

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ПРИРОДОКОРИСТУВАННЯ**  
**ФАКУЛЬТЕТ МЕХАНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ**  
**ТЕХНОЛОГІЙ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

# **КВАЛІФІКАЦІЙНА РОБОТА**

Першого (бакалаврського) рівня вищої освіти  
на тему «**Розробка веб-додатку автоматизованої бібліотечної  
інформаційної системи**»

Виконав студент 4 курсу групи Іт-41сп  
Спеціальності 126 «Інформаційні системи та технології»

Сембай Віталій Русланович

Керівник: к.т.н., в.о. доцента Падюка Р. І.

**ДУБЛЯНИ-2024**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ПРИРОДОКОРИСТУВАННЯ**  
**ФАКУЛЬТЕТ МЕХАТРОНІКИ, ЕНЕРГЕТИКИ ТА ІНФОРМАЦІЙНИХ**  
**ТЕХНОЛОГІЙ**  
**КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Перший (бакалаврський) рівень вищої освіти

Спеціальність 126 «Інформаційні системи та технології»

«ЗАТВЕРДЖУЮ»

Завідувач кафедри \_\_\_\_\_

д.т.н., проф. А. М. Тригуба

« \_\_\_\_ » \_\_\_\_\_ 202\_ р.

**ЗАВДАННЯ**

на кваліфікаційну роботу студенту

Сембаю Віталію Руслановичу

1. Тема роботи: «Розробка веб-додатку автоматизованої бібліотечної інформаційної системи»
2. Керівник роботи в.о. доцента, Падюка Роман Іванович  
затверджені наказом по університету від 27.11.2023 року № 641/к-с.
3. Строк подання студентом роботи 10.06.2024 р.
4. Вихідні дані до роботи: вимоги до автоматизованих бібліотечних систем; методика проектування інформаційних систем; технічне завдання на проектування бібліотечної інформаційної системи
5. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)  
Вступ
  1. Аналіз предметної області.
  2. Постановка задачі та планування проекту.
  3. Проектування автоматизованої бібліотечної інформаційної системи.
  4. Охорона праціВисновки та пропозиції  
Список використаної літератури.

6. Перелік ілюстраційного матеріалу (з точним зазначенням обов'язкових схем та моделей): огляд інтелектуальних інформаційних систем, аналіз мови програмування і середовище програмування, етапи створення автоматизованої бібліотечної інформаційної системи і пов'язані з ними технології, база даних автоматизованої бібліотечної інформаційної системи та загальних вигляд користувачького інтерфейсу системи.

7. Консультанти з розділів:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1, 2, 3,	<i>Падюка Р.І., в.о. доцента кафедри ІТ</i>		
4	<i>Городецький І.М., доцент кафедри фізики, інженерної механіки та безпеки виробництва</i>		

8. Дата видачі завдання

30 листопада 2023 р.

Календарний план

№ з/п	Назва етапів дипломного проекту	Терміни виконання етапів роботи	Примітка
1.	<i>Написання першого розділу</i>	<i>30.11.23-02.02.24</i>	
2.	<i>Виконання другого розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>03-28.02.24</i>	
3.	<i>Виконання третього розділу та аркушів ілюстраційного матеріалу до нього</i>	<i>01.03-30.04.24</i>	
4.	<i>Написання розділу «Охорона праці»</i>	<i>01-15.05.24</i>	
5.	<i>Завершення оформлення розрахунково-пояснювальної записки та аркушів ілюстраційного матеріалу</i>	<i>15-30.05.24</i>	
6.	<i>Завершення роботи цілому</i>	<i>01 -10.06.24</i>	

Студент \_\_\_\_\_ Сембай.В.Р.  
(підпис)

Керівник роботи \_\_\_\_\_ Падюка.Р.І.  
(підпис)

УДК: 004.9 : 631.1

Розробка веб-додатку автоматизована бібліотечна інформаційна система.

Сембай В.Р. Кафедра ІТ. - Дубляни, Львівський НУП, 2024.

Кваліфікаційна робота: 72 с. текст. част., 24 рис. 17 джерел.

Проведено аналіз автоматизованої інформаційної системи бібліотеки з метою визначення її ефективності в управлінні ресурсами та обслуговуванні користувачів. Пропоновані вдосконалення системи включають удосконалення інтерфейсу користувача та розширення його функціональних можливостей, таких як можливість переглядати та оцінювати книги, ділитися рекомендаціями та організовувати книжкові клуби.

Здійснено огляд інтелектуальних інформаційних систем, таких як Koha та Evergreen, які є важливими інструментами для автоматизованих бібліотечних інформаційних систем.

Описано особливості розробки веб-додатку автоматизованої бібліотечної інформаційної системи та вибрано засоби розробки серед яких мова програмування Python та середовище розробки PyCharm.

Розроблено IDEF0 діаграми декомпозиції процесу, такі як «Наповнення бази даних», «Перевірка обмежень автоматизованої бібліотечної інформаційної системи» та «Проектування автоматизованої бібліотечно-інформаційної системи» як основу для розробки розумних інформаційних систем для бібліотек.

Розроблено заходи щодо охорони праці

Ключові слова: веб-додаток, зберігання даних, Управління бібліотечними ресурсами, пошук книг, інформаційні системи, реєстрація користувачів, SQLite.

Key words: Web application, Data storage, Library resource management, Book search, Information systems, User registration, SQLite.

## ЗМІСТ

ВСТУП.....	6
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	7
1.1 Сучасний стан електронних бібліотек .....	7
1.2 Загальні відомості про автоматизовані бібліотечні інформаційні системи та їх функції .....	10
1.3. Огляд інтелектуальних інформаційних систем аналогів .....	14
2. ПОСТАНОВКА ЗАДАЧІ ТА ПЛАНУВАННЯ ПРОЕКТУ.....	21
2.1 Мета та задачі проекту .....	21
2.2 Вибір засобів реалізації проекту .....	22
2.3 Огляд можливостей середовища розробки .....	28
2.4 Особливості розробки бази даних .....	32
3. ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ БІБЛІОТЕЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	36
3.1 Структурно-функціональне моделювання автоматизованої бібліотечної інформаційної системи .....	36
3.2 Розробка бази даних для автоматизованої бібліотечної інформаційної системи.....	43
3.3 Розробка та особливості користувацького інтерфейсу системи ..	47
4. ОХОРОНА ПРАЦІ .....	54
4.1 Аналіз небезпек при роботі з комп'ютером .....	54
4.2 Освітлення та вентиляція робочої зони .....	56
4.3 Інструкція з охорони праці під час роботи за комп'ютером .....	58
ВИСНОВКИ ТА ПРОПОЗИЦІЇ .....	60
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	61
ДОДАТКИ .....	63

## ВСТУП

Інформаційні системи мають велике значення в усіх сферах життя людини, тому суспільство зацікавлене в їх розвитку та підвищенні ефективності.

Розвиток сучасних комп'ютерів дозволив автоматизувати обробку даних у всіх сферах людської діяльності. Промислові технології, управління економікою на всіх рівнях, наукові дослідження, освіта, транспорт, видавнича справа тощо неможливо уявити без систем обробки даних.

З появою персональних комп'ютерів і комп'ютерних мереж сфера застосування комп'ютерної техніки значно розширилася. Найбільш поширеною системою обробки даних є інформаційна система (ІС).

У сучасному суспільстві слово «інформація» є повсюдним. Ми надаємо йому дуже широкий діапазон значень і зазвичай можемо пояснити це на інтуїтивному рівні.

Інформація надходить через телефони, комп'ютерні мережі, радіо і телебачення, газети і журнали, зберігається в бібліотеках, архівах, базах даних тощо.

Метою проведеної сертифікаційної роботи є забезпечення точності, достовірності та повноти даних.

Це також включає контроль якості даних, забезпечення точності обробки інформації, мінімізацію помилок і зберігання даних у безпечному середовищі.

Проектована автоматизована бібліотечно-інформаційна система має забезпечувати каталогізацію літератури, періодичних видань, управління процесом видання та повернення літератури, можливість резервування та замовлення ресурсів.

Це включає пошук за назвою, автором, темою, класифікаційним номером, роком публікації, ключовими словами та іншими критеріями.

# 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Сучасний стан електронних бібліотек

Сучасна бібліотека – це не лише місце для зберігання книг, а й своєрідний електронний архів, місце, де можна отримати доступ до інформації як на традиційних носіях, так і в електронному форматі[10].

Вона спрямована на забезпечення рівного доступу до якісної інформації, яка може надати якісну інформацію в контексті розвитку процесів гуманізації, інформатизації, відкритості освіти та формування освітніх інформаційних потреб.

На сьогоднішній день загальновизнано, що найбільш ефективно інформаційно-бібліотечне обслуговування населення досягається шляхом створення електронних бібліотек. Електронна бібліотека складається з сукупності електронних документів і являє собою систему з єдиним підходом до створення, зберігання та організації різноманітної інформації. Останнім часом з'явилися різні тлумачення терміна «електронна бібліотека»[11].

Електронна бібліотека – це інформаційна система, яка забезпечує безпечно зберігання та ефективно використання різноманітної колекції електронних документів.

Основним завданням електронної бібліотеки є інтеграція інформаційних ресурсів та ефективна навігація в них.

Вона виконує такі функції: забезпечує збереження друкованої продукції з бібліотечних фондів; сприяє забезпеченню доступу локальних і віддалених користувачів до інформаційних ресурсів; розширювати інформаційні послуги; вирівнювання рівня технологічного розвитку різних організацій. В електронних бібліотеках необхідно передбачити можливість імпорту або видалення

інформаційних об'єктів, їх інтеграції, реструктуризації та інших подібних операцій.

Технічно електронні бібліотеки реалізовані на персональних комп'ютерах, підключених до локальної мережі та керованих мережевою операційною системою. Текст документа, що надійшов від видавництва, перевіряється операторами та коректорами та звіряється з друкованим виданням для подальшого використання читачами. Електронний текст перевірених документів заархівовано та включено до електронної бібліотеки. Окрім видавців, іншим джерелом отримання електронного тексту документів є комп'ютерні мережі.

Створення та використання електронних бібліотек здійснюється шляхом накопичення, зберігання, обліку та структурування електронної інформації; організації орієнтуються в інформаційному просторі, доступному через електронні бібліотеки даних; шляхом забезпечення ефективного доступу до користувачів через телекомунікаційні мережі, а також шляхом навчання користувачів.

Електронні бібліотеки мають такі переваги:

- надає інформацію користувачам на місці оформлення замовлення – потрібен лише комп'ютер, підключений до мережі Інтернет;
- надає більше можливостей для пошуку та обробки інформації, оскільки практично будь-яке слово в тексті може бути пошуковим виразом;
- пропонує можливість обміну певною інформацією, що значно спрощує завдання фізичного копіювання рідко використовуваних документів, а також надає спочатку доступ до окремого документа для завдання, яке попередньо має бути переміщено до репозиторію, де воно було розташоване;
- надає бібліотекам та архівам можливість забезпечити широкий доступ користувачів до своїх фондів шляхом їх розміщення в комп'ютерній мережі;



- дає бібліотекам можливість постійно оновлювати свою інтелектуальну власність, оскільки легше оновлювати електронні версії документів, ніж друковані;
- інформація доступна 24 години на добу і всюди;
- Інформаційні документи можна надсилати в різних формах (текст, схеми, аудіо дані, відео).

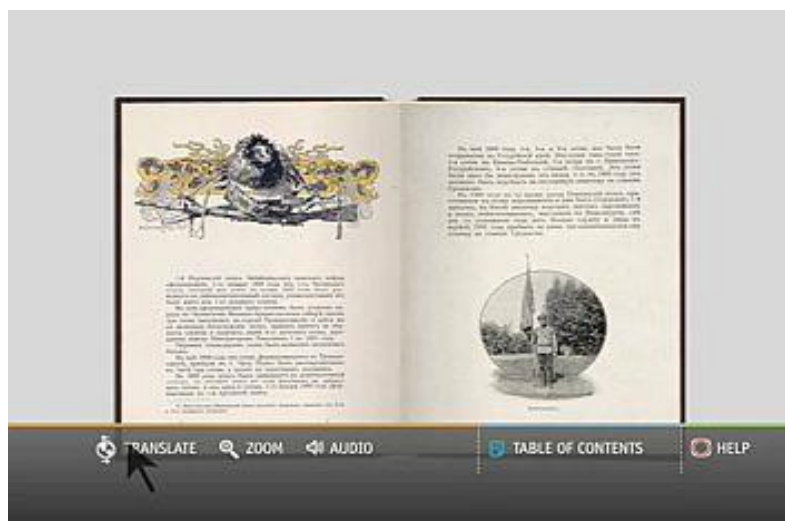


Рисунок 1.1. - Проект «Гутенберг» — онлайн-волонтерська ініціатива з оцифрування, архівування та поширення культурних творів.

Заснована в 1971 році Майклом Гарттом, це найстаріша універсальна електронна бібліотека. Більшість статей у цій колекції є повними текстами книг, які є відкритими (Рисунок 1.1.).

Цей проект намагається забезпечити максимально вільний і довгостроковий доступ до колекції, використовуючи відкриті формати даних, які можна завантажити на різні типи комп'ютерів. Станом на серпень 2010 року бібліотека включає понад 40 000 книг у вільному доступі, а також аудіо книги, музику тощо.

Електронні бібліотеки, як і традиційні бібліотеки, є універсальними та спеціалізованими. Однак якщо в традиційних бібліотеках спеціалізація

визначається лише відбором літератури, то в електронних бібліотеках вона визначається набором додаткових програмних засобів, що дозволяють здійснювати різні маніпуляції з об'єктами зберігання.

## **1.2 Загальні відомості про автоматизовані бібліотечні інформаційні системи та їх функції**

Автоматизована бібліотечна інформаційна система (АБІС) — це система планування бібліотечних ресурсів, яка використовується для відстеження бібліотечних колекцій від замовлення до придбання до розповсюдження відвідувачам бібліотеки. АБІС забезпечує обробку, аналіз та узагальнення обробки та представлення бібліотечних фондів документів (як традиційних (паперових), так і електронних інформаційних ресурсів), обслуговування бібліотечних фондів на всіх етапах від комплектування до замовлення[6].

Кожен читач (відвідувач) і екземпляр мають унікальний ідентифікатор у базі даних, що дозволяє АБІС відстежувати активність. Великі бібліотеки використовують АБІС для замовлення та придбання книг та інших колекцій, їх каталогізації, розповсюдження, резервування матеріалів та відстеження повернення. Менші бібліотеки часто використовують деякі з цих функцій. Усі великі бібліотеки світу використовують АБІС[6].

Головним елементом традиційної АБІС стає електронний каталог (ЕК) з усіма його функціями. Всі властивості традиційної бібліотеки при роботі в АБІС зберігаються, не сильно змінюючись, насамперед, зберігаються фонди в тому вигляді до якого всі звикли. В ЕК присутні посилання на повні тексти, малюнки, аудіо- і відео- матеріали, а також посилання на ресурси Інтернет[6].

Основним компонентом традиційної АБІС є система автоматизації бібліотек (САБ) . САБ надає користувачам найбільш зручний і відповідний доступ до фондів Бібліотеки з конкретної точки зору Бібліотеки. САБ відомий своїми можливостями поповнення, обслуговуванням , доставкою ЕС

користувачам, підтримкою бібліотечних форматів і стандартів , а також наданням мовних засобів для ЕС[6].

У традиційному використанні АБІС користувачі завжди були стурбовані інформаційною підтримкою, мовною підтримкою (друковані та електронні системи класифікації, авторитетні файли тощо)[6].

Система індексування, яка використовується при розробці електронної бібліотеки, суттєво відрізняється від тієї, яка використовується в традиційній АБІС. Наприклад, індексація повного тексту дозволяє знайти не тільки сам текст, але й потрібні частини тексту. авторитетні файли тощо).

Бібліографічні описи також часто відрізняються від бібліографічного опису , що використовується в традиційній АБІС. Широко використовувані формати метаданих , такі як Dublin Core, дозволяють не тільки описувати Інтернет-ресурси, але й створювати скорочені бібліографічні описи конкретних публікацій. У деяких випадках може бути відсутнім бібліографічний опис і, отже, електронний каталог взагалі. Є різних списків, у тому числі за автором, назвою та рейтингом. Очевидно, що для відносно невеликого фонду ЕВ такий список не так вже й важко знайти[6].

