
                        </svg>
                        Оpubліковано <?=$new['datetime']?>
                    </div>
                </div>
            <?php endforeach; ?>
        <?php endif; ?>
<?php

$category_id = $_GET['id'];
if (!is_numeric($category_id)) exit();

```

```

$category=get_category_by_id($category_id);
if($category_id==3):
?>
<div class="container-fluid">
  <div class="row">
    <div class="col-md-2"></div>
    <div class="col-md-8">
      <div class="row">
        <div class="col-md-3"><h1 class="my-3">Книги</h1></div>
        <div class="col-md-3"></div>
        <div class="col-md-6" style="margin-top: 5px;">
          <div class="my-4"><div><form method="POST"
action="search.php"> <input type="text" name="text" placeholder="Пошук"
value= "<?=$_COOKIE['name']?>" required> <input type="submit" name="enter"
value="Пошук"></form></div>

          </div></div></div>

      <?php
        $books = get_books();
      ?>
      <div class="card mb-4">
        <?php foreach ($books as $book):?>
          <div class="row">
            <div class="card mb-1" style="margin-left: 15px"></div>

            <div class="card-body" style="width: 40%;">
              <h4 class="card-title"><?=$book['name']?></h4>
              <p class="card-text">Автор: <?=$book['avtor']?></p>
              <p class="card-text">Рік випуску:
<?=$book['rik_vypusku']?></p>
              <p class="card-text">Видавництво:
<?=$book['publication']?></p>
              <p class="card-text">Кількість сторінок:
<?=$book['kil_stor']?></p>
              <p class="card-text">Кількість книг:
<?=$book['kil_books']?></p>
              <?php
                if(empty($_COOKIE['user'])):
                  ?>
                  <p class="card-text">Авторизуйтеся щоб читати</p>
                <?php else:?>
                  <a href="<?=$book["text"]?>" download="" class=" btn
btn-primary">Завантажити</a>
                </div>
              <?php endif;?>
            </div>

            <hr>
          <?php endforeach; ?>
        <?php endif; ?>
      </div>

      <?php
        include ("include/footer.php");
      ?>

```



## Лістинг коду 3 – Код лістингу сторінки адміністратора

```

    <!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Адмін Панель</title>
</head>
<body>

    <?php
        if(empty($_COOKIE['admin'])):
            ?>
            <?php for($n=1; $n<1000;$n++){?>
                ти не адмін
            <? }?>
            <?php else:??>
<h1>Адмін Панель</h1>
<a href="index.php">На головну сторінку</a>
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$dbname = "biblioteka";

$conn = mysqli_connect($servername, $username, $password, $dbname);
if (mysqli_connect_errno()) {
    echo 'Помилка підключення до БД:
(' . mysqli_connect_errno() . ') : ' . mysqli_connect_error();
    exit();
}
?>

<?php

include ("del/del_regist.php");
include ("del/del_news.php");
include ("del/del_books.php");
include ("del/del_categorias.php");
include ("del/del_janr.php");
include ("del/del_coment.php");

if (isset($_POST["title"])) {
    //Если это запрос на обновление, то обновляем
    if (isset($_GET['red_id'])) {
        $sql = mysqli_query($conn, "UPDATE `news` SET `title` =
'{"$_POST['title']}' , `image` = '{"$_POST['image']}'
        , `content` = '{"$_POST['content']}' , `datetime` =
'{"$_POST['datetime']}' WHERE `id`={"$_GET['red_id']}");
    } else {
        //Иначе вставляем данные, подставляя их в запрос
        $sql = mysqli_query($conn, "INSERT INTO `news` (`title`,

```

```

`image`,`content`,`datetime`) VALUES ('{$_POST['title']}',
      '{$_POST['image']}' , '{$_POST['content']}',
      '{$_POST['datetime']}'");
    }

    //Если вставка прошла успешно
    if ($sql) {
        echo '<p>Успешно!</p>';
    } else {
        echo '<p>Произошла ошибка: ' . mysqli_error($conn) . '</p>';
    }
}

if (isset($_GET['red_id'])) {
    $sql = mysqli_query($conn, "SELECT `id`,`title`,`image`,`content`,`datetime` FROM `news` WHERE `id`={$_GET['red_id']}");
    $news = mysqli_fetch_array($sql);
}

if (isset($_GET['del_news_id'])) {

    $sql = mysqli_query($conn, "DELETE FROM `news` WHERE `id` =
    {$_GET['del_news_id']}");
    if ($sql) {
        echo "<p>Користувач видалений.</p>";
    } else {
        echo '<p>Сталася помилка: ' . mysqli_error($conn) . '</p>';
    }
}

if (isset($_GET['del_news_id'])) {
    $sql = mysqli_query($conn, "DELETE FROM `news` WHERE `id` =
    {$_GET['del_news_id']}");
    if ($sql) {
        echo "<p>Користувач видалений.</p>";
    } else {
        echo '<p>Сталася помилка: ' . mysqli_error($conn) . '</p>';
    }
}

