

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет  
імені Петра Могили  
Факультет комп'ютерних наук  
Кафедра інтелектуальних інформаційних систем**

**ДОПУЩЕНО ДО ЗАХИСТУ**

Завідувач кафедри інтелектуальних  
інформаційних систем, д-р техн. наук,  
проф.

Ю. П. Кондратенко

« \_\_\_\_ » \_\_\_\_\_ 2022 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

**АНАЛІТИЧНА СИСТЕМА ДЛЯ МОНІТОРИНГУ І  
КОНТРОЛЮ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ  
БІБЛІОТЕКИ**

Спеціальність 124 «Системний аналіз»

**124 – МКР – 607.21610208**

Студентка \_\_\_\_\_ А. В. Кирилова

«17» лютого 2022 р.

Консультант \_\_\_\_\_ І. В. Кулаковська  
канд. фіз.-мат. наук, доцент

«17» лютого 2022 р.

**Миколаїв – 2022**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	3
ВСТУП .....	5
1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ .....	7
1.1 Аналіз діяльності шкільної бібліотки.....	7
1.2 Огляд аналогів застосунку.....	9
1.3 Технічне завдання.....	18
Висновки до розділу 1 .....	18
2 ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПРОЕКТУВАННЯ І СТВОРЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ .....	19
2.1 Поняття CRM-системи.....	19
2.2 Класифікація CRM-систем .....	21
2.3 Принципи роботи Web-додатків на основі ASP.NET .....	25
2.4 Огляд засобів роботи з веб-застосунками.....	27
Висновки до розділу 2 .....	35
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ .....	36
3.1 Логічна структура сайту .....	36
3.2 Проектування діаграм на мові uml .....	38
3.3 Проектування бази даних .....	41
3.4 Панелі адміністратора та бібліотекаря.....	44
3.5 Панель користувача.....	48
Висновки до розділу 3 .....	50
4 МЕТОДИЧНА ЧАСТИНА.....	<b>Ошибка! Закладка не определена.</b>
5 ОХОРОНА ПРАЦІ НА РОБОЧОМУ МІСЦІ АДМІНІСТРАТОРА БІБЛІОТЕКИ.....	<b>Ошибка! Закладка не определена.</b>
ВИСНОВКИ.....	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	52
ДОДАТОК А Лістинг коду .....	55

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ**

БД – база даних.

ПЗ – програмне забезпечення.

СУБД – система управління базами даних.

API – Application Programming Interface.

CRM – Customer relationship management.

IIS – Internet Information Services.

MVC – Model–view–controller.

# **Пояснювальна записка**

**до магістерської кваліфікаційної роботи**

на тему:

## **«АНАЛІТИЧНА СИСТЕМА ДЛЯ МОНІТОРИНГУ І КОНТРОЛЮ ВЗАЄМОДІЇ З КОРИСТУВАЧАМИ БІБЛІОТЕКИ»**

Спеціальність 124 «Системний аналіз»

**124 – МКР – 607.21610208**

Студентка \_\_\_\_\_ А. В. Кирилова

«16» лютого 2022 р.

Консультант \_\_\_\_\_ І. В. Кулаковська  
канд. фіз.-мат. наук, доцент

«16» лютого 2022 р.

**Миколаїв – 2022**

## ВСТУП

У наш час важливим завданням інформаційних технологій є автоматизація взаємодії з людиною для отримання максимального прибутку. Одним з основних інструментів для цього є CRM-системи. CRM-система враховує індивідуальні характеристики окремих клієнтів, що дозволяє на гідному рівні взаємодіяти як із постійними, так і з новими партнерами. Використання такої технології в аспекті управління підприємством позитивно впливає на рівень конкурентоспроможності. Також CRM-системи можна використовувати в некомерційних організаціях, наприклад бібліотеки, з метою покращення якості обслуговування, створення інформаційної бази та спрощення роботи персоналу. Використання корпоративної інформаційно-аналітичної системи допомагає підвищити ефективність роботи бібліотеки, упорядковує всі робочі процеси.

Темою магістерської роботи є «Аналітична система для моніторингу і контролю взаємодії з користувачами бібліотеки».

Метою є дослідження аналітичної системи для моніторингу і контролю взаємодії з користувачами бібліотеки шляхом створення веб-сайту бібліотеки з інформаційно-довідковою системою, призначеною для того щоб тримати під контролем ситуацію з заявками, зверненнями і етапами роботи учнів з книгами.

Тема є актуальною оскільки сучасна школа потребує оновлення бібліотечного простору з новими бібліотечними послугами, щоб створити такий простір потрібно прикласти немало зусиль і зробити абсолютне перезавантаження шкільної бібліотеки щоб підготувати її до нових потреб читачів. Для цього потрібно розробити систему управління взаємодії з учнями призначена якої виконання наступних завдань: консолідація даних клієнтів; відстеження взаємодій і активності.

Запланована система має виконувати такі функції:

- забезпечувати авторизацію та розмежування доступу;

- ведення картки виданих книг;
- статистичний аналіз популярності книг, активності співробітників та клієнтів тощо;
- список книг бібліотеки;
- пошук за базою даних;
- формування звітів з персоналу та відвідувачів;
- зворотній зв'язок з користувачами.

Об'єктом дослідження є аналітична система для взаємодії з користувачами бібліотеки.

Предметом дослідження є засоби та методи створення веб-сайтів моніторингу і контролю та реалізації аналітичних систем бібліотеки.

Для реалізації проекту необхідно виконати наступні завдання:

- розглянути теоретичні аспекти створення Web-сайтів;
- провести аналіз існуючих аналогів по темі МКР – її переваг та недоліків;
- розглянути засоби створення веб-застосунків;
- провести аналіз засобів створення баз даних для Web-сайтів;
- визначити структуру і призначений для користувача інтерфейс сайту;
- провести аналіз працездатності створеного сайту.

У спеціальній частині буде проведено аналіз умов праці адміністратора бібліотки та визначено можливості безпечного виконання роботи без шкідливого впливу на здоров'я людини.

Основні задачі спеціальної частини:

- проаналізувати умови праці адміністратора бібліотки;
- розрахувати оптимальний рівень освітлення на робочому місці;
- розглянути вимоги до мікроклімату;
- розглянути вимоги до обладнання та організації робочого місця;
- визначити основні правила пожежної безпеки.

## **1 АНАЛІЗ ПРЕДМЕТНОЇ СФЕРИ**

### **1.1 Аналіз діяльності шкільної бібліотеки**

Бібліотеки існують тисячі років. З давніх-давен вони були зберігачами духовних багатств суспільства, помічниками всіх, хто ними користувався. Згодом людина змінюється; розвивається наука, техніка; ускладнюються шкільні програми; іншими стають діти, але людство не придумало кращого способу у розвиток свого інтелекту, ніж читання. Але сьогодні книжка перестала бути єдиним джерелом знань.

У сучасному світі дедалі частіше на зміну книгам приходять комп'ютери, електронні та цифрові носії. Книги стають незатребуваними, припадають пилом на полицях у бібліотеках і магазинах. Сучасному батькові простіше натиснути кнопку пульта, ніж прочитати своїй дитині книгу. Адже розвиток дитини відбувається тільки при живому спілкуванні. Наслідком недостатнього спілкування дітей із книгами стають мовні порушення, порушення розумових процесів, слабкий розвиток комунікативних функцій. Все це негативно відбивається на загальному стані дитини та на формуванні її як особистості.

У житті дитини шкільна бібліотека має стати як перший вчитель, як перша прочитана книга, як перший справжній друг. Адже бібліотека – це вісь, навколо якої обертається культурне життя школи та світу.

Сучасна шкільна бібліотека – це освітній та соціальний центр школи; безпечне, правильно організоване, затишне, дружелюбне сучасне середовище для всіх учасників освітнього процесу; найпривабливіше місце у школі.

Мета роботи шкільної бібліотеки – сприяти формуванню мотивації користувачів до саморозвитку, самоосвіти через надання необхідних інформаційних ресурсів, через забезпечення відкритого та повноцінного доступу до інформації.

## Основні функції бібліотеки:

Освітня – підтримувати та забезпечувати освітні цілі, сформульовані у концепції школи та у шкільній програмі.

Інформаційна – надавати можливість використовувати інформацію незалежно від її виду, формату та носія.

Культурна – організовувати заходи, які виховують культурну та соціальну самосвідомість, сприяють емоційному розвитку учнів.

Шкільна бібліотека виконує великий обсяг роботи щодо надання користувачам необхідного інформаційного матеріалу. Стрімкий розвиток нових інформаційних технологій актуалізувало проблему підвищення якості бібліотечно-інформаційного обслуговування учнів та педагогічних працівників, від рівня якого багато в чому залежить досягнення нової якості освіти. Актуальним є вирішення завдання взаємодії школи з бібліотекою. Це підкріплюється педагогічним потенціалом бібліотеки навчального закладу.

Зараз намітилося три провідні підходи, три погляди педагогічної науки та практики на роль шкільних бібліотек та діяльність шкільного бібліотекаря. Суть першого підходу – в обґрунтуванні необхідності створення на базі шкільних бібліотек нового підрозділу – медіатеки, або шкільного бібліотечно-медіацентру, доповнення функції бібліотекаря обов'язками медіаспеціаліста; другий у тому, що бібліотека має стати інформаційним центром школи, а бібліотекаря доведеться взяти він функції інформатора; третій – у розгляді шкільних бібліотек як педагогічного структурного підрозділу освітніх закладів зі специфічними функціями.

І освітній заклад, і бібліотека має на меті єдину мету: сприяти формуванню мотивації людини до саморозвитку, самоосвіти через надання необхідних інформаційних ресурсів, через забезпечення відкритого та повноцінного доступу до інформації.

Тому істотним компонентом функціонування навчальної бібліотеки є погляд неї як на інформаційно-бібліотечне середовище, систематизующее наявні інформаційні ресурси, орієнтовані навчальні потреби, що забезпечує



відкритий доступом до інформації, формує інформаційну культуру всіх учасників освітнього процесу, надає комфортні умови для самостійної роботи учнів.

Під сучасною шкільною бібліотекою ми розуміємо структурний підрозділ загальноосвітнього закладу, який бере участь у навчально-виховному процесі з метою забезпечення права учасників навчального процесу на безоплатне користування бібліотечно-інформаційними ресурсами.

## 1.2 Огляд аналогів застосунку

В цьому розділі буде розглянуто аналогічні додатки. Виявлено їх переваги та недоліки для подальшого використання при створенні наступних проєктів.

### SendPulse CRM

Мультиканальний сервіс, який помічає CRM-системи, дає можливість робити розсилку електронної пошти, створювати чат-боти, розробляти мобільні кредити та використовувати інші інструменти для управління бізнесом (див. рис. 1.1).

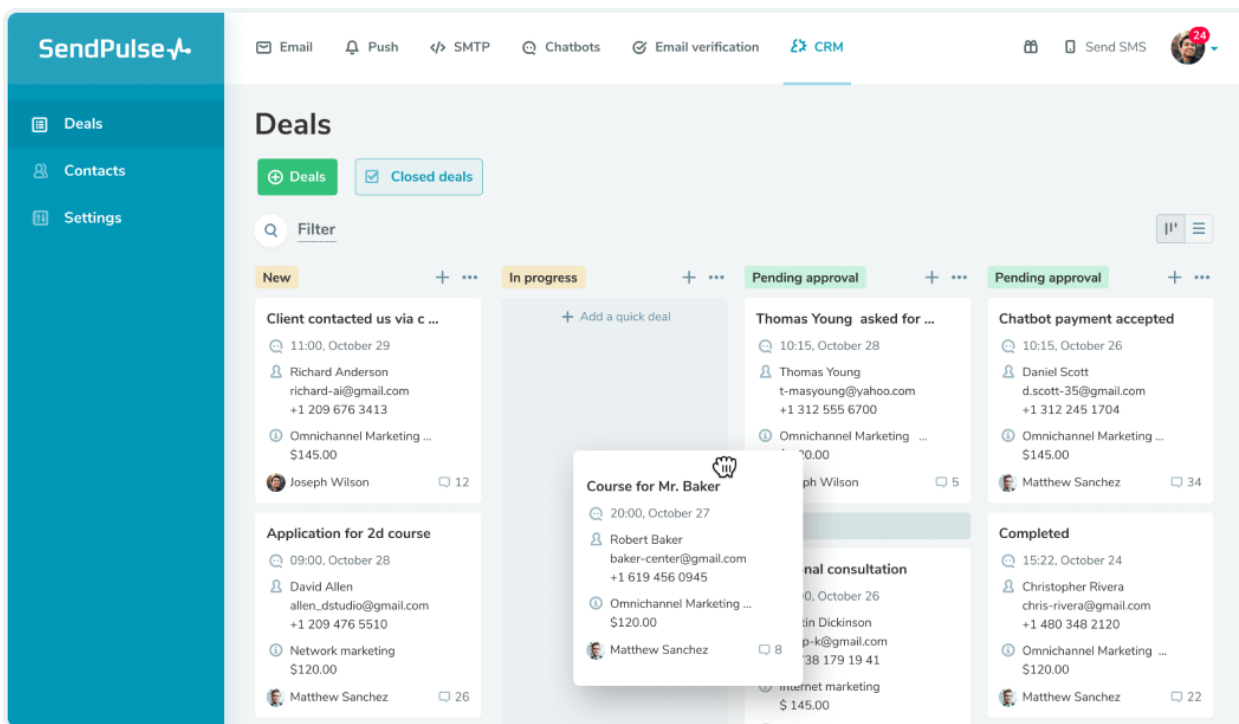


Рис. 1.1. Система «SendPulse CRM»

Угоди можна створювати як автоматично, так і вручну. Усі історії роботи з лідами та покупцями зберігаються у картках клієнта, які легко редагувати. Передбачено фільтрацію угод за різними параметрами, наприклад, сумою, датою створення, статусом, джерелом тощо.

За замовчуванням угоди відображаються у вигляді канбану дошки, однак за необхідності їх можна трансформувати в список (див. рис. 1.2).

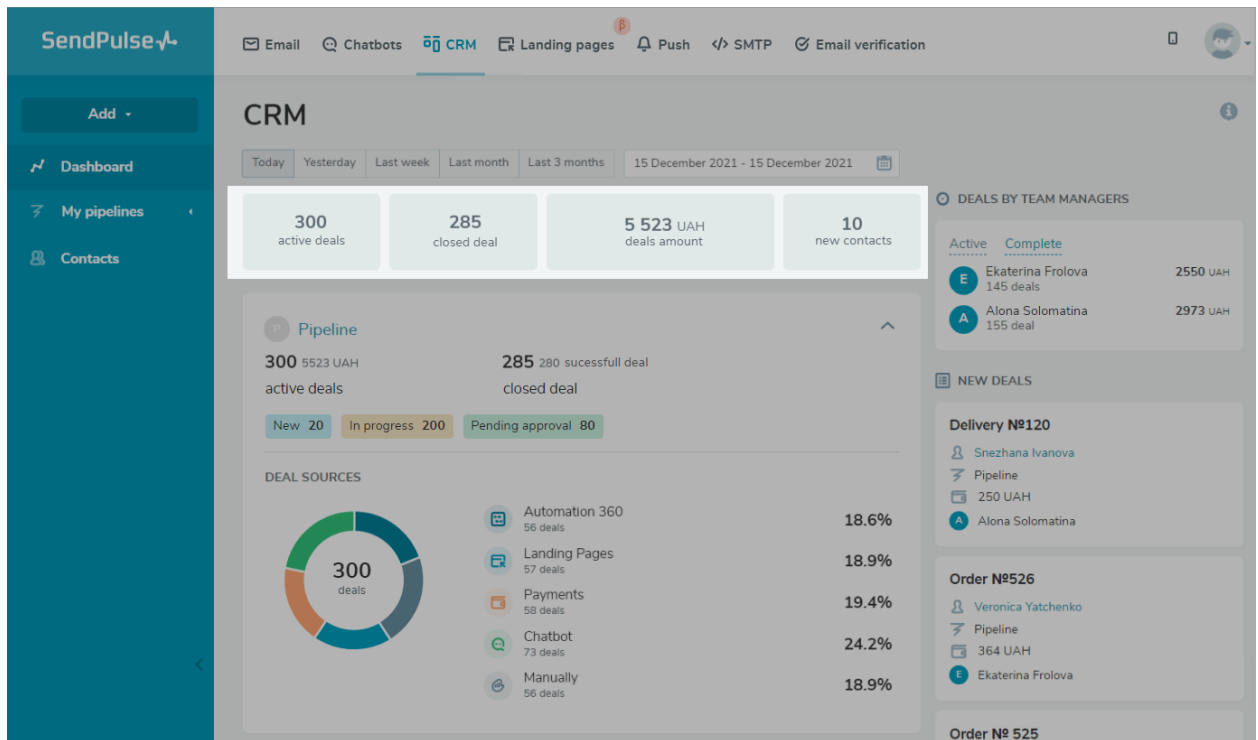


Рис. 1.2. Внутрішній інтерфейс системи керування

Основні функції і особливості:

- управління угодами відбувається по методу Канбан;
- створення справ як вручну, так і автоматично;
- збереження історії взаємодії з клієнтами;
- використання різних каналів комунікацій з клієнтами: електронна пошта, SMS, Viber, WhatsApp, Facebook Messenger, Telegram;
- коментарі в карточках клієнтів;
- використання тегів для сегментації клієнтів;
- фільтрація справ за періодом часу, статусу, суммі, джерелу та іншим параметрам.

**Плюси:**

- простота в використанні, ідеальна для тех, хто тільки починає роботу з CRM;
- можливість налаштування рівня доступу членів команд;
- стабільність роботи системи;
- інтеграції з внутрішніми сервісами SendPulse;
- весь функціональний доступний, починаючи з безкоштовної версії.

