

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ МУЗИЧНОЇ ОНЛАЙН-КРАМНИЦІ
SOFTWARE IMPLEMENTATION OF AN ONLINE MUSIC STORE

Виконав: студент(ка) 4 курсу, групи КНТ-137
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Кунденко Д.С.

(прізвище та ініціали)

Керівник Голуб Т.В.

(прізвище та ініціали)

Рецензент Скрупський С.Ю.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Кунденка Дениса Сергійовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація музичної онлайн-крамниці.
Software Implementation of an Online Music Store

керівник проєкту (роботи) Голуб Тетяна Василівна, к.т.н.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) травень 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне
завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Розробка архітектури програми.
3. Основні рішення щодо реалізації компонентів системи. 4. Експлуатація,
тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Голуб Т.В., асистент		
Нормоконтролер	Липовець М.В., асистент		

7. Дата видачі завдання “1” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз та дослідження предметної області	2-5 тижні	Розділи 1, 2
3	Розробка архітектури програми	6 тиждень	Розділ 2
4	Розробка програми	7-9 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	10 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	11 тиждень	Додатки
7	Захист роботи	12 тиждень	

Студент(ка)

_____ Кунденко Д.С.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Голуб Т.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 81 с., 33 рис., 3 табл., 4 дод., 21 джерело.

ОНЛАЙН-КРАМНИЦЯ, ЕЛЕКТРОННІ ПРОДАЖІ, РОЗРОБКА, ПРОЄКТУВАННЯ, АРХІТЕКТУРА, ДИЗАЙН, ВЕБ-ЗАСТОСУНОК.

Об'єкт дослідження – процес розробки та функціонування онлайн-крамниць, що пропонують цифрові товари.

Предмет дослідження – методи розробки та підтримки функціонування онлайн-крамниць, що пропонують цифрові товари.

Метою роботи є програмна реалізація онлайн-крамниці, що пропонує цифрові товари на прикладі електронних продажів музичних композицій, альбомів виконавців, тощо.

Матеріали, методи та технічні засоби: мова програмування PHP, фреймворк Laravel, система керування базою даних MySQL.

Результати. Розроблено онлайн-крамницю, що пропонує користувачу придбати музичні композиції, альбоми або ж збірки в цифровому форматі. Онлайн-крамницю реалізовано у вигляді веб-застосунку, що реалізує електронний продаж цифрових послуг клієнтам та наповнення каталогу адміністратором.

Практична цінність роботи полягає в розробці онлайн-крамниці музичних композицій, альбомів та збірок в цифровому форматі у вигляді кроссплатформного веб-застосунку, що реалізує зручні інтерфейси для взаємодії клієнтів та адміністраторів, регулює питання продажу та використання цифрових товарів.

Висновки. Розроблено програмне забезпечення, що реалізує онлайн-крамницю музичних композицій.

Галузь використання – користувачі Інтернету, слухачі музики та аудіофіли.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 81 pages, 33 figures, 3 tables, 4 appendixes, 21 sources.

ONLINE STORE, ELECTRONIC SALES, DEVELOPMENT, DESIGN, ARCHITECTURE, DESIGN, WEB APPLICATION.

The object of research is the process of developing and operating online stores offering digital products.

The subject of the research is methods of developing and supporting the functioning of online stores offering digital goods.

The purpose of this work is to programmatically implement an online store that offers digital products based on the example of electronic sales of music compositions, albums of artists, and so on.

Materials, methods, and technical tools: PHP programming language, Laravel framework, MySQL database management system.

Results. An online store has been developed that offers the user to purchase music tracks, albums, or compilations in digital format. The online store is implemented in the form of a web application that implements electronic sales of digital services to customers and filling the catalog by the administrator.

The practical value of the work lies in the development of an online store of music compositions, albums and compilations in digital format in the form of a cross-platform web application that implements convenient interfaces for interaction between customers and administrators, regulates the sale and use of digital goods.

Conclusions. Software has been developed that implements an online store of music tracks.

Field of usage: Internet users, music listeners, and audiophiles.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз предметної області.....	10
1.1 Продаж цифрових товарів та даних.....	10
1.1.1 Види цифрових товарів	10
1.1.2 Особливості продажу цифрових товарів	13
1.1.3 Продаж музикальних композицій у якості цифрових товарів	14
1.2 Дослідження існуючих аналогів	14
1.2.1 Сервіс «Bandcamp»	15
1.2.2 Сервіс «Soundcloud»	16
1.2.3 Застосунок «iTunes Store»	17
1.2.4 Порівняння існуючих аналогів	19
1.3 Постановка завдання	20
1.3.1 Функціональність системи.....	21
1.3.2 Вимоги до інтерфейсу.....	21
1.3.3 Не функціональні вимоги до продукту.....	21
1.3.4 Оцінка надійності, продуктивності і підтримки.....	22
1.3 Висновки за розділом 1	22
2 Розробка архітектури програми.....	24
2.1 Вибір мови програмування.....	24
2.2 Вибір фреймворку для розробки.....	30
2.3 Загальна структура додатку.....	33
2.4 Висновки за розділом 2	34
3 Основні рішення щодо реалізації компонентів системи.....	35
3.1 Проектування системи на основі предметної області	35
3.2 Висновки за розділом 3	43
4 Експлуатація, тестування та експериментальне дослідження програми	44
4.1 Опис застосування.....	44

4.2 Умови виконання програми.....	44
4.3 Інструкція по експлуатації програми	45
4.3.1 Звернення до програми.....	45
4.3.2 Вхідні й вихідні дані	45
4.4 Методика та результати тестування	46
4.4.1 Тестування функціональності.....	46
4.4.2 Тестування сумісності	48
4.4.3 Тестування ергономічності	48
Висновки	50
Перелік джерел посилання	52
Додаток А Технічне завдання	55
Додаток Б Опис програми	61
Додаток В Текст програми	65
Додаток Д Слайди презентації.....	75

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

GUI – графічний інтерфейс користувача;

MVC – Model-view-controller;

LTV – позиттєва цінність клієнта;

SEO – пошукова оптимізація;

UML – Unified Modeling Language;

БД – база даних;

ОС – операційна система

ПЗ – програмне забезпечення;

ПК – персональний комп’ютер;

СКБД – система керування базами даних;

ТЗ – технічне завдання.

ВСТУП

Цифрові товари не можна пощупати руками, але сьогодні їх купують всі: від електронних книг до аудіо та програмного забезпечення – людству потрібно все.

Цифрові товари не треба ні зберігати, ні доставляти – ідеальний варіант для запуску свого інтернет-магазину або онлайн-крамниці, особливо в період карантину, коли через COVID-19 люди не готові зайвий раз виходити з дому за посилками, зате не проти почитати і повчитися чомусь новому.

Для підприємців продаж цифрових товарів має масу переваг. Товар завжди в наявності, не потрібно шукати місце для зберігання товарів, а крім того, в більшості випадків покупець отримує товар відразу після оплати.

У цифрових товарів є дві переваги: для них не потрібен склад і їх не потрібно доставляти поштою або кур'єром. Покупець просто оплачує замовлення і отримує посилання для скачування на електронну пошту. І головне – товар творений один раз, а продавати його можна скільки завгодно разів людям з усього світу.

Проте продаж цифрових товарів має і певний ряд особливостей і нюансів, які необхідно брати до уваги.

Саме тому є актуальною розробка системи «Музична онлайн-крамниця «Music Hall», яка б реалізувала можливості прийому та обробки запитів користувачів, перегляду каталогу товарів для користувача, користування зручним графічним інтерфейсом користувача (GUI) з модулем оплати та регулювала би питання продажу і використання цифрових даних.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Продаж цифрових товарів та даних

Цифровий товар або продукт – нематеріальний актив або контент, який продається або поширюється в Інтернеті [1]. У подібних продуктів є ряд переваг, особливо для підприємця, який займається їх реалізацією [1]-[2]:

- цифровий продукт легше поширювати, адже не потрібні склади і постачальники, щоб зберігати і доставляти товар. Процес відбувається фактично в пару натискань миші;
- клієнт отримує товар практично відразу після замовлення;
- ніяких додаткових витрат на матеріали, упаковку, збірку товарів і замовлень;
- досить висока маржинальність;
- цифровий продукт завжди буде в наявності, тому власник може продавати необмежену кількість товарів (копій товару);
- продаж цифрових товарів часто здійснюється в ніші онлайн-освіти, яка в свою чергу має високий потенціал зростання. За прогнозами до 2025 року в ніші буде сконцентровано \$319 мільярдів [1]-[2].

Резюмуючи, головна перевага цифрових продуктів в тому, що вони створюються один раз і можуть бути продані сотні разів без необхідності поповнювати запаси товару [1]-[2].

1.1.1 Види цифрових товарів

Онлайн-курс. Вже тривалий час не втрачає популярність культура «Зроби власноруч» (DIY). Якщо фахівець в якійсь сфері хоче поділитися своїми знаннями з аудиторією, то онлайн-курс – це чудовий варіант. Він створює його один раз, а потім продає необмежену кількість разів. Він може використовувати свої знання і таланти, допомагаючи іншим і проводячи індивідуальний коучинг для людей, які цікавляться певною темою [3].

Короткий посібник, гайд або чек-лист. Це хороший варіант для тих, хто має досвід і знання в будь-якій сфері і хоче поділитися ними з іншими, заробивши на цьому. Мета – дозволити клієнту отримати цінний досвід і лайфхаки швидше без вивчення тривалих відеокурсів і читання книг [3].

Графічні шаблони або мокапы. Подібні матеріали користуються великим попитом у тих, хто працює або навчається онлайн. Клієнти можуть використовувати ці шаблони як для створення веб-сайту, так і для дизайну одягу – вибір обмежений лише фантазією [3].

Шаблон. Можете запропонувати різні шаблони листів для email-розсилок, таблиць в Google Sheets, текстових документів, які користуються попитом і значно спрощують багато процесів.. Наприклад «10 прикладів успішних рекламних email-кампаній, які довели свою ефективність». Це також може бути шаблон статті в блозі, де одержувачу необхідно заповнити тільки прогалини, щоб скоротити процес публікації статті в блозі [3]-[4].

Додатки, плагіни та програмне забезпечення. Якщо замовник хоче продати будь-яку програму, яка передбачає завантаження інсталяційного файлу, це можна легко зробити це за допомогою платформ електронної комерції [3]-[4].

Членський доступ до ексклюзивного контенту або сайту. Це веб-сайт, який надає доступ лише підписаним користувачам. Контент сайту повністю захищений. Як тільки користувач входить в систему, він отримує доступ до ексклюзивного контенту. Замовник може запропонувати як постійне членство, так і оновлювану щомісяця або щороку підписку [3]-[4].

Доступ до платного співтовариства (Facebook і т.п.) Для постійних клієнтів і людей, які хочуть отримати більше інсайдерської інформації по певній темі [3]-[4].

Printables або роздруківки. Можна знайти безліч роздруківок на ресурсах, типу Pinterest або Etsy. На сайті знайдеться безліч шаблонів, наклейок, запрошень, розкладів, планувальників, ілюстрацій, матеріалів для скрапбукінгу і т. д. [3]-[4].

Стокові фото і відео. Це відмінна модель для фотографів та відеооператорів, які хочуть продавати свої продукти [3]-[4].

Фото і відео – фільтри або пресети. Багато користувачів соціальних мереж зацікавлені в красивій обробці фотографій і відео в один клік. Вирішити проблему багатьох людей можна за допомогою продажу пресетів як на телефон, так і на комп'ютер [3]-[4]. Одним клацанням миші фотографія буде змінена сотнями встановлених варіантів кольору, відтінків, тіні, контрасту і багато чого іншого. Це відмінний варіант товару, так як люди бажають мати якісні фото і відео, щоб ділитися ними в соціальних мережах [3]-[4].

Теми для електронних платформ. Створення шаблону сайту або інтернет-магазину – досить трудомісткий процес, який вимагає професійних навичок, але якщо ви з цим впоралися такі шаблони можна організовано продавати через Інтернет [3]-[4].

Шрифт. Тут все просто. Стандартні шрифти жахливо приїлися, а за їх допомогою складно зробити що-небудь оригінальне. Тому високий попит на нестандартні і виразні шрифти, які, як правило, платні [3]-[4].

Музика і аудіо – можна пропонувати біти і звуки різних інструментів, плагіни для музичного програмного забезпечення, стокову музику, а також звукові ефекти [3]-[4].

Ліцензії на цифрові активи. Фотограф, дизайнер або творець музики – може продавати ліцензію на свої цифрові активи, щоб клієнти могли використовувати їх у своєму власному контенті. Важливо не забути нанести водяний знак на всі продукти, щоб захистити їх від копіювання [5].

Електронні книги. Їх популярність зростає місяць за місяцем. У 2018 році в США було продано 538 мільйонів електронних книг, і ніша тільки продовжує набирати обертів. Користувач може вибрати будь-який жанр книг – список тем практично нескінченний.

1.1.2 Особливості продажу цифрових товарів

На початку, необхідно чітко визначити «на якому полі йде гра». У випадку з фізичними товарами або послугами все буде виглядати і працювати зовсім по іншому. Досліджуючи ж ринок цифрових нематеріальних товарів слід чітко визначити які у нього є особливості, які важливі для даної ситуації [6].

Кожна додаткова продана копія вже створеного продукту не коштує нічого. Створення продукту коштує грошей. Розробка програмного забезпечення або застосунку (ПЗ), створення відео-курсу, написання книги – все це коштує грошей, часу, інших ресурсів і т.д., але тиражування, кожна продана додаткова копія найчастіше не коштує нічого. Безумовно, в разі складного продукту можуть бути додаткові витрати, наприклад на підтримку, тощо. Але найчастіше, після того, як цифровий продукт був створений – вже немає різниці: продати одну копію або десять тисяч [3]-[6].

Залучення клієнта коштує грошей. Це чи особливість саме цифрових продуктів, але так – залучення клієнтів коштує грошей. Пошукова оптимізація (SEO), реклама, email маркетинг, і т.д. Особистий бренд, який є основним каналом залучення клієнтів в інфобізнесах, теж коштує грошей. Кожен прийшов клієнт обходиться компанії в якісь гроші. І якщо в компанії добре налаштована аналітика, то навіть відомо в які [3]-[6].

Пожиттєва цінність клієнта (LTV) більше ціни одного продукту. Зазвичай, компанії, які продають цифрові товари, мають більше одного товару в своїй лінійці продуктів [3]-[6]. Це може бути кілька програм, кілька курсів, кілька книг. І часто клієнти, які купили один продукт, купують і другий, а то і третій. У випадку з ПЗ – це можуть бути додаткові ліцензії, або покупка «старшої версії». Таким чином, за весь час, що компанія працює з клієнтом, вона отримає з нього грошей більше, ніж коштує один продукт [3]-[6].

Є й інші фактори, які можуть вплинути на прийняття рішення [3]-[6]:

- фізична неможливість надання продукту або аналога;
- накладні витрати, пов'язані з продажем;
- неможливість вийти за межі регламентів і налагоджених процесів;
- юридичні аспекти.

1.1.3 Продаж музикальних композицій у якості цифрових товарів

Рано чи пізно кожен виконавець замислюється про те, щоб почати продавати свою творчість і заробити на цьому капітал. Деяким щастить більше: їх помічає продюсерський центр (так званий, рекорд-лейбл), пропонує контракт і вже самостійно розкручує і продає музику [5]-[6]. Музиканту залишається лише писати музику і отримувати свій відсоток з продажів. Мінусом в даному випадку є цей самий відсоток. Як правило, він досить низький, щось близько десяти відсотків з продажу. З іншого боку можна зосередитися на творчості і не забивати голову комерційними питаннями [5]-[6].

Але не всім щастить потрапити під крило лейблу. Багато виконавців, навіть з хорошим матеріалом, залишаються в тіні [5]-[6]. У них є свої слухачі, і вони навіть дають живі виступи, проте не можуть зацікавити компанії звукозапису. Благо в наш час, коли Інтернет – магазини поступово витісняють реальні, у таких музикантів з'являється реальний шанс продати свої композиції і отримати чесно зароблені гроші. Мова, звичайно, піде про популярних майданчиках, на яких можна продавати свою музику [5]-[6].

1.2 Дослідження існуючих аналогів

На сьогоднішній день на просторах Інтернету існує достатньо багато онлайн майданчиків [7]-[9], що у якості цифрових товарів пропонують придбати музикальні композиції. Проте слід зазначити, що більшість з них можна розділити на дві групи.

Перша група надає саме можливість купити ту чи іншу музику. Умовно кажучи, виконавець або автор може завантажити свій альбом, а користувач може купувати окремі пісні або цілий альбом. Такий ресурс зазвичай не надає можливість прослуховування музики (окрім демо-записів) або збереження музики до плейлистів користувача.

Друга група надає можливість при регулярній оплаті слухати музику, що доступна у сервісі. Тобто користувач має так би мовити «безлімітний» доступ до усієї фонотеки сервісу, але за умови регулярної передплати.