АБІС зазвичай складається з реляційної бази даних, програмного забезпечення , яке взаємодіє з базою даних, і двох графічних інтерфейсів користувача: одного для перегляду та іншого для співавторів.

Більшість окремих функцій програмного забезпечення АБІС є функціонально завершеними модулями, які об'єднані в загальний інтерфейс.

Приблизний перелік модулів включає:

- придбання фондів (замовлення, виставлення рахунків та отримання);
- каталогізація (внесення та бібліографічний опис примірників);
- обіг (видача/повернення примірників читачам);
- періодику (відстеження журналів та газет);

- OPAC — загальнодоступний електронний каталог та інтерфейс для користувачів.

Сучасна бібліотечна інформаційна технологія в основному базується на автоматизованих бібліотечно-інформаційних системах (АБІС) (системи автоматизації бібліотек «LAS»), які забезпечують користувачам обробку, аналіз і синтез, обробку та відображення каталогів бібліотечних документів ((Те саме, що традиційний папір). ) та електронні інформаційні ресурси) забезпечують бібліотечні колекції на кожному етапі від комплектування до замовлення.

Основним способом організації інформації в АБІС є створення баз даних. База даних (скорочено БД) — це впорядкована сукупність логічно взаємозалежних даних, які використовуються разом і призначені для задоволення інформаційних потреб користувачів. Технічно містить систему керування базою даних. Дані - у потоках документів, насамперед вміст і текст, що містяться в певному конверті, який має формат і фізичний носій, призначений для передачі та зберігання. Це: рукописи, книги, компакт-диски, мультимедійні ресурси, цифрові об'єкти, електронні документи тощо. - первинні інформаційні матеріали.[6].

Основне завдання бази даних полягає в тому, щоб забезпечити збереження великих обсягів інформації (так звані записи даних) і дозволити користувачам або прикладним програмам отримати до них доступ. Таким чином, база даних складається з двох частин: інформації, що зберігається, і системи її керування.

Щоб забезпечити ефективний доступ, записи даних організовані як набір подій (елементів даних).

Розрізняють структуровані та неструктуровані бази даних. Структуровані бази даних використовують структуровані дані, тобто структурований опис типу події з використанням схеми даних, також відомої як модель даних. Модель

даних описує об'єкти та зв'язки між ними. Існує декілька моделей (або типів) баз даних, основні з яких: плоска, ієрархічна, мережева та реляційна[6].

Основною функцією бази даних будь-якої організації є забезпечення прозорого управління інформаційними потоками, які підтримують операційну діяльність організації та визначають її майбутнє. Наявність комп'ютеризованої системи керування даними не гарантує ефективного використання даних. Така система є лише інструментом управління даними, і її необхідно ефективно використовувати для досягнення бажаних результатів. Таким чином, проблема автоматизації організаційної діяльності полягає не в установці комп'ютерів і програмного забезпечення, а в їх ефективному використанні[6].

Переваги ефективного використання бази даних в організації:

- інтерпретація та представлення даних шляхом структурування й упорядкування необробленої («сирої») інформації;
- поширення інформації серед визначеного кола людей у потрібний час;
- захист даних і контроль доступу до даних;
- контроль дублювання даних, використання даних як на зовнішньому, так і на внутрішньому рівнях.

За останні тридцять років розвиток бібліотек значною мірою визначався прогресом у комп'ютерних системах та інформаційних технологіях. Комп'ютери та технології автоматичної обробки інформації негайно почали використовуватися в бібліотеках, де були доступні значні ресурси і багато рутинних дій з обробки даних потрібно було частково або повністю формалізувати. Виникла перша автоматизована бібліотечно-інформаційна система (АБІС). Перше покоління АБІС базувалося на великих і середніх комп'ютерах. Ці системи не мають широкого поширення і використовуються лише у великих бібліотеках по всьому світу. На початку 1980-х років поява персональних комп'ютерів змінила використання комп'ютерних засобів:

бібліотеки почали створювати АБІС на базі персональних комп'ютерів. Іншими словами, система розташована на комп'ютері, а користувачі працюють у режимі розподілу часу.

Пізніше з'явилася АБІС на основі локальних комп'ютерних мереж. Ці системи охоплюють частину або весь традиційний бібліотечний цикл, включаючи: книговидавництва, зручні інтерфейси, властиві програмним системам для персональних комп'ютерів, сприяють їх широкому поширенню в бібліотеках.

### **1.3. Огляд інтелектуальних інформаційних систем аналогів**

Основними компонентами АБІС є реляційна база даних, спеціалізоване програмне забезпечення для взаємодії з цією базою даних і графічний інтерфейс користувача.

Будь-яка АБІС розробляється на основі класичного структурного програмування, в якому основним є принцип побудови систем за модульним принципом. Звичайно, кількість розроблених модулів і програм в різних системах різна, але основні модулі і програми присутні в кожній системі. До них належать: адміністрування, що включає функції модифікації та налаштування системи; збір коштів; каталогізація; науково-дослідна інформація та замовлення публікацій; обслуговування читачів; створювати Інтернет-сервіси, що виконують функції пошуку, упорядкування та каталогізації; надання бізнес-технологій; блок запису гравця; Модуль міжбібліотечного абонементу та електронного документообігу[12].

Kooha — це перша безкоштовна автоматизована бібліотечна інформаційна система (АБІС) . Розробка ABIS Kooha почалася в 1999 році в Новій Зеландії. Перше встановлення відбулося в січні 2000 року. Назва «kooha» означає подарунок, безкоштовний мовою маорі (Нова Зеландія)[12].

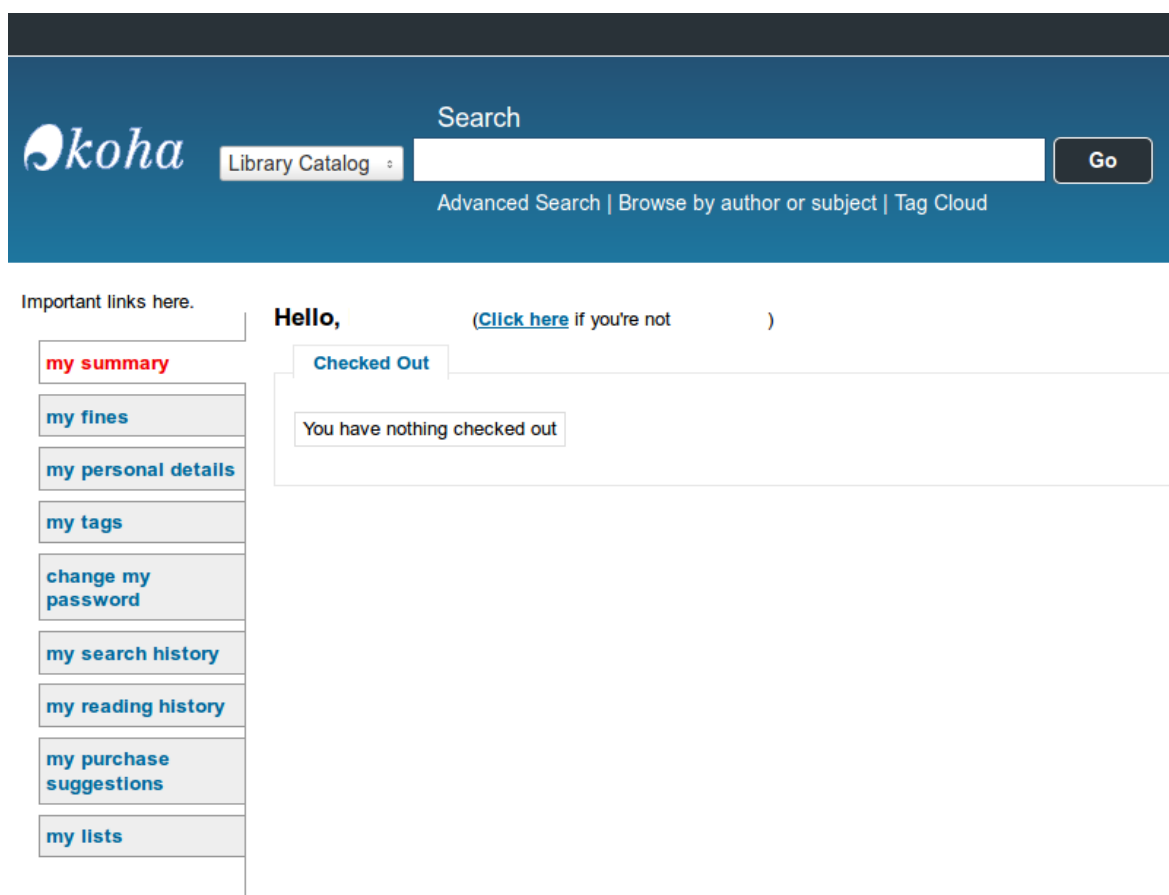


Рисунок 1.2 Автоматизована бібліотечна Інформаційна система «Коха»

Koha розроблено для підтримки технологічних процесів традиційних бібліотек. Koha побудовано на основі стандартів протоколу та бібліотеки, що забезпечує взаємодію між Koha та іншими бібліотечними системами та технологіями (рис. 1.2)[12].

Серверна частина АБІС Коха написана на мові програмування Perl, клієнтська частина інтерфейсу (електронний каталог) АБІС Коха написана на мові XHTML з використанням CSS 2.0. Інтерфейс бібліотеки вимагає налаштування браузера для роботи з JavaScript[12].

Koha має більшість функцій, очікуваних від програмного забезпечення АБІС, зокрема:

- Інтерфейс для бібліотекарів і читачів (відвідувачів)

- пошук
- Проста система контролю колекцій для невеликих бібліотек , включаючи дані про бюджетні витрати та ціни. (зі списком видавництв і конвертацією валют)
- Модуль каталогізації з інтегрованим клієнтом Z39. 50
- Книговидання та керування читачами
- Можливість працювати з будь-яким відділом комплекту, відвідувачами (каталог відвідувачів), примірниками (каталог копій), валюта та інші дані бібліотечних даних
- Система підтримки періодичних видань (журнали чи газети)[12].

Система керування бібліотекою Koha: Koha — це комплексна інтегрована бібліотечна система (ILS), розроблена для того, щоб допомогти бібліотекам керувати своїми колекціями, відвідувачами та різними бібліотечними заходами.

У 2005 році компанія Liblime (Огайо) оновила систему Koha, до неї інтегрували підтримку Zebra - високошвидкісної контекстуальної бази даних, що збільшило швидкість пошуку в АБІС Koha та поліпшило масштабованість системи.

OpenBiblio — це проста у користуванні система автоматизації бібліотек , написана на РНР, яка включає традиційні функції АБІС: , які підтримують онлайн-каталоги, видання книг, каталогізацію та керування персоналом і користувачами читайте.



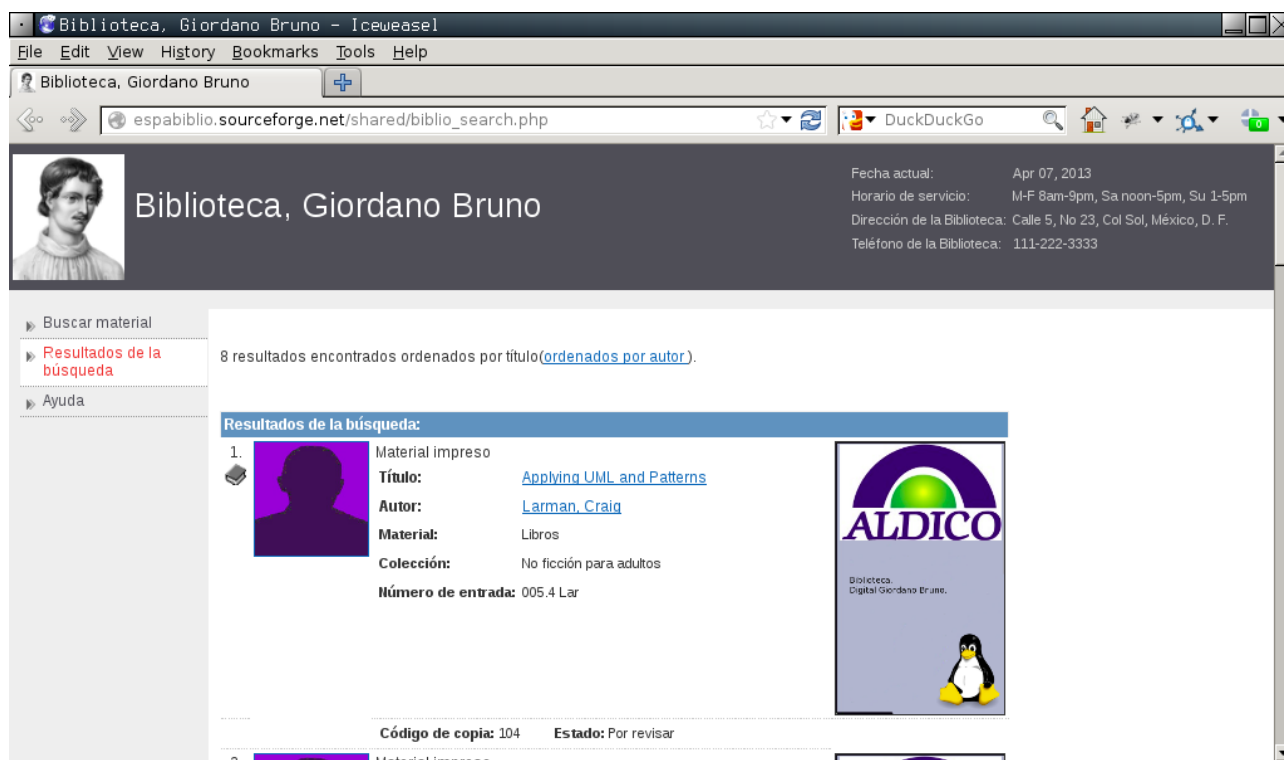


Рисунок 1.3 Інтегрована бібліотечна інформаційна система з відкритим вихідним кодом «OpenBiblio»

Це програмне забезпечення популярне в малих і сільські бібліотеки по всьому світу завдяки своїй простоті, широкій мовній підтримці та детальній документації. Призначений для невеликих бібліотек (рис.1.3) [12].

Хоча ця система ще знаходиться в стадії розробки, вона вже широко використовується в невеликих бібліотеках і архівах по всьому світу. Дослідники з Національного автономного університету Мексики рекомендували використовувати систему в місцевих бібліотеках Мексики, особливо тому, що вона підтримує мову науатль. Національна бібліотека Вірменії рекомендувала використовувати OpenBiblio для 900 малих бібліотек країни (менше 40 тис. томів). і сільські бібліотеки .

Ця система була перекладена на іспанську мову кастильським учителем і використана в початковій школі в Чилі. Крім того, за словами Вернера

Вестерманна з групи Educ caliber, Чилі, Колумбія, Куба та Венесуела висловили інтерес до цієї програми.

Департамент розвитку бібліотек штату Вайомінг розробив серію навчальних матеріалів для невеликих бібліотек, зацікавлених у впровадженні OpenBiblio. Дослідники з факультету інформаційних наук Федерального університету Параїби також обговорили використання OpenBiblio для навчання майбутніх бібліотекарів системам автоматизації бібліотек.

OpenBiblio має такі функціональні можливості:

- Підтримка формату запису MARC21
- Колекція
- Каталогізація: створення, редагування або видалення бібліографічних записів, включаючи імпорт записів у форматах MARC і MARCXML
- Інтернет-каталог публічного доступу (OPAC)
- Адміністрування: конфігурація та керування системою, включаючи бібліотеки, персонал, документацію, і параметри сайту
- Статистична звітність

Хоча OpenBiblio надає всі функції, необхідні для бібліотеки малого чи середнього розміру, але вона не включає багато більш складних функцій, включаючи колекцію та керування серійними файлами, які пропонують інші інтегровані бібліотечні системи. джерело, наприклад Koha або Evergreen.

Evergreen — це інтегрована бібліотечна система, розроблена Асоціацією PINES (Джорджія, США) для використання у великих публічних бібліотеках і загальнодержавних бібліотечних системах, які потребують обробки мільйонів документів із сотень бібліотек[13].

