

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,  
управління та адміністрування  
Кафедра інформаційних технологій

**Кваліфікаційна робота бакалавра**

на тему: Розробка веб-додатку «Music shop»  
\_\_\_\_\_

Виконав студент групи КН-20  
спеціальності 122 Комп'ютерні науки  
Загарін Костянтин Юрійович

Керівник Ткач Т.Б., к.ф.-м.н.

Консультант \_\_\_\_\_

Рецензент Перелигін Б.В., к.т.н.,  
доцент

Одеса 2023

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ТЕРМІНІВ І ПОЗНАЧЕНЬ .....	5
ВСТУП .....	6
1 СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНТЕРНЕТ-МАГАЗИНУ .....	8
1.1 Аналіз предметної області .....	8
1.2 Аналіз аналогів.....	10
1.3 Вимоги до системи.....	14
2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ПРОЕКТУ.....	16
2.1 Аналіз HTML.....	16
2.2 Аналіз CSS .....	19
2.2.1 Правила CSS .....	21
2.3 Мови програмування JavaScript .....	22
2.4 MySQL.....	28
2.5 Бібліотеки ReactJS.....	35
3 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	39
3.1 Формування технічного завдання до системи .....	39
3.2 Проектування системи.....	39
3.3 Макет веб-сторінки.....	43
3.4 Проектування системи баз даних .....	49
4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ.....	55
4.1 Керівництво користувача .....	55
4.2 Керівництво адміністратора.....	58
ВИСНОВКИ.....	61
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ.....	62

**ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ТЕРМІНІВ І ПОЗНАЧЕНЬ**

HTML – Hypertext Markup Language.

CSS – Cascading Style Sheets.

SQL – Structured Query Language.

SEO – Search Engine Optimization.

БД – База Даних.

РБД – Розподілені Бази Даних.

СКБД – Система Керування Базами Даних.

CD – Compact Disc.

DVD – Digital Versatile Disc.

## ВСТУП

У нашому світі музика завжди займала особливе місце – це частина нашого життя. Вона є філософією та психологією людини, способом людини передати свої почуття світу. І метою даного дипломного проекту – це створення музичного онлайн центру «Sonic Highways» де можна прослухати та придбати найрізноманітніші вінілові платівки, касети, CD та DVD диски з музикою з широким спектром вибору.

Великий вибір музичних альбомів різного жанру та напрямку від виконавців блюзу кінця 19 століття, до сучасних музикантів, які зараз починають займатись музикою. Для цього треба створити інформаційну систему з максимально зручним та зрозумілим для звичайного користувача інтерфейсом з яким у користувачів не будуть виникати проблеми з навігацією по сайту. А для формування у відвідувачів уявлення про діяльність цього онлайн центру та його можливості, сайт надавав би всю потрібну документацію відео та фотоматеріали.

Для реалізації поставленої мети були виділені основні задачі розробки програмного продукту:

- обрати програмні засоби, за допомогою яких буде реалізовуватись система спортивного центра;
- зробити аналіз існуючих аналогів та зібрати необхідні матеріали, які дозволили би створити всі необхідні умови для зручної роботи користувачів з програмою;
- спроектувати роботу системи;
- створити надійну функціональну оболонку програмного продукту, яка дозволяла би працювати системі без помилок, а також з максимальною швидкістю;
- реалізувати зрозумілу для адміністраторів логічну модель, на якій буде показано структуру та всі взаємодії на сайті;

- створити базу даних для сайту, для ефективного зберігання, оновлення та доступу до даних;
- розробити документацію для користувача, що допоможе зрозуміти роботу сайту і не заплутатись у навігації;
- розробити документацію адміністратора для підтримання функціоналу сайту, тобто для безпомилкового виконання певних операцій.

Дипломна робота містить в собі 60 сторінок, 20 рисунків, 10 посилань.

# 1 СПЕЦИФІКАЦІЯ ВИМОГ ДО ІНТЕРНЕТ-МАГАЗИНУ

## 1.1 Аналіз предметної області

Веб-сайт це один або кілька веб-документів, пов'язаних між собою одним доменним ім'ям. Веб-документи також мають певне призначення, тему та дизайн. Компанії фірми створюють веб-сайти в основному в комерційних цілях для залучення клієнтів. Приватні особи створюють їх для особистого користування[1].

Розглянемо складові веб-сайту.

Дизайн – перше, на що ми звертаємо увагу при відвідуванні сайту, це його візуальне оформлення. Це означає, що шрифти, колірні схеми, кнопки, розділи та загальний стиль сайту повинні правильно поєднуватися та ідеально підходити до його теми та змісту.

Зміст – власник проекту розміщує на сторінці статті, фотографії та ілюстрації, посилання, відео та аудіо матеріали, щоб інформувати користувачів про тему проекту.

Код – все, що відбувається при натисканні на будь-яку кнопку або посилання, кодується програмістом. Завдяки йому користувач перенаправляється на потрібну сторінку, а на сервері виконуються віддалені команди. Втім, все працює так, як задумано творцями.

Веб-сайт створюється для таких цілей як:

- онлайн присутність – веб-сайти дозволяють організаціям, компаніям та індивідуальним користувачам бути присутніми в Інтернеті. Вони допомагають представити інформацію про продукти, послуги, професійний досвід, контактні дані та іншу важливу інформацію для клієнтів і відвідувачів;
- маркетинг і реклама – веб-сайти є потужним інструментом для маркетингу і реклами. Вони дозволяють підприємствам просувати свої продукти і послуги, привертати нових клієнтів, залучати увагу до

- свого бренду та збільшувати продажі. Надають для клієнтам можливість зробити покупку з будь-якого місця, де є доступ до Інтернету;
- збір інформації – веб-сайти можуть слугувати як інструмент для збору інформації від користувачів. Наприклад, компанії можуть використовувати форми зворотного зв'язку для отримання запитів, коментарів і пропозицій від клієнтів;
  - підтримка клієнтів.

Найпопулярніші типи сайтів:

- сайт-візитка – різноманітні сторінки з інформацією про компанію, її діяльність, проекти та послуги.
- лендінг – односторінкові веб-сайти, що мають на меті надати інформацію про продукти, послуги та контактну інформацію
- інтернет-магазини;
- продажні створюються з метою продати товар.

Головне призначення програмного продукту – це надання інформації користувачам у мережі інтернет. У наш час інтернет-магазин дозволяють покупцям цілодобово шукати, переглядати, вибирати та замовляти товари або послуги зручним способом, не виходячи з дому, включаючи зображення, описи, ціни, рейтинги тощо [2].

Покупці в будь який час можуть використовувати функцію пошуку та фільтрації, щоб знайти конкретний товар або послугу за різними параметрами, такими як категорія, ціновий діапазон, рейтинг тощо. І тут же зробити замовлення, заповнити необхідну інформацію, таку як адреса доставки, обрати зручний спосіб оплати, для оформлення замовлення.

Інтернет магазин має перевагу над іншими способами торгівлі в тому, що він не має географічних обмежень. Очевидно, що ніхто не поїде в звичайний магазин, розташований в іншому кінці міста. Якщо тільки там не продається якийсь унікальний товар. У більшості випадків звичайні магазини відвідують ті покупці, які проживають або працюють десь поблизу. Клієнту не

потрібно їздити по декількох магазинах у пошуках потрібного товару за вигідною ціною.

З усією інформацією він може ознайомитися на сайті, не виходячи з дому. Інтернет-магазини здатні приймати замовлення 24 години на добу без обіду та вихідних в зручний для клієнта час. Інтернет-магазин може працювати на всю країну і навіть на весь світ. У нього немає такої жорсткої регіональної прив'язки. Це, безсумнівно, розширює коло потенційних покупців. Програмний продукт призначений ознайомлювати користувачів з товарами Sonic Highways та надавати можливість за користувачам оформлювати та сплачувати замовлення цих товарів. Також веб-ресурс відповідає за просування товарів на ринку, збільшення обсягів продажу та залучення нових покупців, які активно користуються мережею інтернет.

## 1.2 Аналіз аналогів

Discogs – це онлайн-база даних та ринок для музичних записів, включаючи вінілові платівки, CD, касети та інші формати. На сайті користувачі можуть шукати, додавати та купувати музику, а також спілкуватися з іншими колекціонерами та любителями музики (рис.1.1).

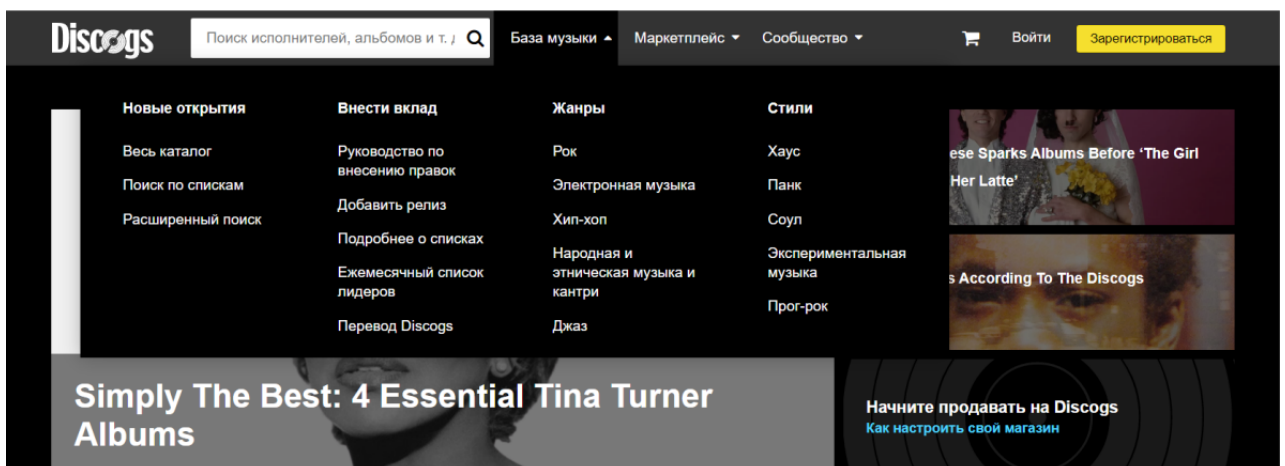


Рисунок 1.1 – Сторінка сайту Discogs



Для прикладу розглянемо його вид. Дизайн та інтерфейс сайту створений по формулі – стислість сестра таланту. Дуже просте оформлення без овердизайну, максимум яскравих елементів та мінімум впливаючий вікон на головній сторінці сайту. Це може бути плюсом та мінусом одночасно. Плюс в тому що сайт не зустрічає користувача перевантаженою панеллю функцій, і тим самим не відлякує потенційного клієнта. З іншого боку це мінус тому що всю іншу потрібну інформацію приховують в випадаючому меню, та інколи це виглядає не коректно. Наприклад поряд із search box «Пошук», де можна знайти потрібний товар по даті жанру тощо, є вкладка с базою музики де також можна шукати потрібний товар по жанрам, стилям і т.д.

Два елемента з однаковими функціями поруч один з одним, але один має вигляд випадаючого меню, а інший це search box. Виглядає як не зовсім доцільне заповнення та використання простору сторінки. Але пошук на цьому сайті якісний. З інших переваг є:

- пошук і браузінг – Discogs має потужний пошуковий інструмент, який дозволяє користувачам знаходити музику за артистами, назвами альбомів, лейблами тощо. Користувачі можуть також переглядати різні чарти, новинки та рекомендації;
- колекція та додавання записів – користувачі можуть створювати свою власну музичну колекцію на Discogs, додаючи записи до свого профілю. Вони можуть вказувати деталі про стан запису, видачу, обкладинку та інші характеристики;
- купівля та продаж – Discogs також служить як музичний ринок, де користувачі можуть купувати та продавати записи. Є можливість укладання угоди безпосередньо між продавцем та покупцем, а також захисту покупців у разі проблем з транзакціями;
- спілкування та громадськість – Discogs має соціальну складову, де користувачі можуть спілкуватися один з одним, залишати відгуки,

обговорювати музику та розділяти свої знання і досвід з колекціонування.

- дизайн та інтерфейс – зручність використання сайту, легкість навігації та привабливість дизайну;
- спілкування та спільнота – оцінка функцій спілкування та взаємодії між користувачами, включаючи форуми, коментарі та рейтинги.

**Queen – Bohemian Rhapsody / Sweet Lady**

Лейбл: [Max \(3\) – 049](#)

Формат: [Пластинки, 7", 45 RPM, Single](#)

Страна: [Turkey](#)

Дата релиза: [1976](#)

Жанр: [Rock](#)

Стиль: [Pop Rock](#)

[Все изображения](#)

**Содержание** Скрыть сведения об участниках записи

A	Bohemian Rhapsody Written-By – <a href="#">Mercury*</a>	5:48
B	Sweet Lady Written-By – <a href="#">May*</a>	3:58

**Компании и т. п.**

Дистрибьютор – [Max Plakçılık](#)

**Заметки**

The song "Bohemian Rhapsody" misprinted on back cover as "Bohemia Rhapsody".

From LP. A Night At The Opera.

**Релиз** (639990)

[Редактировать релиз](#)  
[Все версии этого релиза](#)  
Обновлен недавно

[Добавить в коллекцию](#)

[Добавить в желаемое](#)

**Маркетплейс** 2 в продаже от 249,90 \$

[Купить винил](#) [Продать винил](#)

**Статистика**

Есть у:	46	Последний:	30 апр. 2023 г.
Хотят:	255	Минимум:	53,74 \$
Рейтинг:	4 / 5	В среднем:	376,34 \$
Оценок:	13	Максимум:	510,75 \$

☆☆☆☆☆ [Поделиться](#)

Рисунок 1.2 – Проблеми з відображенням реклами

Як показано на рисунку (рис.1.2) вище, на сайті присутні проблеми с пустими ділянками сторінок що ніяк не використовуються. На другому рисунку це пов'язано з неправильним відображенням блоків з рекламним контентом. Це не результат роботи Google Ads, який скриває рекламу, тому що в такому випадку ці сірі вікна зникли б з сторінки або замінилися на інший контент. Проблема саме в непровантажуваності певних блоків сайту.