**Мінуси:**

- немає дашборда, для одночасного спостереження таких метрик, як кількість нових, завершених успішно та незавершених справ за період часу;
- не можна створювати задачі для співробітників.

### **CRM система AmoCRM**

AmoCRM — система управління взаємовідносинами з клієнтами, що дозволяє автоматизувати продаж та організувати роботу відділу маркетингу всередині невеликого підприємства.

Впровадження AmoCRM дозволяє оптимізувати внутрішні процеси компанії, та покращити якість роботи менеджерів з продажу та керівника відділу завдяки вбудованому інструментарію. Система допомагає організувати відділ продажу. Система дозволяє контролювати роботу менеджерів, фіксувати всі заявки та запити клієнтів, вести облік усіх угод компанії, клієнтів та продажів (див. рис. 1.3).

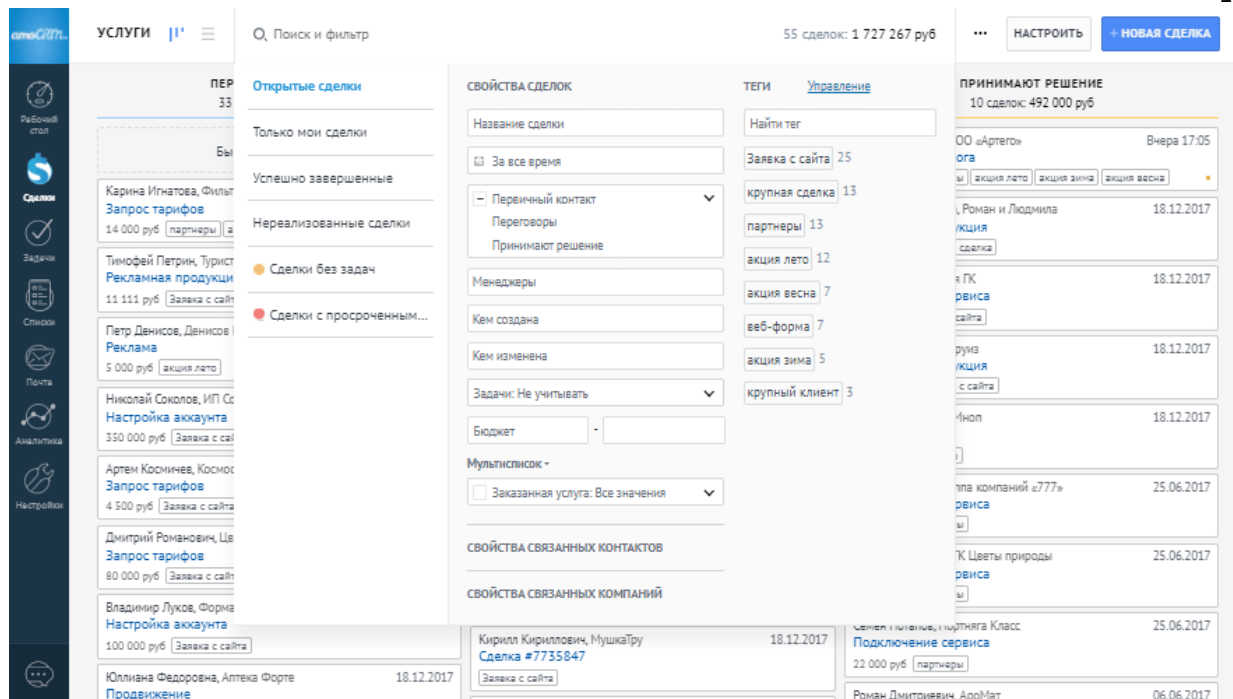


Рис. 1.3. Интерфейс AmoCRM

Интерфейс програми складається з кількох розділів, які відповідають за роботу відділу продажу. Доступ до модуля аналітики, внутрішнього месенджера, угод та завдань здійснюється через головне меню системи – докладно розглянемо основні інструменти сервісу (див. рис. 1.4).

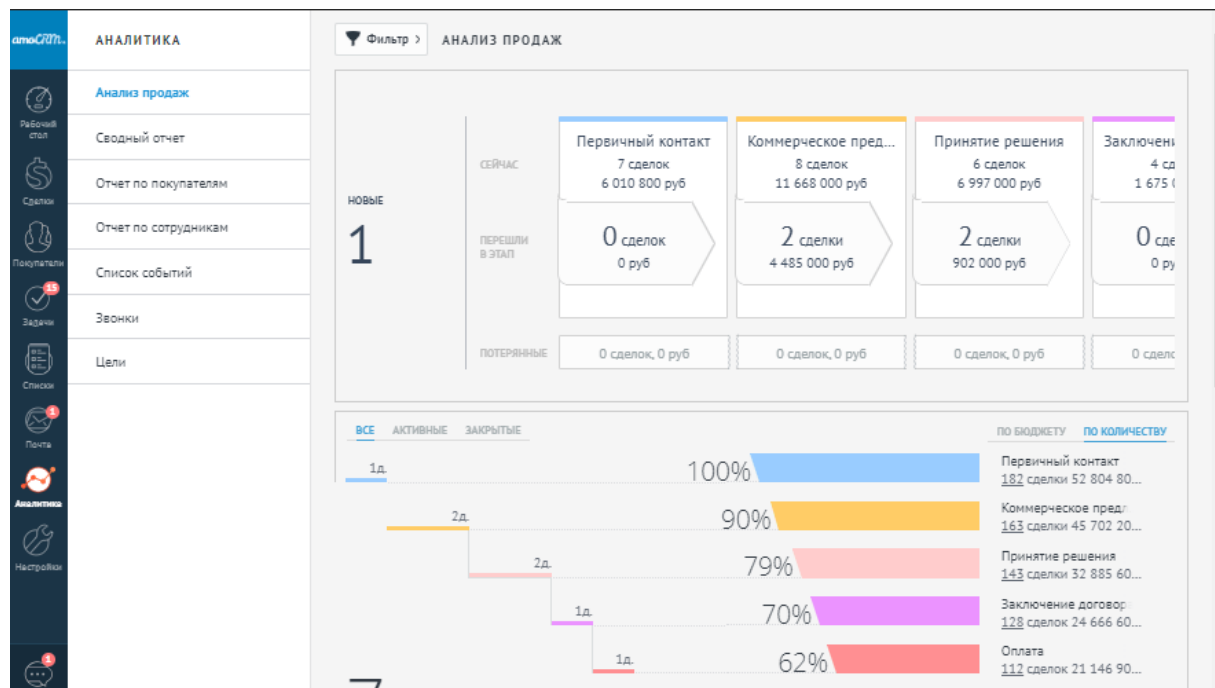


Рис. 1.4. Аналітика AmoCRM

### Основні функції та особливості:

- фіксація звернень від клієнтів із будь-яких каналів;
- мобільний додаток зі сканером візиток, який дозволяє відразу додати контакти;
- зберігання переписки по електронній пошті всередині CRM;
- вихідні дзвінки із системи;
- вбудований месенджер для спілкування з співробітниками;
- автоматична воронка продаж і воронка повторних звернень;
- інтерфейс розробника — можна додати свої функції.

### Плюси:

- інтеграція з більш ніж 50 провайдерами телефонії для офісу;
- налаштування справ з тегами і додатковими параметрами;
- інтеграція з більш ніж 200 сервісами.

### Мінуси:

- іноді API працює некоректно;
- немає широкого функції для роботи з бізнес-процесами;
- техпідтримка може довго реагувати на звернення.

Система управління взаємовідносинами з клієнтами AmoCRM призначена для бізнес-стартапів або організації роботи відділу продажів невеликих за розміром підприємств, що спеціалізуються на торгівлі або працюють у сфері послуг. Простий інструментарій та інтуїтивний зрозумілий інтерфейс дозволяють організувати процес продажу для молодих компаній та розібратися у нюансах e-commerce.

### **CRM система EasyWeek**

Одна з найкращих CRM на ринку з функціями онлайн-записів. Крім стандартних функцій CRM, сервіс надає створюваний міні веб-сайт, який можна підключити до свого домену. Для тих, у кого вже є сайт, CRM надає доступ до онлайн-записів, які можна розмістити на вашому сайті за 5 хвилин і

отримати ліди. Форму збору лідов можна змінювати та доповнювати під свої потреби. Також є повідомлення про повідомлення по SMS, електронній пошті, Telegram тощо (див. рис. 1.5).

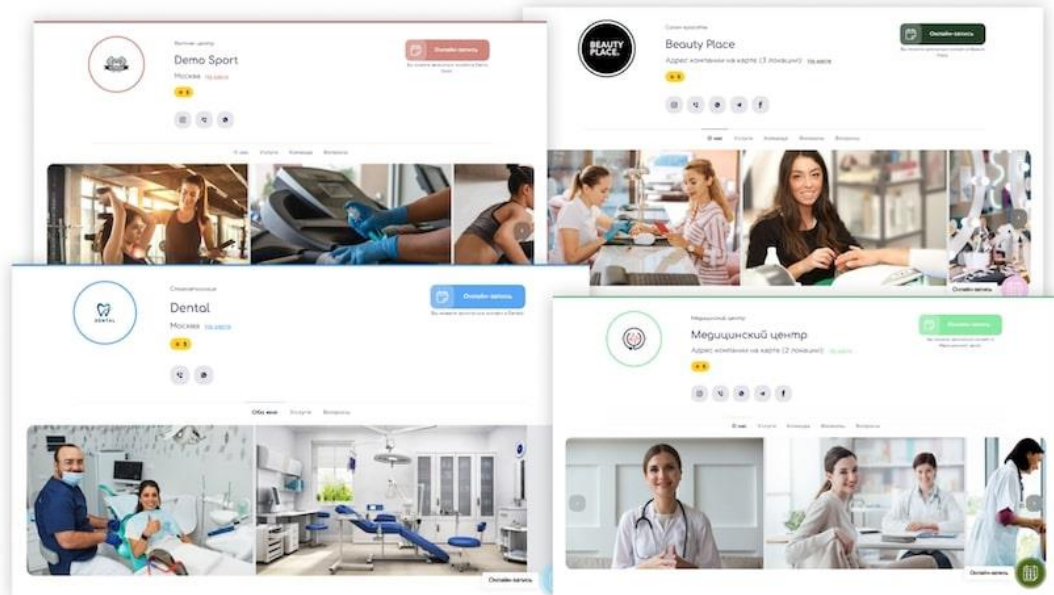


Рис. 1.5. EasyWeek

Сервіс для різних галузей. Найбільш ефективно можна використовувати для: салонів краси та б'юті індустрії, медичних центрів, стоматологій, автошкіл, станцій тех. обслуговування, масажних салонів, лазень та ін.

Можливості:

- онлайн-запис через віджет і сайт;
- електронний календар и журнал;
- клієнтська база;
- веб-сайт, який можна підключити до свого домену;
- установку в соц. медіа;
- статистика і аналітика;
- оповіщення клієнтів;
- програма знижок, лояльності, сертифікатів;
- фінансовий и зарплатний модуль;
- можливість публікації компанії в каталозі EasyWeek;

- мобільні додатки для IOS та Android.

**Плюси:**

- проста у використанні, світлий і зрозумілий інтерфейс (є темна тема);
- швидко працює як система управління, так і віджет записи;
- оперативна техпідтримка.

**Мінуси:**

- нема ланцюжків по роботі з лідами;
- нема воронки продажів.

Таблиця 1.1

**Порівняльна характеристика**

<b>Назва сервісу/функціонал</b>	<b>AmoCRM</b>	<b>EasyWeek</b>	<b>SendPulse</b>
Робота з угодою/проектом	+	+	+
Телефонія	+	+	+
Чат та відео	Чат	Чат и відео	Чат
Календарі	+	+	-
Жива стрічка	-	+	+

Продовження таблиці 1.1

Управління завданнями	Контроль термінів, делегування, звіти, облік часу за завданнями, шаблони завдань, чек-листи, зайнятість працівників	Контроль термінів, делегування, звіти, облік часу за завданнями, шаблони завдань, чек-листи, зайнятість працівників	Контроль термінів, делегування, облік часу, коментування задач
Оповіднення	+	+	+
Бізнес процеси	Налаштування оплати рахунків, заяв на відпустку та відрядження, облік вихідних документів, отримання подрахункових авансів та інші завдання.	Налаштування оплати рахунків, заяв на відпустку та відрядження, облік вихідних документів, отримання подрахункових авансів та інші завдання	Шаблони бізнес-процесів, облік документів, налаштування оплати рахунків, інші завдання



Закінчення таблиці 1.1

Завдання	Ставити завдання, коментувати, делегувати	Ставити завдання, коментувати, лайкати; створювати опис, встановлювати статус, призначати відповідального, відстежувати	Ставити завдання, створювати опис, встановлювати статус, призначати відповідального
E-mail-клієнт	+	+	+
Налаштування інтерфейсу	+	+	Частково
Різні права доступу	+	+	+
Звітність	Рахунки, авансові звіти	Рахунки, авансові звіти	Рахунки, авансові звіти, конструктор звітів, угоди
Воронка продажів	+	+	+
Спільна робота з документами	+	+	+
Інтеграція із соціальними мережами	+	+	+

### **1.3 Технічне завдання**

Головною задачею, є розробка онлайн платформи (сайту) бібліотки, оснащеною системою бронювання.

Мета дипломної роботи полягає у створенні інформаційної системи, що буде простою і зручною у використанні та мати приємний дизайн інтерфейсу.

Реалізація такої теми має за мету провести дослідження наявних, поширених і функціонуючих CRM-систем, відмічаючи їх досвід та найкращі аспекти. Аналітична система для моніторингу і контролю взаємодії з користувачами потрібна для вибудовування грамотного діалогу з читачами: вона дозволяє не допускати помилок і, як наслідок, учень буде задоволений обслуговуванням.

Опрацювавши переліки переваг і недоліків, необхідно розробити інформаційну підсистему автоматизації робочого місця працівника бібліотеки, а також провести аналіз уже реалізованої системи.

#### **Висновки до розділу 1**

Проаналізувавши предметну сферу і оцінивши актуальність вибраної теми, можна зробити висновок що в наші дні веб застосунки використовуються всюди, ІТ-технології дають можливість кожній компанії, установі або окремій людині формувати інформаційний простір між людьми.

Створення якісної системи – це складний процес, що вимагає використання сучасних методів розробки для забезпечення зручної та доступної навігації.

## **2 ТЕХНОЛОГІЇ ТА ПІДХОДИ ДО ПРОЕКТУВАННЯ І СТВОРЕННЯ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ**

### **2.1 Поняття CRM-системи**

Система управління взаємодією з клієнтами (Customer Relationship Management System, CRM-система) – корпоративна інформаційна система, призначена для автоматизації CRM-стратегії компанії, зокрема, для підвищення рівня продажів, оптимізації маркетингу і поліпшення обслуговування клієнтів шляхом збереження інформації про клієнтів (контрагентів) і історії взаємин з ними, встановлення і поліпшення бізнес-процедур і подальшого аналізу результатів. Її основні принципи наступні.

1. Наявність єдиного сховища інформації, що забезпечить доступ до всіх відомостей про попередню і заплановану взаємодію з клієнтами в будь-який момент.
2. Постійний аналіз зібраної інформації про клієнтів і підготовка даних для прийняття відповідних організаційних рішень - наприклад, сегментація клієнтів на основі їх значущості для компанії.

Цей підхід має на увазі, що при будь-якій взаємодії з клієнтом по будь-якому каналу, співробітникам компанії доступна повна інформація про всі взаємодії з цим клієнтом і рішення приймається на основі цієї інформації (див. рис. 2.1).

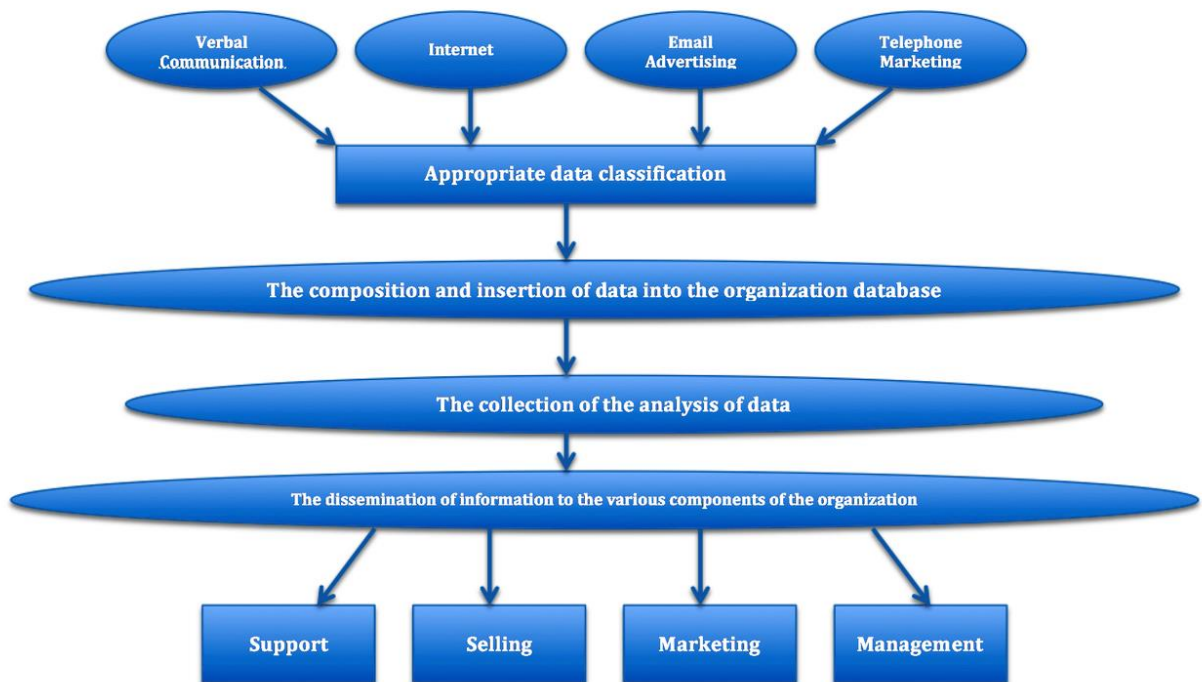


Рис. 2.1. Типи компонентів CRM-систем

Переваги. CRM-система враховує індивідуальні характеристики окремих клієнтів, що дозволяє на гідному рівні взаємодіяти як з постійними, так і новими партнерами. Основними перевагами є такі:

- перелік інструментів управління компанією в одній програмі. CRM дозволяють управляти всіма складовими бізнесу – проектами, персоналом, обліком клієнтів, комунікаціями, фінансами, документообігом, часом, сайтом, розсилками тощо;
- всі дані – в одному місці. CRM може зберігати всі дані по проектам, клієнтам, контактам в одному місці, тому вся інформація за всі роки завжди доступна для співробітників. При цьому CRM мають високий рівень захисту даних, а також мають опції для розмежування доступу до них, що запобігає витоку інформації;
- ефективна оптимізація роботи персоналу. Робота через CRM дозволяє побачити, хто із співробітників чим зайнятий, яке у кого навантаження, і наскільки ефективний кожен працівник, що дає

можливість розподіляти роботу найбільш оптимальним чином, а також автоматизувати контроль термінів;

- економія коштів. CRM мають широкий набір комунікацій, що дозволяє економити десятки тисяч в рік на оплату телефонних дзвінків;
- простота впровадження. При цьому працювати з CRM можна відразу, а співробітники легко освоюють програму в процесі роботи через інтуїтивно зрозумілий інтерфейс.