Під час виконання дослідження зосередимо увагу на ресурсах, першої групи.

1.2.1 Сервіс «Bandcamp»

Можливо, назва даного сервісу вам ні про що не скаже, але його можна сміливо рекомендувати тим, хто всерйоз вирішив зайнятися продажем музики [7]. Так, він менш розпіарний, ніж вищезгадані гіганти індустрії, але ігнорувати Bandcamp не варто [7].

Важливий момент – це не зовсім магазин в звичному розумінні. Всі функції магазину присутні, проте є ряд відмінностей. Ви викладаєте свої треки, оформляєте свою унікальну сторінку і отримуєте під неї власний домен третього рівня [7]. За свою музику ви самі встановлюєте ціну. Мільйонів і навіть тисячі доларів там не заробити, але отримувати якийсь прибуток можна [7].

Bandcamp допускає завантаження тільки аудіофайлів в loseless-форматі (WAV/AIFF/FLAC), мотивуючи це якістю звуку при стрімінгу [7]. Цілком солідний аргумент, та й проблемою це навряд чи може стати – по виходу зі студії на руках у музикантів майже завжди є файли в потрібному форматі [7].

На рисунку 1.1 наведено приклад головної сторінки сервісу «Bandcamp».

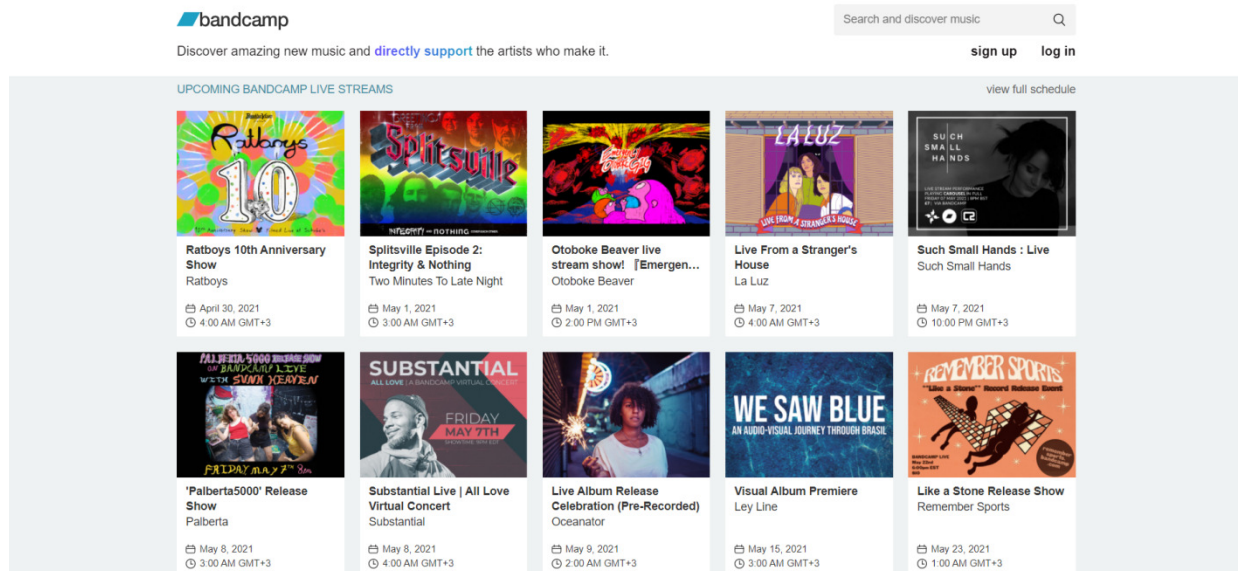


Рисунок 1.1 – Головна сторінка сервісу «Bandcamp»

1.2.2 Сервіс «Soundcloud»

«Soundcloud» існує в безкоштовній і платній версії, а різниця полягає в малоцікавому додатковому функціоналі і обмеженням по завантаженню треків [8]. У безкоштовній версії можна завантажити до 100 хвилин матеріалу, платна версія ж гарантує безліміт. Така модель продажу сервісу, здається, і привела до боргової ями Soundcloud. Більшість ігнорує платну версію, адже 100 хвилин – це мінімум 2 альбоми. А головне, завжди можна знести мало актуальний матеріал і звільнити простір [8].

Також, Soundcloud не дозволяє слухачам завантажувати треки – тільки стримить. Для скачування у розробників був окремий сервіс SoundDrain, який працював з ще одним міні-сервісом для платежів Minno [8]. Все для користувачів: просто, зручно і гнучко. Нажаль, через малу прибуток розробники Soundrain і Minno згорнули сервіс пару років тому. Тепер продати можна лише із зазначенням Buy-Link на сервіси типу iTunes або Bandcamp при завантаженні композиції. Тобто, Soundcloud – на 99% є ознайомчим майданчиком-стрімінгом [8].

Маючи в арсеналі відмінний функціонал з аналізу і статистики прослуховувань, можливість таргетити контент і величезну базу користувачів, Soundcloud все-таки залишається популярним [8].

На рисунку 1.2 наведено приклад сторінки сервісу «Soundcloud».

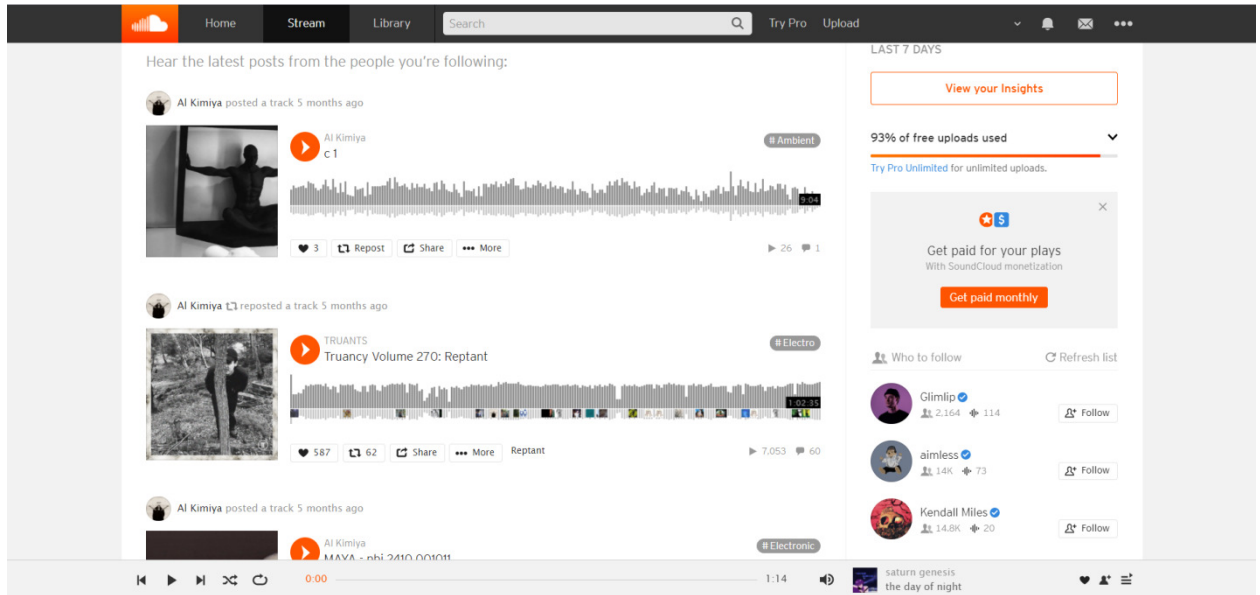


Рисунок 1.2 – Приклад сторінки сервісу «Soundcloud»

1.2.3 Застосунок «iTunes Store»

За статистикою магазином «iTunes Store» користуються порядком півмільярда чоловік [9]. Щодня від туди викачуються мільйони музичних композицій. Число користувачів і продажів зростає щодня. Сьогодні зустріти людину, яка б не чула про iTunes Store просто неможливо. Навіть якщо ця людина ніколи ним не користувався. Тому логічно звернути увагу на цей західний майданчик для продажу музики [9].

Незважаючи на те, що плюси і переваги iTunes Store очевидні, у майданчика є свої мінуси і недоліки, які, можливо, відлякують більшість музикантів. Розглянемо спочатку переваги [9].

По-перше, величезна аудиторія. Комуś буде складно в це повірити, але в ряді країн люди купують музику просто для ознайомлення. Вони не

знають виконавця, ніколи не чули його творів [9]. Але вони куплять альбом і якщо він їм не сподобається, залишать довгий і неприємний відгук на офіційній сторінці музиканта в iTunes. Тепер уявіть, скільки людей з маси в 500 мільйонів, можуть завантажити ваш альбом для ознайомлення? Навіть тисяча покупок буде приємним сюрпризом і непоганий надбавкою до зарплати [9].

По-друге, автоматична розкрутка. Користувачі iTunes, як втім і будь-якого іншого великого онлайн-магазину, люблять залишати відгуки та коментарі [9]. А немає реклами краще, ніж рекомендація одного. Виставивши свій опус в iTunes можна в кілька разів підвищити число своїх слухачів. Вони будуть приходити за рекомендацією друзів і товаришів, ті в свою чергу порекомендують вашу музику ще кому-небудь. Запуститься ланцюгова реакція. Успішна розкрутка неминуче веде до підвищення числа продажів, а відповідно і прибутку [9].

На рисунку 1.3 наведено приклад сторінки застосунку «iTunes Store».

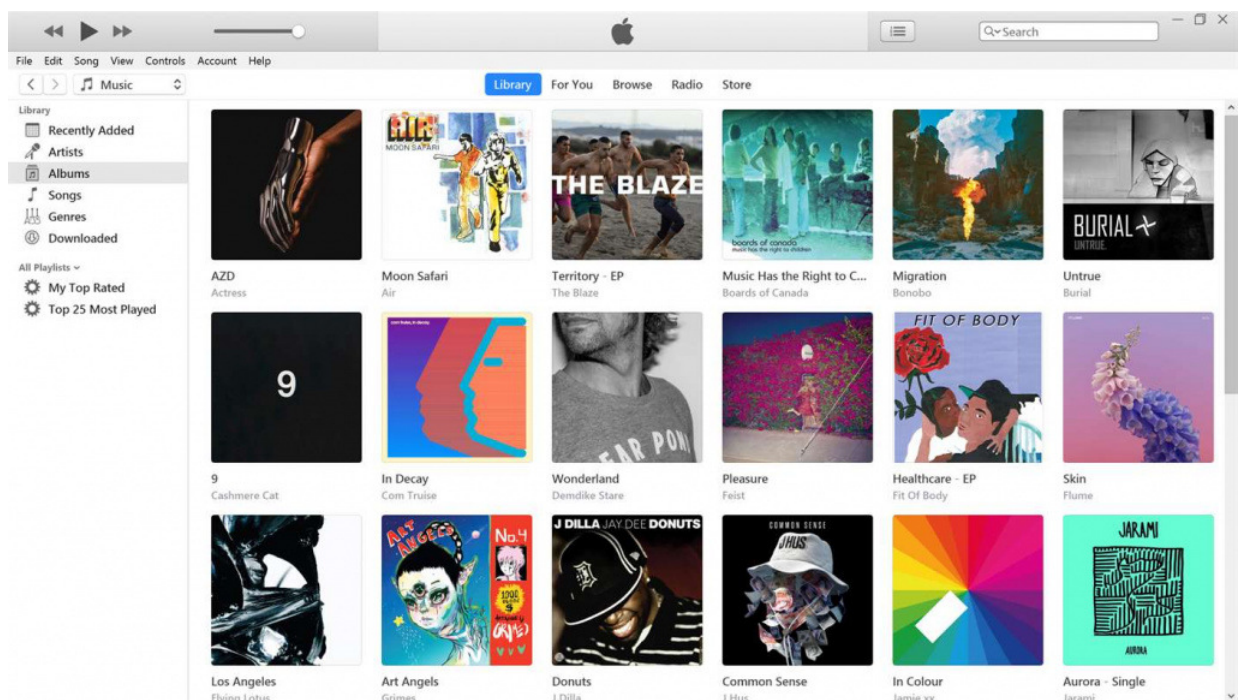


Рисунок 1.3 – Приклад сторінки застосунку «iTunes Store»

1.2.4 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). У таблиці будуть порівнюватися попередні аналоги із онлайн-крамницею, що розробляється – «Music Hall». Серед критеріїв виділимо наступні:

- додаткові вимоги до авторів – чи є у платформи додаткові вимоги для верифікації авторів, що затримає та ускладнить роботу із сервісом;
- додаткові вимоги до аудіо файлів – чи є додаткові вимоги до форматів аудіофайлів, які розміщуються та надаються користувачу у якості покупки;
- наявність обмежень за ціною – чи є певні обмеження за найнижчою або найвищою ціною аудіо файлу;
- функціонал для ознайомлення із альбом або композицією – чи надає сервіс функціонал для ознайомлення із демо-версією продукту.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	Bandcamp	Soundcloud	iTunes	Music Hall
Додаткові вимоги до авторів	Не має	Не має	Наявні	Не має
Додаткові вимоги до аудіофайлів	Наявні	Не має	Наявні	Не має
Наявність обмежень за ціною	Не має	Є (продаж через сторонні сервіси)	Є (додаткові виплати самій платформі)	Не має
Функціонал для ознайомлення із альбомом//композицією	Наявний	Наявний	Обмежений	Наявний

Як видно із порівняння більшість ресурсів мають свої недоліки та нюанси, що ускладнюють їх використання, особливо для молодих авторів.

Так, наприклад, iTunes вимагає складну верифікацію авторів з використанням особливих документів, оформлення яких пов'язане із посольством США. Саме тому платформа не підходить для більшості молодих авторів. Більш того файли підвантажуються в форматі AAC, що в більшості випадків пов'язано із використанням техніки Apple. Bandcamp більш розрахований для молодих авторів, але також має додаткові вимоги до аудіофайлів. А найважливіше, що сервіс домі не дуже відомий та має досить обмежену аудиторію. Soundcloud – платформа виключно для демонстрації музики і сам продаж-купівля виконуються з використанням сторонніх ресурсів, що затримує розповсюдження та використання.

Саме тому, можна заключити, що досі актуальною є задача розробки нового сервісу, що надавав би можливість оформити продаж-купівлю аудіофайлів в якості цифрових товарів.

1.3 Постановка завдання

За результатами аналізу проблемної області та дослідження існуючих аналогів для вирішення проблемних питань, сформулюємо постановку задачі.

Метою є розробка нової музичної онлайн-крамниці. Система повинна забезпечувати можливості зручних:

- перегляду фонотеки;
- вибору композицій;
- створення та оплати замовлення;
- редагування фонотеки системи;
- опрацювання замовлень, коментарів, тощо.

Межі системи:

- система буде мати за основу одне з програмних забезпечень для створення інтерфейсу онлайн-магазину;

- система буде використовувати систему керування базами даних (СКБД) MySQL для роботи з базою даних (БД);
- система буде використовувати модуль оплати;
- система буде використовувати один із існуючих алгоритмів шифрування даних для забезпечення безпеки конфіденційних даних.

Розглянемо вимоги до розроблюваної системи за їх типом [10]-[11].

1.3.1 Функціональність системи

Функціональні вимоги

- система повинна приймати та обробляти запити користувачів;
- система повинна виводити каталог товарів користувачу;
- система повинна по електронній пошті відсилати користувачу підтвердження замовлення;
- система повинна надавати змогу змінювати, додавати та видаляти інформацію про товар;
- система повинна забезпечувати інтерфейс з модулем оплати;
- система повинна видавати звіти на вимогу службових осіб, які працюють у компанії.

1.3.2 Вимоги до інтерфейсу

Для забезпечення ергономічності, зручності та зрозумілості інтерфейс розроблюваного веб-застосунку повинен бути зручним та інтуїтивно-зрозумілим.

1.3.3 Не функціональні вимоги до продукту

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини застосунку:

- серверна операційна системи (ОС);
- веб-сервер ХАМРР;
- інтерпретатор РНР5 та РНР7;
- СКБД MySQL-сервер.

Мінімальні вимоги до клієнтської частини:

- персональний комп'ютер (ПК) із встановленою ОС;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB для безперебійної роботи;
- підключення до мережі Internet;
- встановлений веб-браузер останньої версії.

1.3.4 Оцінка надійності, продуктивності і підтримки

Для забезпечення достатньої надійності та продуктивності система має відповідати певним вимогам:

- система повинна працювати постійно без збою;
- займатися підтримкою системи повинен адміністратор та модератор;
- кожен користувач системи повинен мати ідентифікований логін та пароль;
- бажано щоб програма працювала без додаткового налаштування;
- дана система безкоштовна в межах компанії.

1.3 Висновки за розділом 1

Виконано огляд проблеми розробки та підтримки електронного продажу цифрових товарів. Виділено основні шляхи для направи для розроблення нового ПЗ у вигляді онлайн-крамниці для продажу-купівлі

цифрових товарів у вигляді музичних композицій. Систему буде розроблено у вигляді кроссплатформного веб-застосунку.