Але в усьому іншому інформація представлена правильно – рік, жанр, назва, виконавець, формат, хто приймав участь у створенні. Вказана ціна і

також рейтинг серед клієнтів. Знизу знаходяться рекомендації з іншими альбомами того ж виконавця, і коментарі користувачів.

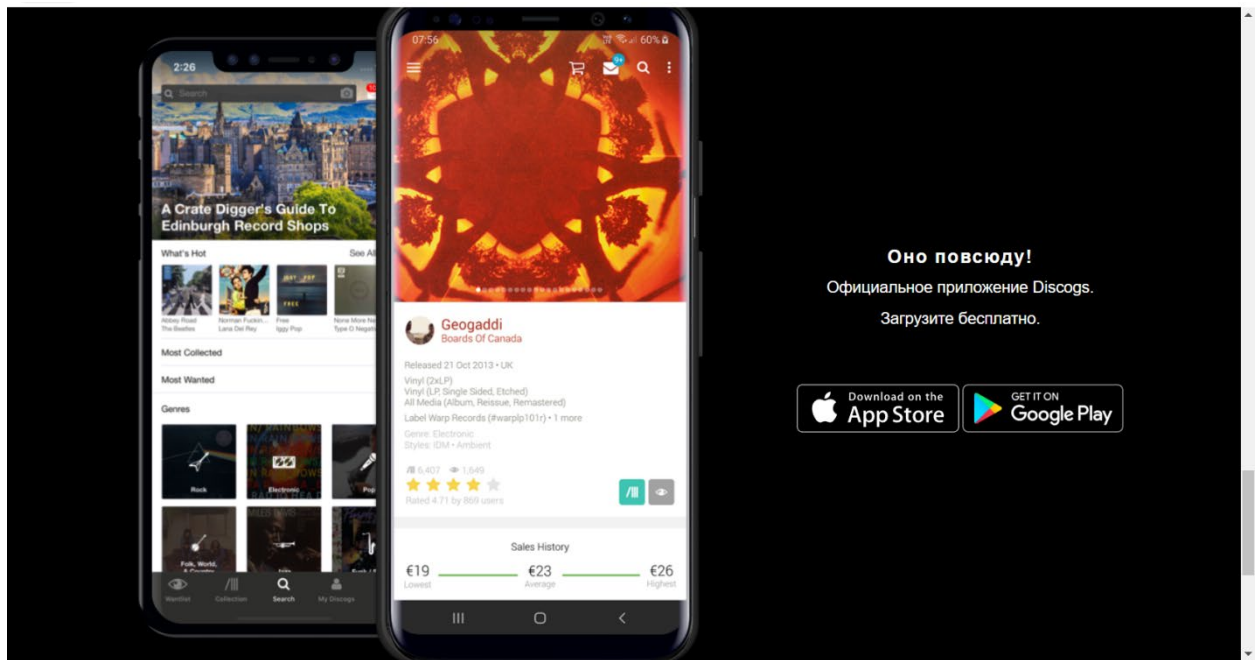


Рисунок 1.3 – Мобільна версія

Окремої уваги треба приділити й мобільній версії цього сайту (рис.1.3). Якщо опуститися в футер сторінки, то можна побачити додаток для телефонів. Здається що він більш опрацьований, ніж комп'ютерна версія. Тому що на мобільній версії більше елементів, а ніж для комп'ютерної. Відчувається легкий овердизайн, тому що в очі одразу кидається велика різноманітність меню, кнопок, переходів та іншого, що для екрану телефону почувається занадто. Всього цього не хватає для варіанту сайту на екрані монітору комп'ютеру.

Це може свідчити про те, що цей сайт розроблявся в першу чергу для мобільних пристроїв, як зручний додаток що можна відкрити в будь-який час, в будь-якому місті. Або навпаки – після розробки сайту для зручного користування комп'ютером з'явився час та ресурси на якісну і детальну розробку адаптивних версій.

### 1.3 Вимоги до системи

Специфікація вимог, також відома як документація, – це процес запису всіх системних і користувацьких вимог у письмовому вигляді. Ці вимоги повинні бути чіткими, повними, всеосяжними і послідовними. На початку нашої роботи ми збираємо всі вимоги з різних джерел. Під час аналізу та переговорів ми аналізуємо та розуміємо ці вимоги.

Тепер необхідно скласти офіційний документ, що описує ці вимоги. Це те, що ми називаємо вимогами до проекту. Точніше, це процес чіткого і точного документування всіх потреб і обмежень користувачів і системи. На різних етапах розробки програми програмний продукт повинен відповідати всім визначеним вимогам, зазначеним у технічному завданні.

У першу чергу мають бути виявлені та виправлені усі помилки в системах та функціях, які доступні всім користувачам (фронтенд) так і внутрішня логіка сайту що схована від звичайних користувачів (бекенд). Має бути гарний дизайн сайту, щоб у потенційних клієнтів виникало бажання користуватися послугами саме цього сайту, також зручного і простого функціоналу.

До нього належать: пошук, сортування, фільтрування, підбір товарів або послуг, функція кошика бажань. Також це може стрічка новин та подій, комунікація з клієнтами та користувачами між собою, це все покращує якість обслуговування і приваблює клієнтів. Для того, щоб сайт інтернет-магазину міг відкриватись з різних пристроїв – необхідно створити адаптовану верстку.

Інтернет-магазин повинен видавати користувачу готову веб сторінку, сформовану за запитом користувача на сервері, яка відображає необхідну інформацію та кнопки для здійснення наступних запитів.

Програмний продукт повинен виконувати наступні дії:

- зручний та легкодоступний каталог товарів, де товари будуть групуватися за категоріями, підкатегоріями та мітками. Каталог повинен

мати можливість додавання нових товарів, описів, зображень, характеристик;

- пошук та фільтрація: магазин повинен мати функцію пошуку, щоб користувачі могли швидко знайти потрібний товар;
- корзина покупок: користувачі повинні мати можливість додавати товари до корзини покупок, переглядати та редагувати вміст корзини та оформляти замовлення;
- зручна система оплати;
- авторизації на сайті;
- вартість товару;
- контактна інформація та підтримка.

В результаті розробки вимог до інтернет-магазину було проведено аналіз предметної області, де було визначено принцип розробки та роботи сайту-магазину, види таких сайтів. Проведено аналіз аналогів інтернет-магазинів. І визначено вимогу до функціоналу системи, які функції повинні бути у системі для її найкращої роботи.

## 2 АНАЛІЗ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОЗРОБКИ ПРОЕКТУ

### 2.1 Аналіз HTML

HTML (Hypertext Markup Language – мова гіпертекстової розмітки) – це мова, яка використовується для опису структури сторінок документа, дозволяючи формувати звичайний текст в абзаци, заголовки, списки та інші структури, а також посилання на інші сторінки. Це текстова мова, в якій інструкції форматування, що називаються тегами, розміщуються в розділах документа, які містять певну інформацію. Теги вказують браузеру, як інформація повинна бути відформатована і представлена на екрані [3].

HTML є стандартною мовою розмітки для створення веб-сторінок і веб-документів. Вона використовується для структурування та представлення вмісту в Інтернеті. HTML визначає структуру документа за допомогою різних тегів і елементів, що описують різні частини веб-сторінки.

HTML це одна з основних мов для розробки веб-сторінок разом з CSS (Cascading Style Sheets) і JavaScript. CSS використовується для стилізації веб-сторінок, а JavaScript – для програмування розумного поведінки сторінки. Разом ці три технології (HTML, CSS і JavaScript) створюють сучасний веб-досвід.[4]

«Для чого взагалі потрібен цей HTML?» Відповідь дуже проста – комп'ютери дурні за своєю природою. Якщо ви скажете певну фразу людині, вона зрозуміє, що вона звернена до неї і що від неї чекають саме цього. Комп'ютери так не вміють. Тому потрібно пояснити, кому саме адресована інформація, що вона означає і що з нею робити. Тільки тоді комп'ютер може жити.

За допомогою HTML можна створювати структуру документа, вставляти текст, зображення, посилання, таблиці, форми та інші елементи, що роблять веб-сторінки зручними та естетично привабливими.

Основною одиницею в HTML є елементи. Елементи складаються з тегів, які визначають тип елемента. Теги в HTML записуються в кутових дужках (< >). Наприклад, <p> – це тег для визначення абзацу, <h1> – тег для заголовка першого рівня тощо.

Крім тегів, HTML також використовує атрибути, які надають додаткову інформацію про елементи. Що є атрибутами у HTML коді?

Атрибути – це поля даних, які посилаються на файл, але не є його частиною. Вони не враховуються при обчисленні розміру файлу і можуть бути скопійовані або видалені без зміни самого файлу. Атрибути вказуються всередині тегів і мають значення, яке вказується в подвійних або одинарних лапках. Наприклад,  – тег <img> використовує атрибути src і alt для відображення зображення та альтернативного тексту відповідно (рис. 2.1).

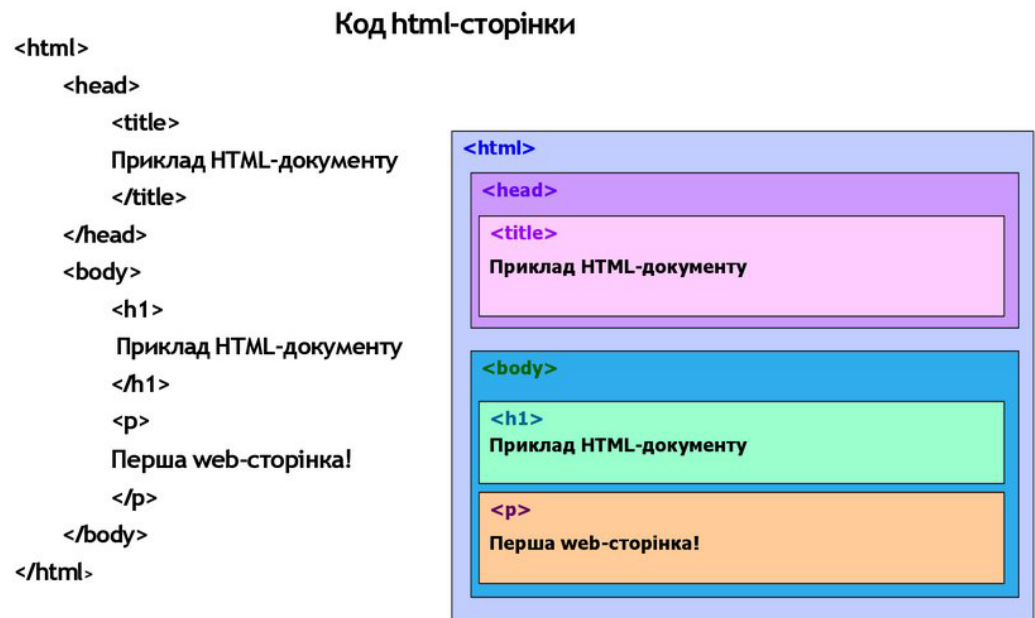


Рисунок 2.1 – Приклад шаблону в HTML

HTML має кілька переваг, які роблять його популярним і широко використовуваним для розробки веб-сторінок:

**Простота вивчення та використання.** HTML має простий синтаксис і легко вивчається навіть для початківців. Він базується на тегах і елементах, що робить його зрозумілим і доступним.

**Безкоштовність.** Однією з найбільших переваг HTML є те, що вона є безкоштовною і не потребує придбання спеціального програмного забезпечення: Оскільки HTML не вимагає плагінів, немає необхідності підтримувати різні плагіни, необхідні для роботи будь-якого програмного забезпечення. Тому розробка всього веб-сайту на HTML є дуже економічно вигідною з точки зору бізнесу, оскільки не потрібно платити за ліцензію.

**Універсальність.** HTML є стандартом для розмітки веб-сторінок і підтримується всіма сучасними веб-браузерами. Це означає, що створені з використанням HTML сторінки будуть відображатись однаково на різних пристроях і веб-браузерах.

**Багатофункціональність.** HTML можна використовувати на будь-якій платформі, будь то комп'ютери, мобільні пристрої чи інші пристрої з веб-браузерами. Це робить HTML універсальним засобом створення веб-сторінок для різних платформ в будь-яких браузерах.

**Сумісність з пошуковими системами.** HTML має структуровану натуру, що дозволяє пошуковим системам легко індексувати та розуміти вміст сторінок. Це сприяє покращенню SEO (оптимізації для пошукових систем) і забезпечує кращу видимість веб-сторінок у пошукових результатах.

**Підтримка мультимедіа.** HTML дозволяє вставляти зображення, аудіо, відео та інші мультимедійні елементи безпосередньо на веб-сторінки. Це дає можливість створювати мультимедійний контент і поліпшувати взаємодію з користувачами.

**Розширюваність.** HTML підтримує використання CSS і JavaScript, що дозволяє створювати більш складні та динамічні веб-сторінки. CSS дозволяє



стилізувати сторінки, а JavaScript – надавати їм інтерактивність та функціональність.

## 2.2 Аналіз CSS

CSS (скорочення від Cascading Style Sheets) – це спеціальна мова (мова таблиць стилів), яка визначає зовнішній вигляд (як і де відображаються елементи веб-сторінки) документів, написаних мовами розмітки даних CSS використовується в документах, розмічених мовами HTML, XHTML і XML. Після того, як пан на ім'я Tim Berners Lee винайшов World Wide Web, мова HTML, деякий час, використовувалася тільки для визначення структури вмісту сторінки. Розробники могли тільки помічати текст: «це – заголовок» чи «це – параграф», використовуючи HTML-теги, такі як `<h1>` та `<p>`. Трішки пізніше почали з'являтися дуже специфічні теги. Ти мабуть ще пам'ятаєш, що були такі теги як `<font>` чи `<bgcolor>` – оформлення веб-сторінок здійснювалося безпосередньо всередині вмісту документа. Ці теги відрізнялися від інших, тим, що вони визначали дизайн, а не структуру, а у більшості інших тегів були такі властивості як `color` чи `border` [5].