Недоліки. Крім цілого списку переваг у CRM є кілька мінусів. Зазвичай недоліки носять тимчасовий характер, вони пов'язані з додатковими зусиллями і витратами на впровадження і засвоєння нової системи.

- щоб забезпечити ефективну роботу, компанії доведеться залучити додаткового фахівця, який виконуватиме її обслуговування. Крім того, потрібно подбати про зберігання даних, що теж може збільшити витрати;
- незважаючи на те, що CRM автоматизує різні процеси всередині компанії, важливо, щоб людський фактор певною мірою залишався;
- купівля ПО – це лише частина процесу, пов'язаного з удосконаленням продажів і маркетингу. Багато хто забуває про необхідні витрати на поточну підтримку.

CRM-системи дозволяють відстежувати комунікацію з клієнтами в чатах з менеджерами, по телефону, email і іншим каналам; вимірювання продуктивності і продуктивності; автоматизація рутинних процесів.

## **2.2 Класифікація CRM-систем**

Види CRM-системи, що це, для чого використовується кожна, які особливості в залежності від типу – відповіді на ці та інші питання цікавлять керівників, які обирають програму управління бізнесом. Зазвичай користувачі орієнтуються на назву CRM-системи, вибираючи лідера того чи іншого рейтингу, а також область, для якої розроблена програма. Але щоб зробити

правильний вибір, потрібно розуміти, які бувають CRM-системи та які існують відмінності між ними. Тому докладно розглянемо всі види CRM-програм, представлених на ринку (див. рис. 2.2).

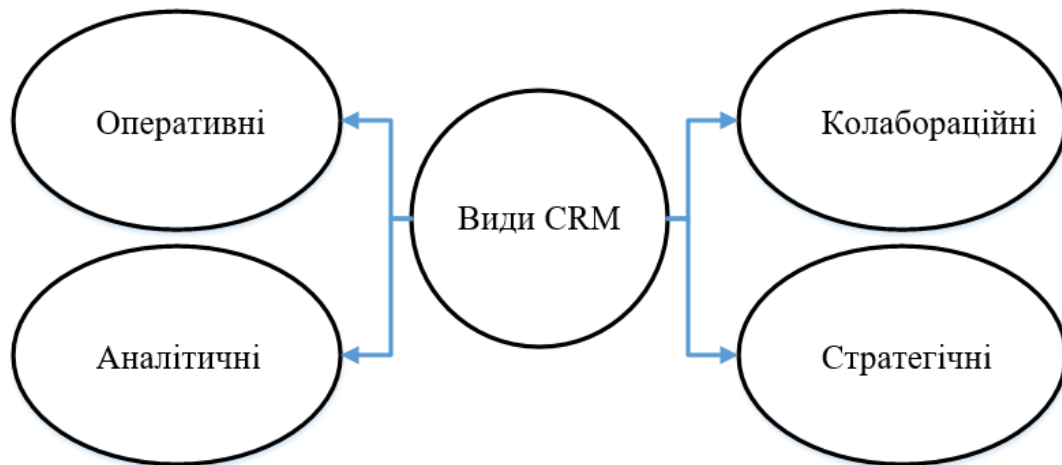


Рис. 2.2. Види CRM-систем

### **Оперативні**

Говорячи про CRM і які бувають їх типи, варто почати з операційних. У цей вид включені звичні більшості програми управління продажами і персоналом. Будь-яка CRM платформа такого типу розроблена для виконання широкого спектра завдань, завдяки чому вони використовуються у різних відділах та компаніях.

Характеристика CRM-систем операційного типу включає такі моменти:

- робота з базою клієнтів та контрагентів. Дані по кожній контактній особі зберігаються в CRM та активно використовуються у роботі;
- автоматизація системи CRM. Кожна програма такого типу виключає зі списку щоденних справ дрібні завдання – починаючи від деяких телефонних дзвінків та заповнення документів та закінчуючи складанням звітів;

- маркетингові настройки. Це може бути розсилка електронною поштою або аналіз вирви продажів. Майже всі сучасні CRM-системи операційного типу передбачають такий функціонал;
- інтеграція з CMS сайтів, поштою чи IP-телефонією. Оскільки кожна програма такого типу – це система управління клієнтами, CRM, яку використовують у комерційних компаніях, уявити без такого функціоналу неможливо.

Завдяки набору опцій програми такого виду активно використовуються в компаніях, де тією чи іншою мірою є продаж товарів або послуг, – CRM-система для управління взаємовідносинами з клієнтами незамінна.

### **Аналітичні**

Розглядаючи інші типи CRM-систем, не можна не згадати і менш універсальні програми, наприклад, аналітичні. Як випливає з назви, вони використовуються для виконання конкретних завдань - це аналіз даних. Тому, як і інші спеціалізовані програмні продукти, CRM-системи такого виду використовуються у роботі окремих відділів чи вузькопрофільних компаній.

Так, всі CRM-системи аналітичного типу розробляються з урахуванням інтересів відділів маркетингу, які регулярно аналізують ефективність продажів у різних напрямках, від поведінки покупців до вибору каналів просування. Специфіка неминуче далася знаки і на наборі інструментів, якими оснащені такі програми (див. рис. 2.3).

## Можливості аналітичної CRM



Рис. 2.3. Види CRM-систем

### **Колабораційні**

Будь-яка класифікація CRM-систем включає програми спільної роботи різних компаній чи віддалених підрозділів однієї організації, тому такі платформи поширені у сегменті B2B. Колабораційна CRM буде корисною, якщо в одному проекті задіяно кілька підрядників — це допомагає вести всі процеси в одному програмному середовищі, що гарантує прозорість під час виконання завдань та виключає конфлікти на технологічному рівні.

### **Стратегічні**

Поділяючи всі існуючі CRM-системи на типи, не можна не згадати про такий вид, як стратегічні. Найчастіше класифікація CRM обходиться без згадки платформ з цього сегмента — багато хто воліє їх поєднувати з вищезазначеними типами.

Які CRM-системи за функціоналом можна зарахувати до стратегічних? Ті, що орієнтовані працювати з клієнтами. Такі платформи розроблені для активного збору даних про замовників та ретельного аналізу отриманих відомостей. Тому такі програми найчастіше поєднують функціонал CRM різних типів, запозичуючи окремі інструменти для виконання конкретних



завдань з вибудовування бізнесу, орієнтованого на запити клієнтів. Це свого роду симбіоз рішень із платформ оперативного та аналітичного типу.

Використання CRM такого виду допомагає підвищити ефективність продажу, зробити правильні акценти на товарах та послугах, а також сформуванати акцентовані комерційні пропозиції.

### 2.3 Принципи роботи Web-додатків на основі ASP.NET

Web-додатки є особливий тип програм, побудованих з архітектури «клієнт-сервер». Особливість їх полягає в тому, що сама Web-програма знаходиться і виконується на сервері. Клієнт при цьому отримує лише результати роботи. Робота програми ґрунтується на отриманні запитів від користувача (клієнта), їх обробці та видачі результату [1]. Передача запитів та результатів їх обробки відбувається через Інтернет (рис. 2.4).

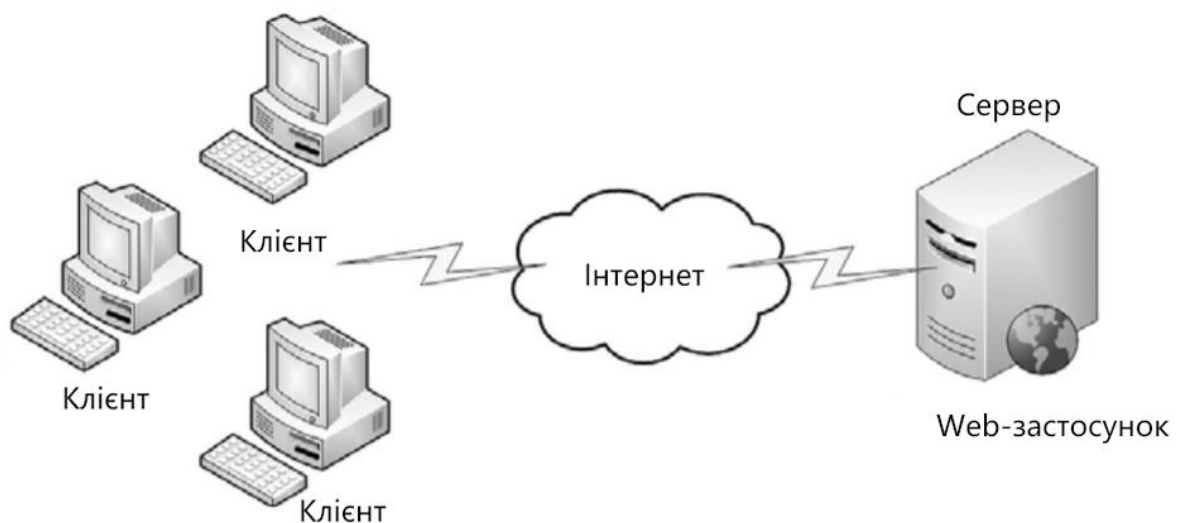


Рис. 2.4. Архітектура Web-додатку

Відображення результатів запитів, а також прийом даних від клієнта та його передачею на сервер зазвичай займається спеціальний додаток – браузер (Internet Explorer, Mozilla, Opera тощо. буд.). Як відомо, однією з функцій браузера є відображення даних, отриманих з Інтернету, у вигляді сторінки, описаної мовою HTML, отже, результат, переданий сервером клієнту, має бути представлений цією мовою [2]. На стороні сервера Web-додаток виконується

спеціальним програмним забезпеченням (Web-сервером), який приймає запити клієнтів, обробляє їх, формує у вигляді сторінки, описаної мовою HTML, і передає його клієнту. Одним з таких Web-серверів є Internet Information Services (IIS) компанії Microsoft. Це єдиний Web-сервер, який здатний виконувати Web-програми, створені з використанням технології ASP.NET. У процесі обробки запиту користувача Web-програма компонує відповідь на основі виконання програмного коду, що працює на стороні сервера, Web-форми, сторінки HTML, іншого вмісту, включаючи графічні файли. В результаті, як уже було сказано, формується HTML-сторінка, яка і надсилається клієнту. Виходить, що результат роботи Web-програми ідентичний результату запиту до традиційного Web-сайту, однак, на відміну від нього, Web-додаток генерує HTML-код залежно від запиту користувача, а не просто передає його клієнту у тому вигляді, в якому цей код зберігається в файл на стороні сервера. Тобто Web-додаток динамічно формує відповідь за допомогою виконуваного коду – так званої частини, що виконується. За рахунок наявності частини, що виконується, Web-додатки здатні виконувати практично ті ж операції, що і звичайні Windows-додатки, з тим лише обмеженням, що код виконується на сервері, як інтерфейс системи виступає браузер, а як середовище, за допомогою якої відбувається обмін даними, - Інтернет. До найбільш типових операцій, що виконуються Web-додатками, відносяться:

- прийом даних від користувача та збереження їх на сервері;
- виконання різних дій на запит користувача: вилучення даних з бази даних (БД), додавання, видалення, зміна даних у БД, проведення складних обчислень;
- автентифікація користувача та відображення інтерфейсу
- системи, що відповідає даному користувачеві;
- відображення оперативної інформації, що постійно змінюється, тощо.

## 2.4 Огляд засобів роботи з веб-застосунками

### 2.4.1 Вибір фреймворку для розробки серверної частини

Фреймворки дозволяють будувати програми швидко, просто та ефективно, а також використовувати готові фрагменти коду, позбавляючи необхідності писати його з нуля. Їх вибір із усього різноманіття залежить від потреб конкретного проекту.

#### Веб-платформа Spring MVC-Java

Через 10 років після першого випуску середа Spring MVC все ще випереджає конкурентів. Згодом він перетворився на повну структуру MVC і перетворився на всеосяжну структуру Java. Як багаторівнева структура J2EE / Java, що об'єднує спеціальну науку про додатки, вона надає розробникам програмного забезпечення повний набір інструментів для налаштування веб-додатків, розробки додатків та проектів безпеки (рис. 2.5).

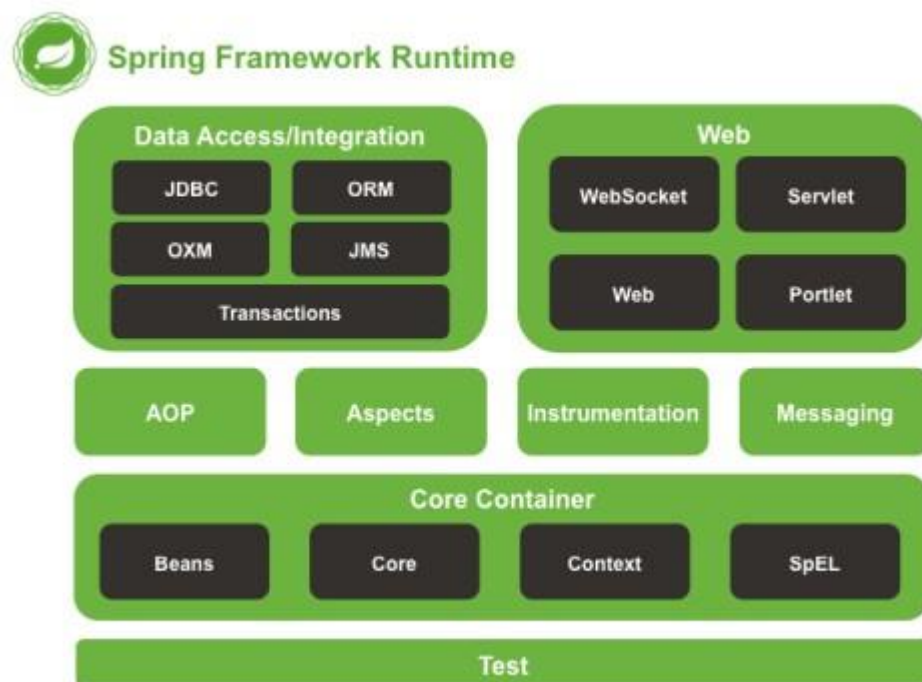


Рис. 2.5. Структура Spring MVC

Фреймворк відомий своєю розвинуеною екосистемою, яка надає безліч додаткових опцій, від веб-служб SOAP, REST API до безпечного шифрування та авторизації.

#### Переваги:

- легкий контейнер із підвищеною модульністю;
- може бути розгорнутий на різних платформах;
- гнучке використання застосування залежностей;
- тристороння конфігурація, включаючи конфігурацію XML, Java та анотацій;
- полегшені дані випробувань упорскування через POJO.

#### Недоліки:

- крива навчання крута;
- початківцям складно реалізувати;
- на розгортання потрібен час.

#### **Веб-платформа JSF-Java**

Oracle вважає, що JSF – це практичне веб-середовище Java, призначене для спрощення інтегрованої розробки веб-інтерфейсу. Хоча ця структура не дуже хороша для розробки Java, документація, надана Oracle, проста у використанні. Це стандартизована технологія з багатими інструментами та бібліотеками, які можуть відкрити вам шлях у дуже складних програмах. Специфікація JSF дозволяє повторно використовувати і розширювати стандартні компоненти інтерфейсу користувача. Це зменшує необхідність створення та підтримки програм Java для запуску та рендерингу програм користувальницького інтерфейсу на цільовому клієнті. Мало того, він також забезпечує просту передачу даних і повторно використовувані компоненти інтерфейсу користувача.

#### Переваги:

- оскільки він може створювати повторно використовувані компоненти, він ефективніший і узгоджений;
- відмінні постачальники, що підтримують продаж JSF, такі як EJB (IBM, Oracle, JBoss і т.д.);

- готові до використання компоненти, такі як Apache, Infragistics, Richfaces тощо;
- зручно для розробників, особливо для тих, хто використовує IDE.

Недоліки JSF:

- крута крива навчання;
- не розглядав інструменти закладок;
- не розширюється;
- підтримка Raw Ajax;
- відсутність гнучкості.