?>
<h2>Таблиця користувачів</h2>
<table border='1'>
<tr>
<td>Номер</td>
<td>Логін</td>
<td>Пароль</td>
<td>Вид користувача</td>
</tr>
<?php
    $sql = mysqli_query($conn, 'SELECT `id`,`login`,`pass`,`name`,`admin`
FROM `regist`');
    while ($result = mysqli_fetch_array($sql)) {

        echo

            "<tbody>".

```

```

        '<tr>' .
            "<td>{$result['id']}</td>" .
            "<td>{$result['login']}</td>".
            "<td>{$result['pass']}</td>" .
            "<td>{$result['name']}</td>" .
            "<td>{$result['admin']}</td>".
            "<td><a
href='?del_regist_id={$result['id']}'>Видалити</a></td>".
            '</tr>'.
            "</tbody>";

    }

    ?>
</table>

<h3>Форма додавання</h3>
<form action="add/add_regist.php" method="post">
    <table>
        <tr>
            <td>Логін:</td>
            <td><input type="text" name="login" id="title"></td>
        </tr>
        <tr>
            <td>Пароль
            <td><input type="text" name="pass" id="image"> </td>
        </tr>
        <tr>
            <td>Імя
            <td><input type="text" name="name" id="name"> </td>
        </tr>
        <tr>
            <td>Адмін
            <td><input type="text" name="admin" id="admin"> </td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="OK"></td>
        </tr>
    </table>
</form>

<h2>Таблиця новин</h2>
    <table border='1'>
        <tr>
            <td>Номер</td>
            <td>Заголовок</td>
            <td>Зображення</td>
            <td>Зміст</td>
            <td>Дата</td>
        </tr>
        <?php
            $sql = mysqli_query($conn, 'SELECT `id`, `title`,
`image`, `content`, `datetime` FROM `news`');
            while ($result = mysqli_fetch_array($sql)) {

                echo
                "<tbody>".

```

```

    '<tr>' .
        "<th scope='row'>{$result['id']}</th>" .
        "<td>{$result['title']}</td>" .
        "<td>{$result['image']}</td>" .
        "<td>{$result['content']}</td>" .
        "<td>{$result['datetime']}</td>" .
        "<td><a href='?del_news_id={$result['id']}'>Видалити</a></td>"
    .
        "<td><a href='?red_id={$result['id']}'>Редагувати</a></td>" .
    '</tr>'.
    "</tbody>";
}

?>
</table>
<h3>Форма змін</h3>
<form action="" method="post">
    <table>
        <tr>
            <td>Заголовок:</td>
            <td><input type="text" name="title" value="<?=
isset($_GET['red_id']) ? $news['title'] : ''; ?>"></td>
        </tr>
        <tr>
            <td>Зображення:</td>
            <td><input type="text" name="image" value="<?=
isset($_GET['red_id']) ? $news['image'] : ''; ?>"></td>
        </tr>
        <tr>
            <td>Зміст:</td>
            <td><input type="text" name="content" value="<?=
isset($_GET['red_id']) ? $news['content'] : ''; ?>"></td>
        </tr>
        <tr>
            <td>Дата:</td>
            <td><input type="text" name="datetime" value="<?=
isset($_GET['red_id']) ? $news['datetime'] : ''; ?>"></td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="OK"></td>
        </tr>
    </table>
</form>

<h3>Форма додавання</h3>
<form action="add/add_news.php" method="post">
    <table>
        <tr>
            <td>Заголовок:</td>
            <td><input type="text" name="title" id="title"></td>
        </tr>
        <tr>
            <td>Зображення
            <td><input type="text" name="image" id="image"> </td>
        </tr>
        <tr>
            <td>Зміст
            <td><input type="text" name="content" id="content"> </td>

```

```

    </tr>
    <tr>
        <td>Дата
        <td><input type="text" name="datetime" id="datetime"> </td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="OK"></td>
    </tr>
</table>
</form>

<h2>Таблиця категорій</h2>
    <table border='1'>
    <tr>
        <td>Номер</td>
        <td>Назва</td>

    </tr>
    <?php
        $sql = mysqli_query($conn, 'SELECT `id`, `title` FROM `categories`');
        while ($result = mysqli_fetch_array($sql)) {

            echo "<tbody>".
                '<tr>' .
                    "<th scope='row'>{$result['id']}</th>" .
                    "<td>{$result['title']}</td>" .
                    "<td><a
href='?del_category_id={$result['id']}'>Видалити</a></td>" .