Таке прагнення до збільшення стилістичних можливостей призвело до того, що оригінальні теги структурування, такі як `<table>`, стали все більше застосовуватися для дизайну сторінок, замість структурування тексту. Багато нових тегів, такі як `<blink>`, підтримувалися тільки одним браузером. «Вам потрібен браузер X, щоб переглядати цей ресурс» – така відмова стала звичайним явищем на web-сайтах.

Водночас, в якийсь момент стало зрозуміло, що код сторінки стає громіздким і нечитабельним, марним і врешті-решт ніхто не буде ним користуватися. Тому було прийнято рішення відокремити структуру сторінки (HTML) від її візуального оформлення (CSS). Однак після такого розділення

потреба в елементах стилів відпала, і поступово такі теги та атрибути стали застарілими і були видалені розробниками.

Наразі HTML у чистому вигляді має дуже обмежений набір інструментів і не може вирішити специфічні дизайнерські та функціональні завдання, які стоять перед веб-майстрами. Наприклад, найпоширеніше питання всіх веб-майстрів – «Як прибрати підкреслення з посилання?» або «Як змінити стиль посилання при наведенні на нього курсору» Це неможливо зробити за допомогою одного лише HTML. І таких питань безліч. Ось тут і з'являється CSS, оскільки він вирішує більшу частину роботи, пов'язаної зі стилем сторінки.

Використання CSS (Cascading Style Sheets) для стилізації веб-сторінок має численні переваги:

- розділення зовнішнього вигляду від вмісту: CSS дозволяє відокремити стиль і дизайн веб-сторінки від її HTML-структури. Це полегшує підтримку і редагування коду, оскільки зміни в стилях можуть бути внесені централізовано, без потреби змінювати HTML-код;
- декілька дизайнів сторінки для різних пристроїв. Наприклад, на екрані дизайн буде розрахований на велику ширину, під час друку меню не виводитиметься, а на смартфоні меню буде внизу, під вмістом;
- відображати один і той же документ в різних стилях;
- зменшення часу завантаження сторінок сайту за рахунок перенесення правил відображення в окремий CSS-файл. В цьому випадку браузер завантажує тільки структуру документа і дані, що зберігаються на сторінці, а стильові правила цих даних завантажуються браузером тільки один раз і кешуються;
- загальні стилі та зручність управління: CSS дозволяє визначати загальні стилі, які можуть бути застосовані до багатьох елементів одночасно. Це полегшує зміну вигляду всієї веб-сторінки або групи елементів зміною всього лише одного правила стилю;

- відновлюваність і модульність: CSS дозволяє використовувати класи і ідентифікатори для надання стилів окремим елементам. Це дозволяє легко змінювати стилі для конкретних елементів без впливу на решту сторінки. Крім того, CSS може бути організований у модульні файли, що полегшує управління і редагування стилів;
- переносимість та кросс-браузерна сумісність: CSS є стандартною технологією, яка підтримується всіма сучасними веб-браузерами. Це забезпечує консистентний вигляд сторінок на різних платформах і пристроях. Крім того, CSS дозволяє використовувати вендорні префікси для забезпечення сумісності з різними браузерами;
- простота подальшої зміни дизайну. Не потрібно правити кожну сторінку, а лише змінити CSS-файл;
- покращення доступності та SEO: Використання CSS дозволяє відокремити структуру веб-сторінки від її вигляду, що полегшує сприйняття інформації пошуковими системами та технологіями адаптивного читання (наприклад, екрани читачів для осіб з вадами зору). Крім того, CSS надає більше можливостей для оптимізації сторінки для пошукових систем (SEO), таких як використання ключових слів у заголовках і мета-даних.

Ці переваги роблять CSS потужним інструментом для створення привабливих, гнучких і легко керованих веб-сторінок з високою якістю дизайну і функціональністю.

### 2.2.1 Правила CSS

Правила в CSS працюють по каскаду (пріоритету, вазі). Це дозволяє отримати передбачуваний результат у випадку, коли до одного елемента, одночасно, застосовуються декілька стилевих правил.

Структура CSS-правила показана на рис. 2.2.



розробки. Він відмінно інтегрується з версткою та серверною частиною (фронтенд та бекенд) [6].

Усі веб-додатки та веб-сайти створюються за допомогою трьох технологій: HTML, CSS та JavaScript. Остання є «мозком» розробки і керує взаємодією та користувацьким досвідом.

Варто зазначити, що все частіше компанії не обмежуються роботою із JavaScript лише в браузері, а й користуються платформою Node.js, середовищем виконання JavaScript-коду, для написання серверних застосунків. Яскравими прикладами проектів за такої комбінації є Netflix та PayPal.

Тож багатоплановість застосування – це ще кілька балів до перспективності JavaScript.

HTML, CSS і JavaScript – це три стовпи, на яких працюють веб-додатки: HTML – це фреймворк, CSS – це красивий візуальний дизайн (CSS3 дозволяє анімацію), а JavaScript – це логіка, інтерактивність, взаємодія з користувачем. Якщо ви бачите на сторінці динамічні елементи, з якими можна взаємодіяти, наприклад, увімкнути аудіо- чи відеоплеєр або прокласти маршрут на карті, ви можете бути впевнені, що тут задіяний JavaScript. Іншими словами, фронтенд- і JavaScript-інженери (розробники) потребують усіх трьох технологій. Основними особливостями мови є динамічність, гнучкість у роботі з функціями та універсальність, а також можливість створення прототипів. Вона сумісна з усіма сучасними браузерами, легко інтегрується з HTML-макетами і може бути налаштована для зв'язку з серверами.

Серед інших переваг:

- 1) Тип даних визначається, коли змінній або константі присвоюється значення.
- 2) В JavaScript функції можна як виконувати, так і повертати, передавати їх як параметри іншим функціям і привласнювати як значення змінних.
- 3) Методологія об'єктно-орієнтованого програмування дає змогу представити програму у вигляді сукупності об'єктів.

- 4) Дозволяє частково перенести бізнес-логіку із сервера на сторону користувача, тобто виконувати код в браузері, що своєю чергою зменшує навантаження на сервери.

JavaScript має розвинену інфраструктуру та активну спільноту. Так, веб-розробники можуть працювати з великою кількістю бібліотек і фреймворків як React, Angular і Vue, декількома пакувальниками, як Webpack, Gulp, та допоміжними бібліотеками як Lodash, axios, та іншими.

З одного боку, JavaScript – це просто, адже результати можуть відображатися в браузері. З іншого боку, це означає постійні оновлення, які потрібно відстежувати.

Є кілька обмежень, про які варто згадати. Найважливішим з них є те, що він не може працювати з файловою системою, тобто не може читати файли. Крім того, JavaScript не підтримує віддалений доступ до системи, що робить його непридатною мовою для використання в мережевих додатках.

Особливу увагу потрібно звертати на захист на стороні клієнта, адже деякі недогляди й помилки в коді можуть використати зловмисники.

Код JavaScript також по-різному інтерпретується різними браузерами, а старіші версії, такі як IE 9 (Internet Explorer 9), взагалі не підтримуються. Деякі браузери читають код JavaScript дещо по-різному, що може призвести до того, що веб-сайти та веб-додатки не відобразатимуться або працюватимуть некоректно. Динамічний набір тексту також іноді вважається недоліком. Однак цю проблему можна вирішити за допомогою TypeScript (транскриптора JavaScript).

На JavaScript вплинуло багато мов, і метою було створити мову, схожу на Java, але водночас просту у використанні для непрограмістів. JavaScript має багато рис об'єктно-орієнтованої мови, але відрізняється від традиційних мов через спосіб реалізації прототипування. Він відрізняється від традиційних об'єктно орієнтованих мов способом роботи з об'єктами. JavaScript також має риси, характерні для функціональних мов (функції як першокласні об'єкти, об'єкти як списки, переривання, анонімні функції, закриття тощо)

JavaScript – це інтерфейсна мова, але не єдина: Переважна більшість JS-розробників – це фронтенд-розробники, тобто люди, які пишуть чисті програми, що запускаються в браузері користувача. Наприклад, якщо ви коли-небудь користувалися Gmail або веб-додатком (наприклад, розрахунок рахунку або обробка документів у Вчасно), то це односторінковий додаток. Це означає, що ви нікуди не переходите зі сторінки, а все, що вам потрібно зробити, робиться в додатковому вікні, все це написано на JavaScript з використанням різних фреймворків. Наприклад, Gmail написаний на Angular, розробленому Google, Facebook – на React і так далі.

У бекенді JavaScript знаходиться на стороні сервера. Тут він використовується як `node.js`, движок V8, який береться з браузера (на якому власне і працює JavaScript) і розміщується на сервері. Таким чином, він виконує всі команди JS, є досить швидким і працює досить добре.

Зараз Java майже повністю перейшла у великомасштабну бекенд-розробку. Точніше, в корпоративну розробку, тобто автоматизацію на великих підприємствах. Багато хто каже, що Java є специфічною для backend, але це не зовсім так.

Зазвичай бекенд і фронтенд мають однаковий розмір; якщо ми говоримо про Java, бекенд набагато більший за фронтенд, тому фронтенд зазвичай не розглядається. Але бувають і винятки. Наприклад, звичайний додаток, написаний на Java, має веб-інтерфейс. Якщо ви представляєте велику компанію з одним, двома або тридцятьма тисячами співробітників, було б дуже дорого встановити спеціальне програмне забезпечення для кожного з них. За ним потрібно стежити, оновлювати тощо. Зробивши це програмне забезпечення веб-інтерфейсом, його можна використовувати з браузера.

На Java написані величезні додатки, фактично найбільші додатки в нашій індустрії. Майже кожен банк, кожна телекомунікаційна компанія, кожна фінтех-компанія, кожна платіжна система тощо. написані на Java. Виняток становлять стартапи, які зазвичай пишуть на Node.js. Всі великі компанії мають Java під капотом: Shell, Oracle, Wall Street Journal і т.д. Чому? Тому що

фундамент повинен бути стабільним і гарантовано працювати, а Java для цього підходить якнайкраще.

Синтаксис Java та JavaScript також цікавий. Обидві мови схожі на C. Це означає, що Java-програмісти можуть легко читати JavaScript-код. Звичайно, якщо цей код написаний правильно. Насправді, JavaScript – дуже вільна мова і, навпаки, Java – дуже сувора мова; Java вимагає, щоб все було так, як має бути, тому для Java-програмістів дуже важливо вміти писати код на JavaScript.

Ім'я файлу повинно бути таким же, як і у класу, і мати такий же розмір. Класи повинні бути розташовані там, де вони повинні бути, тобто в одному каталозі. Ім'я каталогу має бути однаковим. Якщо вона знаходиться десь в іншому місці, в неправильному стані або іншим чином неправильна, то нічого не буде скомпільовано. Java дуже сувора в цьому питанні. Саме тому на Java пишуться величезні системи, тому що навіть маленький компонент не скомпільується, якщо він написаний трохи неправильно. Будь-який код, навіть якщо він був написаний 20 років тому, зараз можна легко підтримувати.

JavaScript – протилежність цьому. Ви можете вводити все, що завгодно. Він динамічний, і ви можете писати дати, числа, слова, картинки і навіть дати в одній змінній, змішуючи їх між собою. Більше того, JS також є мультипарадигмальною мовою, тому ви можете писати майже все, що завгодно. В результаті, якщо ви не дуже добре знаєте JS, ви знайдете якийсь код абсолютно незрозумілим. Вам доведеться перестрибнути в іншу парадигму, щоб зрозуміти його. Крім того, мова дуже вільна, тому ви можете переписувати її самостійно.

В результаті, якщо ви пишете JavaScript на певному фреймворку, такому як jQuery, цей код виглядає зовсім не схожим на JavaScript. Таким чином, JS дозволяє програмістам створювати все, що вони хочуть. Це універсальна мова, і її можна розробляти дуже швидко, тому що її дуже легко і безкоштовно змінювати. З іншого боку, мова не дозволяє розробляти дуже великі системи. Це пов'язано з тим, що кожен програміст пише у власному стилі, що



ускладнює розуміння для всіх. З іншого боку, в Java все строго, все базується на правилах, і якщо ви зробите крок вбік, треба буде перероблювати все знову.

З них Java має вбудовану підтримку безпеки і є мовою вибору для всіх фінтех-компаній, оскільки безпека важлива, особливо тому, що JavaScript має проблеми з безпекою. Однак її можна дуже швидко розробити.

Для обох мов створено багато фреймворків і бібліотек. Однак справедливо і зворотне: Java – це основна, серйозна мова зі зворотною сумісністю. Фактично, все, що розроблено на Java, пережило десятиліття.

Ця мова стикається з дуже серйозними викликами, тому що автоматизація в гігантських компаніях розробляється за допомогою Java. Як наслідок, Java – єдина мова у світі, яка має цілий стек фреймворків, і коли ви ставите один фреймворк на інший фреймворк, інший фреймворк стає ще одним фреймворком. Більше того, існують фреймворки, які з'єднують всі ці фреймворки.

Java має велику кількість глибоко розроблених, повноцінних рішень для всіх видів завдань. Навіть готові рішення вимагають багато часу для розуміння, оскільки вони призначені для вирішення дійсно складних завдань. Однак вони можуть все. Більше того, майже для кожної задачі існує не одне повноцінне рішення, а кілька альтернатив. І важливо знати, яка альтернатива є найкращою – Java така ж велика, як і Всесвіт, адже на Java вже дуже давно все пишеться. І потрібно знати, як з усіма ними працювати.