### **Веб-фреймворк Grails-Java**

Grails – це веб-фреймворк Java з відкритим вихідним кодом, призначений для порівняння продуктивності розробників зі стандартними значеннями та API-інтерфейсами за угодою та конфігурацією. Він інтегрований з Java (JVM) та надає потужні функції, включаючи метапрограмування під час компіляції, асинхронне програмування та предметно-орієнтовані мови. Grails використовує потужну технологію перегляду, інтегровану підтримку NoSQL/ORM, плагіни та впровадження залежностей, що підтримуються Spring для оновлень. Дозволяє легко інтегрувати та взаємодіяти з контейнерами Java, Java EE та JVM.

Перевага:

- grails забезпечує швидкий цикл розробки;
- ідеально підходить для малих та середніх проектів;
- різні плагіни на вибір;
- проста та зрозуміла документація;
- ви можете змінити конфігурацію без перезапуску сервера.

Недоліки:

- використання мов часу виконання може бути важким;
- розробка багатопотокових додатків, у GORM можуть виникнути проблеми;

- комплексний процес інтеграції;
- час затримки.

Виходячи з короткого огляду перерахованих вище Java фреймворків, для реалізації інформаційної системи був обраний фреймворк Spring. Вибір пояснюється найбільшою універсальністю даного фреймворку. Spring дозволяє розробнику виконувати завдання будь-якої складності від роботи з базою даних до процедур тестування системи.

## 2.4.2 PHP

PHP – це сценарна мова на сервері, що використовується для створення статичних веб-сайтів або динамічних веб-сайтів або веб-додатків. PHP розшифровується як попередній процесор Hypertext, який раніше був персональним домашнім сторінками [5].

PHP – скрипти можна інтерпретувати лише на сервері, на якому встановлено PHP. Для клієнтських комп'ютерів, що отримують доступ до скриптів PHP, потрібен лише веб-браузер (див. рис. 2.6).



Рис. 2.6. Принцип роботи PHP-додатку

Файл PHP містить теги PHP і закінчується розширенням ".php".

Переваги PHP:

- PHP є відкритим і безкоштовним кодом;
- Низький поріг входження порівняно з іншими мовами, такими як JSP, ASP тощо;
- більшість серверів веб-хостингу підтримують PHP за замовчуванням на відміну від інших мов, таких як ASP, яким потрібен IIS. Це робить PHP економічно вигідним вибором;
- PHP регулярно оновлюється;

- PHP має вбудовану підтримку роботи з MySQL; це не означає, що ви не можете використовувати PHP з іншими системами управління базами даних;
- з PHP можна розгорнути свою програму в декількох різних операційних системах, таких як Windows, Linux, Mac OS тощо.

### 2.4.3 CSS

CSS розшифровується як каскадні таблиці стилів. Саме мова кодування надає веб-сайту його вигляд та макет. Поряд з HTML, CSS є основним для веб-дизайну. Без нього веб-сайти все ще будуть простим текстом на білому тлі [7].

До розробки CSS у 1996 році Всесвітнім консорціумом веб-сторінок (W3C) веб-сторінки були надзвичайно обмежені як за формою, так і за функцією. Ранні браузері представляли сторінку як гіпертекст - звичайний текст, зображення та посилання на інші гіпертекстові сторінки. Про макет взагалі не можна було говорити, а лише абзаци, що проходили по всій сторінці в одному стовпчику.

CSS дозволив зробити декілька нововведень у макеті веб-сторінок, наприклад, можливість:

- вказати шрифти, окрім стандартних для браузера;
- вказати колір та розмір тексту та посилань;
- нанесіть кольори на фони;
- зберігати елементи веб-сторінки у блоках і розміщувати їх на сторінці.

Вони помістили "стиль" у таблиці стилів, і вперше веб-сторінки могли бути розроблені.

Першим комерційним браузером для читання та використання CSS був Internet Explorer 3 Microsoft у 1998 році. На сьогоднішній день підтримка деяких функцій CSS залежить від браузера до браузера. W3C, який все ще контролює та створює веб-стандарти, нещодавно випустив новий стандарт для

CSS – CSS3. За допомогою CSS3 розробники сподіваються, що всі основні браузери будуть читати та відображати всі функції CSS однаково.

#### 2.4.4 Bootstrap

Bootstrap – це потужний інструментарій – це набір інструментів HTML, CSS та JavaScript для створення та створення веб-сторінок та веб-додатків.

Після випуску відкритого коду у 2011 році Bootstrap став популярним дуже швидко, і не без підстав. Веб-дизайнерам та веб-розробникам подобається Bootstrap, оскільки він гнучкий та простий у роботі [12]. Основні його переваги:

- підтримує широку сумісність веб-переглядачів;
- пропонує стійкий дизайн за допомогою повторно використовуваних компонентів;
- дуже простий у використанні;
- пропонує широку розширюваність за допомогою JavaScript;
- оснащений вбудованою підтримкою плагінів jQuery та програмним API JavaScript;
- може використовуватися з будь-яким IDE чи редактором та будь-якою технологією та мовою на стороні сервера, від ASP.NET від PHP до Ruby on Rails.

За допомогою програми Bootstrap веб-розробники можуть зосередитись на роботі над розробкою, не турбуючись про дизайн, і швидко отримати добре виглядаючий веб-сайт. І навпаки, це дає веб-дизайнерам міцну основу для створення цікавих тем Bootstrap.

#### 2.4.5 MySQL Workbench

MySQL Workbench – універсальний інструмент для роботи та управління базами даних. З його допомогою можна створювати та редагувати таблиці та інші об'єкти, керувати доступом користувачів та повноцінно адмініструвати БД. Workbench може стати в нагоді розробникам, адміністраторам БД та архітекторам систем [15].



Незважаючи на назву, він може працювати не лише з MySQL, але й з іншими СУБД: Oracle Database, Microsoft SQL Server, PostgreSQL та іншими.

Одна з головних рис Workbench – візуальне моделювання. Воно дозволяє побачити всі таблиці та зв'язки між ними у наочному вигляді. Це набагато спрощує проектування бази даних.

За допомогою SQL-скриптів досить легко створити базу даних із кількох таблиць. У ній легко орієнтуватися і всю схему можна пам'ятати. Такі БД зустрічаються лише у невеликих чи навчальних проектах.

У реальних системах бази даних містять десятки та сотні таблиць, процедур, уявлень та тригерів. Неможливо утримати всі ці об'єкти та зв'язки між ними в голові. І коли до такої схеми потрібно додати нову таблицю або розібратися в існуючій структурі, хочеться побачити всі таблиці перед очима.

Із цим може допомогти інструмент MySQL Workbench. Він спрощує багато аспектів роботи з БД: створення таблиць і зв'язків між ними, створення тригерів і процедур, візуальне представлення схеми, налаштування БД та створення резервних копій.

Мета багатьох інформаційних систем – перетворити дані в інформацію, щоб генерувати знання, які можна використовувати для прийняття рішень. Для цього система повинна бути здатна приймати дані, ставити дані в контекст і надавати інструменти для агрегації та аналізу. База даних створена саме для такої мети.

База даних – це організований збір пов'язаної інформації. Це організований збір, тому що в базі даних всі дані описуються та пов'язані з іншими даними [16].

В контексті баз даних варто розглянути поняття СУБД.

На комп'ютері база даних має вигляд одного або декількох файлів. Для того, щоб дані в базі даних були прочитані, змінені, додані або видалені, програмна програма повинна мати доступ до неї. Це призначення категорії програмних програм, званих системами управління базами даних (СУБД). Система управління базами даних (СУБД) – це комплекс програмних засобів,

необхідних для створення структури нової бази, її наповнення, редагування вмісту і відображення інформації. Найбільш поширеними СУБД є MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

Пакети СУБД зазвичай надають інтерфейс для перегляду та зміни дизайну бази даних, створення запитів та розробки звітів. Більшість цих пакетів розроблені для роботи з певним типом баз даних, але, як правило, сумісні з широким спектром баз даних [16].

Наприклад, база Apache OpenOffice.org може використовуватися для створення, зміни та аналізу баз даних у форматі відкритої бази даних (ODF). Microsoft SQL Access використовується для роботи з базами даних у власному форматі бази даних Microsoft Access. I Access, і Base мають можливість читати і писати в інші формати бази даних.

Microsoft Access та Open Office Base – приклади персональних систем управління базами даних. Ці системи в основному використовуються для розробки та аналізу баз даних для одного користувача. Ці бази даних не призначені для спільного використання в мережі чи Інтернеті, а замість цього встановлюються на певному пристрої та одночасно працюють з одним користувачем.

Існує кілька типів систем управління базами даних. Ось перелік семи загальних систем управління базами даних:

- ієрархічні бази даних;
- мережеві бази даних;
- реляційні бази даних;
- об'єктно-орієнтовані бази даних;
- графічні бази даних;
- бази даних моделей ER;
- бази даних документів;
- бази даних NoSQL.

Найпопулярнішою формою бази даних сьогодні є реляційна база даних. Популярними прикладами реляційних баз даних є Microsoft Access, MySQL та Oracle.

Реляційна база даних – це та, в якій дані впорядковані в одну або кілька таблиць. Кожна таблиця має набір полів, які визначають характер даних, що зберігаються в таблиці. Запис – це один екземпляр набору полів у таблиці [20].

У реляційній базі даних всі таблиці пов'язані одним або декількома полями, так що можна з'єднати всі таблиці в базі даних через спільні поля (поля), які мають спільне. Для кожної таблиці одне з полів ідентифікується як первинний ключ. Цей ключ є унікальним ідентифікатором для кожного запису в таблиці.

## **Висновки до розділу 2**

У другому розділі було детально розглянуто методи створення веб додатків та обрано засоби що будуть використані для створення сайту і системи бронювання. У ході написання розділу було розглянуто поняття CRM системи та бази даних, проаналізовано типи існуючих систем. Набуті знання буде використано при розробці проекту.

### 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

#### 3.1 Логічна структура сайту

Структура, іншими словами, ієрархія сайту – це його навігація.

Структура сайту заснована на категоризації і ділиться на:

- категорії – основні і найбільш важливі підсторінки на сайті, які нас найбільше цікавлять. Вони завжди повинні бути присутніми в меню;
- підкатегорії – інші підсторінки, пов'язані з основними підсторінками. Ті, що менш важливі або глибоко розташовані, не обов'язково повинні бути в меню – їх можна додати, наприклад, в макет сторінки.

Структура сайту для пошукових робіт – це посилальні зв'язки всередині сайту. Чим більше посилань на сторінку, тим вона пріоритетнее всередині сайту, і тим частіше пошуковик буде її сканувати [26].

Пошукові системи зроблять все можливе, щоб сканувати будь-які сайти без спеціальних вказівок. Тим не менш, ви можете допомогти пошуковику коректніше сприймати ваш сайт, якщо пропрацюєте чітку структуру.

Отже, структура позитивно впливає на те, як пошукові системи сканують і індексують ваш контент, але при цьому вона також впливає на користувачів. Сприйняття сайту користувачами все більш нерозривно пов'язане з ранжируванням в пошуковій видачі.

Види структур сайтів:

- проста лінійна. При такій організації подальша сторінка посилається на попередню і на головну. Ця форма підходить для невибагливих ресурсів (елементарних портфоліо, тематичних блогів), які не сильно розраховують на просування, а хочуть лише розповісти про що-небудь відвідувачеві. Лінійна структура буває ускладненою, з відгалуженнями. Тоді від будь-якої сторінки може відходити інша, підпорядкована;

- блокова. Іноді таку структуру називають ґратчастою. Сторінки, крім головної, рівнозначні, облінковані одна з одною. Структура досить важка в реалізації і підходить не для всіх ресурсів. Часто використовується в каталогах;
- деревоподібна. Це загальнопоширена багаторівнева структура, яку застосовують практично всі сайти. Вона універсальна і підходить для ресурсів різних типів. Послідовність сторінок утворює гілку з розділів і підрозділів, які пов'язані один з одним свого роду стовбуром або стрижнем. Може бути передбачено кілька рівнів вкладеності. Треба розуміти, що більше 3 рівнів вже викличе складнощі при просуванні і сприйнятті.

Для даного проекту була обрана структура деревовидного типу.

На рис. 3.1. зображено загальну структуру сайту.

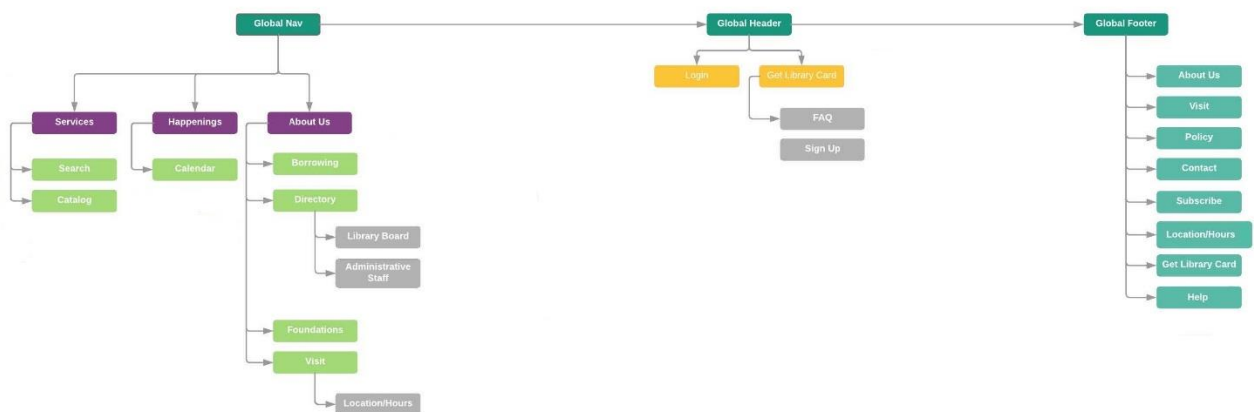


Рис. 3.1. Структура сайту

Структура навігації, яку було використано, складається з трьох різних меню. Глобальний заголовок, який містить швидкі посилання, глобальна навігація зі спадним меню для легкого перегляду вузлів (сторінок), і навігація в нижньому колонтитулі, яка містить деякі з найважливіших посилань на сайті. Нижній колонтитул використовується як додатковий, оскільки багато користувачів розуміють, що там зазвичай розміщується важлива інформація [27].

### 3.2 Проектування діаграм на мові uml

Для детального опису функціональних вимог до програмного забезпечення скористаємося діаграмою варіантів використання (use cases). Діаграма варіантів використання (англ. use-case function diagram) – діаграма, описуюча, яка розробляється програмною системою, доступна кожній групі користувачів [30].

Користувачі бібліотеки можуть переглянути бібліотечний каталог в Інтернеті, щоб знайти різноманітні ресурси – книги, періодичні видання, аудіо- та візуальні матеріали чи інші предмети, які знаходяться під контролем бібліотеки. Користувачі можуть зарезервувати або поновити товар, надати відгук і керувати своїм обліковим записом (див. рис. 3.2).

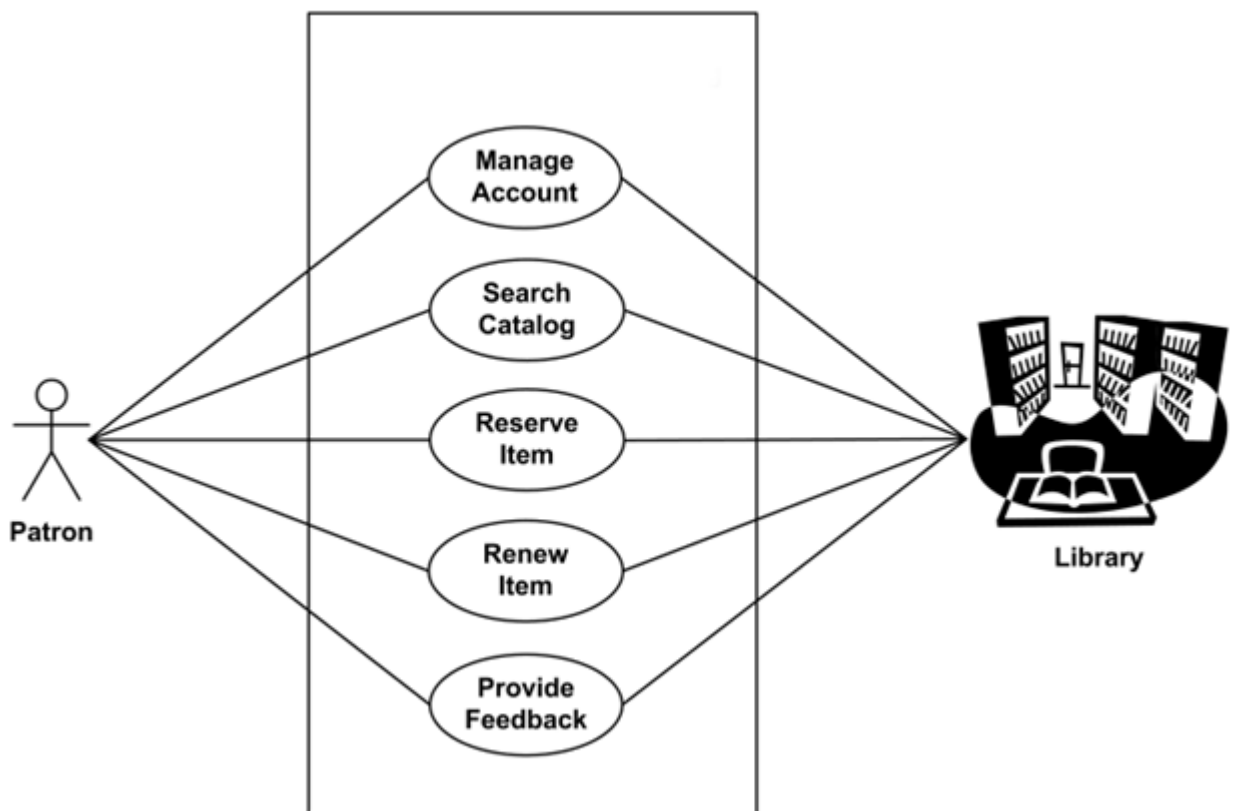


Рис. 3.2. Система управління бібліотекою

У наведеному проекті бібліотечної системи є три користувача Бібліотекар, Студент та Адміністратор. І бібліотекар, і студент можуть виконувати такі дії після входу в систему управління бібліотекою (див. рис. 3.3).