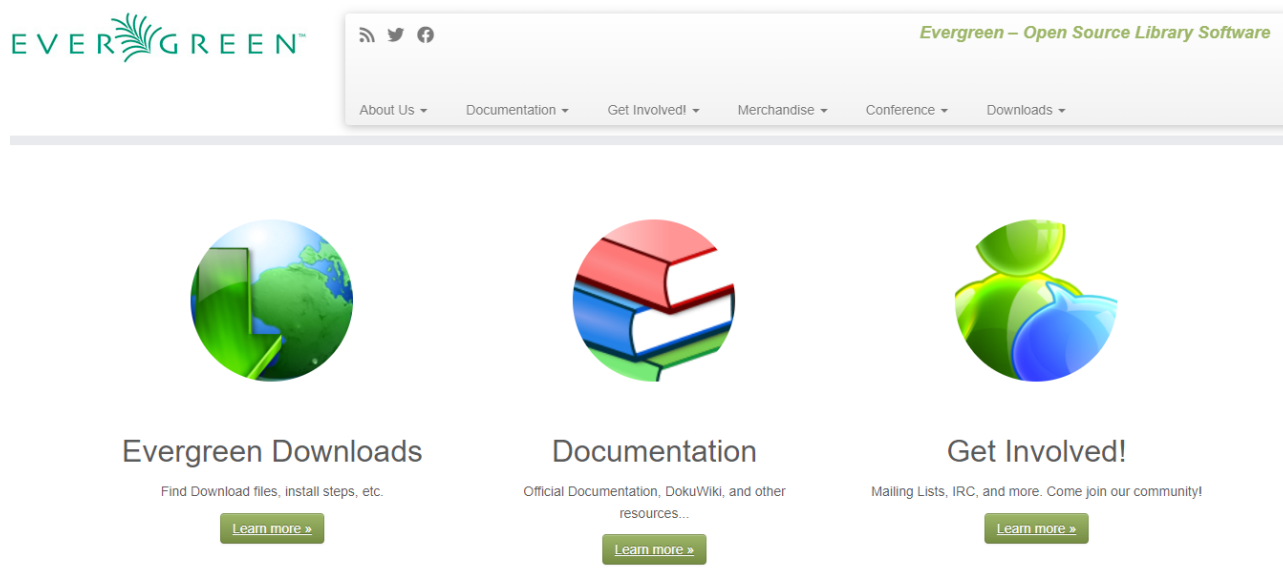


Рисунок 1.4 інтегрована бібліотечна система «Evergreen»

Це високоякісне бібліотечне програмне забезпечення, яке допомагає користувачам бібліотек знаходити бібліотечні матеріали, а також допомагає бібліотекам керувати, каталогізувати та розповсюджувати ці матеріали (рис.1.4)

Це програмне забезпечення розроблено для широкомасштабного розгортання у великих публічних бібліотеках і корпоративних середовищах штату. Однак Evergreen також можна використовувати в окремих бібліотеках-філіях. Сотні бібліотек використовують Evergreen.

Основна частина програми написана на мові програмування Perl, окремі модулі розроблені на мові C. База даних Evergreen реалізована в середовищі PostgreSQL. Інтерфейс електронного каталогу написаний на JavaScript і XHTML, тому для доступу до нього можна використовувати будь-який сучасний веб-браузер[13].

Evergreen забезпечує підтримку щоденних бібліотечних операцій, таких як архівування, реєстрація клієнтів і доступ до онлайн-каталогу. Evergreen має можливість прораховувати та контролювати умови видачі та повернення документів. Система дозволяє відстежувати місцезнаходження будь-якої книги,

компакт-диска чи іншого бібліотечного матеріалу. Evergreen має зручний інтерфейс каталогу веб-бібліотеки, що дозволяє читачам знаходити те, що їм потрібно, де б вони не були[13].

Можливості Evergreen включають в себе:

- Комплектування
- Каталогізація: надходження до фондів, введення бібліографічної інформації, класифікація та індексація документів
- Книговидача
- Онлайновий каталог публічного доступу (OPAC)
- Статистична звітність
- Пошук / введення інформації за допомогою протоколу Z39.50

Процес адаптації ABIS Evergreen за допомогою інтернаціоналізація, як система заснована на американських стандартах. Архітектура Evergreen є складною, і зміна конфігурації системи вимагає значних зусиль.

## 2. ПОСТАНОВКА ЗАДАЧІ ТА ПЛАНУВАННЯ ПРОЕКТУ

### 2.1 Мета та задачі проекту

Метою проекту є створення програмного забезпечення, яке спрощує та покращує управління бібліотечними ресурсами та послугами.

Очікується, що ця програма стане зручним інструментом для бібліотекарів та користувачів, що дозволить ефективно керувати книжковими фондами, реєструвати читачів, процесами видачі та повернення книг, а також мати зручний доступ. Отримувати доступ до ресурсів бібліотеки через веб-інтерфейс.

Наша мета – покращити доступність та якість обслуговування користувачів бібліотеки шляхом впровадження сучасних технологій та автоматизації рутинної роботи.

Такий веб-додаток допоможе підвищити ефективність роботи бібліотеки, скоротити витрати часу на адміністративні процедури та покращити обслуговування користувачів. Користувачі зможуть швидко знаходити потрібні книги, робити онлайн-запити, резервувати матеріали та отримувати повідомлення про статус запитів.

Адміністратори зможуть ефективно керувати каталогами книг, відстежувати видачу та повернення книг, а також аналізувати статистику використання бібліотечних ресурсів для покращення управління ними. Розробка цього додатка відповідає сучасним вимогам бібліотечної галузі, сприяючи цифровізації та модернізації цієї галузі.

Запропонована автоматизована бібліотечно-інформаційна система надає користувачам функцію створення персонального профілю, щоб вони могли зберігати історію пошуку, попередні документи та документи замовлення. Система може використовувати алгоритми та інтелектуальні технології для

надання рекомендацій користувачам. Він може аналізувати їхні інтереси, історію читання та інші дані, щоб надавати персоналізовані рекомендації щодо нового вмісту чи тем, які можуть їх зацікавити.

## 2.2 Вибір засобів реалізації проекту

**Мова програмування.** Python — потужна мова програмування, яку легко освоїти. Він має ефективну структуру даних високого рівня та простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис і динамічна типізація Python, а також його інтерпретований характер роблять його ідеальним для створення сценаріїв і швидкої розробки додатків у багатьох областях на більшості платформ. (рис.2.1)[14].

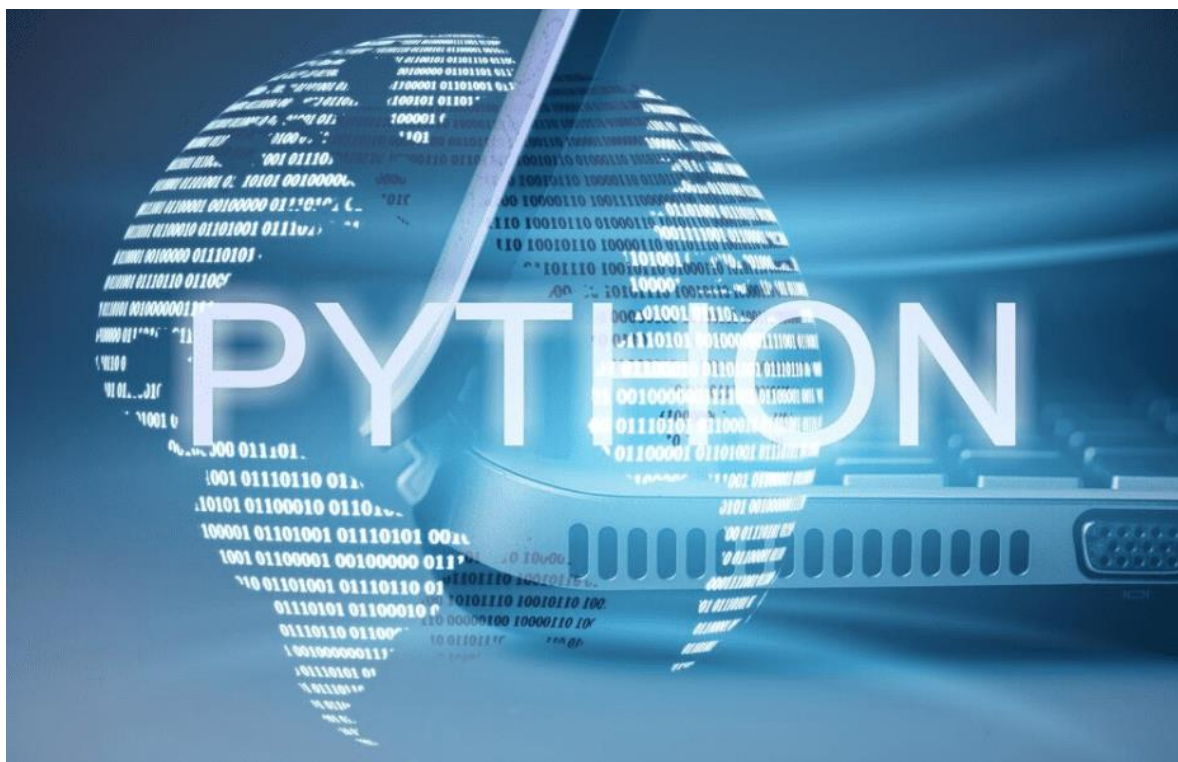


Рисунок 2.1- Мова програмування Python

Python має ефективні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів і той факт, що це інтерпретована мова, роблять його ідеальним для створення сценаріїв і швидкої розробки додатків у багатьох галузях промисловості на більшості платформ[14].

Мова Python широко використовується на великих підприємствах, оскільки підтримує різні парадигми програмування: об'єктно-орієнтоване, функціональне, структурне, імперативне та аспектно-орієнтоване. Python має велику, повну стандартну бібліотеку з автоматичним керуванням пам'яттю та динамічними функціями.

Багата стандартна бібліотека є однією з визначних пам'яток мови Python. Існують інструменти для роботи з багатьма мережевими протоколами та Інтернет-форматами, наприклад, модулі для написання HTTP-серверів і клієнтів, для аналізу та генерації повідомлень електронної пошти, для роботи з XML тощо.

На додаток до стандартної бібліотеки існує багато інших бібліотек, які забезпечують інтерфейси для всіх системних викликів на різних платформах; зокрема, на платформі Win32 підтримуються всі виклики Win32 API, а також COM не меншою мірою, ніж у Visual Basic або Delphi. Існує велика кількість бібліотек програм для Python у багатьох різних галузях: веб-розробка, бази даних, обробка зображень, обробка тексту, чисельні методи, програми операційної системи тощо[15].

Набір модулів для роботи з операційною системою, що дозволяє писати кросплатформні програми. Існують модулі для роботи з регулярними виразами, кодуванням тексту, мультимедійними форматами, протоколами кодування, зберіганням, серіалізацією даних, модульним тестуванням тощо[14].

Python підтримує динамічну типізацію, тобто тип змінної визначається лише під час виконання. Серед базових типів варто відзначити підтримку цілих

чисел довільної довжини і комплексних чисел. Python має багату бібліотеку для роботи з рядками, особливо закодованими в Unicode[14].

Python є стабільною та популярною мовою. Він використовується в багатьох проектах і з різними функціями: як основна мова програмування або для створення розширень додатків та інтеграції. На Python реалізовано велику кількість проектів, він також активно використовується для створення прототипів майбутніх програм.

Також Python відомий своєю універсальністю, що робить його популярним серед розробників з різним рівнем досвіду. Його простий синтаксис, який допомагає вам писати чіткий і зрозумілий код, є однією з головних причин, чому він такий популярний. Крім того, Python дуже розширюваний за допомогою багатьох бібліотек і фреймворків для різних типів завдань. Його інтерпретатор доступний на більшості платформ, а його відкритий вихідний код заохочує активну розробку мови та внески спільноти розробників[14].

Крім того, Python інтегрується з кодом C, що розширює його можливості та допомагає оптимізувати продуктивність. Навколо цієї мови також існує дружня та підтримуюча спільнота, яка завжди готова надати підтримку та допомогу як новим, так і досвідченим розробникам. Поєднання цих факторів робить Python чудовим вибором для різноманітних проектів, від веб-розробки до наукових досліджень[14].

Він також має ефективні високорівневі структури даних і простий, але ефективний підхід до об'єктно-орієнтованого програмування. Елегантний синтаксис Python, динамічна обробка типів і той факт, що це інтерпретована мова, роблять його ідеальним для створення сценаріїв і швидкої розробки додатків у багатьох галузях промисловості на більшості платформ[14].

Всі ці переваги цієї мови зробили її популярною і затребуваною сьогодні, дозволяючи Python розвиватися швидкими темпами, але незважаючи на всі



очевидні переваги, Python також має деякі недоліки, які, на мою думку, переважають його переваги.

До недоліків мови програмування Python можна віднести:

- Продуктивність. Більшість розробників, включаючи творця мови, погоджуються, що Python не такий гнучкий, як мав би бути. Це тому, що Python є інтерпретованою мовою. Але навіть у порівнянні з іншими інтерпретованими мовами ми виявили, що Python менш ефективний. Але це можна легко вирішити, використовуючи реалізацію цієї або проблемної частини коду на C. У нинішніх умовах це не дуже помітно.
- Синтаксис - так, на мій погляд, синтаксис також є плюсом, тому що якщо ви зміните мову програмування, синтаксис буде для вас незнайомим і трохи дивним, але до нього потрібно лише звикнути.
- Динамічний тип. Через динамічний тип Python споживає більше ресурсів, ніж міг би, але зазвичай це компенсується внутрішнім кешуванням.
- Глобальний ключ інтерпретатора. Зараз це головна проблема продуктивності в Python, як і погана реалізація багатопоточності. Код GPL не змінився з першої версії мови. Це чітко свідчить про те, що він застарів. Ще є надія, що розробники звернуть на це увагу в наступних версіях.

Виходячи з вищесказаного, я продовжую стверджувати, що будь-які недоліки мови переважають її переваги, яких у сучасних реаліях значно більше. І пам'ятайте, що мови розвиваються. Це дає надію, що всі недоліки будуть зменшені або повністю усунені[16].

У даній роботі використовуються три бібліотеки для створення графічного інтерфейсу користувача (GUI) в мові програмування Python:

**Tkinter** - Вбудована бібліотека графічного інтерфейсу, заснована на інструментах Tk (широко використовуються у світі GNU/Linux та інших UNIX-

подібних системах, а також перенесені на Microsoft Windows, Apple Mac OS тощо). Розповсюджується з відкритим вихідним текстом, написаним Стіном Ламхолтом. (Стін Лумхольт) і Гвідо ван Россум. Він включений до стандартної бібліотеки Python.

Tkinter - це окрема бібліотека, але вона складається з кількох окремих модулів, кожен з яких має окремі функції та власну офіційну документацію. Бінарна версія Python також включає додаткові модулі.

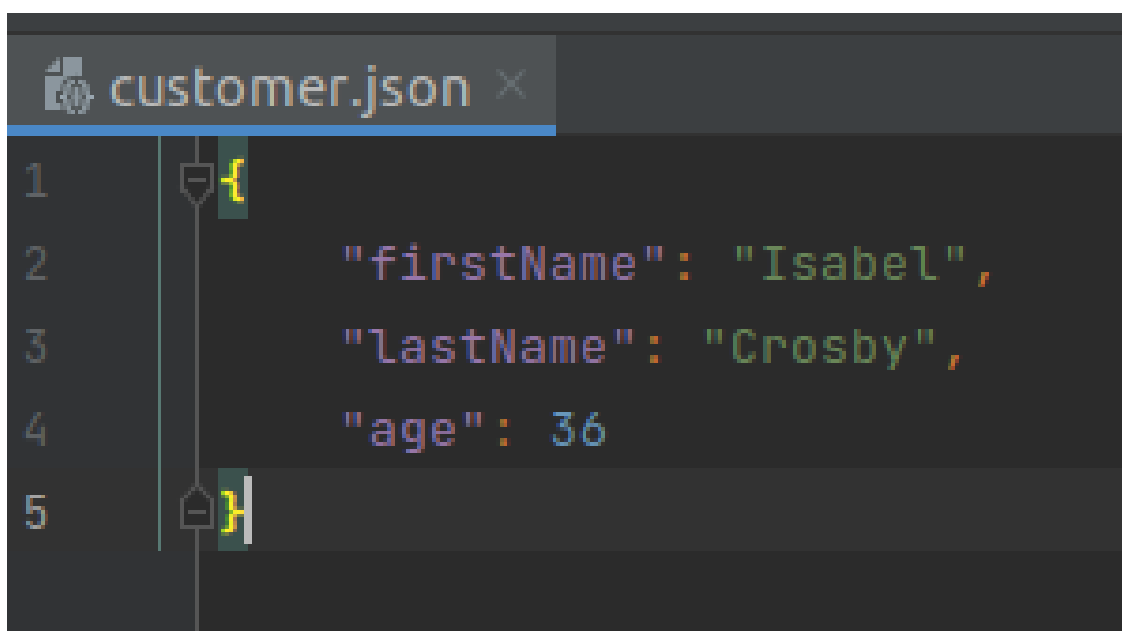
1. Tcl — це динамічно інтерпретована мова програмування, подібна до Python. Хоча його можна використовувати окремо як мову програмування загального призначення, найчастіше його інтегрують у програми C як інтерфейс до механізму створення сценаріїв або інструментарію Tk. Бібліотека Tcl надає інтерфейс C для створення та керування одним або декількома екземплярами інтерпретатора Tcl, виконання команд і сценаріїв Tcl у цих екземплярах, а також додавання спеціальних команд, реалізованих у Tcl або C. Пропозиції. Кожен інтерпретатор має чергу подій і спосіб надсилання подій інтерпретатору для обробки. На відміну від Python, модель виконання Tcl розроблена для спільної багатозадачності, і Tkinter заповнює цю прогалину.
2. Tk - це пакет Tcl, реалізований на C, який додає власні команди для створення та керування віджетами GUI. Кожен об'єкт Tk вбудовує власний екземпляр інтерпретатора Tcl, у який завантажується Tk. Віджети Tk дуже легко налаштувати, але вони виглядають застарілими. Tk використовує черги подій Tcl для створення та обробки подій GUI[17].