Це тому, що JavaScript такий же великий, як і Всесвіт, і для кожної задачі написані десятки або сотні бібліотек. У них також необхідно розбиратися: JS фреймворки поділяються на фронтенд фреймворки: Angular, React і Vue та бекенд: node.js.

Обидві мови нікуди не зникнуть у найближчі 30 років. Обидві мають багато написаного коду і багато шанувальників; JavaScript зараз має відкритий вихідний код і повністю підтримується.

## 2.4 MySQL

MySQL – це система управління базами даних, що використовується для підтримки реляційних баз даних; це програмне забезпечення з відкритим вихідним кодом, яке підтримується корпорацією Oracle. Спочатку її заснувала шведська компанія MYSQL AB, пізніше її придбала Solar Microsystems і, нарешті, корпорація Oracle. Як система баз даних з відкритим вихідним кодом, вихідний код може бути змінений відповідно до наших потреб. Якщо ми купуємо комерційну ліцензію від Oracle Corporation, ми також отримуємо преміум-сервіси. MySQL – це масштабована, швидка і надійна система управління базами даних, яка працює на будь-якій платформі, включаючи Windows, Unix і Linux, і може бути встановлена як на робочому столі, так і на сервері. Її можна встановити на будь-яку настільну або серверну машину. Її також дуже легко вивчити порівняно з іншими системами управління базами даних, доступними на ринку, такими як Microsoft SQL Server, Oracle Database тощо, а також PHP. LAMP – Linux як операційна система, веб-сервер Apache, система управління реляційними базами даних MySQL та об'єктно-орієнтований скрипт PHP [7].

Існує багато топових сайтів, які використовують MySQL. Існує також багато компаній, які використовують MySQL як систему управління реляційними базами даних. Приклади включають YouTube, Facebook, Twitter та ін. MySQL працює за моделлю клієнт-сервер, і сервер MySQL є головним сервером, який обробляє всі команди.

Системою управління базами даних MySQL може користуватися будь-хто, хто має базові знання про структуру мови запитів. Це найпопулярніше програмне забезпечення для управління базами даних з відкритим вихідним кодом, яке можна інтегрувати з багатьма мовами програмування, включаючи Java та Python. На ринку MySQL існує багато можливостей для кар'єрного росту у великих компаніях, таких як Yahoo та Suzuki.

Невеликі переваги використання MySQL такі:

- масштабованість – MySQL пропонує чудову масштабованість для керування та налаштування глибоко вбудованих додатків, що займають менше місця, навіть у великих сховищах з великими обсягами даних. Гнучкість на вимогу є важливою особливістю MySQL: оскільки MySQL – це рішення з відкритим вихідним кодом, оператори електронної комерції можуть повністю налаштувати власні вимоги до сервера баз даних;
- безпека – MySQL – одна з найбезпечніших у світі систем управління базами даних, яку використовують більшість популярних веб-додатків, таких як YouTube, Facebook, Twitter і WordPress. Остання версія MySQL забезпечує підтримку та безпеку обробки транзакцій, що є великою перевагою для бізнесу, особливо для будь-якої електронної комерції, яка вимагає переказу великих сум грошей;
- продуктивність – MySQL розроблена для роботи з найвимогливішими додатками, забезпечуючи при цьому достатню швидкість. Для підвищення продуктивності він надає власний кеш пам'яті та повнотекстові покажчики. Тому, будь то сайт електронної комерції, який щодня виконує мільйони запитів, або будь-яка операційна система обробки транзакцій, MySQL надає унікальну систему зберігання, яка дозволяє адміністраторам налаштовувати сервери MySQL без будь-яких недоліків і забезпечувати високу продуктивність сервера MySQL без будь-яких недоліків і забезпечувати високу продуктивність;
- економічна ефективність – забезпечуючи точне управління та надійність, MySQL зменшує час, що витрачається на усунення негараздів, проблеми з продуктивністю та простої, забезпечуючи значну економію коштів для бізнесу;

- немає простоїв – MySQL пропонує широкий спектр рішень високої доступності з конфігураціями реплікації master/slave і виділеними кластерними серверами для забезпечення цілодобової доступності;
- повний контроль робочого процесу – MySQL – це комплексне рішення, яке пропонує розширення домену та функції самоуправління для автоматизації управління базами даних. Він пропонує мінімальний час встановлення і готовий до використання, може бути встановлений на UNIX, Windows, Linux і Macintosh;
- рішення з відкритим кодом – MySQL з відкритим вихідним кодом вирішує різні проблеми, такі як налагодження, обслуговування, швидке оновлення та покращення користувацького досвіду. Безпечна обробка забезпечує ефективну роботу з великими масивами даних;
- транзакційна підтримка – MySQL – це найпопулярніша на ринку система транзакційних баз даних, оскільки вона забезпечує повну цілісність даних. Він забезпечує необмежену кількість блокувань на рівні рядків, ізоляцію, узгодженість і довговічність, повністю підтримує атомарні транзакції і підтримує транзакції між декількома версіями [8].

У мережевому клієнт-серверному середовищі MySQL Server можна використовувати як окрему програму. Він також доступний у вигляді бібліотеки, яку можна підключити до іншої програми; існує кілька корисних додатків, які підтримують управління базами даних MySQL. Клієнт MySQL, з іншого боку, встановлюється на мережевому комп'ютері; інструкції надсилаються від клієнта MySQL до сервера MySQL, і сервер MySQL діє відповідно.

Хоча MySQL встановлюється на одному комп'ютері, він може надсилати бази даних у різні місця, а користувачі можуть отримувати доступ до однієї і тієї ж бази даних за допомогою різних клієнтських інтерфейсів MySQL. Ці інтерфейси передають оператори SQL-серверу, а результати відображаються на екрані. Функціональністю MySQL можна керувати, використовую-

чи лише існуючі команди SQL, тому немає необхідності вивчати нові команди. Реплікація даних і розбиття таблиць також можуть бути виконані в MySQL, що дозволяє користувачам мати кращу продуктивність і вищу довговічність; Для зберігання даних і доступу до них можна використовувати кілька механізмів зберігання, включаючи NDB і InnoDB.

MySQL написана мовами C та C++ і доступна на багатьох платформах, включаючи Windows, Linux та Mac. Ця реляційна система управління базами даних підтримує мільйони записів і безліч типів даних, включаючи беззнакові та знакові цілі числа до 8 байт, двійкові, varchar, double, char, float, time, blob, enum, text, date, year, date, timestamp, багато інших типів даних, таких як varbinary і просторові типи OpenGIS. Також підтримуються типи рядків фіксованої та змінної довжини. Клієнт MySQL може підключатися до серверів MySQL за допомогою ряду протоколів, таких як TCP/IP. Проте, підтримуються різні інструменти керування клієнтами та утиліти, такі як MySQL Workbench та різні програми командного рядка.

Є кілька проектів, розроблених на базі MySQL, як показано нижче:

- Percona Server з XtraDB – це розширена версія MySQL, яка пропонує горизонтальну масштабованість;
- Drizzle – це система управління базами даних з відкритим кодом, яка є легкою і розроблена на основі MySQL 6.0;
- MariaDB – популярна заміна для MySQL, яка використовує API та команди MySQL.

Тому багато проектів, які потребують швидкого відгуку, зберігання інформації та складного відбору даних, розробляються на основі MySQL. Більшість таких розробок – це веб-сайти. Водночас MySQL активно використовується для реалізації як невеликих (наприклад, блоги, сайти-візитки), так і досить великих завдань (наприклад, інтернет-магазини, сховища даних). В обох випадках для відображення інформації на сторінках сайту використовуються запити до MySQL. При створенні запитів розробники намагаються

максимально використати доступні можливості системи управління базами даних.

Дані повинні бути організовані для зручного зберігання та подальшої обробки. Структура даних може визначати, як виглядатимуть таблиці, що використовуються для зберігання інформації. Таблиця бази даних – це набір полів (стовпців), що відповідають за кожну властивість об'єкта даних.

Наприклад, якщо вам потрібно створити таблицю співробітників компанії, її найпростіша структура буде такою: кожному співробітнику присвоюється унікальний номер, і цей номер зазвичай використовується як первинний ключ таблиці.

Далі персональні дані працівника вносяться в таблицю. Це може бути що завгодно ПІБ, номер відділу, номер телефону, адреса тощо. Згідно з вимогами нормалізації (шість нормальних форматів бази даних) та конфігурації запитів MySQL, поля в таблиці повинні бути атомарними, тобто не містити перелічень або списків. Тому, як правило, таблиці мають окремі поля, наприклад, ім'я, прізвище тощо.

На рис.2.3 наведено тривіальний приклад структури таблиці бази даних. Однак він все ще не повністю відповідає основним вимогам нормалізації. У реальній системі будуть створені додаткові таблиці відділів [9].

Для отримання даних з таблиць бази даних використовуйте спеціальну команду MySQL `Select query`. Для того, щоб сервер бази даних правильно відповів на запит, запит повинен бути правильно побудований. Структура запиту будується наступним чином. Всі звернення до сервера бази даних починаються з ключового слова `select`.

Це основа для всіх запитів в MySQL. Приклади можуть бути різної складності, але принципи дуже схожі. Далі потрібно вказати поля, з яких ви хочете вибрати інформацію, що вас цікавить. Список полів відокремлюється комами після речення `select`. Після того, як всі необхідні поля перераховані, в запиті використовується речення `from` і назва таблиці, щоб вказати об'єкт таблиці, з якого потрібно зробити вибірку.

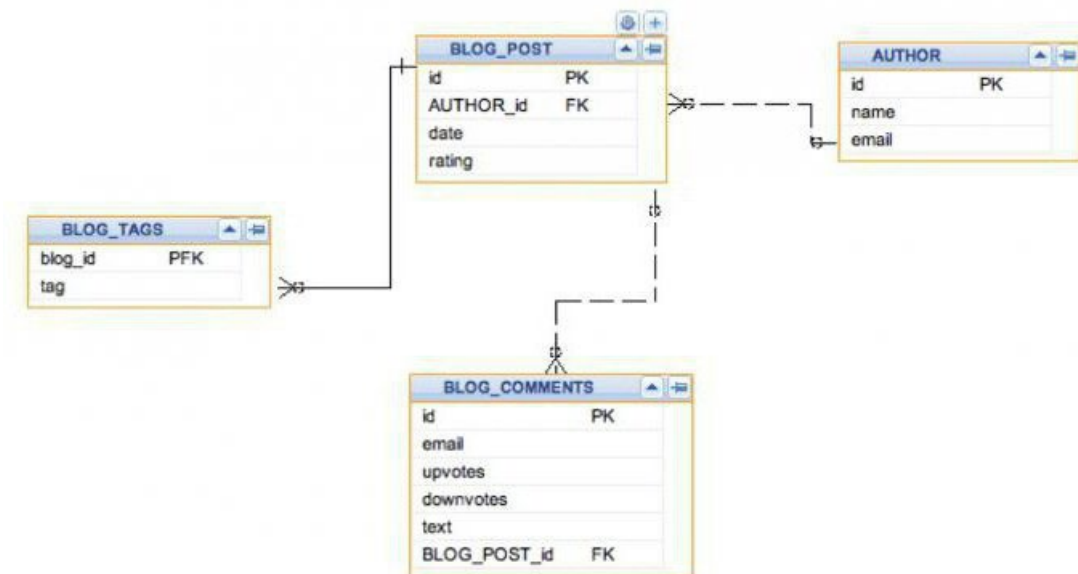


Рисунок 2.3 – Зразок структури в базі даних MySQL

Щоб обмежити вибірку, до запиту MySQL додаються спеціальні оператори, передбачені СУБД: речення `distinct` використовується для відбору неповторюваних (унікальних) даних, а речення `where` – для вказівки умов. Як приклад, застосовний до наведеної вище таблиці, можна розглянути запит, який вимагає інформацію про повне ім'я співробітника, що працює у відділі продажів.

Структура запиту показана на рис. 2.4.

```

Структура даних  Просмотр данных  SQL редактор
1 select surname,
2     name,
3     patronymic
4 from employees
5 where daptment_id = "Продажи"

```

Рисунок 2.4 – Структура MySQL запиту

Однак найважливішою особливістю СУБД є здатність обробляти вкладені запити MySQL, про що вже згадувалося вище. Як він повинен виглядати? Як впливає з назви, видно, що це запит, який складається з ієрархії двох або більше запитів; теорія вивчення властивостей СУБД стверджує, що MySQL не накладає ніяких обмежень на кількість запитів MySQL, які можуть бути вкладені в основний запит. Але реальні експерименти можуть підтвердити, що час відповіді значно збільшується після другого десятка вкладених запитів. У будь-якому випадку, на практиці не існує завдань, які вимагають використання надскладних запитів до MySQL. Кількість вкладених ієрархій, які можуть знадобитися для виконання запиту, може становити 3-5, не більше.

При аналізі прочитаної інформації виникає ряд питань, наприклад, де можна використовувати вкладені запити для вирішення завдань або чи можна їх розбити на прості структури без складності. На практиці вкладені запити використовуються для вирішення складних завдань.

Цей тип завдань також включає ситуації, коли умови, що обмежують подальшу вибірку значень, не відомі заздалегідь.

Вирішити ці проблеми за допомогою звичайних запитів до MySQL неможливо. Запити, структуровані в ієрархії, будуть шукати обмеження, які можуть змінюватися з часом або які не можуть бути відомі заздалегідь. Наступний приклад може бути використаний як складне завдання з урахуванням наведеної вище таблиці Припустимо, вам потрібно знайти основну інформацію про співробітника, який підпорядковується начальнику відділу продажів.

При подачі заяви ідентифікаційний номер невідомий. Тому спочатку необхідно його знайти. Для цього скористайтеся простим запитом, який може знайти розв'язок головної умови і виконати основний запит до MySQL.