### Діяльність бібліотекаря:

- додати книги;
- додати студента;
- видати книгу;
- повернути книги;
- застосувати штраф;
- змінити пароль;
- переглянути звіти.

### Діяльність учнів:

- пошук книги;
- звіт про випуск / повернення книги;
- звіт про штрафи;
- редагувати профіль.

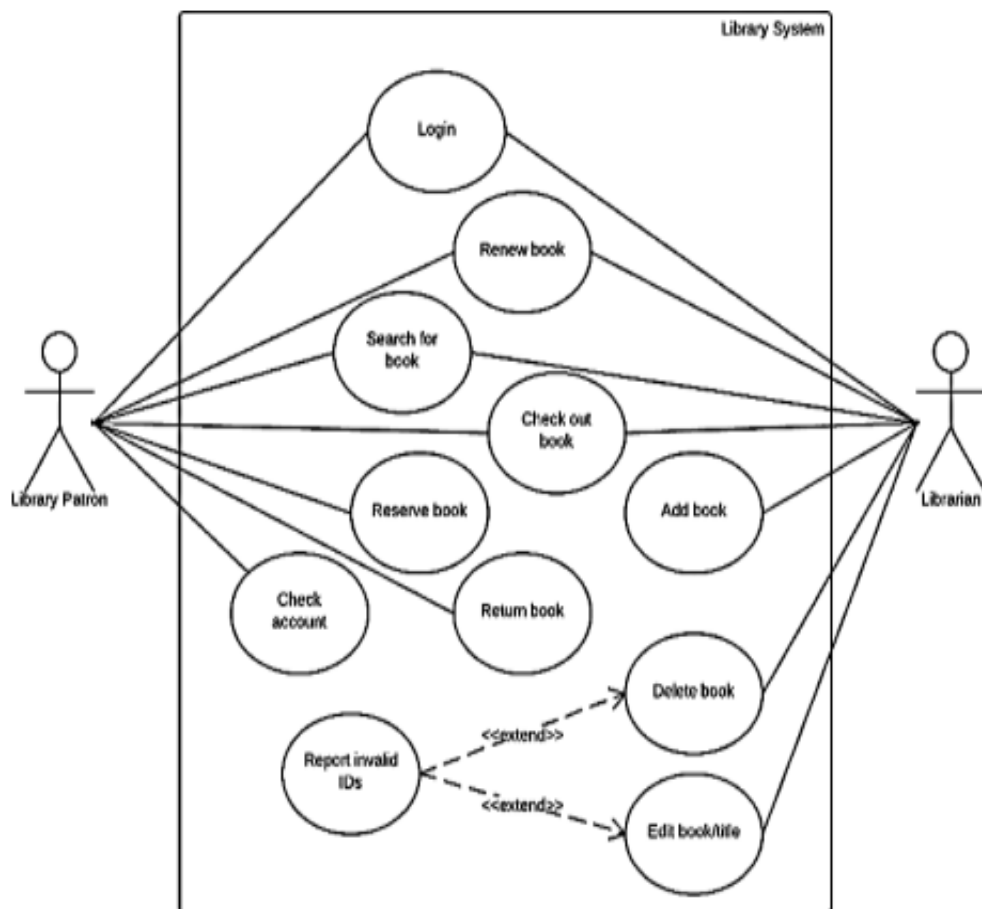


Рис. 3.3. Система відосин читач-бібліотекар

Адміністратор зможе керувати користувачами, наприклад, деактивувати їх, створювати нових бібліотекарів (звичайні користувачі зможуть зареєструватися вручну) або оновлювати свою інформацію. Йому доступні такі панелі як трафік користувачів та статистика книг (див. рис. 3.4).

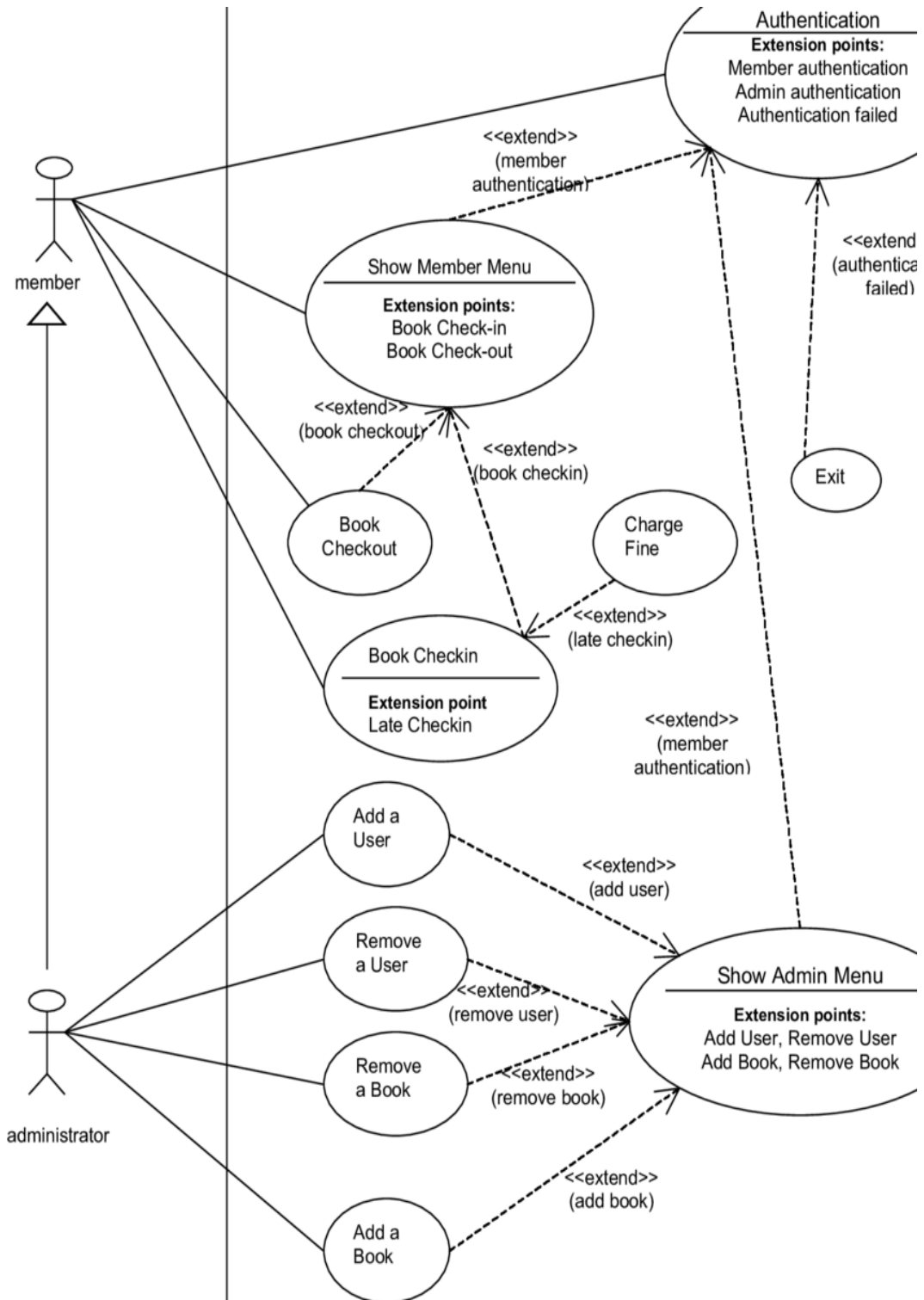


Рис. 3.4. Система відосин користувач-адміністратор



### 3.3 Проектування бази даних

Ключове місце в системі управління взаємодією з клієнтами займає база даних, на проектування якої слід звернути особливу увагу, оскільки існує потреба в роботі з великими обсягами інформації. Виходячи з цього під час розробки структури БД були використані механізми нормалізації. Нормалізація полягає в приведенні структури зберігання даних до нормальних форм. Нормальна форма – це ряд вимог, що пред'являється до структури таблиць для забезпечення цілісності бази даних і усунення надмірних функціональних залежностей [31]. В цій системі база даних буде приведена до перших трьох нормальних форм:

- перша нормальна форма: кожна таблиця повинна мати первинний ключ; кожен атрибут має складатись тільки з елементарних (неподільних) значень і не містити групи, що повторюються (атомарність);
- друга нормальна форма: дані, що повторно з'являються в декількох рядках, виносяться в окремі таблиці;
- третя нормальна форма: дані не повинні зберігатися в таблиці, якщо їх можна отримати з не ключових полів.

Враховуючи всі особливості для кожної сутності була створена таблиця з відповідними атрибутами. Зв'язки між таблицями бази даних визначаються за допомогою ключів двох типів:

- первинний ключ – значення, що унікально ідентифікує кожний запис;
- зовнішній ключ – значення, що відповідає первинному ключу з іншої таблиці.

Розглянемо деякі таблиці бази даних системи, а також їх призначення.

Users – зберігає інформацію про користувачів системи. Структуру полів наведено в таблиці 3.1.

Таблиця 3.1

Структура полів таблиці Users

Назва	Тип	Інформаційний зміст
id	bigint	Унікальний ідентифікатор запису
username	varchar	Нік користувача
first_name	varchar	Ім'я користувача
last_name	varchar	Прізвище користувача
phone	varchar	Номер телефону
birthday	date	Дата народження
email	varchar	Адреса електронної пошти (використовується в якості логіну при вході в систему)
city	varchar	Місто користувача
password	varchar	Пароль
adress	varchar	Адреса користувача
record_created	timestamp	Дата реєстрації профілю

Books – зберігає інформацію про книги. Структуру полів наведено в таблиці 3.2.

Таблиця 3.2

Структура полів таблиці Books

Назва	Тип	Інформаційний зміст
id	bigint	Унікальний ідентифікатор книги
title	varchar	Назва книги
publisher	varchar	Видавництво

published_date	varchar	Дата публікації
page_count	int	Кількість сторінок
raiting	double	Рейтинг
Image_Ink	varchar	Назва файлу (зображення обкладинки)
desription	text	Опис

Структура бази даних представлена малюнку 3.5.

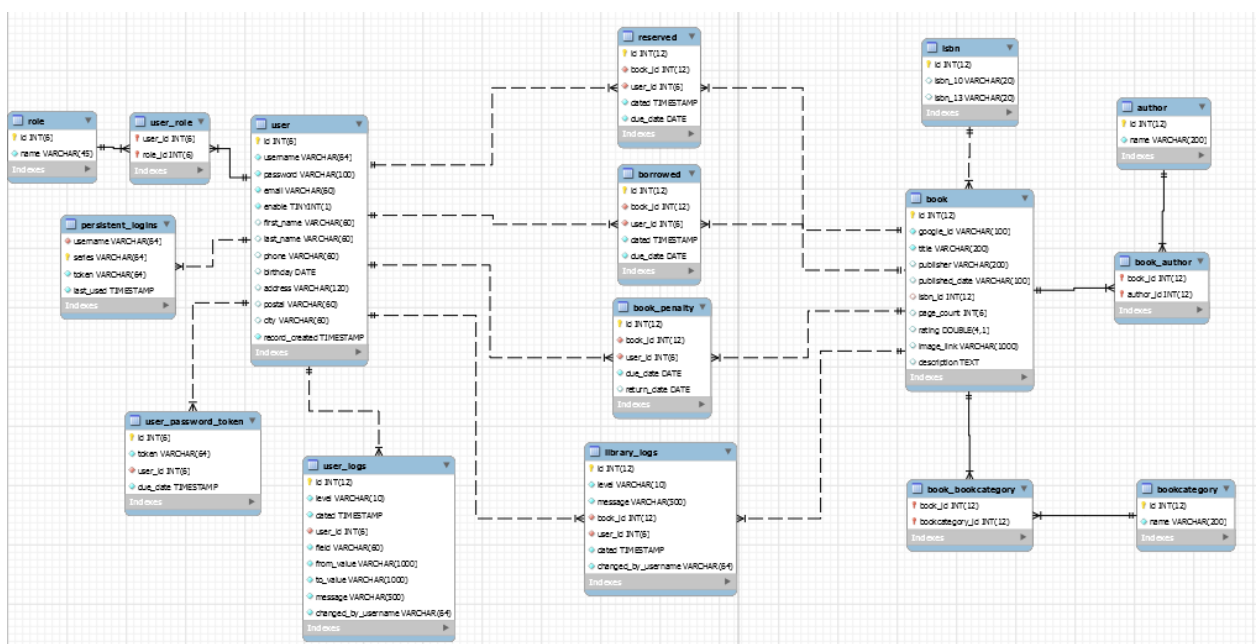


Рис. 3.5. Структура бази даних

Детальний огляд взаємовідносин об'єктів користувача. Відносини користувач-книга, ці таблиці зберігають інформацію про стан книги (наприклад, якщо вона зарезервована, запозичена, має штрафи) або журнали (див. рис. 3.6).

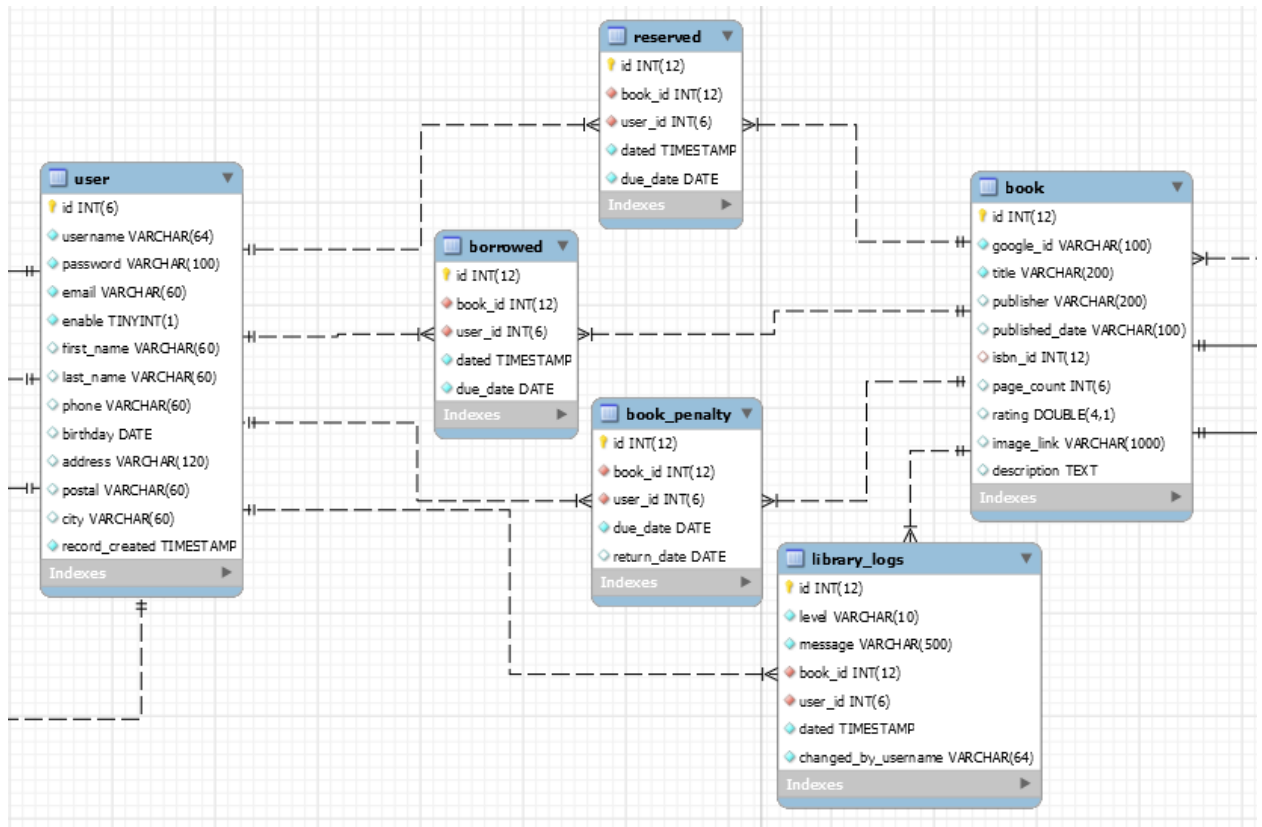


Рис. 3.6. Відносини користувач-книга

### 3.4 Панелі адміністратора та бібліотекаря

Ресурс являє собою електронну бібліотеку, точну систему учета та реєстраційний друкований видання та користувачів, яким дозволений доступ до цих друкованих видань. Ресурсом можна скористатися лише зареєстрованими користувачами, для цього необхідно, щоб один із ресурсів адміністратора заповнив відповідну форму реєстрації користувача та видав логін та пароль. Користувач може брати книги і повертати їх, якщо книга не буде повертатися впродовж трьох днів, то ця книга потрапляє в список прострочених книг користувача, і можливо введення санкцій щодо цих користувачів. Користувач може переглянути список книг, не реєструючись на ресурсі, для цього необхідно зайти на стартову сторінку ресурсу. Адміністратор ресурсу може додавати, видаляти, змінювати дані, як про користувачів, так і про книги. Адміністратор не може брати і повертати книги, щоб отримати цю можливість, йому необхідно зареєструватися як звичайний користувач.