Розробка та підтримка музичної онлайн-крамниці має надати систему, що забезпечує можливості зручних перегляду каталогу, вибору цифрових товарів, оформлення замовлень, поповнення асортименту, опрацювання замовлень. Система повинна бути кроссплатформною. Система також повинна бути простою й зрозумілою користувачу.

2 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

У попередньому розділі було проведено аналіз та дослідження існуючих аналогів майданчиків та онлайн-магазинів продажу музичних композицій. Виходячи з проведеного аналізу, було сформульовано основні можливості, якими повинна оперувати система, що проєктується.

Що стосується технічних та програмних можливостей, то система повинна забезпечувати можливість швидко здійснювати покупку музичних творів з урахуванням всіх нюансів.

Даний розділ містить опис питань, які стосуються безпосередньо структури системи, що розробляється, вибору середовища проєктування, а також визначаються основні та специфічні архітектурні рішення, які має система.

2.1 Вибір мови програмування

PHP – це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом. Говорячи простіше, PHP це мова програмування, спеціально розроблений для написання веб-застосунків (сценаріїв), що виконуються на веб-сервері [12].

Синтаксис мови бере початок з C, Java і Perl. PHP досить простий для вивчення. Перевагою PHP є надання веб-розробникам можливості швидкого створення динамічно генерованих веб-сторінок [12].

Важливою перевагою мови PHP перед такими мовами, як мови Perl і C полягає в можливості створення HTML документів з впровадженими командами PHP [12].

Значною відмінністю PHP від будь-якого коду, що виконується на стороні клієнта, наприклад, JavaScript, є те, що PHP-скрипти виконуються на стороні сервера. Більш того, навіть можна налаштовувати свій сервер таким чином, щоб HTML-файли оброблялися процесором PHP, так що клієнти

навіть не зможуть дізнатися, чи отримують вони звичайний HTML-файл або результат виконання скрипта [12]-[13].

PHP дозволяє створювати якісні Web-додатки за дуже короткі терміни, отримуючи продукти, легко модифікуються і підтримувані в майбутньому. PHP простий для освоєння, і разом з тим здатний задовольнити запити професійних програмістів [12]-[13].

Мова PHP постійно вдосконалюється, і їй напевно забезпечено довге домінування в області мов веб-програмування, принаймні, найближчим часом [12]-[13].

Головним фактором мови PHP є практичність. PHP повинна надати програмісту засоби для швидкого і ефективного вирішення поставлених завдань. Практичний характер PHP обумовлений п'ятьма важливими характеристиками [14]:

- традиційність;
- простота;
- ефективність;
- безпека;
- гнучкість.

Існує ще одна «характеристика», яка робить PHP особливо привабливим: безкоштовне поширення. Причому, з відкритим вихідним кодом [14].

Традиційність. Мова PHP буде здаватися знайомим програмістам, які працюють в різних областях. Багато конструкцій мови запозичені з C і Perl [12]-[13].

Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP – мова, що поєднує переваги Perl і C і спеціально націлений на роботу в Інтернеті, мова з універсальним (правда, за деякими застереженнями) і ясним синтаксисом [12]-[14].

І хоча PHP є досить молодою мовою, вона знайшла таку популярність серед веб-програмістів, що на даний момент є мало не найпопулярнішою мовою для створення web-додатків (скриптів) [12]-[14].

Простота. Сценарій PHP може складатися з 10 000 рядків або з одного рядка – все залежить від специфіки вашого завдання. Вам не доведеться довантажувати бібліотеки, вказувати спеціальні параметри компіляції або що-небудь в цьому роді. Механізм PHP просто починає виконувати код після першої екрануючої послідовності і продовжує виконання до того моменту, коли він зустрине парну екрануючу послідовність. Якщо код має правильний синтаксис, він виконується в точності так, як вказав програміст [12]-[14].

PHP – мова, яка може бути вбудована безпосередньо в HTML-код сторінок, які, в свою чергу будуть коректно оброблятися PHP-інтерпретатором [12]-[14]. Можна використовувати PHP для написання CGI-сценаріїв і позбутися від безлічі незручних операторів виведення тексту. Можна залучати PHP для формування HTML-документів, позбувшись від безлічі викликів зовнішніх сценаріїв [12]-[14].

Велика різноманітність функцій PHP позбавлять від написання багаторядкових користувацьких функцій на C або Pascal [12]-[14].

Ефективність. Ефективність є виключно важливим фактором при програмуванні для багатокористувацьких середовищ, до числа яких відноситься і веб [12]-[14].

Дуже важлива перевага PHP полягає в його рушії. Рушій PHP не є ні компілятором, ні інтерпретатором. Він є транслює інтерпретатором. Такий пристрій рушія PHP дозволяє обробляти сценарії з досить високою швидкістю [12]-[14].

За деякими оцінками, більшість PHP-сценаріїв (особливо не дуже великих розмірів) обробляються швидше аналогічних їм програм, написаних на Perl. Однак, щоб не робили розробники PHP, відкомпільовані виконувані файли будуть працювати значно швидше – в десятки, а іноді і в сотні разів.

Але продуктивність PHP цілком достатня для створення цілком серйозних веб-застосунків [12]-[14].

Безпека. PHP надає в розпорядження розробників і адміністраторів гнучкі та ефективні засоби безпеки, які умовно діляться на дві категорії: засоби системного рівня і засоби рівня за стосунку [12]-[14].

Засоби безпеки системного рівня. У PHP реалізовані механізми безпеки, що знаходяться під управлінням адміністраторів; при правильному налаштуванні PHP це забезпечує максимальну свободу дій і безпеку [12]-[14]. PHP може працювати в так званому безпечному режимі, який обмежує можливості застосування PHP користувачами по ряду важливих показників. Наприклад, можна обмежити максимальний час виконання і використання пам'яті (неконтрольований витрата пам'яті негативно впливає на швидкодію сервера). За аналогією з cgi-bin адміністратор також може встановлювати обмеження на каталоги, в яких користувач може переглядати і виконувати сценарії PHP, а також використовувати сценарії PHP для перегляду конфіденційної інформації на сервері (наприклад, файлу passwd) [12]-[14].

Засоби безпеки рівня програми. У стандартний набір функцій PHP входить ряд надійних механізмів шифрування. PHP також сумісний з багатьма застосунками незалежних фірм, що дозволяє легко інтегрувати його з захищеними технологіями електронної комерції [12]-[14]. Інша перевага полягає в тому, що вихідний текст сценаріїв PHP не можна переглянути в браузері, оскільки сценарій компілюється до його відправки за запитом користувача. Реалізація PHP на стороні сервера запобігає викраденню нетривіальних сценаріїв користувачами, знань яких вистачає хоча б для виконання команди View Source [12]-[14].

Гнучкість. Оскільки PHP є вбудовуванню мовою, вона відрізняється винятковою гнучкістю по відношенню до потреб розробника [12]-[14]. Хоча PHP зазвичай рекомендується використовувати в поєднанні з HTML, він з таким же успіхом інтегрується і в JavaScript, WML, XML та інші мови. Крім того, добре структуровані застосунки PHP легко розширюються в міру

необхідності (втім, це відноситься до всіх основних мов програмування) [12]-[14].

Немає проблем й з залежністю від браузерів, оскільки перед відправкою клієнту сценарії PHP повністю компілюються на стороні сервера. По суті, сценарії PHP можуть передаватися будь-яким пристроям з браузерами. Програмісти, що займаються допоміжними утилітами, можуть запускати PHP в режимі командного рядка [12]-[14].

Оскільки PHP не містить коду, орієнтованого на конкретний веб-сервер, користувачі не обмежуються певними серверами (можливо, незнайомими для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold і Zeus – PHP працює на всіх перерахованих серверах. Оскільки ці сервери працюють на різних платформах, PHP в цілому є платформно-незалежною мовою і існує на таких платформах, як UNIX, Solaris, FreeBSD і Windows [12]-[14].

Нарешті, засоби PHP дозволяють програмісту працювати із зовнішніми компонентами, такими як Enterprise Java Beans або COM-об'єкти Win32. Завдяки цим новим можливостям PHP займає гідне місце серед сучасних технологій і забезпечує масштабування проєктів до необхідних меж [12]-[14].

Безкоштовне розповсюдження. Стратегія Open Source, і поширення вихідних текстів програм в масах, зробило безсумнівно благотворний вплив на багато проєктів, в першу чергу – Linux, хоча і успіх проєкту Apache сильно підкріпив позиції прихильників Open Source. Сказане відноситься і до історії створення PHP, оскільки підтримка користувачів з усього світу виявилася дуже важливим фактором у розвитку проєкту PHP [12]-[14].

Прийняття стратегії Open Source і безкоштовне поширення вихідних текстів PHP зробило неоціненну послугу користувачам. До того ж, чуйне співтовариство користувачів PHP є свого роду «колективною службою підтримки», і в популярних електронних конференціях можна знайти відповіді навіть на найскладніші питання [12]-[14].

Для обґрунтування вибору мови програмування проведемо порівняння найбільш популярних мов програмування для розробки веб-застосунків. Результати порівняння представимо в табличному вигляді – таблиця 2.1. У якості критеріїв виділимо:

- швидкість роботи застосунку – швидкість роботи розробленого застосунку та відповідно необхідність подальшої оптимізації;
- додаткові бібліотеки – чи вимагає розробка встановлення додаткових бібліотек та розширень для розробки;
- парадигми програмування – парадигми програмування, що підтримуються моєю парадигми;
- спільнота та матеріали – наявність та розповсюдженість спільноти розробників та додаткових матеріалів для вивчення мови програмування;
- фреймворки – наявність фреймворків для спрощення та прискорення реалізації архітектури розробки.

Таблиця 2.1 – Порівняння використання фреймворку та CMS під завдання замовника

	PHP	Python	C# (+.Net)
Швидкість роботи застосунку	+	+	+
Додаткові бібліотеки	++	+	±
Парадигми програмування	++	++	++
Спільнота та матеріали	++	+	+
Фреймворки	++	+	±

Виходячи з порівняння розробка з використанням PHP буде більш швидкою, не буде потребувати додаткових витрат ресурсів часу або встановлення додаткових бібліотек. Більш того така розробка буде більш простою в подальшій підтримці.

2.2 Вибір фреймворку для розробки

Laravel – безкоштовний веб-фреймворк з відкритим кодом, призначений для розробки з використанням архітектурної моделі MVC [15]-[16]. Після виходу PHP7 в порівнянні з PHP5, скрипти стали швидше і почали використовувати набагато менше оперативної пам'яті, а в зв'язці з Zend OPcache показують чудові результати. Зокрема сервіс Laravel Forge налаштовує Zend OPcache для досягнення максимальної продуктивності [15]-[16].

Саме тому, коли йдеться про продуктивність того чи іншого PHP-фреймворку, то завжди проводять тестування без кешування, роботи з БД або файлами, в основному здійснюючи безліч викликів до звичайної PHP сторінки [15]-[16]. В цьому плані даний PHP-фреймворк істотно нічим не відрізняється від всіх інших, але коли мова йде про масштабованість, гнучкості, універсальності вбудованих механізмів кешування і швидкості розробки, саме тоді Laravel показує всю свою гнучкість і міць [15]-[16].

Сам Laravel постійно вдосконалюється і слід сучасним тенденціям. Вивчаючи його, ви не відстанете від світу веб-розробки, головне – не зациклюватися на якійсь конкретній версії фреймворку, а розвиватися разом з ним. Для цього необхідно також вивчати нововведення Laravel [15]-[16].

Щорічно проводяться різні конференції, які можна подивитися також і онлайн. Слід зазначити основні можливості Laravel, щоб можна було оцінити масштаб [15]-[16].

Model View Controller (MVC). PHP-фреймворк побудований на базі відомих і надійних компонентів Symfony [15]-[16].

Необхідні модулі для фреймворку підключаються у вигляді пакетів-провайдерів (service provider). У версії Laravel 5.5 досить просто встановити пакет через Composer, і він відразу буде доступний, без необхідності щось писати в коді [15]-[16].

Код фреймворку відокремлений від коду розробника, кожен компонент легко розширюється [15]-[16].

Код веб-проєкту, CSS, JS, HTML-код сторінок розділені в окремі директорії. Фреймворк використовує чудовий шаблонізатор Blade, який дозволяє відокремити верстку від PHP-коду. Сам шаблонізатор настільки простий, що навіть початківець HTML-верстальник зможе легко його подужати [15]-[16].

Зручна маршрутизація, валідація вхідних параметрів.

Кешування, робота зі сховищами файлів, робота з різними БД.

Міграції для бази даних, можна змінювати структуру БД і відкочувати зміни.

Черги завдань, планувальник завдань, консоль, робота з SSH.

Величезний функціонал Eloquent ORM дозволяє повністю убезпечити себе від атак типу SQL Injection, а також завантажувати дані з декількох таблиць (вирішуючи проблему N+1) або ж обробляти дані з БД частинами [15]-[16].

Laravel Collections – можна сказати, що це PHP масиви, але з дуже просунутими можливостями, які економлять масу часу [15]-[16].

Кешування файлів маршрутизації, файлів конфігурації, шаблонів. Це прискорює роботу фреймворка. Відправка повідомлень різними способами: пошта, Slack і т.д., можете дописати самі. Підтримка WebSockets для створення справжніх інтерактивних додатків. Підтримка багатомовності: легко додавайте будь-які мови, а пакет Laravel-lang вже містить безліч перекладів. Інтерфейс командного рядка artisan, який дозволяє генерувати моделі, контролери, повідомлення, запускати завдання з черги завдань і багато іншого [15]-[16].

Laravel Tinker-додатковий пакет, який дозволяє працювати з кодом проєкту з командного рядка.

Величезні можливості для тестування веб-проєкту, включаючи заповнення бази даних тестовими даними [15]-[16].

У фреймворку є навіть власний сайт з бібліотекою пакетів.

У таблиці 2.2 наведено порівняння популярних фреймворків за критеріями. Серед характеристик для порівняння виділімо:

- спільнота та простота вивчення – кількість та активність спільноти, що активно використовує фреймворк та може допомогти із вивченням;
- оновлення та підтримка – швидкість оновлення та релізів фреймворків;
- простота розробки представлень – чи є вбудовані механізми, що спрощують розробку фронтенду, в тому числі представлень;
- підтримка СКБД – кількість СКБД, що підтримує фреймворк.

Таблиця 2.2 – Порівняння CMS

	Laravel	Yii2
Спільнота та простота вивчення	++	++
Оновлення та підтримка	++	–
Простота розробки представлень	±	++
Підтримка СКБД	++	++

Як видно із порівняння всі фреймворки надають широку палітру для розробки веб-застосунків в тому числі для продажу-купівлі цифрових товарів через Інтернет. Проте саме Laravel надає більше інструментів для повноцінної розробки, підтримки та розвитку онлайн-крамниці. Тому вибір був зроблено саме на користь Laravel.

2.3 Загальна структура додатку

Для побудови архітектурного каркасу ПЗ буде використано шаблон проєктування MVC [17]. Ця схема використання декількох шаблонів проєктування, за допомогою яких модель даних програми, користувацький інтерфейс і взаємодія з користувачем розділені на три окремих компоненти.

View – відповідає за збір даних від користувачів в різних формах та обробку їх у відповідності зі специфікацією [17].

Controller – обробляє дані, отримані від користувача для передачі на наступний рівень [17].

Model – сукупність правил, принципів, залежностей поведінки об'єктів предметної області. Це реалізація предметної області в інформаційній системі [17].

На рисунку 2.1 показано, який вигляд має архітектура ПЗ на основі предметної області з використанням шаблону проєктування MVC.