## 2.5 Бібліотеки ReactJS

React – це декларативна, ефективна і гнучка JavaScript-бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє створювати ефективні, масштабовані та перевикористовувані інтерфейси з невеликих окремих частин коду – «компонентів». Але не кожен JavaScript розробник може вважатися кваліфікованим розробником React JS, він додатково має знати фреймворк React та низку інших веб технологій [10].

Багато компаній зараз шукають співробітників зі специфічними навичками та досвідом роботи з React JS; компанії, що використовують React JS, у всіх на слуху. Найпопулярніша з них – Meta (колишній Facebook), інші зі списку Fortune 500 – Dropbox, Khan Academy, Yahoo! Mail, New York Times, Netflix, Facebook, Instagram і WhatsApp, Codecademy, Instagram і Uber.

Основна мета React – мінімізувати помилки, які виникають при розробці користувацьких інтерфейсів. Це досягається завдяки використанню компонентів (самодостатніх логічних частин коду); React абстрагує більшу частину роботи зі створення, дозволяючи вам зосередитися на дизайні.

Ви можете розробляти нові функції в React, не переписуючи наявний код.

React також може рендеритись на сервері, використовуючи Node, і приводити в дію мобільні додатки, які використовують React Native.

Наявність декларативних інтерфейсів робить ваш код більш передбачуваним і його набагато легше налагоджувати. ReactJS є популярним вибором для розробки модерних веб-додатків і забезпечує швидку, ефективну та розширювану розробку інтерфейсів користувача.

Приклад вихідного коду на React наведено на рис. 2.5.

The screenshot shows a dark-themed editor window titled "ІНТЕРАКТИВНИЙ JSX-РЕДАКТОР" with a checked "JSX" toggle. The code defines a `HelloMessage` component and renders it with the name "Петро". The output window on the right, titled "РЕЗУЛЬТАТ", displays "Привіт, Петро".

```

class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Привіт, {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Петро" />,
  document.getElementById('hello-example'),
);

```

Рисунок 2.5 – Приклад вихідного коду на React

Основні риси та переваги ReactJS:

- компонентна структура. ReactJS дозволяє будувати інтерфейс з використанням компонентної структури, де кожен компонент відповідає певній частині інтерфейсу. Це сприяє легкості управління кодом, перевикористанню та підтримці;
- віртуальний DOM. ReactJS використовує віртуальний DOM (Document Object Model), що дозволяє ефективно маніпулювати і оновлювати інтерфейс без необхідності повного перерендерингу сторінки. Це забезпечує швидку та ефективну роботу з великими обсягами даних;
- одностороннє зв'язування даних. У ReactJS використовується одностороннє зв'язування даних, що дозволяє просто керувати станом даних компонентів та оновлювати інтерфейс відповідно до змін даних;
- розширюваність. ReactJS дозволяє розширювати функціональність за допомогою додаткових бібліотек та розширень. Наприклад, React Router дозволяє керувати маршрутизацією, а Redux допомагає управляти станом додатка;

- сумісність. ReactJS можна використовувати в поєднанні з іншими технологіями та бібліотеками. Він може працювати як на клієнтському боці (у веб-браузері), так і на серверному боці за допомогою Node.js.

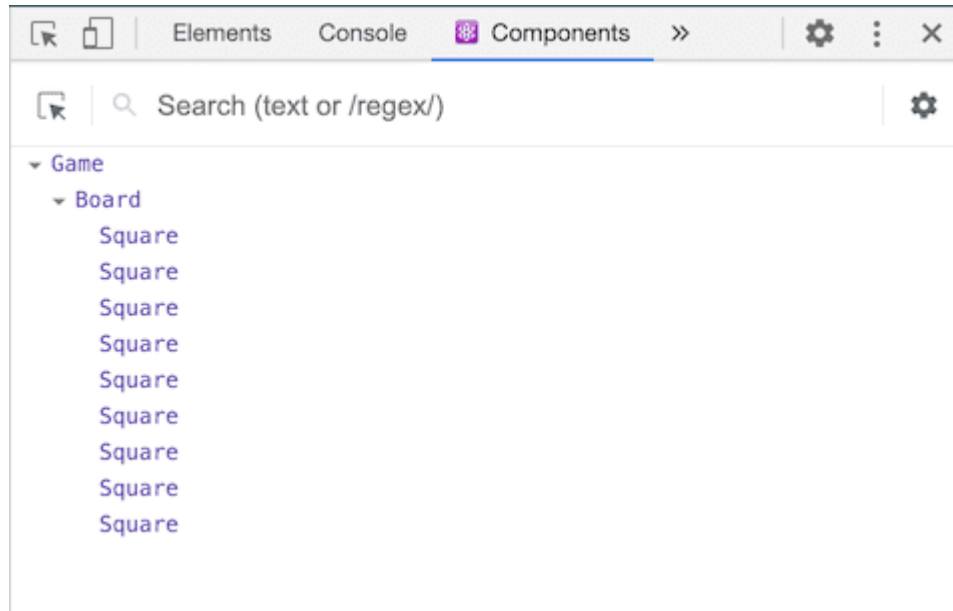


Рисунок 2.6 – Розширення React Devtools

Розширення React Devtools для Chrome та Firefox дозволяє інспектувати дерево React-компонентів у панелі інструментів розробника браузера перевірити props і стан React-компонента (рис. 2.6).

Одна з особливостей React – це використання компонентів. Компоненти є однією з основних концепцій React. Вони є основою, на якій розробник будує інтерфейс користувача (UI). Компоненти в React – це цеглини, з яких будується дім. React дозволяє об'єднати розмітку, CSS і JavaScript у власні «компоненти», багаторазово використовувати елементи інтерфейсу користувача для веб-додатку. У React існує кілька різних видів компонентів.

Загальний алгоритм роботи з компонентом React:

- експорт компонента (export default);
- визначення функції (ключове слово «function»);

- створення розмітки (цей синтаксис називається JSX і дозволяє вставляти розмітку в JavaScript).

Було обрано оптимальні програмні засоби з тими можливостями, які дозволять в поставлені терміни розробити веб-сайт, що відповідає сучасним вимогам клієнтів.

## **3 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ**

### **3.1 Формування технічного завдання до системи**

Технічне завдання (ТЗ) для розробки інтернет-магазину є важливим кроком, щоб уточнити всі деталі та вимоги проекту перед початком розробки. Замовник перед розробником поставив вимоги що до розробки веб-сайту с подальшим використанням в мережі Internet, такі вимоги як:

- зрозумілий функціонал для адміністраторів та клієнтів сайту;
- загальна ідея та мета інтернет-магазину;
- інформація про основні сторінки;
- інтеграції з іншими системами;
- вимоги до безпеки та продуктивності;
- інтеграція з платіжними шлюзами для обробки платежів;
- адаптивний дизайн для оптимального відображення на різних пристроях.

### **3.2 Проектування системи**

Наведемо схему інтернет-магазину (рис.3.1), що побудована за допомогою програми Erwin Process Modeler, для зручності і розуміння поставленої задачі, описані етапи розробки веб-сайту. А саме:

- постанова технічного завдання з усіма вимогами та побажаннями від замовника до розробника;
- правила розробки веб-сайтів, яких будуть дотримуватися для отримання бажаного та якісного результату;
- вибір програмного забезпечення для розробки та подальшого підтримання функціоналу;
- вибір інтернет-провайдеру, який надає доступ сайту до мережі Internet;

- послуга (іншим словом хостинг) яка надає можливість розміщувати файли або сайти на сервері, який цілодобово доступний. Це потрібно для того, щоб люди могли без проблем відвідувати сайт 24/7;
- дотримання законів країни, про захист прав споживачів. Наприклад право на отримання достовірної та якісної інформації, кількість товару, асортимент, інформація про виробника тощо. Також це стосується контенту який розміщується на сайті (особливо платний контент) знижки, акції, рекламні банери, тексти, продукція та інші товари, відео, зображення фізичних обличч, слогани, персональні данні фізичних обличч тощо. Перевірка контенту на сайті на наявність недобросовісної конкуренції, використання торгових марок. За порушення рекламного законодавства України на власника сайту може бути накладено штраф у розмірі п'ятикратної вартості розповсюдженої інформації на сайті, або рекламного контенту.

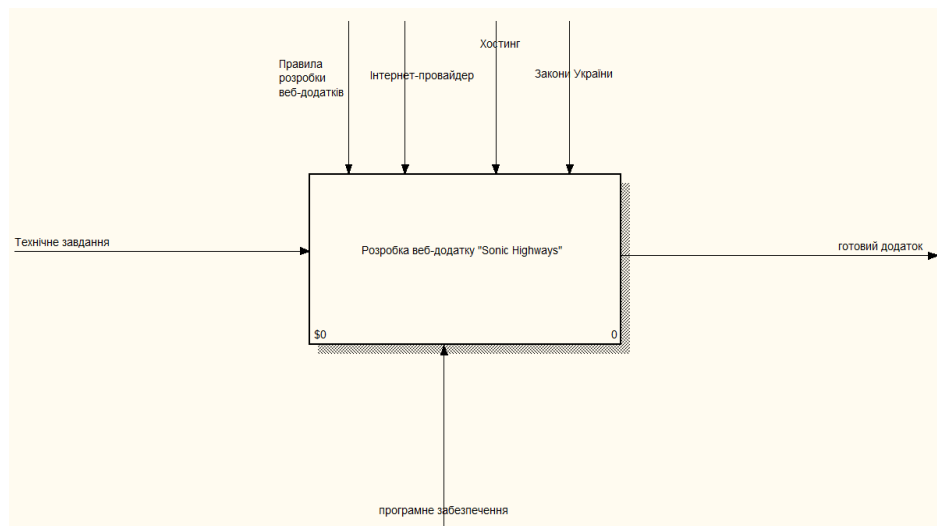


Рисунок 3.1 – Система розробки інтернет-магазину

Спроекуємо більш детально підсистеми, що є важливими для функціонування інтернет-магазину.

На сайті планується купівля товару, тому важливо організувати процеси, які дозволяють ефективно обробляти та виконувати замовлення, отримані від клієнтів (рис.3.2).

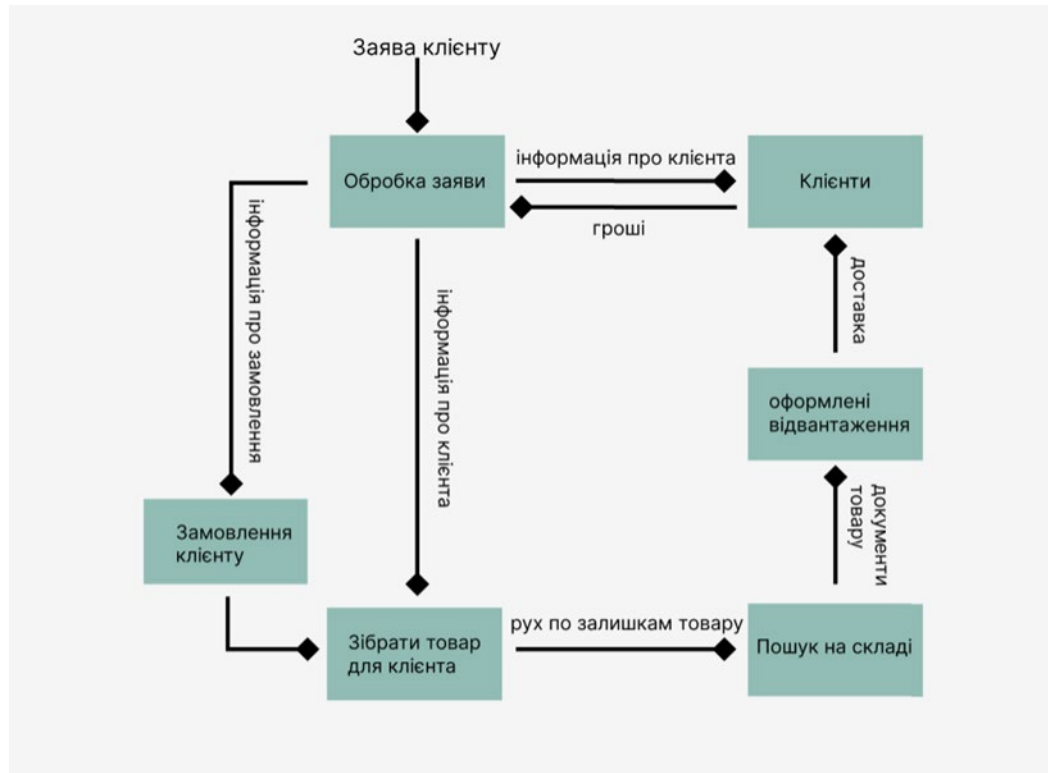


Рисунок 3.2 – Система обробки заказу клієнта

Ця система відіграє важливу роль у забезпеченні безперебійного процесу продажу та доставки товарів в інтернет-магазині. Основні компоненти системи обробки замовлень включають наступне:

- клієнт робить заявку на товар;
- надає всю необхідну інформацію про себе;
- оплата товару;
- інформація про клієнта і замовлення товару надходить до системи, що підбирає товар по параметрам вимогам клієнту;
- по цим параметрам шукається товар на складі;

- товар, разом зі всіма потрібними паперами, відправляється на оформлення на відвантаження;
- клієнт доставкою отримує свій товар.