Бібліотекарі можуть додавати нові книги до бібліотеки. Крім того, вони зможуть перевірити кожну книгу, видана вона чи ні, і перевірити, коли книгу потрібно повернути до бібліотеки. Нарешті вони зможуть відзначити, що читач повернув книгу.

Для того щоб зайти до панелі адміністратора необхідно увійти на сайт з паролем адміністратора, що зазначений у відповідній таблиці бази даних. (див. рис. 3.7).

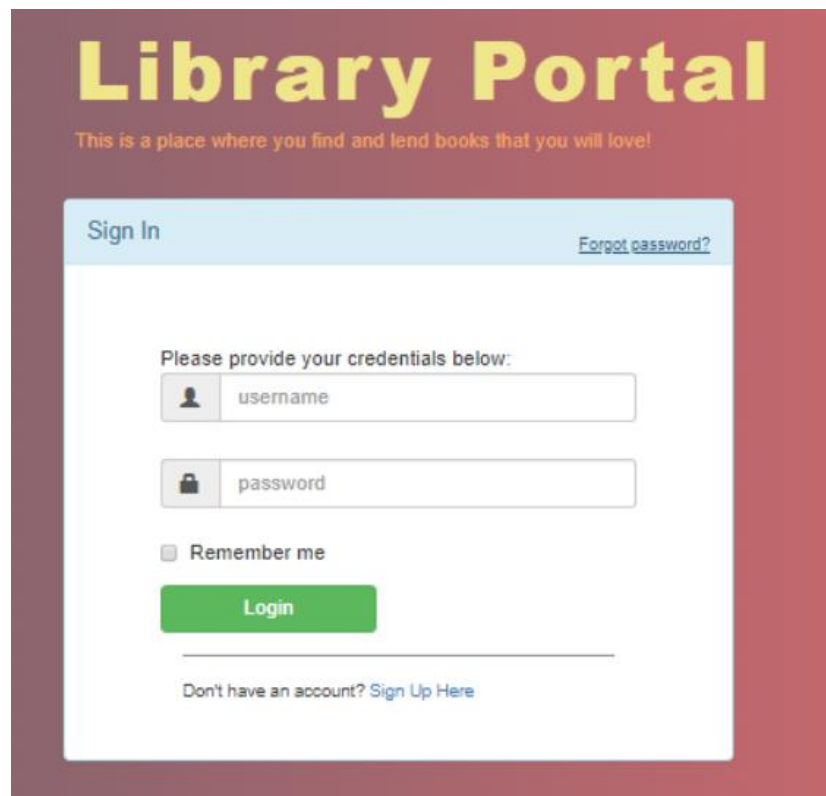


Рис. 3.7. Запрошення увійти до панелі адміністратора

Адміністратор може керувати користувачами, наприклад, деактивувати їх, створювати нових бібліотекарів (звичайні користувачі зможуть зареєструватися вручну) або оновлювати свою інформацію. Йому доступні такі панелі як трафік користувачів та статистика книг (див. рис. 3.8)

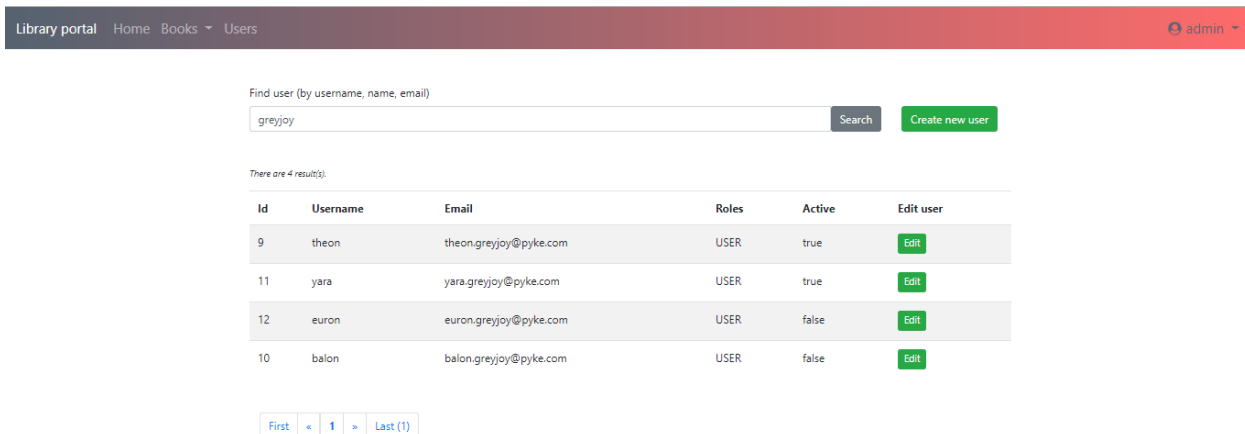


Рис. 3.8. Список користувачів

Адміністратор може знаходити та змінювати профілі користувачів. Також у нього є можливість побачити журнали користувача, щоб перевірити, які зміни були внесені в обліковий запис користувача (див. рис. 3.9).

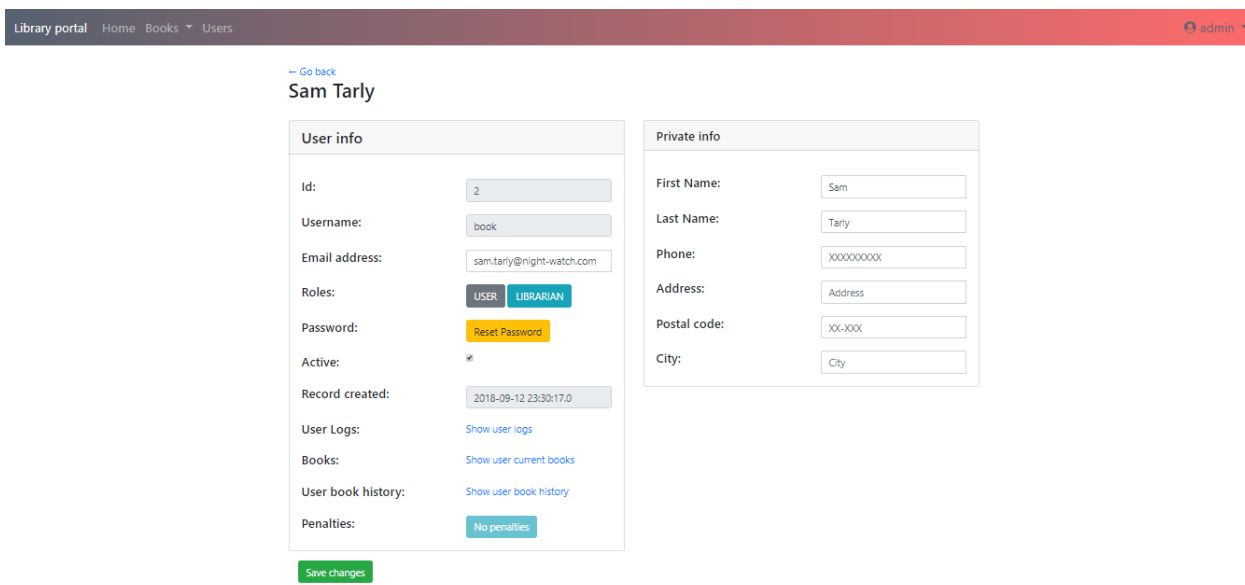


Рис. 3.9. Редагування профілю

Якщо користувач вибрав «Редагувати книги», то йому буде введено список усіх книг і надано можливість видалити та редагувати вже наявні книги та додати нову книгу. Якщо користувач вибрав додати книгу, то йому буде надано сторінку «Створити книгу», де необхідно заповнити відповідну форму. Бібліотекар може додавати нові книги до бібліотеки. Дані про нові книги

отримуються з Google Book API, коли виконується пошуковий запит (див. рис. 3.10).

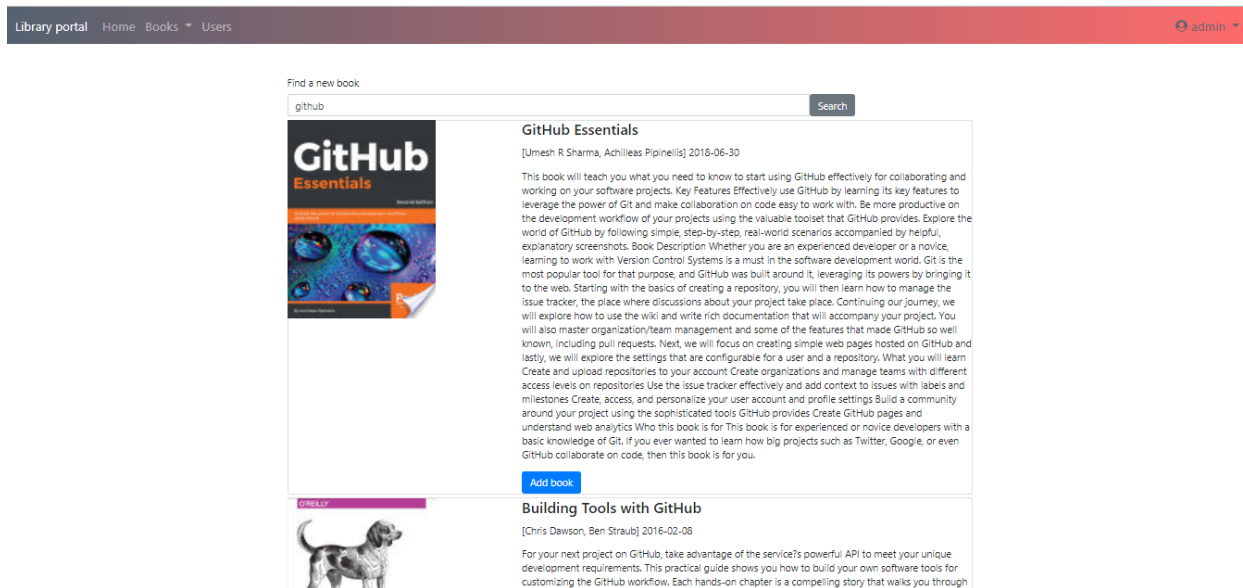


Рис. 3.10. Додавання книги

Бібліотекар може брати і повертати книги. Крім того, він може перевірити історію книги, щоб зрозуміти, хто і коли робить якісь дії з книгою (див. рис. 3.11).

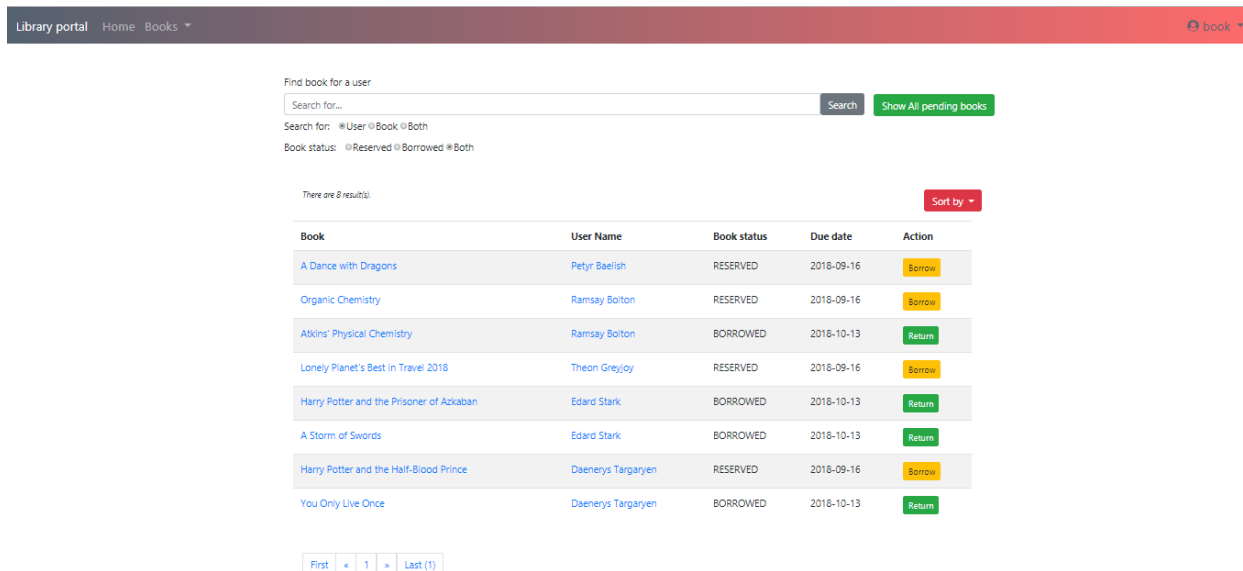
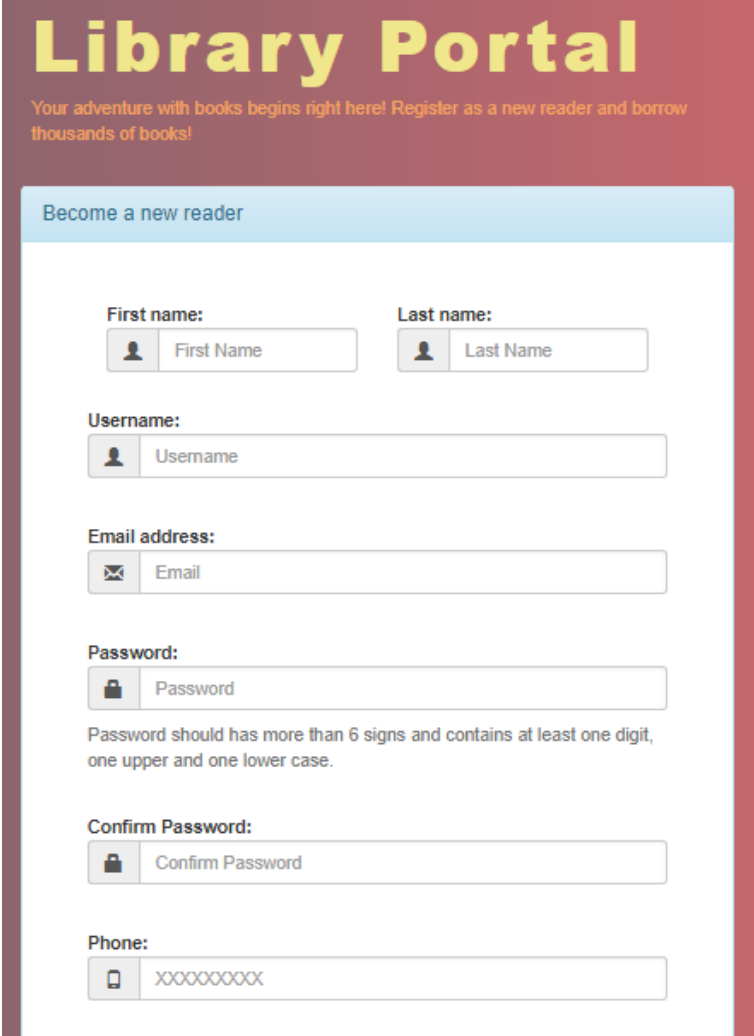


Рис. 3.11. Історія книги

При переході по ссилку «Вийти» користувач закінчує роботу в системі.

### 3.5 Панель користувача

Читачі – це найчисельніша група користувачів. Після входу до свого профілю вони зможуть знайти та взяти деякі книги з бібліотеки. Для того щоб отримати можливість позичити книгу, необхідно увійти на сайт або зареєструватись. Зробити це можна через форму реєстрації. Після заповнення всіх полів інформація про користувача стане доступною у базі даних та у панелі адміністратора (див. рис. 3.12).



The image shows a registration form titled "Library Portal" with the subtitle "Your adventure with books begins right here! Register as a new reader and borrow thousands of books!". The form is titled "Become a new reader" and contains the following fields:

- First name:** Input field with a person icon and placeholder "First Name".
- Last name:** Input field with a person icon and placeholder "Last Name".
- Username:** Input field with a person icon and placeholder "Username".
- Email address:** Input field with an envelope icon and placeholder "Email".
- Password:** Input field with a lock icon and placeholder "Password". Below this field is a note: "Password should has more than 6 signs and contains at least one digit, one upper and one lower case."
- Confirm Password:** Input field with a lock icon and placeholder "Confirm Password".
- Phone:** Input field with a phone icon and placeholder "XXXXXXXXXX".

Рис. 3.12. Форма реєстрації

Після реєстрації користувачеві стане доступним його профіль на сайті, який він може змінити. Звичайний користувач обмежений в діях що може виконувати на сайті, йому не доступна панель адміністрування.



Після реєстрації вони можуть шукати потрібну книгу, переглянути її деталі та забронювати її (див. рис. 3.13).