                '</tr>'.
                "</tbody>";

        }

    ?>
</table>

<h3>Форма додавання</h3>
<form action="add/add_categories.php" method="post">
    <table>
        <tr>
            <td>Назва:</td>
            <td><input type="text" name="title" id="title"></td>
        </tr>
        <tr>
            <td colspan="2"><input type="submit" value="OK"></td>
        </tr>
    </table>
</form>

<h2>Таблиця книг</h2>
    <table border='1'>
    <tr>
        <td>Номер</td>
        <td>Назва</td>
        <td>Автор</td>

```

```

        <td>Рік випуску</td>
        <td>Видавництво</td>
        <td>Кількість сторінок</td>
        <td>Кількість книг</td>
        <td>Номер жанру</td>
        <td>Зображення</td>
        <td>Текст</td>
    </tr>
<?php
    $sql = mysqli_query($conn, 'SELECT `id`, `name`,
`avtor`, `rik_vypusku`, `publication`, `kil_stor`, `kil_books`, `id_janr`, `img`,
`text` FROM `books`');
    while ($result = mysqli_fetch_array($sql)) {

        echo
        "<tbody>".
        ' <tr>' .
            "<th scope='row'>{$result['id']}</th>" .
            "<td>{$result['name']}</td>" .
            "<td>{$result['avtor']}</td>" .
            "<td>{$result['rik_vypusku']}</td>" .
            "<td>{$result['publication']}</td>" .
            "<td>{$result['kil_stor']}</td>" .
            "<td>{$result['kil_books']}</td>" .
            "<td>{$result['id_janr']}</td>" .
            "<td>{$result['img']}</td>" .
            "<td>{$result['text']}</td>" .
            "<td><a
href='?del_books_id={$result['id']}'>Видалити</a></td>" .

        ' </tr>'.
        "</tbody>";

    }

?>
</table>

<h3>Форма додавання</h3>
<form action="add/add_books.php" method="post">
    <table>
        <tr>
            <td>Назва:</td>
            <td><input type="text" name="name" id="name"></td>
        </tr>
        <tr>
            <td>Автор
            <td><input type="text" name="avtor" id="avtor"> </td>
        </tr>
        <tr>
            <td>Рік випуску
            <td><input type="text" name="rik_vypusku" id="rik_vypusku"> </td>
        </tr>
        <tr>
            <td>Видавництво
            <td><input type="text" name="publication" id="publication"> </td>

```

```

</tr>
<tr>
  <td>Кількість сторінок
  <td><input type="text" name="kil_stor" id="kil_stor"> </td>
</tr>
<tr>
  <td>Кількість книг
  <td><input type="text" name="kil_books" id="kil_books"> </td>
</tr>
<tr>
  <td>Номер жанру
  <td><input type="text" name="id_janr" id="id_janr"> </td>
</tr>
<tr>
  <td>Зображення
  <td><input type="text" name="img" id="img"> </td>
</tr>
<tr>
  <td>Текст
  <td><input type="text" name="text" id="text"> </td>
</tr>
<tr>
  <td colspan="2"><input type="submit" value="OK"></td>
</tr>
</table>
</form>

<h2>Таблиця жанрів</h2>
<table border='1'>
<tr>
  <td>Номер</td>
  <td>Назва</td>

</tr>
<?php
  $sql = mysqli_query($conn, 'SELECT `id`, `name` FROM `janr`');
  while ($result = mysqli_fetch_array($sql)) {

    echo
    "<tbody>".
      '<tr>' .
        "<th scope='row'>{$result['id']}</th>" .
        "<td>{$result['name']}</td>" .
        "<td><a href='?del_janr_id={$result['id']}'>Видалити</a></td>"
    .

      '</tr>'.
      "</tbody>";

  }

?>
</table>
<h3>Форма додавання</h3>
<form action="add/add_janr.php" method="post">
  <table>
  <tr>

```

```

        <td>Назва:</td>
        <td><input type="text" name="name" id="name"></td>
    </tr>
    <tr>
        <td colspan="2"><input type="submit" value="OK"></td>
    </tr>
</table>
</form>

<h2>Таблиця відгуків</h2>
    <table border='1'>
    <tr>
        <td>Номер</td>
        <td>Відгук</td>
        <td>Імя</td>

    </tr>
    <?php
        $sql = mysqli_query($conn, 'SELECT `id`, `text`, `name` FROM
`coment`');
        while ($result = mysqli_fetch_array($sql)) {

            echo "<tbody>".
                '<tr>' .
                    "<td>{$result['id']}</td>" .
                    "<td>{$result['text']}</td>".
                    "<td>{$result['name']}</td>" .
                    "<td><a
href='?del_coment_id={$result['id']}'>Видалити</a></td>".
                '</tr>'.
                "</tbody>";
        }

    ?>
</table>

<?php endif;?>

</body>
</html>

```



ДОДАТОК В – Диск із кваліфікаційною роботою бакалавра