**Модуль Tkinter ttk** (тема tk) є розширенням tcl/tk із новим набором віджетів. ttk використовує нову систему віджетів. Цей механізм підтримує теми та стилі оформлення. Це дозволяє віджетам ТТК виглядати більш природно в різних операційних системах[17].

ttk (тема tk) є розширенням tcl/tk із новим набором віджетів. ttk використовує нову систему віджетів. Цей механізм підтримує теми та стилі оформлення. Це дозволяє віджетам ТТК виглядати більш природно в різних операційних системах. Це відносно нова функція Tkinter, тому я поясню трохи про встановлення цього модуля. Щоб використовувати ttk з попередніми версіями Python, ви повинні встановити його самостійно. Python-ttk наразі розміщено на [svn.python.org](http://svn.python.org). Модуль — це окремий файл (ttk.py), який потрібно розмістити в каталозі, де Python зможе його знайти. Quick Jump для версії 2.x, Quick Jump для версії 3.x.

З точки зору програміста, основна відмінність між новими та старими віджетами полягає в тому, що віджети ttk не мають опцій для налаштування їх зовнішнього вигляду. Наприклад, порівняйте кількість ПАРАМЕТРОВ ЗА ЗАМОВЧУВАННЯМ у старому та новому віджетах кнопок. Зовнішній вигляд віджета ТТК налаштовується за допомогою тем і стилів. В іншому випадку використання віджета ttk подібне до його аналога tk.

**Модуль json** у Python надає інструменти для роботи з форматом даних JSON (JavaScript Object Notation), який широко використовується для обміну даними між різними мовами програмування.



```
customer.json x
1 {
2     "firstName": "Isabel",
3     "lastName": "Crosby",
4     "age": 36
5 }
```

Рисунок 2.2 – Модуль json в роботі з JSON форматом

Крім того, Модуль json забезпечує функції серіалізації (перетворює об'єкти Python у формат JSON) і десеріалізації (перетворює дані JSON на об'єкти Python). Важливо зазначити, що json підтримує основні типи даних Python, але не всі типи даних можна серіалізувати в JSON без додаткового налаштування. (рис.2.2)

### 2.3 Огляд можливостей середовища розробки

**Середовище розробки.** PyCharm — це інтегроване середовище розробки для мови програмування Python. Надає інструменти аналізу коду, засоби перегляду графіки, інструменти для виконання модульних тестів і підтримку веб-розробки Django. PyCharm розроблено чеською компанією JetBrains на основі IntelliJ IDEA.

PyCharm розроблено JetBrains, відомою міжнародною компанією, яка розробляє інструменти програмування різними мовами, такими як Java, Kotlin, C#, F#, C++, Ruby, Python, PHP, JavaScript та багато інших. Спочатку заснована як IntelliJ, сьогодні компанія пропонує різноманітні інструменти для командної роботи та підтримки розробки для основних мов програмування. (рис.2.3)

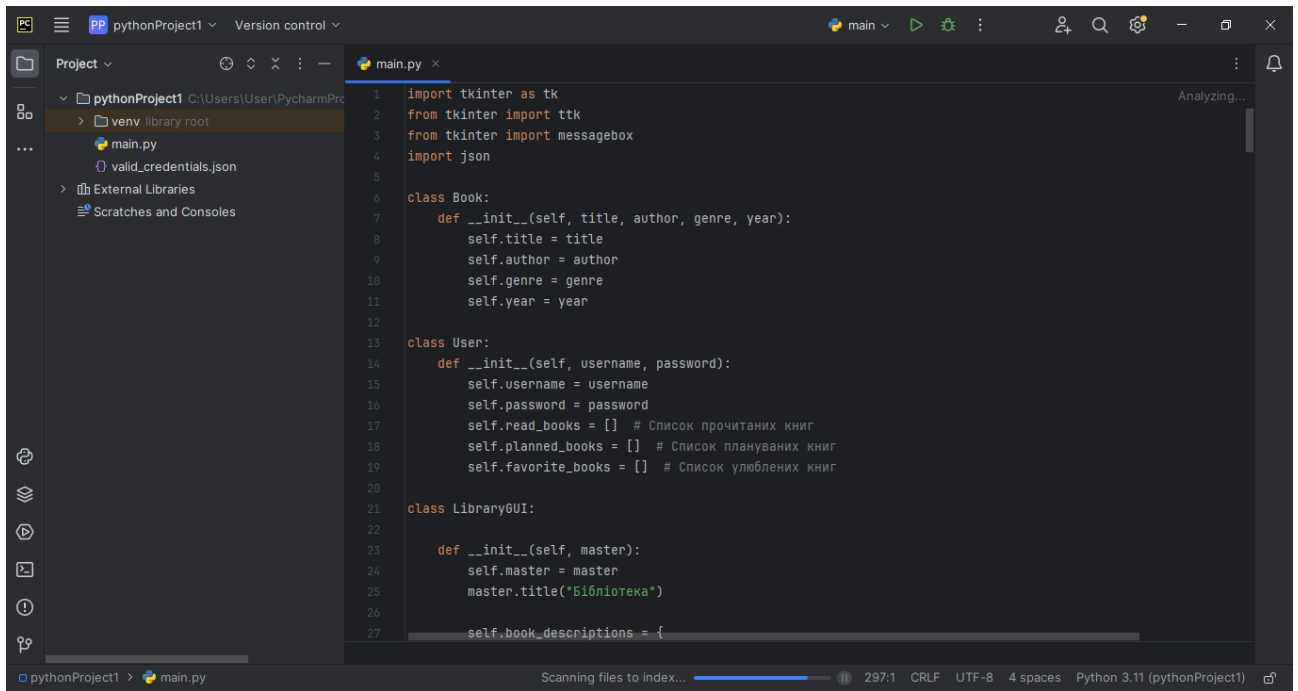


Рисунок 2.3 – Інтерфейс PyCharm інтегрованого середовища розробки (IDE) для мови програмування Python

Багатофункціональний редактор коду для мови Python під назвою PyCharm створений для швидкої та ефективної розробки програмного забезпечення. Наявна велика кількість інструментів, які дозволяють сконцентруватись на написанні саме бізнес-логіки застосунку не відволікаючись на рутинні речі: налаштування віртуального середовища, підключення системи контролю версій, встановлення зовнішніх бібліотек тощо.

PyCharm — це потужний інструмент розробки Python, який пропонує кілька важливих функцій для підвищення продуктивності програміста. Налагоджувачі дозволяють ефективно знаходити та виправляти помилки в коді, а засоби рефакторингу допомагають оптимізувати структуру програми без втрати її функціональності. Крім того, PyCharm підтримує інтеграцію з різними системами контролю версій, такими як Git і Mercurial, і надає зручний інтерфейс для роботи з віртуальним середовищем Python, дозволяючи розділяти залежності частин і бібліотеки для кожного проектух[3].

Крім того, автоматичне завершення коду в PyCharm спрощує процес написання програми, прискорює розробку та зменшує ризик помилок. Статичний аналіз коду, також включений у цей редактор, допомагає програмістам виявляти потенційні проблеми в коді на ранніх стадіях процесу розробки, допомагаючи підтримувати високу якість коду протягом усього проекту[3].

Використовуйте віртуальне середовище PyCharm, щоб зробити свій код чистішим і організованішим, уникнути конфліктів і покращити процес розробки.

Ми розглянули ключові функції та можливості PyCharm, які роблять його потужним інструментом для програмування на Python. Він надає практичні інструменти для управління проектами, налагодження коду, рефакторингу, інтеграції з системами контролю версій тощо. Ми вважаємо, що це значно спрощує процес розробки та дозволяє писати чистіший та ефективніший код.

**Erwin Data Modeler** (стилізований як erwin, але раніше ERwin) — комп'ютерна програма для проектування та документування баз даних. Моделі даних допомагають візуалізувати структури даних, забезпечуючи ефективні процеси для організації, управління та керування такими аспектами бізнес-операцій, як складність даних, технологія баз даних і середовище розгортаннях[5].

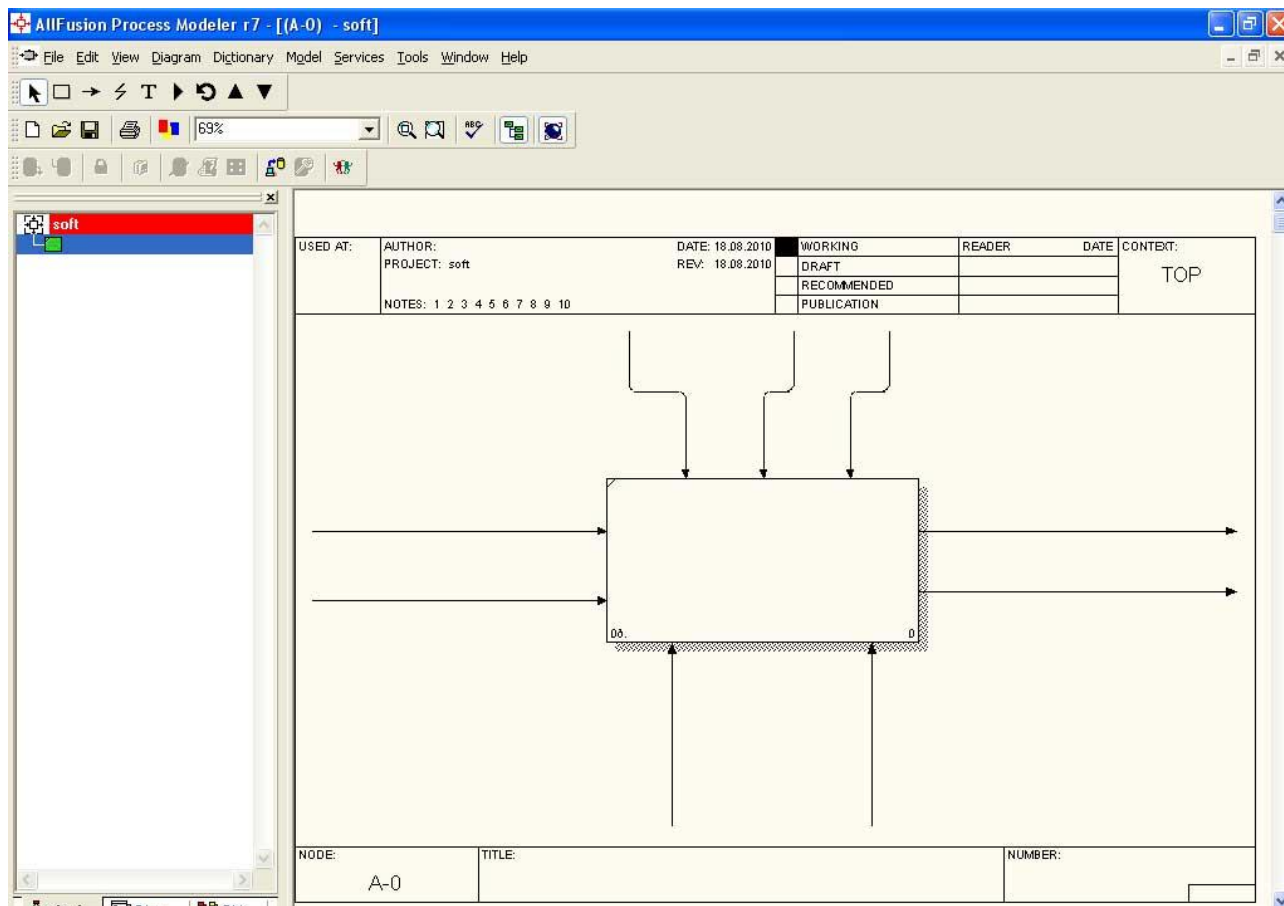


Рисунок 2.4 – Інтерфейс Erwin Data Modeler інструмент для моделювання даних

Програмний інструмент базується на методології IDEF1X, хоча тепер він також підтримує діаграми, що відображаються в альтернативній інженерній нотації для інформаційних технологій, а також нотації розмірного моделювання. (рис.2.4)

Користувачі можуть використовувати erwin Data Modeler для створення концептуальних моделей даних і створення логічних моделей даних, які залежать від конкретної технології бази даних. Ця діаграматична модель може бути використана для створення фізичної моделі даних. Потім користувачі можуть керувати технічною мовою визначення даних, необхідною для створення схем для певної системи керування базами даних. Програмне забезпечення містить функції для редагування графічної моделі, включаючи діалогові вікна для визначення кількості зв'язків сутностей, обмежень бази даних, індексів та

унікальності даних. Erwin підтримує три мови моделювання даних: IDEF1X, варіант комп'ютерної розробки, розроблений Джеймсом Мартіном, і нотацію розмірного моделювання.

Програмне забезпечення дозволяє користувачам створювати моделі даних шляхом зворотного проектування існуючих баз даних на основі кількох різних форматів. Інша функція включає можливість Ервіна створювати багаторазові стандарти дизайну: «включає стандарти іменування, стандарти типів даних, шаблони шаблонів тощо». Програмне забезпечення містить кілька функцій для зміни способу відображення моделі даних, включаючи параметри кількох кольорів, шрифтів, схем, тематичних областей і макетів[5].

Загалом, Erwin Data Modeler є важливим інструментом для розробників і архітекторів даних, який допомагає їм ефективно створювати, аналізувати та керувати структурами даних у великих підприємствах.

За допомогою Erwin Data Modeler розробники можуть ефективно проектувати оптимальні структури даних, що підвищує продуктивність і якість розроблених систем. Такі інструменти є важливою частиною процесу розробки програмного забезпечення для компаній будь-якого розміру, де доступні великі обсяги даних і точна організація даних важлива.

## 2.4 Особливості розробки бази даних

**База даних.** SQLite — це легка вбудована система керування реляційною базою даних (СУБД). На відміну від багатьох інших СУБД, таких як MySQL і PostgreSQL, він інтегрується безпосередньо у вашу програму без потреби в окремому сервері бази даних[2].

Особливість SQLite полягає в тому, що він не використовує клієнт-серверну модель, тобто драйвер SQLite не є окремим процесом, з яким взаємодіє



додаток, а надає бібліотеку, до якої компілюється програма, а драйвер стає частиною програми. . Тому виклики функцій бібліотеки SQLite (API) використовуються як протокол обміну. Такий підхід зменшує витрати, час відгуку та спрощує програму. SQLite зберігає всю базу даних (включно з визначеннями, таблицями, індексами та даними) в одному стандартному файлі на комп'ютері, на якому працює програма. Легкість реалізації досягається тим, що перед початком транзакції весь файл сховища бази даних блокується; Функції ACID спеціально реалізовані шляхом створення файлу журналу[4].