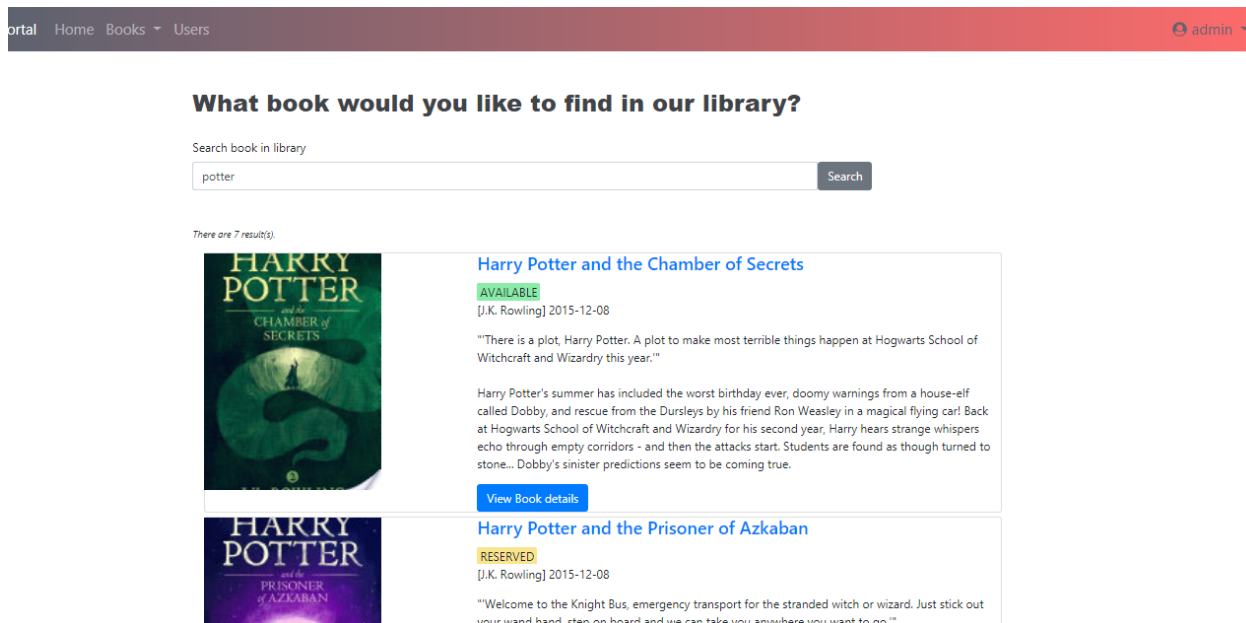


Рис. 3.13. Доступні книги

На сторінці «Всі книги» користувач може взяти книгу, і тоді назви книги буде додано до його списку взятих книг (див. рис. 3.14).

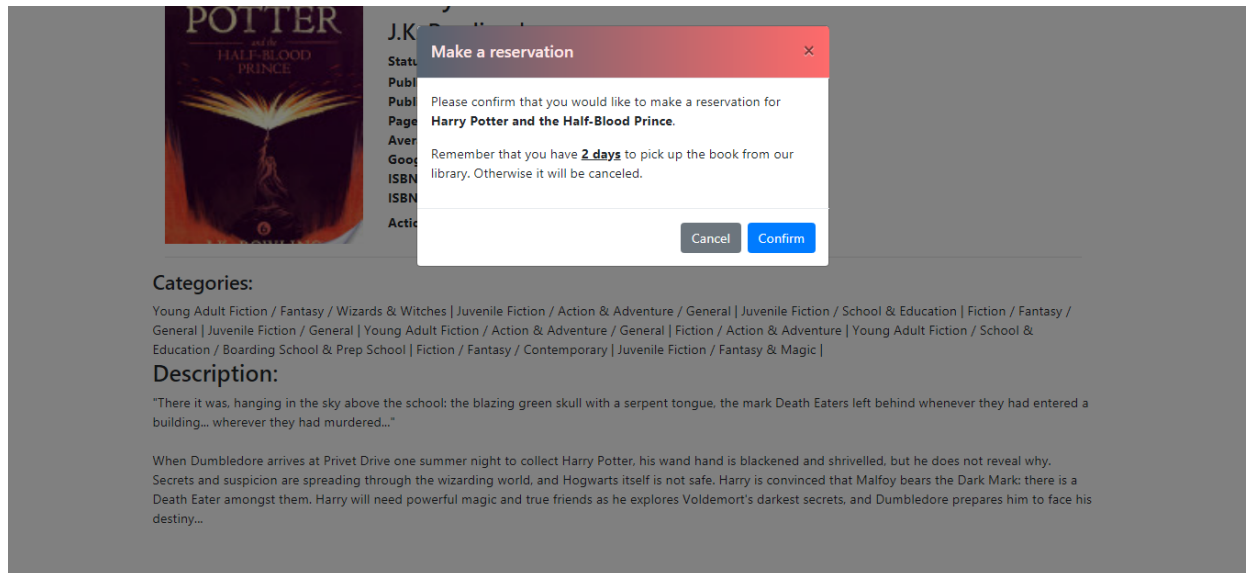


Рис. 3.14. Бронь книги

Якщо книга була заброньована, на сайті з'явиться відповідна позначка (див. рис. 3.15).

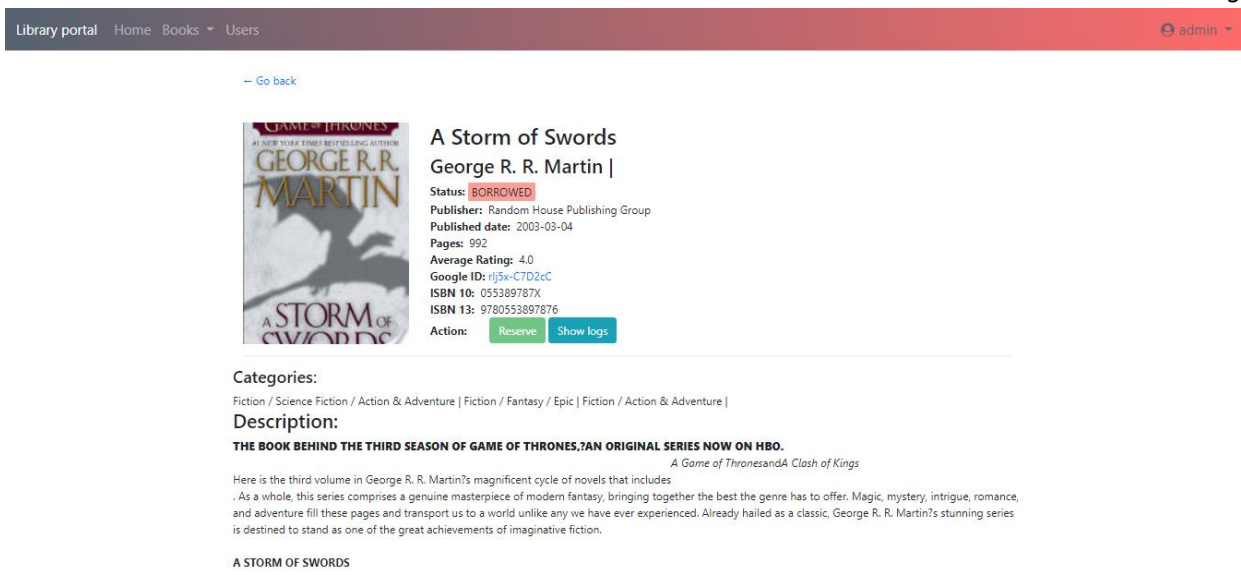


Рис. 3.15. Заброньована книга

Якщо книга була повернена у виділений час, користувач отримає повідомлення. Таким чином користувач зможе відстежувати історію своєї книги та свої штрафи (див. рис. 3.16).

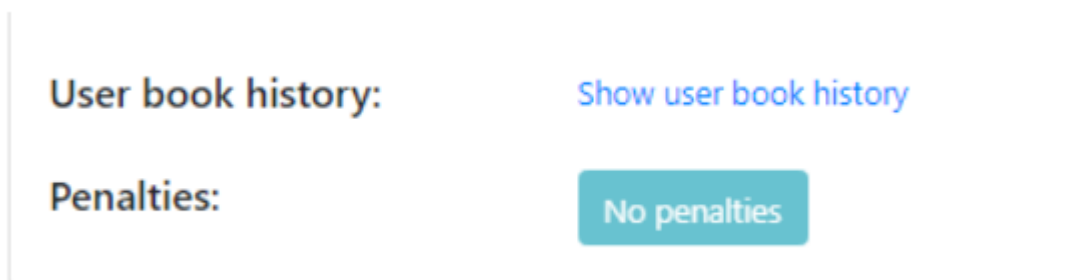


Рис. 3.16. Історія читача

При переході по ссилку «Вийти» користувач закінчить роботу в системі.

### Висновки до розділу 3

У даному розділі дипломної роботи був проведений огляд створеного сайту та системи бронювання.

Була розроблена модель структури сайту та макет головної сторінки.

Були описані функції кожної сторінки, можливості адміністратора та користувача. Розглянуто таблиці створеної бази даних.

Створено сайт та перевірено працездатність створеної системи.

## ВИСНОВКИ

У магістерській кваліфікаційній роботі була досліджена аналітична система для моніторингу і контролю взаємодії з користувачами бібліотеки.

Для даного проекту було розроблено план проектних завдань та розраховані терміни виконання кожного завдання. Вивчено та проаналізовано методи та засоби розробки веб сторінок: відповідне програмне забезпечення, мови програмування та фреймворки, створена база даних і розроблено систему взаємодії з користувачами бібліотеки.

У процесі роботи було ретельно досліджено предметну область та вироблено практичні навички роботи з аналітичними системами.

Створена система була протестована і задовольняє всім вимогам, поставленим на етапі постановки завдання.

Були вирішені наступні завдання:

- проведено аналіз предметної сфери;
- обрано технології розробки веб-додатків;
- створено схему та макет сайту;
- реалізована база даних;
- розроблено систему взаємодії з користувачами бібліотеки.

У спеціальному розділі було проаналізовано приміщення, у якому може використовуватися дана система. Було визначено загальні правила безпеки в офісному приміщенні, правила організації робочого простору та правила пожежної безпеки на підприємстві.

У методичній частині було розроблено лабораторні роботи для відпрацювання навичок роботи з електронними ресурсами.

Мета дослідження щодо проектування аналітичної системи та створення інструменту – веб-сайту бібліотеки з інформаційно-довідковою системою, призначеною для контролю ситуацій з заявками, зверненнями і етапами роботи учнів з книгами досягнута. Розроблена аналітична система дозволяє побудувати грамотний діалог з читачами і відповідає поставленим цілям.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Будилів В. А. Основи програмування для Інтернету / В. А. Будилів. – СПб. : БХВ-Петербург, 2003. - 736 с.
2. Глушков С.В. Програмування Web-сторінок / С. В. Глушков, І. А. Жакін, Т. С. Хачиров. – Х.: Фоліо, 2005. - 390 с.
3. Калиновський А. І. Юзабіліті: як зробити сайт зручним / А. І. Калиновський. – Мн. : "Нове знання", 2005. - 220 с.
4. Крістофер Шмітт CSS. Рецепти програмування / Крістофер Шмітт. – СПб. : "БХВ-Петербург", 2007. - С. 592.
5. Лещев Д. Створення інтерактивного web-сайту: навчальний курс / Д. Лещев. – СПб. : Пітер, 2003. - 544 с.
6. Ломов А. Ю. HTML, CSS, скрипти: практика створення сайтів / А. Ю. Ломов. – СПб. : БХВ-Петербург, 2006. - 416 с.
7. Мак-Дональд М. Microsoft ASP.NET 4 з прикладами на С # 2010 для професіоналів, 4-е изд. / М. Мак-Дональд, А. Фрімен, М. Шпуста – М.: ТОВ "І.Д. Вільямс", 2011. - 1424 с. : іл. Парал. тит. англ.
8. Пономаренко В. С. Основи технологій Internet: навчальний 55 посібник / В. С. Пономаренко, С. В. Минухин, І. А. Торохтій. – Х.: Вид. ХГЕУ, 2001. - 256 с.
9. Прохоренко Н. HTML, Javascript, PHP і MySQL. Джентльменський набір Web-майстра / Н. Прохоренко. – СПб. : БХВПетербург, 2008. - 640 с.
10. Сандереон Стівен. ASP.NET MVC Framework з прикладами на С # для професіоналів / Сандереон Стівен; пер. з англ. – М.: ТОВ "І.Д. Більямс", 2010. - 560 с.
11. Створення Web-сторінок і Web-сайтів. Самовчитель / під ред. В. Н. Печникова. – М.: Изд-во Тріумф, 2006. - 464 с.
12. Соколов С. А. HTML і CSS в прикладах, типових рішеннях і завданнях. Професійна робота / С. А. Соколов. – М.: "Вільямс", 2007. - 416 с.

13. Томсон Л. Розробка Web-додатків на PHP і MySQL – пров. з англ. - 2-е изд., Испр. – СПб. : "ДиаСофтЮП", 2003. - 672 с.
14. Чистов Д.В., Шуремов Є.Л., Заложнев А.Ю. Інноваційні технології внутрішньфірмового керування. – М.: Видавництво ВивідАрт, 2010. – 200-356 с.
15. Шуремов Є. Л. Інформаційні технології управління взаємовідносинами із клієнтами. – М.: Видавництво ІС-Паблішинг, 2005. – 86 с.
16. Ульман Л. Основи програмування на PHP / Л. Ульман. – М.: ДМК Пресс, 2001. - 288 с.
17. Ерік А. Мейер CSS-каскадні таблиці стилів: докладний посібник / А. Ерік Мейер. – М.: "Символ", 2006. - С. 576.
18. Будилів В. А. Основи програмування для Інтернету / В. А. Будилів. – СПб. : БХВ-Петербург, 2003. - 736 с.
19. Глушков С.В. Програмування Web-сторінок / С. В. Глушков, І. А. Жакін, Т. С. Хачиров. – Х.: Фоліо, 2005. - 390 с.
20. Крістофер Шмітт CSS. Рецепти програмування / Крістофер Шмітт. – СПб. : "БХВ-Петербург", 2007. - С. 592.
21. Лещев Д. Створення інтерактивного web-сайту: навчальний курс / Д. Лещев. – СПб. : Пітер, 2003. - 544 с.
22. Ломов А. Ю. HTML, CSS, скрипти: практика створення сайтів / А. Ю. Ломов. – СПб. : БХВ-Петербург, 2006. - 416 с.
23. Мак-Дональд М. Microsoft ASP.NET 4 з прикладами на C # 2010 для професіоналів, 4-е изд. / М. Мак-Дональд, А. Фрімен, М. Шпуста – М.: ТОВ "І.Д. Вільямс", 2011. - 1424 с. : іл. Парал. тит. англ.
24. Пономаренко В. С. Основи технологій Internet: навчальний 55 посібник / В. С. Пономаренко, С. В. Минухин, І. А. Торохтій. – Х.: Вид. ХГЕУ, 2001. - 256 с.
25. Прохоренко Н. HTML, Javascript, PHP і MySQL. Джентльменський набір Web-майстра / Н. Прохоренко. – СПб. : БХВПетербург, 2008. - 640 с.

26. Створення Web-сторінок і Web-сайтів. Самовчитель / під ред. В. Н. Печникова. – М.: Изд-во Тріумф, 2006. - 464 с.
27. Соколов С. А. HTML і CSS в прикладах, типових рішеннях і завданнях. Професійна робота / С. А. Соколов. – М.: "Вільямс", 2007. - 416 с.
28. Томсон Л. Розробка Web-додатків на PHP і MySQL - пров. з англ. - 2-е изд., Испр. – СПб. : "ДиаСофтЮП", 2003. - 672 с.
29. Троелсен Е. Мова програмування C # 2010 і платформа .NET 4.0, 5-е изд. / Е. Троелсен; пер. з англ. – М.: ТОВ "І.Д. Вільямс", 2011. - 1392 с.
30. Ульман Л. Основи програмування на PHP / Л. Ульман. – М.: ДМК Пресс, 2001. - 288 с.
31. Ерік А. Мейер CSS-каскадні таблиці стилів: докладний посібник / А. Ерік Мейер. – М.: "Символ", 2006. - С. 576.
32. Байков В.Д. Інтернет: пошук інформації та просування сайтів / В.Д. Байків. – СПб.: БХВ-Санкт-Петербург, 2000. – 288 с.
33. Науково-методична підтримка розробки наукових електронних бібліотек / Акімов С.І., Єлізаров А.М., Єршова Т.В., Когаловський М.Р., Федоров А.О., Хохлов Ю.Є// Електронні бібліотеки: Російський науковий електронний журнал. - 2005 - Том 8 - Вип. 1 [Електрон. ресурс]. – Режим доступу: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2005/part1>
34. Паршукова, Г. Б. Інформаційна компетентність інженера: віртуальний простір: навчальний посібник [Текст]/Новосиб. держ. техн. ун-т. - Новосибірськ: Вид-во НДТУ, 2005 (у бібліотеці НДТУ - 22)
35. Закон України «Про охорону праці» (№ 2694-ХІІ від 14.10.1992 р.)
36. Кодекс цивільного захисту України (№ 5403-VI від 02.04.2020р.)
37. Закон України «Про основи національної безпеки України» (№ 2411-VI/2411-17 від 01.07.2010 р.)