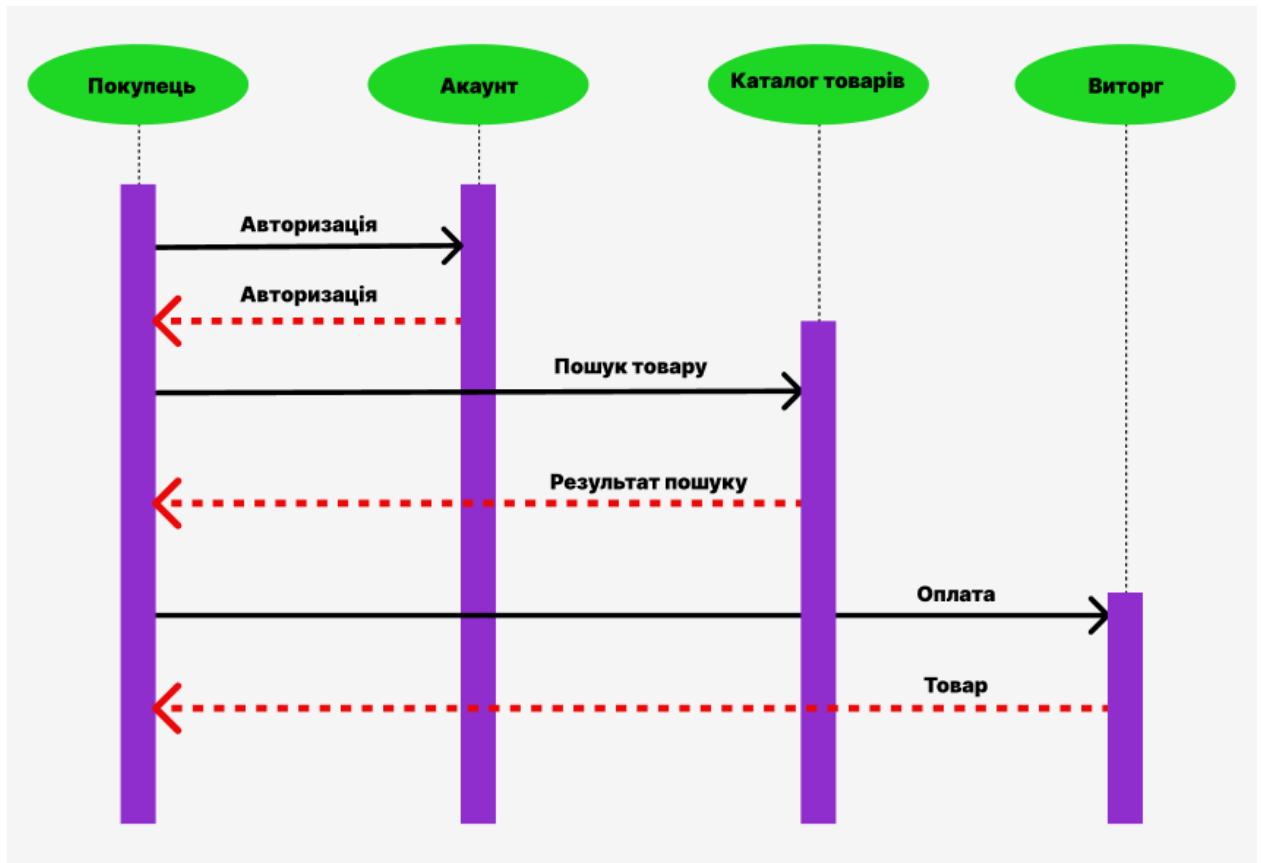


Рисунок 3.3 – Діаграма послідовності роботи інтернет-магазину

Ця система онлайн-покупок є надійним інструментом розробки програмного забезпечення, який дозволяє компаніям створювати індивідуальні покупки (рис. 3.3).

Він надає клієнтам зручний, безпечний і зручний спосіб здійснювати транзакції та купувати продукти. Він оснащений такими функціями, як керування кошиком для покупок, інтеграція платіжного шлюзу, відстеження замовлень.

Це також дозволяє розробнику адаптувати програмне забезпечення до своїх потреб, дозволяючи їм налаштовувати інтерфейс користувача та дода-



вати власні функції. Система онлайн-покупок полегшує підприємствам швидке налаштування повноцінної системи електронної комерції. Спочатку покупець проходить авторизацію на сайті, після авторизації покупець переходить до каталогу товарів, йому представлені товари. Клієнт платить за товар, товар надходить клієнту. Магазин отримує виторг.

### 3.3 Макет веб-сторінки

Наступним кроком було створення макетів сторінок магазину.

На рисунку показано макет головної веб-сторінки інтернет-магазину (рис.3.4).

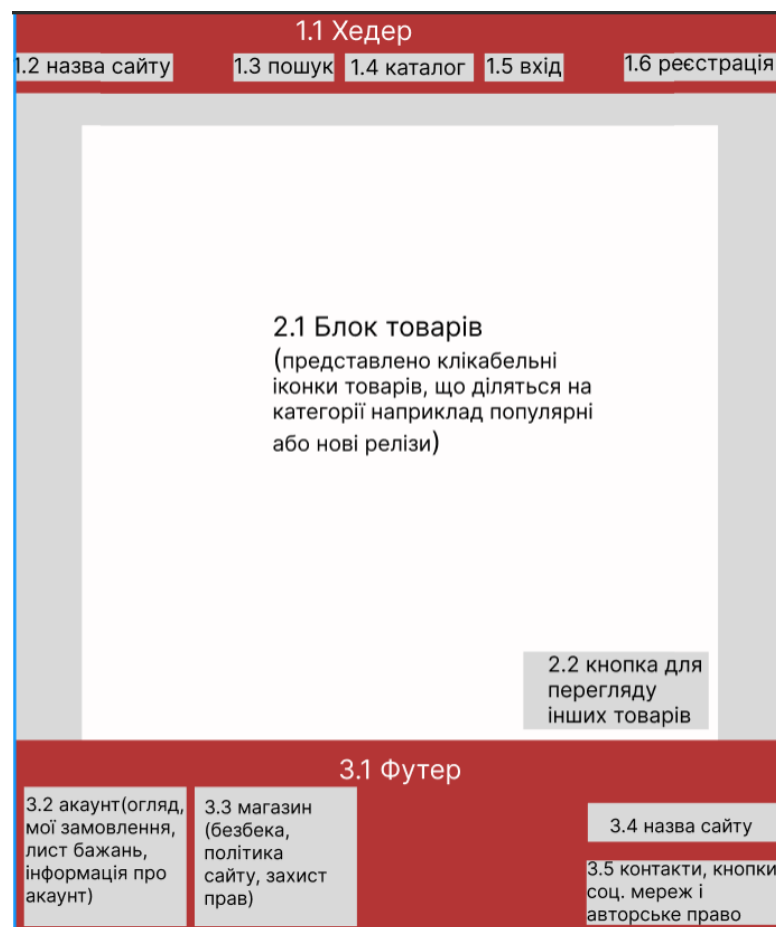


Рисунок 3.4 – Макет головної веб-сторінки

Як можна побачити – в хедері представлені назва сайту, search box каталог, кнопка реєстрації та авторизації В центральній частині сайту по середині є популярні/нові релізи після натиснення яких відвідувач сайту потрапляє в розділ с потрібним товаром.

В футері також розміщено назву сайту, поштову скриньку і телефон, по якому можна зв'язатись с адміністрацією, кнопки посилання на соціальні мережі. Присутні також різноманітні технічні розділи, що пояснюють правила сайту – «Про безпеку», «Політика користувача», «Про захист прав користувачів». Є розділи що пов'язані с обліковими записами користувача – в розділі «Огляд» можна побачити останню активність облікового запису. В «Мої замовлення» показується список вже придбаних товарів, а в «Лист бажань» ті товари, що планує купити користувач сайту. Вкладка «Інформація про акант» містить в собі інформацію про дату створення облікового запису, нікнейм або ім'я, свої данні тощо.

Кожен інтернет магазин повинен мати вікно покупки товарів, для покупців. На рисунку знизу представлено можливий варіант такого вікна на ньому безпосередньо має бути кнопка «Придбати» що дозволяє купити обрану продукцію, також кнопка «Додати до кошику», це треба для того, щоб зберегти товар що зацікавив покупця, але він не впевнений що потрібно його придбати зараз. Не менш головне це правильно вказана актуальна ціна. В цьому вікні, опціонально може бути деяка інша інформація про товар, яка вплине на рішення(рейтинг, кількість придбання тощо). Перше, що перед покупкою треба зауважити – наявність товару, вона постійно повинна бути точною, перед тим, як робити замовлення. Код товару – це швидкий спосіб знайти товар на магазині клієнту, та на складі – продавцю.

По макету було побудовано головну сторінку інтернет-магазину, де було відтворено всі ідеї та вимоги, які були поставлені системі. Без втрати функціоналу (рис.3.5).

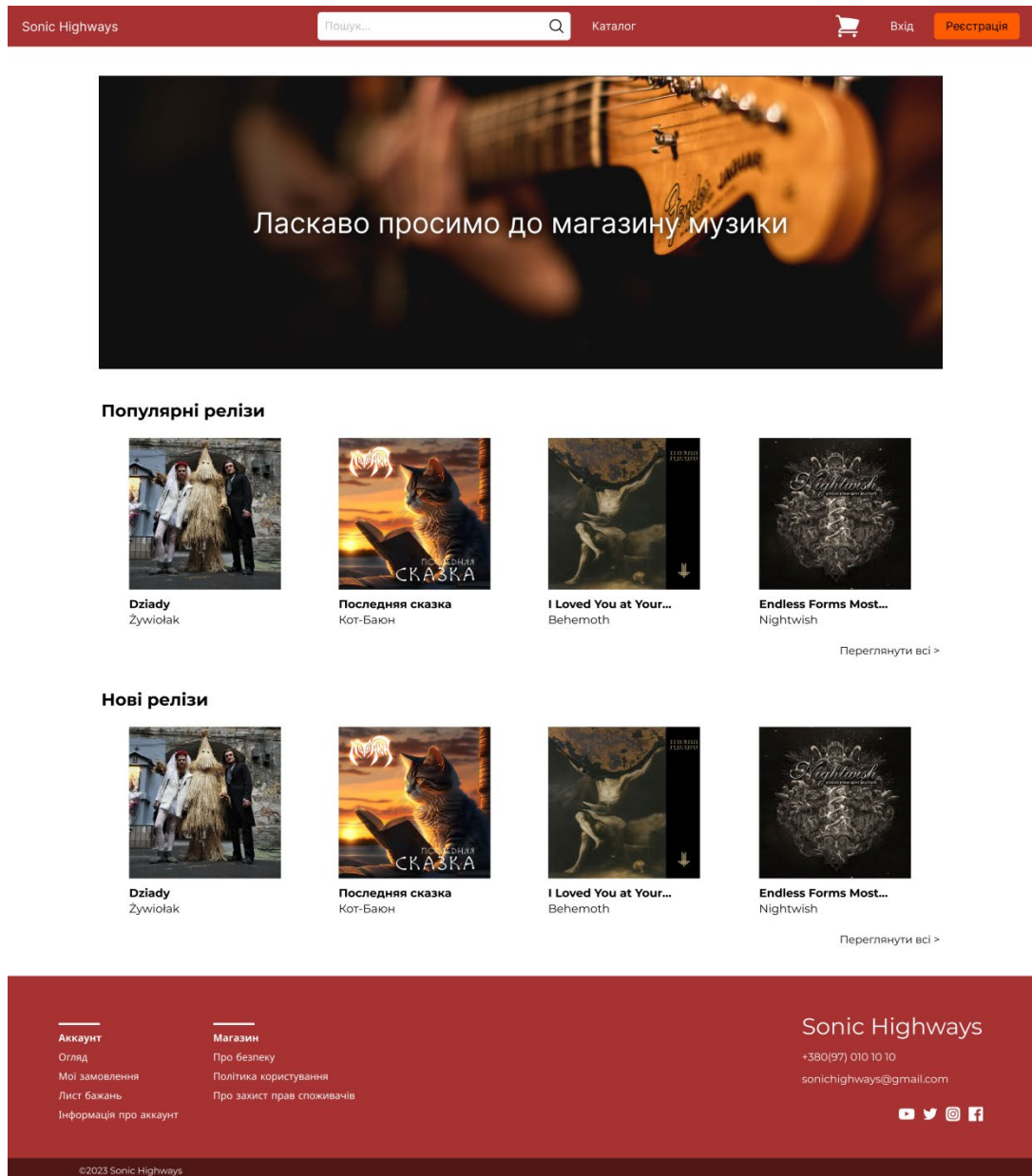


Рисунок 3.5 – Головна веб-сторінка


На сторінці товару, як правило, повинна бути інформація про альбом: його назва, назва виконавця, рік релізу, країна, обкладинка, жанр, стиль, номер товару, список пісень, в якому форматі є в наявності на сайті (CD диски, вініл тощо). Макет сторінки представлений на рис.3.6.



Рисунок 3.6 – Макет сторінки товару

Корисним буде наявність секції коментарів, де користувачі можуть залишити свої відгуки на товар, поділитись враженнями, порадити або навпаки не радити тратити гроші (рис. 3.7).