SQLite відомий своєю простотою використання. Ця простота робить його ідеальним для використання в невеликих проектах, прототипах або ситуаціях, коли потрібно швидко створити базу даних. Крім того, SQLite займає низькі ресурси, що дозволяє йому ефективно працювати на широкому спектрі пристроїв, таких як мобільні телефони та вбудовані системи. (рис.2.5)

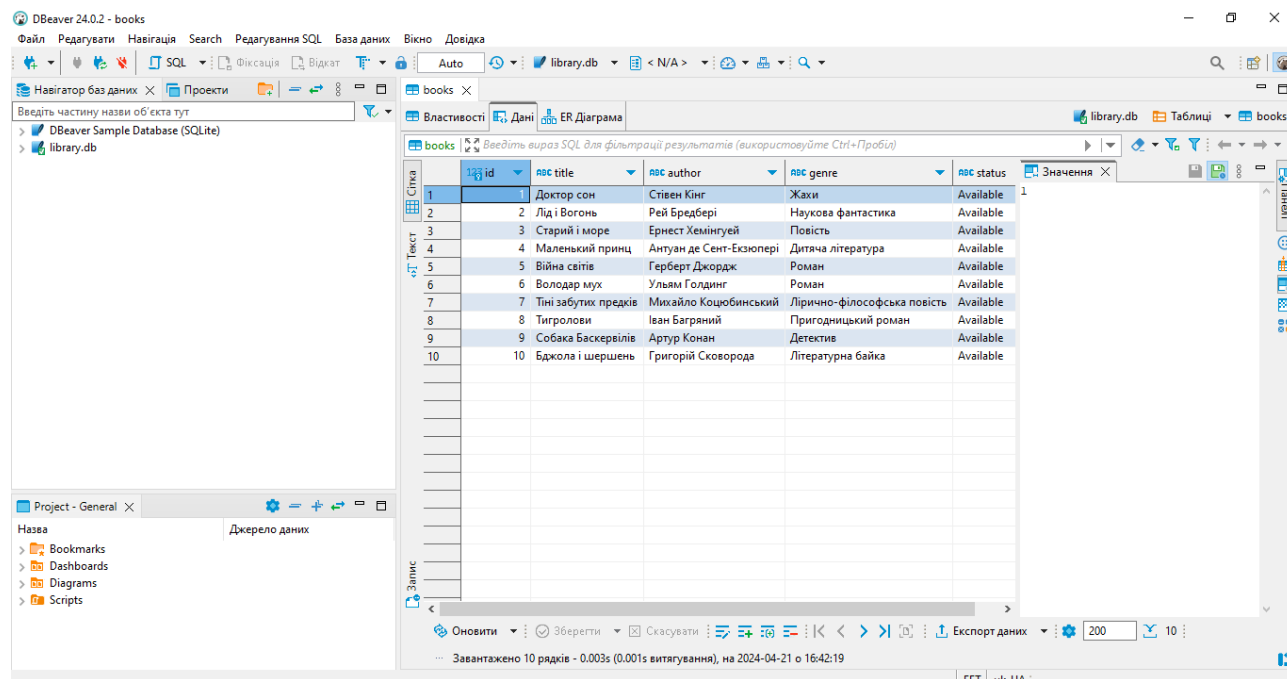


Рисунок 2.5 – Загальний вигляд створення бази даних і виконання SQL-запитів у Python.

З точки зору функціональності, SQLite підтримує більшість стандартних запитів SQL, що дозволяє розробникам легко інтегрувати SQLite в існуючі системи. Він також підтримує транзакції з підтримкою ACID для забезпечення надійності та цілісності даних. SQLite підтримує різні типи даних, такі як INTEGER, TEXT, REAL і BLOB, і є крос-платформним, тому він працює в багатьох операційних системах[4].

SQLite розширюється за допомогою вбудованих модулів і додаткових бібліотек, що дозволяє розробникам додавати функціональні можливості за потреби. Усі ці функції роблять SQLite дуже популярним інструментом для розробки додатків, яким потрібно зберігати та обробляти невеликі та середні обсяги даних.

SQLite — це проста у використанні інтегрована реляційна база даних, яка не потребує окремого сервера чи адміністрування, що робить її ідеальною для невеликих проектів або створення прототипів. Його компактність і легкість інтеграції дозволяють вбудовувати його безпосередньо в програмне забезпечення, спрощуючи розгортання та доставку додатків. SQLite підтримує стандартні функції SQL, включаючи транзакції, індексування та підзапити, що дозволяє розробникам створювати складні запити та оптимізувати керування даними.

Однак SQLite не підходить для великих систем із високим навантаженням, оскільки його архітектура використовує файлову систему для зберігання даних і не підтримує багатопотоковість. Це може викликати проблеми з блокуванням даних під час одночасних операцій запису, обмежуючи його ефективність у програмах із високим навантаженням. Крім того, у SQLite відсутні вбудовані механізми автентифікації та авторизації, що може створити проблеми безпеки даних у великих системах.

Незважаючи на ці обмеження, завдяки своїй простоті та легкості використання SQLite залишається популярним вибором для ряду програм, включаючи мобільні програми, веб-сайти та локальні бази даних. Залежно від

конкретних потреб проекту, належне використання SQLite може забезпечити ефективність системи, зменшити витрати на ресурси та спростити розробку.

### 3. ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ БІБЛІОТЕЧНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

#### 3.1 Структурно-функціональне моделювання автоматизованої бібліотечної інформаційної системи

Для моделювання структури та функції функціонування автоматизованої інформаційної системи бібліотеки пропонується використовувати метод ERwin. Функціональна модель ERwin відображає функціональну структуру об'єкта, тобто дії, що відбуваються в інтелектуальній інформаційній системі, і зв'язки між її компонентами.

Використовуючи підхід ERwin, ми маємо модель, що складається з сутностей, атрибутів, зв'язків, ключів, які посилаються один на одного. Відповідно до цього методу основними компонентами моделі є схеми, а всі функції та інтерфейси представлені блоками та дугами.

Використання цього інструменту дозволяє розробнику візуалізувати розроблену модель бази даних і забезпечити її перенесення у фізичну базу даних.

Ми використовували інструмент ERwin Data Modeler, який має графічний інтерфейс користувача, який дозволяє вам взаємодіяти з моделями даних, перетягуючи об'єкти на діаграмі, встановлюючи систему зв'язків, змінюючи властивості та інші операції. Він також надає функції для створення сценаріїв SQL, аналізу моделі, створення документації та інших функцій, корисних для роботи з моделями даних.

Розглянемо інтелектуальну інформаційну систему автоматизованої бібліотеки представлену на рис. 3.1.

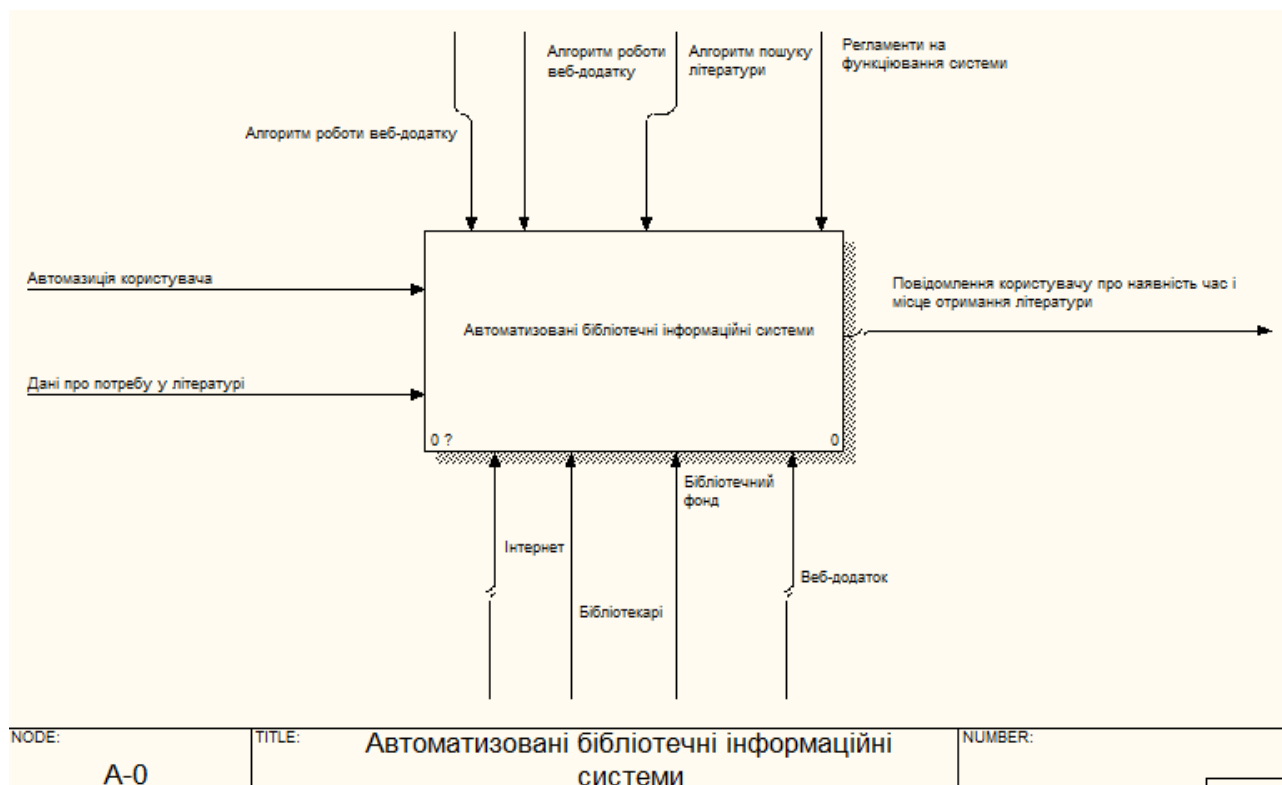


Рис. 3.1 – Робота автоматизованої бібліотеки

Окремими вхідними даними для функції «Проектування автоматизованої бібліотечної інформаційної системи» є:

- Автоматизація користувача;
- Дані про потребу у літературі;

На виході отримаємо автоматизовану бібліотечну інформаційну систему.

Автоматизована інформаційна система бібліотеки ведеться на термін, який може бути встановлений згідно з правилами внутрішнього розпорядку бібліотеки або іншого органу видачі документів. Наприклад, для книг, які мають читачів, можна встановити максимальний час перебування. Механізми та ресурси, включаючи Інтернет, бази даних і запропоновані програмні продукти.

Оскільки контекстна модель — це модель, яка представляє контекст або середовище системи, що аналізується або проектується. Дозвольте мені допомогти вам зрозуміти основи контекстного моделювання.

Дана модель використовується для опису взаємодії між системою та її зовнішнім середовищем, а також для визначення меж системи. Це дозволяє розглядати систему як частину більшої системи або зв'язку з іншими об'єктами.

Цей процес дозволяє детально ознайомитися з послідовністю роботи для досягнення бажаного результату – Проектування автоматичної інформаційної системи бібліотеки.

Пропонована модель включає два рівні деталізації. На першому рівні визначається загальний бюджет часу на роботу з видання літератури читачам. Цей бюджет часу можна розподілити на різні завдання, процеси або етапи, пов'язані з публікацією документа. На другому рівні розглядається планування часу для конкретних завдань або процесів, пов'язаних з публікацією документів. Це може включати такі завдання, як обробка запитів на випуск книг, підготовка замовлень читачів, доставка книг тощо.

Перша розділена на чотири блоки які подано на рис. 3.2. До них належать:

- Літератури;
- Замовлення;
- Отримання;
- Бібліотека;



управлінського впливу належить «Регламенти та функціонування системи», а ресурсами та механізмами є «Необхідна література».

Елементами для операції «Бібліотека» є «Оплата за дану літературу». Водночас вихідними складовими цієї діяльності є «Повідомлення користувачу про наявність час і місце отримання літератури». До процесів управлінського впливу належить «Регламенти та функціонування системи», а ресурсами та механізмами є «Необхідна література».

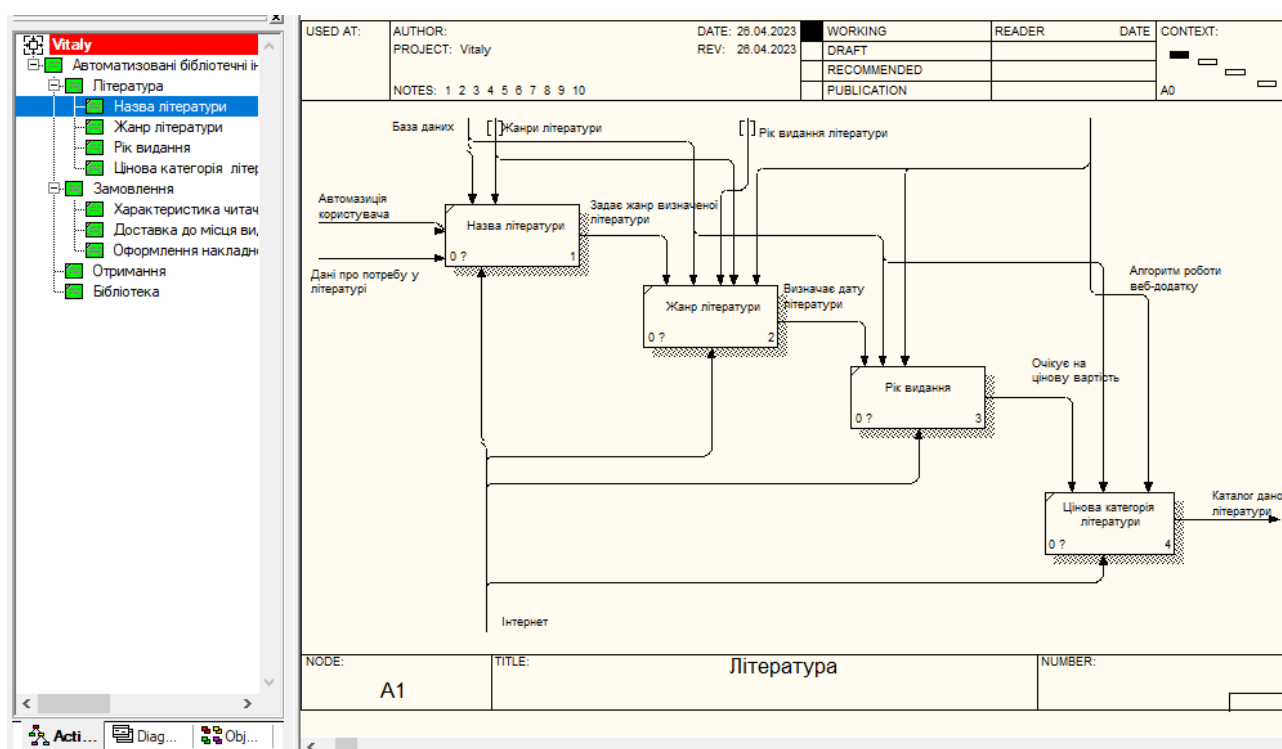


Рис. 3.3 – Діаграма декомпозиції IDEF0 «Заповнення БД»

Враховуючи те, що контекстна діаграма «Проектування автоматизованої бібліотечної інформаційної системи» має два рівні декомпозиції, нами побудовано відповідні моделі цих рівнів. Діаграму декомпозиції запропонованого процесу «Заповнення БД» представлено на рис. 3.3.

Початковими до діяльності «Назва літератури» є «Авторизація користувача». Водночас вихідними складовими цієї діяльності є «Задає жанр



визначеної літератури». У межах управлінського впливу належать – «Жанри літератури», а ресурсами та механізмами є «Необхідна література».

Вхідними до діяльності «Жанр літератури» є «Авторизація користувача». Водночас вихідними складовими цієї діяльності є «Визначає Дані про потребу у літературі». До процесів управлінського впливу належать – «Жанр літератури» та «Рік видання літератури», а ресурсами та механізмами є «Інтернет».

Елементами для операції «Рік видання» є «Очікування на цінову вартість». Водночас вихідними складовими цієї діяльності є «Інформація з БД». Процеси управлінського впливу включають – «Алгоритм роботи веб-додатку», а ресурсами та механізмами є «Інтернет» та «База даних».

Вхідними до діяльності «Цінова категорія літератури» є «Очікує на цінову вартість». вихідними складовими цієї діяльності є «Каталог даної літератури». Характерно до процесів управлінського впливу включають – «Алгоритм роботи веб-додатку», а ресурсами та механізмами є «Необхідна література».

Нами побудовано діаграму декомпозицій IDEF0 процесу «Перевірка обмежень щодо виконання роботи для видачі літератури читачеві», яка зображена на рис. 3.4.