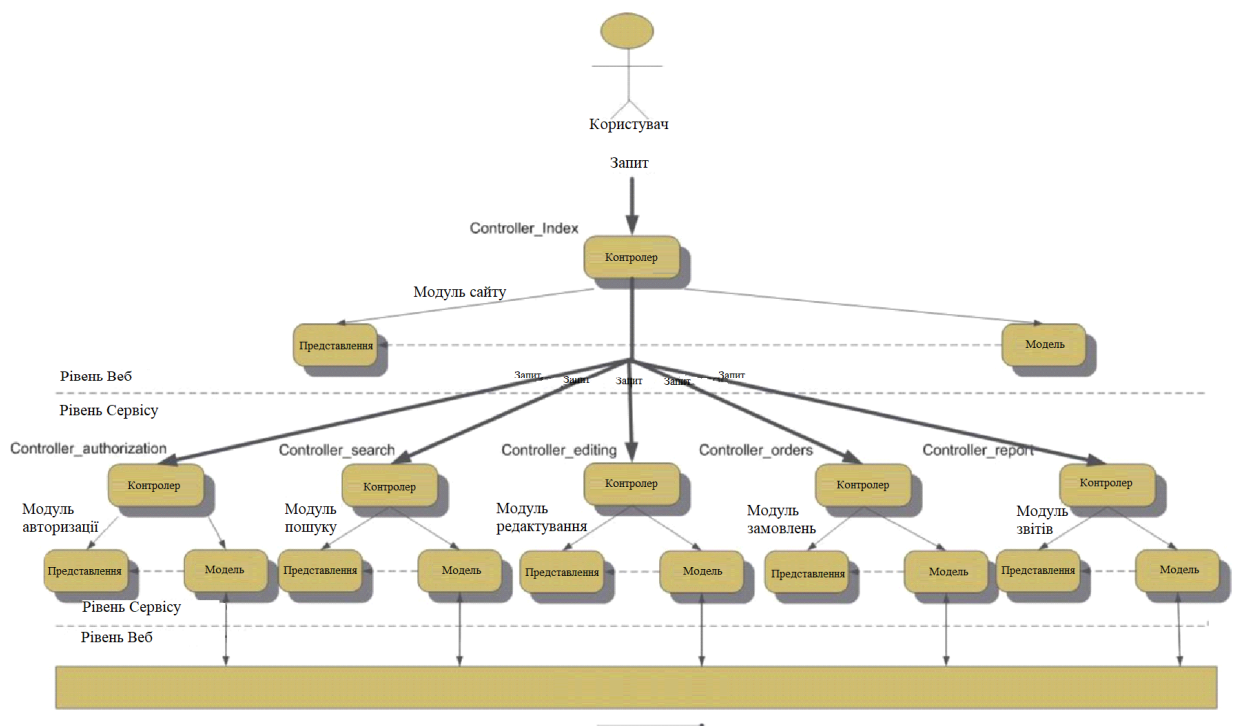


Рисунок 2.1 – MVC схема для користувача

2.4 Висновки за розділом 2

Таким чином, архітектура програмного засобу дозволяє вирішувати задачі, що були поставлені перед системою, пов'язані із продажами цифрових послуг. Засоби, що використовуються, забезпечують реалізацію функцій програми.

Для програмної реалізації можливостей ПЗ – музичної онлайн-крамниці, виділених під час аналізу у попередньому розділі, проведено вибір засобів та обґрунтовано структуру системи, що проектується.

Аналізуючи обрану структуру ПЗ, можна зробити висновок, що вона є лаконічною, але, разом з тим, досить працездатною, використовує ресурси ПК раціонально.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проектування системи на основі предметної області

Unified Modeling Language (UML) – уніфікована мова моделювання, що використовується у парадигмі об'єктно-орієнтованого програмування та є невід'ємною частиною уніфікованого процесу розроблення ПЗ [18]-[19]. UML заснована на діаграмах, що дають можливість представити систему у такому вигляді, щоб її можна було легко перевести у програмний код.

Діаграми класів – статичні структурні діаграми, що описують структуру системи, демонструють класи систем, їх атрибути, методи і залежності між класами [18]-[19].

Діаграма класів системи, що проектується приведена на рисунку 3.1.

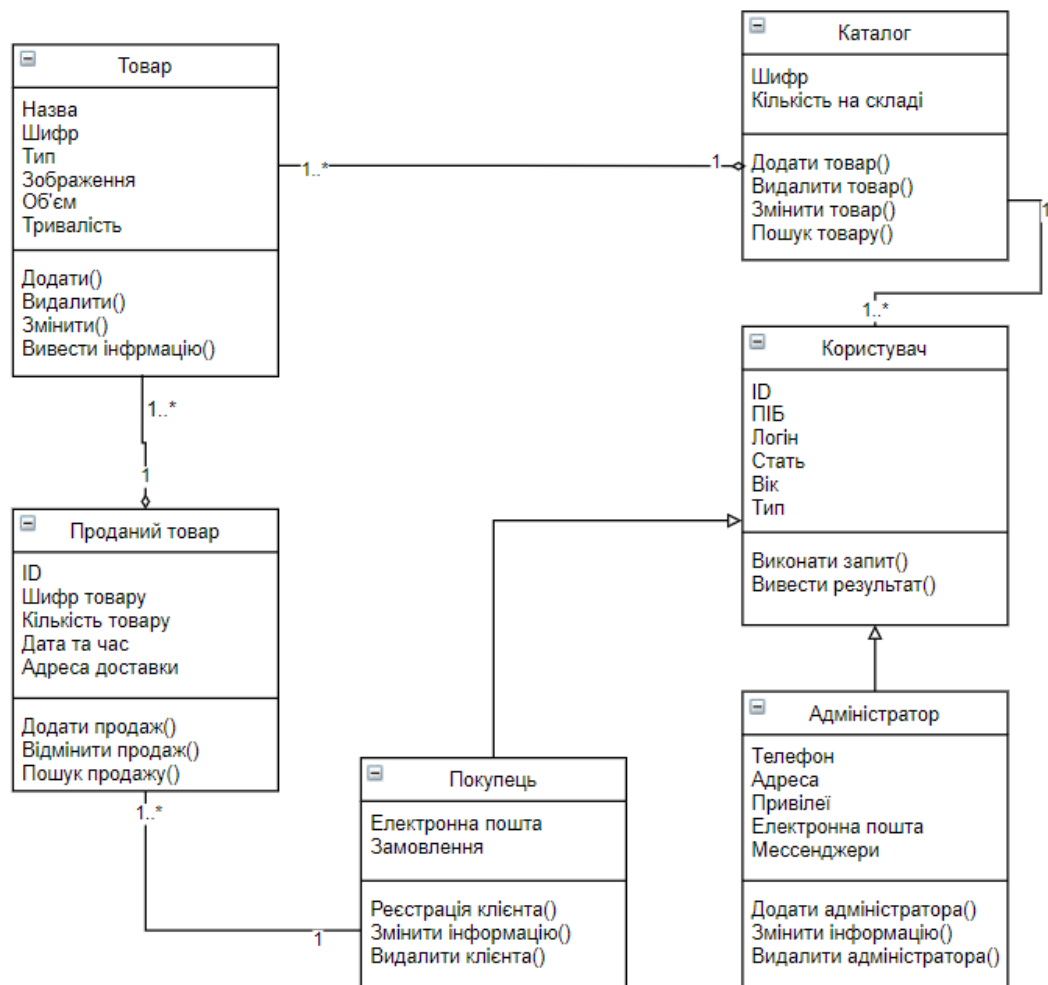


Рисунок 3.1 – Діаграми класів системи

Діаграми пакетів – структурні діаграми, основним вмістом яких є пакети і відношення між ними. Ці діаграми слугують, в першу чергу, для організації елементів в групи по яким-небудь ознакам з метою спрощення структури і організації роботи з моделлю системи [18]-[19].

Діаграма пакетів системи, що проєктується приведена на рисунку 3.2.

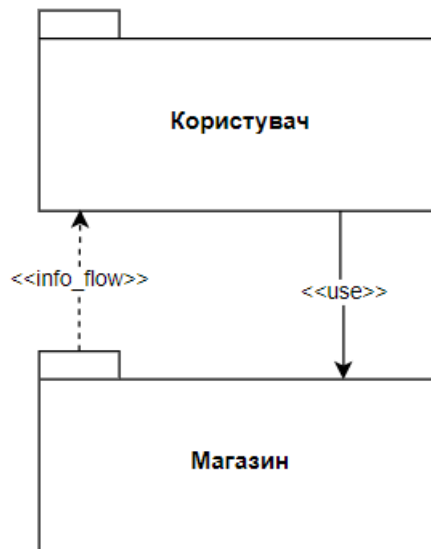


Рисунок 3.2 – Діаграми пакетів системи

Діаграма компонентів – статична структурна діаграма, що показує розбиття програмної системи на структурні компоненти і зв'язки (залежності) між компонентами. В якості фізичних компонент можуть виступати файли, бібліотеки, модулі, виконавчі файли, пакети і т. п. [18]-[19].

Діаграма компонентів системи, що проєктується приведена на рисунку 3.3.

Діаграми прецедентів застосовуються для моделювання вигляду системи з точки зору прецедентів (або варіантів використання). Даний вид діаграм використовується для моделювання динамічних аспектів системи, так само як і діаграми станів та діяльності [18]-[19].

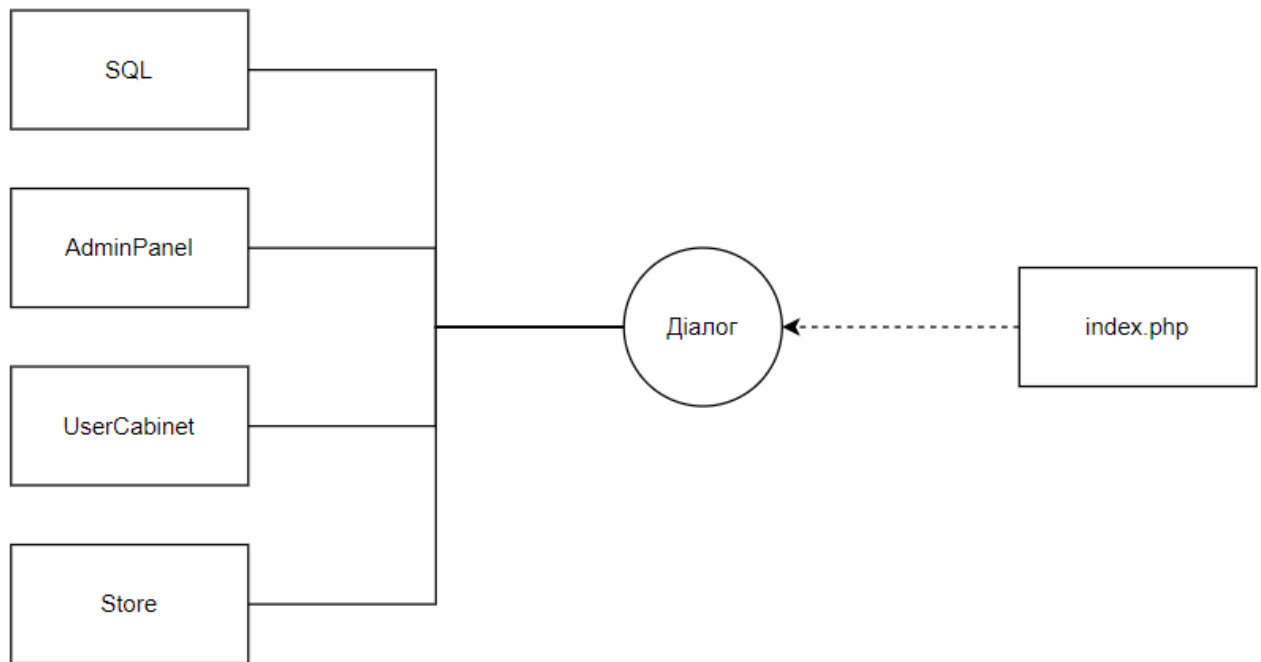


Рисунок 3.3 – Діаграми компонентів системи

Діаграма прецедентів – діаграма, на якій зображено відношення між акторами та прецедентами в системі [18]-[19].

Діаграми прецедентів зазвичай включають в себе:

- прецеденти;
- акторів;
- відношення залежності, узагальнення та асоціації.

Діаграма прецедентів для системи, що проектується, приведена на рисунок 3.4.

Специфікація сценарію використання.

У даному прецеденті описується загальна робота системи. Представимо потік подій.

Базовий потік (Б – крок базового потоку):

Б1 Користувач запускає систему;

Б2 Користувач авторизується;

Б3.1.1 Адміністратор додає товар;

Б3.1.2 Адміністратор змінює товар;

Б3.1.3 Адміністратор видаляє товар;

Б3.2 Покупець шукає товар;

Б3.3.1 Покупець придбав товар;

Б3.3.2 Покупець замовляє товар;

Б3.3.3 Покупець оплачує товар;

Б4 Вихід з системи.

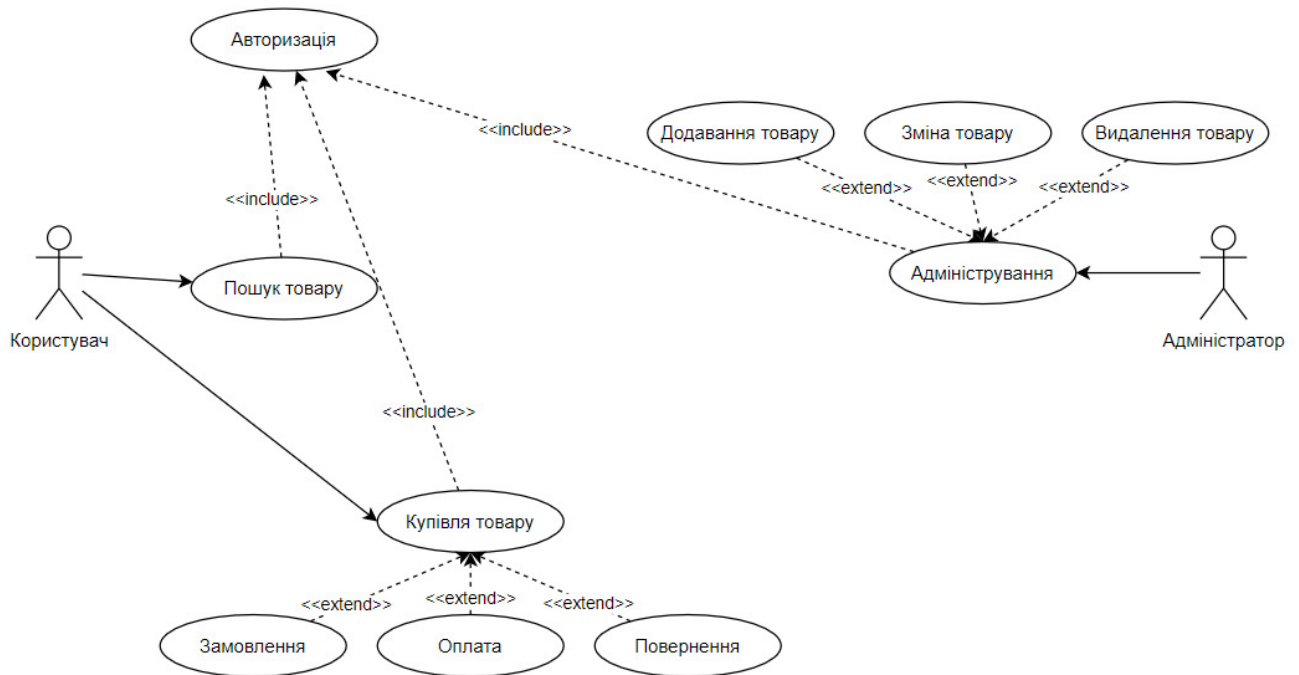


Рисунок 3.4 – Загальна діаграма прецедентів

Альтернативні потоки (А – крок альтернативного потоку).

А.1 Авторизація три рази пройшла не успішно, то на кроці Б2;

Система блокується:

А.2 Покупець відмовляється від товару;

Процедура повернення:

Передумови:

Комп'ютер підключений до локальної мережі;

Постумови:

Користувач може виконувати роботу з системою.

Діаграми станів зображають всі можливі стани, в яких може знаходитись конкретний об'єкт, а також зміни стану об'єкту, які відбуваються в результаті впливу деяких подій на цей об'єкт. У більшості об'єктно-орієнтованих методів діаграми станів будуються для єдиного класу, щоб показати динаміку поведінки єдиного об'єкту [18], [19].

Зазвичай діаграми станів складаються з наступних елементів:

- прості та складні стани;
- переходи разом з асоційованими подіями та діями.

Діаграма станів для системи, що проєктується, приведена на рисунку 3.5.

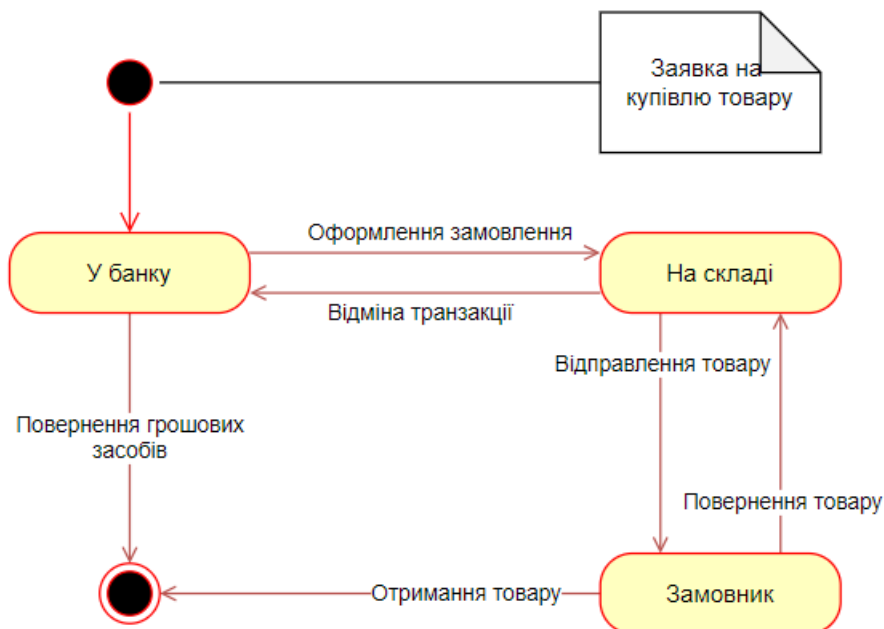


Рисунок 3.5 – Діаграма станів придбання товару

Діаграма діяльності показує переходи між видами діяльності. Модель видів діяльності може подавати в графічній формі потік подій для прецеденту [18]-[19].