Sonic Highways  [Каталог](#) [Вхід](#) [Реєстрація](#)



**Behemoth - I Loved You at Your Darkest (Tour Edition)**  
 Лейбл: Nuclear Blast GmbH  
 Номер каталогу: NB 5319-0  
 Формат: CD  
 Країна: Німеччина  
 Реліз: 2020  
 Жанр: Metal  
 Стиль: Death Metal

**Треклист**

№	Назва	Тривалість
01	Solve	2:05 хв.
02	Wolves ov Siberia	2:54 хв.
03	God = Dog	3:59 хв.
04	Ecclesia Diabolica Catholica	4:50 хв.
05	Bartzabel	5:01 хв.
06	If Crucifixion Was Not Enough	3:17 хв.
07	Angelvs XIII	3:41 хв.
08	Sabbath Mater	4:57 хв.
09	Havohej Pantocrator	6:05 хв.
10	Rom 5:8	4:23 хв.
11	We Are the Next 1000 Years	3:24 хв.
12	Coagula	2:05 хв.
<b>Бонус-треки</b>		
13	Wolves ov Siberia	2:57 хв.
14	God = Dog	3:59 хв.
15	Ecclesia Diabolica Catholica	4:54 хв.
16	Bartzabel	5:02 хв.
<b>Загальна тривалість альбому</b>		<b>63:31 хв.</b>

**Відгуки**

[Додатик відгук](#)

**thesatanist69** 23 квітня, 2021  
 Мощний альбом, мне нравится!

[Відповісти](#)

**Товар є в наявності**

[Додати до кошика](#)

[Придбати](#)

Ціна: **€850.00**


Рейтинг: 5.0

Придбали: 4382

Код товару: [1236874]

★★★★★ [←](#)

**Відео**



Behemoth - Bartzabel (2020)  
 Provided by YouTube

**Акаунт**

- [Огляд](#)
- [Мои замовлення](#)
- [Лист бажань](#)
- [Інформація про акаунт](#)

**Магазин**

- [Про нас](#)
- [Про безпеку](#)
- [Політика користування](#)
- [Про захист прав споживачів](#)

**Sonic Highways**

+380(97) 010 10 10  
 sonichighways@gmail.com

[YouTube](#) [Twitter](#) [Instagram](#) [Facebook](#)

©2023 Sonic Highways

Рисунок 3.7 – Сторінка товару



Нижче зображено процес пошуку товарів на сайті. Зліва в колонці знаходяться фільтри, що допомагають швидше знайти потрібне – жанр, стиль, рік, формат. Результат пошуку виводиться на екран, показана його ціна та можливість додати до кошику (рис. 3.8).

Sonic Highways

Пошук...  Каталог

**Активні фільтри:**

**Результати пошуку**  
1-1 з 1 < Поп. Наст. > **Сортувати**  **Показати**

Назва	Ціна
 <p><b>I Loved You at Your Darkest (Tour Edition) (2020)</b> Behemoth Тип: Альбом Лейбл: Unkknown Оцінка: 5.0</p>	<p>€850.00 <input type="button" value="До кошику"/></p>
 <p><b>Opvs Contra Natvram (2022)</b> Behemoth Тип: Альбом Лейбл: Unkknown Оцінка: 5.0</p>	<p>€850.00 <input type="button" value="До кошику"/></p>

**Жанр**  
Rock  
Metal  
Folk  
Acoustic  
Soundtrack  
[Більше...](#)

**Стиль**  
Daedric Black Metal  
Atmospheric Black Metal  
Folk Metal  
Folk Rock  
Punk Rock  
[Більше...](#)

**Рік**  
Задати діапазон  
2023  
2022  
2021  
2022  
[Більше...](#)

**Формат**  
WEB  
CD  
Vinyl  
Касета  
DVD  
[Більше...](#)

---

**Акаунт**  
Огляд  
Мій замовлення  
Лист бажань  
Інформація про акаунт

**Магазин**  
Про безпеку  
Політика користування  
Про захист прав споживачів

Sonic Highways  
+380(97) 010 10 10  
sonichighways@gmail.com

©2023 Sonic Highways

Рисунок 3.8 – Пошук товару

В хедері сайту вказано кількість сторінок з потрібним товаром (їх може бути більше або менше), можливість сортування товару за алфавітним порядком, роками, популярність, спаданню тощо, та кількість потрібного товару що знайшовся по результатам пошуку.

### 3.4 Проектування системи баз даних

База даних (БД) – це організаційна структура, призначена для зберігання, маніпулювання та обробки взаємопов'язаної інформації, часто у великих обсягах. Це певний набір даних, пов'язаних між собою спільними ознаками або характеристиками та організованих, наприклад, в алфавітному порядку.

Поєднуючи великі обсяги даних в одній базі, можна групувати всі види інформації, такі як особисті дані клієнтів, історія замовлень, каталоги товарів тощо, і створювати безліч різних варіацій. Найбільшою перевагою баз даних є швидкість, з якою можна вводити та використовувати необхідну інформацію. Завдяки спеціальним алгоритмам, що використовуються в базі даних, ви можете легко знайти потрібні вам дані всього за кілька секунд.

Працювати з інформацією в базах даних також простіше і швидше, оскільки між інформацією в базі даних існують певні взаємозв'язки, і зміна одного рядка також змінить інші рядки. База даних для веб-сайту може зберігати інформацію у вигляді пов'язаних таблиць. У базі даних зберігається вся інформація (дані про клієнтів, прайс-листи, списки товарів), яка необхідна і корисна для роботи веб-сайту. Для створення запитів до бази даних часто використовуються мови структурованих запитів; SQL дозволяє додавати, редагувати і видаляти інформацію, що міститься в таблицях.

Основними СУБД є:

- об'єктно-реляційна система управління базами даних Oracle Database;
- вільна система управління базами даних PostgreSQL;
- система керування базами даних Microsoft SQL Сервер;
- вільна система управління базами даних MySQL.

Такі системи управління відрізняються централізованою обробкою запитів, забезпечують надійність, доступність та безпеку БД. Найбільш популярною системою управління є MySQL, вона дає зручний доступ для управління БД та підтримує велику кількість таблиць різних типів.

Розподілені бази даних – це сукупність логічно пов'язаних між собою БД, які є розподіленими у комп'ютерній мережі.

Переваги РБД:

- ця модель відображає інформацію в найбільш простій для користувача формі;
- заснована на розвиненому математичному апараті, який дозволяє досить лаконічно описати основні операції з даними;
- дозволяє створювати мови маніпулювання даними не процедурного типу;
- маніпулювання даними на рівні вихідний БД і можливість зміни.

Недоліки РБД:

- найповільніший доступ до даних;
- трудомісткість розробки.

СУБД – це програмний інструмент, який діє як міст між базою даних та її користувачами; він полегшує створення, підтримку та спільне використання баз даних різними користувачами за допомогою набору мов та програмних засобів.

Сучасна програма СУБД складається з ядра, процесора мови баз даних, підсистеми підтримки виконання та сервісної програми, яка надає додаткові функції для обслуговування інформаційної системи.

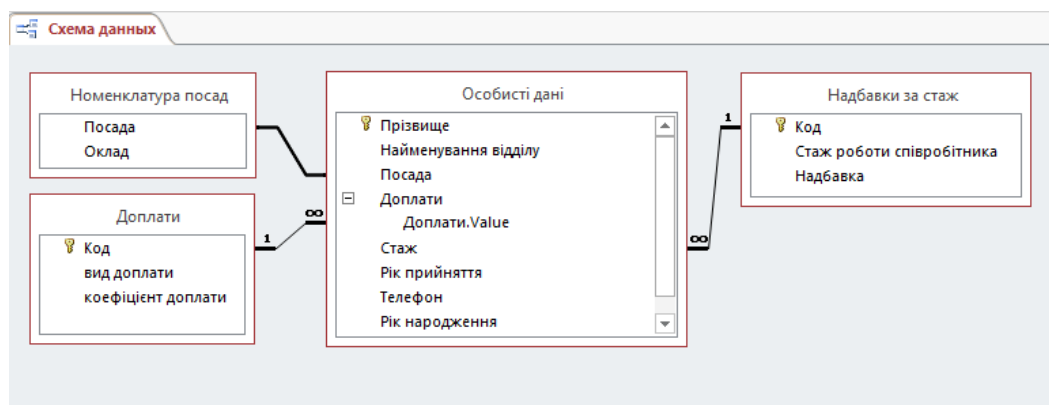


Рисунок 3.9 – Приклад бази даних зі сформованими зв'язками між таблицями



Проектування системи баз даних (СБД) – це процес створення оптимальної структури та організації бази даних для ефективного зберігання, оновлення та доступу до даних. Цей процес включає в себе кроки, такі як аналіз вимог, моделювання даних, нормалізацію, вибір СУБД та фізичне проектування. Нижче наведені основні кроки проектування СБД.

**Аналіз вимог.** Спочатку потрібно зрозуміти вимоги користувачів і бізнесу до системи баз даних. Це включає визначення типів даних, їх об'єму, виділення основних сутностей та взаємозв'язків між ними (рис. 3.9).

**Моделювання даних.** Після аналізу вимог створюється модель даних, яка відображає структуру даних та зв'язки між ними. Це може бути схема бази даних у вигляді сутність-відношення (Entity-Relationship, ER-модель) або інша формальна модель, така як UML.

**Нормалізація.** Після створення моделі даних проводиться нормалізація. Цей процес допомагає усунути аномалії та дублікати даних шляхом розбиття таблиць на менші, де кожна таблиця містить тільки необхідні дані та відповідає певним нормальним формам (наприклад, перша, друга, третя нормальні форми).

**Вибір СУБД.** На цьому етапі обирається підходяща система управління базами даних (СУБД), яка задовольняє потреби проекту. Рішення може бути засноване на функціональних можливостях, швидкості, масштабованості, безпеці та інших факторах.

**Фізичне проектування.** Після вибору СУБД проводиться фізичне проектування, що означає визначення конкретних таблиць, полів, індексів та інших об'єктів бази даних. Важливо враховувати фізичні обмеження апаратного забезпечення та оптимізувати структуру бази даних для швидкого доступу до інформації.

**Розробка додаткової функціональності.** На цьому етапі можна розробити додаткову функціональність, таку як зберігання процедур, тригери, перегляди та інші об'єкти, які допоможуть в ефективному використанні бази даних.

Реалізація та тестування. Останній етап – реалізація проекту в кодї та тестування. Створюються таблиці, індекси, запити та інші об'єкти бази даних. Дані імпортуються і тестуються на відповідність вимогам та коректну роботу.

Важливо враховувати принципи проектування СБД, такі як надійність, ефективність, безпека та масштабованість. Також рекомендується документувати всі етапи проектування для подальшого супроводу та розширення СБД.

Для підвищення зручності використання СУБД використовують веб-додатки, які дозволяють керувати сервером баз даних через графічний інтерфейс, виконувати спеціальні команди та змінювати вміст таблиць і баз даних. Наприклад, phpMyAdmin використовується для управління базами даних MySQL, а pgAdmin – для управління PostgreSQL. Ви також можете знайти ці програми для керування базами даних на нашому віртуальному хостингу cPanel.

На рисунку зверху показано проектування системи баз даних для інтернет магазину Sonic Highways (рис.3.10).

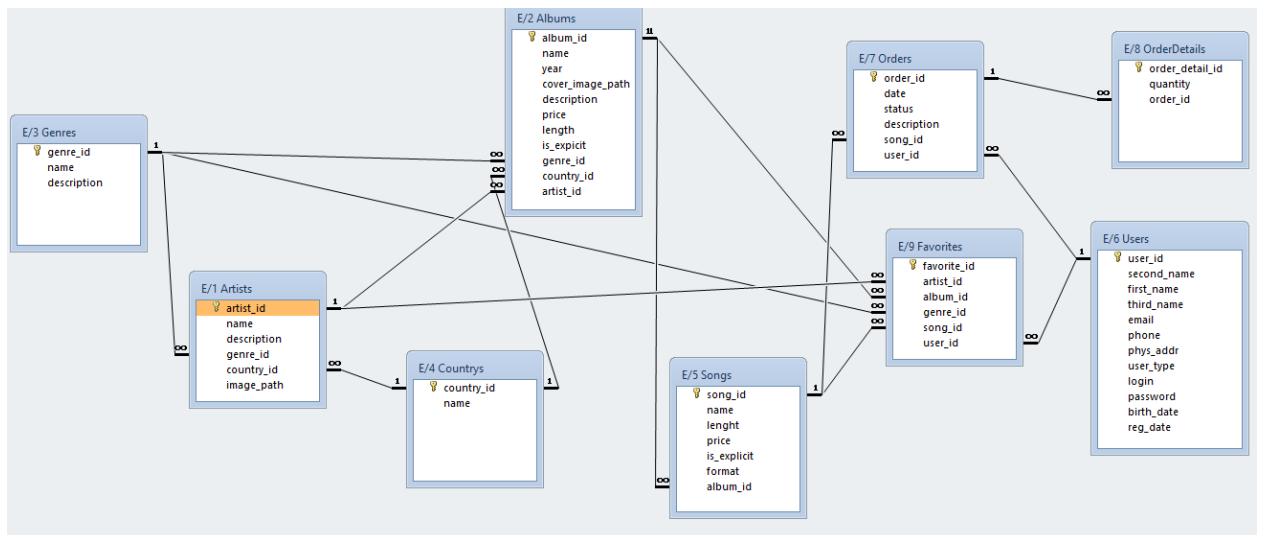


Рисунок 3.10 – Організація системи БД для інтернет-магазину

Вона має наступну структуру:

- таблиця с жанрами «Genres» має в собі назви та опис жанрів, вона пов'язана с таблицею альбомів «Albums» зв'язком «один-до-багатьох» – тобто в різних альбомів може бути один жанр. В таблиці «Albums» також є такі параметри як назва, рік, обкладинка, описання, ціна, довжина альбому і назви інших таблиць, що з'єднані з цією;
- таблиця «Albums» пов'язана с таблицею виконавців «Artists» зв'язком «один-до-багатьох», один артист може мати багато альбомів;
- також таблиця «Albums» пов'язана с таблицею країн «Countries» зв'язком «один-до-багатьох», що позначає наявність декількох альбомів з однієї країни;
- таблиця «Albums» пов'язана с таблицею пісень/синглів «Songs» зв'язком «один-до-багатьох» – багато пісень в одному альбомі;
- таблиця «Artists» має в собі ім'я, опис та зв'язок «один-до-багатьох» з таблицею улюблених виконавців/пісень «Favorites»;
- таблиця «Songs» має зв'язок «один-до-багатьох» с таблицею заказів «Orders».
- таблиця заказів «Orders», де вказано статус, дату замовлення, повинна мати зв'язок с таблицею деталей замовлення «Orders – Details».
- таблиця заказів «Orders» має зв'язок с таблицею користувачів «Users» «один-до-багатьох», тобто один користувач може зробити багато замовлень.
- таблицею користувачів «Users» зв'язком «один-до-багатьох» поєднана з таблицею улюблених виконавців/пісень «Favorites», таким чином у одного користувача може бути декілька улюблених виконавців.