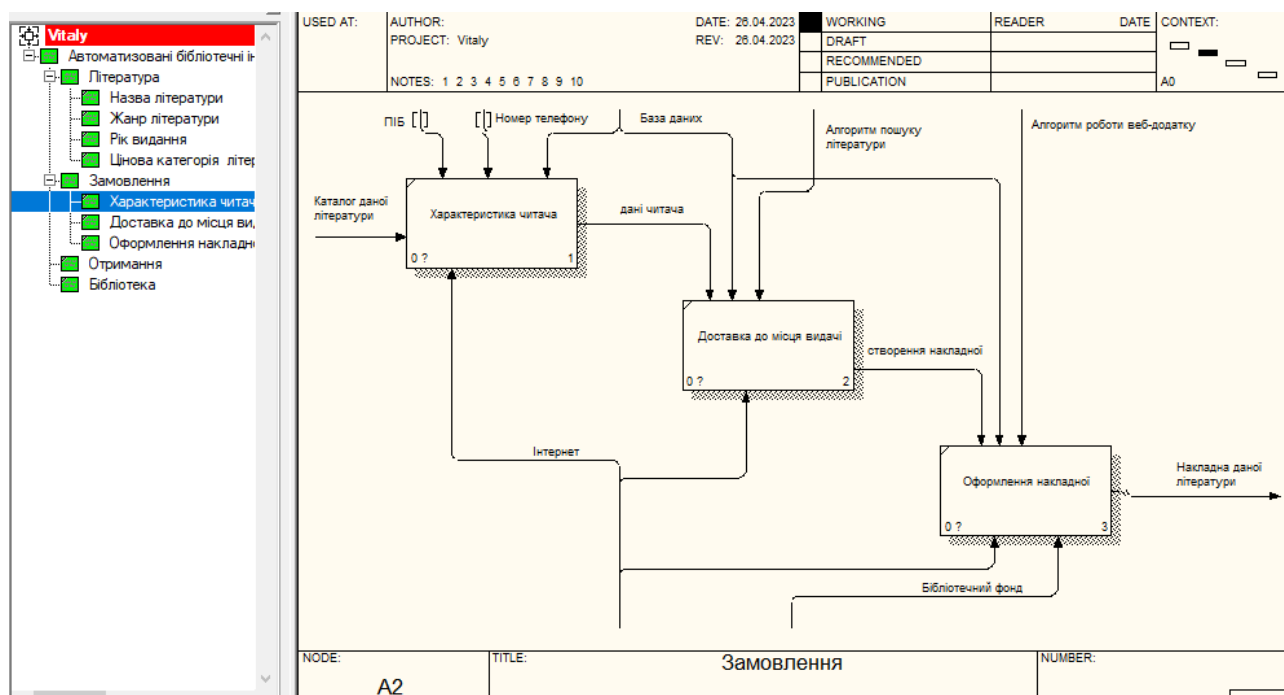


Рис. 3.4 - Діаграма декомпозиції IDEF0 "Перевірка обмежень автоматизованої бібліотечної інформаційної системи "

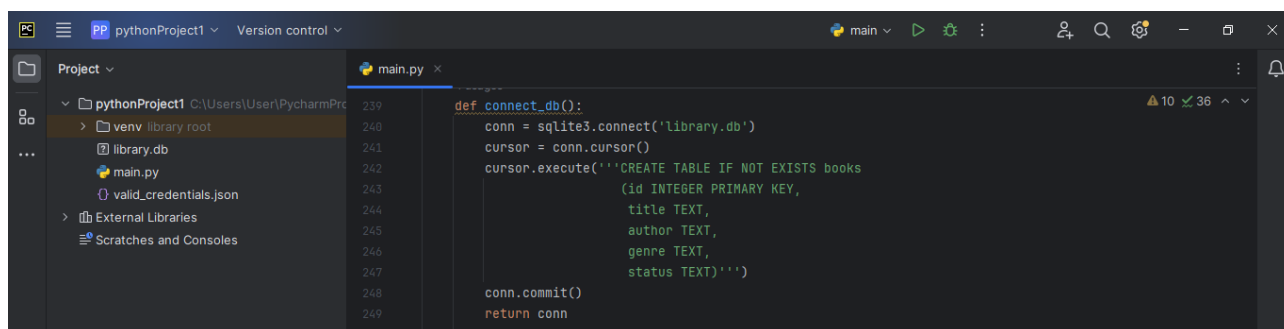
Початковими до діяльності «Характеристика читача» є «Каталог даної літератури». Водночас вихідними складовими цієї діяльності є «Дані читача». У межах управлінського впливу належать – «ПІБ», «Номер телефону», «Інформація з БД», а ресурсами та механізмами є «Інтернет».

Вхідними до діяльності «Доставка до місця видачі» є «Створення накладної». Водночас вихідними складовими цієї діяльності є «Накладна даної літератури». До процесів управлінського впливу належать – «Алгоритм пошуку літератури», а ресурсами та механізмами є «Інтернет».

Елементами для операції «Оформлення накладної» є «Створення накладної» та. Водночас вихідними складовими цієї діяльності є «Дані читача». Процеси впливу управління включають – «Алгоритм роботи веб-додатку», а ресурсами та механізмами є «Інтернет» та «Бібліотечний фонд».

## 3.2 Розробка бази даних для автоматизованої бібліотечної інформаційної системи

Для початку роботи розробки бази даних ми створюємо таблицю під назвою "books"



```
def connect_db():
    conn = sqlite3.connect('library.db')
    cursor = conn.cursor()
    cursor.execute('CREATE TABLE IF NOT EXISTS books
                   (id INTEGER PRIMARY KEY,
                    title TEXT,
                    author TEXT,
                    genre TEXT,
                    status TEXT)')
    conn.commit()
    return conn
```

Рисунок 3.5 - Підключення до бази даних SQLite і створення таблиці "books"

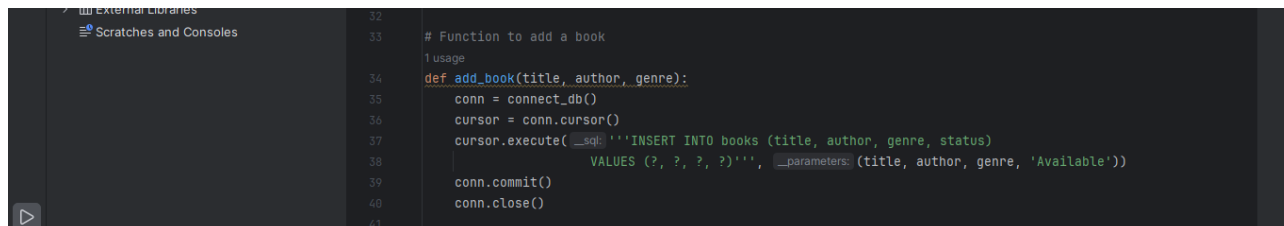
У цій частині коду функція `connect_db` підключається до бази даних SQLite і створює таблицю книг, якщо вона не існує. (рис.3.5)

У таблиці Книги визначено п'ять полів:

- ID: Унікальний ідентифікатор книги (ціле число), який є первинним ключем таблиці.
- title: назви книг (текст).
- author: Автор книги (текст).
- genre: Книга (текст) жанр.
- status: Статус книги (текст).

Функція `add_book` використовується для додавання нових книг до бази даних. Ця функція дозволяє користувачам вводити інформацію про книгу, наприклад: В. Назва, автор, рік видання та інші відомості. Після введення всіх

необхідних даних інформація про вашу книгу буде збережена в базі даних для подальшого використання. (рис.3.6)

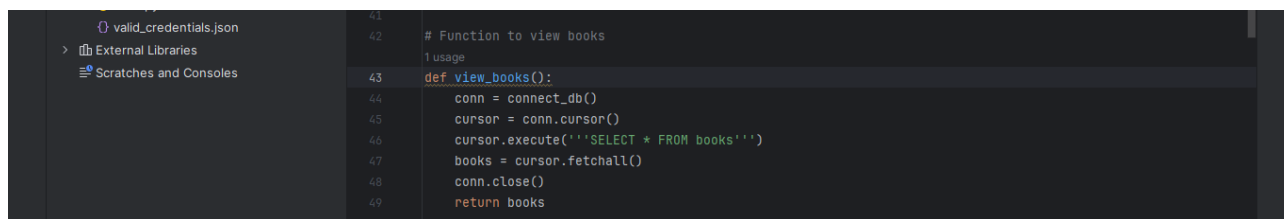


```
32
33 # Function to add a book
34 1 usage
35 def add_book(title, author, genre):
36     conn = connect_db()
37     cursor = conn.cursor()
38     cursor.execute( _sql: '''INSERT INTO books (title, author, genre, status)
39                     VALUES (?, ?, ?, ?)''', _parameters: (title, author, genre, 'Available'))
40     conn.commit()
41     conn.close()
```

Рисунок 3.6 – Код функцій add\_book

У рамках розробки веб-додатку автоматизованої інформаційної системи бібліотеки реалізовано функцію add\_book, яка допомагає системним адміністраторам додавати нові книги до каталогу бібліотеки. Ця функція містить форму, де адміністратори можуть вводити всю необхідну інформацію про книгу, таку як назва, автор, рік видання, жанр і номер ISBN. Окрім основних даних книги, add\_book також надає можливість завантажувати обкладинки книг для подальшого відображення в каталозі бібліотеки. Після введення всієї необхідної інформації та завантаження зображень дані будуть збережені в базі даних, що полегшить користувачам системи пошук і запозичення нових квитанцій. Функція add\_book створена таким чином, щоб забезпечити максимальну зручність та ефективність процесу поповнення бібліотечних фондів новими виданнями.

Функція view\_books використовується для отримання списку всіх книг, які зберігаються в базі даних. Ця функція шукає в базі даних і створює список, що містить інформацію про всі книги, наприклад: В. Назва, автор, рік видання, жанр. Після створення списку функція повертає його, і користувач може переглядати всі доступні книги. (рис.3.7)



```
41
42 # Function to view books
43 # Usage
44 def view_books():
45     conn = connect_db()
46     cursor = conn.cursor()
47     cursor.execute('SELECT * FROM books')
48     books = cursor.fetchall()
49     conn.close()
50     return books
```

Рисунок 3.7 – Код функцій view\_books

При розробці веб-додатку бібліотечної автоматизованої інформаційної системи реалізовано функцію view\_books, призначену для відображення каталогу бібліотеки користувачу. Ця функція забезпечує зручний інтерфейс для перегляду доступних книг, де користувачі можуть використовувати різні фільтри та критерії пошуку для пошуку конкретних публікацій. На сторінці view\_books кожна книга представлена з детальною інформацією, такою як назва, автор, рік видання, жанр і короткий опис, а також фото обкладинки. Користувачі мають можливість переглянути додаткові відомості про книгу та її наявність у бібліотеці, а також взяти онлайн. Функція view\_books реалізована з урахуванням інтуїтивно зрозумілої навігації та надання користувачам зручного доступу до вичерпної інформації про книжковий фонд бібліотеки, тим самим сприяючи продуктивній взаємодії з системою.

Функція update\_status використовується для оновлення статусу книг у базі даних. Ця функція дозволяє користувачам змінювати статус окремої книги. Наприклад, ви можете вказати, чи була книга вже прочитана чи ще очікує на прочитання. Щоб оновити статус книги, ця функція приймає назву книги та новий статус як аргументи, шукає книгу за такою назвою в базі даних і оновлює її статус. Після успішного оновлення статусу функція відображає повідомлення про зміну статусу книги. (рис.3.8)

```

51 # Function to add a book to planned or read list
52 def update_status(book_id, status):
53     conn = connect_db()
54     cursor = conn.cursor()
55     cursor.execute(_sql: '''UPDATE books SET status=? WHERE id=?''', _parameters: (status, book_id))
56     conn.commit()
57     conn.close()

```

Рисунок 3.8 – Код функцій як update\_status

У ході створення веб-додатку для автоматизації бібліотечної інформаційної системи була реалізована функція update\_status, призначена для оновлення статусу книг у системі. Ця функція дозволяє адміністраторам редагувати інформацію про доступність книги, вказуючи поточний статус книги: доступна для запозичення, видана або відсутня в бібліотеці. Крім того, адміністратори можуть встановлювати терміни повернення книг, відстежувати запозичення та бронювання, а також вносити інші необхідні зміни в статус книги на основі фактичних обставин. Функція update\_status інтегрована в інтерфейс адміністрування системи, надаючи адміністраторам зручні та ефективні інструменти для управління фондами бібліотеки та оптимізації всієї роботи бібліотеки.

Функція "search\_books" використовується для пошуку книг у базі даних за ключовими словами та категоріями. Ця функція дозволяє шукати книги за назвою або жанром, ввівши ключові слова. Ця функція аналізує вашу базу даних і вибирає книги, ключові слова яких відповідають назві або жанру книги. Коли ви виконаєте пошук, ця функція повертає список знайдених книг. (рис.3.9)

```

59 # Function to search books by genre or title
60 def search_books(keyword, search_by):
61     conn = connect_db()
62     cursor = conn.cursor()
63     if search_by == 'Genre':
64         cursor.execute(_sql: '''SELECT * FROM books WHERE genre=?''', _parameters: (keyword,))
65     elif search_by == 'Title':
66         cursor.execute(_sql: '''SELECT * FROM books WHERE title LIKE ?''', _parameters: ('%' + keyword + '%',))
67     books = cursor.fetchall()
68     conn.close()
69     return books

```

Рисунок 3.9 – Код функцій search\_books

При розробці веб-додатку бібліотечної автоматичної інформаційної системи реалізовано функцію `search_books`, що надає користувачам зручний інтерфейс для пошуку книг у каталозі бібліотеки. Ця функція дозволяє використовувати різні критерії пошуку, такі як назва книги, автор, жанр, рік видання та інші параметри, щоб швидко і точно знаходити потрібні документи. Результати пошуку відображаються у вигляді списку книг із детальною інформацією та зображеннями обкладинок, що дозволяє користувачам ознайомитися з книгами та отримати доступ до детального перегляду вибраного видання. Функціональність `search_books` розроблена, щоб зробити процес пошуку ефективнішим та інтуїтивно зрозумілішим, зробити систему доступнішою та зручнішою для користувача, а також покращити загальний досвід використання веб-програми.

### **3.3 Розробка та особливості користувацького інтерфейсу системи**

Цей веб-додаток забезпечує зручний доступ до каталогу книг, можливість пошуку за різними критеріями та можливість замовити літературу на самовивіз. Особливу увагу приділено дизайну, який відповідає останнім стандартам веб-розробки та забезпечує оптимальний перегляд на різних пристроях. Крім того, системні адміністратори мають доступ до розширеного керування базами даних, кредитного моніторингу та можливостей статистичного аналізу.

```

71 # GUI
72 1 usage
73 def main():
74     def login_button_click():
75         username = username_entry.get()
76         password = password_entry.get()
77         role = login(username, password)
78         if role == 'admin':
79             admin_menu()
80         elif role == 'user':
81             user_menu()
82         else:
83             error_label.config(text='Неправильний пароль або користувач')

```

Рисунок 3.10 – Код функції login\_button\_click

Функція login\_button\_click (рис.3.10) є частиною зовнішнього веб-додатку автоматизованої бібліотечно-інформаційної системи та відповідає за автентифікацію користувача. При натисканні кнопки входу функція отримує дані з полів введення логіна та пароля. Далі викликається функція login(username, password), щоб перевірити, чи правильна комбінація логіна та пароля в базі даних. Якщо перевірка пройшла успішно, а роль користувача — адміністратор, виконується функція admin\_menu(), і адміністратор перенаправляється до меню адміністрування системи. Якщо роль користувача — «Користувач», виконується функція user\_menu(), і звичайний користувач перенаправляється до головного меню системи. Якщо логін або пароль неправильні, користувач отримає відповідне повідомлення про помилку через error\_label.config(text='Password or user is incorrect'). Інтерфейс автоматизованої бібліотечно-інформаційної системи розроблено з урахуванням зручності та інтуїтивності для різних категорій користувачів.

login\_frame: Цей фрейм використовується для входу. Користувачі повинні ввести ім'я користувача та пароль для доступу до системи. Після введення правильних облікових даних користувач перенаправляється до відповідного фрейму (admin\_frame або user\_frame) залежно від його ролі. (рис.3.11)





```

132 # Login Frame
133 login_frame = Frame(root)
134 login_frame.pack()
135
136 username_label = Label(login_frame, text='Користувач:')
137 username_label.grid(row=0, column=0)
138 username_entry = Entry(login_frame)
139 username_entry.grid(row=0, column=1)
140
141 password_label = Label(login_frame, text='Пароль:')
142 password_label.grid(row=1, column=0)
143 password_entry = Entry(login_frame, show='*')
144 password_entry.grid(row=1, column=1)
145
146 login_button = Button(login_frame, text='Увійти', command=login_button_click)
147 login_button.grid(row=2, columnspan=2)
148
149 error_label = Label(login_frame, text='')
150 error_label.grid(row=3, columnspan=2)