Кожен прецедент можна моделювати за допомогою одного або декількох графів видів діяльності. Діаграма діяльності за своєю сутністю є

блоком-схемою, яка демонструє, як потік управління переходить від однієї діяльності до іншої.

Подія, джерелом якої служить суб'єкт-прецедент, що ініціює, це та ж сама подія, що запускає виконана графа видів діяльності. Процес виконання послідовно переходить від одного стану до іншого.

Діаграма діяльності у загальному випадку складається з:

- станів діяльності та станів дії;
- переходів.

Діаграма діяльності для системи, що проектується, приведена на рисунку 3.6.

Діаграма кооперації призначена для специфікації структурних аспектів взаємодії. Головна особливість діаграми кооперації полягає в можливості графічно уявити не тільки послідовність взаємодії але і всі структурні відносини між об'єктами що беруть участь в цій взаємодії [18]-[19].

Діаграми кооперації для системи, що проектується, приведені на рисунках 3.7, 3.8.

Діаграма послідовності показує учасників взаємодії і послідовність повідомлень, якими вони обмінюються [18]-[19].

Діаграма послідовності зображує виключно ті об'єкти що безпосередньо беруть участь у взаємодії і не показуються можливі статичні асоціації з іншими об'єктами. Для діаграм послідовності ключовим моментом є саме динаміка взаємодії об'єктів у часі.

Діаграми послідовності для системи, що проектується, приведені на рисунках 3.9 та 3.10.

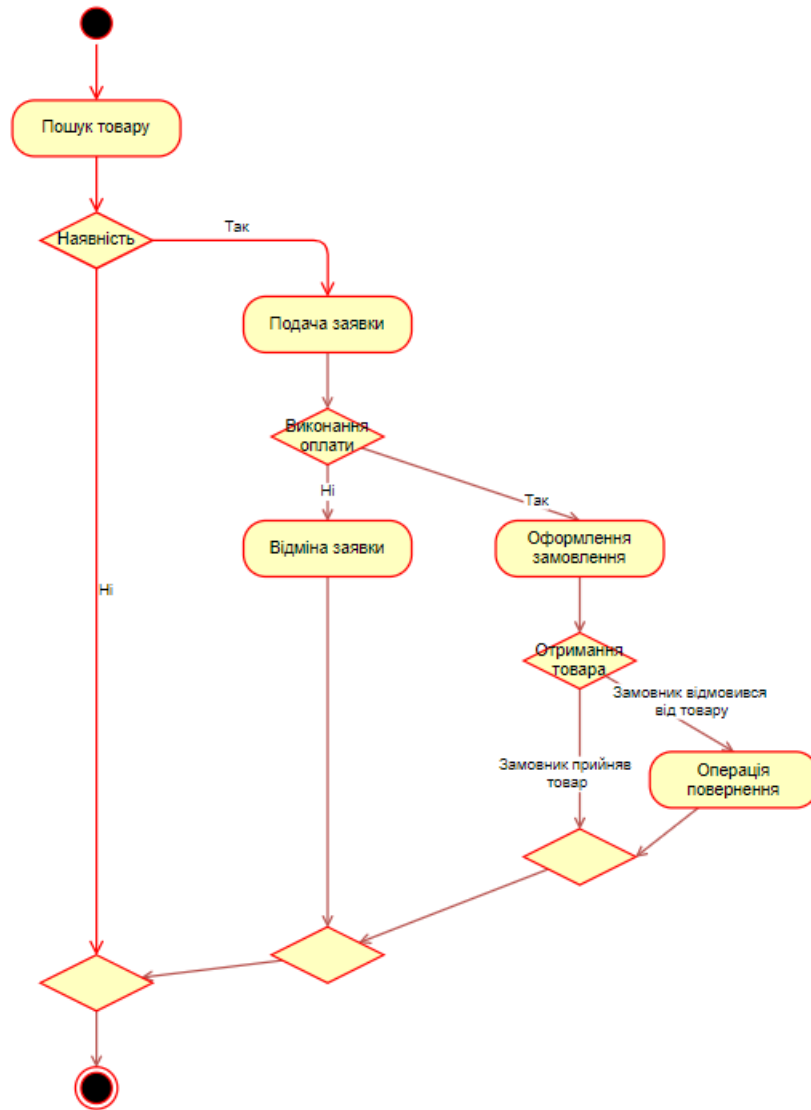


Рисунок 3.6 – Діаграма діяльності придбання товару

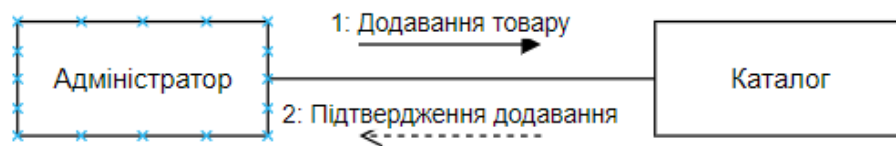


Рисунок 3.7 – Діаграма кооперації «Адміністратор - Каталог»

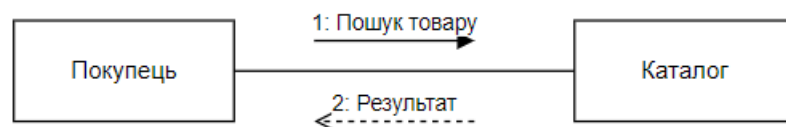


Рисунок 3.8 – Діаграма кооперації «Покупець - Каталог»

Діаграми послідовності для системи, що проектується, приведені на рисунках 3.9 та 3.10.

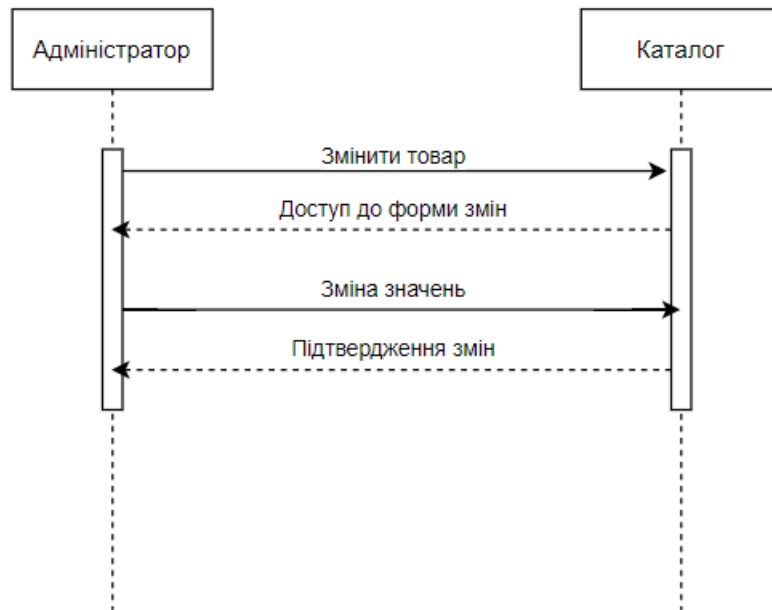


Рисунок 3.9 – Діаграма послідовності «Адміністратор – Зміна товару»

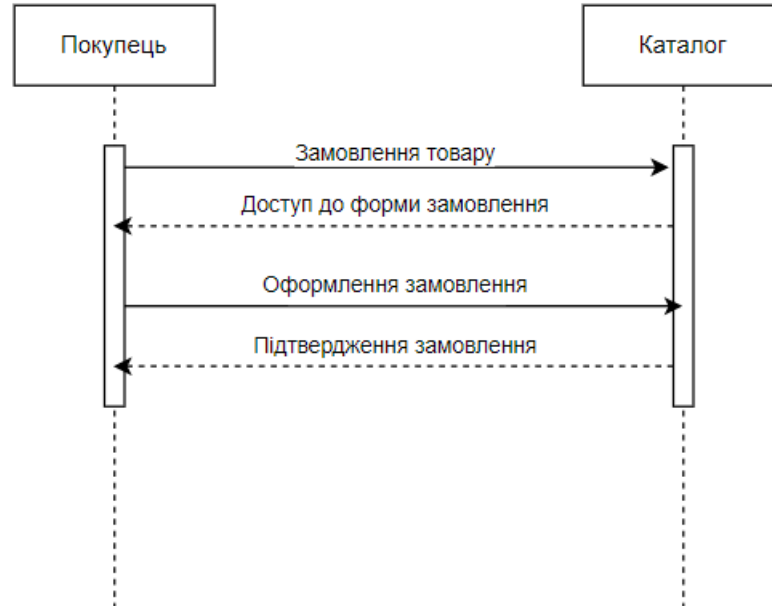


Рисунок 3.10 – Діаграма послідовності «Покупець – Замовлення товару»

3.2 Висновки за розділом 3

Для програмної реалізації музичної онлайн-крамниці було проаналізовано сучасні засоби для веб-розробки на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної дипломної кваліфікаційної роботи бакалавра. Було виявлені необхідні засоби. Отже, для розробки практичної частини роботи обрано:

- ① мову програмування PHP;
- ↑ ① фреймворк Laravel;
- ↓ ① патерн архітектури MVC.

Проаналізувавши обрану клієнт-серверну архітектуру можна зробити висновок: застосування, що розроблюється є надійним, зручним у використанні, має відкриту структуру, що відкриває широкі можливості по адаптації та масштабуванню.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Опис застосування

Метою проекту є розробка корпоративного сайту для музичного інтернет-магазину «Music Hall». Система призначена для залучення якомога більше клієнтів до музичного інтернет-магазину, що призводить до збільшення активів підприємства.

Система необхідна для збільшення кількості клієнтів музичного інтернет-магазину «Music Hall» та обсягу продажу товарів, а також завдяки можливості. Система призначення для можливості перегляду асортименту та за необхідністю швидко здійснювати замовлення необхідного товару, з урахуванням всіх особливостей щодо форми та структури товару, та щодо здійснення доставки до пункту призначення.

Програма має виконувати наступні функції:

- прийом та обробка запитів користувачів;
- можливість перегляду каталогу товарів для користувача (з можливістю фільтрації та користувацьким кошиком);
- надсилання користувачу підтвердження замовлення по електронній пошті;
- надавати змогу змінювати, додавати та видаляти інформацію про товар;
- забезпечувати інтерфейс з модулем оплати;
- видавати звіти на вимогу службових осіб, які працюють у компанії.

4.2 Умови виконання програми

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини застосунку:

- серверна ОС;

- веб-сервер XAMPP;
- інтерпретатор PHP5 та PHP7;
- СКБД MySQL-сервер.

Мінімальні вимоги до клієнтської частини:

- ПК із встановленою ОС;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB для безперебійної роботи;
- підключення до мережі Internet;
- встановлений веб-браузер останньої версії.

4.3 Інструкція по експлуатації програми

4.3.1 Звернення до програми

Для звернення до системи необхідно запустити сервер за допомогою іконки Start XAMPP, відкрити браузер, ввести адресу сайту та за необхідністю зареєструватися або авторизуватися на головній сторінці. Після відкриття доступу до системи, можливо переходити за посиланнями на головній сторінці веб-застосунку для виконання відповідних функцій.

4.3.2 Вхідні й вихідні дані

Програма не має конкретних вхідних даних для обробки, і ключем до доступу до роботи з нею є комбінація з логіна і пароля користувача чи адміністратора.

Вихідною інформацією є дані, що хоче отримати користувач від системи (наприклад, власний профіль, список покупок, результати пошуку та ін..), систематизована в таблицях бази даних з можливістю фільтрування за допомогою SQL-запитів для представлення користувачеві в табличній формі на сторінках веб-застосунку.

4.4 Методика та результати тестування

4.4.1 Тестування функціональності

На рисунку 4.1-4.4 зображено основні сторінки веб-зстосунку та виконання функцій на них.

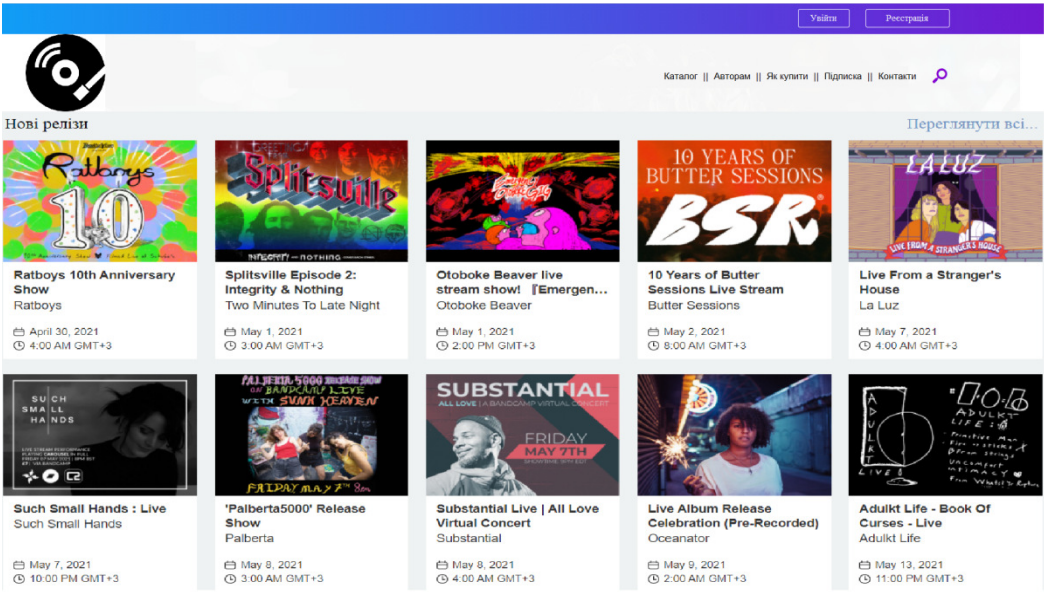


Рисунок 4.1 – Головна сторінка сайту

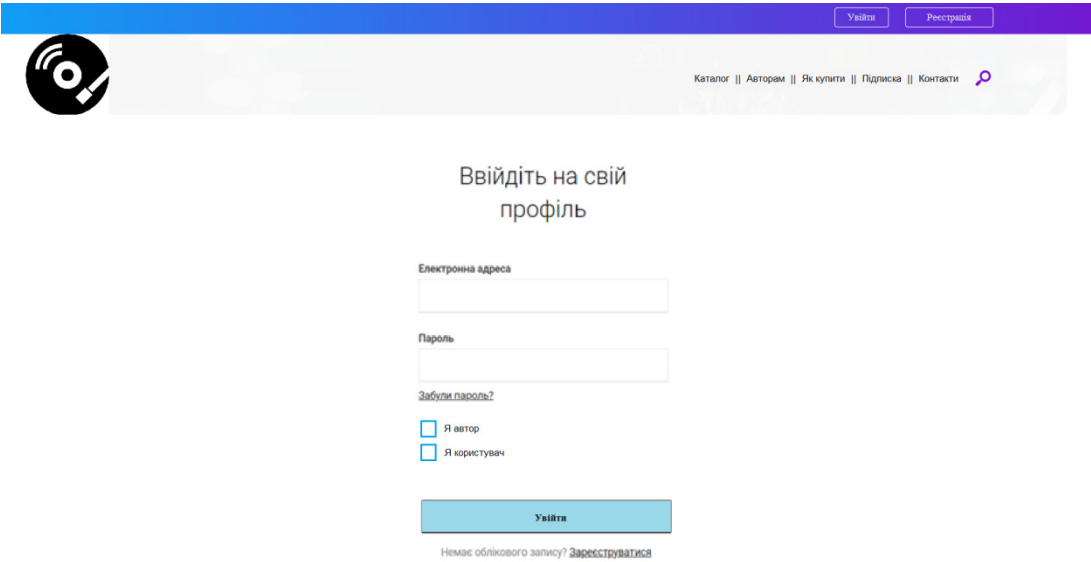


Рисунок 4.2 – Авторизація користувача

Інтерфейс веб-застосунку музичної онлайн-крамниці представлений на рисунках 4.1- 4.4. Розташування пунктів меню, кнопок, полів для вводу і т.п. найбільш оптимально для швидкої і ефективної роботи з даними веб-застосунком. Запускаючи даний застосунок, користувач потрапляє на головне вікно сайту (рис. 4.1), натискаючи один з пунктів меню є можливість переглянути запропоновані музичні композиції.