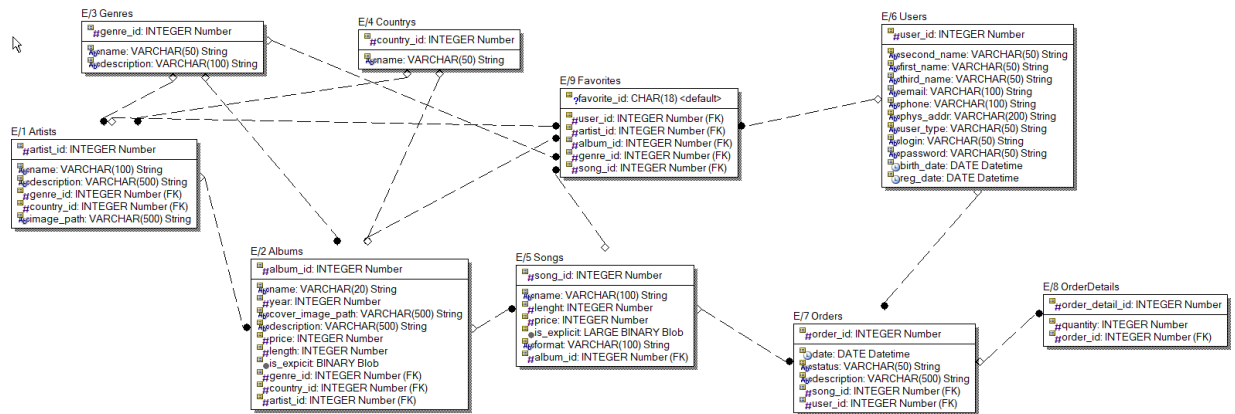


Рисунок 3.11 – Логічна модель

У ході проектування інформаційної системи було сформовано технічне завдання що показує систему розробки сайту. Зроблено макети веб-сторінок та побудовано логічну модель бази даних інтернет-магазину з допомогою середовища ERwin. Яка повністю відповідає вимогам функціоналу сайту (рис. 3.11).

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

### 4.1 Керівництво користувача

Інструкція користувача – довідково-інформаційний документ, в якому зібрані відомості щодо правильного, безпечного і ефективного використання описуваного продукту. Знання базових положень посібника з експлуатації – це можливість уникнути збоїв, поломок і передчасних виходів техніки з ладу. Користувачі часто вважають за краще вивчати всі тонкощі роботи з програмним забезпеченням на практиці, навіть не читаючи інструкцій, що додаються до нього. Як наслідок, посібники, які мають допомагати користувачам, не полегшують їм життя, а багато корисних функцій програмних продуктів залишаються незрозумілими і недоторканими [11].

Щоб цього не сталося, посібники користувача повинні відповідати своїй назві. Автор посібника повинен ретельно продумати, як люди будуть використовувати програмний продукт і які проблеми вони будуть шукати в тексті посібника. Текст посібника слід створювати з урахуванням цих ідей.

Основними факторами, що визначають якість посібника користувача, є, звичайно, його структура і сам текст. Загальна якість посібника залежить від того, наскільки добре ці компоненти відповідають потребам користувача. Структура посібника має бути зрозумілою, простою і побудованою навколо призначення та використання програмного продукту. Іншими словами, вона повинна відповідати на запитання:

- що це за продукт?;
- кому і коли він потрібен?;
- як з ним працювати: установка і настройка, перелік вирішуваних завдань і оптимальні способи їх реалізації, способи вирішення можливих проблем.

Структуруючи інструкцію, потрібно думати з точки зору потреб користувачів програмного продукту. Тоді, коли вони знаходять те, що їм потрібно,

вони отримують вичерпну інформацію про те, як вирішити поставлене завдання, коли і для чого використовувати цю функцію програми. Оскільки користувачі не читають весь посібник, а лише переглядають потрібні їм розділи, кожен розділ повинен бути самодостатнім і надавати всю необхідну інформацію.

Крім правильної розбивки тексту практичне керівництво повинне відповідати таким умовам:

- обов'язкові правила і моменти, що вимагають особливої уваги користувача, слід виділяти в тексті;
- текст бажано забезпечити перехресними посиланнями так, щоб при необхідності користувач не мучився в пошуках згаданого терміну;
- верстка керівництва повинна бути зручною для користувача;
- при необхідності текст слід супроводжувати відповідними текстовими ілюстраціями;

Ігнорування цих деталей може зіпсувати інструкцію та ускладнити розуміння тексту користувачами. Хороший посібник користувача характеризується не лише продуманою структурою, але й якістю тексту. Тут немає фіксованих критеріїв і єдиного стилю. Основні принципи полягають у тому, що текст інструкції повинен надавати користувачеві чіткі та вичерпні відповіді, не перевантажувати читача непотрібними та надлишковими технічними деталями, а також бути легким для читання та грамотним. Посібники користувача створюються для користувача, а не для опису програмного продукту, і повинні відповідати різним вимогам:

- головна персона в керівництві завжди користувач і його запити;
- чіткість характеристик;
- стислість тексту.

Для інтернет-магазину Sonic Highways було розроблено своє керівництво користувача, для того щоб допомогти користувачеві зрозуміти як користу-

ватись магазином, робити покупки, ознайомитися з функціоналом. Це все допоможе не загубитись в орієнтації сайту.

Головна сторінка. На головній сторінці магазину розміщені такі функції як реєстрація/авторизація, каталог, поле пошуку, в яке можна ввести ключові слова або назву товару, який користувач шукає. Використовуйте фільтри або категорії, щоб обмежити результати пошуку за ціною, брендом, розміром тощо.

Обрання товару. Перегляньте результати пошуку та натисніть на товар, щоб переглянути детальну інформацію. Ознайомтесь з описом товару, характеристиками, фотографіями та відгуками попередніх покупців

Додавання товару до кошика. Якщо вам сподобався товар і ви бажаєте придбати його, натисніть кнопку "Додати до кошика" або "Купити зараз". Зазвичай ви зможете вибрати кількість одиниць товару, яку ви хочете придбати.

Оформлення замовлення. Перейдіть до свого кошика або відповідної сторінки для перегляду обраних товарів. Переконайтеся, що список товарів та їх кількість правильні. Введіть дані доставки, включаючи адресу, контактну інформацію та спосіб оплати. Перевірте інформацію ще раз перед підтвердженням замовлення.

Оплата. Виберіть зручний спосіб оплати, такий як кредитна картка, платіжна система чи готівка при отриманні (залежить від доступних варіантів). Введіть необхідну інформацію для оплати та підтвердіть транзакцію.

Підтвердження замовлення. Після успішної оплати ви отримаєте підтвердження замовлення на електронну пошту або на сторінці підтвердження. Зазвичай цей електронний лист містить інформацію про замовлення, включаючи номер замовлення, товари, адресу доставки та іншу важливу інформацію.

Слідкування за замовленням. За допомогою номера замовлення або облікового запису на сайті ви зможете слідкувати за статусом вашого замов-

лення. Ви отримаєте оновлення про обробку, доставку та прибуття замовлення.

Повернення товару. Якщо ви не задоволені отриманим товаром, зверніться до політики повернень магазину. Більшість магазинів надають певний строк для повернення товару та вимоги щодо його стану.

## 4.2 Керівництво адміністратора

Керівництво адміністратора – це складова частина експлуатаційної документації або набір документів, що розробляється на будь-яку програму чи автоматизовану систему. За допомогою керівництва адміністратора відповідальні користувачі системи отримують можливість керувати її функціонуванням – виконувати певні операції із забезпечення порядку роботи АСК, розподіляти права доступу до неї, редагувати дані та виправляти помилки. Це детальний опис ролі адміністратора та процедур, які він повинен виконувати для успішного управління певною системою, організацією або процесом [12].

Керівництво адміністраторів часто плутають із керівництвами системних програмістів (адміністраторів). Це не так, адже мета цього документа – організація та підтримка цільового застосування АСК, а не забезпечення її технічної працездатності.

Керівництво адміністратора може включати наступні елементи:

- опис ролі – чіткий опис обов'язків та відповідальностей адміністратора;
- процедури роботи – інструкції з виконання різних завдань, які входять в обов'язки адміністратора. Наприклад, управління користувачами, налаштування системи, резервне копіювання даних тощо;
- керівництво з безпеки – правила та процедури щодо забезпечення безпеки системи або мережі, включаючи автентифікацію, захист від несанкціонованого доступу, збереження даних тощо;



- конфігурація та налаштування – інструкції щодо налаштування різних параметрів системи, програмного забезпечення або обладнання;
- відновлення системи – процедури відновлення після виникнення неполадок або аварійних ситуацій, включаючи резервне копіювання, відновлення даних та відновлення роботи системи;
- документація – інформація про систему, включаючи схеми мережі, конфігураційні файли, списки користувачів та паролі, доступні ресурси тощо;
- підтримка та контакти – контактна інформація або процедури звернення для отримання допомоги або підтримки.

Керівництво адміністратора допомагає адміністраторам бути організованими, ефективно виконувати свої обов'язки та забезпечувати безперебійну роботу системи, мережі або процесу, за яким вони відповідають.

Для інтернет-магазину Sonic Highways було розроблено керівництво для адміністратора з наступними пунктами.

#### Управління товарами:

- додавання товарів – для додавання нових товарів в магазин ви можете використовувати адміністративну панель. Для кожного товару введіть назву, опис, характеристики, ціну, зображення та інші відповідні дані;
- редагування товарів – забезпечте можливість редагування існуючих товарів, включаючи зміну опису, ціни, наявності та інших деталей;
- керування запасами – слідкуйте за кількістю товарів на складі та оновлюйте інформацію про наявність.

#### Керування замовленнями:

- огляд замовлень – ви маєте доступ до списку замовлень, які були зроблені покупцями. Ви можете переглядати інформацію про кожне замовлення, включаючи деталі покупки, адресу доставки та статус замовлення;

- оновлення статусу – після обробки замовлення ви можете оновити його статус, наприклад, позначити як прийняте, оплачене, відправлене тощо. Це дозволяє покупцям відстежувати їх замовлення.

#### Управління користувачами:

- реєстрація та автентифікація – забезпечте можливість користувачам реєструватися в магазині та автентифікуватися для здійснення покупок;
- керування обліковими записами – можливість перегляду, редагування та видалення облікових записів користувачів;
- обробка запитів на підтримку – забезпечте можливість користувачам звертатися до служби підтримки через електронну пошту, чат або інші канали комунікації.

#### Оплата та доставка:

- інтеграція платіжних систем – налаштуйте платіжні шлюзи, щоб дозволити користувачам здійснювати оплату своїх замовлень;
- співпраця з службами доставки – укладіть угоди з службами доставки для зручної та ефективної доставки замовлень.

#### Маркетинг та просування:

- промоакції та знижки – забезпечте можливість запуску промоакцій, знижок або акційних пропозицій для залучення клієнтів.

Для інтернет-сайту було розроблено керівництва для користувачів та адміністраторів, для того, щоб користувачі та власники мали розуміння про безпечне користування сайту, згідно його правил, яке не буде порушувати законів.

## ВИСНОВКИ

У ході виконання дипломної роботи було спроектовано і розроблено сервісний додаток у вигляді музичного інтернет магазину “Sonic Highways”, де можна, перебуваючи в будь-якому місці, прослухати та зручно придбати найрізноманітніші вінілові платівки, касети, CD та DVD диски с музикою з широким спектром вибору. При визначені вимог до системи було проведено аналіз предметної області та вже існуючих аналогічних веб-магазинів.

У розробці сайту використано такі програмні засоби як мова розмітки HTML, Java Script, мова стилів CSS, що відповідає за зовнішній вигляд веб-сторінок сайту. Система MySQL, яка відповідає за функціонування баз даних А також ReactJS, для створення і розширення інтерфейсів.

Для сайту було сформовано керівництва користувача та адміністратора, в яких знаходяться певні довідки для власників і клієнтів з правами та обов’язками, які допоможуть уникнути збоїв у роботі сайту.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Використання освітніх веб-ресурсів. URL: <https://hostkoss.com/b/uk/what-is-website/> (дата звернення 03.05.2023).
2. Специфікація вимог. URL: <https://visuresolutions.com/uk/blog/requirements-specification/> (дата звернення 03.05.2023).
3. Основні поняття HTML лекція 3. URL: <https://sites.google.com/> (дата звернення 03.05.2023).
4. Ніксон Р. – Створюємо динамічні веб-сайти за допомогою PHP, MySQL, JavaScript, CSS і HTML, 2019 (дата звернення 03.05.2023).
5. Що таке CSS. URL: <https://css.in.ua/> (дата звернення 03.05.2023).
6. JavaScript – клієнтська мова програмування, що робить сторінки сайту інтерактивними. URL: <https://astwellsoft.com/uk/blog/tehnology/javascript.html> (дата звернення 04.05.2023).
7. Структура баз даних. URL: <https://apereps.kpi.ua/shco-take-basa-danykh> (дата звернення 08.05.2023).
8. Основні поняття та переваги MySQL URL: <https://uk.education-wiki.com/2686691-what-is-mysql-database> (дата звернення 04.05.2023).
9. MySQL – запит у запиті. MySQL: приклади запитів. URL: <https://hi-news.pp.ua/kompyuteri/700-mysql-zapit-u-zapit-mysql-prikladi-zapitv-vkladen-zapiti-mysql.html> (дата звернення 04.05.2023).
10. Посібник: з ReactJS URL: <https://uk.legacy.reactjs.org/tutorial/tutorial.html> (дата звернення 09.05.2023).
11. Керівництво користувача для користувача. URL: <https://wiki.cuspu.edu.ua/> (дата звернення 10.05.2023).
12. Теліженка О.М. та Глівенка С.В. – Адміністративний менеджмент, 2023