```

Рисунок 3.11 – Код функції login\_frame

У рамках розробки веб-додатку Автоматизованої бібліотечно-інформаційної системи створено login\_frame, який відповідає за аутентифікацію користувачів. Ця рамка інтегрована в інтерфейс користувача системи та містить поля для введення логіна та пароля, а також кнопку для запуску процесу входу. Компонент login\_frame взаємодіє з серверною частиною системи, щоб перевірити дані, введені користувачем, і визначити роль користувача в системі (адміністратор або звичайний користувач). Якщо спроба входу не вдається, відображається відповідне повідомлення про помилку, яке сповіщає користувача про те, що логін або пароль неправильні.

dmin\_frame: Цей фрейм забезпечує адміністративний доступ до системи. Адміністратори мають розширені функції керування базами даних, такі як: В. Додавати, видаляти та редагувати інформацію про книгу. Ви також можете переглянути список користувачів та їх дії в системі. (рис.3.12)



```

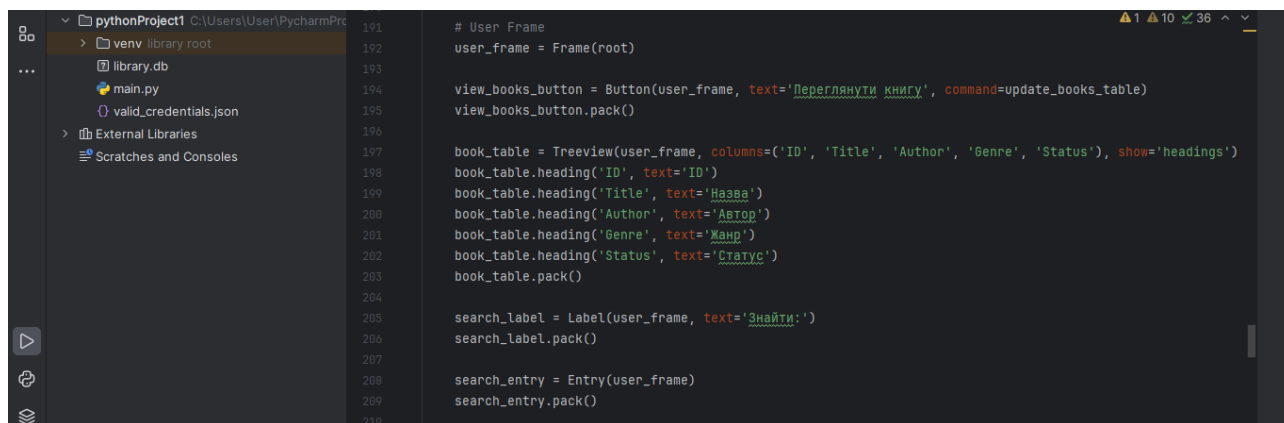
152 # Admin Frame
153 admin_frame = Frame(root)
154
155 title_label = Label(admin_frame, text='Назва книги:')
156 title_label.grid(row=0, column=0)
157 title_entry = Entry(admin_frame)
158 title_entry.grid(row=0, column=1)
159
160 author_label = Label(admin_frame, text='Автор:')
161 author_label.grid(row=1, column=0)
162 author_entry = Entry(admin_frame)
163 author_entry.grid(row=1, column=1)
164
165 genre_label = Label(admin_frame, text='Жанр:')
166 genre_label.grid(row=2, column=0)
167 genre_entry = Entry(admin_frame)
168 genre_entry.grid(row=2, column=1)
169
170 add_book_button = Button(admin_frame, text='Додати книгу', command=add_book_button_click)
171 add_book_button.grid(row=3, columnspan=2)

```

Рисунок 3.12 – Код функції `dmin_frame`

У веб-додатку системи автоматизації бібліотечної інформації реалізовано `admin_frame` для управління та управління ресурсами бібліотеки. Ця структура надає адміністраторам доступ до багатьох розширених функцій, зокрема: Додавайте, редагуйте та видаляйте книги в базі даних, керуйте користувачами, відстежуйте стан позики та повернення тощо. Він інтегрований у звичайний інтерфейс користувача системи та має чітко визначену структуру, яка спрощує навігацію та забезпечує зручний доступ до всіх необхідних параметрів керування. Крім того, `admin_frame` надає можливість створювати звіти та статистичні дані для аналізу роботи бібліотеки та планування подальших заходів щодо оптимізації бібліотеки.

`user_frame`: Цей фрейм призначений для звичайних користувачів. Переглядайте список доступних книг, шукайте книги за ключовим словом або жанром і оновлюйте статус прочитаних книг. Звичайні користувачі не мають доступу до адміністративних функцій у системному адмініструванні. (рис.3.13)



```
191 # User Frame
192 user_frame = Frame(root)
193
194 view_books_button = Button(user_frame, text='Переглянути книгу', command=update_books_table)
195 view_books_button.pack()
196
197 book_table = Treeview(user_frame, columns=('ID', 'Title', 'Author', 'Genre', 'Status'), show='headings')
198 book_table.heading('ID', text='ID')
199 book_table.heading('Title', text='Назва')
200 book_table.heading('Author', text='Автор')
201 book_table.heading('Genre', text='Жанр')
202 book_table.heading('Status', text='Статус')
203 book_table.pack()
204
205 search_label = Label(user_frame, text='Знайти:')
206 search_label.pack()
207
208 search_entry = Entry(user_frame)
209 search_entry.pack()
210
```

Рисунок 3.13 – Код функції user\_frame

У ході створення веб-додатку для автоматизації бібліотечної інформаційної системи була реалізована функція user\_frame. Вона призначена для забезпечення функціональності, яка дозволяє користувачам взаємодіяти з ресурсами бібліотеки. Ця структура включає можливість переглядати каталог доступних книг, шукати конкретні публікації за різними критеріями, а також процес запозичення та замовлення необхідної літератури. user\_frame спрощує взаємодію користувача з системою та забезпечує інтуїтивно зрозумілий інтерфейс із швидким доступом до всіх необхідних функцій. Крім того, в цьому рамках можуть бути реалізовані особисті облікові записи користувачів для перегляду історії позик, бронювань і відгуків про прочитані книги, підвищуючи зручність і персоналізацію взаємодії з системою.

Інтерфейс користувача веб-додатку автоматизованої бібліотечно-інформаційної системи має бути простим, зручним та інтуїтивно зрозумілим для користувачів. Основними елементами інтерфейсу будуть два поля введення та кнопка входу в систему. (рис.3.14)

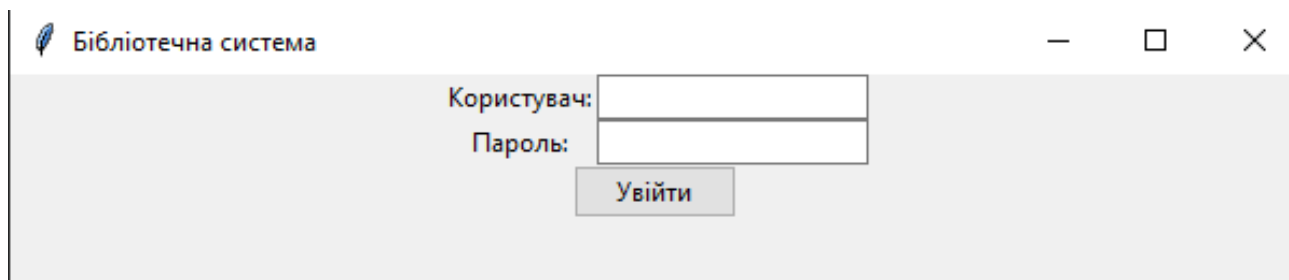


Рисунок 3.14 – Форма входу до автоматизованої бібліотечної інформаційної системи.

Після запуску програми на головному екрані з'явиться форма входу. Ця форма повинна містити два текстових поля: , одне для введення імені користувача (логіна) і одне для введення пароля. Поля мають бути чітко позначені відповідними мітками, щоб полегшити орієнтацію користувача.

Кнопку «Увійти» слід розмістити під полями введення. Кнопка має бути достатньо великою, щоб привертати увагу користувача та чітко вказувати на її функцію. Після натискання кнопки «Вхід» система перевірить введені дані та, якщо вони вірні, надасть доступ до основного інтерфейсу автоматизованої бібліотечної інформаційної системи.

Фон головного екрану можна зробити нейтральним і приємним для ока, щоб користувачі не відволікалися від основних елементів форми. Щоб покращити роботу користувача, ви також можете додати можливість показувати або приховувати паролі, кнопку відновлення пароля, якщо ви його забули, і посилання для реєстрації нового користувача, якщо така функція підтримується системою.

Після входу в автоматизовану інформаційну систему бібліотеки користувач потрапляє в головний інтерфейс, зручний для додавання та перегляду книг. У верхній частині екрана є поля для введення інформації про книги: назви, автори та жанри. Ці поля мають відповідні мітки, щоб користувачі могли легко зрозуміти, яку інформацію куди вводити. Поруч із полями введення

є кнопка «Додати книгу», за допомогою якої користувачі можуть додавати нові книги до бібліотечної системи. (рис.3.15)

The screenshot shows a window titled "Бібліотечна система". At the top, there is a form with three input fields labeled "Назва книги:", "Автор:", and "Жанр:". Below these fields is a button labeled "Додати книгу". Below the form is a table with the following data:

ID	Назва	Автор	Жанр	Статус
1	Доктор сон	Стівен Кінг	Жахи	Available
2	Лід і Вогонь	Рей Бредбері	Наукова фантастика	Available
3	Старий і море	Ернест Хемінгуей	Повість	Available
4	Маленький принц	Антуан де Сент-Екзюпері	Дитяча література	Available
5	Війна світів	Герберт Джордж	Роман	Available
6	Володар мух	Ульям Голдинг	Роман	Available
7	Тіні забутих предків	Михайло Коцюбинський	Лірично-філософська повість	Available
8	Тигролови	Іван Багряний	Пригодницький роман	Available
9	Собака Баскервілів	Артур Конан	Детектив	Available
10	Бджола і шершень	Григорій Сковорода	Літературна байка	Available

At the bottom of the table, there is a button labeled "Вихід".

Рисунок 3.15 – Інтерфейс автоматизованої бібліотечної інформаційної системи

Під полями введення та кнопкою «Додати книгу» є таблиця доступних книг.

Таблиця містить кілька стовпців: ID: — це унікальний ідентифікатор книги, який дозволяє легко відрізнити одну книгу від іншої. Назва: — це назва книги, яка допомагає користувачам швидко знайти потрібну книгу. Автор: імена авторів книг, які допомагають ідентифікувати книги за авторством. Жанр: жанри книг, які допомагають легко шукати та сортувати книги за жанром. Статус : статус доступності книги . У цьому випадку всі книги мають статус «доступно», тобто книгу можна забрати.

У таблиці наведено різні книги разом із їх авторами та жанрами. Ця таблиця оновлюється автоматично, коли додаються нові книги, що дозволяє користувачам бачити оновлений список доступних книг. Інтерфейс створено для забезпечення легкого та швидкого доступу до інформації про книги та ефективного управління бібліотечними фондами.

## 4. ОХОРОНА ПРАЦІ

### 4.1 Аналіз небезпек при роботі з комп'ютером

Виявилося, що під час роботи за комп'ютером найбільшому ризику піддаються зорова, моторна та нервово-психічна системи. Незрозуміло, що саме турбує, радіація чи постійна статична поза.

Тривала робота за комп'ютером також може призвести до розвитку синдрому зап'ястного каналу через надмірне навантаження на зап'ястя при роботі з клавіатурою і мишкою. Крім того, безперервна робота в закритому, недостатньо провітрюваному приміщенні може викликати головний біль, запаморочення і загальне зниження працездатності через брак свіжого повітря і надмірне накопичення вуглекислого газу. Важливо враховувати психологічні фактори, такі як стрес і виснаження, які можуть виникнути через велике робоче навантаження та відсутність належного відпочинку.

Працюючи за комп'ютером протягом тривалого часу, важливо звернути увагу на фактори, які можуть спричинити порушення зору, наприклад, неправильне налаштування яскравості екрана, недостатній контраст і занадто дрібні шрифти, які можуть призвести до напруги очей і погіршення зору. Крім того, робота в умовах недостатнього освітлення або, навпаки, занадто яскравого світла може негативно вплинути на зоровий апарат, викликати напругу очей і головний біль.

Погана поза за комп'ютером може призвести до хронічних болів у спині, шії та плечах, причиною яких може бути неправильне робоче місце або незручне крісло. Тривале сидіння в одному положенні без регулярних перерв для розминки може спричинити застій крові та збільшити ризик розвитку варикозного розширення вен.

Робота в шумному середовищі може призвести до зниження концентрації та підвищення рівня стресу, що негативно впливає на загальну продуктивність і може спричинити психоемоційне виснаження. Крім того, тривале використання комп'ютера може спричинити порушення сну, якщо ви не дотримуєтеся правил гігієни сну, наприклад, уникаєте використання електронних пристроїв перед сном і дотримуєтеся розпорядку дня.

Дисплеї є основним джерелом небезпеки. Вони випромінюють різні типи випромінювання, включаючи рентгенівське, ультрафіолетове, інфрачервоне та електромагнітні хвилі. Для кожного з цих випромінювань розроблені гранично допустимі норми, але вони досить умовні і відрізняються від країни до країни. Хоча стандарт передбачає, що опромінюється все тіло людини, насправді опромінюється лише верхня частина тіла. Хоча наведені вище критерії встановлюються окремо для кожного типу впливу, насправді всі зони діють одночасно і їх комплексний вплив ще не вивчений.

Крім того, відеотермінали порушують баланс позитивно та негативно заряджених іонів у повітрі. Електростатичне поле дисплея притягує негативні іони, виводячи всю атмосферу з рівноваги. Це також шкідливо для вашого здоров'я. Після години роботи біля монітора можна помітити, що негативні іони майже повністю зникають. Тому в робочу зону за комп'ютером має надходити свіже повітря. У зв'язку з усіма цими небезпеками розміри столів і стільців для роботи з комп'ютером чітко регламентовані. Тому хитка постава негативно позначається на опорно-руховому апараті. Стіл повинен бути містким зі спеціальною підставкою для ніг, а робочий стілець – регульованим по висоті, сидінням і спинкою, нахиленими під кутом.

Є два джерела радіації. системний блок і монітор.

1. Системний блок створює тільки електромагнітні поля (випромінювання). Також є певний шум від вентилятора, але ця проблема

очевидна для всіх і не потребує знання електроніки. Пошкодження, спричинені електромагнітними полями, безумовно присутні, коли напруженість поля висока. Однак польові комп'ютери набагато менші за мобільні телефони.

2. У моніторі є два основних шкідливих елементи. Бета-випромінювання (простіше потік електронів), яке фактично створює зображення на екрані, і висока напруга (досягає 16-20 кіловольт, як в інших телевізорах) викликають іонізацію повітря. Бета-випромінювання поширюється через монітор у двох напрямках: вперед і назад. У старих телевізорах і моніторах випромінювання досягало 1-2 метрів від екрана (ми всі пам'ятаємо, коли нам радили сидіти далі 2-3 метрів від телевізора). Тож вийшов дуже потужний прожектор і випустив у нас електронний промінь. У процесі він вибиває електрони з молекул повітря та перетворює їх на позитивні іони, які дуже шкідливі для людини. В даний час рівні бета-випромінювання в моніторах настільки низькі, що електрони відлітають за кілька сантиметрів від екрана. Основне випромінювання монітора спрямоване назад. Тому «дальність ураження» становить 1,5 метра. Цього слід уникати. Висока напруга дозволяє захоплювати електрони з молекул повітря і перетворювати їх на шкідливі катіони. Це надихає, оскільки виробники моніторів і телевізорів стикаються з дедалі суворішими вимогами щодо використання високої напруги.

## **4.2 Освітлення та вентиляція робочої зони**

Освітлення робочої зони необхідно організувати таким чином, щоб створювати комфортні умови для роботи за комп'ютером. Зокрема, використання штучного та природного освітлення має мінімізувати відблиски на екрані керування. Вентиляція робочих приміщень відіграє важливу роль у підтримці здорового мікроклімату, забезпечуючи постійний приплив свіжого повітря та видаляючи затхле повітря, сприяючи підвищенню продуктивності праці та



зниженню стомлюваності. Оптимальне освітлення знижує навантаження на очі і підвищує загальну ефективність роботи, а достатня вентиляція запобігає надмірному накопиченню в повітрі вуглекислого газу та інших шкідливих речовин, що позитивно впливає на здоров'я користувачів системи.