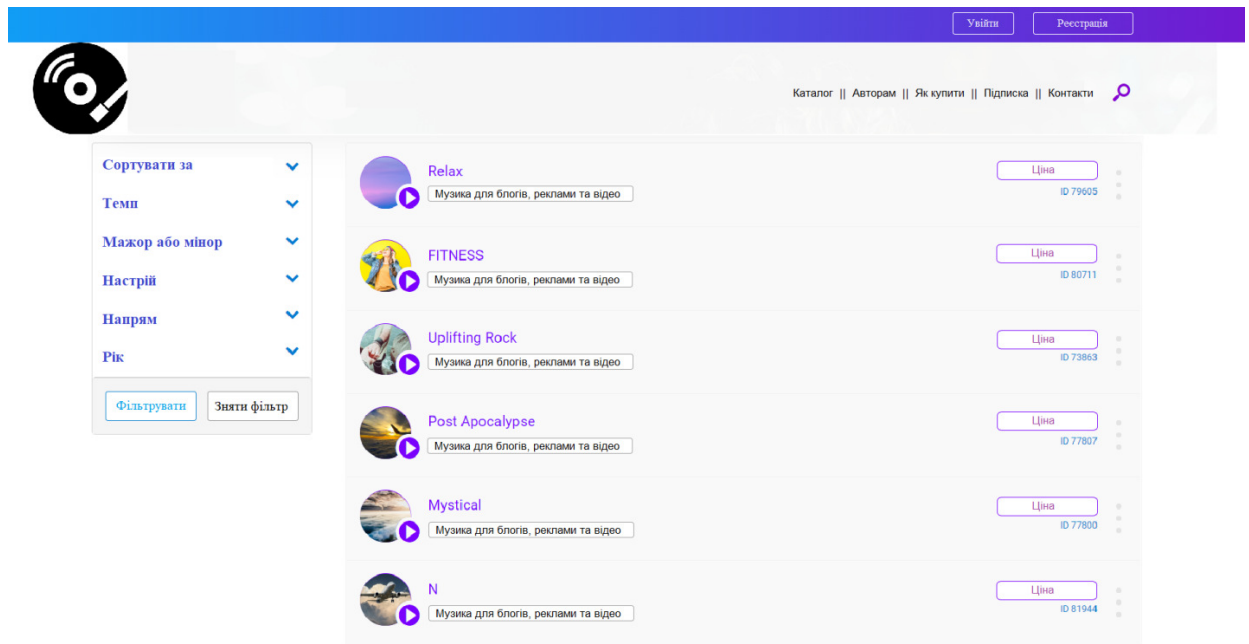


Рисунок 4.3 – Каталог треків

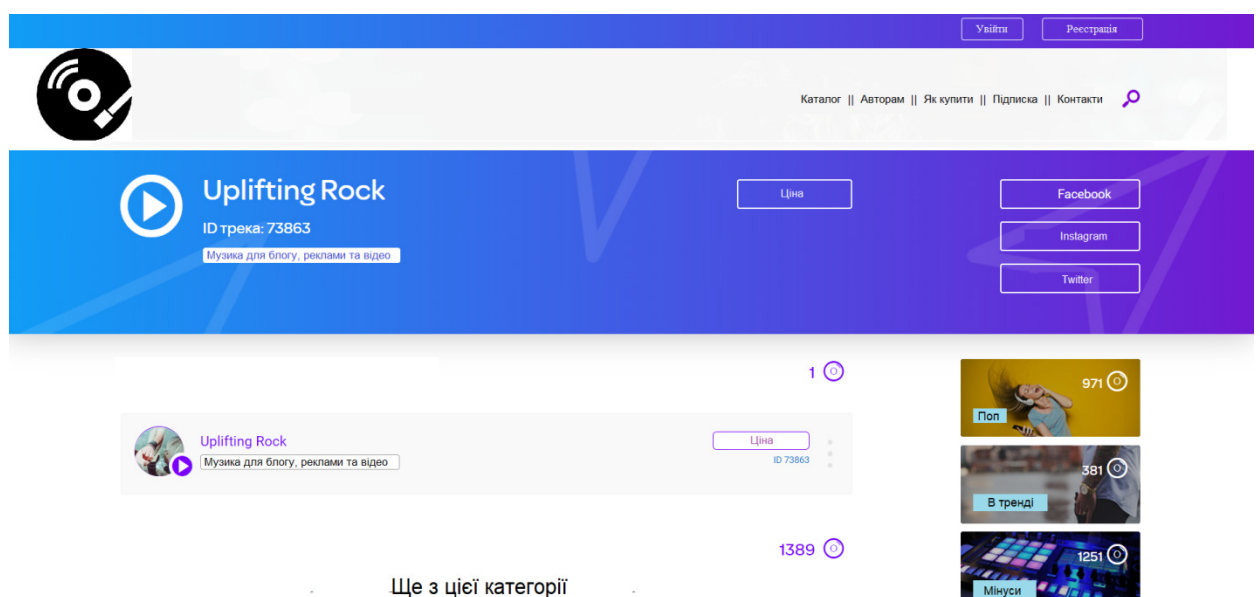


Рисунок 4.4 – Демо-версія треку

4.4.2 Тестування сумісності

Browsershots дозволяє тестувати сайти не тільки в Firefox, Opera, IE и Safari, а й в екзотичних браузерях (Dillo, Eipiphany, Flock, Galeon, Konqueror, Seamonkey) [20]. Також наявна можливість встановлення роздільної здатності, глибини кольору та підтримки JavaScript та Flash [20].

На рисунку 4.5 наведений знімок екрану роботи з Browsershots.

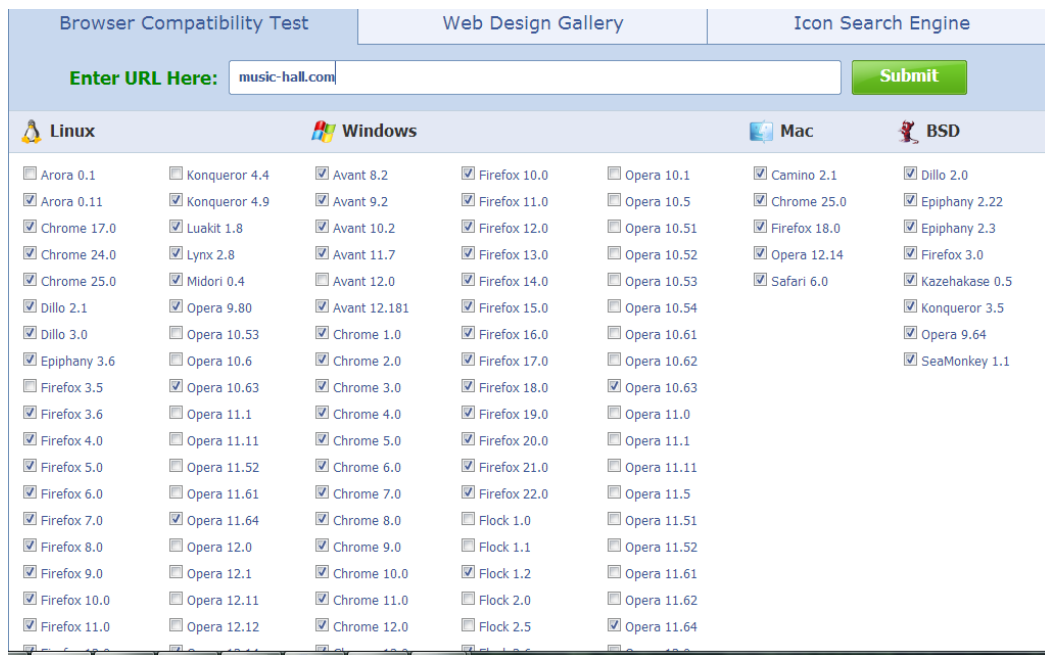


Рисунок 4.5 – Робота з Browsershots

4.4.3 Тестування ергономічності

Програмне забезпечення TechSmith Morae – це зручне рішення для перевірки зручності використання ПЗ, веб-сайтів, мобільних пристроїв та обладнання, роботи з фокус-групами, інтерв'ювання, етнографічних та інших досліджень, вивчення кваліфікації користувачів [21]. Досить відкрити TechSmith Morae і вибрати з вбудованих шаблонів спосіб дослідження.

TechSmith Morae складається з трьох програмних продуктів: Recorder, Observer і Manager, і пов'язує воєдино всі дані, які зберігаються в цифровому форматі в одному місці і індексуються єдиною монтажною лінійкою для

негайного збору та аналізу інформації. Морае надає всі необхідні засоби для перегляду, аналізу, складання графіків і надання результатів всього одним кліком [21].

На рисунку 4.6 наведений результати тестування в Морае 3.

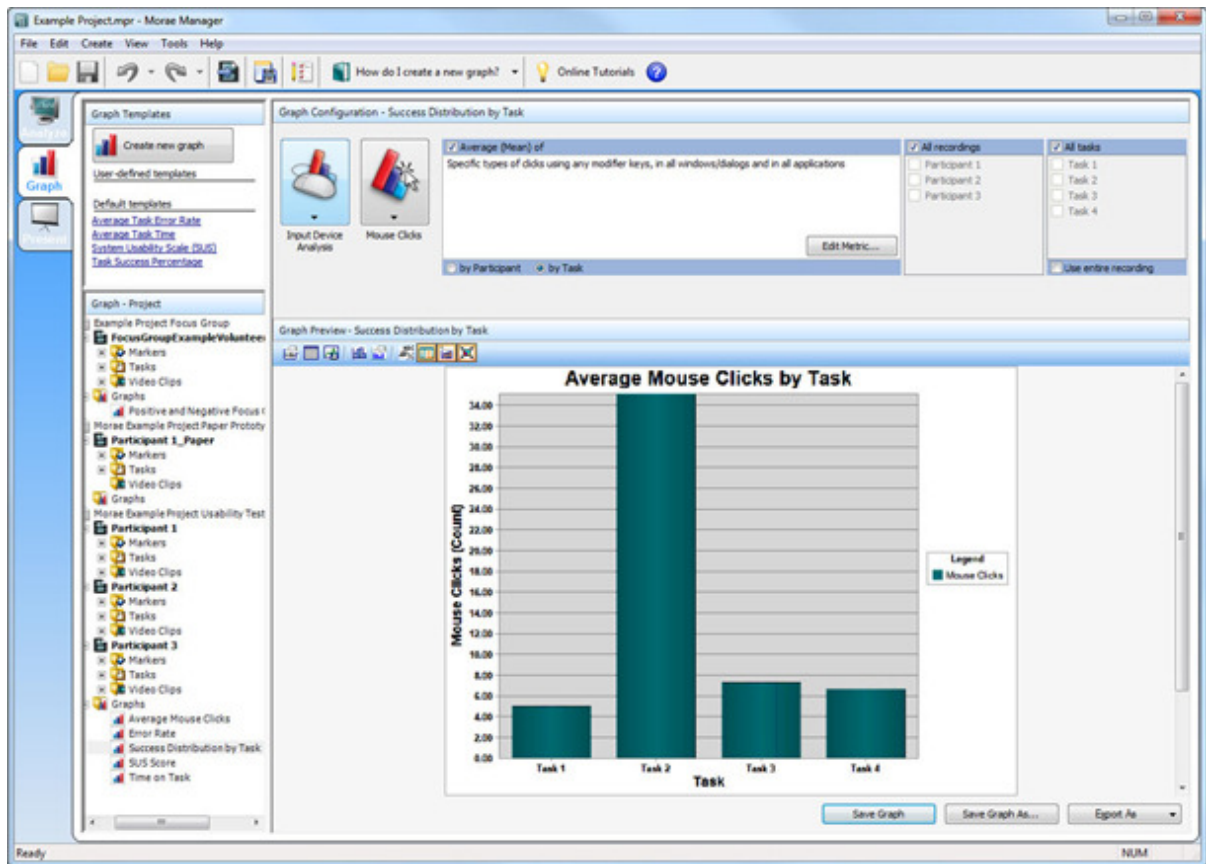


Рисунок 4.6 – Тестування в Морае 3

ВИСНОВКИ

В ході виконання даної дипломної кваліфікаційної роботи бакалавра було проведено огляд та аналіз методів розробки та підтримки систем для онлайн-продажу цифрових товарів. Відібрані матеріали були систематизовані, проаналізовані, узагальнені та надані у першому розділі пояснювальної записки.

Для цього було виконано наступні завдання:

- розглянуто проблемну область;
- проаналізовано особливості продажу цифрових товарів через мережу Інтернет;
- виконано проєктування веб-застосунку;
- розроблено веб-застосунок;
- розроблено сценарій та проведено тестування.

Для програмної реалізації музичної онлайн-крамниці у вигляді веб-застосунку було проаналізовано сучасні засоби для проєктування та веб-застосунків на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної дипломної кваліфікаційної роботи бакалавра. Було виявлені необхідні засоби. Отже, для розробки практичної було використано: мову програмування PHP, фреймворк Laravel; СКБД – MySQL для проєктування та розробки бази даних з метою централізованого збереження інформації; веб-сервер XAMPP.

В результаті виконання завдання було отримано веб-застосунок, що відповідає всім вимогам представленим в технічному завданні, зручний для використання та виконує всі необхідні функції, для інтернет-продажу цифрових товарів. Застосунок виконує наступні функції:

- реєстрація на сайті;
- авторизація на сайті;
- перегляд каталогу (фонотеки);
- додавання композиції до кошику;

- видалення композиції з кошику;
- вилучення визначеного користувача з бази даних системи;
- редагування сайту адміністратором;
- оформлення покупки.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам представленим в технічному завданні, зручна для використання та виконує всі необхідні функції.

Усі завдання випускної дипломної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Пурський О. І. Емпіричне визначення функції рекламних витрат в електронній торгівлі [Текст] / О. І. Пурський, І. О. Мороз, Д. П. Мазоха // Бізнес Інформ. – 2017. – № 6. – С. 268-274.
2. Christina I. D. Digital marketing strategy in promoting product [Текст] / I. D. Christina, Fenni Fenni, D. Roselina // Менеджмент та підприємництво: тренди розвитку. – 2019. – Вип. 4. – Р. 58-66.
3. Doroshenko V. S. The Examples of Digitalization of Foundry Production: Virtual Engineering, Digital Twin, Additive Technologies [Текст] / V. S. Doroshenko, O. V. Tokova // Control systems & computers. – 2020. – № 5. – Р. 64-69.
4. Завсєгдашня І. В. Інформаційна система управління товарними запасами для умов малого бізнесу у сфері e-commerce [Текст] / І. В. Завсєгдашня, С. А. Рубан, О. В. Пилипенко, О. О. Завсєгдашня, І. О. Филипова // Гірничий вісник. – 2020. – Вип. 108. – С. 44-50.
5. Hliebova A. The modern models of e-commerce organization: theoretical aspect [Текст] / A. Hliebova, A. Zozulia // Економіка і регіон. – 2020. – № 1. – С. 144-150.
6. Gevorgyan L. Big data in e-commerce [Текст] / L. Gevorgyan // Науковий вісник [Одеського національного економічного університету]. – 2017. – № 6. – С. 25-32.
7. Сервіс «Bandcamp» / [Electronic resource]. – Access mode: <https://bandcamp.com/>
8. Сервіс «Soundcloud» / [Electronic resource]. – Access mode: <https://soundcloud.com/discover>
9. Офіційний сайт для завантаження застосунку «iTunes Store» / [Електрон. ресурс]. – Режим доступу: <https://www.apple.com/ua/itunes/>

10. Матвієнко О. Вимоги до функціональних можливостей програмних засобів (для бібліотечних процесів) [Текст] / О. Матвієнко // Бібліотечний вісник. – 1996. – № 4. – С. 8-9.
11. Kruglik V. Information system of software distribution [Текст] / V. Kruglik // Інформ. технології в освіті : зб. наук. пр. – 2011. – Вип. 9. – С. 169-174.
12. Meloni J.C. Sams teach yourself PHP, MySQL® and Apache [Text] : all in one / Julie C. Meloni. – 5th ed. – Indianapolis, Indiana : Sams, 2012. – XVI, 653 p.
13. Пономаренко Н.Н. ВЕБ-программирование: HTML, CSS, JavaScript, jQuery, PHP, MySQL [Текст] : учеб. пособие / Н. Н. Пономаренко ; Нац. аэрокосм. ун-т им. Н. Е. Жуковского "Харьк. авиац. ин-т". – Харьков : ХАИ, 2014. – 143 с.
14. Шолом П. С. Електронний навчальний підручник на базі PHP Framework Yii2 [Текст] / П. С. Шолом, А. Ю. Котвицька, В. Ф. Самарчук // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2016. – № 23. – С. 112-116.
15. Laravel - The PHP Framework For Web Artisans / [Electronic resource]. – Access mode: <https://laravel.com/>
16. Багнюк Н. В. Аналіз та порівняння можливостей розробки веб-ресурсів засобами PHP-фреймворку Laravel і CMS WordPress [Текст] / Н. В. Багнюк, В. М. Мельник, О. В. Казмірчук // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2017. – № 26. – С. 10-15.
17. Щербакан Д. В. Использование шаблона проектирования MVC для создания ВЕБ – приложений [Текст] / Д. В. Щербакан // Автоматизация технологических и бизнес-процессов. – 2014. – № 2. – С. 85-88.
18. Императивное программирование и объектно-ориентированное моделирование: Java, UML, OCL [Текст] : учеб. пособие для студентов высш. учеб. заведений / А. Ф. Верлань [и др.]. – Одесса : Экология, 2013. – 430 с.