## ДОДАТОК А

### Лістинг коду

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сьогодні - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
<dd>gdgsgshr</dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
```

on the Internet tend to repeat predefined chunks as necessary, making this the first true generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic words etc. </dd>

```
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
<p></p>
<center><b>O</b>u are going to use a passage </b></center>
<p>Автор: O</p>u are going to use a passage <br>
Жанр: O</p>u are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">O</b>u are going to use a passage </b></center>
<p>Автор: O</p>u are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
</div>
<div id="content">
<p></p>
<h2>Library</h2>
<?php
if (isset($_SESSION['uid'])) {
if ($_SESSION['uid'] == 1) {
echo "<p>
</p>";
}
} else {
echo " <p>
</p>";
}
?>
</div>
<div id="footer">
<p class="copyright">&copy; Your site 2014 | Created by Asem Omarova | <a
href="https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%8
2%D0%B5%D0%BA%D0%B0">www.aseplib.com</a> </p>
<p></p>
</div>
</div>
</div>
</body>
</html>
```



### bd.php

```
<?php
session_start();
mysql_connect('localhost','admin','123') or die(mysql_error());
mysql_select_db('lib') or die(mysql_error());
mysql_query('SET NAMES cp1251');
include('uni-auth.php');
if(USER_LOGGED) {
if(!check_user($UserID)) logout();
?>
<p>Здравствуйте, <?php echo $UserName; ?>!</p>
<h2>Ваш ID: <?php echo $UserID; ?>.</h2>
<h4><a href="?logout">Выход</a></h4>
<?php
}
else { ?>
<form method="POST" action="<?php echo $_SERVER['PHP_SELF']; ?>">
<table>
<tr>
<td>Имя<input type="text" class="fields_contact_us" name="user"></td>
</tr>
<tr>
<td>Пароль:<input type="password" class="fields_contact_us" name="pass"></td>
</tr>
<tr>
<td colspan="2"><input type="submit" class="fields_contact_us" name="login"
value="Войти"></td>
</tr>
</table>
</form>
<?php
}
?>
```

### uni-auth.php

```
<?php
##Определяем константы
define('USERS_TABLE','users');
define('SID',session_id());
##Определяем функции
//Функция выхода.
//Пользователь считается авторизованным, если в сессии присутствует uid
//см. "Действия - если пользователь авторизован".
function logout() {
unset($_SESSION['uid']); //Удаляем из сессии ID пользователя
die(header('Location: '.$_SERVER['PHP_SELF']));
}
//Функция входа.
//Все выбранные поля записываются в сессию.
```

```
//Таким образом, при каждом просмотре страницы не надо выбирать их заново.  
//Для обновления информации из БД можно пользоваться этой же функцией - имя и  
пароль  
//хранятся в сессиях  
function login($username,$password) {  
$result = mysql_query("SELECT * FROM `".USERS_TABLE."` WHERE  
`username`='".$username.'" AND `password`='".$password.'");  
or die(mysql_error());  
$USER = mysql_fetch_array($result,1); //Генерирует удобный массив из  
результата запроса  
if(!empty($USER)) { //Если массив не пустой (это значит, что пара имя/пароль  
верная)  
$_SESSION = array_merge($_SESSION,$USER); //Добавляем массив с пользователем  
к массиву сессии  
mysql_query("UPDATE `".USERS_TABLE."` SET `sid`='".$SID.'" WHERE  
`uid`='".$USER['uid'].'"");  
or die(mysql_error());  
return true;  
}  
else {  
return false;  
}  
}  
//Функция проверки залогинности пользователя.  
//При входе, ID сессии записывается в БД.  
//Если ID текущей сессии и SID из БД не совпадают, производится logout.  
//Благодаря этому нельзя одновременно работать под одним ником с разных  
браузеров.  
function check_user($uid) {  
$result = mysql_query("SELECT `sid` FROM `".USERS_TABLE."` WHERE  
`uid`='".$uid.'"); or die(mysql_error());  
$sid = mysql_result($result,0);  
return $sid==SID ? true : false;  
}  
##Действия - если пользователь авторизован  
if(isset($_SESSION['uid'])) { //Если была произведена Авторизация, то в  
сессии есть uid  
//Константу удобно проверять в любом месте скрипта  
define('USER_LOGGED',true);  
//Создаём удобные переменные  
//Все поля таблицы пользователей записываются в сессии (см. стр. 35-37)  
//Таким образом, после добавления нового поля в таблицу надо дописать лишь  
одну строку  
$UserName = $_SESSION['username'];  
$UserPass = $_SESSION['password'];  
$UserID = $_SESSION['uid'];  
}  
else {
```

```
define('USER_LOGGED',false);
}
##Действия при попытке входа
if (isset($_POST['login'])) {
if(get_magic_quotes_gpc()) { //Если слешы автоматически добавляются
$_POST['user']=stripslashes($_POST['user']);
$_POST['pass']=stripslashes($_POST['pass']);
}
$user = mysql_real_escape_string($_POST['user']);
$pass = mysql_real_escape_string($_POST['pass']);
if(login($user,$pass)) {
header('Refresh: 3');
die('Вы успешно авторизировались!');
}
else {
header('Refresh: 3;');
die('Пароль неправильный!');
}
}
##Действия при попытке выхода
if(isset($_GET['logout'])) {
logout();
}
?>
t2.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav">
<?php
echo "Сьогодні - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="#about" class="about"><span>Контакти</span></a></li>
</ul>

```

```
<form action="search.php" method="post" id="forms">
<p class="search_t">Поисковый запрос должен быть не менее 4-х
символов.</p><br>
<p><input name="search" type="text" size="25" maxlength="40">
<input class="search_b" name="submit_s" type="submit" value="Искать">
</p>
</form>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
```

```
<p></p>
<center><b>Oу are going to use a passage </b></center>
<p>Автор: Oу are going to use a passage <br>
Жанр: Oу are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">Oу are going to use a passage </b></center>
<p>Автор: Oу are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
</div>
<div id="content">
<center>
<font color="#0000FF" size="4"><b>Список книг </b></font>
</center>
<?php
// ваш адрес где находится, хостится ваша база данных
$sdd_db_host='localhost';
// Имя базы данных с которой вы хотите работать, так как их может быть
множество
$sdd_db_name='lib';
// логин доступ к базе данных
$sdd_db_user='admin';
// пароль доступа к базе данных
$sdd_db_pass='123';
// устанавливаем связь с сервером
@mysql_connect($sdd_db_host,$sdd_db_user,$sdd_db_pass);
mysql_query("set names cp1251"); /*Кодировка*/
// переключаемся на нужную нам базу данных
@mysql_select_db($sdd_db_name);
// делаем выборку из таблицы
$result=mysql_query('SELECT Nazvanie, Book_ID FROM `Book` order by
Nazvanie');
// берем результаты из каждой строки
if (isset($_SESSION['uid'])) {
if ($_SESSION['uid'] == 1) {
echo "<p align='right'>
<a href='insert.php'>Добавить книгу</a></p>";}}
while($row=mysql_fetch_array($result))
{
if (isset($_SESSION['uid'])) {
if ($_SESSION['uid'] == 1) {
echo "<p><a href='d.php?id=".$row['Book_ID']."'>".$row['Nazvanie']."</a>
<a href='delete.php?id=".$row['Book_ID']."'>Удалить</a>
<a href='upd.php?id=".$row['Book_ID']."'>Изменить книгу</a></p>";
}
}
```

```
} else {
echo "<p><a
href='d.php?id=".$row['Book_ID']."'>".$row['Nazvanie']. "</a><br></p>";
}
}
?>
</div>
<div id="footer">
<p class="copyright">&copy; Your site 2014 | Created by Asem Omarova | <a
href="https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%8
2%D0%B5%D0%BA%D0%B0">www.aseplib.com</a> </p>
<p></p>
</div>
</div>
</div>
</body>
</html>
```

#### insert.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сьогодні - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
```

```
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
<dd>gfuskhf,shjssfsg</dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
<p></p>
<center><b>Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Ou are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
</div>
<div id="content">
<tr>
<td><table width="690" border="0" cellspacing="0" cellpadding="0">
<tr>
<!--Подключаем левый блок сайта-->
<td valign="top">
<form action="add_book.php" method="post">
Название книги: <br><input type="text" name="Nazvanie" id='Nazvanie'><br>
Количество страниц:<br> <input type="text" name="Kol_vo_str"
id='Kol_vo_str'><br>
```

```
Количество экземпляров: <br><input type="text" name="Kol_vo_exmemplar"
id='Kol_vo_exmemplar'><br>
Цена:<br> <input type="text" name="Cena" id='Cena'><br>
ISBN:<br> <input type="text" name="ISBN" id='ISBN'><br>
UDK:<br> <input type="text" name="UDK" id='UDK'><br>
ВВК:<br> <input type="text" name="ВВК" id='ВВК'><br>
Год издания:<br> <input type="text" name="Gog_Izd" id='Gog_Izd'><br>
ID_Издательства:<br> <input type="text" name="Izdat_ID" id='Izdat_ID'><br>
ID_Категории:<br> <input type="text" name="p_kateg_ID" id='p_kateg_ID'><br>
Текст:<br> <textarea type="text" name="Text" id='Text'></textarea><br>
<tr>
<td colspan="2"><input type="submit" name="submit" id="submit"
value="OK"></td>
</tr>
</form>
</form>
<p>&nbsp;</p> </td>
</tr>
</table></td>
</tr>
<div id="footer">
<p class="copyright">&copy; Your site 2014 | Created by Asem Omarova | <a
href="https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%8
2%D0%B5%D0%BA%D0%B0">www.aseplib.com</a> </p>
<p></p>
</div>
</div>
</div>
</body>
</html>
```

#### add\_book.php

```
<?php
//include("bd.php");
/* Если существует в глобальном массиве $_POST['title'] опр. ячейка, то мы
создаем простую переменную из неё. Если переменная пустая, то уничтожаем
переменную. */
if (isset($_POST['Nazvanie'])) {$Nazvanie = $_POST['Nazvanie']; if ($Nazvanie
== '') {unset($Nazvanie);}}
if (isset($_POST['Kol_vo_str'])){$Kol_vo_str = $_POST['Kol_vo_str']; if
($Kol_vo_str == '') {unset($Kol_vo_str);}}
if (isset($_POST['Kol_vo_exmemplar'])){$Kol_vo_exmemplar =
$_POST['Kol_vo_exmemplar']; if ($Kol_vo_exmemplar == '')
{unset($Kol_vo_exmemplar);}}
if (isset($_POST['Cena'])) {$Cena = $_POST['Cena']; if ($Cena == '')
{unset($Cena);}}
if (isset($_POST['ISBN'])) {$ISBN = $_POST['ISBN']; if ($ISBN == '')
{unset($ISBN);}}
```



```
if (isset($_POST['UDK'])) {$UDK = $_POST['UDK']; if ($UDK == '')
{unset($UDK);}}
if (isset($_POST['BBK'])) {$BBK = $_POST['BBK']; if ($BBK == '')
{unset($BBK);}}
if (isset($_POST['Gog_Izd'])) {$Gog_Izd = $_POST['Gog_Izd']; if ($Gog_Izd ==
'') {unset($Gog_Izd);}}
if (isset($_POST['Izdat_ID'])) {$Izdat_ID = $_POST['Izdat_ID']; if ($Izdat_ID
== '') {unset($Izdat_ID);}}
if (isset($_POST['p_kateg_ID'])) {$p_kateg_ID = $_POST['p_kateg_ID']; if
($p_kateg_ID == '') {unset($p_kateg_ID);}}
if (isset($_POST['Text'])) {$Text = $_POST['Text']; if ($Text == '')
{unset($Text);}}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сьогодні - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
```

```
<dd>believable. If you are going to use a passage of Lorem Ipsum, you need to be sure there
isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators on
the Internet tend to repeat predefined chunks as necessary, making this the first true generator
on the Internet </dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
<p></p>
<center><b>Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Ou are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
</div>
<?php
if (isset($Nazvanie) && isset($Kol_vo_str) && isset($Kol_vo_exemplar) &&
isset($Cena) && isset($ISBN) && isset($UDK) && isset($BBK) && isset($Gog_Izd)
&& isset($Izdat_ID) && isset($p_kateg_ID) && isset($Text))
{
/* Здесь пишем что можно заносить информацию в базу */
$result = mysql_query ("INSERT INTO Book
(Nazvanie,Kol_vo_str,Kol_vo_exemplar,Cena,ISBN,UDK,BBK,Gog_Izd,Izdat_ID,p_ka
teg_ID,Text) VALUES ('$Nazvanie',
'$Kol_vo_str', '$Kol_vo_exemplar', '$Cena', '$ISBN', '$UDK', '$BBK', '$Gog_Izd', '$
Izdat_ID', '$p_kateg_ID', '$Text')");
if ($result == 'true') {echo "<p>Ваша книга успешно добавлена!</p>";}
else {echo "<p>Ваша книга не добалена!</p>";}
}
```

```
}
else
{
echo "<p>Вы ввели не всю информацию, поэтому книга в базу не может быть
добалена.</p>";
}
?>
</td>
</tr>
</table></td>
</tr>
<!--Подключаем нижний графический элемент-->
</table>
</body>
</html>
contact.php
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сегодня - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
?>
</form>
```

```
</div>
<div id="content">
<p>Контакти<br>
Адрес: Казахстан, 050013,<br>
г.Алматы, пр.Абая 14,<br>
тел.+7 (7272) 67-28-83<br>
e-mail: asem_metis@mail.ru</p>
<div id="footer">
<p class="copyright">&copy; Your site 2014 | Created by Asem Omarova | <a
href="https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%8
2%D0%B5%D0%BA%D0%B0">www.asemlib.com</a> </p>
<p></p>
</div>
</div>
</div>
</body>
</html>
```

#### d.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251"/>
<title>Бібліотека</title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сьогодні - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
```

```
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
<dd>27 ноября 2014 г. в 11.00 ч. в Зале выставок состоится открытие книжно-
иллюстративной выставки «1 декабря - День Первого Президента Республики
Казахстан».</dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
<p></p>
<center><b>Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Ou are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
</div>
<div id="content">
<?php
$id = $_GET['id'];
// ваш адрес где находится, хостится ваша база данных
$sdd_db_host='localhost';
// Имя базы данных с которой вы хотите работать, так как их может быть
множество
$sdd_db_name='lib';
// логин доступ к базе данных
```

```
$sdd_db_user='admin';  
// пароль доступа к базе данных  
$sdd_db_pass='123';  
// устанавливаем связь с сервером  
@mysql_connect($sdd_db_host,$sdd_db_user,$sdd_db_pass);  
mysql_query("set names cp1251"); /*Кодировка*/  
// переключаемся на нужную нам базу данных  
@mysql_select_db($sdd_db_name);  
// делаем выборку из таблицы  
$result=mysql_query('SELECT Text FROM `Book` where Book_ID='.$id.'');  
$row=mysql_fetch_array($result);  
// берем результаты из каждой строки  
echo $row['Text'];  
?>  
</div>  
<div id="footer">  
<p class="copyright">&copy; Your site 2014 | Created by Asem Omarova | <a  
href="https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%8  
2%D0%B5%D0%BA%D0%B0">www.aseplib.com</a> </p>  
<p></p>  
</div>  
</div>  
</div>  
</body>  
</html>
```

#### **delete.php**

```
<?php  
$id = $_GET['id'];  
// ваш адрес где находится, хостится ваша база данных  
$sdd_db_host='localhost';  
// Имя базы данных с которой вы хотите работать, так как их может быть  
множество  
$sdd_db_name='lib';  
// логин доступа к базе данных  
$sdd_db_user='admin';  
// пароль доступа к базе данных  
$sdd_db_pass='123';  
// устанавливаем связь с сервером  
@mysql_connect($sdd_db_host,$sdd_db_user,$sdd_db_pass);  
mysql_query("set names cp1251"); /*Кодировка*/  
// переключаемся на нужную нам базу данных  
@mysql_select_db($sdd_db_name);  
// делаем выборку из таблицы  
$result=mysql_query('call del_b('.$id.')');  
if ($result == null) {echo "ошибка";} else {echo "Запись №".$id." успешно  
удалена!";}  
?>
```

#### **search.php**

```
<?php
header('Content-type: text/html; charset=windows-1251');
if (isset($_POST['search'])) {$search=$_POST["search"];if ($search == '')
{unset($search);}}
if (isset($_POST['submit_s']))
{
$submit_s = $_POST['submit_s'];
}
if (isset($submit_s))
{
if (empty($search) or strlen($search) < 3)
{
exit("<p>Поисковый запрос не введен, либо он менее 3-х символов.</p>");
}
//$search = trim($search);
//$search = stripslashes($search);
//$search = htmlspecialchars($search);
}
else
{
exit("<p>Вы обратились к файлу без необходимых параметров.</p>");
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251" />
<title><?php echo $search; ?></title>
<link rel="stylesheet" type="text/css" media="screen" href="style.css" />
</head>
<body>
<div id="container">
<div id="top_nav"><p class="topnav"><?php
echo "Сегодня - ".date("d F Y")."<br>";
echo "Текущее время - ".date("H:i:s");
?></p></div>
<div id="header"><h1>Бібліотека</h1></div>
<div id="main_nav">
<div id="navcontainer">
<ul id="navlist">
<li><a href="index.php" class="home"><span>Головна</span></a></li>
<li><a href="t2.php" class="book"><span>Книги </span></a></li>
<li><a href="contact.php" class="about"><span>Контакти</span></a></li>
</ul>
<form action="search.php" method="post" name="form_s" id="forms">
<p class="search_t">Поисковый запрос должен быть не менее 4-х
символов.</p><br>
```

```
<p><input name="search" type="text" size="25" maxlength="40">
<input class="search_b" name="submit_s" type="submit" value="Искать">
</p>
</form>
</div>
</div>
<div id="contentwrapper">
<div id="left">
<h3>Авторизація </h3>
<form id="form1" method="post" class="contact_us" action="">
<?php
include("bd.php");
?>
</form>
<h3>Новини</h3>
<dl>
<dt class="news" align="center">26.11.2014 </dt>
<dd> There are many variations of passages o</dd>
<dt class="news" align="center">24.11.2014 </dt>
<dd>Ou are going to use a passage of Lorem Ipsum, you need to be sure there
isn't a.</dd>
<dt class="news" align="center">21.11.2014</dt>
<dd> There are many variations of passages of Lorem Ipsum available, but the majority have
suffered alteration in some form, by injected humour, or randomised words which don't look
even slightly believable. If you are going to use a passage of Lorem Ipsum, you need to be sure
there isn't anything embarrassing hidden in the middle of text. All the Lorem Ipsum generators
on the Internet tend to repeat predefined chunks as necessary, making this the first true
generator on the Internet. It uses a dictionary of over 200 Latin words, combined with a handful
of model sentence structures, to generate Lorem Ipsum which looks reasonable. The generated
Lorem Ipsum is therefore always free from repetition, injected humour, or non-characteristic
words etc.</dd>
</dl>
</div>
<div id="right">
<h4>Новинки </h4>
<p></p>
<center><b>Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Ou are going to use a passage <br>
Год: 2014</p>
<p></p>
<center><b align="right">Ou are going to use a passage </b></center>
<p>Автор: Ou are going to use a passage <br>
Жанр: Наукова література<br>
Год: 2014</p>
<h4>More info</h4>
<p>privet</p>
```