Згідно з правилами, під час роботи за комп'ютером світло має падати з лівого боку, а відстань між очима та екраном має становити близько 50 сантиметрів. Крім того, відрегулюйте стілець так, щоб ваші очі були на одній висоті з центром монітора. Експерти стверджують, що найбільший стрес для саамських очей викликає робота за комп'ютером.

Для забезпечення відповідного рівня освітлення необхідно використовувати загальне та місцеве освітлення, причому загальне освітлення забезпечує рівномірне освітлення всієї робочої зони, а місцеве освітлення – пряме освітлення робочого місця, де знаходиться комп'ютер. Використання ламп з регульованою яскравістю дозволяє регулювати освітлення в залежності від часу доби та особистих потреб користувача, створюючи комфортні умови під час тривалої роботи. Що стосується вентиляції, важливо забезпечити ефективну систему повітрообміну, включаючи як природну вентиляцію через вікна, що відкриваються, так і механічну вентиляцію за допомогою всмоктування та повітродувок для забезпечення оптимального рівня вологості та температури. Такий підхід до організації освітлення та вентиляції робочих зон може сприяти розробці веб-додатків автоматизованих бібліотечних інформаційних систем, підвищити загальну ефективність роботи та знизити ризик ризику професійних захворювань.

Я виявив, що мої очі перестають моргати, коли я довго дивлюся на екран. Через це очі червоніють і сльозяться, а зір погіршується. Мала відстань до екрану, дрібний шрифт, мерехтіння, різне освітлення в кінцевому підсумку призводять до короткозорості. Якщо ваші очі почервоніли, сльозяться, ви відчуваєте печіння і починає боліти голова, це ознаки того, що ваші очі вже

втомилися і потребують відпочинку. Але, звичайно, краще не доводити себе до такого стану.

### 4.3 Інструкція з охорони праці під час роботи за комп'ютером

Персонал, який працює з ЕОМ, повинен виконувати вимоги інструкцій, складених на основі санітарних норм і правил, виконувати вимоги безпеки при роботі та уникати виробництва, яке є небезпечним або шкідливим для інших працівників або виводить з ладу ЕОМ особиста відповідальність за виготовлення елементів.

До шкідливих і небезпечних факторів при роботі з комп'ютером належать:

- Електростатичне поле.
- Електромагнітне випромінювання.
- Наявність сильного іонізуючого випромінювання.
- Місцева втома, загальна втома.
- Напруга очей.
- Ризик ураження електричним струмом.
- Небезпека пожежі.

Режим праці та відпочинку під час роботи за комп'ютером повинен бути організований залежно від виду та виду професійної діяльності. Види робіт поділяються на 3 групи:

- ❖ Група А – читання інформації на екрані комп'ютера з попередніми вимогами;
- ❖ Група Б – робота з введення інформації;
- ❖ Група Б – створення в діалоговому порядку.

Основною роботою з комп'ютером слід вважати роботу, яка передбачає не менше 50% часу роботи за комп'ютером. Для видів робіт встановлено 3 види важкості та напруженості роботи з ЕОМ, які визначаються таким чином:

Для групи А - загальна кількість символів, які можна прочитати при роботі з комп'ютером, але не більше 60000 знаків;

для групи Б – загальна кількість символів, прочитаних або введених під час роботи з комп'ютером, але не більше 40 000 знаків;

для групи В - загальний час роботи безпосередньо з комп'ютером, але не більше 6 годин за час роботи з комп'ютером. Ви повинні регулярно робити перерви під час роботи за комп'ютером, щоб забезпечити оптимальну продуктивність і здоров'я.

Перед початком роботи необхідно переконатися, що екран комп'ютера має антиблокуюче покриття (крім групи А) з коефіцієнтом відбиття не більше 0,5. Покриття також повинно забезпечувати видалення статичної електрики з поверхні екрану, іскор і накопиченого пилу. Корпус дисплея повинен забезпечувати захист від іонізуючого та неіонізуючого випромінювання.

Необхідно перевірити робоче положення комп'ютера, відстань між стіною, вікном і столом не менше 0,8 м. Відстань між партами повинна бути не менше 1,2 м. Не дозволяється знаходити друге робоче місце ззаду.

## ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Проведено аналіз автоматизованої інформаційної системи бібліотеки з метою визначення її ефективності в управлінні ресурсами та обслуговуванні користувачів. Пропоновані вдосконалення системи включають удосконалення інтерфейсу користувача та розширення його функціональних можливостей, таких як можливість переглядати та оцінювати книги, ділитися рекомендаціями та організовувати книжкові клуби.

Подібну оцінку було зроблено для інтелектуальних інформаційних систем, таких як Koha та Evergreen, які є важливими інструментами для автоматизованих бібліотечних інформаційних систем.

Рекомендована автоматизована бібліотечно-інформаційна система надає користувачам можливість створювати персональні профілі та отримувати персоналізовані рекомендації за допомогою алгоритмів та розумних технологій.

Контекстна діаграма для впровадження процесу з використанням інтелектуальних інформаційних систем та автоматизованих бібліотечних інформаційних систем включає два рівні деталізації, що дозволяє розглянути загальний бюджет часу та конкретне планування часу, можливе для завдань або процесів, пов'язаних із публікацією документів.

IDEF0 розробив діаграми декомпозиції процесу, такі як «Наповнення бази даних», «Граничне тестування автоматизованої бібліотечно-інформаційної системи» та «Проектування автоматизованої бібліотечно-інформаційної системи» як основу для розробки розумних інформаційних систем для бібліотек.

Функціональний опис, через представлення варіантів використання, допомагає краще зрозуміти автоматизовану бібліотечну інформаційну систему, вказуючи перелік функцій, які система повинна виконувати для встановлення загальних вимог.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Маенда, Д. (2019). Designing databases, including schema definition. *Книга How to Speak Machine: Laws of Design for a Digital Age* (с. 186). Penguin.
2. sqlite3 — інтерфейс DB-API 2.0 для баз даних SQLite [Електронний ресурс]. Режим доступу: <https://docs.python.org/uk/3.9/library/sqlite3.html>
3. PyCharm: [Електронний ресурс]. Режим доступу: <https://pycharm.blogspot.com/2017/09/blog-post.html>
4. Що таке SQLite, для чого і як його використовувати [Електронний ресурс]. Режим доступу: <https://foxminded.ua/prohramuvannia-baz-danykh-na-sqlite/>
5. Erwin Data Modeler [Електронний ресурс]. Режим доступу: [https://en.wikipedia.org/wiki/Erwin\\_Data\\_Modeler](https://en.wikipedia.org/wiki/Erwin_Data_Modeler)
6. Поняття про АБІС і їх роль у розвитку бібліотечних технологій [Електронний ресурс]. Режим доступу: [http://moodle.dk.rv.ua/pluginfile.php/771/mod\\_resource/content/1/%D0%90%D0%91%D0%A2%20%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%202.pdf](http://moodle.dk.rv.ua/pluginfile.php/771/mod_resource/content/1/%D0%90%D0%91%D0%A2%20%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%202.pdf)
7. PyCharm як найкраща IDE для розробки ПО на Python [Електронний ресурс]. Режим доступу: <https://foxminded.ua/ru/pycharm-eto/>
8. Кунанец Н. Особливості впровадження автоматизованої бібліотечної системи АЛЕРН у Львівській національній науковій бібліотеці імені В. Стефаника // Вісн. Нац. Ун-ту “Львів Політехніка”. 2008. №631
9. Юрченко І.В., Сікора В.С. Програмування мовою Python: навчальний посібник.– Чернівці: Чернівецький національний університет, 2022.– 104 с.

10. Іванович, П. Ю. (2014). Бібліотека як потужний інформаційний центр сучасності. *Міжнародна наукова конференція «Місце і роль бібліотек у формуванні національного інформаційного простору»*
11. Овсієнко, А. (2021). Анотація. у *АДАПТАЦІЯ БІБЛІОТЕК ДО РЕАЛІЙ ЦИФРОВОГО ПРОСТОРУ* (с. 255).
12. Самоленко, Я.В. (2020). Структура АБІС. *Поняття про АБІС і їх роль у розвитку бібліотечних технологій*, с. 5.
13. Безкоштовні АБІС – вільне відкрите програмне забезпечення [Електронний ресурс]. Режим доступу: <http://nbuv.gov.ua/node/1336>
14. Підручник з Python [Електронний ресурс]. Режим доступу: <https://docs.python.org/uk/3/tutorial/index.html>
15. Python [Електронний ресурс]. Режим доступу: <https://www.wikidata.uk-ua.nina.az/%D0%9F%D0%B0%D0%B9%D1%82%D0%BE%D0%BD.html>
16. Переваги і недоліки мови Python [Електронний ресурс]. Режим доступу: <https://blog.ithillel.ua/articles/perevagi-i-nedoliki-movi-python>
17. tkinter — Інтерфейс Python до Tcl/Tk [Електронний ресурс]. Режим доступу: <https://docs.python.org/uk/3/library/tkinter.html>

## ДОДАТКИ

### Код Розробки веб-додатку автоматизованої бібліотечної інформаційної системи

```
import sqlite3

from tkinter import *

from tkinter.ttk import *

# Sample user data (replace with your actual user data)

user_data = [

    {'username': 'admin', 'password': 'admin', 'status': 'admin'},

    {'username': 'user', 'password': 'user', 'status': 'custom'}

]

# Function to connect to the database

def connect_db():

    conn = sqlite3.connect('library.db')

    cursor = conn.cursor()

    cursor.execute('''CREATE TABLE IF NOT EXISTS books

                    (id INTEGER PRIMARY KEY,

                     title TEXT,

                     author TEXT,

                     genre TEXT,

                     status TEXT)''')

    conn.commit()

    return conn
```

```
# Function to log in

def login(username, password):

    if username == 'admin' and password == 'adminpass':

        return 'admin'

    elif username == 'user' and password == 'userpass':

        return 'user'

    else:

        return None

# Function to add a book

def add_book(title, author, genre):

    conn = connect_db()

    cursor = conn.cursor()

    cursor.execute(''INSERT INTO books (title, author, genre,
status)

                                VALUES (?, ?, ?, ?)'', (title, author, genre,
'Available'))

    conn.commit()

    conn.close()

# Function to view books

def view_books():

    conn = connect_db()

    cursor = conn.cursor()

    cursor.execute(''SELECT * FROM books'')

    books = cursor.fetchall()

    conn.close()

    return books
```



```

# Function to add a book to planned or read list

def update_status(book_id, status):

    conn = connect_db()

    cursor = conn.cursor()

    cursor.execute(''UPDATE books SET status=? WHERE id='',
(status, book_id))

    conn.commit()

    conn.close()

# Function to search books by genre or title

def search_books(keyword, search_by):

    conn = connect_db()

    cursor = conn.cursor()

    if search_by == 'Genre':

        cursor.execute(''SELECT * FROM books WHERE genre=?'',
(keyword,))

    elif search_by == 'Title':

        cursor.execute(''SELECT * FROM books WHERE title LIKE ?'',
('%' + keyword + '%',))

    books = cursor.fetchall()

    conn.close()

    return books

# GUI

def main():

    def login_button_click():

        username = username_entry.get()

```

```
password = password_entry.get()
role = login(username, password)
if role == 'admin':
    admin_menu()
elif role == 'user':
    user_menu()
else:
    error_label.config(text='Неправильний пароль або користувач')
```

```
def add_book_button_click():
    # Add book to the treeview
    title = title_entry.get()
    author = author_entry.get()
    genre = genre_entry.get()
    add_book(title, author, genre)
    success_label.config(text='Книгу успішно додано')
    update_books_table()

    # Clear text entries
    title_entry.delete(0, END)
    author_entry.delete(0, END)
    genre_entry.delete(0, END)

def update_books_table():
    books = view_books()
    for i in book_table.get_children():
        book_table.delete(i)
```

```
        for book in books:
            book_table.insert('', 'end', values=(book[0], book[1],
book[2], book[3], book[4]))
```

```
def search_button_click():
    keyword = search_entry.get()
    search_by = search_by_var.get()
    books = search_books(keyword, search_by)
    search_listbox.delete(*search_listbox.get_children())
    for book in books:
        search_listbox.insert('', 'end', values=(book[0],
book[1], book[2], book[3], book[4]))
```

```
def admin_menu():
    login_frame.pack_forget()
    user_frame.pack_forget()
    admin_frame.pack()
```

```
def user_menu():
    login_frame.pack_forget()
    admin_frame.pack_forget()
    user_frame.pack()
```

```
def logout():
    admin_frame.pack_forget()
    user_frame.pack_forget()
    login_frame.pack()
```

```
root = Tk()

root.title('Бібліотечна система')

root.geometry("600x100")

# Login Frame

login_frame = Frame(root)

login_frame.pack()

username_label = Label(login_frame, text='Користувач:')

username_label.grid(row=0, column=0)

username_entry = Entry(login_frame)

username_entry.grid(row=0, column=1)

password_label = Label(login_frame, text='Пароль:')

password_label.grid(row=1, column=0)

password_entry = Entry(login_frame, show='*')

password_entry.grid(row=1, column=1)

login_button = Button(login_frame, text='Увійти',
command=login_button_click)

login_button.grid(row=2, columnspan=2)

error_label = Label(login_frame, text='')

error_label.grid(row=3, columnspan=2)

# Admin Frame

admin_frame = Frame(root)
```

```
title_label = Label(admin_frame, text='Назва книги:')
title_label.grid(row=0, column=0)

title_entry = Entry(admin_frame)
title_entry.grid(row=0, column=1)

author_label = Label(admin_frame, text='Автор:')
author_label.grid(row=1, column=0)

author_entry = Entry(admin_frame)
author_entry.grid(row=1, column=1)

genre_label = Label(admin_frame, text='Жанр:')
genre_label.grid(row=2, column=0)

genre_entry = Entry(admin_frame)
genre_entry.grid(row=2, column=1)

add_book_button = Button(admin_frame, text='Добавити книгу',
command=add_book_button_click)

add_book_button.grid(row=3, columnspan=2)

success_label = Label(admin_frame, text='')
success_label.grid(row=4, columnspan=2)

book_table = Treeview(admin_frame, columns=('ID', 'Title',
'Author', 'Genre', 'Status'), show='headings')

book_table.heading('ID', text='ID')
book_table.heading('Title', text='Назва')
book_table.heading('Author', text='Автор')
book_table.heading('Genre', text='Жанр')
```

```
book_table.heading('Status', text='Статус')

book_table.grid(row=5, columnspan=2)

update_books_table()

exit_button = Button(admin_frame, text='Вихід', command=logout)
exit_button.grid(row=6, columnspan=2)

admin_frame.pack_forget()

# User Frame

user_frame = Frame(root)

view_books_button = Button(user_frame, text='Переглянути книгу',
command=update_books_table)

view_books_button.pack()

book_table = Treeview(user_frame, columns=('ID', 'Title',
'Author', 'Genre', 'Status'), show='headings')

book_table.heading('ID', text='ID')

book_table.heading('Title', text='Назва')

book_table.heading('Author', text='Автор')

book_table.heading('Genre', text='Жанр')

book_table.heading('Status', text='Статус')

book_table.pack()

search_label = Label(user_frame, text='Знайти:')

search_label.pack()
```

```
search_entry = Entry(user_frame)

search_entry.pack()

search_by_var = StringVar(user_frame)

search_by_var.set('Title')

search_by_menu = OptionMenu(user_frame, search_by_var, 'Title',
                              'Genre')

search_by_menu.pack()

search_button = Button(user_frame, text='Знайти',
                        command=search_button_click)

search_button.pack()

search_listbox = Treeview(user_frame, columns=('ID', 'Title',
                                                'Author', 'Genre', 'Status'), show='headings')

search_listbox.heading('ID', text='ID')

search_listbox.heading('Title', text='Назва')

search_listbox.heading('Author', text='Автор')

search_listbox.heading('Genre', text='Жанр')

search_listbox.heading('Status', text='Статус')

search_listbox.pack()

exit_button = Button(user_frame, text='Вихід', command=logout)

exit_button.pack()

user_frame.pack_forget()
```

```
root.mainloop()

if __name__ == "__main__":
    main()

import sqlite3
```