19. Постіл С. Д. UML. Уніфікована мова моделювання інформаційних систем [Текст] : навч. посіб. / С. Д. Постіл ; Ун-т держ. фіск. служби України. – Ірпінь : Ун-т держ. фіск. служби України, 2019. – 321 с.

20. Browsershots / [Electronic resource]. – Access mode: <http://browsershots.org/>

21. TechSmith Morae / [Electronic resource]. – Access mode: <https://www.techsmith.com/tutorial-morae.html>

ДОДАТОК А
Технічне завдання

Вступ

Метою проєкту є розробка корпоративного сайту для музичної онлайн-крамниці «Music Hall». Система призначена для залучення якомога більше клієнтів до музичного інтернет-магазину, що призводить до збільшення активів підприємства.

A.1 Підстава для розробки

Підставою для розробки слугує завдання на випускну роботу на тему «Програмна реалізація музичної онлайн-крамниці», затверджене наказом Національного університету «Запорізька політехніка» № 103 від 30 березня 2021 р.

A.2 Призначення розробки

Система необхідна для збільшення кількості клієнтів музичного інтернет-магазину «Music Hall» та обсягу продажу товарів, а також завдяки можливості. Система призначення для можливості перегляду асортименту та за необхідністю швидко здійснювати замовлення необхідного товару, з урахуванням всіх особливостей щодо форми та структури товару, та щодо здійснення доставки до пункту призначення.

Програма має виконувати наступні функції:

- прийом та обробка запитів користувачів;
- можливість перегляду каталогу товарів для користувача (з можливістю фільтрації та користувацьким кошиком);
- надсилання користувачу підтвердження замовлення по електронній пошті;
- надавати змогу змінювати, додавати та видаляти інформацію про товар;

- видавати звіти на вимогу службових осіб, які працюють у компанії.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Функціональні вимоги

- система повинна приймати та обробляти запити користувачів;
- система повинна виводити каталог товарів користувачу;
- система повинна по електронній пошті відсилати користувачу підтвердження замовлення;
- система повинна надавати змогу змінювати, додавати та видаляти інформацію про товар;
- система повинна забезпечувати інтерфейс з модулем оплати;
- система повинна видавати звіти на вимогу службових осіб, які працюють у компанії.

A.3.2 Вимоги до надійності

Програма повинна коректно реагувати на наступні виняткові ситуації: не вірність введення логіну або паролю.

База даних з даними компанії повинна буди захищеною паролем, який відомий лише певній групі осіб. Повинні бути збережені наступні властивості інформації: конфіденційність, цілісність, доступність. Пароль повинен шифруватися за допомогою технології S-DES.

A.3.3 Умови експлуатації

Програма може зберігатися та переноситися на будь-яких існуючих носіях інформації.

Експлуатація програми здійснюється згідно експлуатаційної документації на розроблену програму, яка відповідає стандартам і містить інформацію, необхідну для її освоєння та експлуатації.

A.3.4 Вимоги до складу та параметрів технічний засобів

Мінімальні системні вимоги та програмне забезпечення , що необхідні для функціонування серверної частини застосунку:

- серверна ОС;
- веб-сервер XAMPP;
- інтерпретатор PHP5 та PHP7;
- СКБД MySQL-сервер.

Мінімальні вимоги до клієнтської частини:

- ПК із встановленою ОС;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB для безперебійної роботи;
- підключення до мережі Internet;
- встановлений веб-браузер останньої версії.

A.3.5 Вимоги до маркування і пакування

Програма може бути записана на переносний носій або розповсюджуватися через хмарні сховища. На впакуванні повинна бути назва програми – «Музична онлайн-карамниця».

A.3.6 Вимоги до транспортування й зберігання

Вимоги до транспортування й зберігання аналогічні вимогам, пропонованим до носія, на якому записана програма.

A.3.7 Вимоги до програмної документації

Програмний продукт повинен поставлятися у наступному складі:

- технічне завдання;
- опис програми;
- текст програми.

A.4 Стадії та етапи розробки

Виділяють такі етапи в розробці програмного забезпечення:

→ ① бирання та аналіз вимог. Проводиться збір вимог до програмного забезпечення, їх систематизація, документування, аналіз, виявлення протиріч, неповноти, рішення конфліктів в процесі розробки програмного забезпечення;

↑ ① розробка ТЗ. На основі раніше зібраних вимог формується вихідний документ (ТЗ) на проєктування програмного забезпечення. Описується всі вимоги до створюваного продукту;

↓ ① створення логотипу та дизайн основних сторінок;

↖ ① налаштування БД;

↗ ① розробка структури. Дозволяє оцінити інформативність та компоновку елементів системи на початковому етапі розробки;

↘ ① підключення адміністративного інтерфейсу. Має містити повне управління настройками системи, окремих модулів та контентом сайту;

↙ ① робота над програмним кодом. Розробка основного функціоналу системи;

← ① наповнення та тестування. Професійні тестувальники знаходять «баги» у системі та фіксують їх. Заповнюється базовий «контент», щоб сайт мав привабливіший вигляд для перших користувачів;

➔ ① розміщення та відкритий доступ, технічна підтримка. Соціальна мережа розміщується на сайті унтернет-магазину та стає доступною для всіх користувачів;

↓ ① оформлення документації.

A.5 Порядок контролю та приймання системи

Приймання системи здійснюється відповідно до складу та змісту робіт, наведених у Технічному завданні, і відповідно до умов контракту.

Звіт про проведену роботу направляється Замовнику на паперовому носії в 2-х екземплярах, підписаних належним чином, а також на електронному носії.

Вимоги до оформлення проміжних звітів і підсумкового звіту узгоджуються з Виконавцем при підписанні контракту.

Одночасно з підсумковим звітом про виконання робіт Виконавець надає Замовнику підписаний зі сторони Виконавця Акт здачі-приймання виконаної роботи.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Назва програмного продукту – «Музична онлайн-крамниця».

ПЗ було розроблено з використанням мови програмування PHP та фреймворк Laravel.

Б.2 Функціональне призначення

Функціональні вимоги

- система повинна приймати та обробляти запити користувачів;
- система повинна виводити каталог товарів користувачу;
- система повинна по електронній пошті відсилати користувачу підтвердження замовлення;
- система повинна надавати змогу змінювати, додавати та видаляти інформацію про товар;
- система повинна забезпечувати інтерфейс з модулем оплати;
- система повинна видавати звіти на вимогу службових осіб, які працюють у компанії.

Б.3 Опис логічної структури

Система повинна бути побудована за технологією «клієнт-сервер» та бути розрахована на велику кількість користувачів. Загальна схема представлена на рисунку Б.1.

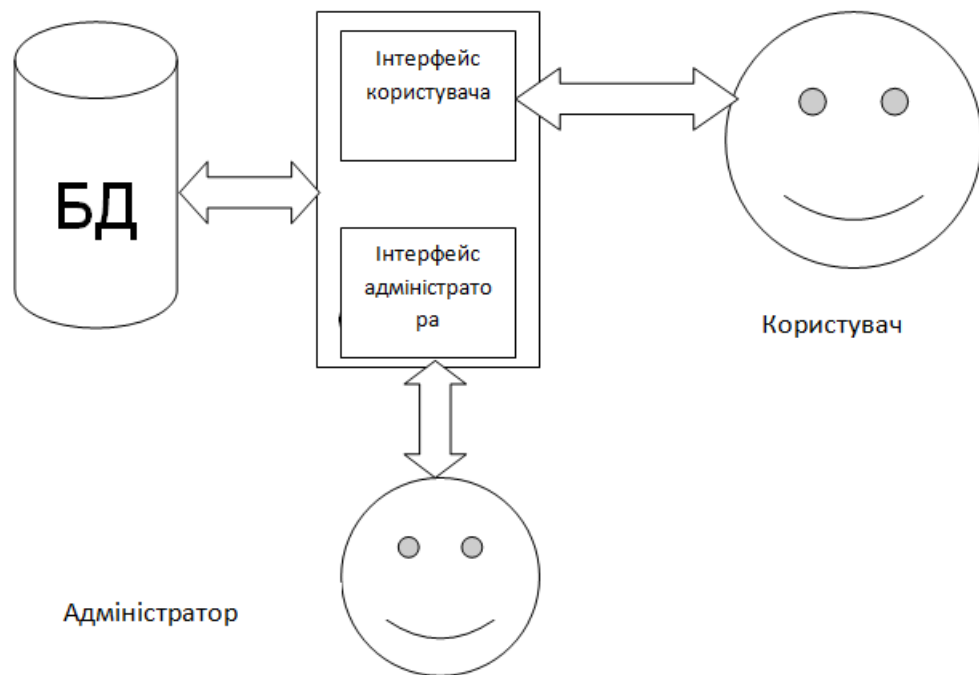


Рисунок Б.1 – Схема системи

Б.4 Використані технічні засоби

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини застосунку:

- серверна ОС;
- веб-сервер XAMPP;
- інтерпретатор PHP5 та PHP7;
- СКБД MySQL-сервер.

Мінімальні вимоги до клієнтської частини:

- ПК із встановленою ОС;
- процесор із тактовою частотою від 1,5 GHz;
- обсяг оперативної пам'яті – не менше 3GB для безперебійної роботи;
- підключення до мережі Internet;
- встановлений веб-браузер останньої версії.

Б.5 Виклик і завантаження

На наданому для хостингу веб-вузла на MySQL-сервері необхідно створити БД.

Необхідно отримати логін і пароль користувача БД з правами адміністратора, дисковий простір для БД, вказати адресу головної сторінки графічного інтерфейсу користувача `index.php`.

Використовуючи phpMyAdmin «Експорт», вибравши шлях до файлу дампу БД і вказавши кодування, перенести БД на новий сервер, або скопіювати папку з базою даних у директорію розміщення даних баз даних MySQL-сервера.

Для завантаження сайту і взаємодії з ним достатньо вказати в адресному рядку браузера `musichall.ua`.

Б.6 Вхідні та вихідні дані

Вхідними даними є інформація про користувача: логін та пароль людини, що запрошує доступ до системи. Вхідна інформація зберігається у текстовій формі в таблицях бази даних на сервері. Також до вхідної інформації можна віднести всі дані, що відносяться до певного користувача.

Вихідною інформацією є дані, що хоче отримати користувач від системи (наприклад, власний профіль, замовлення, результати пошуку та ін..), систематизована в таблицях бази даних з можливістю фільтрування за допомогою SQL-запитів для представлення користувачеві в табличній формі на сторінках Web-додатку.

ДОДАТОК В
Текст програми

```

class MConfig {
public class CheckoutController : Controller
{
    MusicStoreEntities shopDB = new
MusicStoreEntities();
    const string PromoCode = "FREE";

    //
    // GET: /Checkout/AddressAndPayment

    public ActionResult AddressAndPayment()
    {
        return View();
    }

    //
    // POST: /Checkout/AddressAndPayment

    [HttpPost]
    public ActionResult
AddressAndPayment(FormCollection values)
    {
        var order = new Order();
        TryUpdateModel(order);

        try
        {
            if (string.Equals(values["PromoCode"],
PromoCode,
StringComparison.OrdinalIgnoreCase) ==
false)
            {
                return View(order);
            }
            else
            {
                order.Username = User.Identity.Name;
                order.OrderDate = DateTime.Now;

                //Save Order
                shopDB.Orders.Add(order);
                shopDB.SaveChanges();

                //Process the order
                var Crt =
SHCrt.GetCrt(this.HttpContext);
                Crt.CreateOrder(order);

                return RedirectToAction("Complete",
                    new { id = order.OrderId });
            }
        }
    }
}

```

```

        catch
        {
            //Invalid - redisplay with errors
            return View(order);
        }
    }

    //
    // GET: /Checkout/Complete

    public ActionResult Complete(int id)
    {
        // Validate customer owns this order
        bool isValid = shopDB.Orders.Any(
            o => o.OrderId == id &&
            o.Username == User.Identity.Name);

        if (isValid)
        {
            return View(id);
        }
        else
        {
            return View("Error");
        }
    }
}

public class SHCrtController : Controller
{
    MusicStoreEntities shopDB = new
    MusicStoreEntities();

    //
    // GET: /SHCrt/

    public ActionResult Index()
    {
        var Crt = SHCrt.GetCrt(this.HttpContext);

        // Set up our ViewModel
        var viewModel = new SHCrtViewModel
        {
            CrtItems = Crt.GetCrtItems(),
            CrtTotal = Crt.GetTotal()
        };

        // Return the view
        return View(viewModel);
    }

    //
    // GET: /Store/AddToCrt/5

```

```

public ActionResult AddToCrt(int id)
{
    // Retrieve the album from the database
    var addedAlbum = shopDB.Albums
        .Single(album => album.AlbumId == id);

    // Add it to the shopping Crt
    var Crt = SHCrt.GetCrt(this.HttpContext);

    Crt.AddToCrt(addedAlbum);

    // Go back to the main store page for more
shopping    return RedirectToAction("Index");
}

//
// AJAX: /SHCrt/RemoveFromCrt/5

[HttpPost]
public ActionResult RemoveFromCrt(int id)
{
    // Remove the item from the Crt
    var Crt = SHCrt.GetCrt(this.HttpContext);

    // Get the name of the album to display
confirmation    string albumName = shopDB.Crts
                    .Single(item => item.RecordId ==
id).Album.Title;

    // Remove from Crt
    int itemCount = Crt.RemoveFromCrt(id);

    // Display the confirmation message
    var results = new SHCrtRemoveViewModel
    {
        Message = Server.HtmlEncode(albumName) +
            " has been removed from your shopping
Crt.",

        CrtTotal = Crt.GetTotal(),
        CrtCount = Crt.GetCount(),
        ItemCount = itemCount,
        DeleteId = id
    };

    return Json(results);
}

//
// GET: /SHCrt/CrtSummary

```

```

[ChildActionOnly]
public ActionResult CrtSummary()
{
    var Crt = SHCrt.GetCrt(this.HttpContext);

    ViewData["CrtCount"] = Crt.GetCount();

    return PartialView("CrtSummary");
}
}
namespace MvcMusicStore.Models
{
    public partial class SHCrt
    {
        MusicStoreEntities shopDB = new
MusicStoreEntities();

        string SHCrtId { get; set; }

        public const string CrtSessKey = "CrtId";

        public static SHCrt GetCrt(HttpContextBase context)
        {
            var Crt = new SHCrt();
            Crt.SHCrtId = Crt.GetCrtId(context);
            return Crt;
        }

        // Helper method to simplify shopping Crt calls
        public static SHCrt GetCrt(Controller controller)
        {
            return GetCrt(controller.HttpContext);
        }

        public void AddToCrt(Album album)
        {
            // Get the matching Crt and album instances
            var CrtItem = shopDB.Crts.SingleOrDefault(
c => c.CrtId == SHCrtId
&& c.AlbumId == album.AlbumId);

            if (CrtItem == null)
            {
                // Create a new Crt item if no Crt item
exists

                CrtItem = new Crt
                {
                    AlbumId = album.AlbumId,
                    CrtId = SHCrtId,
                    Count = 1,
                    DateCreated = DateTime.Now

```

```

        };

        shopDB.Crts.Add(CrtItem);
    }
    else
    {
        // If the item does exist in the Crt, then
add one to the quantity
        CrtItem.Count++;
    }

    // Save changes
    shopDB.SaveChanges();
}

public int RemoveFromCrt(int id)
{
    // Get the Crt
    var CrtItem = shopDB.Crts.Single(
Crt => Crt.CrtId == SHCrtId
&& Crt.RecordId == id);

    int itemCount = 0;

    if (CrtItem != null)
    {
        if (CrtItem.Count > 1)
        {
            CrtItem.Count--;
            itemCount = CrtItem.Count;
        }
        else
        {
            shopDB.Crts.Remove(CrtItem);
        }

        // Save changes
        shopDB.SaveChanges();
    }

    return itemCount;
}

public void EmptyCrt()
{
    var CrtItems = shopDB.Crts.Where(Crt =>
Crt.CrtId == SHCrtId);

    foreach (var CrtItem in CrtItems)
    {
        shopDB.Crts.Remove(CrtItem);
    }
}

```

```

        // Save changes
        shopDB.SaveChanges();
    }

    public List<Crt> GetCrtItems()
    {
        return shopDB.Crts.Where(Crt => Crt.CrtId ==
SHCrtId).ToList();
    }

    public int GetCount()
    {
        // Get the count of each item in the Crt and
sum them up
        int? count = (from CrtItems in shopDB.Crts
                        where CrtItems.CrtId == SHCrtId
                        select
(int?)CrtItems.Count).Sum();

        // Return 0 if all entries are null
        return count ?? 0;
    }

    public decimal GetTotal()
    {
        decimal? total = (from CrtItems in shopDB.Crts
                           where CrtItems.CrtId ==
SHCrtId
                           select (int?)CrtItems.Count *
CrtItems.Album.Price).Sum();
        return total ?? decimal.Zero;
    }

    public int CreateOrder(Order order)
    {
        decimal orderTotal = 0;

        var CrtItems = GetCrtItems();

        // Iterate over the items in the Crt, adding
the order details for each
        foreach (var item in CrtItems)
        {
            var orderDetail = new OrderDetail
            {
                AlbumId = item.AlbumId,
                OrderId = order.OrderId,
                UnitPrice = item.Album.Price,
                Quantity = item.Count
            };

            // Set the order total of the shopping Crt

```

```

        orderTotal += (item.Count *
item.Album.Price);

        shopDB.OrderDetails.Add(orderDetail);
    }

    // Set the order's total to the orderTotal
count    order.Total = orderTotal;

    // Save the order
    shopDB.SaveChanges();

    // Empty the shopping Cart
    EmptyCart();

    // Return the OrderId as the confirmation
number    return order.OrderId;
}

// We're using HttpContextBase to allow access to
cookies.
public string GetCrtId(HttpContextBase context)
{
    if (context.Session[CrtSessKey] == null)
    {
        if
(!string.IsNullOrEmpty(context.User.Identity.Name))
        {
            context.Session[CrtSessKey] =
context.User.Identity.Name;
        }
        else
        {
            // Generate a new random GUID using
System.Guid class
            Guid tempCrtId = Guid.NewGuid();

            // Send tempCrtId back to client as a
cookie
            context.Session[CrtSessKey] =
tempCrtId.ToString();
        }
    }

    return context.Session[CrtSessKey].ToString();
}

public void MigrateCrt(string userName)
{

```

```

        var SHCrt = shopDB.Crts.Where(c => c.CrtId ==
SHCrtId);

        foreach (Crt item in SHCrt)
        {
            item.CrtId = userName;
        }
        shopDB.SaveChanges();
    }
}
{
    if (input.Length <= length) {
        @input
    } else {
        @input.Substring(0, length)<text>...</text>
    }
}

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table>
    <tr>
        <th>
            Genre
        </th>
        <th>
            Artist
        </th>
        <th>
            Title
        </th>
        <th>
            Price
        </th>
        <th></th>
    </tr>

    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Genre.Name)
            </td>
            <td>
                @Truncate(item.Artist.Name, 25)
            </td>

```

```

        <td>
            @Truncate(item.Title, 25)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new {
id=item.AlbumId }) |
            @Html.ActionLink("Details", "Details", new {
id=item.AlbumId }) |
            @Html.ActionLink("Delete", "Delete", new {
id=item.AlbumId })
        </td>
    </tr>
}
<h2>Details</h2>

<fieldset>
    <legend>Album</legend>

    <div class="display-label">Genre</div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Genre.Name)
    </div>

    <div class="display-label">Artist</div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Artist.Name)
    </div>

    <div class="display-label">Title</div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Title)
    </div>

    <div class="display-label">Price</div>
    <div class="display-field">
        @Html.DisplayFor(model => model.Price)
    </div>

    <div class="display-label">AlbumArtUrl</div>
    <div class="display-field">
        @Html.DisplayFor(model => model.AlbumArtUrl)
    </div>
</fieldset>
<p>
    @Html.ActionLink("Edit", "Edit", new { id=Model.AlbumId
}) |
    @Html.ActionLink("Back to List", "Index")
</p>
</table>

```

ДОДАТОК Д
Слайди презентації

Національний Університет «Запорізька політехніка»

Програмна реалізація музичної онлайн- крамниці

Software Implementation of an Online Music Store

Кунденко Денис Сергійович (КНТ-137)

Керівник: Голуб Т.В.

Рисунок Д.1 – Слайд 1

Зміст роботи

Об'єкт дослідження – процес розробки та функціонування онлайн-крамниць, що пропонують цифрові товари.	Предмет дослідження – методи розробки та підтримки функціонування онлайн-крамниць, що пропонують цифрові товари.	Метою роботи є програмна реалізація онлайн-крамниці, що пропонує цифрові товари на прикладі електронних продажів музичних композицій, альбомів виконавців, тощо.
--	---	---

Рисунок Д.2 – Слайд 2

Функціональні вимоги до розробки

Функціональні вимоги:

- система повинна приймати та обробляти запити користувачів;
- система повинна виводити каталог товарів користувачу;
- система повинна по електронній пошті відсилати користувачу підтвердження замовлення;
- система повинна надавати змогу змінювати, додавати та видаляти інформацію про товар;
- система повинна забезпечувати інтерфейс з модулем оплати;
- система повинна видавати звіти на вимогу службових осіб, які працюють у компанії.

3

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерій порівняння	Bandcamp	Soundcloud	iTunes	Music Hall
Додаткові вимоги до авторів	Не має	Не має	Наявні	Не має
Додаткові вимоги до аудіофайлів	Наявні	Не має	Наявні	Не має
Наявність обмежень за ціною	Не має	Є (продаж через сторонні сервіси)	Є (додаткові виплати самій платформі)	Не має
Функціонал для ознайомлення із альбомом//композицією	Наявний	Наявний	Обмежений	Наявний

4

Рисунок Д.4 – Слайд 4

Вибір мови програмування для розробки

	PHP	Python	C# (+.Net)
Швидкість роботи застосунку	+	+	+
Додаткові бібліотеки	++	+	±
Парадигми програмування	++	++	++
Спільнота та матеріали	++	+	+
Фреймворки	++	+	±

5

Рисунок Д.5 – Слайд 5

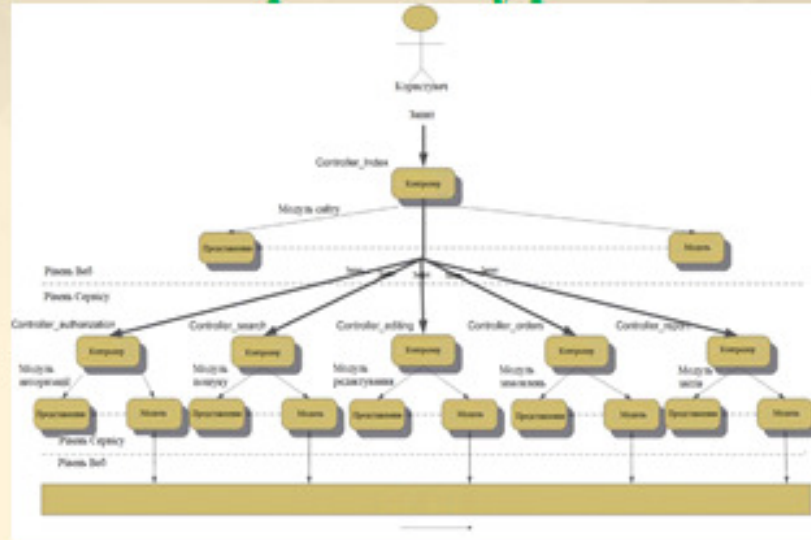
Вибір фреймворку для розробки

	Laravel	Yii2
Спільнота та простота вивчення	++	++
Оновлення та підтримка	++	-
Простота розробки представлень	±	++
Підтримка СКБД	++	++

6

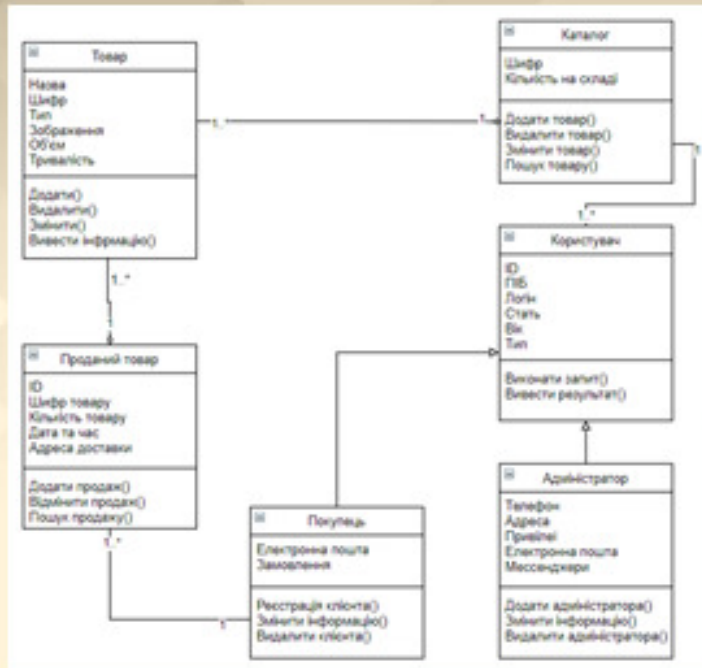
Рисунок Д.6 – Слайд 6

MVC схема проектування архітектури



7

Рисунок Д.7 – Слайд 7



8

Рисунок Д.8 – Слайд 8

Діаграма класів

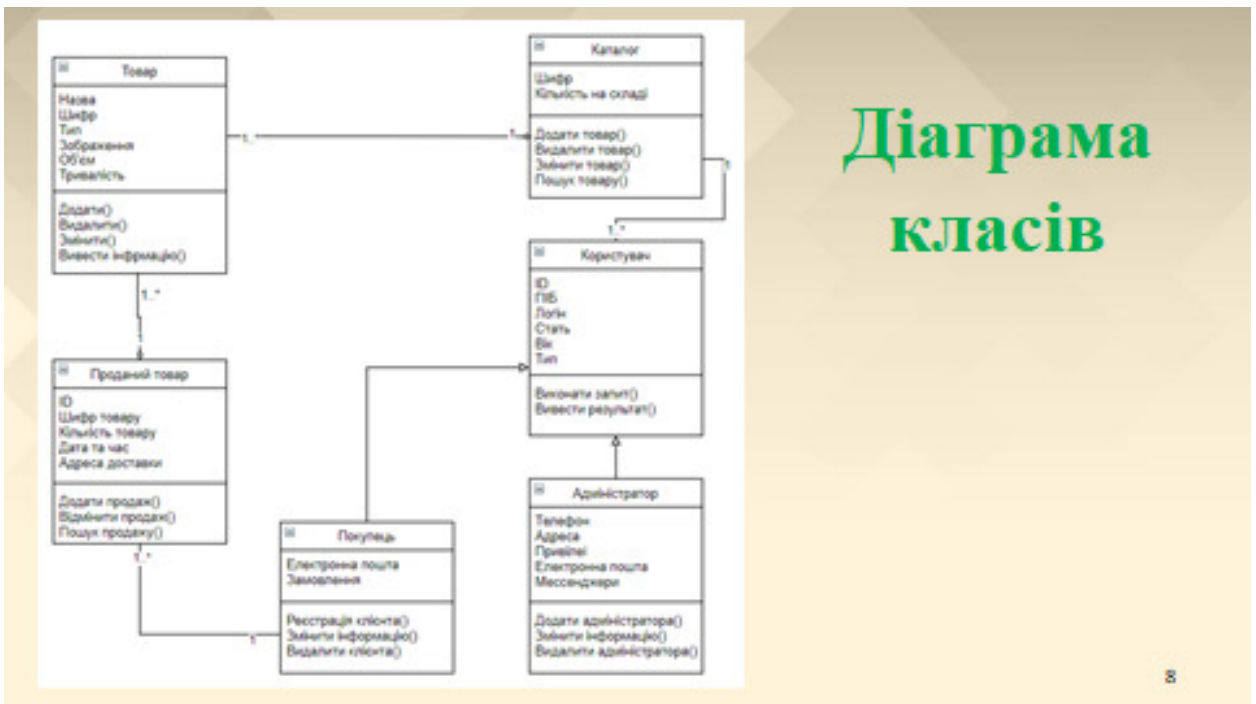


Рисунок Д.9 – Слайд 9

Загальна діаграма прецедентів

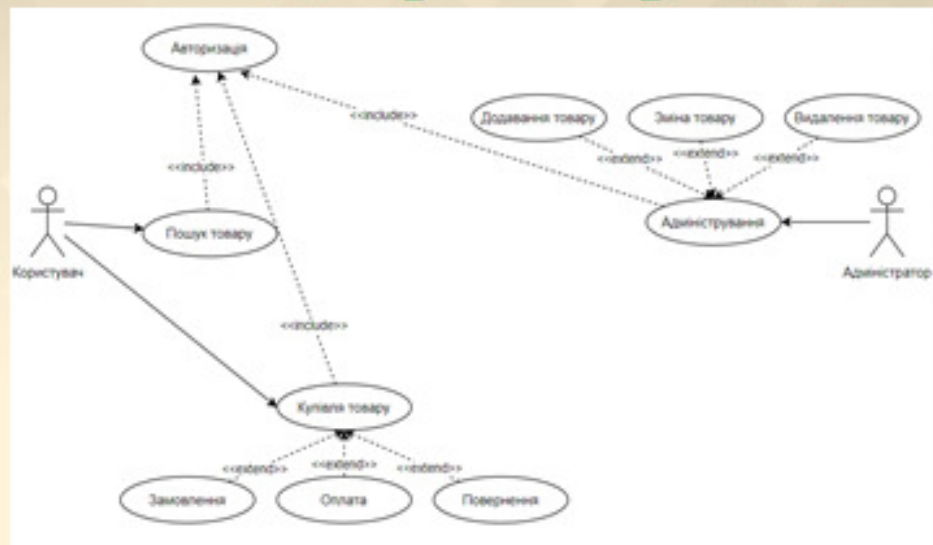


Рисунок Д.10 – Слайд 10

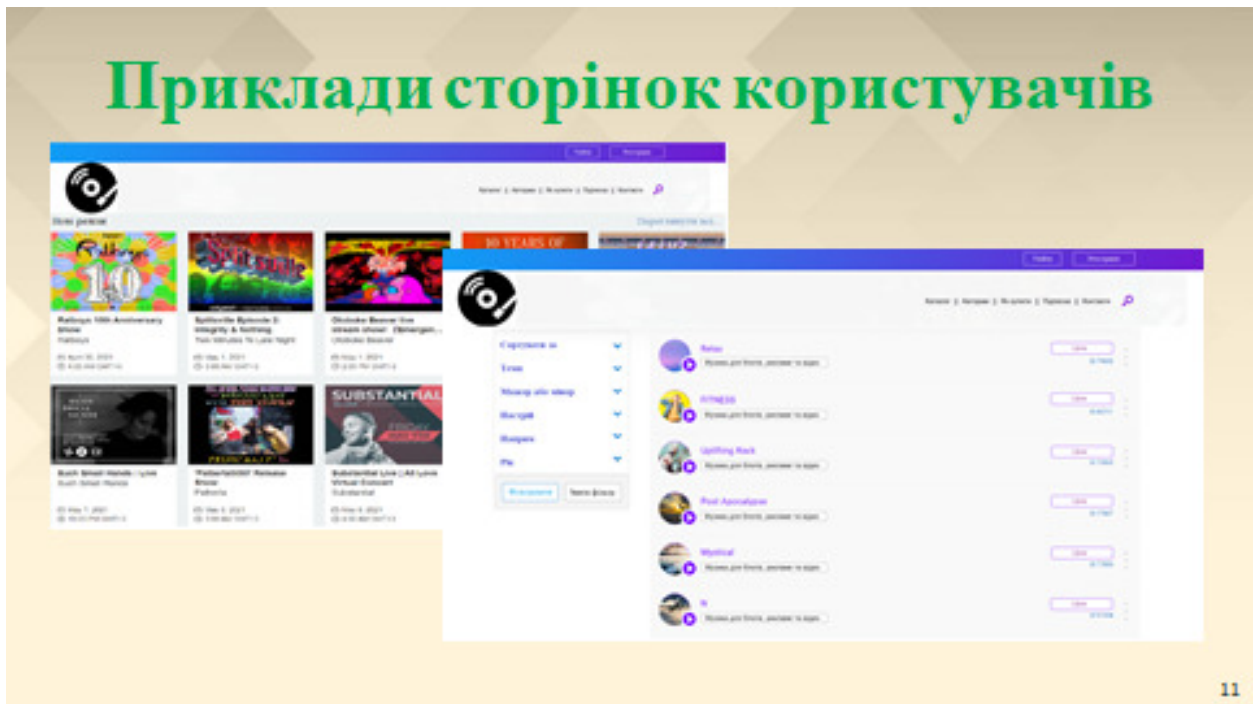


Рисунок Д.11 – Слайд 11

Висновки

В ході виконання роботи було проведено огляд та аналіз методів розробки та підтримки систем для онлайн-продажу цифрових товарів.

Для програмної реалізації було проаналізовано сучасні засоби для проектування та веб-застосунків на основі виявлених особливостей та архітектури застосунку, що розробляється в рамках даної роботи.

В результаті виконання завдання було отримано веб-застосунок, що відповідає всім вимогам представленим в технічному завданні, зручний для використання та виконує всі необхідні функції, для інтернет-продажу цифрових товарів. Застосунок виконує наступні функції:

- реєстрація та авторизація на сайті;
- перегляд каталогу (фонотеки);
- додавання композиції до кошику;
- редагування сайту адміністратором;
- оформлення покупки.

Рисунок Д.12 – Слайд 12